Understanding Unspent Transaction Outputs in Cardano

emurgo.io

Unspent Transaction Outputs (UTxOs) are an important concept for new Cardano developers to understand. Every transaction on the Cardano Settlement Layer (SL) has at least one input and at least one output. The amount of funds in your wallet is the sum of all your UTxOs and those UTxOs will become the inputs for future transactions; the blockchain records the collective history of those transactions.

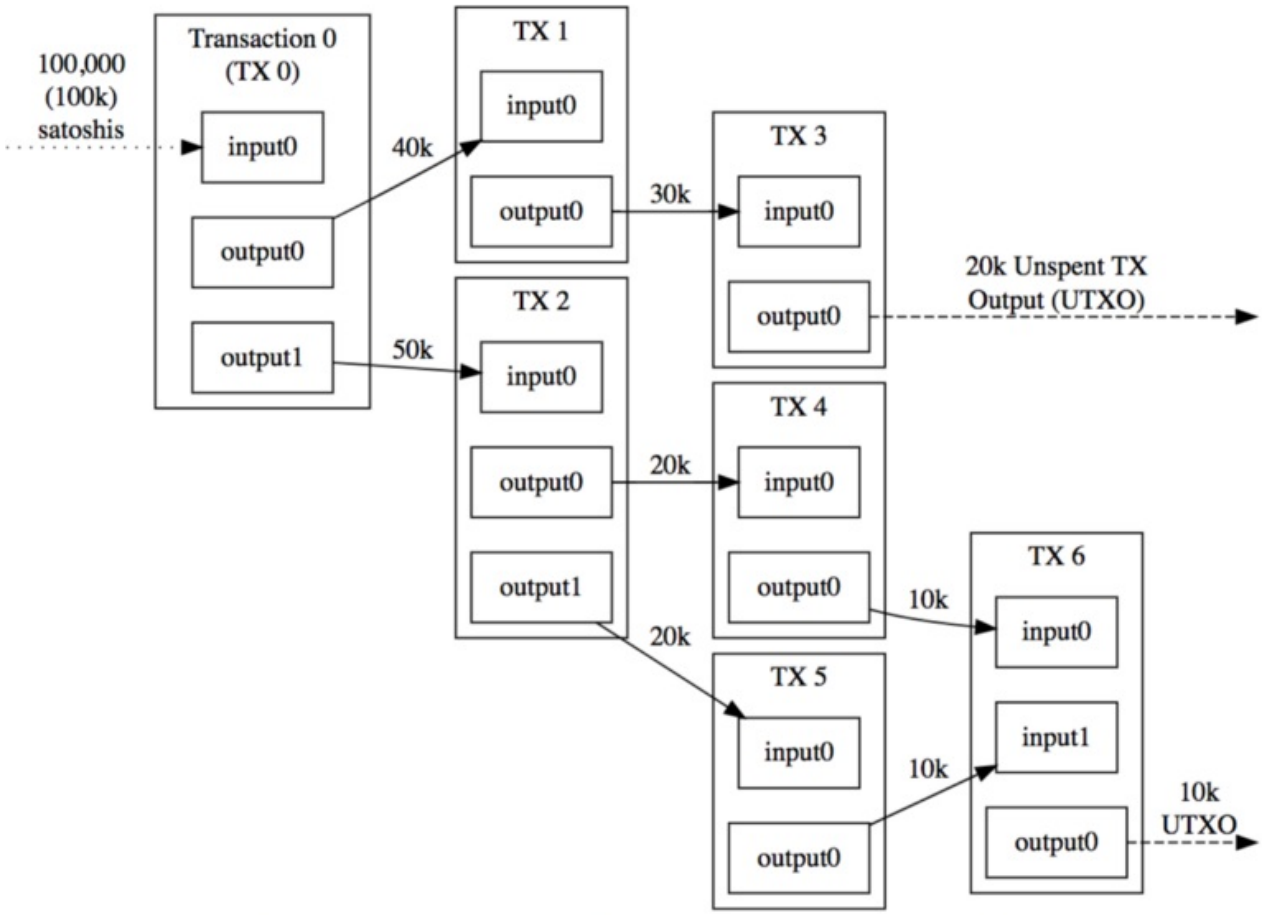
The Cardano Settlement Layer documentation on transactions sums it up quite nicely, "Inputs and outputs carry information about *money flow*: inputs inform where the money came from, and outputs inform where the money [goes] to." The UTxO model was introduced by Bitcoin.

For example, let's say you have a Cardano wallet with 10,000 ADA and you want to send 3,000 ADA to your spouse. For the sake of simplicity, in this example, we will use whole numbers and forget about network fees normally incurred by transactions. Now let's assume that your underlying total amount of ADA in your wallet is actually composed of three separate UTxOs that add up to 10,000 ADA. You might have 4,000 ADA in one UTxO, 2,000 ADA in another UTxO, and 4,000 ADA in another.

The transaction will need to send 3,000 ADA to your spouse from one of the UTxOs containing 4,000 ADA and it will also need to send 1,000 ADA back to your wallet as a "change address." The process of figuring out which UTxOs to use in a transaction is called the Coin Selection Algorithm — originally created for Bitcoin. Alternatively, if you had 10 separate UTxOs all containing 1,000 ADA each, then you would only need to create a transaction with an input index that contained three UTxOs.

This seems, initially, somewhat unconventional when we think of how traditional bank accounts work; if you want to send someone $3,000 the amount is simply subtracted from your $10,000 total balance. This traditional model is known as the "account based" model and is used by a variety of other cryptocurrencies, including Ethereum.

In the UTxO model, an input contains a transaction ID, which is a BLAKE2b-256 hash of the transaction and an index of the outputs using the transaction. An output contains the amount of money being sent as well as an address where we want money to be sent to, which is just a BLAKE2b-256 hash of the addresses public key.



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

Cardano will use a combination of both the UTxO and Accounting Based models. An Extended UTxO model will exist on the Settlement Layer and the Accounting Based model will exist on the Computation Layer. Because Cardano will use both models it is called a "Chimeric Ledger."

Sebastien Guillemot talks at length about the original Chimeric Ledger paper, written by IOHK ghostwriter Joachim "Zahnentferner," and helps to simplify, and understand, the underlying concepts in his video. You may want to watch Guillemot's video and return to this article for added context. Cardano will either use transaction translations, or a new hybrid transaction type, to execute transactions, and transfer funds, between the UTxO ledger and the Accounting Based ledger(s).

Translating between Account-to-UTxO transactions is fairly straightforward. However, UTxO-to-Account translations is a little more difficult. They can be executed in maximum of O(n+m) transactions, although the code is not guaranteed to obtain the smallest possible number of transactions.Zahnentferner says, "Whenever a UTXO-based transaction has more than one sender or more than one receiver, it cannot be translated into a single equivalent account based transaction. But it can be translated into a list of account-based transactions." The only issue with this conversion is a lack of atomicity. If you remember back to your Database 101 class, an atomic transaction is one that is either committed or rolled back in totality. This issue can be avoided by introducing a new wrapper type for UTxO-to-Account based transactions or using the aforementioned hybrid transaction type.

There are two primary benefits to the, stateless, UTxO model. Firstly, privacy is increased when change addresses and new unspent outputs (i.e., the money you sent to someone else) are close to 1:1. Distributed apps will be unable to store user data in the same way a bank account, or accounting based model, like Ethereum, would, because every new transaction uses a new address. The second benefit is in terms of scalability. Transactions can be processed in a parallel fashion.

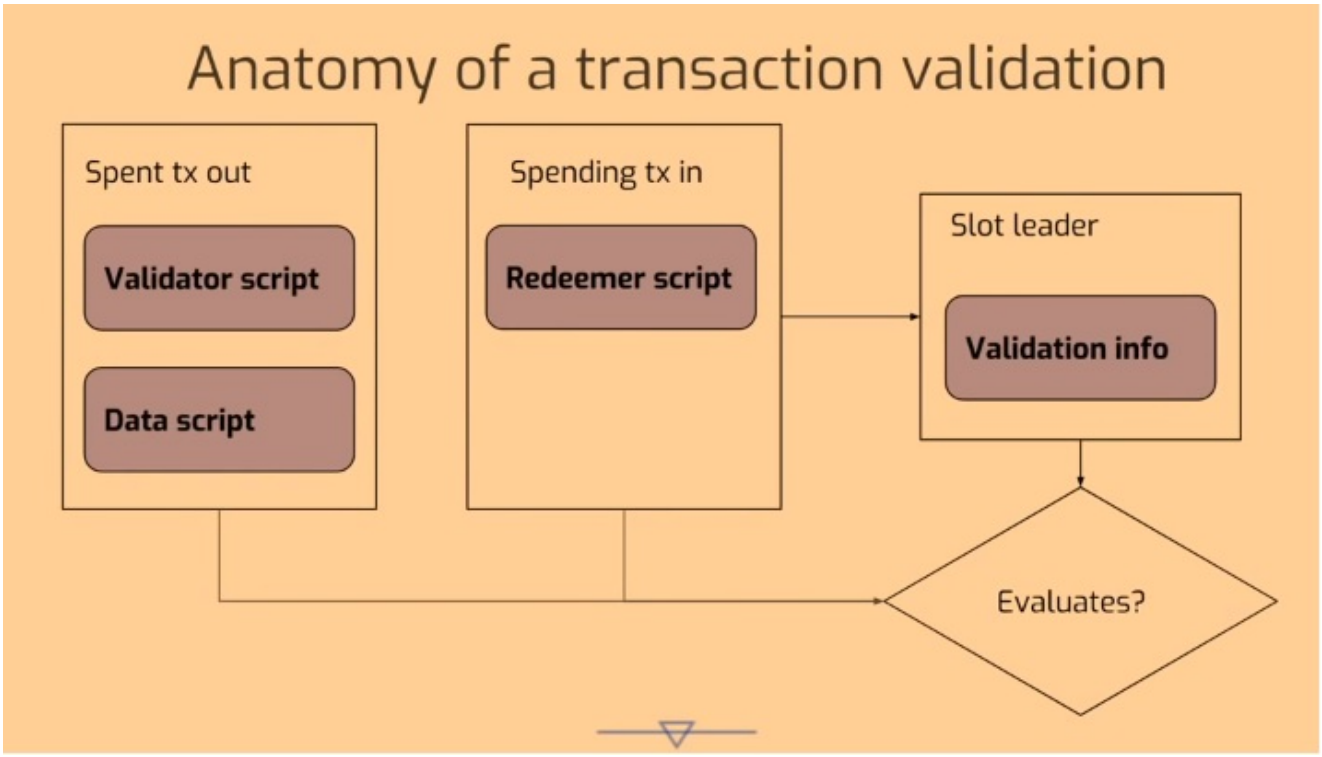
A third advantage is that state does not have to be *managed* on a global scale. This is important for when you want to execute smart contracts on Plutus, which actually uses the Cardano Settlement Layer. Eventually, the Plutus Core will be integrated into the Computation Layer, however.

How will the Computation Layer (which is a sidechain) and the Settlement Layer (the main chain) interact and allow transactions between each other? The process is defined through both *cross-chain certification* and *ad-hoc threshold multisignatures*. This is a new technology that Cardano will be introducing into the cryptocurrency space in the Goguen release. Although this is beyond the scope of this particular post, you can learn more in the Proof-of-Stake Sidechains paper. The underlying implementation will use the aforementioned Chimeric Ledger properties, however this is still being developed as of April 2019.

The stateless nature of UTxOs fits nicely with functional programmability, which can be formally verified. One argument against UTxOs is that they make smart contracts complex, but this is in the context of a traditional, imperative, programming paradigm. Plutus uses a stateless architecture, which is modeled on Haskell. Traditional programmers, who are used to the imperative model of languages like C/C++, Javascript, ASP, PHP, etc., will benefit by diving into Haskell in order to learn how to write Plutus contracts. In the imperative model, programmers are used to managing state, on both a local and global level.

In his speech at a recent Coinscrum conference, Michael Peyton explains, "We can still model things that are stateful. In the Ethereum model you change the state of something directly, here it's more the case that we need to keep track of our state as we go, and pass it along, which is often how we deal with state in a functional programming world."

In order to support smart contracts on Plutus, which runs on the Cardano Settlement Layer (SL), Cardano wallets use an Extended UTxO Model. This extended model is a creative concept in blockchain technology. It has two primary components, which include extra data carried by traditional UTxOs and additional wallet backend structure that helps facilitate off chain code involved in on-chain code execution.



Anatomy of a transaction validation

What that really means is that wallets will need to support scripts in order to create the smart contract space. Outputs are locked by validator scripts and inputs are unlocked with redeemer scripts. Robert Kornacki simplifies the process in more detail in his video Plutus Contracts in a Nutshell.

In addition to the usual UTxO set, validator/redeemer scripts are "interested in additional addresses and a wallet with scripts needs to maintain an additional UTxO set, which we call the *script UTxO*." Script addresses were introduced in the Byron release. Cardano, and the underlying UTxO system, is evolving with each release.

Shelley

Shelley has introduced four new address/UTxO types: base addresses, pointer addresses, enterprise addresses, and reward account addresses. An accounting based model is used for the staking rewards addresses.

The Design Specification paper for Shelley explains the new types, "A base address directly specifies the staking key that should control the stake for the address. A pointer address indirectly specifies the staking key that should control the stake for the address. It references a stake key by a stake key pointer which is a location on the blockchain of the stake key registration certificate for that key. Enterprise addresses carry no stake rights whatsoever and thus using them allows completely opting out of participation in the proof of stake protocol. Exchanges or other organisations that control large amounts of ada – but hold it on behalf of other users – may wish to follow a policy of not exercising stake rights. Reward account addresses are used to distribute rewards for participating in the PoS protocol."

The design specification paper has only recently been released, but future articles from EMURGO on the subject of staking will explain this in more detail.

Stake Amplification

PoSv3 cryptocurrencies are based on proof-of-work systems and can suffer from users attempting to increase their *apparent stake* and increase their chances of being selected to create a new blockUsers do this through self-spending and increasing their UTxO count. Fortunately, Cardano does not suffer from this issue as it is based on a rigorous peer reviewed development process and a completely different proof-of-stake system.

Dust

Dust is the term for when small UTxOs gather in a user's wallet over time. These UTxOs are so small that it costs more in transaction fees than they are worth to send over the network. The distribution of these UTxOs depends upon the underlying coin selection algorithm used by the wallet. Sebastien Guillemot has extensively covered the effects that different coin selection algorithms have on creating dust. If you are interested, you can watch his video or refer to the blog post that he reviews, which is originally based on the formal wallet specification paper.

He highlights a number of interesting points, two of which stand out: by choosing a coin selection algorithm that keeps transaction amounts and change amounts close to 1:1 you maximize the pseudo-anonymity of the transactions, because it is harder for people to identify who is actually receiving a cryptocurrency payment and who is receiving the change. The second point is that the coin selection algorithm may not matter all that much for individual wallets, but for an exchange it can have serious efficiency consequences.

Storage Consequences

The formal Cardano wallet specification document states that, "Obviously, storing all checkpoints of the UTxO leads to unbounded memory usage. Thankfully, however, the blockchain protocol defines a 'security parameter' k which guarantees that we will never have to roll back past k slots, and hence don't have to store more than k checkpoints. Currently, k is set to 2160." With a maximum block size of 2MB only 4GB of data needs to be stored in addition to around 13.5 gigabytes for pending transactions. The paper also discusses the aforementioned issue of coin selection and says that a poor algorithm will unnecessarily grow the UTxO set and that, "input selection should attempt to keep the size of the UTxO steady."

Cardano also expects to be able to handle a very large UTxO state usingAVL+ Trees for User Issued Assets. How this will be implemented is an entire subject of its own, however, and is slated for the Goguen release.

Conclusion

What is exciting about both the Chimeric Ledger and the Extended UTxO model is that developers at IOHK and EMURGO are not constrained by the past-time limitations of blockchain technology; researchers and developers are willing to create new concepts to solve difficult problems.

Cardano is a high assurance blockchain. It can be argued that institutions, and governments, with billions, and trillions, in currency reserves will not put their funds on a smart contract enabled blockchain without the formal proofs necessary to ensure security. Many banks are still running COBOL and other legacy

technology with unsupported hardware, partly due to incompetent management, but also because the systems are high assurance and have worked for so long. However, at some point in the future, the cost of not transitioning away from aging infrastructure will become too great; aged centralized systems with single points of failure will come to represent significant financial risk.

Bitcoin has proven, over the last ten years, that the UTxO model is both safe and secure. Cardano's extension to that model, through either hybrid transactions, or transaction conversions, along with multichain support and Extended UTxO script, will prove to be both straightforward and trustworthy; as a decentralized and secure, smart contract enabled, high assurance cryptocurrency system, Cardano represents a significant technological and historical accomplishment.

About EMURGO

EMURGO drives the adoption of Cardano and adds value to ADA holders by building, investing in, and advising projects or organizations that adopt Cardano’s decentralized blockchain ecosystem. EMURGO leverages its expertise in blockchain R&D as well as its global network of related blockchain and industry partners to support ventures globally.

EMURGO is the official commercial and venture arm of the Cardano project, registered in Tokyo, Japan since June 2017 and in Singapore since May 2018. EMURGO is uniquely affiliated and works closely with IOHK to grow Cardano’s ecosystem globally and promote the adoption of the Cardano blockchain. To learn more about the project, visit the EMURGO website.

|| **Click here to subscribe to the EMURGO Newsletter** ||

Follow EMURGO on Social Media

- Official Homepage: emurgo.io
- Twitter (English): @emurgo_io
- Twitter (Japanese): @Emurgo_Japan
- Youtube: EMURGO
- Telegram: EMURGO Announcements
- Facebook: @emurgo.io
- Instagram: @emurgo_io
- LinkedIn: @emurgo_io

About Yoroi Wallet

- Yoroi Twitter: @YoroiWallet
- Yoroi Homepage: https://yoroi-wallet.com/

About Cardano

- Cardano Forum: https://forum.cardano.org/
- Cardano Telegram: https://t.me/CardanoGeneral
- Cardano Reddit: