

# Bitcoin Transaction Lifecycle

## Dalla Mempool alla Conferma: RBF e CPFP

Valerio Vaccaro

Satoshi Spritz Connect

18 Novembre 2025

- 💻 Sviluppatore Bitcoin ed Esperto Hardware
- 🔥 Contributore a progetti Bitcoin open source
- ⚠️ Appassionato di hardware fai-da-te (DIY)
- Ingegnere Bitcoin e Liquid presso Blockstream

## Social

- 👤 **LinkedIn** [linkedin.com/in/valeriovaccaro](https://linkedin.com/in/valeriovaccaro)
- 🐙 **Github** [github.com/valerio-vaccaro](https://github.com/valerio-vaccaro)
- **Telegram** [t.me/valeriovaccaro](https://t.me/valeriovaccaro)

Questa presentazione è distribuita sotto la licenza Creative Commons [CC BY-SA 4.0](#).

Le immagini utilizzate in questa presentazione sono proprietà dei rispettivi autori e sono incluse solo a fini educativi e illustrativi.

May this presentation inspire you to become more self-sovereign!



## Contenuti della Presentazione

- 🔑 Transaction Lifecycle
- ⚙️ Struttura di una Transazione Bitcoin
- 🚀 Creazione e Propagazione
- ⚙️ Mempool: Cosa è e Come Funziona
- 🔥 Mining e Conferma
- 📈 Fee Rate e Priorità
- ⚙️ Replace-By-Fee (RBF)
- 💛 Child-Pays-For-Parent (CPFP)
- 💡 Strategie Pratiche
- ⚠️ Considerazioni di Sicurezza

## Cos'è?

- 🚀 Il percorso completo di una transazione Bitcoin dalla creazione alla conferma
- ⚙️ Processo multi-step che coinvolge wallet, nodi, miner e blockchain
- ⚙️ Comprende aggiornamento wallet, creazione, propagazione, validazione, attesa nella mempool e mining

## Perché è Importante?

- 💡 Comperderlo aiuta a ottimizzare fee e tempi di conferma
- 🛡️ Permette di gestire situazioni di emergenza (transazioni bloccate)
- 📈 Essenziale per applicazioni che richiedono conferme rapide
- ⚠️ Conoscere RBF e CPFP può salvare transazioni “bloccate”







## Componenti Base

- 🔑 **Version:** Versione del protocollo della transazione
- 🏠 **Inputs:** Riferimenti a UTXO precedenti (prevout)
- 🏠 **Outputs:** Nuovi UTXO creati dalla transazione
- 🔒 **Locktime:** Blocco/altezza minima per validità





## Input e Output

- 🏠 Input:
  - Previous TXID + Output Index
  - ScriptSig (firma + dati di sblocco)
  - Sequence Number
- 🏠 Output:
  - Amount (in satoshi)
  - ScriptPubKey (condizioni di spesa)

## Creazione della Transazione

-  **Aggiornamento wallet:** Cerco nuovi UTXO
-  **Creazione Output:** Creazione della transazione base e aggiunta Output
-  **Selezione UTXO:** Wallet sceglie UTXO da spendere
-  **Calcolo Fee:** Determina fee rate appropriato
-  **Firma:** Firma digitale degli input con chiave privata
-  **Serializzazione:** Conversione in formato binario/hex

## Propagazione nella Rete

-  Transazione inviata a nodi peer connessi
-  Ogni nodo valida e propaga ad altri peer
-  Diffusione esponenziale nella rete Bitcoin
-  Validazione include: firme, doppia spesa, format

# ⚙️ Mempool: Cosa è e Come Funziona

## Cos'è Mempool?

- 🏠 **MemPool**: Pool di transazioni non ancora confermate
- ⚙️ Ogni nodo mantiene la propria mempool locale e potenzialmente **differente**
- 📈 Ordinato per fee rate (più alto = più prioritario)
- ⚙️ Transazioni **valide** in attesa di essere incluse in un blocco

## Caratteristiche del Mempool

- 🔥 **Dinamico**: Continuamente aggiornato con nuove transazioni
- 🔥 **Espulsione**: Transazioni vecchie o invalide vengono rimosse (-> [mempush.com](https://mempush.com))
- 📈 **Prioritizzazione**: Fee rate determina l'ordine
- ⚠️ **Limitato**: Dimensione massima per evitare DoS

## Processo di Mining

- 1 ⚙️ Miner seleziona transazioni dalla mempool
- 2 📄 Costruisce un blocco candidato con transazioni prioritarie
- 3 🔥 Calcola hash del blocco cercando nonce valido
- 4 🚀 Quando trova hash valido, propaga il blocco

## Conferma della Transazione

- ✓ **1 conferma:** Transazione inclusa in un blocco
- ✓ **6 conferme:** Standard per transazioni importanti
- 🛡️ Ogni blocco successivo aumenta sicurezza
- 🗑️ Dopo conferma(e), transazione è difficilmente reversibile (-> irreversibile)

### Cos'è il Fee Rate?

- 🪙 **Fee Rate:** Fee pagata per byte di transazione (sat/vB)
- 📈 Determina la priorità nel mempool
- 🚀 Fee rate più alto = conferma più rapida
- 📊 Calcolato come:  $\text{fee} / \text{size\_in\_bytes}$

### Strategie di Fee

- 🚀 **High Priority:** Fee rate alto per conferme immediate
- ⚙️ **Standard:** Fee rate medio per conferme entro 1-3 blocchi
- ⚠️ **Low Priority:** Fee rate basso, può richiedere ore/giorni
- 💡 **Fee Estimation:** Wallet stimano fee rate ottimale

## ⚙ E se le fee non bastano?



# ⚙️ Replace-By-Fee (RBF)

## Cos'è RBF?

- ⚙️ **Replace-By-Fee:** Meccanismo per sostituire una transazione non confermata
- ⚠️ La può usare solo chi è capace di firmare gli input (sender)
- 🚀 Permette di aumentare il fee rate di una transazione già propagata
- 🛡️ Standardizzato in BIP 125

## Come Funziona RBF?

- 1 ⚙️ Creare nuova transazione con stesso input
- 2 📈 Aumentare fee rate (o aggiungere output o cambiarli)
- 3 🚀 Propagare nuova transazione nella rete

### Requisiti per RBF

- 📈 **Fee Increase:** Nuova fee deve essere maggiore
- ⚙️ **Same Inputs:** Deve spendere almeno gli stessi input
- 🛡️ **No Double Spend:** Non può spendere input già confermati

### Tipi di RBF

- ⚙️ **Full RBF:** Qualsiasi transazione può essere sostituita
- 🔑 **Opt-in RBF:** Solo transazioni con nSequence appropriato ( $< 0xFFFFFFFF$ )
- 🛡️ La maggior parte dei nodi supporta Opt-in RBF

## ⚙️ Replace-By-Fee (RBF) - Esempio Pratico

### Scenario

Transazione Originale:

Input: 100,000 sats

Output: 99,000 sats

Fee: 1,000 sats (1 sat/vB)

Status: Bloccata nel mempool

### Soluzione con RBF

Nuova Transazione (RBF):

Input: 100,000 sats (stesso)

Output: 98,000 sats

Fee: 2,000 sats (2 sat/vB)

Status: Confermata rapidamente

# 💛 Child-Pays-For-Parent (CPFP)

## Cos'è CPFP?

- 💛 **Child-Pays-For-Parent**: Transazione figlia paga per transazione genitore
- 🚀 Strategia alternativa a RBF
- ⚠️ La può usare solo chi può spendere un output
- ✂️ Crea nuova transazione che spende output della transazione bloccata
- ⚙️ Fee della transazione figlia include fee per entrambe

## Quando Usare CPFP?

- ⚠️ Quando RBF non è disponibile (nSequence final)
- 🔑 Quando si controlla un output della transazione bloccata
- 🚀 Per accelerare transazioni ricevute (non proprie)
- 🛡️ Quando si vuole evitare di rivelare nuovi input

### Come Funziona CPFP?

- 1 🏠 Identificare output della transazione bloccata
- 2 ⚙️ Creare nuova transazione che spende quell'output
- 3 📈 Includere fee sufficiente per entrambe le transazioni
- 4 🚀 Miner includerà entrambe per ottenere fee totale

## 👉 Child-Pays-For-Parent (CPFP) - Dettagli Tecnici

### Calcolo Fee CPFP

Fee Totale Necessaria = Fee Parent + Fee Child + Fee Extra

Esempio:

Parent TX: 1,000 sats fee, 500 vB

Child TX: 500 vB

Fee Rate Target: 5 sat/vB

Fee Child =  $(500 + 500) * 5 - 1,000 = 4,000$  sats

## 👉 Child-Pays-For-Parent (CPFP) - Esempio Pratico

### Scenario

Transazione Parent (bloccata):

Input: 100,000 sats

Output 1: 50,000 sats (tuo)

Output 2: 49,000 sats

Fee: 1,000 sats (1 sat/vB)

Status: Bloccata nel mempool

### Soluzione con CPFP

Transazione Child:

Input: 50,000 sats (Output 1 della parent)

Output: 45,000 sats

Fee: 5,000 sats (include fee per parent)

Status: Entrambe confermate insieme

## Quando Usare RBF?

- 🔑 Transazione propria con nSequence modificabile
- 🚀 Bisogno di accelerare conferma rapidamente
- ✍️ Si può modificare output o aggiungere fee
- 🛡️ Situazione più semplice e diretta

## Quando Usare CPFP?

- ⚠️ RBF non disponibile (nSequence final)
- 🪙 Si controlla un output della transazione bloccata
- 🤝 Transazione ricevuta che si vuole accelerare
- ⚙️ Si vuole evitare di rivelare nuovi input

# ⚠ Considerazioni di Sicurezza





## Rischi di RBF

- ⚠ **Double Spend**: Transazione originale può essere sostituita
- 🔗 **Privacy**: RBF può rivelare che si controlla un ulteriore input
- 🛡 **Acceptance**: Non tutti i wallet accettano transazioni RBF





## Rischi di CPFP

- 💰 **Costo**: Può richiedere fee significativamente più alte
- ⚙ **Complessità**: Richiede controllo di output specifico
- ⚙ **Timing**: Entrambe le transazioni devono essere incluse insieme
- 🛡 **Acceptance**: Alcuni wallet non supportano coin selection (necessaria per CPFP)

## Strumenti di Fee Estimation

-  **Bitcoin Core:** estimatesmartfee RPC
-  **Mempool.space:** API pubblica per fee estimation
-  **Blockstream Esplora:** Visualizzazione mempool
-  Wallet integrano questi strumenti automaticamente

## Strategie di Ottimizzazione

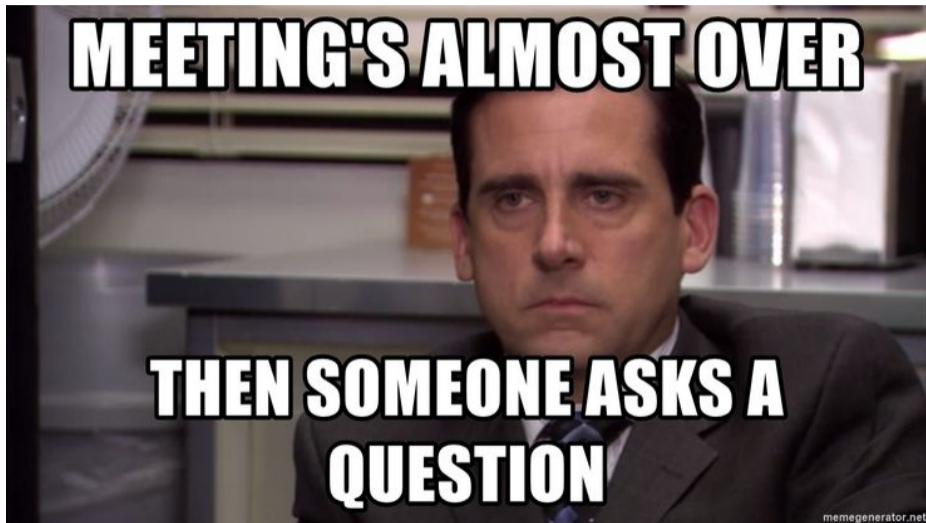
-  **Batching:** Combinare più pagamenti in una transazione
-  **SegWit:** Usare indirizzi SegWit per ridurre dimensioni
-  **Coin Selection:** Scegliere UTXO per minimizzare fee
-  **Timing:** Inviare durante periodi di basso traffico

## Fattori che Influenzano il Mempool

- 📈 **Volume Transazioni:** Più transazioni = più competizione
- 🔥 **Congestione:** Periodi di alta attività aumentano fee
- ⚙️ **Dimensione Blocchi:** Limite 1MB per blocco
- ⚙️ **Politiche Nodi:** Diversi nodi hanno politiche diverse

## Comportamento del Mempool

- 🔥 **Espulsione:** Transazioni vecchie vengono rimosse
- 📈 **Prioritizzazione:** Fee rate determina ordine
- ⚙️ **Limite dimensione:** Se imposto limiti di dimensione scarto le transazioni con fee più basse in caso di picchi









## Risorse Principali

- **BIP 125:** Opt-in Full Replace-by-Fee Signaling
- **Bitcoin Core Documentation:** Mempool and Transaction Relay
- **Mastering Bitcoin:** Andreas M. Antonopoulos
- **Bitcoin Developer Guide:** [bitcoin.org/developer-guide](https://bitcoin.org/en/developer-guide)

## Link Utili

- [Mempool.space](https://mempool.space) - Visualizzazione mempool
- [Blockstream Esplora](https://blockstream.info/esplora) - Explorer Bitcoin
- [Bitcoin Core RPC](https://bitcoincore.org/en/rpc/)
- [BIP 125](https://bitcoin.org/en/bips/125)

-  Federazione di gruppi locali di Bitcoiner
-  Eventi gratuiti e privacy oriented
-  BITCOIN ONLY
-  Satoshi Spritz Connect online settimanale
-  Orientato all'apprendimento della self-sovereign
-  Tutte le settimane un evento online -> Satoshi Spritz Connect

## Links

- [satoshispritz.it](https://satoshispritz.it)
- [t.me/SatoshiSpritzConnect](https://t.me/SatoshiSpritzConnect)

- 🇮🇹 Comunità Italiana di Bitcoiners, totalmente gratuita
- 🤖 BITCOIN ONLY
- 🎓 Focus su educazione e sviluppo di progetti
- 📋 Progetti:
  - 📁 Sviluppo nodi Bitcoin
  - 🧑💻 Uso di Hardware Wallet
  - 💻 Filosofia open source
  - 🇮🇹 Installazione di Debian
  - 🎲 Mnemoniche & Dadi
  - ... e molto altro

## Links

- [officinebitcoin.it](https://officinebitcoin.it)