pairfy

Pairfy,
A P2P ecommerce protocol based on trust rating and blind peers

Juan C. Rey*

November 24, 2023

Abstract. Pairfy is an electronic commerce protocol that uses smart contracts to decentralize functional requirements necessary for the processes of selling and buying a physical product. Any member of the community can post a product for sale and any member of the community can express an intention to purchase that product. If the stock of a product is 15, only the 15 fastest people who express their intention to buy will be able to occupy a slot and secure a negotiation session. The slot of a product is a concept that represents the availability to open a negotiation session similar to a sell order on a DEX. A slot can be released if the buyer or seller cancels the session. A negotiation session is defined as the process of trading a product for a limited time. The buyer and seller generate the negotiation context by engaging in bilateral communication.

1 Introduction

Pairfy is an electronic commerce protocol that uses smart contracts to decentralize functional requirements necessary for the processes of selling and buying a physical product. Any member of the community can post a product for sale and any member of the community can express an intention to purchase that product. If the stock of a product is 15, only the 15 fastest

[*] @pairfy website: `www.pairfy.io` - Plutus pioneer first cohort. This personal project was born as a proposal to Y Combinator in 2019 and has been activated for contribution to the open source community.

people who express their intention to buy will be able to occupy a slot and secure a negotiation session. The slot of a product is a concept that represents the availability to open a negotiation session similar to a sell order on a DEX. A slot can be released if the buyer or seller cancels the session. A negotiation session is defined as the process of trading a product for a limited time. The buyer and seller generate the negotiation context by engaging in bilateral communication.

2 Requirements

2.1 Negotiation session

The negotiation session is a 4-stage synchronous process *Waiting*, *Locking*, *Delivery*, *Finish*. A stage cannot start if the previous stage has not finished, the transitions are sequential not parallel.

2.1.1 Waiting

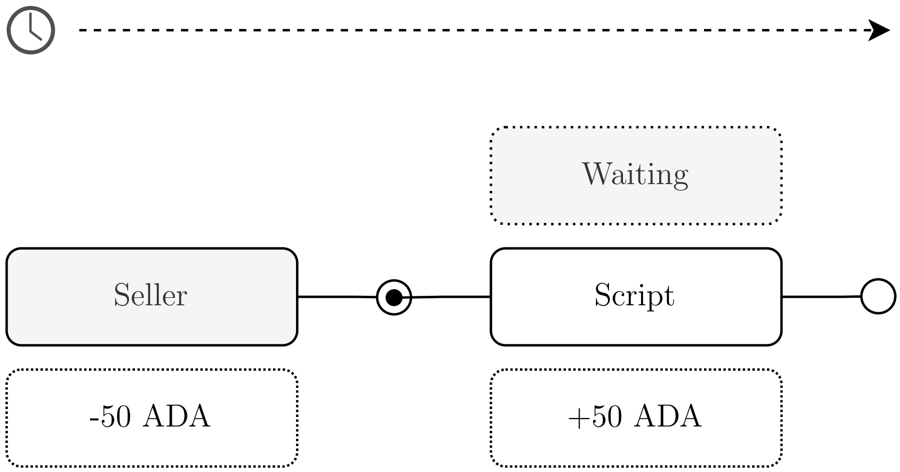

Figure 1: Script deployment

When a seller offers a product to the public by activating a slot (sale order) a small plutus script with state machine logic is activated in the blockchain. The seller must lock an amount greater than 0 ADA as collateral. If the seller acts in bad faith during the negotiation session the seller will lose the collateral. If the seller is a good agent during the negotiation session the collateral will return to him. Collateral guarantees that the seller behaves honestly. This mechanism of coercion allows to generate trust in potential buyers. It also allows the seller to increase their trust rating. Example. Alice publishes a book with 20 units of stock. She activates only 10 slots (sale orders) each with a collateral of 20 ADA. In the first few hours 7 books were sold and now there are only 3 slots left. Alice decides to activate the 10 remaining slots due to demand. Bob sells the same product with the same stock but offers collateral of 5 ADA. The community prefers to buy the books from Alice since she is more committed to fulfilling her obligation by offering 20 ADA collateral. Other factors increase the seller's trust rating such as the total number of successful sales, seniority, or profile information. In the background for each activated slot individual scripts are deployed on the blockchain waiting to be occupied by buyers.

2.1.2 Locking

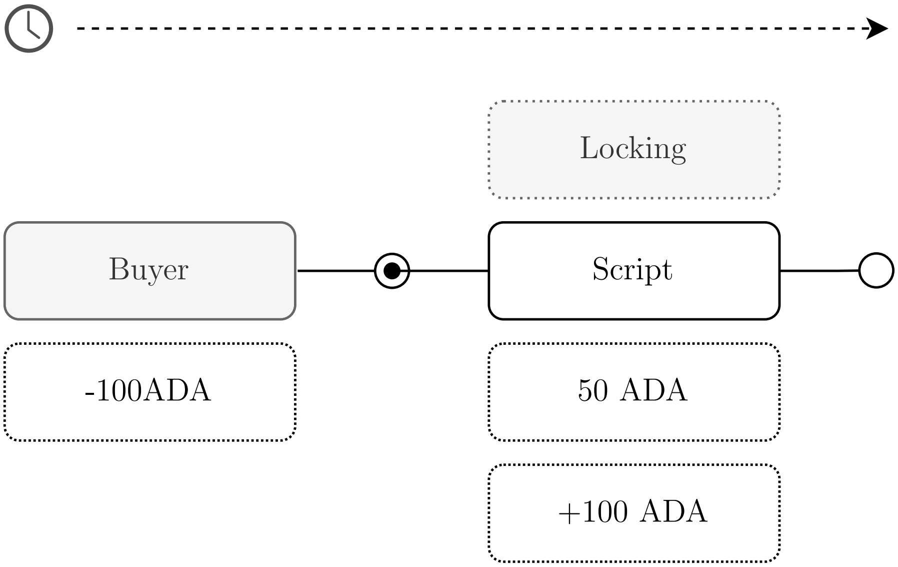

Figure 2: Session locked

In Figure 2 you can see the collateral of 50 ADA given by the seller and the price of the product 100 ADA given by the buyer. When the buyer presses the buy button a slot is occupied and the state of the script transitions from *Waiting* to *Locking*. Blocking funds allows participants to advance their obligations. The seller's obligation is to deliver the correct product with the correct specifications. The buyer's obligation is to receive the product and pay the price. It is important to clarify that the buyer's obligation to pay the price is guaranteed when he occupies a slot since the amount in ADA of the product price is a necessary condition to occupy a slot.

From this point the seller can start with questions such as: What is the delivery point? Description of delivery point? Any questions necessary to guarantee the effective delivery of the product.

All information provided by both actors in the user interface is contained within a websocket instance and can only be observed by the blind peers in case of a dispute. The websocket server does not store any type of information about the negotiation session once legal certainty is declared about the business between the parties. Legal certainty is declared by the blind peers and refers to the absence of reasonable doubt regarding compliance with the obligations that correspond to the buyer and the seller.

2.1.3 Delivery

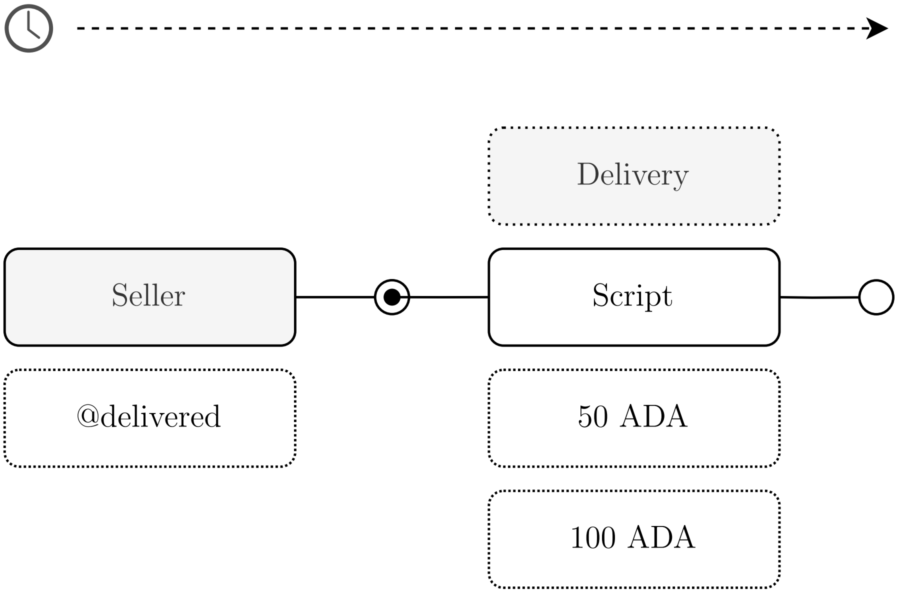

Figure 3: Delivery confirmed by seller

When the shipping company confirms the effective delivery of the product the seller can invoke the @delivered endpoint. The script transitions from *Locking* to the *Delivery* state which indicates that the seller has fulfilled its delivery obligation. Finally, the buyer confirms whether he received the product by invoking the @received endpoint. By doing so, the script transitions to the last state *Finish* which releases the funds to the seller's wallet.

2.1.4 Finish

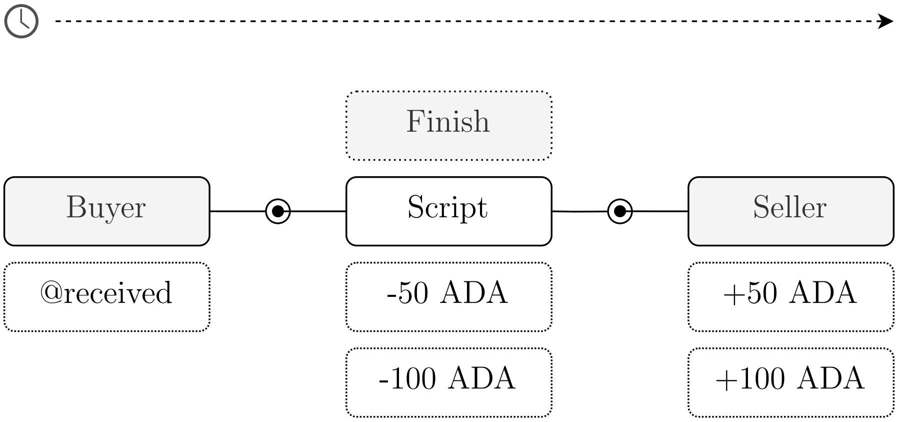

Figure 4: Delivery confirmed by buyer

2.1.5 State machine

A state machine also known as a finite state machine (FSM), is a mathematical model used to describe the behavior of a system or process that can exist in a limited number of distinct states. The system transitions from one state to another in response to specific events or input conditions and these transitions are defined by a set of rules or conditions known as transitions.

A deterministic finite state machine (DFSM) is a specific type of state machine that exhibits deterministic behavior, meaning that for any given state and input there is a unique and unambiguous next state. In a deterministic state machine, the transition from one state to another is uniquely determined by the current state and the input with no ambiguity or randomness involved.

Figure 5 shows the deterministic state machine concepts applied to the steps of a negotiation session. In Cardano's EUTXO model contracts need an initial transaction to activate their design logic and establish the starting state. When the seller activates a slot his wallet deploys a plutus script that receives the collateral in ADA and necessary parameters for the session.

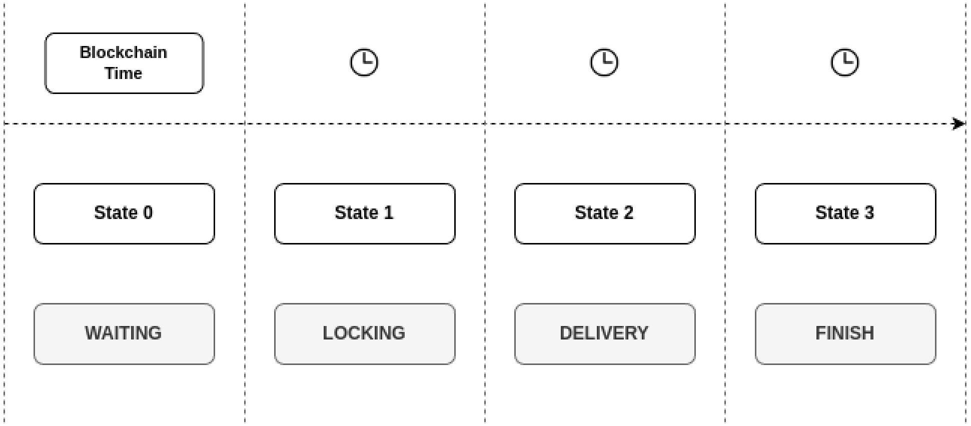

Figure 5: Session states

```
sessionState :: SessionState
sessionState = SessionState {
                              cState = 0
                              sLabel = “waiting”
                              tDuration = 100
                              cSlot = False
                              pDelivered = False
                              pReceived = False
                            }
```

The data type *sessionState* represents the initial state of the plutus script once deployed by the seller. These variables will remain in the default state indefinitely until the buyer’s wallet interacts with the script taking the slot.

```
sessionState :: SessionState
sessionState = SessionState {
                              cState = 1
                              sLabel = “locking”
                              tDuration = 100
                              cSlot = True
                              pDelivered = False
                              pReceived = False
                            }
```

cSlot represents the variable that indicates whether the script has been occupied by a buyer. The boolean value True means that the slot has been occupied by a buyer which makes the script transition from *Waiting* to *Lock-*

ing.

```
sessionState :: SessionState
sessionState = SessionState {
                                cState = 2
                                sLabel = “delivery”
                                tDuration = 100
                                cSlot = True
                                pDelivered = True
                                pReceived = False
                              }
```

pDelivered is a variable of type boolean that represents the delivery of the product or not. At this point the seller has invoked the @delivered endpoint stating that he has fulfilled his obligation to deliver the product. This action makes the script transition from *Locking* to *Delivery*

```
sessionState :: SessionState
sessionState = SessionState {
                                cState = 3
                                sLabel = “finish”
                                tDuration = 100
                                cSlot = True
                                pDelivered = True
                                pReceived = True
                              }
```

pReceived is a boolean variable that represents whether the buyer confirms receipt of the product or not. This variable is modified only by the buyer using the @received endpoint. By doing this, the script transitions to its final state *Finish* which releases the funds to the seller.

2.2 Blind peers

Connecting the blockchain to the real world is an expensive challenge at least for trading physical products. Shipping companies would have to connect their vehicles to supervised oracles and the products must have an infallible tracker to check if were delivered. A lot of infrastructure deployed around the world is necessary to achieve a fully decentralized ecommerce protocol. Clearly this represents a very difficult challenge to achieve in contrast to another option which is to use the human brain as a source of truth. With the right conditions the human brain can make logical reasoning about facts or propositions to declare a truth. For example, the peer review system in the academic world allows a research paper to be reviewed by randomly assigned experts. A blind reviewer does not know who is the author of the paper he is reviewing and the author also does not know who the reviewers are. The system maintains confidentiality to prevent biases, ensuring a rigorous and impartial assessment. Serving as a crucial quality control mechanism, it identifies errors, provides constructive feedback to authors and contributes to the overall accuracy and reliability of published work. A system of blind peers trained in conflict resolution can decide on a dispute in a negotiation session in the event that the buyer or seller fails to comply with their obligations or another problem arises that does not allow the natural conclusion of the session. A blind peer system can decide on a disputed trading session.

Some characteristics of the profile of a mediator are:

1. Mediators must stay neutral and impartial to ensure a fair and unbiased assessment of disputes, fostering trust in the mediation process for all parties involved.

2. Effective communication is vital for mediators, involving active listening, skillful facilitation of discussions and clear conveyance of information to aid parties in mutual understanding and resolution.

3. A mediator needs strong problem-solving skills to navigate complex disputes and find mutually acceptable solutions.

4. Mediators require patience as mediation processes take time, allowing parties to express themselves, explore solutions and reach agreements at their own pace.

5. Mediators must uphold ethical standards ensuring the confidentiality of the mediation process to establish and maintain the crucial element of trust for successful resolution.

6. Mediators require a comprehensive skill set encompassing technical, legal, and cultural knowledge to effectively carry out their duties.

2.2.1 Appeal

Peer-to-Peer cryptocurrency exchange services such as localbitcoins or binance P2P have proven to be systems that work for secure trades. These systems are based on trust rating and involve real-world action on the part of the buyer such as making a transaction to the bank account provided by the seller. In Binance P2P, disputes between buyers and sellers can be addressed through an appeal process. Common reasons for initiating an appeal involves problems with payment confirmation, disagreements over payment quality, or disputes regarding trade terms. To protect the cryptocurrency involved in the trade an escrow system locks the funds.

Both the buyer and seller are afforded the opportunity to present evidence or explanations supporting their case during the appeal. A mediation team at Binance reviews the appeal carefully considering the evidence and arguments put forth by both parties. Subsequently, Binance reaches a decision which may involve upholding the original trade agreement, releasing funds from escrow to the appropriate party, or taking other actions based on the specific circumstances of the dispute.

In the case of Pairfy the seller is the party that has control over the terms of the negotiation. The UI view of a published product contains the product description, seller name, seniority, trust rating, number of successful sales, number of total sales, and the terms that any buyer must meet to be able to negotiate with the seller. These terms must be clear, enumerated and determined. Any buyer who accepts the seller's terms must comply with them.

It is not convenient for the seller to use the @delivered endpoint if he has not fulfilled his obligation since he has to prove it to a level of certainty at the risk of losing his collateral. The seller also cannot use the endpoint by mistake since the user interface would have mechanisms to prevent this. It is also not convenient for the buyer to use the @received endpoint without having received the product since he would lose the funds corresponding to the price. He also cannot do it by mistake since the UI does not allow it.

If the buyer does not comply with his obligation to receive the product, the seller can initiate the appeal. If the intention of abuse is proven, then the buyer loses the equivalent of the collateral proposed by the seller. Both parties can resort to appeal in case of dispute, a group of blind mediators can take action on a case-by-case basis.

2.2.2 Pool

According to the principle of least cardinality in data modeling which states that one-to-many relationships such as between the author and the N of published posts, the identifier that relates them should be stored on the "many" side. In other words, when data modeling is designed posts should store the author identifier, authors should not store identifiers about their posts. As long as the author has a small number of published posts there are no problems with storing and consulting the information. The problem arises when the author has a large number of published posts, the storage of post identifiers in one of the author variables represents a scaling problem and the query time can be affected. The application of this principle in the protocol architecture allows high horizontal scalability. Pairfy uses a master-slave pattern in which the master is the main contract of the protocol which manages the pool of mediators. Slaves are the individual scripts that are in charge of the trading sessions. For each activated slot of a product a script is deployed waiting to be occupied by a buyer.

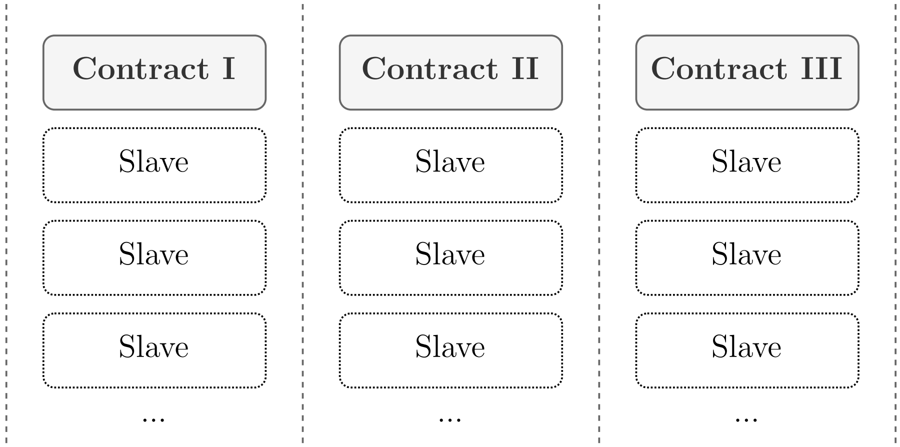

Figure 6: Pairfy instances

Figure 6 shows a grouping of consecutive contracts representing horizontal scaling to ensure high concurrency and availability of the Pairfy service. When a seller enables a product slot a slave script with state machine logic is deployed in the blockchain waiting for a buyer to occupy it to start a negotiation session. Each slave script has by default the *PubKeyHash* of the master contract in its datum. A slave cannot interact with another contract other than its master.

Master contracts do not have any information about their slaves, their function is to provide mediation service in case of a dispute during a negotiation session by enabling an endpoint called @appeal. The endpoint is designed to be used by slave scripts and requires an ADA fee to prevent abuse.

The mediator pool is basically a set of *PubKeyHash* each corresponding to the wallet of a certified and active mediator. When @appeal is triggered by a slave the master contract mint an NFT and within the NFT datum the contract adds three (3) *PubKeyHash* contained in the pool of mediators. Subsequently the contract removes those three *PubKeyHash* from the pool queue.

The slave receives the NFT and adds to its own datum the three *PubKeyHash* that are within the NFT datum. After this process the slave has exact information about the mediators authorized to decide on the current dispute. Mediators are the only ones who can activate special @endpoints of the slave to decide the dispute.

2.2.3 Designation

The community using governance can change the policies and parameters of the protocol. The community can also decide who meets the characteristics to be a mediator or not. Mediators can have two states: active and inactive. Only active mediators can enter the pool.

2.2.4 Rewards

To avoid bias and inactivity, mediators are rewarded every 30 days with utility token. The amount will be decided by community governance. Only active days in the pool can be rewarded.