**Analisi e comparazione dei sistemi DeFI che usano AMM e Liquidity Pools**

Cocco Mattia 65336 e Lepuri Tomas 65358

Università degli Studi di Cagliari

Corso di Laurea Magistrale in Informatica

Blockchain and Smart Contracts

Anno Accademico 2024-2025

Indice

[DeFI 3](#_Toc202343933)

[Licenza 3](#_Toc202343934)

# Decentralized Finance

La DeFI, o **Finanza Decentralizzata**, consiste in un insieme di protocolli e applicazioni, generalmente basate su tecnologie blockchain, con lo scopo di offrire servizi finanziari senza la necessità di intermediari centralizzati, come banche, broker o assicurazioni, di cui bisogna necessariamente fidarsi.

Nella finanza tradizionale infatti (centralizzata), ogni transazione e servizio in generale sono gestiti da un ente centrale, rigidamente regolamentato da requisiti e autorizzazioni, con una trasparenza limitata e velocità delle operazioni strettamente legata ai ritmi lavorativi del settore.

Al contrario, la DeFI sfrutta gli **smart contracts**, programmi autonomi e immutabili, in continua esecuzione su blockchain pubbliche, come Ethereum, che permettono di automatizzare operazioni come scambi di valute, prestiti, gestione di assets e rendimenti, garantendo più accessibilità, trasparenza, rapidità e disponibilità dei servizi.

Possiamo infatti riassumere alcuni principi cardine:

* **Decentralizzazione** o **disintermediazione**: elimina l’esigenza di fiducia verso banche o altre istituzioni.
* **Trasparenza**: il codice è open-source e le transazioni sono registrate su blockchain pubblicamente consultabili.
* **Accessibilità**: è possibile accedere a questi servizi con meno controlli bancari o territoriali, dato che molti provider di servizi hanno comunque bisogno di dotarsi di misure di controllo e KYC per poter essere regolamentati nell’UE o simili.
* **Composability**: i protocolli DeFI possono essere integrati tra loro, creando applicazioni più complesse.

La nascita della Finanza Decentralizzata può essere ricondotta alla diffusione di Ethereum, che ha introdotto la possibilità di scrivere questi contratti intelligenti. Tra i primi protocolli, vale la pena citarne alcuni:

* **MakerDAO**: il protocollo che permise la prima stablecoin decentralizzata.
* **Compound**: un protocollo di tasso di interesse autonomo.
* **Uniswap**: uno swap di criptovalute decentralizzato, che verrà approfondito in seguito.

Con la sua evoluzione, questo ecosistema è arrivato a comprendere decine di miliardi di dollari in Total Value Locked.

**Automated Market Makers**

Un AMM, o **Automated Market Maker**, è un tipo di algoritmo, impiegato principalmente negli Exchange Decentralizzati, che consente lo scambio di token senza la necessità di una controparte diretta umana (senza un order book).

Nei mercati tradizionali la domanda di beni o valute viene abbinata all’offerta, mentre gli AMM utilizzano una formula matematica per determinare il prezzo dei token, sulla base della disponibilità nei liquidity pools: sostanzialmente degli smart contracts che detengono riserve di due o più token, come verrà spiegato meglio al capitolo successivo.

Un noto esempio è Uniswap v2, che utilizza la seguente formula:

Dove:

* ***x*** e ***y*** sono le quantità dei due token nel pool.
* ***k*** è una costante fissa.
* Ogni swap modifica ***x*** e ***y***, mantenendo ***k*** costante.

Supponiamo di avere un pool ETH/DAI con:

* ***x*** = 10 ETH
* ***y*** = 200 DAI
* Quindi ***k*** = 10 ⋅ 200 = 2000

Se uno swap aggiungesse 1 ETH al pool, l’utente riceverebbe meno DAE in cambio di ETH dato che dovrebbe essere soddisfatta 11 ⋅ y = 2000.

Questo è solo un modello di AMM, chiamato **Constant Product**, dato che k è una costante invariabile, il prezzo si adatta dinamicamente al rapporto tra le quantità dei token e la curva dei prezzi è iperbolica.

Esistono altri modelli di AMM, tra cui:

* **Constant Mean**: che generalizza il modello a più di due token, permette di dare pesi diversi ai token che influiscono sulla determinazione dei prezzi, con formula:
* **Logarithmic Market Scoring Rule**: utilizzato per i prediction markets, permette di calcolare il prezzo marginale di ogni outcome e ha formula del costo:

Dove ***qi*** è la quantità acquistata dall’evento *i* e ***b*** è un parametro di liquidità.

Altri esempi di attori rilevanti nel campo degli AMM, oltre Uniswap, sono:

* **SushiSwap**: una fork di Uniswap che implementa delle differenze nel meccanismo di incentivazione per i fornitori di liquidità.
* **PancakeSwap**: AMM Constant Product che opera sulla Binance Smart Chain, con focus su token BEP-20.
* **Balancer**: AMM Constant Mean che consente la creazione di pool con più di due token e pesi personalizzabili associati.
* **Augur**: piattaforma di predizione del mercato che utilizza appunto un AMM LMSR.

**Esempio implementativo**

L’applicazione dei concetti appena descritti può essere osservata perfettamente nel **core smart contract** di Uniswap v2.

La funzione di **swap** prende come parametri le quantità dei token e l’indirizzo che riceverà i token in uscita, in seguito controlla subito che almeno uno dei due amount sia positivo e la liquidità del pool sia sufficiente:

A computer screen shot of text

AI-generated content may be incorrect.

Il contratto invia all’utente i token richiesti e verifica quanto è stato depositato in cambio, confrontando i saldi attuali con le riserve, per poi calcolare gli importi in ingresso:

A screen shot of a computer code

AI-generated content may be incorrect.

Viene applicata la **fee** di Uniswap dello 0.3%, che equivale a moltiplicare per 997/1000:

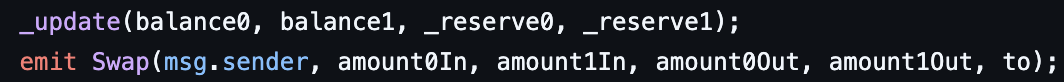


Viene effettuato il controllo fondamentale, che si assicura che nessuno possa ottenere più di quanto dovrebbe e che il valore del pool non diminuisca:



Che equivale a controllare che:

Infine vengono aggiornate le riserve e viene emesso l’**evento** di Swap, che verrà salvato nel log della blockchain:



In sintesi:

* L’utente dice quanto vorrebbe ricevere.
* Il contratto calcola se sono stati ricevuti abbastanza token in cambio, tenendo conto anche della fee con la formula matematica.
* In caso sia tutto corretto invia i token richiesti, aggiornando le riserve, altrimenti la transazione fallisce.

**Liquidity pools**

Un **Liquidity Pool** consiste in una riserva di due o più token che vengono bloccati tramite uno smart contract che consente agli utenti di scambiarli automaticamente, secondo regole predefinite, solitamente tramite un AMM.

Gli utenti che forniscono i fondi al pool sono detti **Liquidity Providers** e ricevono in cambio degli LP tokens, che rappresentano la loro quota nel pool, in questo modo con l’aumentare del valore del pool tramite le fees, i providers vedranno una plusvalenza della loro quota.

Naturalmente se il valore dei token (come ETH) diminuisce fortemente rispetto al momento del deposito, un Liquidity Provider può perdere denaro anche se il valore totale del pool è cresciuto tramite le fees, rischio chiamato **Impermanent Loss**.

Sostanzialmente, automated market makers e liquidity pools lavorano in simbiosi:

* Il pool contiene i due token.
* Gli utenti possono swapparli.
* I Liquidity Providers guadagnano sulle commissioni.
* I prezzi vengono determinati dinamicamente sulla base del rapporto tra token.

**Esempio implementativo**

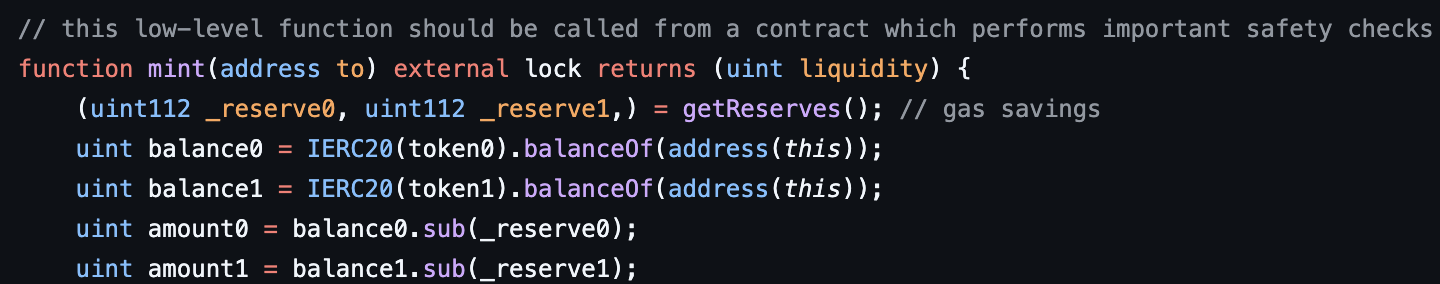
Anche in questo caso, nel core contract di Uniswap v2 sono presenti alcune funzioni che permettono di comprendere meglio i concetti teorici.

La prima funzione restituisce le riserve correnti dei token e il timestamp dell’ultima operazione, utile per calcolare i prezzi medi ponderati nel tempo (TWAP), le riserve sono variabili di stato che vengono aggiornate a ogni swap o cambiamento della liquidità:

A black screen with text

AI-generated content may be incorrect.

La funzione **mint** invece viene chiamata ogni volta che un Liquidity Provider aggiunge liquidità al pool. Innanzitutto calcola il saldo dei token, leggendo le riserve attuali:



In seguito, se un provider è il primo in assoluto, la liquidità viene calcolata come radice quadrata del prodotto delle quantità di token, altrimenti è calcolata sulla base dei token aggiunti alle riserve esistenti:

A computer screen with white text

AI-generated content may be incorrect.

Infine, se la liquidità è sufficiente, vengono emessi gli LP token, vengono aggiornate le riserve e viene emesso l’evento di **Mint**:

A computer screen with white text

AI-generated content may be incorrect.

La funzione **burn**, al contrario, permette ai Liquidity Providers di prelevare la loro quota, “bruciando” gli LP token. Per prima cosa legge le riserve e i saldi attuali, calcola quanti token vuole bruciare l’utente e quanti ne deve restituire all’indirizzo to:



In seguito, distrugge i token, aggiorna le riserve e genera l’evento **Burn**:

A computer screen shot of text

AI-generated content may be incorrect.

**Attori Principali**

**Bibliografia**

Andrea Cesaretti - Auto-Paired DeFi Pools: Illusory Liquidity and Price Manipulation. A Critical Review.

Yongge Wang UNC Charlotte - Automated Market Makers for Decentralized Finance (DeFi).

Hayden Adams, Noah Zinsmeister, Dan Robinson – Uniswap v2 Core.

Coinbase.com

# Licenza

© 2025 Mattia Cocco & Tomas Lepuri – Licenza CC BY 4.0