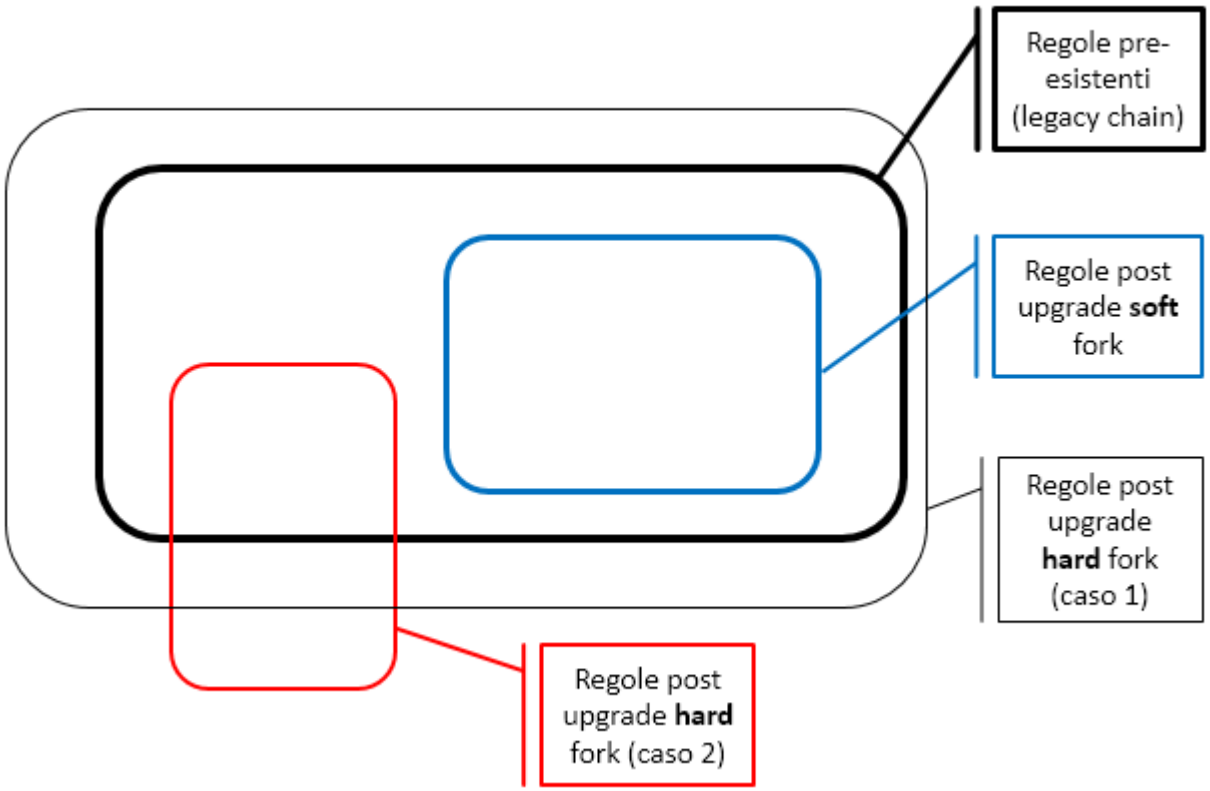
SOFT FORK, HARD FORK e SPLIT: differenze

Le definizioni di hard fork e soft fork sono varie e non sempre perfettamente coerenti. Entrambi sono un cambiamento del protocollo che potrebbe condurre a uno split della catena. L'unico modo per evitare lo split, è avere sufficiente consenso da parte della community sulle nuove regole del protocollo.

Un esempio di hard fork è la creazione di blocchi maggiori di 1mb (blocksize). In questo caso, il fork può essere visto come un rilassamento delle regole del protocollo (caso 1 nel grafico), poiché si allarga il protocollo ad alcune possibilità prima negate, ovvero l'esistenza di blocchi maggiori di 1mb. In altri casi che possiamo immaginare, l'hard fork potrebbe comportare allo stesso tempo una restrizione di alcune regole e un allargamento di altre (caso 2). Ad ogni modo è sicuramente vero che **l'hard fork rende validi blocchi di un certo tipo che erano prima invalidi**.

Definire un soft fork è più difficile, poiché ci sono definizioni non del tutto coerenti fra loro, come vedremo a breve. Secondo alcune teorie, è una modifica che comporta esclusivamente un restringimento delle regole del protocollo. Un esempio è l'introduzione al limite alla dimensione del blocco (blocksize): Fino al luglio 2010 non c'erano limitazioni al peso dei blocchi, ma nella versione 0.3.1 rc1 del client Bitcoin è stato impostato un tetto massimo a 1mb, senza una particolare discussione in merito. A quei tempi le transazioni bitcoin erano ben lontane dal possibile raggiungimento di quel tetto e non si prevedeva che quella scelta sarebbe stata al centro di un terremoto politico. Ad ogni modo, tornando alla teoria diciamo che il soft fork rende invalidi blocchi che prima potevano essere validi (tutti quelli con peso superiore a 1mb), perciò restringe le

possibilità permesse dal protocollo. Per tornare allo stadio precedente al soft fork, è sempre necessario un hard fork (nel caso in esame, che rimuova il limite al blocksize).



In caso di soft fork, a differenza di un hard fork, vecchie versioni del software riconoscono come validi i blocchi della chain generati col nuovo software. La ragione è mostrata nel grafico in modo semplice: i blocchi post soft fork sono un sotto-insieme di quelli possibili nella catena originaria (o legacy). Per questo motivo, spesso si dice che il soft fork sia retro-compatibile, anche se lo è solo in senso molto stretto. Infatti, supponendo che il traffico di transazioni fosse già stato elevato nel 2010, possiamo immaginare che se alcuni miners avessero creato blocchi di oltre 1mb dopo l'upgrade, si sarebbero potute creare due blockchain, con un effettivo split. A onor di cronaca, va specificato che ai tempi della 0.3.1 la limitazione a 1mb non era stata afatto percepita come fork, poiché nessun nodo o blocco creato da un miner sarebbe stato rifiutato: il traffico di transazioni era molto limitato e non esistevano sufficienti transazioni per

riempire i blocchi oltre a quella soglia.

Abbiamo (per ora) definito **soft fork** tramite queste caratteristiche:

(1) Una modifica che restringe le regole del protocollo, rendendo invalidi blocchi che prima erano validi

(2) Una modifica per cui vecchie versioni del software riconosceranno i blocchi e le transazioni generati da nuove versioni del software (retrocompatibilità)

there are "soft" rule changes and "hard" rule changes. "Soft" changes tighten up the rules– old software will accept all the blocks and transactions created by new software, but the opposite may not be true. "Soft" changes do not require the entire network of miners and merchants and users to upgrade or be left behind.

"Hard" changes modify the rules in a way that old, un-upgraded software consider illegal. At this point it is much, much more difficult (some might say impossible) to roll out "hard" changes, because they require every miner and merchant and user to upgrade.

(Gavin Andresen, Bitcoin Core maintainer, 2012, vedi qui)

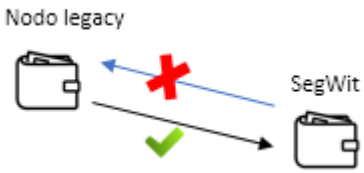
Tuttavia, se le definizioni date sono valide, il soft fork più famoso del momento, l'upgrade SegWit, non può essere considerato un soft fork. L'upgrade a SegWit era stato progettato come hard fork, ma con un po' di ingegno nel team di Core sono riusciti a trasformarlo in una soft fork, almeno secondo una specifica definizione di soft fork:

(3) Una soft fork è una modifica del protocollo a cui si converge nel momento in cui la maggioranza di hash power la adotta

"A softfork is defined as a consensus change that converges as long as majority of the hashrate adopts it"

(Pieter Wuille, Bitcoin Core maintainer, Blockstream co-founder, vedi qui)

SegWit non ha le stesse caratteristiche di un soft fork come quello finora descritto sulla limitazione del blocksize. Con SegWit la parte della firma non viene più inclusa nel blocco (blocksize), ma viene scambiata dai nodi a parte, in un pacchetto dati separato. A livello di transazione fra nodo e nodo, i nodi vecchi non vedono questo pacchetto dati separato, quindi non riescono a ricevere transazioni di tipo SegWit (non vedendo la parte della firma dove si aspettano di trovarla, ovvero nel blocksize). Questo invalida la definizione (2): il vecchio software non riesce a ricevere transazioni del nuovo tipo. Anche se i nuovi nodi Core sono progettati sia per effettuare transazioni a firma separata SegWit sia vecchie transazioni di tipo legacy e quindi possono inviare bitcoin anche a nodi non SegWit, non possono farlo se i bitcoin da spedire si trovano in un output di tipo SegWit (cioè ricevuti da una precedente transazione SegWit). Se si tentasse la transazione, questa restituirebbe un errore: missing witness data.



La definizione **(2)** andrebbe quindi rivista. Non è più vero che la soft fork è:

(2.a) *Una modifica per cui vecchie versioni del software riconosceranno* ***le transazioni****generate da nuove versioni del software (retrocompatibilità)*

Infatti le vecchie versioni del software non riconosceranno **alcune delle possibili transazioni** generate da nuove versioni del software

Tuttavia le vecchie versioni del software riconosceranno **i blocchi** generati dai miners SegWit, anche se questi blocchi contengono output SegWit che non sono riconosciuti dai nodi legacy (suona strano, ma è così). Quindi la soft fork rimane:

(2.b) *Una modifica per cui vecchie versioni del software riconosceranno **i blocchi** generati da nuove versioni del software (retrocompatibilità)*

Questo fatto è importante, perché significa che la rete può accettare i blocchi SegWit come la propria blockchain, e non un (hard) fork incompatibile, come se fosse la blockchain di un'altra moneta, ad esempio Litecoin.

Nel merito della definizione **(2.b)**, va fatta però una precisazione:

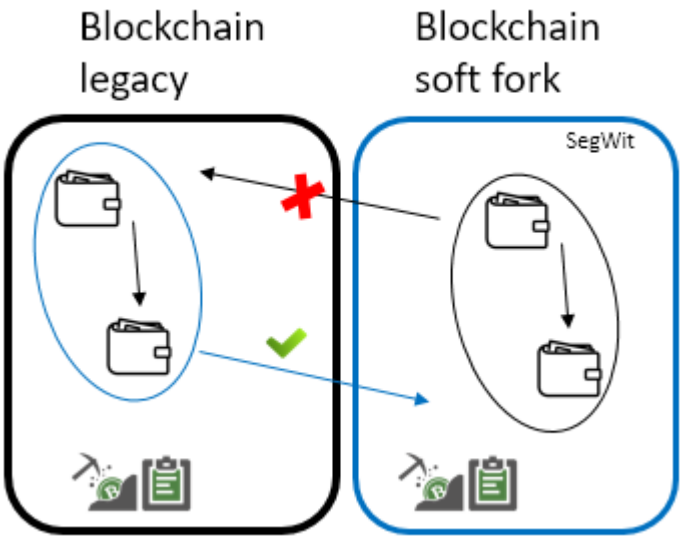
– **il software dei nodi riconoscerà i blocchi** generati da nuove versioni del software (seppur non possa accettare alcune delle transazioni provenienti da output proprio di quei blocchi: quelle provenienti da output segwit)

– **il software dei miners legacy non riconoscerà i blocchi** creati dai miners SegWit

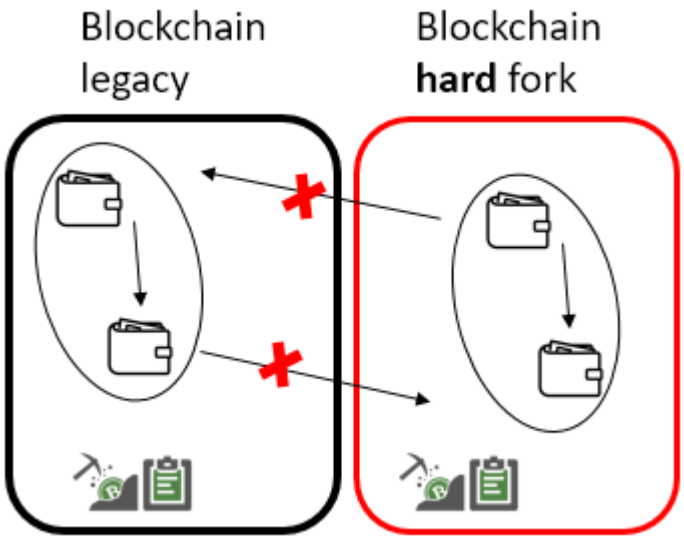
Quest'ultimo fatto è fondamentale, ed è ciò che mette a rischio di split la blockchain. Infatti, se solo una parte dei miners fa upgrade, si creano due catene inconciliabili: la catena legacy escluderà qualsiasi blocco che abbia almeno una transazione SegWit.

Infine, dobbiamo rivedere anche la definizione **(1)**, per cui affermavamo che il soft fork rende invalidi blocchi che prima erano validi. In effetti, la catena SegWit può includere tutti i tipi di blocchi che oggi sono considerati validi nella catena legacy (vedremo perché bisogna precisare "oggi"). Qualsiasi transazione che oggi viene validata nella catena legacy potrebbe essere validata anche nella catena SegWit, mentre non vale il contrario (perché i miners legacy non

ammettono transazioni SegWit).



In caso di Hard Fork la catena che fa upgrade non avrebbe questo vantaggio:



Si tratta di vantaggio per un motivo preciso: la catena che fa upgrade può validare nella blockchain tutti i tipi di transazione, potendo così "accontentare" sia gli utenti che vogliono usare client legacy, sia gli utenti che vogliono usare client SegWit.

In definitiva, se SegWit è legittimamente un "soft fork", le definizioni **(1)** e **(2)** di soft fork sono poco precise, se non sbagliate. Conviene utilizzare la definizione di Wuille:

(3) una soft fork è una modifica del protocollo a cui si converge nel momento in cui la maggioranza di hash power la adotta (Pieter Wuille)

> A softfork is defined as a consensus change that converges as long as majority of the hashrate adopts it. Yes, a non-witness transaction spending a segwit output without signature would be valid to old nodes, but 1) they won't see that (such a transaction is nonstandard to everyone) 2) such a transaction won't be mined (same as 2) and thus 3) the majority chain will not accept such transactions. – Pieter Wuille May 16 at 17:40

Se la maggioranza dei miners accetta Segwit (scaricano un software compatibile), tutte le transazioni (legacy e SegWit) saranno approvate nella catena SegWit e i client (anche quelli legacy) faranno "reorg", riconoscendo quest'ultima come l'unica valida. Questo è il meccanismo di "blockchain reorganization" di cui trattato **sopra.** La particolarità di questa caratteristica (e che la differenzia da un generico hard fork) è che qualsiasi client, legacy o SegWit, riconoscerà come blockchain valida solo la catena SegWit, scartando l'altra, in qualunque momento la catena SegWit supererà in lunghezza quella legacy (ovvero quando avrà maggiore hashpower e lavoro da parte dei miners). In caso di reorg, nel momento in cui tutti i client della rete riconoscono SegWit come la catena valida, la catena legacy muore. Viceversa, se la catena SegWit non supererà in lunghezza la legacy, quindi non raggiungerà mai il sostegno della maggioranza di miners e hashpower, verrà probabilmente abbandonata.

Per finire, bisogna segnalare un'ultima e importante sfaccettatura: abbiamo detto che la catena SegWit accetta blocchi con tutti i tipi di transazione che vengono effettuati oggi, anche quelli legacy, ma questo fatto *potrebbe* non essere vero in futuro. Infatti i vecchi nodi legacy non vedono la parte della firma delle transazioni SegWit, quindi vedono i bitcoin che sono output di transazioni SegWit come "di nessuno". Se dei bitcoin sono di nessuno, te ne puoi appropriare, facendo una transazione di tipo legacy (aggiungendo quindi la parte della firma) e mandandoli verso un tuo indirizzo: questa caratteristica degli output SegWit è

conosciuta come “anyone can spend”. La possibilità di sfruttare l’”anyone can spend” mette a rischio la catena che fa soft fork, se non ha la maggioranza dell’hashrate. Infatti una transazione legacy che sfrutti un output anyone can spend è compatibile con il protocollo legacy e può essere quindi validata dai miners all’interno dei blocchi della catena legacy, permettendo il “furto” dei bitcoin presenti in qualsiasi output SegWit. Ovviamente i miners della catena SegWit in “soft fork” non potranno mai accettare blocchi in cui compaiono transazioni che spendono a proprio piacimento i bitcoin degli utenti con client SegWit (in effetti derubandoli), di conseguenza rifiuteranno quei blocchi. In effetti quindi SegWit renderebbe invalidi dei blocchi che in base al protocollo legacy sono validi e la definizione (1) di soft fork che avevamo dato torna valida.

Su Litecoin, dove è stato approvato SegWit, nessuno ha mai sfruttato la caratteristica “anyone can spend”, perché un blocco che includa una transazione di quel tipo creato da un miner verrebbe orfanato, ovvero rifiutato dalla totalità degli altri miners, che supportano le attuali regole di consenso all’unanimità (tutti hanno approvato SegWit). Nessun miner quindi spenderebbe soldi e tempo convalidando bocchi in cui ci sono transazioni che rubano soldi dagli output SegWit. Ma se non c’è un vasto supporto “politico” all’upgrade, non si possono garantire le stesse condizioni. In definitiva, suona come uno scherzo, ma se SegWit non avesse la maggioranza di hashpower si avrebbe una situazione per cui: un nodo potrebbe accettare un blocco, ma non accettare di ricevere output da transazioni SegWit di quel blocco; output che tuttavia potrebbe tentare di spendere (rubando i bitcoin), pur rispettando il protocollo (legacy), e rimanendo quindi nell’incertezza se questa transazione verrà validata o meno dai blocchi successivi dei miners, sapendo che se venisse validata si creerebbe uno split della blockchain, che non si saprà quanto durerà e quanto catastrofico potrà essere sull’intero ecosistema.

L’ESPRESSIONE DEL VOTO POLITICO E DELLE INTENZIONI DI UPGRADE

Situazioni catastrofiche nel sistema Bitcoin non si sono mai verificate. Questo anche grazie ad un softisticato sistema di “voto politico”, per cui viene espressa una preferenza nei confronti di un upgrade. Generalmente, solo quando una proposta è condivisa alla quasi unanimità, si procede all’upgrade. Quando in Italia siamo passati dalla Lira all’Euro, non tutti erano d’accordo con la scelta politica. Ma quando ormai era inevitabile che si sarebbe fatto il passaggio, anche i più strenui difensori dello status quo si sono dovuti rassegnare e accettare l’Euro. Se qualcuno si fosse tenuto le lire pretendendo di utilizzarle come moneta, magari sarebbe riuscito a continuare a scambiarle all’interno di una comunità ristretta, ma di certo il fenomeno non sarebbe stato un problema a livello di sistema né avrebbe potuto minare la solidità dell’Euro. Nel Bitcoin funziona allo stesso modo: immaginiamo il protocollo Bitcoin come il taglio e la stampa di una certa banconota. La banconota ha valore solo perché c’è consenso fra i cittadini (e le istituzioni che questi formano) che quella stampa rappresenti esattamente una moneta con quello specifico valore. Nel momento in cui abbiamo deciso che la Lira non valeva più niente, è diventata carta straccia, e le *consensus rules* che stanno alla base del “protocollo” sono state modificate: “accettare solo banconote con scritto sopra Euro”.

Nel 2012 gli sviluppatori di Core hanno formulato una proposta di miglioramento di Bitcoin, il BIP 34 (Bitcoin Improvement Proposal numero 34), per cui si è stabilito un metodo sofisticato di “segnalazione” della volontà di fare l’upgrade da parte della rete. Questo metodo è stato poi perfezionato con BIP9 nel 2015.

Il funzionamento è il seguente: i miners segnalano l’intenzione di aderire a una proposta di cambiamento del protocollo tramite il blocco creato, che viene inserito nella blockchain e propagato alla rete insieme a un dettaglio

importante: un messaggio inserito nel “block version”, ovvero in una stringa di 32bit. Ad esempio:

```
Version field: 00001000 00000000 00000010 00000000
                    ^      ^           ^
                    |      |           Supporto un aumento dei btc a 50 milioni
                    |      Non voglio bene a Salvini né ai gattini
                   Supporto i blocchi grandi 16mb
```

Quando si propone un particolare Bitcoin Improvement Proposal (BIP) per modificare il protocollo, si indica quale sia il bit per sostenerlo. Quando un miner crea un blocco della blockchain, dichiara che è pronto all’upgrade a quel determinato BIP indicando 1 anziché 0 nel corrispondente bit del version field.

Più un miner è ricco e potente, maggiore è la quantità di blocchi che crea rispetto ai miners concorrenti, perciò il voto non viene espresso nella modalità 1 testa = 1 voto (o 1 macchina = 1 voto), bensì con potenza di calcolo. Maggiore la potenza, maggiore influenza può avere sulle decisioni della rete.

A **questo link** vedete i risultati in tempo reale del voto dei miners

È curioso notare che i miners possano esprimersi anche a favore di proposte non ancora concrete, ovvero in favore di idee che non hanno ancora dato vita a un software definitivo, come invece avviene nel caso dei BIP. Ad esempio, oggi il client btc1 della proposta Segwit2x è ancora in fase di testing, eppure i miners già stanno indicando l’intenzione di adottarlo, scrivendo all’interno della transazione “coinbase” le lettere: “NYA”. La sigla sta per New York Agreement, il patto stretto il 23 maggio su proposta di Barry Silbert, per cui la maggioranza di miners e grosse aziende di servizi in btc concordano sull’upgrade SegWit + 2mb di blocco.

Dunque per proposte indefinite o per comunicare un qualsiasi tipo di messaggio, i miners anziché il version field del

blocco possono usare la coinbase, che all'interno del nuovo blocco creato è la transazione che genera nuovi bitcoin (la ricompensa del miner per aver scoperto il blocco). Già Satoshi Nakamoto aveva inserito un messaggio nella coinbase del Genesis block (il primo blocco Bitcoin), con il famoso testo: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks". Il messaggio non solo segnava con precisione la data della nascita della blockchain Bitcoin, ma poteva già essere interpretato proprio come un messaggio politico.

THRESHOLD: LA SOGLIA DI VOTO DA RAGGIUNGERE PER L'UPGRADE

Un elemento fondamentale per ogni upgrade Bitcoin è la soglia di segnalazione. Generalmente un BIP specifica una percentuale di supporto che deve raggiungere la segnalazione dei miners prima che, di concerto, tutti facciano l'upgrade in una specifica data. Se la soglia non è raggiunta, l'upgrade non viene fatto da nessuno, nemmeno da chi lo ha proposto. Questo fattore è fondamentale, poiché se non si è a conoscenza del supporto verso una particolare proposta, il rischio è uno split della rete e della blockchain, con scenari anche catastrofici.

Non è detto che tutte le proposte sul mercato abbiano una threshold di segnalazione. E questo è un male. La tipologia di segnalazione di voto che abbiamo visto appartiene soltanto ai miners, che sono solo una fetta della rete. Non c'è invece un sistema analogo di voto da parte degli utenti. L'upgrade che alcuni utenti avevano minacciato il 1 agosto 2017, ovvero BIP 148, era un **UASF**, ovvero User Activated Soft Fork: un upgrade "attivato" dagli utenti, non dai miners (altrimenti chiamato **MASF**: Miner Activated Soft Fork).