# Establishing a Blockchain-enabled, Highly Liquid, Auction-based Employee Rewards Marketplace

## DRAFT VERSION 0.7

Brendan Ward[1], Albert Eloyan[2], Alex Norta[3]

[1] Universal Recognition Token, 7 World Trade Center, New York 10006, USA
brendan@urtoken.net
[2] University of California, Berkeley, USA
ae@berkeley.edu
[3] Large-Scale Systems Group, Tallinn University of Technology,
Akadeemia tee 15A, 12616 Tallinn, Estonia
alex.norta.phd@ieee.org

**Abstract.** American corporations spend over $90 billion each year on employee gifts to be awarded via employee recognition programs that acknowledge outstanding employee achievement. Typically, these gifts come in the form of gift cards or merchandise. Although some employees receive gifts they want and appreciate, a not insignificant amount of the time these gifts are seen as unwanted or undesirable by employees. Consequently, many gifts are discarded altogether by employees. We propose to monetize the market of unwanted corporate gifts by matching those gifts with individuals wishing to purchase them using blockchain-technology such as smart contracts. Employees will be able to liquidate their unwanted gifts for cash or marketplace points (at the employer's option) and do in an immutable and traceable way; value realized by gifts in this manner can be spent on alternate gifts or saved toward a future gift of greater value (employee's option). Accurate market-demand information regarding offered prizes will become available via this system; relative demand as well as customer-driven price data will be available to employers to better select prizes and prize vendors for future gifts.

**Key words:** employee recognition program, corporate gifts, monetize, blockchain, smart contract, market-demand information, high liquidity, matching gift preferences, customer-driven

## IMPORTANT NOTICES

YOU MUST READ THE FOLLOWING BEFORE CONTINUING. YOU ARE ADVISED TO READ THIS CAREFULLY BEFORE READING THIS DRAFT WHITEPAPER, ACCESSING OR MAKING ANY OTHER USE OF THIS DRAFT WHITEPAPER.

This Draft Confidential Whitepaper (this ”Draft Whitepaper”) has been prepared by Universal Recognition, Inc. for use by prospective purchasers to whom the Company expects to offer the opportunity to purchase UR Tokens ( the ”UR Tokens”). Unless the context requires otherwise, in this Whitepaper the terms ”Company,” ”Token Issuer”, ”we,” ”us” and ”our” refer to Universal Recognition, Inc. each of their subsidiaries and affiliates, as applicable, and all dollar ($) amounts set forth herein refer to United States dollars.

This Whitepaper supersedes in its entirety any and all previous draft whitepapers, marketing memoranda, term sheets, verbal or written communications or other information regarding the proposed offering described herein.

THIS DRAFT WHITEPAPER SHALL NOT CONSTITUTE AN OFFER TO SELL OR THE SOLICITATION OF ANY OFFER TO BUY UR TOKENS NOR SHALL THERE BE ANY SALE OF SUCH TOKENS IN ANY JURISDICTION IN WHICH SUCH OFFER, SOLICITATION OR SALE WOULD BE UNLAWFUL. ANY OFFER TO PURCHASE THE UR TOKENS TO BE OFFERED BY THE COMPANY WILL BE MADE SOLELY PURSUANT TO A PRIVATE PLACEMENT MEMORANDUM TO BE MADE AVAILABLE BY THE COMPANY.

INFORMATION CONTAINED IN THIS DRAFT WHITEPAPER IS SUBJECT TO COMPLETION OR AMENDMENT. THE UR TOKENS DESCRIBED IN THIS DRAFT CONFIDENTIAL WHITEPAPER MAY NOT BE SOLD PRIOR TO DELIVERY OF A FINAL WHITEPAPER AND FINAL PRIVATE PLACEMENT MEMORANDUM.

ANY FUTURE OFFER AND SALE OF UR TOKENS WILL NOT BE REGISTERED WITH THE SECURITIES AND EXCHANGE COMMISSION (THE ”SEC”) UNDER THE SECURITIES ACT OF 1933, AS AMENDED (THE ”SECURITIES ACT”), OR ANY STATE SECURITIES ACT. IF AND WHEN OFFERED PURSUANT TO A PRIVATE PLACEMENT MEMORANDUM, THE UR TOKENS WILL BE OFFERED AND SOLD IN RELIANCE ON EXEMPTIONS FROM THE REGISTRATION REQUIREMENTS OF SUCH ACTS.

ANY FUTURE OFFERING WILL BE MADE IN THE U.S. IN RELIANCE UPON THE AVAILABILITY OF AN EXEMPTION FROM THE REGISTRATION PROVISIONS OF THE SECURITIES ACT BY VIRTUE OF THE INTENDED COMPLIANCE WITH THE PROVISIONS OF RULE 506(C) OF REGULATION D. ACCORDINGLY, ANY PROSPECTIVE PURCHASER OF UR TOKENS MUST BE VERIFIED AS AN ”ACCREDITED INVESTOR”, AS DEFINED IN RULE 501 OF REGULATION D. GENERALLY, THIS WILL REQUIRE ALL PROSPECTIVE PURCHASERS TO PROVIDE CERTAIN

DOCUMENTATION TO THE COMPANY OR ITS VERIFICATION AGENT THAT SUCH PURCHASER HAS A NET WORTH OF AT LEAST $1 MILLION (WITH THE CALCULATION OF NET WORTH EQUALING THE EXCESS OF TOTAL ASSETS OVER TOTAL LIABILITIES, DISREGARDING FROM THIS CALCULATION THE VALUE OF THE PURCHASER'S PRIMARY RESIDENCE AND ANY ASSOCIATED DEBT SECURED BY THE PURCHASER'S PRIMARY RESIDENCE; BUT ONLY TO THE EXTENT ANY SUCH DEBT DOES NOT EXCEED THE FAIR MARKET VALUE OF SUCH RESIDENCE) OR THAT SUCH PURCHASERâĂŹS GROSS INCOME HAS EQUALED OR EXCEEDED $200,000 DURING EACH OF THE LAST TWO YEARS AND FOR THE CURRENT YEAR. PROSPECTIVE ACCREDITED INVESTORS WILL BE REQUIRED TO REPRESENT AND/OR DEMONSTRATE TO THE SATISFACTION OF THE COMPANY OR ITS VERIFICATION AGENT THAT: (i) THEY HAVE SUCH SOPHISTICATION, KNOWLEDGE AND EXPERIENCE IN FINANCIAL AND BUSINESS MATTERS THAT THEY ARE CAPABLE OF EVALUATING THE MERITS AND RISKS OF THIS PURCHASE AND (ii) THEY ARE ABLE TO BEAR THE ECONOMIC RISKS OF THIS PURCHASE, INCLUDING A TOTAL LOSS OF THEIR PURCHASE PRICE.

ANY INVESTMENT IN UR TOKENS PURSUANT TO A FUTURE OFFERING WILL NOT HAVE BEEN APPROVED OR DISAPPROVED BY THE SEC OR ANY OTHER REGULATORY AUTHORITY NOR HAS ANY AUTHORITY PASSED UPON OR ENDORSED THE MERITS OF ANY PROPOSED OFFERING OR THE ACCURACY OR ADEQUACY OF THIS DRAFT WHITEPAPER. ANY REPRESENTATION TO THE CONTRARY IS A CRIMINAL OFFENSE. THIS DRAFT WHITEPAPER DOES NOT CONSTITUTE AN OFFER OR SOLICITATION TO ANYONE IN ANY STATE OR IN ANY OTHER JURISDICTION.

NO PERSON HAS BEEN AUTHORIZED IN CONNECTION WITH ANY FUTURE OFFERING OF UR TOKENS TO GIVE ANY INFORMATION OR MAKE ANY REPRESENTATIONS OTHER THAN THOSE CONTAINED IN A FINAL WHITEPAPER OR FINAL PRIVATE PLACEMENT MEMORANDA AND, IF GIVEN OR MADE, SUCH INFORMATION AND REPRESENTATIONS MUST NOT BE RELIED UPON. UNDER NO CIRCUMSTANCES SHALL THE DELIVERY OF THIS DRAFT WHITEPAPER OR FINAL SALE MADE PURSUANT TO A FINAL PRIVATE PLACEMENT MEMORANDUM CREATE ANY IMPLICATION THAT THERE HAS BEEN NO CHANGE IN THE FACTS OR THE AFFAIRS OF THE COMPANY DESCRIBED IN THIS DRAFT WHITEPAPER SINCE THE DATE HEREOF, OR THAT THE INFORMATION CONTAINED HEREIN IS CORRECT AS OF ANY TIME SUBSEQUENT TO THE DATE OF THIS DRAFT WHITEPAPER.

THIS DRAFT WHITEPAPER MAY NOT BE FORWARDED OR DISTRIBUTED TO ANY OTHER PERSON AND MAY NOT BE REPRODUCED IN ANY MANNER WHATSOEVER. ANY FORWARDING, DISTRIBUTION

OR REPRODUCTION OF THIS WHITEPAPER IN WHOLE OR IN PART IS UNAUTHORIZED. FAILURE TO COMPLY WITH THIS DIRECTIVE MAY RESULT IN A VIOLATION OF CERTAIN APPLICABLE LAWS.

YOU ARE STRONGLY ADVISED TO CONSULT YOUR INDEPENDENT LEGAL AND TAX ADVISORS IN RESPECT OF THE LEGALITY IN YOUR JURISDICTION OF YOUR PARTICIPATION IN ANY FUTURE OFFERING OF UR TOKENS AND ANY TAX IMPLICATIONS ASSOCIATED THEREWITH.

## Table of Contents

1 Legal Disclosure....................................................................6

2 Problem and Solution...........................................................6

2.1 Problem with employee rewards...........................................7

2.2 Solution.......................................................................................8

3 Product and Architecture......................................................9

3.1 Goal model.................................................................................9

3.2 System architecture...............................................................11

3.3 Dynamic protocols..................................................................12

3.4 Employee (Auctioneer) interface application......................15

3.5 URT holder (Bidder) interface application.........................18

4 Token Economics...................................................................20

4.1 Token model.............................................................................21

4.2 Transaction scenario..............................................................22

4.3 System-throughput throttling...............................................24

4.4 Assumed financial risks.........................................................25

5 Technical Challenges..............................................................26

6 Technical Solutions.................................................................27

6.1 Mass-data management.........................................................28

6.2 Smart-contract management.................................................28

6.3 Verifying the soundness of smart contracts........................30

6.4 Identity authentication............................................................30

7 Conclusion and Future Work.................................................31

For more information about the token structure, the team, roadmap, please go to our website [1].

[1] https://urtoken.net/

## 1 Legal Disclosure

Nothing herein constitutes an offer to sell, or the solicitation of an offer to buy, any tokens, nor shall there be any offer, solicitation or sale of Universal Recognition Tokens (URT) in any jurisdiction in which such offer, solicitation or sale would be unlawful. You should carefully read and fully understand this whitepaper and any updates and the ”Important Notices” on the facing pages of this Whitepaper. Every potential token purchaser will be required to undergo an on-boarding process that includes identity verification and certain other documentation, which you should read carefully and understand fully because you will be legally bound. Please make sure to consult with appropriate advisors and others.

This white paper describes our current vision for the URT Network platform. While we intend to attempt to realize this vision, please recognize that it is dependent on quite a number of factors and subject to quite a number of risks. It is entirely possible that the URT Network platform will never be implemented or adopted, or that only a portion of our vision will be realized. We do not guarantee,represent or warrant any of the statements in this white paper, because they are based on our current beliefs, expectations and assumptions, about which there can be no assurance due to various anticipated and unanticipated events that may occur.

Please know that we plan to work hard in seeking to achieve the vision laid out in this white paper, but that you cannot rely on any of it coming true. Blockchain, crypto-currencies and other aspects of our technology and these markets are in their infancy and will be subject to many challenges, competition and a changing environment. We will try to update our community as things grow and change, but undertake no obligation to do so.

## 2 Problem and Solution

Employee recognition programs are a longstanding feature of corporate workplaces [17]. Most of us have seen these programs in action in some way, shape or form, and the purposes of these programs are apparent: to reward desired workplace behavior, to publicly acknowledge those behaviors and to encourage those behaviors in more employees. Such occasions include:

- Meeting and exceeding sales goals
- Employment Anniversary (1 year, 5 year,..)
- Project Milestones
- Employee Birthdays
- Holidays (Christmas / New Year)
- Seasonal Events (Spring)
- Promotions
- Weddings / New Babies
- Health / Wellness

– Meeting Project Milestones
– Rollout of New Product/Service
– Opening New Office/Factory

### 2.1 Problems with employee rewards

Market research performed by Bersin and Deloitte showed that in 2012, companies in the United States spent between 1-2% of their total payroll outlay on those programs, and additionally in this year, $46 billion was spent on those programs. Subsequent studies place 2013 spending at $77 billion and 2015 spending at $90 billion[2]. Most of this spending goes toward items such as gift cards and merchandise. A 2015 study by the Incentive Federation found that in businesses using non-cash award types, gift cards and merchandise were the two most broadly used types of rewards, with almost 90% of businesses using gift cards and about three-quarters using merchandise in their various employee reward programs.

Corporate spending on these categories is significant; however, that spend may not be effectively directed, resulting in waste. Let us look at both categories: gift card spend and merchandise.

According to a marketing survey of Fortune 500 company employees, just over one third admitted that they had a company-issued gift card that they simply hadn't used, either wholly or in part[3]. Furthermore, according to the Tower Group, cards not redeemed within 60 days aren't ever likely to be redeemed at all resulting in billions of dollars of purchased gift card value going unspent each year.

As regards merchandise spending: is that spending going towards items that employees really want? One major US retailer has offered the following employee gifts in the last five years:

– Wristwatch
– Howard Miller Rosewood Clock
– iCOOL Xtreme Waterproof Cooler Backpack
– Aluminum Folding Picnic Table with 4 Seats

It seems apparent that to some extent, both gift cards and corporate merchandise are sitting unused on shelves, in boxes or in dusty garages. In short, companies are spending 1-2% of payroll on programs that offer minimal or, in some cases, nonexistent return on investment. The system is extraordinarily wasteful and inefficient especially given how much money is being spent. However, due to that waste, there is also opportunity present.

[2] http://www.incentivemag.com/News/Industry/Incentive-Federation-$90-Billion-Market-Size/

[3] https://www.cbsnews.com/news/why-retailers-love-gift-cards-but-you-shouldnt/

### 2.2 Solutions

Our solution leverages blockchain technology [22] to magnify employee satisfaction while simultaneously minimizing employer cost and waste. The primary feature of our system is a virtual marketplace [23] that allows employees to auction gifts while preserving the value of those gifts for later use and/or accumulation toward higher-value gifts.

When an employee receives a gift or wins a prize, he can take it right then for himself or choose to auction it and receive cash (or alternatively, the employer can choose to credit employee with internal reward *points* or *miles* that have the equivalent value of cash and are redeemable for prizes off of a fixed-price prize menu – e.g., the ”United Airlines Mileage Plus mall” framework). This system allows employees to exchange an otherwise undesirable gift for value to be spent on a desirable gift. Additionally, this marketplace liquidity adds perceived value to employee rewards by allowing the value of multiple smaller gifts to be combined and applied toward a future gift of greater value. Our system is intended to mitigate the chance of HR or management forcing upon an employee an inappropriate gift (e.g. a voucher to a high-end steakhouse being presented to a vegan employee) when an employee could instead make his or her own choice.

An additional benefit to a highly liquid marketplace is that the need to only give generic gifts is eliminated; management no longer needs to hand out gifts from an endless stack of Starbucks or Barnes and Noble gift cards. Because a notable feature of this system is its geographic and economic flexibility, vendors across the world will be able to enter the system, and the value of their gifts will be determined by marketplace demand. When an employee auctions off an unwanted gift voucher from, say, a noted Boston seafood restaurant, the URT Holders will determine the value of that voucher by means of their bids. Over time, employers will receive data about which prizes their employees are keeping versus auctioning; due to the fact that blockchain is a distributed ledger, they will also be able to see general demand for items that are auctioned, meaning that even if the employer misses the mark on giving an employee a suitable prize, at least they picked one that has maximal ”exchange” or ”trade-in” value for the employee. Access to this type of data will help employers and HR departments determine the viability of prizes (and prize vendors) and allow a much more accurate selection of gifts to be utilized as corporate rewards in the first place.

Millennials are joining the workforce in large numbers, so the time is right to sharpen and fine-tune recognition programs. Remember that today’s employees rarely ever spend years or decades with a single employer, as their parents and grandparents often did. For better or worse, that level of employee loyalty simply no longer exists. A corollary of that is that a less loyal workforce generally leads to higher turnover. Despite that, the Bersin study found that employers with ”recognition-rich culture” and improved recognition programs (defined as companies in the top quintile of that measure) enjoyed average voluntary turnover rates 31% lower than other companies. From a business perspective, this is a notable and desirable statistic, and clearly a counterweight to decreased employee loyalty.

So why not give the power of choice to the persons being rewarded? Rather than spending time and effort curating a list of gifts from a supplier's list, or presenting employees with branded gifts that may end up in a box or on a shelf, why not reward employees with gifts that match with their beforehand submitted sets of preferences and needs? An employee can select what they want with far greater accuracy than an HR staff member can, and a self-selected gift or prize will surely be meaningful and memorable to an employee.

## 3 Product and Architecture

We present the URT-system in orthogonal aspects starting with Section 3.1 that comprises the requirement sets and involved stakeholders. Note that the requirements fall into the categories of functional goals and non-functional quality goals. The functional goals are arranged as a hierarchically refined tree. Associated to the functional tree are quality goals and stakeholders to denote which functional goals they concern. Section 3.2 shows the static architecture of the URT-system that is derived from the goal model. The additional aspect of the architecture model is inserted information-exchange elements that the goal model does not address. Finally, Section 3.3 addresses the dynamic runtime behavior of the URT-system and the embedded utilization of blockchains.

### 3.1 Goal model

An agent-oriented model type termed goal model [25] is depicted in Figure 1. In more detail, AOM goal models comprise a simple notation as is depicted in Figure 1. Functional goals of a system are depicted as parallelograms, 'quality goals' or non-functional requirements are clouds, and agents with specified roles that may be human or artificial are stick figures. The root element of a goal model is a 'value proposition' that is depicted as a functional goal. The value proposition denotes the overall systems goal based on which the rest of the model is hierarchically decomposed. Attached to functional goals are quality goals and roles that also hold for lower-level, refining functional goals.

The central functional goal of model in Figure 1 is labeled as *Establish a liquid preference-matching recognition program.* That way, we adhere to the solution put forward in Section 2.2 where the solution of the URT-system is mapped out. With respect to quality goals, Figure 1 shows most are associated to the value proposition root functional goal. Thus, these are quality goals that also affect all hierarchically refining functional goals below the main value proposition. We now turn to describing each quality goal at length.

*Cost minimizing*, on the other hand, indicates that the marketplace must also serve those who bid on auction items wanting to pay lower-than-par/retail prices for items they find desirable. *Waste avoiding* infers that a better matching

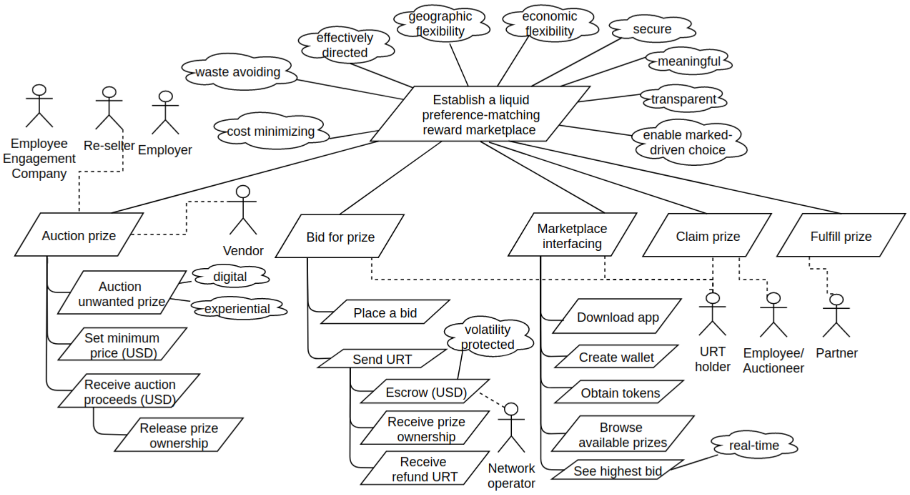


**Fig. 1.** The goal model for the URT system.

of employee preferences with their received gifts is achieved. In a secondary step, this is achieved by auctioning. *Effectively directed* describes a system that is efficient in identifying and matching people with varying preferences in a way that is economically beneficial to both parties involved. This is an inherent virtue of an auction-style bidding system. *Geographically flexible* means the items/prizes that are up for auction are not geographically limited when it comes to redemption. Auctioneers and bidders from all over the world can exchange on a marketplace that has a diverse inventory and that is agnostic to location. *Economic flexibility* refers to the URT-system offering a market place where free market forces can unfold that caters to the needs of a diverse set of stakeholders. *Secure* indicates that the Universal Recognition Marketplace must be protected against unauthorized usage attempts such as denial of service attacks and scammers, while providing services to trusted partners and token holders. *Meaningful* assumes that the gifts match with the declared preferences of an employee and auctioneer. At the same time, the employer expects the gift to trigger the expected positive reinforced for enhancing employee loyalty and motivation. *Transparent* refers to ensuring that the marketplace partners and URT holders have a clear understanding of auction mechanics, data flow and lot settlements and are aware of each other's previous reputation and activities on the marketplace. Finally, the quality goal *enable market-driven choice* refers to an ideal support of the URT-system for guiding an employee and auctioneer to the right gifts that are perceived as most rewarding.

*Digital* means that experiential items put up for auction can be digitally represented (e.g. gift cards, vouchers, etc.) to ensure accurate record-keeping. *Experiential* means prizes that are NOT material goods but instead revolve around experiences (e.g. dining, spa, museum, movies, travel, etc). *Volatility protecting* pertains to protecting both auctioneers and bidders against currency volatility for the duration of the auction. Lastly, *real-time* describes that bidders must receive real-time information about an ongoing auction in order to effectively participate.

### 3.2 System architecture

Based on the goal model above, we deduce a UML component diagram [3] to specify the static structure of the URT-system. Figure 2 depicts a UML notation with components as labeled rectangles. The latter we further refine with hierarchically nested sub-components and attached provided- and required interfaces depicted as a line with a circle and required interface as a lines with a cup. Actors (roles in goal models) indicate their interaction with components and the main addition in component diagrams versus goal models are the data exchanges that provided- and required interfaces denote.

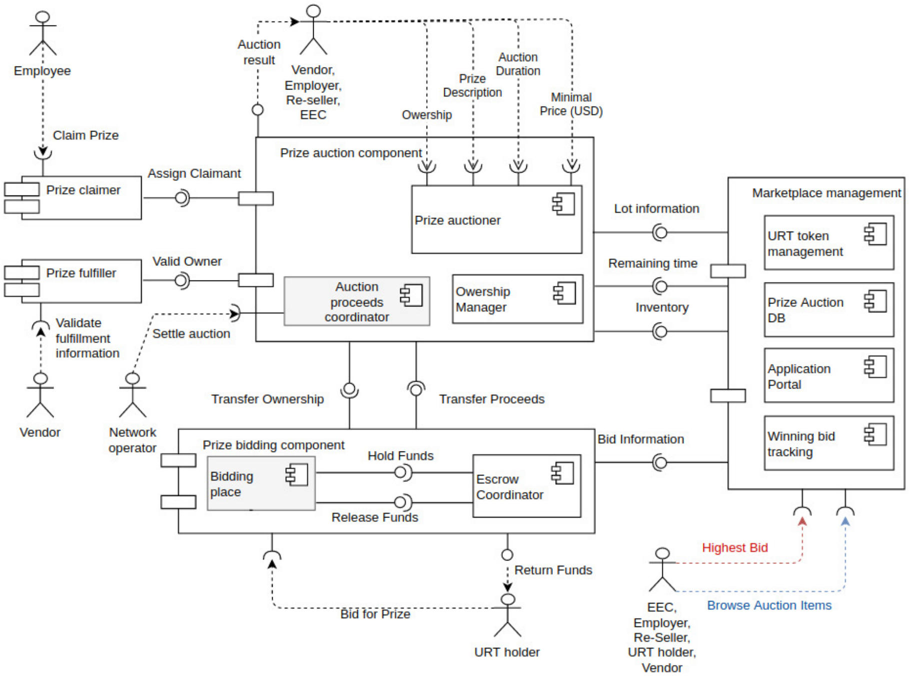


**Fig. 2.** The component-model architecture of the URT system.

The prize auction component in the top center of Figure 2 has an embedded component for ownership management. We omit refinement beyond first refinement levels in the architecture due to space limitation. Note the dashed gray box denotes URT use correspondingly with the goal models in Section 3.1. These components have several provided and required interfaces.

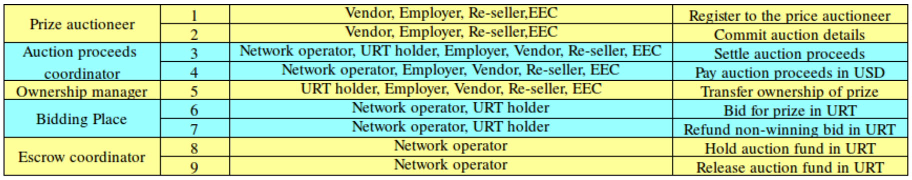
The *Prize Auction Component* has multiple required interfaces that are provided by the Partner Actor (Employer, Employee Rewards Company, Vendor, Reseller): the auction lot description (what is the nature of the prize up for auction, e.g: dinner for two at a restaurant in Paris), ownership information (who is the current owner of the item/prize), auction duration (how long will the auction be open for bidding) and the minimum price (the minimum amount the current owner wishes to receive for the auction to complete). The *Prize Auction Component* in turn has providing interfaces that relay some of that information (auction details, the remaining time and inventory) to the *Marketplace Management Component* to be provided to the EEC, Employer, Re-seller, URT holder and Vendor Actors via the URN Mobile app. The *Prize Auction Component* also asserts Auction Validity via a providing interface of the same name to the *Prize Bidding Component.*

Once the *Prize bidding component* receives auction validity through its corresponding required interface, it can start accepting bids. This happens through a receiving interface and is provided by URT holder actors. There are multiple internal interfaces within the *Prize bidding component* that debit the bid amount in URT from the URT holder, and store the funds into escrow via the *Escrow Coordinator* embedded component for the duration of the auction. Once the *Prize auction component* receives a signal from the Network Operator actor via its required interface indicating that the auction is complete and settled it will initiate an ownership transfer via its provided interface to the corresponding required interface of the *Prize bidding component.* The latter will then record the change of ownership of the auction loot item and request a release of funds from escrow via the embedded *Escrow Coordinator* and transfer proceeds via a provided interface to the *Prize auction component.* Finally, the *Prize auction component* has two provided interfaces for URT holder actors: one is to transfer the auction lot to the highest bidder and the other is to return funds to those whose bids were ultimately not sufficient to win the auction.

Lastly, there is a *Prize claimer* feedback component which has a required interface to receive reports and feedback from URT holder actors regarding their claimed items (for example, the URT holder went to a Spa Day experience that he was the winning bidder for). It has a corresponding provided interface to the *Prize auction component* to help with the assertion of auction validity and guard against Partner actors with sufficiently negative reviews from being able to initiate further auctions.

### 3.3 Dynamic protocols

For showing the dynamic behavior of the URT-system, we consider UML sequence diagrams [24]. Figure 3 and Figure 4 depict entities with labels and dashed vertical timelines attached. Directed horizontal arrows denote messages between the entities while bars are termed activations for showing active processing threads. Dashed directed arcs between entities are return messages for returning processing control. Additionally, the activations of an entity may also act asynchronously.



| | | | |
|---|---|---|---|
| Prize auctioneer | 1 | Vendor, Employer, Re-seller,EEC | Register to the price auctioneer |
| | 2 | Vendor, Employer, Re-seller,EEC | Commit auction details |
| Auction proceeds coordinator | 3 | Network operator, URT holder, Employer, Vendor, Re-seller, EEC | Settle auction proceeds |
| | 4 | Network operator, Employer, Vendor, Re-seller, EEC | Pay auction proceeds in USD |
| Ownership manager | 5 | URT holder, Employer, Vendor, Re-seller, EEC | Transfer ownership of prize |
| Bidding Place | 6 | Network operator, URT holder | Bid for prize in URT |
| | 7 | Network operator, URT holder | Refund non-winning bid in URT |
| Escrow coordinator | 8 | Network operator | Hold auction fund in URT |
| | 9 | Network operator | Release auction fund in URT |

**Table 1.** Blockchain transactions for the URT system.

Storing events on a blockchain is costly when it comes to computing power and transaction fees. Since this can become a performance and scalability bottleneck, it is prudent to define the minimal set of transactions that allow for the operation of the URT marketplace.

Table 1 lists the operations with the event, column giving identification numbers of respective blockchain operations. Additionally, we also list the stakeholder set per operation and a corresponding explanation per operation. The first column categorizes the operations into subsets that are assigned to respective components from Figure 6. Note the event IDs we use below in the sequence diagrams are to show at which moments in the protocols the blockchain is used.

We now provide a more thorough description of each operation: operation 1 registers the Partner (Vendor, Employee Rewards Company, Employer, Reseller) with the price auctioneer; operation 2 commits the auction details (item description, auction duration and minimum price to be fetched); operation 3 settles the auction by identifying the highest bidder at the time of close; operation 4 pays the auction proceeds to the Partner that put up the lot for auction; operation 5 records the transfer of ownership from the auctioneer to the highest bidder identified in operation 3; operation 6 submits a bid for an auction lot in URT; operation 7 refunds URT to the bidder who was not the winner/highest bidder at the time of settlement; operation 8 commits all confirmed bids into an auction escrow fund; operation 9 releases the winning bid from escrow to the auctioneer and releases non-winning bids to bidders.

The sequence diagram in Figures 3 and 4 depict as UML entities the components and actors depicted in Section 3.2. Note the use of pseudo-code commands for the messages between the entities that suffice to demonstrate the dynamic protocol behavior. The numbered red dots correspond to the blockchain-operation events in Table 1.

The *Partner* in the top left of Figure 3 commences with a register(profile) message to the *Prize Auctioner* module, which is an event that is immutably stored in a blockchain. The *Prize Auctioner* module returns an acknowledgment indicating that the *Partner* is now eligible to start an Auction. This will be followed by a *push(Prize)* message, which, upon acknowledgment by the *Prize Auctioner module*, commences the auction of the prize.

Bidding starts at the bottom right of Figure 3 when the *URT holder* pushes a *request(Bid)* message to the bidding module and each bid is recorded on chain. The *Bidding Place* component acknowledges the incoming bid and proceed to check the requester's URT balance. If the balance is sufficient, the *Bidding Place*

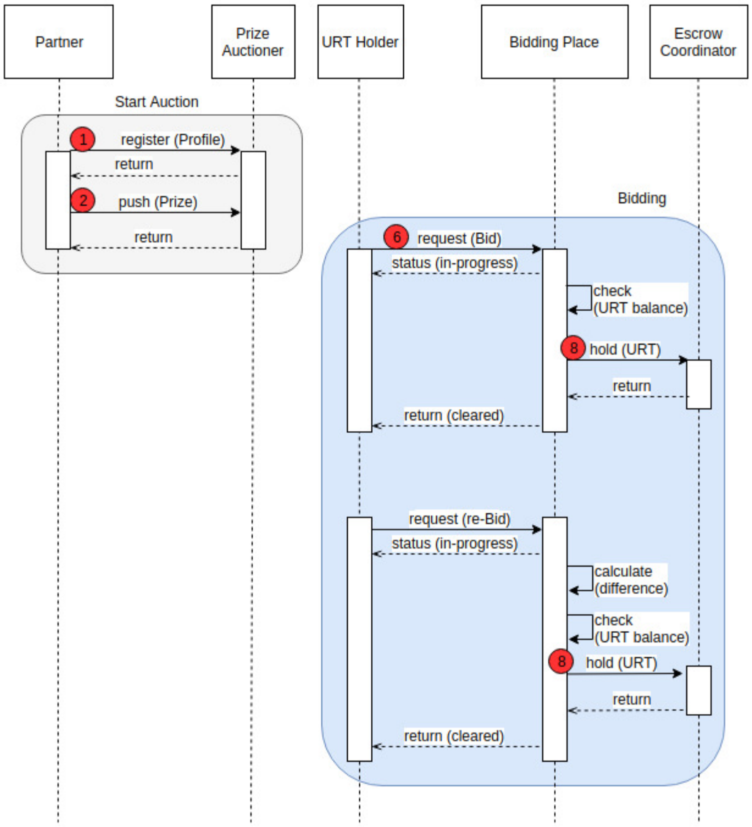


**Fig. 3.** Exchange protocol for starting an auction and bidding for gifts.

component sends a *hold(USD)* message to the *Escrow Coordinator*. The escrow operation is immutably recorded in the blockchain and when the *Escrow Coordinator* returns an acknowledgment, the *Bidding Place* is ready to return a bid acceptance message to the bidder, thus completing the action sequence. Should there be a higher bid accepted and recorded from another URT holder, the original bidder has an option to re-bid by submitting a higher bid. The data flow and sequence of events is similar to submitting an original bid with the additional step of calculating the difference between the already debited amount stored in escrow and the current bid. This difference is converted to USD and stored in escrow, followed by an acknowledgment returned to the URT holder by the bidding module.

The settlement sequence in Figure 4 is initiated when the Settle Auction signal is sent to the *Prize Auctioneer* by the *Network Operator* and recorded on chain. The *Prize Auctioneer* acknowledges that the settlement request has been received and immediately push a request to the *Bidding Place* for the winning proceeds. The *Bidding Place* in turn submits a release(URT) message to the

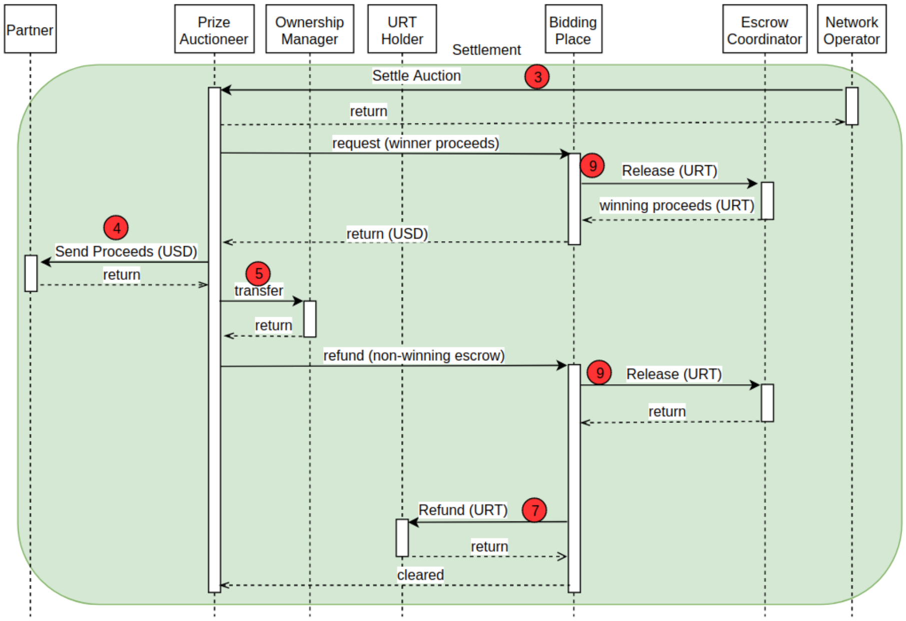


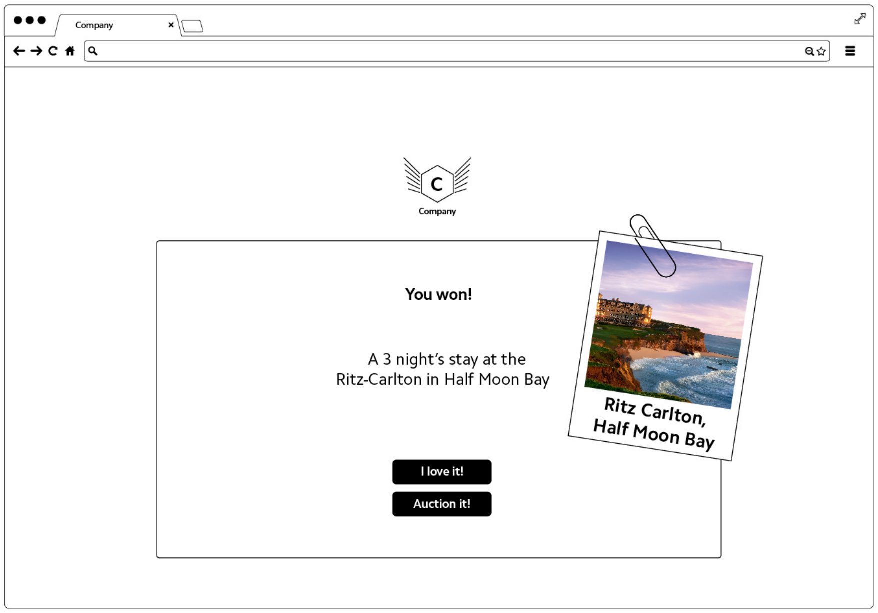
**Fig. 4.** Exchange protocol for settling a gift auction.

*Escrow Coordinator* to release the winning proceeds. Next, the *Bidding Place* forwards the funds to the *Prize Auctioneer*, which ultimately sends the winning proceeds to the *Partner* that initiated the auction. Both of these events are immutably stored on chain, to transparently track the disbursement of auction proceeds and the return of funds to non-winning bidders. The *Prize Auctioneer* next submits a request for refund of non-winning proceeds. The *Bidding Place* pushes a release message into the *Escrow Coordinator*, and forwards the URT to the original *URT holder*. These actions are also stored on chain and are denoted with labels 9 and 7 in Figure 6. Finally, once all funds have been returned, the *Binding Place* clears the auction with the *Prize Auctioneer* and the lot is settled.

### 3.4 Employee (Auctioneer) interface

The status quo of corporate recognition can be described in the following user flow. Most companies define certain milestones to celebrate as part of their employee recognition programs (either internal or through a vendor). On these milestones, employees usually get a notification that they have been issued a reward. This happens either through an internal employee portal, email, in person, or through a connected dashboard hosted and maintained by a third party, external vendor. The example in Figure 5 uses e-mail as the delivery mechanism:

An employee will receive the e-mail containing a message from the company, a description of their prize and instructions on how to redeem the prize. Most prizes can be reduced to a digital representation: gift cards have a claim or QR



You won!

A 3 night's stay at the Ritz-Carlton in Half Moon Bay

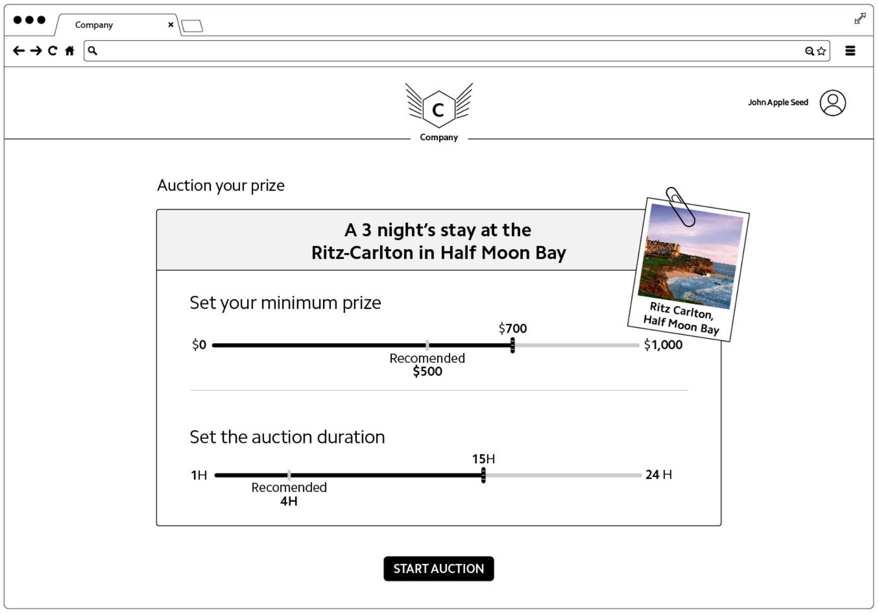
**Fig. 5.** Email to the employee who is informed about receiving a prize.

code, experiences like dinners, spas and others have an on-line voucher from the providing vendor. Once presented with the prize an employee will have an option to claim it by revealing the gift card code or having the voucher sent to them, or auction it. The *Auction it* button in Figure 6 will be provided by the URT-system, along with an API that can easily be embedded within the company's existing employee rewards flow.

Similarly, in cases of physical gifts, URT network participants will be instructed to first reveal the prize and ask their employees whether they would like to claim it or to auction it. This way, physical items will only be shipped once and to the correct recipient, either the employee or the winner of the auction. If the employee likes the prize and decides to claim it, they will be either presented or emailed the digital voucher to redeem. If they click the *Auction it* button, the page will take them to the Auction setup flow.

In Figure 6 the employee sets 2 important parameters using a slider: the minimum price they would like to receive for the item and the duration of the auction. Both parameters will have recommended values preset that will maximize the number of the participants' likelihood of a successful auction, e.g., setting the minimum too high might exclude valid bids, setting the auction duration too short might not attract a lot of bidders, etc.

Once the parameters are set, the auction goes live and the item will appear in the *Live now* tab of the URT mobile application, where URT holders can browse, view and submit bids for the auction. Once the set auction time runs



Auction your prize

A 3 night's stay at the Ritz-Carlton in Half Moon Bay

Set your minimum prize

$0 — $700 — $1,000

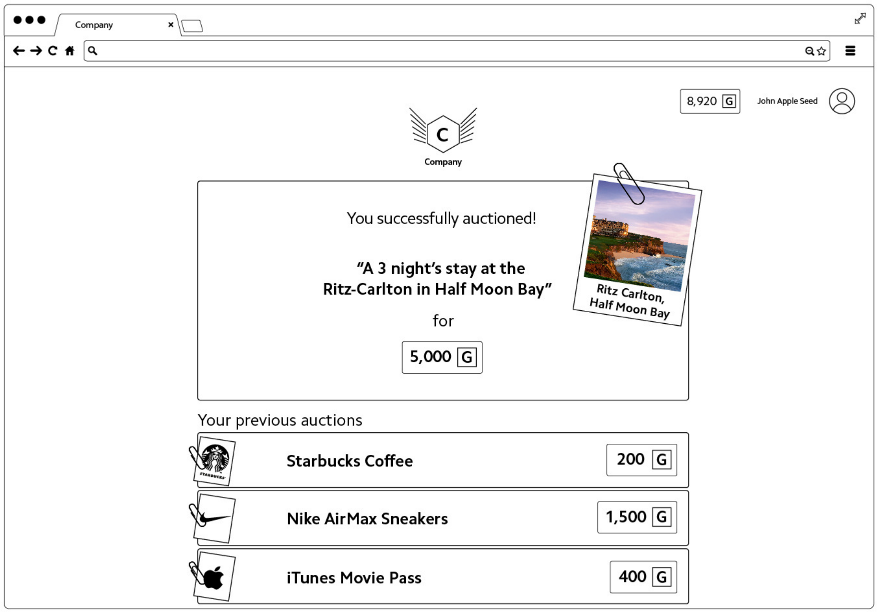
Recomended $500

Set the auction duration

1H — 15H — 24 H

Recomended 4H

**Fig. 6.** Choosing whether to accept or auction a received prize.



You successfully auctioned!

"A 3 night's stay at the Ritz-Carlton in Half Moon Bay"

for

5,000 G

Your previous auctions

| | |
|---|---|
| Starbucks Coffee | 200 G |
| Nike AirMax Sneakers | 1,500 G |
| iTunes Movie Pass | 400 G |

**Fig. 7.** Announcing a successful auction completion.

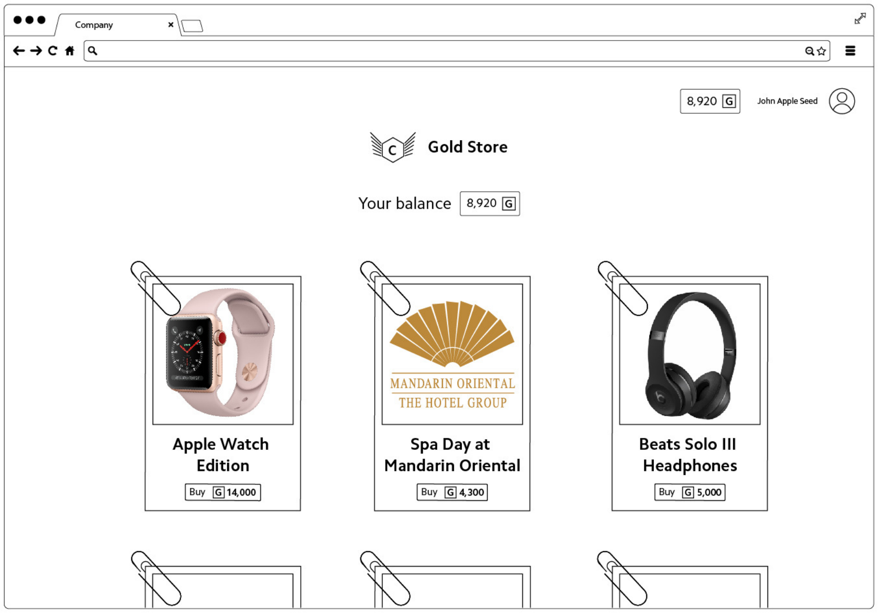


**Fig. 8.** Spending the balance on other gifts.

out, the network operator will select the highest valid bit and successfully settle the auction if the bid amount is above the minimum set by the auctioneer. Once the auction is complete as depicted in Figure 7, the employee will then be credited with the auction proceeds in either USD or the internal currency of their company or employee engagement vendor, e.g., Gold, Gems, Kudos, etc. Finally as Figure 8 shows, employees can choose to spend their balance on other prizes provided by the company, or accrue auction proceeds of multiple smaller prizes toward a large one.

### 3.5 URT holder (Bidder) interface application

Participants of the Universal Recognition Marketplace interact with the network through the URM mobile app available on both iOS and Android devices. As Figure 9 conceptually shows, the app is built to support three core functions:

1. Allowing URT holders to access their URT wallet and check their balance and transaction history.
2. Browse through a list of ongoing and upcoming auctions.
3. Submit bids on auctions.

We describe each of these functions and more in detail in the below user and app flow diagram. Those wishing to participate in the Universal Recognition Marketplace are first required to download the iOS or Android version of the URT Wallet app from the Apple Store or Google Play Store respectively.

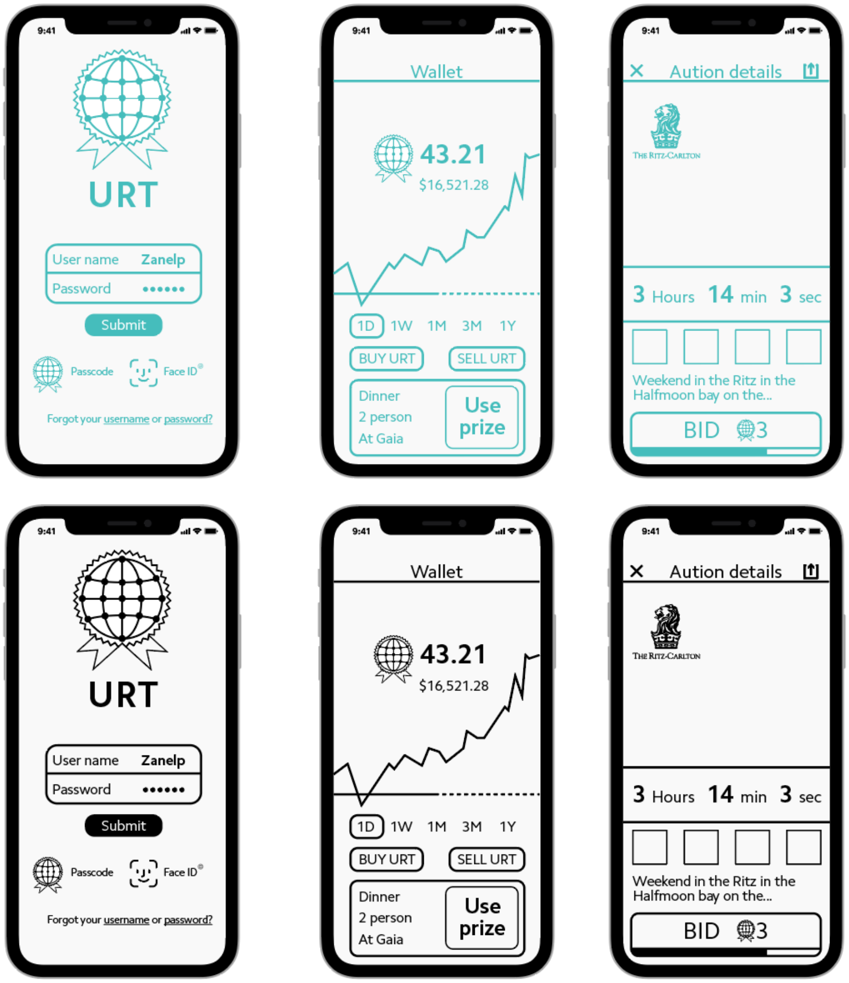


**Fig. 9.** Mobile-phone application user experience.

New users then need to register by providing either their e-mail address or phone number and a password. Once registered and verified (through a confirmation e-mail or text message), users have the option to enable more convenient authentication methods such as FaceID or passcode. Users also have the option to reset their password through a recovery link.

The main section of the URT Wallet app is the *Live Now* tab that lists all currently ongoing and upcoming auctions available for bid in the marketplace. Here users are able to browse various listings by seeing an image of the prize, brief description and information about the auction, such as how much time is remaining, how many bids have already been submitted and what the current highest bid is. From here, users have an option to expand the Auction item and view more images and details about the lot, or submit a bit by double-tap confirmation (once to set the bid, once to confirm).

The *Auction Details* view expands information about the lot, such as:

- More images and a detailed description of the item
- The vendor, location and claim restrictions (if applicable)
- The bidding history and timeline (to transparently view how many bids have been submitted at what price)
- A more functional ability to bid (by either increasing the current highest bid by a suggested amount or entering manually)

Users have an ability to set notifications for upcoming auctions that have been announced but have not started yet. They can also view past auctions and favorite certain items in case similar ones will be available in the future (they can browse their favorites list in the *Favorites* tab).

Lastly, the place where URT holders can check their current balance, transaction history and available prizes is the *Wallet* tab. The main screen shows the value of their URT holdings over a configurable period of time (day, week, month, year, all) followed by quick actions of either selling or buying more URT on the market. Here, they can also view their past transaction history including which auctions did they submit bids for including details about that auction. This list displays winning auctions at the top - URT holders who have successfully bid and won auctions, can see their available prizes that include the digital claim codes and fulfillment instructions (if provided by the vendor).

## 4 Token Economics

As a preamble for this section, we stress that the URT is planned to be issued as an ERC20 token[4] based on a Solidity smart contract template that belongs to the Ethereum system. An ERC20 smart contract template defines a set of rules that an Ethereum token has to implement. Note that there are multiple ways to convert URT to USD. The exact method will be carried out by crypto exchanges that are extrinsic to the URT system.

To meet the URT quality goals and become a universal, global, scalable and self-sufficient system, the URT network should have a built-in mechanism of value exchange. This mechanism should be based on an independent means of payment and be generic enough to accommodate for future system requirements and network growth. Strictly speaking, such a mechanism could use existing fiat payment rails or an existing blockchain-based crypto-currency. Still, using either of these would have technological and/or economical performance implications. For example, using fiat payment rails yields transactional inefficiencies. Transaction costs are likely to be high compared to the price of prizes. The reasons for an

[4] https://www.ethereum.org/token

employee (auctioneer) not achieving the maximum potential value for a prize are not instantaneous transaction times, limited geographic coverage for a single currency, a limited audience of the chosen rails, and so on. Last but not least, by now all existing fiat rails have poor to no compatibility with blockchain technology while alternatively, using Ether or another crypto-currency as a unit of account renders a) the system dependent on Ether price fluctuations uncorrelated with URT network performance and b) the used unit of account independent of the total value of prizes in the system. Essentially, Ether and many other crypto-currencies are not backed by the real value of any assets which diminishes their suitability for our needs.

### 4.1 Token model

To guarantee higher flexibility, suitability and lower technological and economic coupling of the URT network with existing payment rails, it is highly desirable that the mechanism of value exchange is independent, system-specific and linked to the value of assets it transfers. To achieve these goals, we create a URT utility token issued on the blockchain solely for use in transactions on the URT network. URT removes transactional inefficiencies and is ultimately indirectly linked to the value of all the auctioned assets in the system.

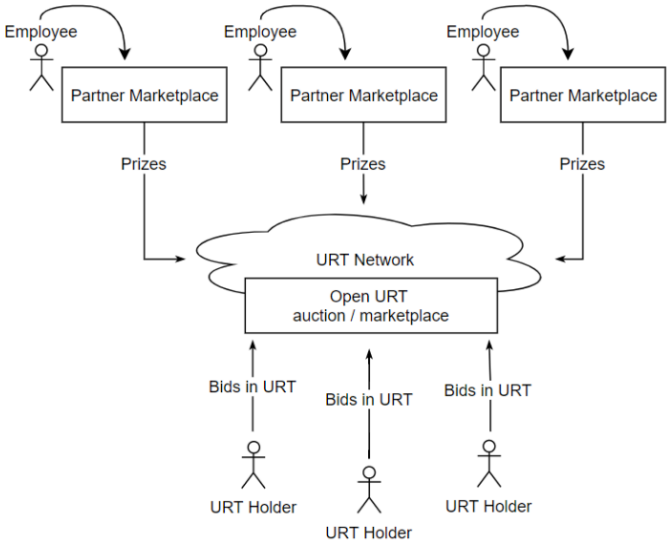


**Fig. 10.** Bidding on the URT Network.

The URT network brings liquidity to the currently largely illiquid prize pools by means of matching prize auctioneers with a community of URT holders. As Figure 10 shows, to get access to the entire URT network, participating partners need to integrate with the URT token as a common intra-network means of payment. The bids at the auctions are preformed in URT so that external participants and URT holders are able to take part in the auctions.

To understand this setup better, below we examine a common, though slightly simplified, transaction scenario and a corresponding value flow.

### 4.2 Transaction scenario

As an example, Employee A in Figure 11 places his prize at an auction on the URT platform. One URT holder bids 10 URT and another URT holder bids 15 URT. The URT holder who has made the highest bid receives the prize and Employee A receives a USD equivalent of 15 URT.

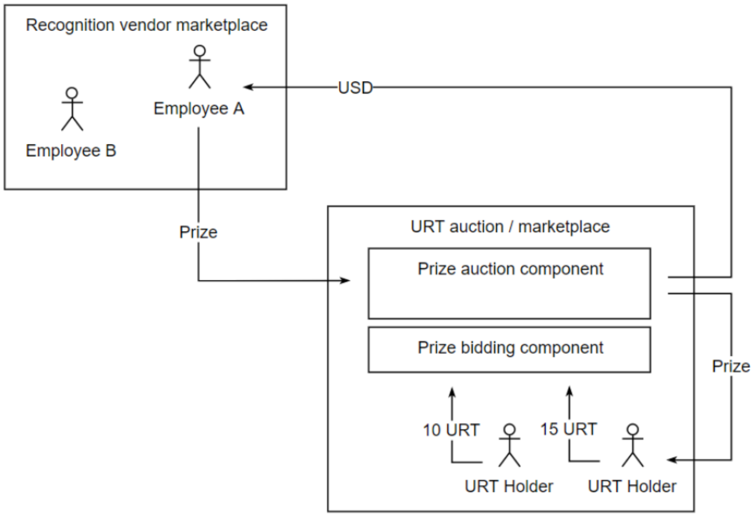


**Fig. 11.** Bidding example with employees and URT holders.

Figure 12 depicts the auction steps in further details. These steps are textually listed below.

1. The Auctioneer puts up a prize for auction and sets a minimum accepted amount in USD.
2. The Partner tells the URT Network Operator via an API that there is a prize available for auction together with the minimum amount in URT. The URT Network Operator displays the prize on the URT marketplace in an app.
3. Other members and external URT token holders bid for the prize inside the app with URT.
4. URT from bidders are automatically deposited into a special escrow smart contract until the bidding is complete.
5. Upon the completion of the auction, if the reservation price has been exceeded at the moment of completion, the highest bid is automatically released to the partner and the lower bids are returned.

6. 6. The winning URT bid is converted to USD and credited to the auctioning employee. Alternatively, the partner can choose to credit the Auctioneer with internal *points*, or *miles* that have the equivalent value of USD and are redeemable for prizes off of a fixed price prize menu – e.g., the *United Airlines Mileage Plus mall* framework.

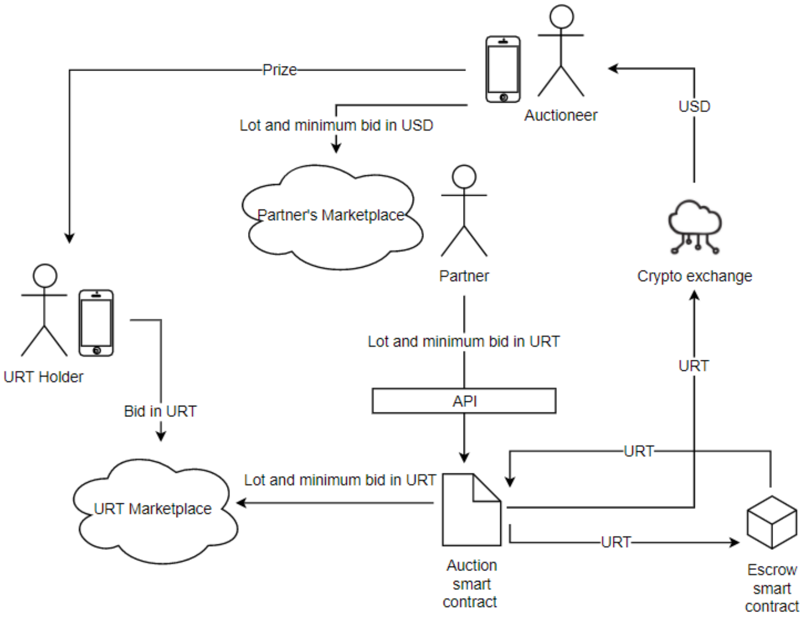


**Fig. 12.** Auctioning steps involving Partners, URT holders and Auctioneers.

The resulting value flow in this scenario is as depicted in Figure 13. A digit at each arrow represents the order in which value is transferred between parties. First, a URT Holder places a bid and transfers URT to an auction smart contract (1). Next, the auction smart contract stores URT in a special escrow smart contract (2). If that bid wins, the URT are released from escrow (3), automatically converted to USD (4), sent to an Auctioneer (5) and the prize is sent to the URT Holder (6). We omit for now the leg of the value flow is not depicted in Figure 13 in case the bid does not win. Still, the resulting protocol comprises the following steps - the bid is returned, i.e., URT are taken from escrow and returned to the URT Holder. This flow can be implemented using a single multilateral smart contract with four counterparties, or a number of bilateral smart contracts with two counter parties each. Note that conversion of URT to USD can be implemented in multiple ways. The exact method depends on the eventual choice of conversion rails and will be chosen during implementation.

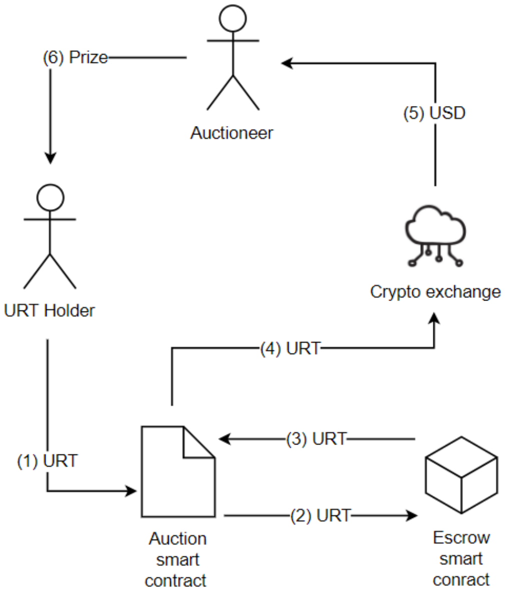


**Fig. 13.** Value flow in smart contracts.

### 4.3 System-throughput throttling

Given that it takes time to establish partnerships and a network, we use a value-based throughput throttling approach to prevent a flooding of the market with too many URT backed by too little inventory. A throughput throttling approach does not render an amount of URT permanently *locked*, or *frozen*, or impossible to exchange, or spend. Limited is the overall amount of URT acceptable by Network Operators as bids at a given point in time. If suddenly a high-demand spike occurs for prizes while Partners are not able to satisfy that demand, throughput throttling is activated.

One option to implement throughput throttling is with a URT smart contract. Each partner and network operator in Figure 14 has a special account, or wallet to receive URT. Throughput thresholds are imposed on those accounts in that the total amount of tokens on all such accounts cannot exceed a threshold at any point in time. Any URT holder can still spend their tokens by bidding and winning prizes on a first-come-first-served basis until a threshold is hit. Throughput throttling should not be regarded as a restrictive measure, though, as network operators and Partners performing URT settlements have an incentive to return received URT back on the market. This measure creates constant supply and demand by inventory refilling the short-cycles in good market conditions. Therefore, the total amount spent does not exceed the threshold at any time and assures sufficient URT liquidity is always available to match the liquidity of prize pools.

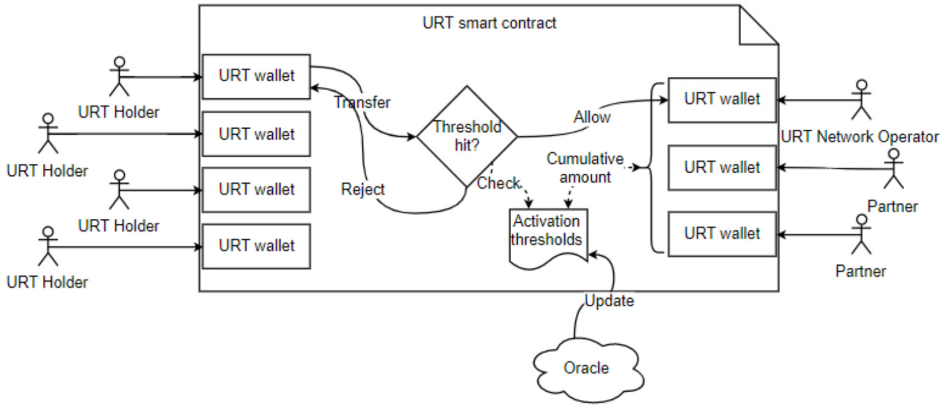


**Fig. 14.** URT throughput throttling in a smart contract.

| Liquidity, USD | 1 000 000 USD | 2 000 000 USD | 5 000 000 USD | 10 000 000 USD | 20 000 000 USD | 30 000 000 USD | 40 000 000 USD |
|---|---|---|---|---|---|---|---|
| Threshold, % of tokens | 2.5% | 5% | 12.5% | 25% | 50% | 75% | 100% |

**Table 2.** Throughput thresholds pegged to network growth.

Throughput thresholds are pegged to the network growth. The more trading volume the network has, the more URT are accepted by converters. The reference schedule in Table 2 may be programmed in a fixed way into the smart contract and allows deviations of +-10% from these thresholds for a higher degree of flexibility.

### 4.4 Joining the URT network and holding

In brief terms, different parties benefit differently from joining the URT network and holding URT. Below we list several examples:

1. Investors and network users:
- Gain access to the marketplaces of a multiplicity of Partners (recognition vendors).
- As the number of URT tokens is fixed, the value of a single URT token is expected to increase as the number of underlying partnerships and network participants (auctioneers) grows.
2. Recognition vendors release and distribute excess inventory via higher demand due to the reach of a much wider audience.
3. Employee satisfaction might increase if they have a chance to auction rewards they might not want to have and, instead, receive cash or purchase rewards that they would like to have.

# 5 Technical Challenges

The URT-system is confronted with several technical issues that require specific attention. The challenges we resolve by considering the use of blockchain technology such as smart contracts for addressing the requirements stipulated in Section 3. Briefly, blockchain technology emerged with the first use-case of the peer-to-peer crypto-currency called Bitcoin [21]. Bitcoin comprises only a small operations set above the blockchain on the protocol layer. Bitcoins use proof-of-work (PoW) as a transaction-validation algorithm. The main problem of PoW is its computational expensive, electricity intensive, lack of performance and scalability. Thus, PoW lacks efficiency for high-volume transactions with the undesirable consequence that a transaction confirmation may take over an hour [14].

Differently to Bitcoins, so-called smart-contract systems have quasi Turing-complete languages such as the lingua franca called Solidity[5] that resembles JavaScript syntax and targets for enactment, e.g., the Ethereum Virtual [29] machine. Note that Solidity is not fully Turing-complete as gas is required for transactions. Thus, when all gas is consumed then a Solidity-written smart contract stops. Ethereum is currently the de-facto smart-contract standard system that is the primary choice for issuing so-called ERC20 tokens based on a Solidity smart-contract template.

Unfortunately Ethereum suffers from several deficiencies. First, PoW for transaction validation is a scalability bottleneck so that Ethereum is not feasible for highly distribute and critical industry projects. Second, Solidity is not a formal language and no tools exist for verification before enactment [6]. The lack of verification means before enactment has resulted in a recent crowd-funding case study to a smart contract hack[6] because of security flaws. Consequently, this security exploit resulted in a loss of ca. $50 million. Ethereum hardforked and created a split yielding two Ethereum versions[7]. Another later Ethereum hardfork[8] was required due to a denial of service attack, and surely additional hardforks lie ahead[9] to introduce proof-of-stake [5] transaction validation with blockchain sharding [19].

Other limiting factors for Ethereum's industry adoption [10] are as follows. A lacking secure and stable virtual machines for better performing proof-of-stake [5] transaction validation. As the security exploit described above shows, formally verifiable smart-contract languages are very important to decide before enactment if a smart contract is sound. Another drawback is that Ethereum requires the downloading of the entire blockchain and thereby, it is not possible

[5] http://solidity.readthedocs.io/en/develop/

[6] https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/

[7] https://bitcoinsmagazine.com/articles/ethereum-classic-hard-forks-diffuses-difficulty-bomb-1484350622/

[8] https://cointelegraph.com/news/ethereum-hard-fork-no-4-has-arrived-as-dos-attacks-intensify

[9] https://forum.daohub.org/t/whats-up-with-casper-proof-of-stake-and-sharding/6309

to manage smart contracts with lite wallets that run on mobile devices that have limited storage capacity. More specifically for lite wallets on mobile-device use simple payment verification (SPV) [13] so that merely downloading block headers suffices when they connect to an arbitrary full node [21]. Consequently, the limited storage capacity of mobile devices is not a bottleneck any more.

Currently, smart-contract evaluation is a research topic for which recent publications exist. It is a challenge to write secure smart contracts [7] due to programs and pseudonymous users calling public methods of third-party programs. The consequence is an insecure combination of trusted and untrusted programs. A solution is the translation of code to the functional programming language **F*** that allows for analyzing and verifying the runtime safety and functional correctness of Ethereum contracts. For the evaluation of smart contracts [12], a set of heuristics exist to check for common pitfalls that results from empiric observations of students. Following best practices during the development phase decreases the likelihood of these common mistakes to occur. The pitfall heuristics cover the aspects of errors in encoding the state machines, failures to use cryptography, misaligning incentives, Ethereum-specific mistakes such as call-stack-, blockcash- and incentive bugs.

Another problem for the URT-system is providing an identification mechanism for the engaging stakeholders. It does not suffice when individuals merely choose an identity without any assurance of matching to the actual identity of a person. For example, employee John Smith signs up to an identification system with a different name and a mobile phone number from a prepaid card. It is subsequently possible for John Smith to sign up several times with different names and mobile phone numbers. Instead, it is important to couple an identity to an authentication process that assures identity accuracy. A good example for such a strong identity is the Estonian e-Residency card [27] where government documentation is used as authentication for an identity. The problem is that a government entity owns not only the identity, but also the means for performing a genesis authentication. If the individual holder of an e-Residency identity card becomes an undesirable in the eyes of that government, the identity card can be switched off to digitally neutralize that individual. The challenge is to establish identity authentication mechanisms for the URT-system that no government owns and where an individual retains ownership and control over the respective authenticated identity. For example, the blockchain-based identification application Civic.com merely requires an email address and a mobile phone number to sign up. No means exist in Civic.com for authenticating an identity corresponds to the actual individual who uses that application

## 6 Technical Solutions

Given the system architecture in Figure 2, for the rapid deployment of the URT-system application, several pre-existing technical solutions exist. Note that the sound understanding of URT-system requirement sets, the architecture and the behavior protocol in Section 3 are an important prerequisite for determining which technical solutions to choose for covering the respective architecture aspect. We assume that it is not realistic to have one single blockchain solution be able to cater for all diverse problems of the URT-system. Instead, each technical problem requires a study of available solutions that are carefully composed.

### 6.1 Mass-data management

For blockchain-based mass data storage about gifts and all involved stakeholders in the URT-system, the InterPlanetary File System[10] (IPFS) [4] is suitable as a content-addressable, peer-to-peer (P2P) and distributed hypermedia protocol that yields an open-source file system.

IPFS uses a highly performance, decentralized block-storage model being an alternative for the regular HTTP protocol. IPFS allows for hyper-links to address data sets in a highly performing block-storage model with data distribution across several computers. A single computer participates with storing data subsets using content addressing, hash-linked lists and allows to develop distributed blockchain applications on top by placing immutable, permanent IPFS links into a blockchain transaction.

More elaborately, IPFS connects diverse computing devices with one file system, similarly to the WWW. The main difference of IPFS is that the system acts as a single BitTorrent swarm, which exchanges objects in one Git repository. That way IPFS caters for a content-addressed block storage model that utilizes content-addressed hyperlinks to form a Merkle directed acyclic graph (DAG). Other features of IPFS are a combination of a distributed hash table without a single point of failure. IFPS is very resilient as the respective nodes are not required to trust each other, except for directly connected nodes. Differently to the HTTP protocol, distributed denial of service (DDoS) attacks [15] are prevented because of bandwidth-saving distributed content delivery. A local file in the IPFS filesystem is available via the HTTP protocol. While files are hash-identified, they are distributed using a BitTorrent-based protocol [1].

For complex operations on large datasets, the blockchain database BigchainDB [20] is a candidate system that also allows for constructing profiles. BigchainDB is decentralized in that each node is owned and controlled by a different person or organization, which allows for the establishment of a marketplace for gifts where also customer organizations maintain their own internal tokens. BigchainDB is a candidate for the URT-system because of the ability of connecting to other decentralized systems such as IPFS, and smart-contract platforms such as Ethereum, Qtum and so on. Specifically the ability of creating and moving digital assets is an important feature for virtual marketplace management.

### 6.2 Smart-contract management

The earlier discussed Ethereum platform is a potential candidate for the development of smart contracts that govern the URT-system. Unfortunately, a

[10] https://ipfs.io/

profound Ethereum bottleneck is the use a PoW algorithm that is a problem with respect to performance and scalability. Based on Section 3, it is predictable that the URT-system creates many blockchain transaction to store immutably critical events. Comparable to Ethereum, the equally non-permissioned Qtum smart-contract solution uses SPV and unspent transaction outputs (UTXO) [11] for lite wallets using merely the headers of transactions for running on mobile devices. At the moment, Qtum uses the Ethereum Virtual Machine (EVM) and according to [18], the EVM suffers from deficiencies such as earlier experienced attacks against mishandled exceptions and against dependencies such as for transaction-ordering, timestamps, and so on.

Although Ethereum has announced the development of a PoS version termed Casper [8], the eventual system release is still unclear. In Qtum, PoS is implemented and functioning while both Ethereum and Qtum use Solidity as a language for developing smart-contracts. A permissioned blockchain-system alternative to Ethereum and Qtum where blockchain access is controlled, is Hyperledger [28], an open-source distributed ledger for enterprise-scaling applications. Hyperledger comprises a set of configurable modules and Fabric [9] is a modular implementation specifically for running smart contracts to provide pluggable implementations of various functions such as token exchange between a URT-system marketplace and customer organizations with own tokens.

Finally, we stress that alternative smart-contract platforms are currently under development. For example, EOS[11] is a novel smart-contract system that positions itself as a blockchain-based operating system for the Internet. A main advantage of EOS is a transaction speed that is approximately three times faster than for Ethereum. Additionally, to overcome the issues related to the novel smart-contract language Solidity, EOS uses C++ with extensive and long-term established libraries available for smart-contract development. The plan is to mature the pre-existing C++ libraries into higher-level object-oriented libraries to improve the speed of smart-contract deployment. Besides the availability of many C++ developers, there also exists a large tool set for code checking and verification. Cardano[12] is a research-driven smart contract platform that offers PoS in a formally developed virtual machine and on top a functional smart contract language using Haskell [26] that allows for a formal verification of smart-contract code before enactment. Tezos[13] is another notable smart-contract platform under development that differentiates itself by aiming for an elaborate governance mechanism of rules for stakeholders to approve of protocol upgrades that are subsequently automatically deployed on the network. Additionally, the smart-contract language of Tezos also allows for formal verification before enactment with techniques that mathematically proves code correctness.

[11] https://eos.io/

[12] https://www.cardanohub.org/en/home/

[13] https://www.tezos.com/

### 6.3 Verifying the soundness of smart-contracts

As stressed in Section 5, developing sound and error free smart-contracts without security flaws is a challenge. Given the currently available tools, it appears that smart-contract verification before enactment is still an open research topic. Security flaws are a challenge that may lead to security exploits such as for the earlier discussed Ethereum hack of a crowd-funding smart contract that resulted in the loss of ca. $50 million. The architecture of Figure 2 suggests that an URT-system implementation utilizes a large set of cross-dependent smart contracts, which may result in concurrency conflicts or dependability issues [2]. Informally, a concurrency conflict occurs when, e.g., one component assume payment before delivery of a gift while a connected component expects an exchange the other way round. Dependability means that the exchanged data between components can be trusted to be of the correct content and format so that no processing error occurs inside the receiving component upon parsing and processing.

Several systems exist for evaluating the soundness of smart contracts that are developed in Solidity. For example, the online service called Securify[14] formally verifies smart contracts and checks coding for critical security issues. Unfortunately, currently only a beta-version of Securify is available while a big problem is the lacking Securify documentation that creates a challenge to understand which formal properties are checked in a smart contract with which specific way. We found an online provided Securify example that is checked by Securify with heuristics for transaction recordings, recursive calls, insecure coding patterns, unexpected Ether flows and the use of untrusted inputs in security operations. Furthermore, the Embark-framework[15] and Populus[16] are tools for smart-contract development and deployment. Unfortunately, also these tools are currently not sufficiently mature for any satisfactory formal verification and evaluation.

### 6.4 Identity authentication

For enabling the URT-system to have collaboration certainty, it is essential to employ a strong and flexible identification mechanism for stakeholders where also their respective authentication is assured. As a solution under development, the Authcoin identity-authentication protocol [16] is a candidate for the URT-system. The Authcoin protocol is an alternative concept to the commonly used public key infrastructures such as central authorities and the PGP Web of Trust (WoT). Authcoin combines a challenge response-based validation and authentication process for domains, certificates, email accounts and public keys, with the advantages of a blockchain-based storage system. The blockchain technology provides a publicly available, transparent and fault tolerant mechanism for storing data in a distributed and decentralized manner. Authcoin is currently

[14] https://securify.ch/

[15] https://github.com/iurimatias/embark-framework

[16] http://populus.readthedocs.io/en/latest/

developed as an application prototype using Qtum smart contracts in a mobile app realization and further explored in ongoing research with students and senior researchers from Estonia and Germany involved. Besides a first conference publication [16], an additional scholarly journal paper publication is in progress.

Briefly, the Authcoin protocol separates validation from authentication. The former proves the following three facts: First, a specific entity can access a specific account for the purpose of validation, e.g., an email account. Second, a specific entity accesses a specified private- and public key. Third, this key pair corresponds to the tested account. The authentication of Authcoin continues the validation procedure by verifying the identity and aims to confirm that the alleged owner is also the actual owner of the public key. Thus, users issue challenges for otherentities and ask they fulfill the challenges that depend on the use-case scenario, the required level of security and the given threat level of the involved entities. Either the entity fails to do so, or is able to successfully complete the challenge and creates a corresponding response.

## 7 Conclusions and Future Work

Many marketplace challenges are unchanging; how to attract and retain the best employee pool, how to provide the most attractive product or service in the marketplace at the lowest possible cost, how to contain costs while increasing revenue. However, new marketplace challenges are always presenting themselves. Among today's challenges are ensuring that capital is efficiently spent and that it produces a positive return; another challenge is to reduce employee turnover and improve employee loyalty, particularly among highly job-mobile millenial employees. Historically, there has been little or no means to ensure that expenditures on employee loyalty programs have a positive effect. Additionally, the rise of a more job-mobile and less 'anchored' workforce is recent, having only gained notice in the last 10-20 years.

Current technology offers us the means to approach such issues substantively. Via our blockchain-based and smart contract-driven system, employers can know that employees who have been recognized for their achievements are receiving more meaningful recognition gifts – because the employees have chosen those gifts themselves. Also, by tracking transaction and bidding activities, employers and prize vendors can know what gifts are most sought-after by the market, can increase the supply of desirable items and either reduce or eliminate undesirable items from the marketplace. In addition, the Bersin study (cited earlier) found that employer with "recognition-rich culture" and improved employee recognition programs experienced 31% lower rates of voluntary turnover compared with companies without those attributes.

The whitepaper elaborates upon our Universal Recognition Token system by detailing functional goals, quality goals and critical stakeholders – and their mutual relationships within the system – within our unique value proposition. We elaborate upon the system with a UML component diagram that illustrates information exchanges along data interfaces. Additionally, three protocol exchanges between the component and stakeholder entities show the dynamic URT system behavior. Key events within the protocols for starting an auction, bidding within an auction and final settlement of an auction are captured and stored within the blockchain. In selecting the set of critical blockchain transactions, our assumed goal is that potential litigation issues either do not occur at all, or are quickly settled inside of the URT system. Ideally and under normal operating conditions, no potential litigation should ever need to reach a traditional court.

Rather than using existing transaction/fulfillment systems or a cryptocurrency as a medium of exchange, we have chosen to use a token economic model in our system. This greatly improves transactional efficiency, simplifies the system requirements (existing currencies are poorly compatible with blockchain technology or else not compatible at all) and provides a direct linkage between the aggregate value of issued URT and the total value of auctioned prize assets available on the system. Auctioneers who can be employees place prizes for gifts into an auction component and URT holders may place bids with the highest bid winning the auction. The payout to the auctioneer is then performed in USD. We address the potential issues of high volatility and token price fluctuation with a mechanism of system-throughput throttling to prevent a flooding of the market with too many URT backed by too little inventory. We assume that throughput throttling can be coded into the respective smart contracts that exist for the stakeholders of the URT system.

There are several technical issues to address and overcome in a system such as ours. For example, the de facto smart-contract standard, Ethereum, suffers from key deficiencies that make it challenging to use; it suffers from performs problems for highly distributed and/or large-scale systems because of the use of proof-of-work algorithm that is computationally expensive. Ethereum also requires the downloading of the entire blockchain, making it challenging to run lite smart-contract wallets on mobile devices (which have limited data storage capacity). In addition, the current standard language for smart contracts, Solidity, is not a formal language. As such, contracts written in Solidity cannot be verified before enactment, which creates an unacceptable security risk. A secure and verifiable method of identity validation is also a firm requirement. We outline and explain our solutions to these obstacles in detail.

Finally, we discuss a paper based feasibility study we have performed; this study examines the prospect for a rapid deployment of the URT-system based on pre-existing blockchain-technology systems. In particular, our focus in this study was on specific, important aspects of the URT system such as mass-data management with blockchain technology, pre-existing solutions for smart-contract management (to include verification) and resolving the issue of identity authentication of URT-system stakeholders. Our study concludes that rapid deployment is feasible, given the set of pre-existing solutions.

## References

1. N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu. Influences on co-operation in bittorrent communities. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 111–115. ACM, 2005.
2. A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):11–33, 2004.
3. D. Bell. Uml basics: The component diagram. *IBM Global Services*, 2004.
4. J. Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.
5. I. Bentov, A. Gabizon, and A. Mizrahi. *Cryptocurrencies Without Proof of Work*, pages 142–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
6. K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, and S. Zanella-Béguelin. Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, PLAS '16, pages 91–96, New York, NY, USA, 2016. ACM.
7. K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, A. Rastogi, T. Sibut-Pinote, N. Swamy, and S. Zanella-Beguelin. Formal verification of smart contracts. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security-PLASâĂŹ16*, pages 91–96, 2016.
8. V. Buterin and V. Griffith. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*, 2017.
9. C. Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.
10. K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.
11. K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer. *On Scaling Decentralized Blockchains*, pages 106–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
12. K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi. *Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab*, pages 79–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
13. D. Frey, M.X. Makkes, P.L. Roman, F. Taïani, and S. Voulgaris. Bringing secure bitcoin transactions to your smartphone. In *Proceedings of the 15th International Workshop on Adaptive and Reflective Middleware*, ARM 2016, pages 3:1–3:6, New York, NY, USA, 2016. ACM.
14. D. Gupta, J. Saia, and M. Young. Proof of work without all the work. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, page 6. ACM, 2018.
15. F. Lau, S.H. Rubin, M.H. Smith, and L. Trajkovic. Distributed denial of service attacks. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 3, pages 2275–2280. IEEE, 2000.
16. B. Leiding and A. Norta. Mapping requirements specifications into a formalized blockchain-enabled authentication protocol for secured personal identity assurance. In *International Conference on Future Data and Security Engineering*, pages 181–196. Springer, 2017.

17. K. Luthans. Recognition: A powerful, but often overlooked, leadership tool to improve employee performance. *Journal of Leadership Studies*, 7(1):31–39, 2000.
18. L. Luu, D.H. Chu, H. Olickel, P. Saxena, and A. Hobor. Making Smart Contracts Smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 254–269, 2016.
19. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 17–30, New York, NY, USA, 2016. ACM.
20. T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto. Bigchaindb: a scalable blockchain database. *white paper, BigChainDB*, 2016.
21. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.
22. M. Pilkington. 11 blockchain technology: principles and applications. *Research handbook on digital transformations*, page 225, 2016.
23. D. Rosu, Dionne M. Aleman, J.C. Beck, M. Chignell, M. Consens, M.S. Fox, M. Grüninger, C. Liu, Y. Ru, and S. Sanner. A virtual marketplace for goods and services for people with social needs. In *Humanitarian Technology Conference (IHTC), 2017 IEEE Canada International*, pages 202–206. IEEE, 2017.
24. J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The (2Nd Edition)*. Pearson Higher Education, 2004.
25. L. Sterling and K. Taveter. *The art of agent-oriented modeling*. MIT Press, 2009.
26. S. Thompson. *Haskell: the Craft of Functional Programming . International Computer Science Series*. Addison-Wesley, March, 1999.
27. L. Uljala and A. Scupola. Virtual entrepreneurship and e-residency adoption. In *Network, Smart and Open*, pages 71–84. Springer, 2018.
28. M. Vukolić. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, pages 3–7, New York, NY, USA, 2017. ACM.
29. G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014.