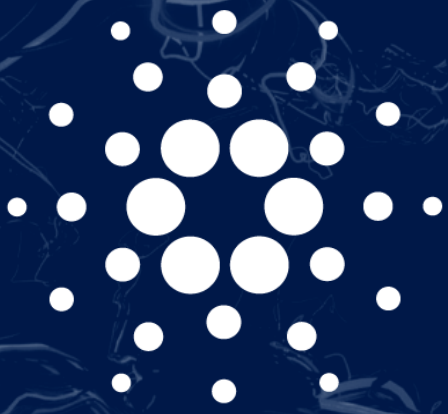


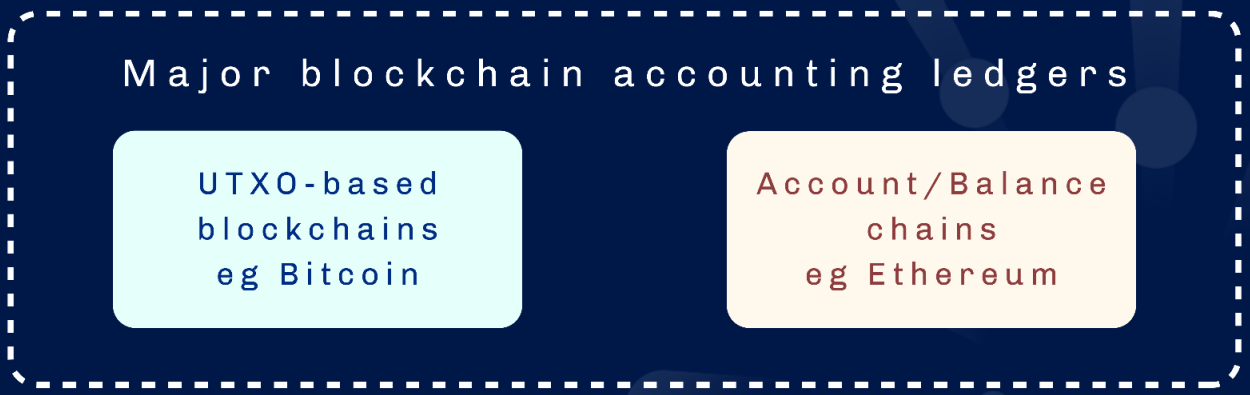
CARDANO

EUTXO Handbook

Introduction

Blockchain networks are complex data structures. Transactions continuously crisscross the chain, creating digital footprints that require careful tracking and management to maintain the integrity and reliability of the underlying ledger.

Two major accounting ledgers exist in the blockchain space: UTXO-based blockchains (Bitcoin, for instance), and Account/Balance chains (Ethereum, and others).

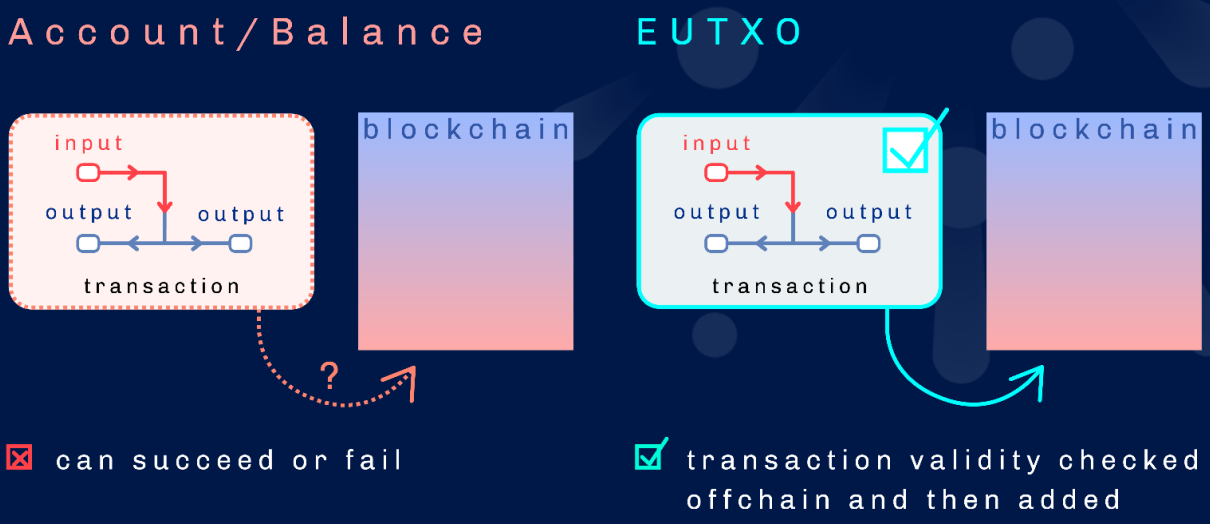


Cardano combines Bitcoin's UTXO model with the ability to handle smart contracts into an Extended Unspent Transaction Output (EUTXO) accounting model. The adoption of EUTXO facilitates the implementation of smart contracts into the Cardano chain.

The EUTXO model offers unique advantages over other accounting models.

The EUTXO model offers unique advantages over other accounting models. For example, the success or failure of transaction validation depends only on the transaction itself and its inputs, and not on anything else on the blockchain. As a consequence, the validity of a transaction can be checked off-chain, before the transaction is sent to the blockchain. A transaction can still fail if some other transaction concurrently consumes an input that the transaction is expecting, but if all inputs are still present,

the transaction is guaranteed to succeed. This is in stark contrast to Ethereum, where transactions can fail mid-execution.



Understanding EUTXO requires an understanding of what a blockchain accounting model is, why it's needed, its functionality, and traits.

What is a blockchain accounting model?

Every company, firm, or commercial entity requires a balance sheet to keep an accurate record of profit, loss, cash flow, and other parameters. By maintaining careful accounting of all this data, companies can, at a glance, visualize their financial status at any given point in time. A company's accounting ledger offers another advantage: The ability to trace the provenance and ownership of funds.

Blockchain networks also require an accounting model to determine who owns what coins (and how many of them), track where those coins go, which ones are used up, and which ones remain available to be spent.

UTXO model vs Account/Balance model: a brief overview

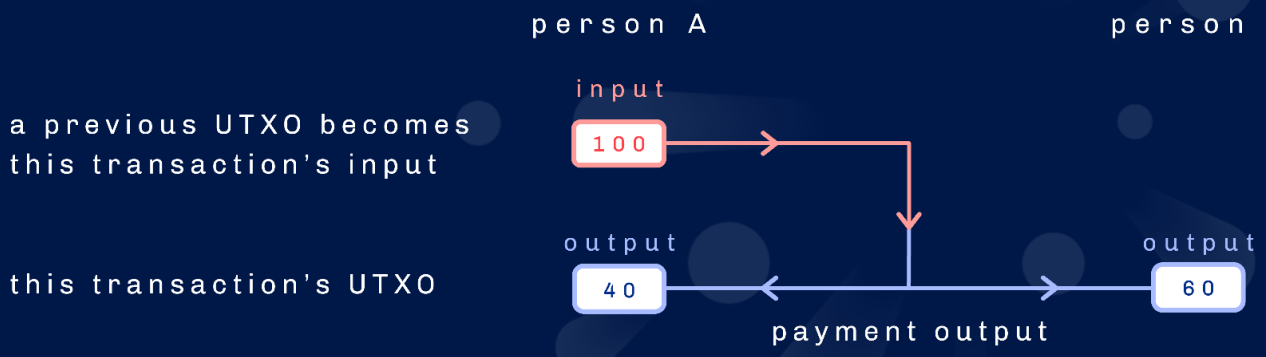
Decades ago, accountants used physical ledger books with handwritten entries to keep records about the movement of funds. Nowadays, companies use electronic versions of the same thing. Blockchains use transactions as records (much like entries on a ledger book) to track provenance and ownership. These transactions contain a lot of information (where the coins come from, where they're going, and whatever change is leftover from these transactions).

Here's a brief overview of the UTXO and Account/Balance models:

UTXO

In a UTXO model, the movement of assets is recorded in the form of a directed acyclic graph where the nodes are transactions and the edges are transaction outputs, where each additional transaction consumes some of the UTXOs and adds new ones. The users' wallets keep track of a list of unspent outputs associated with all addresses owned by the user, and calculate the users' balance.

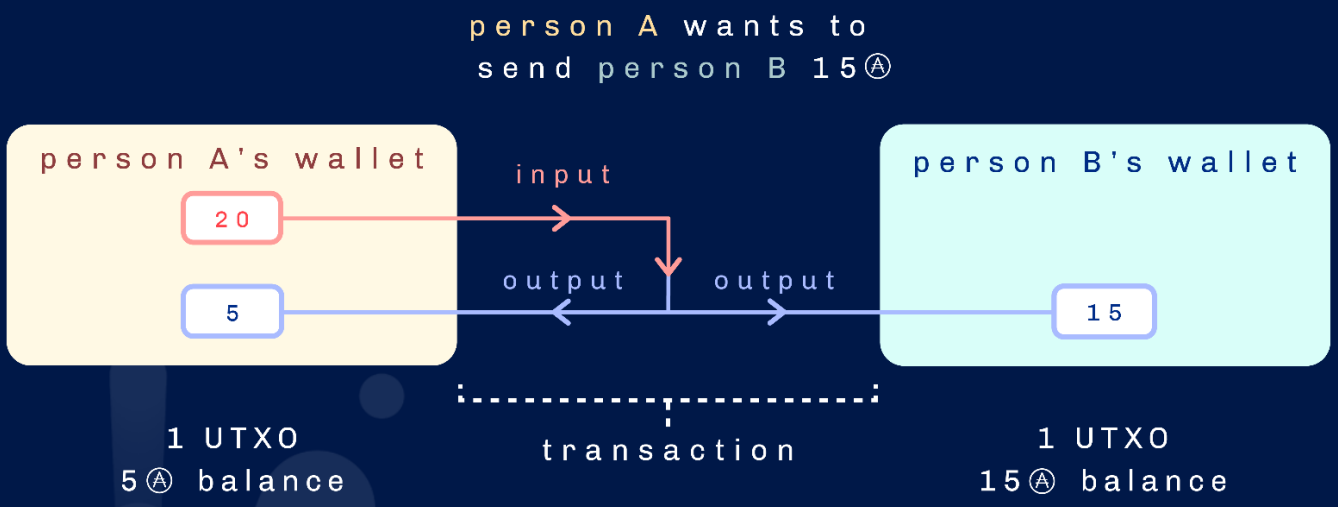
...each additional transaction consumes some of the UTXOs and adds new ones.



UTXO is, in many ways, similar to cash. A good analogy is this: imagine you have $50 in your wallet. This amount could be made up with several combinations: two $20 bills and one $10, four $10 bills and two $5 bills, and many others. But regardless of the permutations, the amount ($50) remains equal. UTXOs work in the same way. Whatever balance you have in your blockchain wallet (say, 150 coins) could be made up with many different UTXO combinations, based on previous transactions, but the balance amount remains the same. In other words, the balance held in a given wallet address is the sum of all unspent UTXOs from previous transactions.

The concept of 'change' in UTXO models

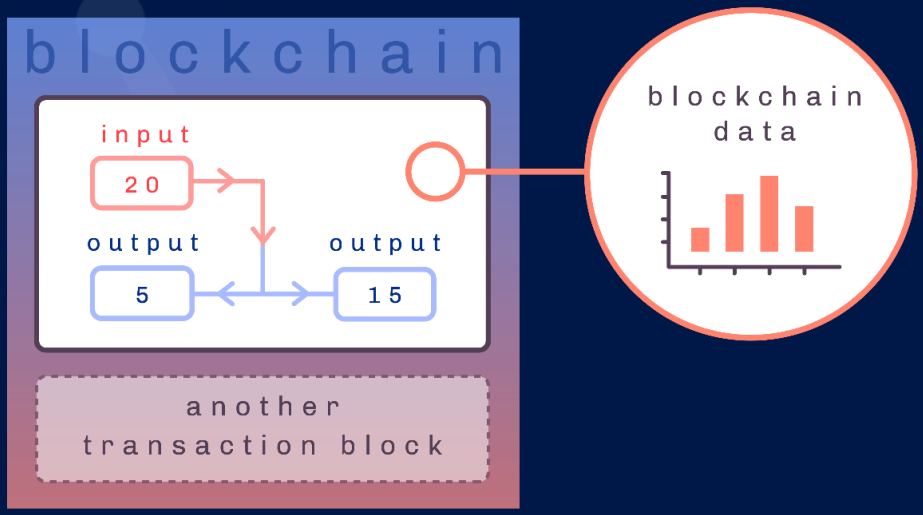
Much like cash transactions in any store, UTXOs introduce ‘change’. When you take out say a $50 bill from your wallet, you cannot tear that bill into smaller pieces to pay for something that costs $15, for example. You have to hand over the entire $50 bill and receive your change from the cashier. UTXOs work in the same way. You cannot ‘split’ a UTXO into smaller bits. UTXOs are used whole, and change is given back to your wallet’s address in the form of a smaller UTXO.



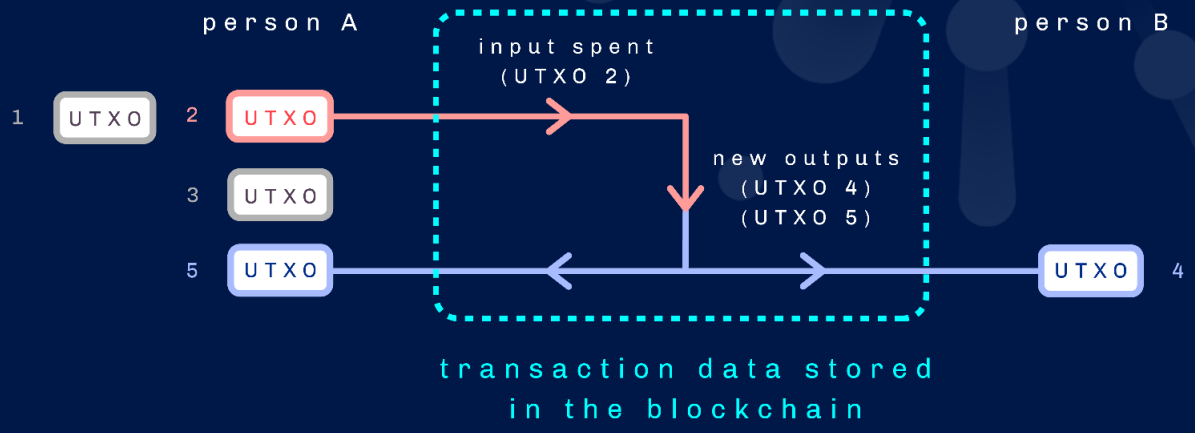
UTXOs are used whole, and change is given back to your wallet's address in the form of a smaller UTXO.

Advantages of the UTXO model

By checking and tracking the size, age, and amount of UTXOs being transferred around, one can extract accurate metrics about the blockchain's usage and financial activity of the chain.



UTXO models offer other advantages. Better scalability thanks to smart contract parallelization and privacy, for example. Also, the transaction logic is simplified, as each UTXO can only be consumed once and as a whole, which makes transaction verification much simpler.



To sum UTXO up:

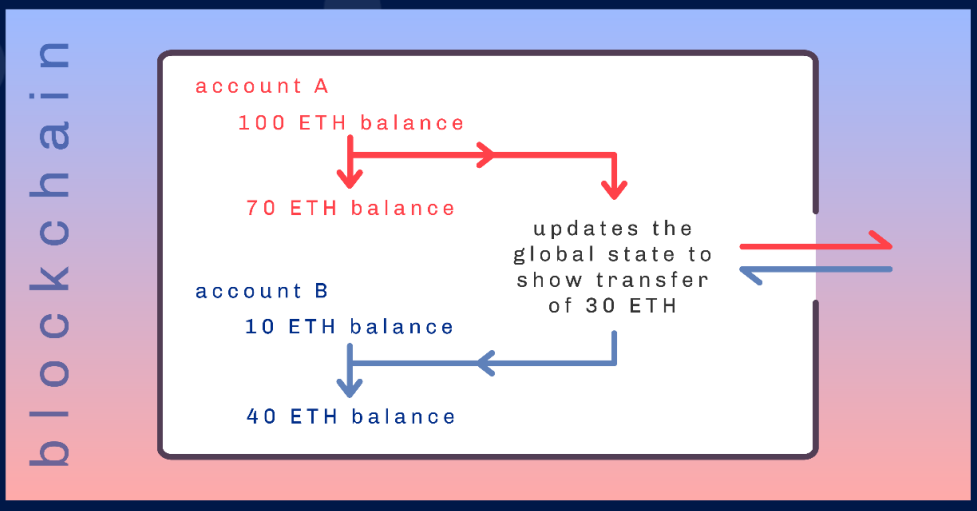
- A UTXO is the output of a previous transaction, which can be spent in the future
- UTXO chains have no accounts. Instead, coins are stored as a list of UTXOs, and transactions are created by consuming existing UTXOs and producing new ones in their place
- Balance is the sum of UTXOs controlled by a given address
- UTXOs resemble cash in that they use 'change', and are indivisible (UTXOs are used whole)

The Account/Balance model

As the name indicates, blockchain models that deploy an Account/Balance accounting model use an account (which can be controlled by a private key or a smart contract) to hold a coin balance. In this model, assets are represented as balances within users' accounts, and the balances are stored as a global state of accounts, kept by each node, and updated with every transaction.

In many respects, Account/Balance chains (such as Ethereum) operate in a similar fashion to traditional bank accounts. The wallet's balance increases when coins are deposited, and decreases when coins are transferred elsewhere. The crucial difference here is that, unlike UTXOs, you can use your balance partially. So for example, if you have 100 ETH in your account, you can send a portion of that (say, 30 ETH) to someone else. The resulting balance will be 70 ETH remaining in your account, and the address where you sent the coins to will increase by 30 ETH. The concept of change does not apply in Account/Balance accounting models as it does in UTXO ones.

Account/Balance chains operate in a similar fashion to traditional bank accounts.



To sum up the Account/Balance model:

- This accounting model resembles how a bank operates
- Users have accounts that hold their coin balance
- It is possible to spent partial balances
- The concept of change does not apply.

The EUTXO model

To understand EUTXO, it is important to understand how transactions work in Cardano. Particularly, the role of transaction outputs and inputs.

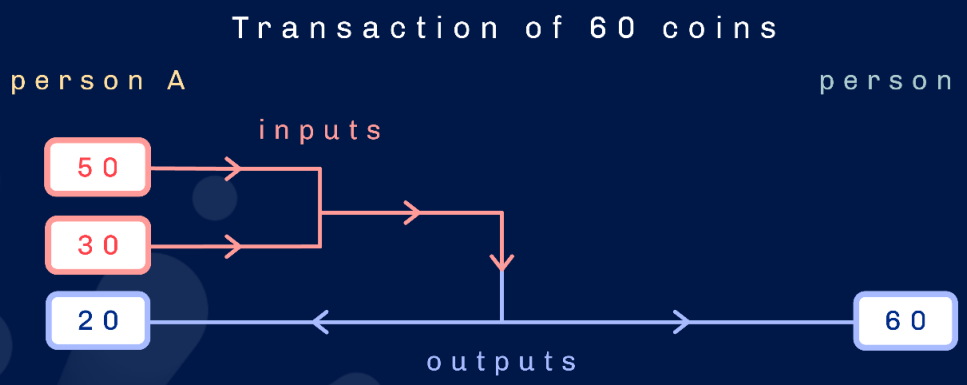
We need to talk about transactions: Outputs and Inputs

Think of a transaction as the action that unlocks previous outputs, and creates new ones.

The term *transaction* usually evokes financial echoes. While such meaning would apply to Bitcoin (since the Bitcoin blockchain is used to move funds between peers), many other blockchains (including Cardano) are far more versatile. In these cases, the term 'transaction' is much more nuanced. One can think of transactions as transfers of value.

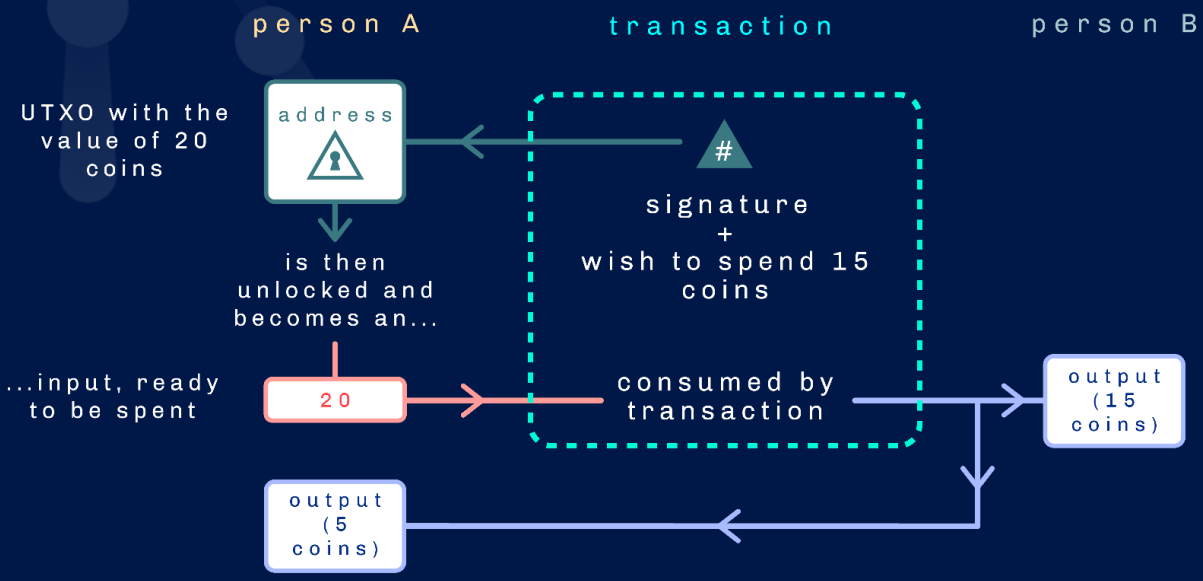
In a blockchain environment, each transaction can have one or multiple inputs, and one or multiple

outputs. The concepts of Inputs and Outputs must be understood, if one wants to understand how a transaction works, and how it relates to UTXO. In abstract terms, think of a transaction as the action that unlocks previous outputs, and creates new ones.



Transaction output

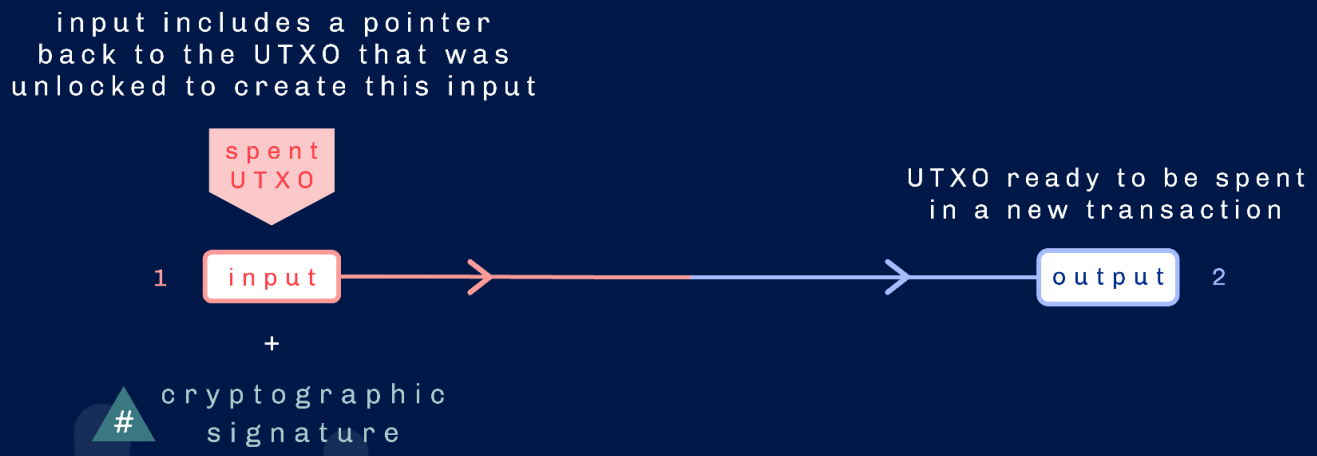
A transaction output includes an address (that you can think of as a lock) and a value. In keeping with this analogy, the signature that belongs to the address is the key to unlock the output. Once unlocked, an output can be used as input. New transactions spend outputs of previous transactions, and produce new outputs that can be consumed by future transactions. Each UTXO can only be consumed once, and as a whole. Each output can be spent by exactly one input, and one input *only*.



Transaction input

A transaction input is the output of a previous transaction. Transaction inputs include a pointer and a cryptographic signature that acts as the unlocking key. The pointer refers back to a previous transaction output, and the key unlocks this output. When an output is unlocked by an input, the blockchain marks the unlocked output as “spent”. New outputs created by a given transaction can then be pointed to by new inputs, and so the chain continues. These new outputs (which have not yet been unlocked, i.e., *spent*) are the UTXOs. Unspent outputs are simply that, outputs that have not yet been spent.

Transaction inputs include a pointer and a cryptographic signature that acts as the unlocking key.

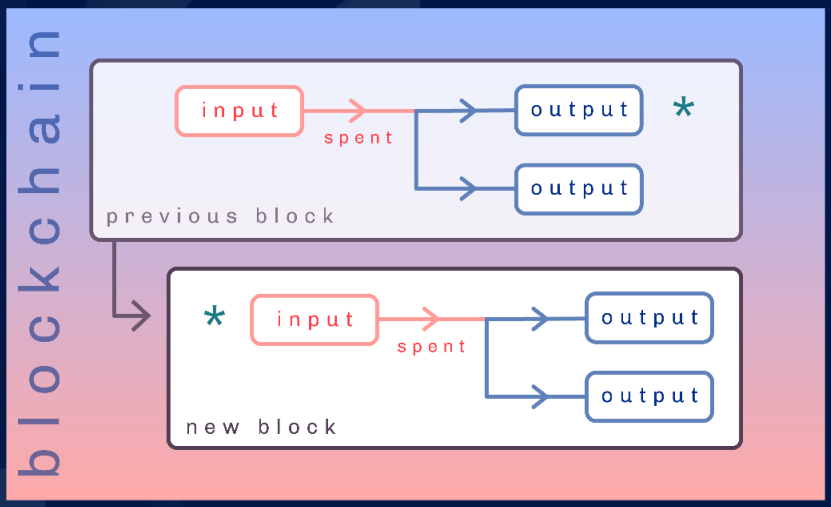


How UTXO works, in a nutshell

In a UTXO accounting model, transactions consume unspent outputs from previous transactions, and produce new outputs that can be used as inputs for future transactions.

Every blockchain node maintains a record of the subset of all UTXOs at all times.

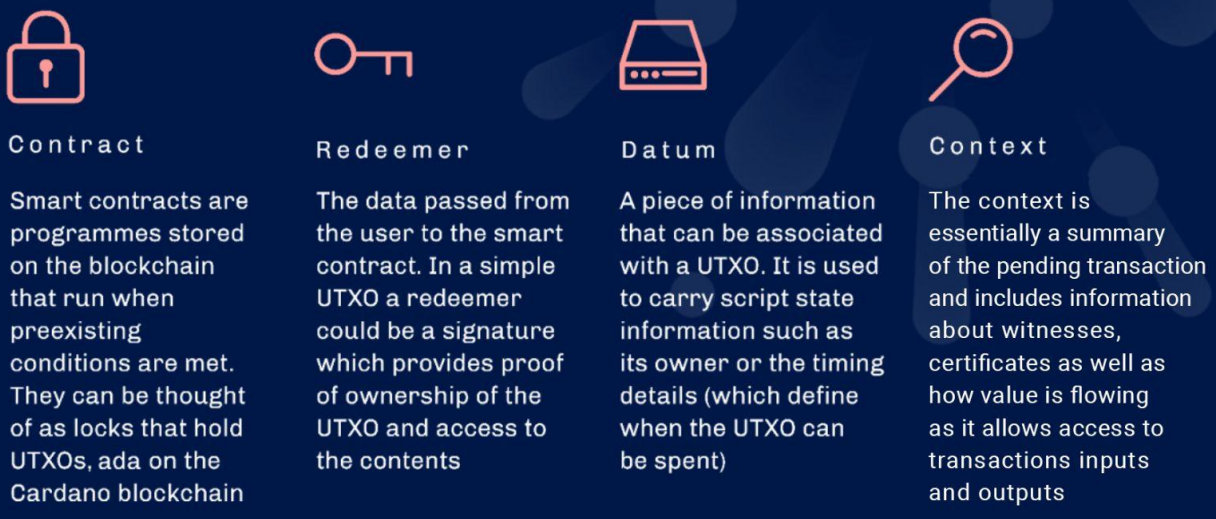
The users' wallets manage these UTXOs and initiate transactions involving the UTXOs owned by the user. Every blockchain node maintains a record of the subset of all UTXOs at all times. This is called the UTXO set. In technical terms, this is the chainstate, which is stored in the data directory of every node. When a new block is added to the chain, the chainstate is updated accordingly. This new block contains the list of latest transactions (including of course a record of spent UTXOs, and new ones created since the chainstate was last updated). Every node maintains an exact copy of the chainstate.



EUTXO is a transaction mechanism combining:

- Smart contracts: these lock-up UTXOs, ada, native assets, and NFTs.
- Redeemers: user-supplied data provided to unlock locked assets & spend them.
- Datum: data such as a high score, user information or other information relevant to your app

- Context: information like metadata about the transaction being validated.



EUTXO: the rationale behind Cardano's choice

Bitcoin's 'vanilla' UTXO accounting model would not suit Cardano, as Cardano is designed to do more than handle payments. Particularly, the need for more programming expressiveness for smart contract functionality required a novel ('Extended') solution.

The 'basic' UTXO model has a limited expressiveness of programmability. Ethereum's Account/Balance accounting model addressed this specific problem with the development of an Account/Balance ledger and associated contract accounts. But by doing so, the semantics of the contract code became far more complex, which had the unwanted effect of forcing contract authors to fully grasp the nuances of the semantics to avoid the introduction of potentially very costly vulnerabilities in the code.

An ‘extended’ UTXO solution would require two pieces of additional functionality that the existing UTXO model could not provide:

1 – To be able to maintain the contract state

2 – To be able to enforce that the same contract code is used along the entire sequence of transactions. This is called *continuity*.

A powerful feature of the EUTXO model is that the fees required for a valid transaction can be predicted precisely prior to posting it.

A powerful feature of the EUTXO model is that the fees required for a valid transaction can be predicted precisely prior to posting it. This is a unique feature not found in Account/Balance models.

Plutus Core

The implementation of EUTXO includes two key elements that differentiates it from an Account/Balance model: script and data. Scripts require a definite, well-specified scripting language, and it is also important to define the type of data

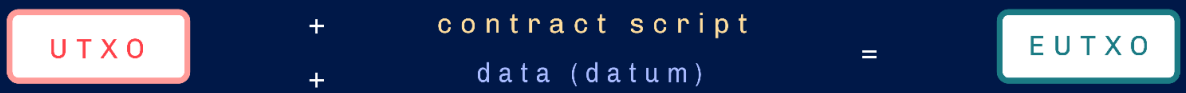
attached to outputs and used as redeemers. Redeemer data is a simple (algebraic) data type that can be easily defined in Haskell.

Plutus Core, Cardano's scripting language, provides these two elements. It is a simple and functional language similar to Haskell. Indeed, a large subset of Haskell can be used to write Plutus Core scripts. Developers do not write any Plutus Core code. A Haskell compiler plug-in generates all Plutus Core scripts.

The implementation of EUTXO includes two key elements that differentiates it from an Account/Balance model: script and data.

Nodes execute these scripts during transaction validation 'live' on the chain. The scripts can lock UTXOs in the form of validator scripts or as minting policies, which control the minting and burning of native tokens.

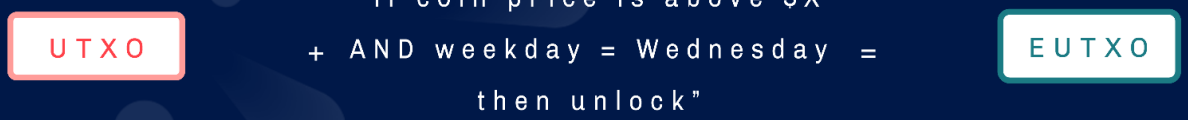
Appropriate Haskell libraries simplify writing such validation logic by providing core data types for the inspection of transactions during validation, and by offering many helper functions and higher level abstractions. This allows contract authors to concentrate on the business logic and not have to worry about too many low-level details.



How does the EUTXO model extend UTXO?

EUTXO extends the 'basic' UTXO model in two directions:

1. It generalizes the concept of 'address' by using the lock-and-key analogy. Instead of restricting locks to public keys and keys to signatures, addresses in the EUTXO model can contain arbitrary logic in the form of scripts. For example, when a node validates a transaction, the node determines whether or not the transaction is allowed to use a certain output as an input. The transaction will look up the script provided by the output's address and will execute the script if the transaction can use the output as an input.



2. The second difference between UTXO and EUTXO is that outputs can carry (almost) arbitrary data in addition to an address and value. This makes scripts much more powerful by allowing them to carry state.

Here, we present what makes EUTXO unique, and why this accounting model is superior to the 'vanilla' UTXO.

EUTXO features	Why is this important
Each UTXO has an address, a value, *and* a datum, which is a piece of contract-specific data. Datum is also known as a *datum object*. The datum is passed in as an additional argument during validation.	It allows a contract to carry some state (the datum) without changing its code.
The validator receives context, which is information about the transaction being validated. This information is passed in as an additional argument of type Context.	The information supplied in the context enables the validator to enforce much stronger conditions than possible with a 'bare' UTXO model. In particular, it can inspect the outputs of the current transaction, which is essential for ensuring contract continuity.
EUTXO provides a certain degree of access to time by adding a validity interval to transactions. This is an interval of *ticks* (defined as	This interval enables any script running during validation to assume that the current tick is within that interval, but the script does

a monotonically increasing unit of progress in the ledger system. This usually corresponds to the block number or block height) during which a transaction can be processed.	not know the precise value of the current tick.

Why is EUTXO superior to UTXO?

- Cardano's decentralization and security improves over those of Bitcoin. Cardano also offers smart contract features superior to Account/Balance chains.
- Multi-asset and smart contract support.
- Greater flexibility, security, stability, and scalability.
- More concurrent transactions on a proof of stake system drawing just a tiny fraction of the energy required by proof of work blockchains.
- IOG has teamed up with seven other UTXO-based blockchains under the UTXO Alliance to push the boundaries of innovation and interoperability, and improve it further, together.

EUTXO as a springboard to scale Cardano in 2022

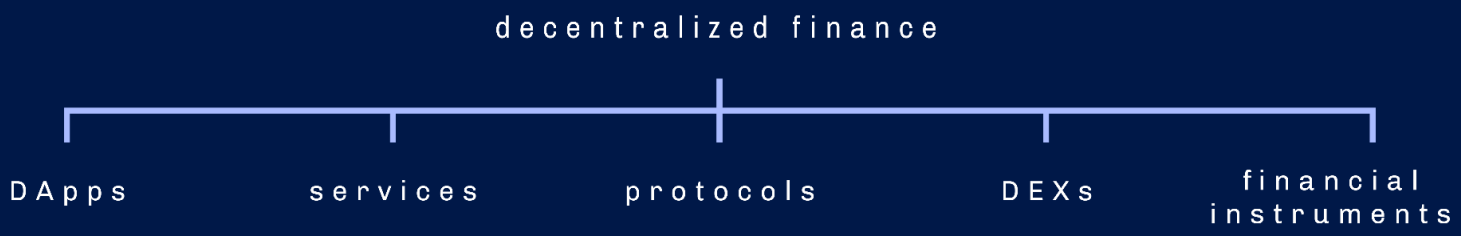
While it does present a new paradigm and way of thinking, EUTXO offers smart contract developers a rather powerful and versatile platform to construct and deploy their applications.

This year, Input Output Global Inc. is leveraging the power of EUTXO to optimize the smart contracts built on Cardano in three ways:

- **Reference inputs** (CIP-0031) – Plutus scripts can inspect transaction inputs without needing to spend them. This means that it is not necessary to create UTXOs simply to inspect the information held by an input
- **Plutus Datums** (CIP-0032) – Datums can be attached directly to outputs instead of datum hashes. This simplifies how datums are used, as a user can see the actual datum rather than having to supply the datum that matches the given hash
- **Script sharing** (CIP-0033) – Plutus script references can be associated with transaction outputs, meaning that they can be recorded on-chain for subsequent reuse. It will not be necessary to supply a copy of the script with each transaction, hugely reducing friction for developers. Reusing scripts in multiple transactions significantly reduces transaction sizes, improving throughput and reducing script execution costs.

Everything you always wanted to know about impermanent loss and were afraid to ask

Decentralized Finance (DeFi) is an umbrella term that refers to decentralized applications (DApps), services, protocols, and financial instruments built on blockchain. It is a relatively new industry segment enabled by decentralized ledger technology, which means there is no single authority with centralized control over the system. And anyone familiar with the DeFi environment probably knows about impermanent loss. It's a simple concept with a misleading name.



Cardano is a third-generation blockchain whose expansive DeFi universe features, among many other DApps, decentralized exchanges (DEXs). These are crypto exchange protocols that enable peers to trade cryptocurrencies with each other. DEXs use two main design architectures: *automated market maker* (AMM) and *order book*. The implementation of AMMs is relatively simple, and this design has since become the de facto choice forAccount/Balance chains, including Ethereum. However, this design has some inherent deficiencies. Their tendency to incur impermanent loss, for example.

Decentralized exchanges (DEXs) are crypto exchange protocols that enable peers to trade cryptocurrencies with each other.

Cardano uses the EUTXO accounting model to track the movement of assets across the chain. EUTXO is deterministic, which offers better predictability of impermanent loss.

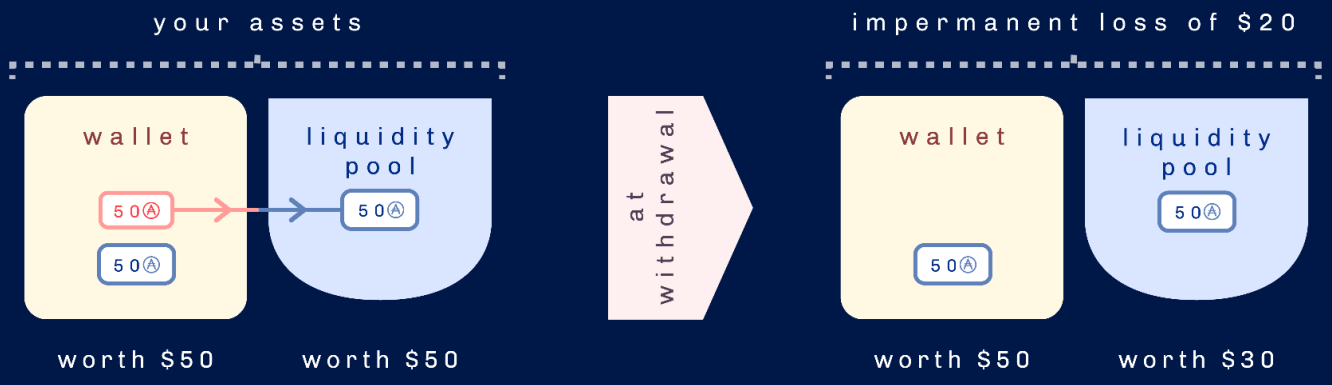
These seemingly unrelated concepts coalesce into a very interesting interplay on the blockchain. This article considers DEX designs and explains why EUTXO offers better predictability of impermanent loss than Account/Balance accounting models.

Impermanent loss: definition

When the total value of assets provided as liquidity is lower than the value that would have accrued had you simply held onto them.

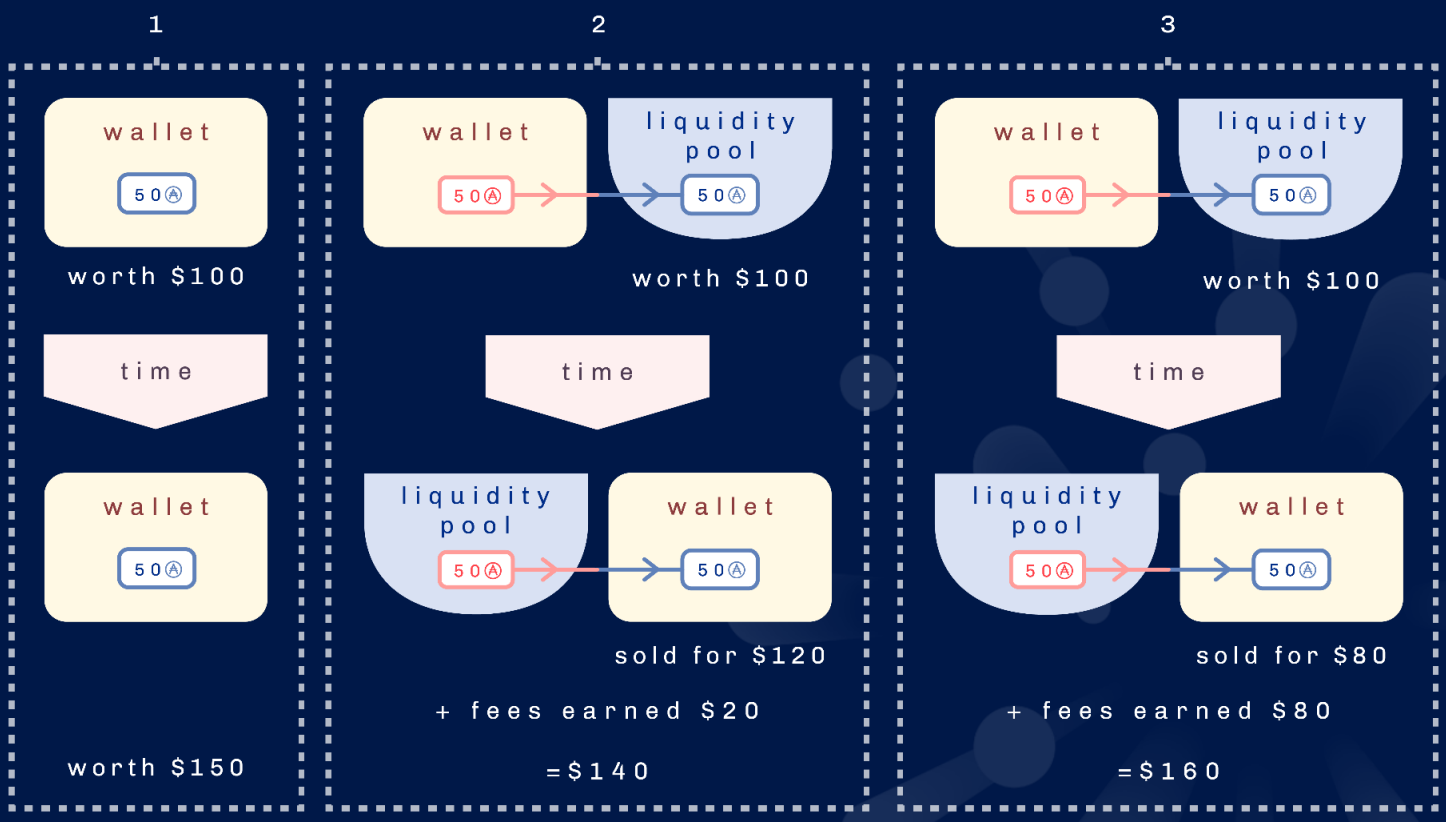
This is the simplest definition of impermanent loss, a concept that inspires dread on liquidity providers.

Impermanent loss occurs when the price of the assets deposited into a liquidity pool changes (upwards or downwards) in relation to when they were deposited. In other words, the worth of your assets when you withdraw them is different to when you deposited them into the liquidity pool.



The name *impermanent* is slightly misleading, as a decrease in token price might only be temporary, and the price might rise again following market or trading conditions, etc. In this case, the loss would be *temporary* (i.e., impermanent), because the price rectified upwards. The change becomes *permanent* only if the dollar price of the token at withdrawal is less than it was when the token was deposited.

One could argue that impermanent loss is the risk that liquidity providers take in exchange for fees earned by trading crypto pairs on liquidity pools. If the loss is greater than the fees earned, the liquidity provider realizes a loss, which might not have happened had they held onto their tokens instead. It is interesting to note that one might not actually lose money, but one's gains might be *less* than if one had just held the tokens.



AMMs vs order-book

Understanding impermanent loss requires a basic understanding of how DEXs work. Currently, DEXs use two design models: AMM and order book. Each comes with a set of advantages and disadvantages when it comes to impermanent loss, which are explored below.

AMM

The Automated Market Maker (AMM) DEX mode enables automated trading of cryptocurrency pairs using smart contracts. These pairs are usually (but not always) an Ethereum-based token and a stablecoin.

AMMs rely on liquidity pools, which are mechanisms that facilitate users to pool their assets into smart contracts. The more liquidity there is in the pool, the easier it becomes to trade on the DEX the pool is associated with, and the higher the fees and rewards earned by liquidity providers. Liquidity pools aggregate the liquidity provided by investors into both sides of the trading pair. The pool uses an algorithm that looks at the current liquidity to calculate the pair's market price at that time. To put it another way, the algorithm considers the availability of a particular asset in the pool to determine its price.

AMMs rely on liquidity pools, which are mechanisms that facilitate users to pool their assets into smart contracts.

AMMs rely almost entirely on liquidity providers to provide liquidity to expand the pool's size and ensure the assets are traded at a fair price. This design trait effectively means that the liquidity providers are the market makers.

Liquidity providers need an incentive to invest, of course. This comes in the form of yield farming, essentially token rewards earned through lending or staking of digital assets.

Order book

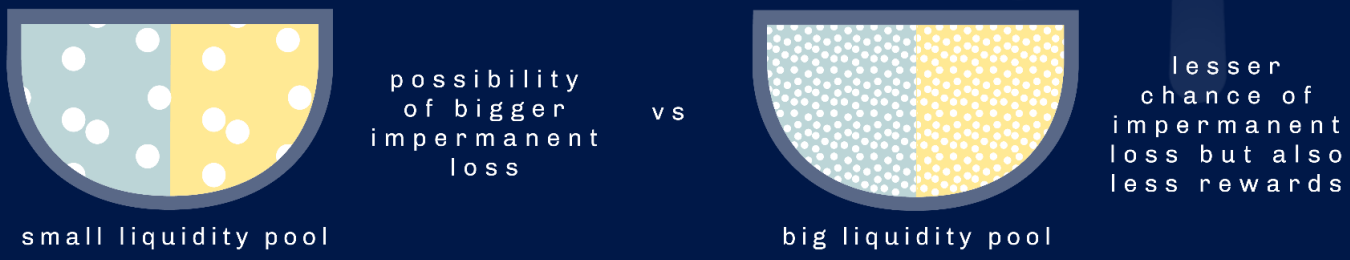
The mechanics behind the order book design have been around the economics field for a long time. It is a very straightforward model. The order book simply lists all the buy/sell (asks/bids, in this context) orders, so when the traders put in their orders, the order book sorts them according to the asset's price. If there is supply and demand, the asset can be traded.

UTXO-based ledgers, like Cardano, are far more suitable for order book architecture, as this design, together with Cardano's EUTXO features, mitigates the effects of impermanent loss.

UTXO-based ledgers, like Cardano, are far more suitable for order book architecture.

The (un)predictability of impermanent loss

Liquidity providers provide liquidity to pools for financial returns. But this brings with it risk. The amount of tokens in the pool and number of liquidity providers contributing to it are major factors on the *possibility* of impermanent loss occurring, and such consideration is important for potential liquidity providers. Frequent impermanent loss leads to pools drying up and liquidity providers looking elsewhere.



Here's the insidious thing about impermanent loss: it is very difficult to predict whether or not it will occur, and to what degree.

Impermanent loss in UTXO-based chains vs Account/Balance ones

Quick intro:

- UTXO-based chains: there are no accounts holding a balance. Instead, users' wallets keep track of a list of unspent outputs associated with all addresses owned by the user, and calculate the users' balance. UTXO is, in many ways, similar to cash transactions. Cardano 's EUTXO model adds a *datum*, which is contract-specific data. This is important as it confers Cardano with the ability to support multi-assets and smart contracts.

- Account/Balance model – This accounting model uses an account (which can be controlled by a private key or a smart contract) to hold a coin balance. In this model, assets are represented as balances within users' accounts, and the balances are stored as a global state of accounts. The state is kept by each node and updated with every transaction.

There are several fundamental differences between these two models, but when it comes to AMMs and impermanent loss, there is one key distinction. AMMs operating on Account/Balance chains (like Ethereum) tend to use a Constant Formula Market Maker (CFMM) pricing formula, which is one of the more commonly used algorithms for AMMs. This formula contains inherent inefficiencies. For example, the Total Value Locked (TVL) -defined as the sum of all staked crypto assets earning rewards, interest, etc.- is distributed across the entire price range, which implies that the price of an asset is equally likely to be $1 or $10,000. Under this assumption, CFMM prices are unrealistic and tend not to reflect actual market conditions. Also, trades on low token volume tend to lead to high slippage (the difference between the expected price of an order and the price when the order actually executes.) While CFMM is a popular choice for AMMs, these inefficiencies might result in the dilution of revenues for liquidity providers. More importantly, this liquidity is subject to impermanent loss.

While CFMM is a popular choice for AMMs, these inefficiencies might result in the dilution of revenues for liquidity providers.

EUTXO and order book DEX design as the bulwark against impermanent loss

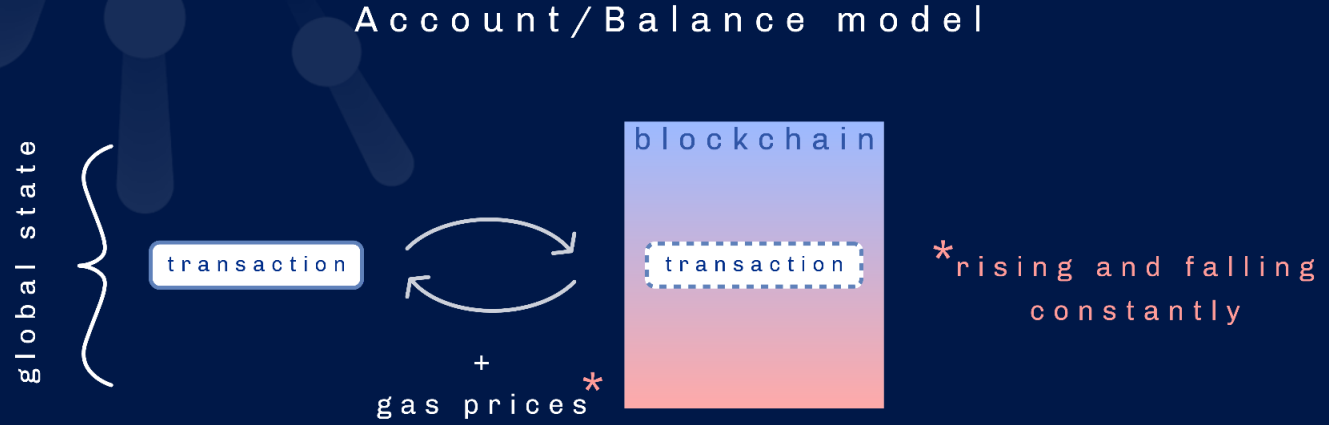
EUTXO architecture's inherent advantages of security, determinism, parallelism, and scalability offer an ideal environment for DEXs using order book design, as it presents stronger resilience to impermanent loss. One key advantage of this design is concentrated liquidity (liquidity that is allocated within a custom price range.) This feature maximizes the liquidity's efficiency and minimizes impermanent loss.

Why global state is not an issue in EUTXO-based chains

Unlike Account/Balance blockchains where every single transaction outcome alters the global state, in UTXO-based blockchains, the validity of a transaction is assessed at the transaction level, and the balance is the sum of remaining UTXOs. At the local state, in other words.

A transaction's gas fees might spike significantly in the interval between the transaction being submitted and validated.

This immediately poses a problem for Account/Balance chains. A multitude of smart contracts and other actors continuously interact and influence the global state, which means that assets and resources are consumed, and gas prices rise and fall all the time. A side effect of this is that transaction fees can (and do) fluctuate. Effectively, this means that a transaction's gas fees might spike significantly in the interval between the transaction being submitted and validated. Consequently, such a transaction might not be accepted by the chain, but the gas fees are taken anyway, potentially leading to financial loss for the user. This is one of the Ethereum chain's main design flaws.



Fee wastage cannot occur in Cardano's EUTXO model, since transactions are processed and validated at the local state. This is achieved by adding a datum (additional data) to the transaction. The datum contains contract-specific information, which is passed to the transaction's validation logic, thus maintaining EUTXO's deterministic context. This effectively means that transaction fees are known in advance, and will not change. A welcome side effect of EUTXO and determinism is that transactions cannot be rearranged by bad actors, another risk of Account/Balance models.

Fee wastage cannot occur in Cardano's EUTXO model, since transactions are processed and validated at the local state.

EUTXO model

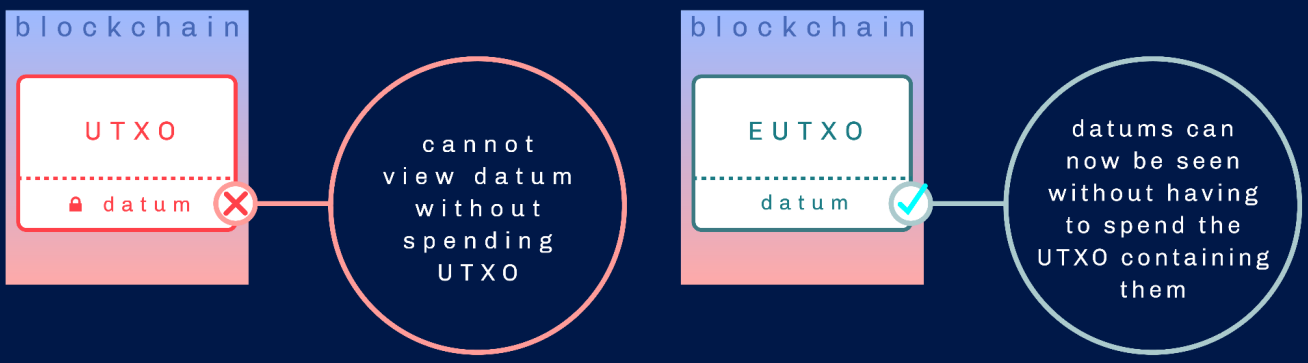
The local nature of transaction validation offers yet another significant advantage: a high degree of parallelism. A node could validate transactions in parallel, as long as those transactions do not attempt to consume the same input. This cannot be done in Account/Balance chains, as transactions must be processed sequentially by design.

Further enhancements

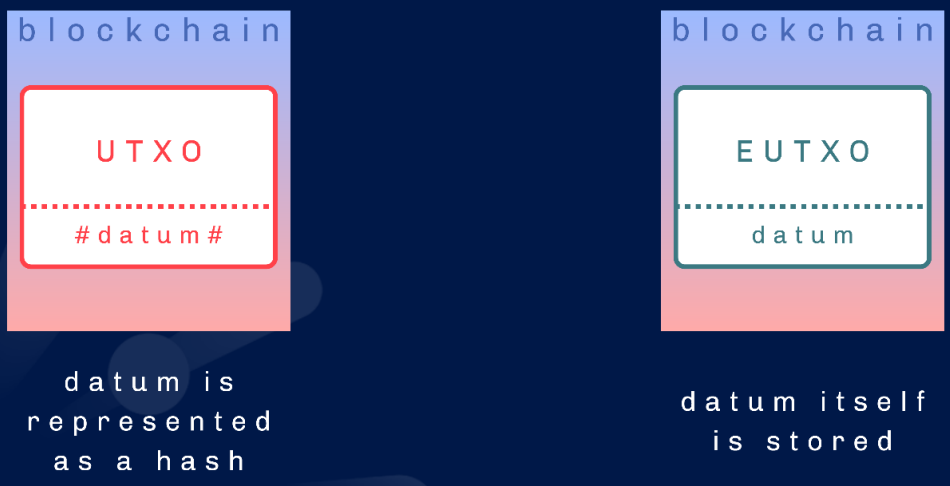
The Plutus platform provides a native smart contract language for the Cardano blockchain.

Upcoming Cardano Improvement Proposals (CIPs) to Plutus include:

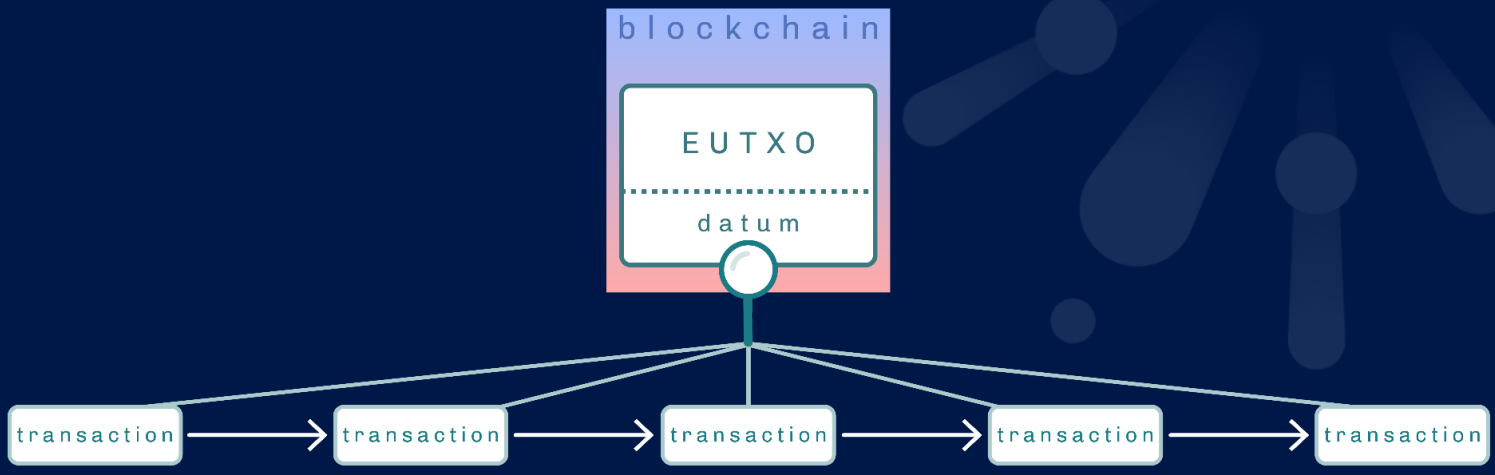
- CIP-31: Reference Inputs



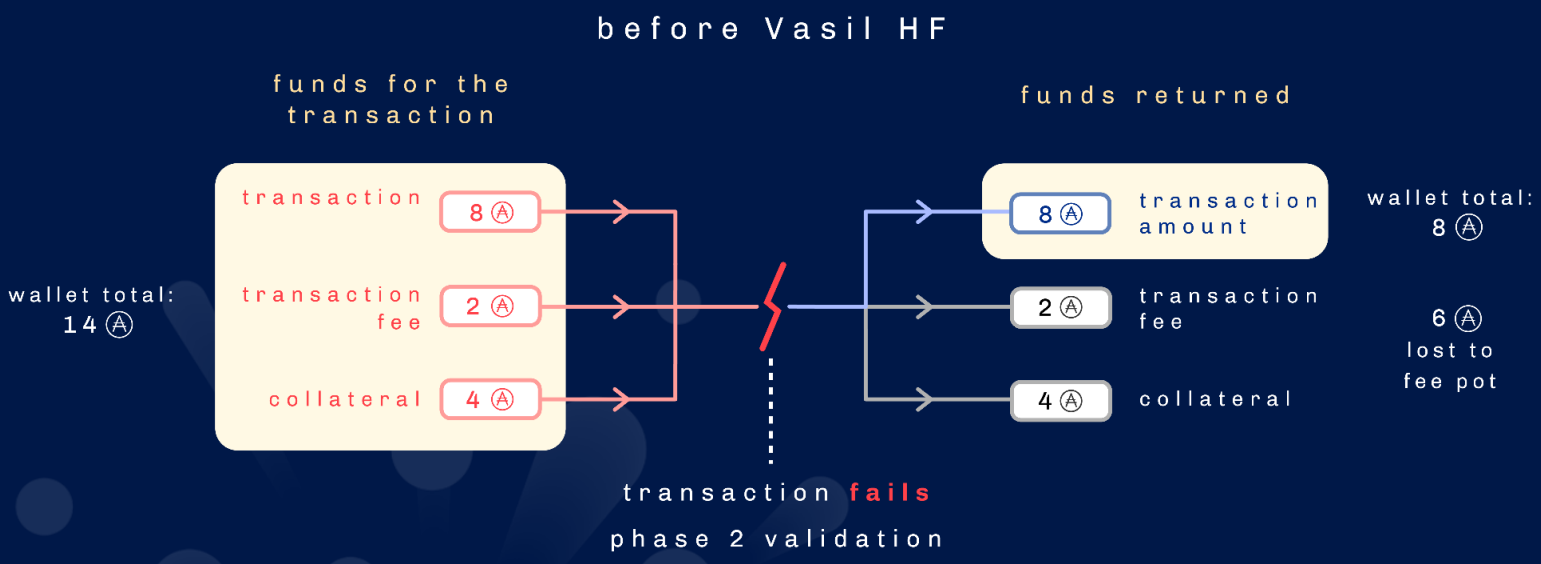
- CIP-32: Inline Datums

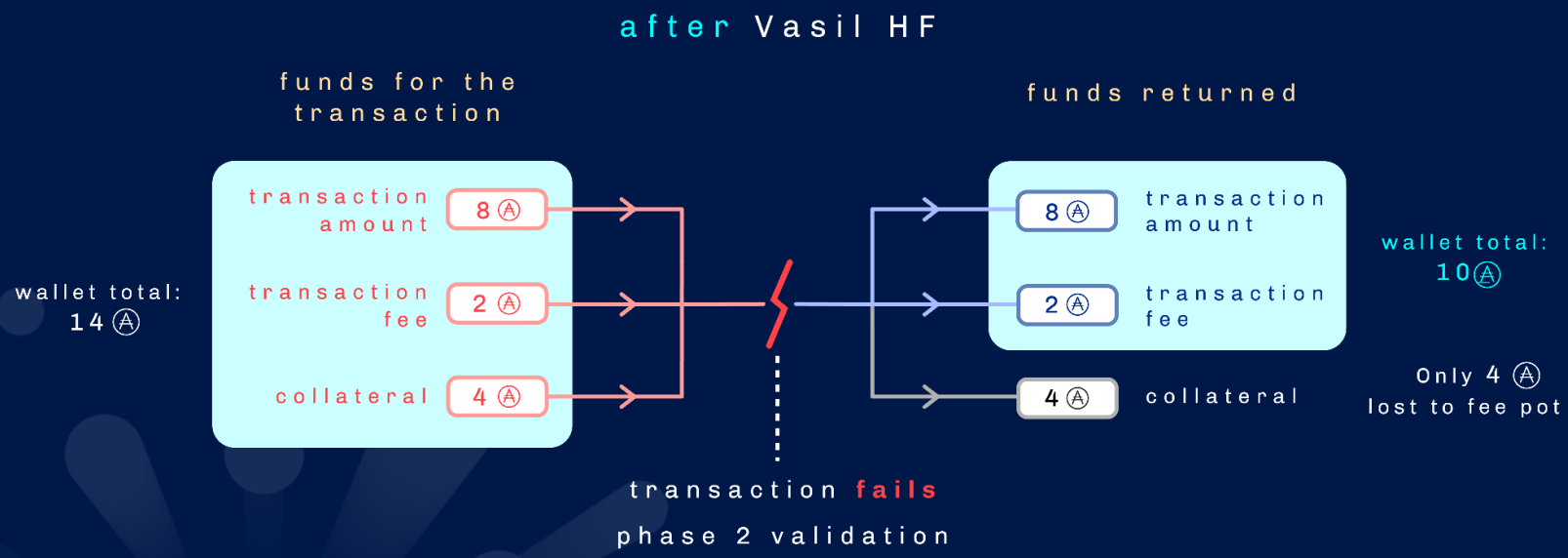


- CIP-33: Reference Scripts



- CIP-40: Collateral Outputs





Conclusion: what makes the EUTXO model innovative and relevant?

Cardano's ledger model extends the UTXO model to support multi-assets and smart contracts without compromising the core advantages of a UTXO model. Our innovative research enables functionality beyond what is supported in any other UTXO ledger, making Cardano a unique competitor in the next-generation blockchain space.

Further reading

Learn more about Cardano's EUTXO model with these sources:

EUTXO whitepaper

Cardano's Extended UTXO accounting model – built to support multi-assets and smart contracts