Supporting Interface Experimentation for Blockchain Applications

Michael Froehlich*
Center for Digital Technology and Management, Germany
froehlich@cdtm.de

Benjamin Moser
Ludwig Maximilian University, Germany
benjamin.moser@campus.lmu.de

Florian Alt
University of the Bundeswehr Munich, Germany
florian.alt@unibw.de

Albrecht Schmidt
Ludwig Maximilian University, Germany
albrecht.schmidt@ifi.lmu.de

ABSTRACT

There is an increasingly diverse range of smart-contract blockchains on which decentralized applications (dApps) are built. However, HCI research has so far failed to address them, focusing primarily on Bitcoin and Ethereum. This is problematic as these new blockchains come with an increasingly diverse set of properties that influence the usability of dApps for end-users. For blockchain interface design guidelines to be valuable for practitioners, they need to acknowledge the heterogeneity of blockchains. However, evaluating novel interface concepts across different blockchains is resource-intensive as each blockchain has to be integrated manually, slowing down research. To address this challenge, we propose a system to support interface experimentation for blockchain applications. The system allows researchers and developers to connect interfaces to a unified API simulating different blockchains and facilitates the configuration, distribution, and evaluation of online experiments. A preliminary evaluation showed promising results.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Applied computing** → **Digital cash**; • **Information systems** → **Digital cash**; • **General and reference** → *Experimentation.*

KEYWORDS

blockchain, cryptocurrency, dapps, web3, hci, interface design, experimentation, support tools

ACM Reference Format:
Michael Froehlich, Benjamin Moser, Florian Alt, and Albrecht Schmidt. 2022. Supporting Interface Experimentation for Blockchain Applications. In *Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference (NordiCHI Adjunct '22), October 8–12, 2022, Aarhus, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3547522.3547676

*Also with Ludwig Maximilian University, University of the Bundeswehr Munich.

NordiCHI Adjunct '22, October 8–12, 2022, Aarhus, Denmark

ACM ISBN 978-1-4503-9448-2/22/10.
https://doi.org/10.1145/3547522.3547676

1 INTRODUCTION

There is an increasingly diverse landscape of blockchain application platforms to develop with [10]. While a few years ago Ethereum was the only smart-contract blockchain available, today alternatives like Cosmos, Solana, Polkadot, or Polygon have emerged and gained traction among developers [15]. At the same time, extant interaction design research on blockchain and cryptocurrency has overwhelming focused on Bitcoin and Ethereum, neglecting other chains [10]. This gap is problematic as these new blockchains offer developers fundamentally different properties – for example w.r.t. transaction speed, throughput, and fees – which in turn influence how end-users can interact with the built decentralized applications (dApps). Taking the researchers' perspectives it is not difficult to see how this gap has formed: Prototyping and evaluating interfaces for different blockchains requires substantial resources, as each blockchain needs to be manually integrated. This consequently makes it costly to experiment with interface concepts on several blockchains and, as a field, has kept us from understanding the heterogeneous effects different blockchain properties may have on application design.

Let's take the design of interface elements for the communication of transaction stati as an example: Previous literature documents that users find transactions hard to understand and misconceptions are frequent (see e.g. [9, 11, 14, 16]). For designers and developers this begs the question, how to best design interface elements that communicate the status and expected completion of a transaction clearly and unambiguously. The non-deterministic nature of blockchains – validating nodes can independently decide which transactions to include in the next block – makes this a non-trivial task. The completion of a transaction may depend on the frequency at which blocks are created, the current state of the network, and the amount of fees allocated for the specific transaction. These properties are all connected to the infrastructure provided by the underlying blockchain a dApp is built on. For example, even simple transactions may take between tens of minutes (e.g. Bitcoin), a few minutes (e.g. Ethereum), and a few seconds (e.g. Bitcoin Lightning or Solana) depending on the blockchain. Design guidelines for such interface elements would thus need to acknowledge the heterogeneity of blockchains and their properties to be valuable for practitioners.

Consequently, to create such guidelines for blockchain interfaces, it is necessary to design interfaces and evaluate them across different blockchains. To address this challenge, we propose a system to support interface experimentation for blockchain applications.

The proposed system allows researchers and developers to connect their interfaces to a unified API that simulates different blockchains and provides a management interface to configure, distribute, and evaluate online experiments. We present an early implementation of the system and report the results of a preliminary study with N=160 participants on Amazon Mechanical Turk (mTurk).

2 PROTOTYPE

We developed a system to support rapid interface experimentation for blockchain applications. In the following we lay out the requirements, its architecture, and implementation.

2.1 Use Case and Requirements

We illustrate the envisioned use case by contrasting an *as-is-scenario* with a *visionary-scenario* [4]. The system has two actors: the interface DEVELOPER and the study PARTICIPANT.

Table 1: Use-Case: As-Is-Scenario and Visionary-Scenario

Situation
Dora is an interface developer for a mobile social payment app that supports multiple cyrptocurrencies. By analyzing comments on the app store she notices that some users complain that transactions are sometimes taking too long to complete or even get stuck. After conducting desk research and some user interviews she realizes that new users often do not understand the connection between fee-amount and transaction speed and thus face difficulties to select the right fee. She decides to prototype different input elements and test them with users before suggesting changes to the production app. She wants to understand which input elements help users select the appropriate fees and is interested in understanding whether different cryptocurrencies require different input elements.
As-Is-Scenario: Dora implements the different input elements on different branches of the Github repository. After collecting qualitative input from a small sample, she wants to test the different interfaces in an online experiment. Due to cost constraints she cannot distribute real cryptocurrency to participants. Instead, she decides to mock the sending of transactions and fees. For each cryptocurrency she starts implementing realistic behavior mocking the fees and transaction speed for the specific experiment she has in mind. After completing the implemention, she deploys the app and creates a document outlining the task instruction for the participants. In another tool she creates a questionnaire. Finally, creates four tasks on Amazon mTurk, each linking to a different version of the app and distributes her experiment.
Visionary-Scenario: Dora prototypes four input elements on a new branch of the Github project. She integrates the API of the blockchain experimentation system. Based of the programatic assignment through the API the respective input element is rendered. To mock sending transactions, she uses the unified interface of the API. She sends the respective cryptocurrency, the amount, and selected transaction, and additional transaction details and the API returns the status of the transaction. After completing the implementation she switches to the web-interface of the experimentation system. She configures the study procedure, adds a questionnaire step and an experimentation step with an appropriate task description. She configures the simulated cryptocurrencies and tested input elements. With the generated link she distributes the experiment via Amazon mTurk.

From the described use case we derived several functional requirements [4] for the system:

- **R1 Blockchain Simulation:** The system should allow the simulation of different blockchains and their core properties. It should provide an common interface to simulate transactions on the supported blockchains to decouple interface implementation and evaluation from the specific blockchain implementation.
- **R2 Experiment Management:** The system should support the configuration and management of experiments. It should be configurable with respect to which cryptocurrencies and which interface variations are part of an experiment and manage subsequent randomized assignment of participants.
- **R3 Rapid Dissemination:** The system should enable a fast dissemination of experiments. Task descriptions for participants and questionnaires should be integrated into the experimentation system to allow for fast distribution.

2.2 Conceptual Architecture

We decomposed the proposed system into several components with specific responsibilities. Figure 1 provides an overview of the conceptual architecture. The experimentation system comprises three subsystems: The BLOCKCHAIN SIMULATOR bundles blockchain simulation functionality and exposes an REST API that integrates with the interface prototypes of the DEVELOPER. The EXPERIMENT CONFIGURATOR provides a management interface for the DEVELOPER to configure and monitor their experiments. The STUDY DISSEMINATION subsystem manages the distribution of the experiment to PARTICIPANTS in accordance to the configuration of the experiment.

The decompostion in subsystem has several advantages. First, decoupling functionality allows composability and re-use. For example, experiments could be easily repeated with different interfaces by duplicating experiment configurations. Second, it allows for maintainability and extensibility. By exposing only a limited interface to other components, the underlying implementation can be changed or improved in the future. For example, new blockchain simulations could be added without affecting existing experiments or the simulation of a specific blockchain could be implemented in a more advanced way.

2.3 Implementation

We implemented the proposed system in a first prototype. We realized the experimentation system using NodeJS[1], ExpressJS[2] and MongoDB[3]. The implemented system supports an abstraction layer to simulate the Bitcoin and Ethereum blockchain and allows for the integration of additional blockchains in the future. Figure 2 shows a low-fidelity interface prototype and the actual realized interface of the experimentation system.

The interface of the EXPERIMENT CONFIGURATOR shows three main pages (see navigation bar on the left). The Blockchain page shows the blockchains that can be simulated. The Questionnaires page shows an overview of existing questionnaires and allows to created new ones. Finally, the Experiments pages shows an overview of the created experiment, allows to create new ones, and configure existing ones. The configuration of an experiment comprises chaining different tasks together, i.e. questionnaire tasks or experiment

[1]https://nodejs.org/ (last-accessed: 2022-05-21)
[2]http://expressjs.com/ (last-accessed: 2022-05-21)
[3]https://www.mongodb.com/ (last-accessed: 2022-05-21)

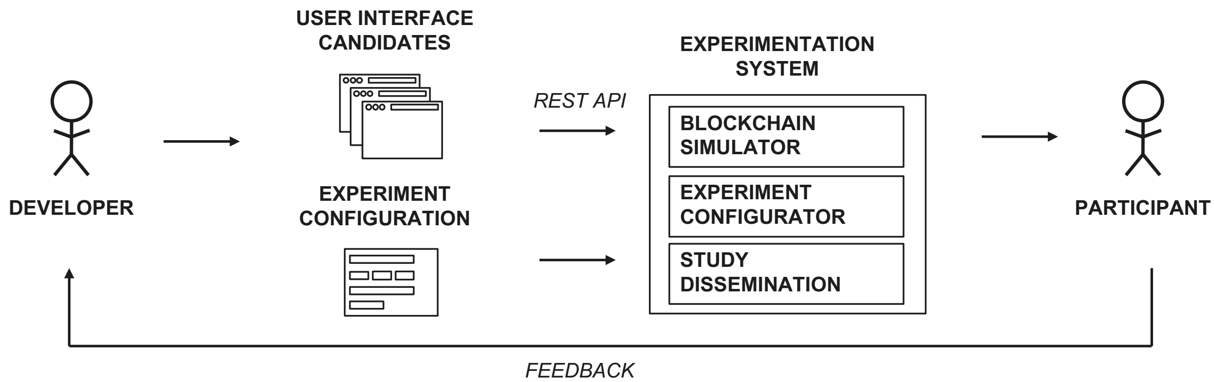

Figure 1: Conceptual overview of the prototyped system.

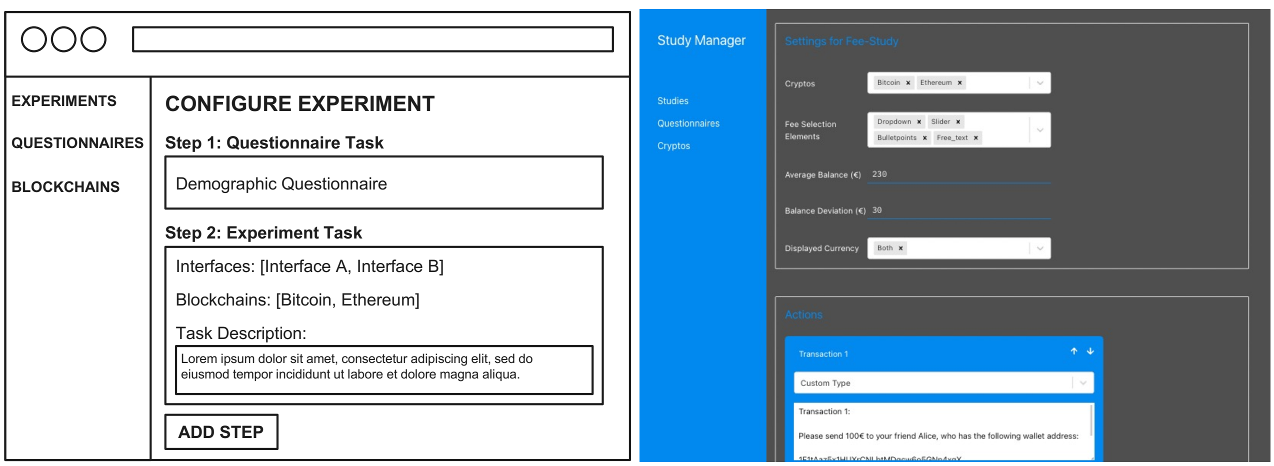

Figure 2: The interface of the Experimentation System: A low fidelity prototype (left) and the realized implementation (right).

tasks. Experiment tasks require specific configuration: identifier for the respective interfaces, which blockchain simulations to use, and the task descriptions that are shown to participants.

3 EVALUATION AND RESULTS

To evaluate the system we designed and ran an initial experiment with it. The experiment evaluated four types of input elements (free input, select, dropdown, slider) for sending transactions with two cryptocurrencies (Bitcoin, Ethereum) in a between-subject online experiment. The main purpose of running the experiment was to test designed system under realistic conditions.

3.1 Experimental Setup

We used a between-subject design to compare different interface elements for selecting fees when sending a transaction in an online experiment with n=160 participants who we recruited from Amazon mTurk. There were 8 experimental conditions (4 input elements times 2 cryptocurrencies). We recruited in total 160 participants who were randomly assigned to one condition by our system. The instructions for the experiment were provided within our system and could be accessed by participants using a dedicated button at all times. Additionally, participants had to fill a questionnaire after completing the user study.

Procedure: During the study participants were provided with three task descriptions asking them to consider a specific scenario under which they should send a cryptocurrency transactions. The task description contained cues about the expected speed at which the user would like the transaction to complete to induce interaction during the fee selection process. For example, "*Please send 10€ to your colleague Tim [...] He made it clear that he needs the money within 30 minutes.*". Participants conducted these transactions with a mobile wallet interface presented in the browser. At the start of the study they were randomly assigned to one of the conditions, which remained the same during the study. The wallet interface integrated the developed system and displayed the respective interface elements.

Collected Data and Hypotheses: We collected several metrics and variables to understand system performance and users' perception. We were specifically interested in the **perceived usability**, **time needed for the fee selection**, and **the selected fee value**. Our hypotheses for all collected output variables were that there would be a difference between cryptocurrencies and input elements.

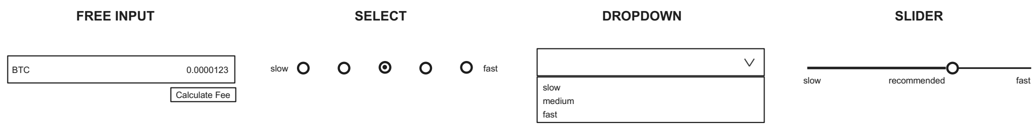

Figure 3: Illustrations of the different input elements used to select the transaction fee during the experiment.

3.2 Results

In total 160 people participated in the experiment. The median age was 32 years. 57 (35.6%) were female, the other 103 (64.4%) were male. 76.9% reported being from the USA. Two thirds of participants had previously made a cryptocurrency transaction. Overall, the analysis showed that the system could successfully be used to collect experimental data, did not suffer performance issues, and supported the intended use cases. We observed that in 15.6% of tasks were not completed correctly. Common mistakes included incorrectly entered receiver addresses or transaction amounts, also reported in literature (see e.g. [7, 14]).

Table 2: Overview of the collected metrics (arithmetic mean).

	Usability	Time	Values
free input	64.55	39.9 sec	0,0668 ETH
select	56.59	15.0 sec	0,0178 ETH
dropdown	60.43	9.7 sec	0,0247 ETH
slider	69.79	11.9 sec	0,0178 ETH
BTC	60.33	18.9 sec	67,13 sat/byte
ETH	64.81	19.1 sec	0,0283 ETH
Total	62.48	19.0 sec	–

Usability: Overall, the mean SUS score [3] was 62.48, which is below average usability compared to general consumer apps [13] and comparable to existing cryptocurrency apps [8, 9]. Regarding input elements, usability was highest (69.79) for the slider. A Kruskal-Wallis test showed significant difference between input elements. The respective Dunn-Bonferroni post-hoc analysis showed differences are only significant between *slider-select* (p=0.006). A Mann-Whitney-U-Test did not find statistical significant differences between cryptocurrencies.

Time: The average time to select a fee was between 9.7 and 39.9 seconds depending on the input element. The *free input* element required the most time. A Welch's Anova showed difference between groups. A post-hoc pairwise Games-Howell test showed statistically significant differences between *free input-dropdown* (=0.024) and *free input-slider* (p=0.028).

Fee Value: A Kruskal-Wallis test showed statistically significant differences in selected transaction fee value by input element for Ethereum (p<0.001) but not for Bitcoin. For Ethereum, pairwise post-hoc comparisons show statistically significant differences for *slider-free input* (p=0.006), *select-free input* (p=0.001), *dropdown-free-input* (p=0.002).

4 DISCUSSION

Overall, the preliminary evaluation of the developed system showed promising results. The implemented system could fulfill the initial requirements and facilitate a blockchain interface experiment including configuration, simulation, and distribution. We did not encounter any technical problems or load issues during the experiment. While mTurk has established itself as popular platform for microtasks and research [12], we observed that some participants attempted to cheat the system – i.e. they they just tried to enter bogus data and click through the prototype as fast a possible. This behavior is in line with previous findings (see e.g. [1]) discussing data quality of mTurk.

4.1 Limitations & Future Work

The results presented in this paper are not without limitations. The implementation of the proposed system is an early version with room for future development. The evaluation presented primarily serves to demonstrate the feasibility of the approach. Future evaluations of the system should aim to understand whether using the system enables researchers and developers to improve their workflow in a more holistic ways. The experimental comparison of input element was conducted with a small sample and tentative findings presented here should be complemented by qualitative research in the future to support interface designers.

Future System Development: The described system is early technical work. To unlock its full value for researchers and developers it will require a larger set of blockchains to be available for simulation. While there new generations of blockchain have become available for developers to build on, a recent literature review shows that there is a research gap in HCI concerning studies that go beyond Bitcoin and Ethereum [10]. A second point for future development concerns the level of sophistication at which blockchain transactions can be simulated. While simulation of simple transactions enables interface experimentation with regards to sending and receiving cryptocurrency, a larger design space can arguable found in the area of smart contracts and dApps [5, 10]. Thus, a way to realistically simulate transactions calling smart contracts on various blockchains may be beneficial for more complex blockchain experiments. From a technical point-of-view, this could be achieved by integrating more sophisticated systems to simulated the underlying blockchain (see e.g. [2, 6]). Another opportunity would be the possibility to not just simulate blockchains, but replay specific states of the blockchain. This would allow testing hypotheses related to cryptocurrency valuations and network congestion.

Future System Evaluation: With researchers and developers being the primary users of the system, future evaluation should test whether the system delivers value to them. This includes questions related to their user experience integrating the API and running the experiments with the system as well as more objective measure like the time and resource savings generated through use of the system. Additionally, future evaluations should analyze whether data gathered via mTurk fulfills the required standards for scientific research in this context and, if not, implement additional services.

5 CONCLUSION

This paper presents a system to support interface experimentation for blockchain applications. In a preliminary evaluation it shows promising results for reducing the time and effort needed to conduct experiments with novel users interfaces. We would like to engage with the HCI community at NordiCHI to discuss how the system could be extended to support researchers, designers, and users beyond experiment driven evaluation of novel interfaces. In line with the conference's themes we would like to explore how users could be empowered to participate not just in the evaluation but the design process itself.

REFERENCES

[1] Douglas J Ahler, Carolyn E Roush, and Gaurav Sood. 2019. The micro-task market for lemons: Data quality on Amazon's Mechanical Turk. *Political Science Research and Methods* (2019), 1–20.

[2] Maher Alharby and Aad van Moorsel. 2019. BlockSim: A Simulation Framework for Blockchain Systems. *SIGMETRICS Perform. Eval. Rev.* 46, 3 (jan 2019), 135–138. https://doi.org/10.1145/3308897.3308956

[3] John Brooke. 1996. SUS: a 'quick and dirty' usability scale. *Usability evaluation in industry* (1996), 189.

[4] Bernd Bruegge and Allen H. Dutoit. 2009. *Object-Oriented Software Engineering Using UML, Patterns, and Java* (3rd ed.). Prentice Hall Press, USA.

[5] Chris Elsden, Arthi Manohar, Jo Briggs, Mike Harding, Chris Speed, and John Vines. 2018. *Making Sense of Blockchain Applications: A Typology for HCI.* Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3174032

[6] Carlos Faria and Miguel Correia. 2019. BlockSim: Blockchain Simulator. In *2019 IEEE International Conference on Blockchain (Blockchain)*. 439–446. https://doi.org/10.1109/Blockchain.2019.00067

[7] Michael Froehlich, Philipp Hulm, and Florian Alt. 2021. Under Pressure. A User-Centered Threat Model for Cryptocurrency Owners. In *2021 the 4th International Conference on Blockchain Technology and Applications* (Xi'an, China) *(ICBTA 2021)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3510487.3510494

[8] Michael Froehlich, Charlotte Kobiella, Albrecht Schmidt, and Florian Alt. 2021. *Is It Better With Onboarding? Improving First-Time Cryptocurrency App Experiences*. Association for Computing Machinery, New York, NY, USA, 78–89. https://doi.org/10.1145/3461778.3462047

[9] Michael Froehlich, Maurizio Raphael Wagenhaus, Albrecht Schmidt, and Florian Alt. 2021. *Don't Stop Me Now! Exploring Challenges Of First-Time Cryptocurrency Users.* Association for Computing Machinery, New York, NY, USA, 138–148. https://doi.org/10.1145/3461778.3462071

[10] Michael Froehlich, Franz Waltenberger, Ludwig Trotter, Florian Alt, and Albrecht Schmidt. 2022. Blockchain and Cryptocurrency in Human Computer Interaction: A Systematic Literature Review and Research Agenda. In *Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 155–177. https://doi.org/10.1145/3532106.3533478

[11] Michael Fröhlich, Felix Gutjahr, and Florian Alt. 2020. *Don't Lose Your Coin! Investigating Security Practices of Cryptocurrency Users.* Association for Computing Machinery, New York, NY, USA, 1751–1763. https://doi.org/10.1145/3357236.3395535

[12] Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 453–456. https://doi.org/10.1145/1357054.1357127

[13] James R Lewis. 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.

[14] Alexandra Mai, Katharina Pfeffer, Matthias Gusenbauer, Edgar Weippl, and Katharina Krombholz. 2020. User Mental Models of Cryptocurrency Systems - A Grounded Theory Approach. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. USENIX Association, 341–358. https://www.usenix.org/conference/soups2020/presentation/mai

[15] Maria Shen and Avichal Garg. 2022. *Developer Report 2021.* Electric Capital. Retrieved 2022-02-11 from https://github.com/electric-capital/developer-reports/blob/master/dev_report_2021_updated_012622.pdf

[16] Artemij Voskobojnikov, Oliver Wiese, Masoud Mehrabi Koushki, Volker Roth, and Konstantin (Kosta) Beznosov. 2021. The U in Crypto Stands for Usable: An Empirical Study of User Experience with Mobile Cryptocurrency Wallets. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 642, 14 pages. https://doi.org/10.1145/3411764.3445407