Case Studies

Bartoletti and Pompianu [5] identify five classes of smart contract applications: Financial, Notary, Game, Wallet, and Library. Our case studies include at least one application per category (Table 1). In addition, we consider scalability solutions.

Financial. These apps include digital tokens, crowdfunding, escrowing, advertisement, insurances and sometimes Ponzi schemes. A study investigating all blocks mined until September 15th, 2018 [9], found that 72.9 % of the high-activity contracts are token contracts compliant to ERC-20 or ERC-721, which have an accumulated market capitalization of US $ 12.7 billion. We have implemented a fungible Prisma token of the ERC-20 standard. Further, we implemented crowdfunding and escrowing case studies. These case studies demonstrate how to send and receive coins with Prisma, which is the basic functionality of financial applications. Other financial use cases can be implemented in Prisma with similar techniques.

Notary. These contracts use the blockchain to store data immutably and persistently, e.g., to certify their ownership. We implemented a general-purpose notary contract enabling users to store arbitrary data, e.g., document hashes or images, together with a submission timestamp and the data owner. This case study demonstrates that Notaries are expressible with Prisma.

Games. We implemented TicTacToe (Section 2), Rock-Paper-Scissors, Hangman and Chinese Checkers. Hangman evolves through multiple phases and hence benefits from the explicit control flow definition in Prisma more than the other game case studies. The game Chinese Checkers is more complex than the others, in regard to the number of parties, the game logic and the number of rounds, and hence, represents larger applications. Rock-Paper-Scissors illustrates how randomness for dApps is securely generated. Every Ethereum transaction, including the executions of contracts, is deterministic – all participants can validate the generation of new blocks. Hence, secure randomness is negotiated among parties: in this case, by making use of timed commitments [3], i.e., all parties commit to a random seed share and open it after all commitments have been posted. The contract uses the sum of all seed shares as randomness. If one party aborts prior to opening its commitment, it is penalized. In Rock-Paper-Scissors both parties commit to their choice – their random share – and open it afterwards. Other games of chance, e.g., gambling contracts, use the same technique.

Wallet. A wallet contract manages digital assets, i.e., cryptocurrencies and tokens, and offers additional features such as shared ownership or daily transaction limits. At August 30, 2019, 3.9 M of 17.9 M (21 %) deployed smart contracts have been different types of wallet contracts [4]. Multi-signature wallets are a special type of wallet that provides a transaction voting mechanism by only executing transactions, which are signed by a fixed fraction of the set of owners. Wallets transfer money and call other contracts in their users stead depending on run-time input, demonstrating calls among contracts in Prisma. Further, a multi-signature wallet uses built-in features of the Ethereum VM for signature validation, i.e., data encoding, hash calculation, and signature verification, showing that these features are supported in Prisma.

Libraries. As the cost of deploying a contract increases with the amount of code in Ethereum, developers try to avoid code repetitions. Contract inheritance does not help: child contracts simply copy the attributes and functions from the parent. Yet, one can outsource commonly used logic to *library contracts* that are deployed once and called by other contracts. For example, the TicTacToe dApp and the TicTacToe channel in our case studies share some logic, e.g., to check the win condition. To demonstrate libraries in Prisma, we include a TicTacToe library to our case studies and another on-chain executed TicTacToe dApp which uses such library instead of deploying the

Author's address:

logic itself. Libraries use a call instruction similar to wallets, although the call target is typically known at deployment and can be hard-coded.

Scalability solutions. State channels [6–8] are scalability solutions, which enable a fixed group of parties to move their dApp to a non-blockchain consensus protocol: the execution falls-back to the blockchain in case of disputes. Similar to multi-signature wallets, state channels use built-in signature validation. We implemented a state channel for TicTacToe[1] to demonstrate that Prisma supports state channels.

Table 1. Categories and Cross-tier calls.

Category	Case study
Financial	Token
	Crowdfunding
	Escrow
Wallet	Multi-signature wallet
Notary	General-purpose notary
Game	Rock Paper Scissors
	TicTacToe
	Hangman
	Chinese Checkers
Library	TicTacToe library
	TicTacToe using library
Scalability	TicTacToe channel

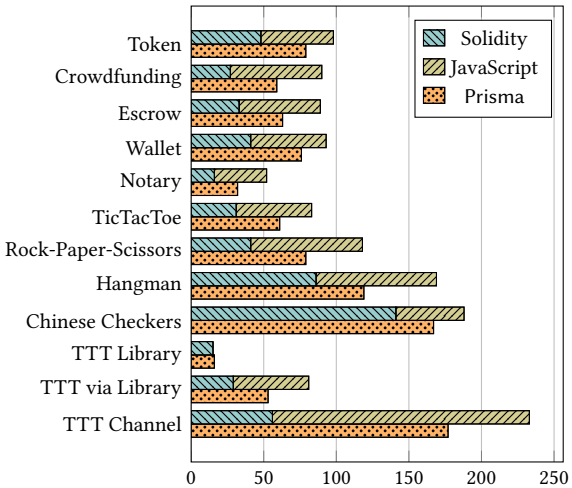

Fig. 2. LOC in Solidity/JavaScript and Prisma.

[1] A general solution is a much larger engineering effort and subject of industrial projects [1, 2]

REFERENCES

[1] 2020. Perun Network. https://perun.network.

[2] 2020. Statechannels. https://statechannels.org/.

[3] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. 2014. Secure Multiparty Computations on Bitcoin. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*. IEEE Computer Society, 443–458. https://doi.org/10.1109/SP.2014.35

[4] Monika Di Angelo and Gernot Salzer. 2020. Wallet Contracts on Ethereum. *CoRR* abs/2001.06909 (2020). arXiv:2001.06909 https://arxiv.org/abs/2001.06909

[5] Massimo Bartoletti and Livio Pompianu. 2017. An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International conference on financial cryptography and data security*. Springer, 494–509.

[6] Stefan Dziembowski, Lisa Eckey, Sebastian Faust, Julia Hesse, and Kristina Hostáková. 2019. Multi-party Virtual State Channels. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11476)*, Yuval Ishai and Vincent Rijmen (Eds.). Springer, 625–656. https://doi.org/10.1007/978-3-030-17653-2_21

[7] Stefan Dziembowski, Sebastian Faust, and Kristina Hostáková. 2018. General State Channel Networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang (Eds.). ACM, 949–966. https://doi.org/10.1145/3243734.3243856

[8] Andrew Miller, Iddo Bentov, Surya Bakshi, Ranjit Kumaresan, and Patrick McCorry. 2019. Sprites and State Channels: Payment Networks that Go Faster Than Lightning. In *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11598)*, Ian Goldberg and Tyler Moore (Eds.). Springer, 508–526. https://doi.org/10.1007/978-3-030-32101-7_30

[9] Gustavo A Oliva, Ahmed E Hassan, and Zhen Ming Jack Jiang. 2020. An exploratory study of smart contracts in the Ethereum blockchain platform. *Empirical Software Engineering* (2020), 1–41.