Platonic.Systems

for Project Ardana

Audit

Authors

Quinn Dougherty `quinn.dougherty@platonic.systems`

Bassam Awad `bassam.awad@platonic.systems`

November 14, 2021 23:43

Table of Contents

List of beliefs

I. Executive summary

Platonic.Systems has conducted an internal Audit parallel to the engineering efforts building `Danaswap`, Ardana stablecoins, and `DANA` governance token mechanisms.

I.1. Recommendations

Recommendations are assigned a five-valued confidence.

I.1.1. Keep Ardana components closed source, or under complex licensing, to prevent forking

This can play a role in reducing vampire attack risk. Confidence in importance: very low

I.1.2. Peg fee structure to governance (i.e. `DANA` holders) to avoid competition

This reduces vampire attack risk, details IV.3.3. Confidence in importance: medium.

I.1.3. Monitor development of Cardano ecosystem for multi-step atomic transactions to guard against flashloan attacks

With mitigation strategy sketches provided in IV.4.1. Confidence in importance: very high

I.1.4. Enforce `aeson >= 2.0.1.0` at build time.

IV.1.2. Confidence in importance: very high.

I.2. Nonissues

During the audit process research was conducted to rule out the following attack vectors.

1. **Reentrancy**
2. **Flashloans** (modulo IV.4.1)
3. **Trade-based price manipulation**

I.3. Insufficient literature

As of this writing, the jury is still out on the following considerations or attack vectors. Research opportunities are detailed in V.2.

1. **Transaction-ordering dependence**: waiting on publications or code from IOHK to determine network fee resolution and their impact on miner incentives.
2. **Cost semantics and gas**: waiting on publications or code from IOHK to determine network fee resolution.
3. **Action-based price manipulation**

II. Preamble

The audit is a preliminary effort to compensate for the fact that proper formal verification before launch is infeasible.

An audit means many things. Let's be precise about what we mean by audit in this document.

Definition II.0.1 (Audit). *An audit is a document provided to the community to guide them in taking informed risk.*

Definition II.0.2 (Community). *A community consists of liquidity providers, investors, swappers, arbitrageurs, governance token holders, and neighboring members/projects of the ecosystem.*

II.0.1. Desiderata

- **The community is the audience. The community is the customer.**
- The audit is **as much as was possible to do before launch time, not exhaustive**.

II.1. Considerations

In this chapter we look at broad concepts and decisions and provide context into the way the team is thinking about them. This section should add indirect value to the process of taking informed risks.

II.2. Attacks

In this chapter we profile threat models, attack vectors, vulnerabilities; mostly on the economic and mechanism design levels, but occasionally on the software implementation level.

III. Considerations

III.1. Code quality

In internal documents, the Ardana team set engineering standards.

1. **Building with `nix`**. Properties of *immutable, reproducible* builds are desirable and use of `nix`[1] for Cardano dapps is standardized across the ecosystem.

2. **Property tests**. These come in the categories domain-driven, tests of parser components, test of state machine components, integration property tests of database components, and also come alongside unit tests. Additionally, developers are made aware of coverage via continuous integration (CI).

3. **Linting and code style**. Enforced via CI.

4. GitHub practices of **code review and successful CI checks** for all merges into main, protected branches.

5. No bottom type allowed.

6. `newtype` constructor-destructor pairs rather than aliases and rather than passing around types like `String`, `Bool`, etc.

III.1.1. Frontend

The frontend has a nice property that the javascript ecosystem has, unlike the haskell ecosystem, `npm audit` which queries a vulnerability database for everything in the build specification. This is a tool that the Ardana frontend team leverages in development, and that which will monitor and fix our build online through continuous delivery. The astute reader will see (Thomson 2021) for information about the vulnerability database `npm audit` relies on.

[1] https://nixos.org/

III.2. Physical and operational security

A dapp consists of onchain and offchain components. An ecosystem like Cardano's confidence in security properties, by-construction or otherwise, of it's underlying decentralized technology (sometimes called "layer one") is fundamental, so we do not discuss it here. We can point to an entrypoint to literature on that[2] and for the advanced reader (Kiayias et al. 2017).

Dapp developers are responsible for securing offchain dapp components. Ardana's CTO composed a treatment of the team's security considerations in (Shapira 2021b). In what follows, we assume the reader has a minimal understanding of the plutus application backend (PAB)[3]

The highlights are simple:

1. No VM, no VM-to-VM attack. Ardana PABs run on bare metal.
2. Keys are both hosted and generated in the Yubi Hardware Security Module 2[4] (HSM). Keys on HSM cannot be read off device.
3. Developers who need to deploy are provisioned Yubikey 5[5], a physical authentication instrument without which Ardana deployments are blocked.
4. Bare metal is located in a Flexential data center, which is thoroughly audited and certified for compliance in numerous sets of industry standards[6][7].
5. Cloudfare's DDOS protection[8].
6. VPN: A Tailscale[9] implementation of Wireguard[10].
7. Firewall
8. Port knock sequence is an additional layer of access verification.

[2] https://why.cardano.org/en/introduction/
[3] https://github.com/input-output-hk/Alonzo-testnet/blob/main/explainers/PAB-explainer.md
[4] https://www.yubico.com/product/yubihsm-2/
[5] https://www.yubico.com/products/yubikey-5-overview/
[6] https://www.flexential.com/system/files/file/2021-03/centennial-flexential-data-center-data-sheet.pdf
[7] https://www.flexential.com/compliance-certifications-and-attestations
[8] https://www.cloudflare.com/ddos/
[9] https://tailscale.com/
[10] https://www.wireguard.com/

III.2.1. Protect yourself

Of course, it is up to individuals in Community to have strong digital security like password hygiene etc. See (Shapira 2021a) for a reasonably complete treatment.

III.3. Wallet integration

Ardana is partnering with Emurgo[11] such that the Yoroi[12] wallet is the means by which users interact with `Danaswap`, and a user's balance of Ardana stablecoins can be found in their Yoroi wallet.

According to a video published November 14th, 2021[13], Project Ardana recommends using Yoroi as a browser extension in Brave browser (which is derived from Chromium). As of the time of current document, Yoroi offers browser extensions in Chrome, Edge, and Firefox as well as smartphone apps for Android and iPhone.

Yoroi's security assurances[14] imply that they take security seriously. Of note is private key storage: encrypted on user's machine, never on third-party servers nor even shared with Yoroi.

III.3.1. Yoroi audit

Yoroi claims to have been rigorously audited.

> Yoroi is a light wallet for Cardano. It's simple, fast and secure. Yoroi is an Emurgo product, engineered by IOHK. And it follows best practices for software in the industry including a comprehensive security audit.

However, it seems artefacts (such as reports) from this audit are private, as of the time of current document.

III.3.2. General security of working with browser extensions

The advanced and paranoid reader may see (Obimbo, Zhou, and Nguyen 2018) to further scrutinize Yoroi. Additionally, assurances in III.1.1 should also contribute to

[11] https://emurgo.io/
[12] https://yoroi-wallet.com/#/
[13] https://youtu.be/j9wvmi0HGu0
[14] https://yoroi-wallet.com/#/faq/4

your assessment that the end-to-end user experience is secure, because the interface between Ardana's website and Yoroi will be continuously monitored modulo the `npm audit` database. Additionally, we recommend the reader see (Shapira 2021a) and implement at least some of it's advice before working with wallets.

III.4. Datastream integrity

The `dUSD` stablecoin will

NOTES from Tuesday call:

III.5. Scalar types

III.5.1. `Danaswap`

The Ardana team uses `PlutusTx.Rational`[15] to represent numbers in the `Danaswap` contract. As of this writing, tolerance (number of decimals needed to evaluate equality) is set to 20.

Belief III.5.1 (FLOPs incompatible with 'Danaswap' requirements)**.** *The 'Danaswap' smart contract requires 1. extreme precision, 2. very large numbers, and 3. exact reproduction across varying hardware. Floating point operations (FLOPs) are incompatible with the requirements.*

Addtionally, the `Danaswap` team does not *a priori* know exactly how big the numbers need to be. Additionally, the `Floating` typeclass is not provided by `plutus`, meaning if `Double` was used somewhere in the `Danaswap` codebase, it would have to be cast to `PlutusTx.Rational` at some point anyway.

III.5.2. The stablecoin

The stablecoin codebase also uses `PlutusTx.Rational`, for ultimately similar reasons.

[15] https://github.com/input-output-hk/plutus/blob/master/plutus-tx/src/PlutusTx/Ratio.hs

III.5.3. `DanaswapStats`

`DanaswapStats` uses vanilla haskell's `Double` type to log data about activities on `Danaswap`.

III.6. Rootfinding

`Danaswap` takes a novel rootfinding strategy of the team's own devising. An in depth treatment will be in a future publication. In the current document's scope is a brief registration of of Ardana's belief that invariant-driven prices will behave desirably, which is to say the implementation will provide behavior that the formal model predicts.

Recall the **invariant equation** from the StableSwap Whitepaper (Egorov 2019, 5). In the formalism provided by our Danaswap Whitepaper (Thomas 2021a, 3), there exists a function $I : S \to \mathbb{R}$ for contract states S such that $I(s) = 0$ is equivalent to the invariant equation. Danaswap borrows everything from StableSwap to vary between constant-product and constant-sum market-making according to a *leverage* parameter, for which we also accept the suggestion found in (Egorov 2019). We either hold all balances constant to solve for a number D, which has the semantics of total amount of coins *when* all coins have equal price, or we hold D and all but a given balance constant and solve for the given balance. Since this looks like solving an equation of some function set to zero, we call this "finding the roots" of a function or "solving for the function's zeros."

Belief III.6.1 (Exactly one positive real root of invariant functions)**.** *Given $n + 1$ invariant functions, one for each of n balances plus one for the constant D, each invariant function has exactly one positive real root.*

In the upcoming publication, we will leverage the contents of a precalculus course in the US to justify this belief, include a derivation from (Egorov 2019) to monic polynomials, and discuss Ardana's rootfinding algorithm with regard to other rootfinding methods in the literature or related ecosystems.

To be clear, if we did not have III.6.1, the invariant equation would either be ambiguous (a subjective choice of *which* positive real root would be needed), or provide negative or complex balances making it effectively undefined.

While Curve uses Newton's method[16] to get zeros of the invariant equation, `Danaswap` uses a home rolled algorithm that leverages III.6.1 to provide arbitrary precision one hundred percent of the time.

[16] `https://en.wikipedia.org/wiki/Newton%27s_method`

III.7. Throughput

TODO: establish throughput problem with language from (Thomas 2021b).

Definition III.7.1 (Fairness)**.** *A DeX's concurrency solution is **fair** if when two people perform an action at the same time, that action is performed for the same price.*

> The Cardano mempool is designed to be "fair." Transactions are processed in a FIFO order regardless or how much in fees they pay (the ledger spec does support a fee market, but cardano-node doesn't take this into account) (Guillemont 2021)

TODO: language to discuss this quote.

Definition III.7.2 (Fragmented)**.** *A UTXO state model is **fragmented** when there is more than one state UTXO in play at a time.*

Definition III.7.3 (Normalized)**.** *A UTXO state model is **normalized** when there is neither disagreement nor redundancy regarding the data stored by the collection of UTXOs.*

Our UTXO state model design is *fragmented* yet *normalized.* In such a model, a fairness guarantee is impossible: to have a fairness guarantee, each pool would have to update eachother, which isn't possible to do in one transaction.

IV. Attacks

IV.1. Denial-of-service

Definition IV.1.1 (Denial-of-service (DoS)). *A **denial-of-service** or **DoS** attack is a class of disruption that prevents intended users from reaching a service, usually accomplished by flooding or congesting.*

Belief IV.1.1 (No unique DoS). *Ardana ecosystem components do not offer a **unique** Denial-of-service (DoS) vector.*

However, we think Community ought to be made aware of *ambient* vulnerabilities in the broader Plutus and Cardano ecosystem.

IV.1.1. Onchain

We rely on (MLabs 2021) to describe three flavors of onchain DoS vector, which essentially target `Validators` or `Redeemers`.

Definition IV.1.2 (Token dust attack). *An attacker crams hundreds of unique tokens with different `CurrencySymbols`/`TokenNames` into a single UTXO intending for it's representation to challenge the `16kb` limit. Then, the UTXO is placed in a `Validator` in such a way that one or more `Redeemers` will need to consume it, blocking transactions on that `Validator`-`Redeemer` pair.*

Definition IV.1.3 (Datum too big). *In the datum field, an attacker puts an unbounded data structure on a UTXO that happens to demand consumption by a `Redeemer` which is critical to honest users.*

Definition IV.1.4 (EUTXO concurrency DoS). *An attacker submits a barrage of vacuous transactions consuming blocking EUTXOs.*

(MLabs 2021) points to (IOHK 2020) section on `Min-Ada-Value` as a mechanism that can be leveraged to block Token dust attack, but it's on the developer to set it and its implementation effects honest users.

> Every output created by a transaction must include a minimum amount of `ADA`, which is calculated based on the size of the output (that is, the number of different token types in it, and the lengths of their names). (IOHK 2020).

With similar drawbacks, fees or discincentives could block EUTXO concurrency DoS, where again honest users are impacted by the mechanism.

Neither ourselves nor (MLabs 2021) provide a strategy against Datum too big.

IV.1.2. Offchain

As of this writing, `plutus` depends on the JSON parsing and encoding library `aeson`[1]. This means that `PAB` artefacts, if the `aeson` version is `< 2.0.1.0`, will be subject to the known DoS vulnerability described in (Kerckhove 2021).

IV.1.2.1. Recommendation

Build system should enforce `aeson >= 2.0.1.0`.

IV.1.3. Conclusion

Denial-of-service (DoS) vectors are currently a part of Cardano. With respect to these vectors, we do not believe Danaswap nor anything in the Ardana ecosystem is better or worse off (IV.1.1). If the build system enforces `aeson >= 2.0.1.0`, a known attack is factored out.

IV.2. Price manipulation

Definition IV.2.1 (Truthfulness of a protocol's beliefs)**.** *A protocol* Π *beliefs are* ***true*** *when it's prices are accurate.*

[1] https://hackage.haskell.org/package/aeson

Definition IV.2.2 (Ideal exchange protocol)**.** *An **ideal** exchange protocol is one who's beliefs are true.*

Definition IV.2.3 (Arbitrage)**.** *When an agent can distinguish any given exchange protocol from an ideal protocol, they can profit by buying underpriced assets and selling overpriced assets. Such an agent is called an **arbitrageur**; we say that they are **doing arbitrage** or **exploiting** an arbitrage opportunity.*

Definition IV.2.4 (Price manipulation)**.** *Given an exchange protocol Π, an attacker **manipulates prices** when they make Π's beliefs less true.*

Price manipulation can be thought of as the **imposition** of arbitrage opportunities. For many protocols, this is a serious attack vector (see the bZx[2] margin trade feature for an example [3]).

IV.2.1. Trade-base manipulation

IV.2.4 roughly corresponds to "market manipulation" found in (Pirrong 2017), which further defines a taxonomy of which here we consider the trade-based flavor. In trade-based manipulation,

> The manipulator buys or sells in quantity, knowing that due to asymmetric information and trade processing and inventory costs prices will move in the direction of their trades (Pirrong 2017, p 5).

Trade-based manipulation in defi is most famously carried out with the help of Flashloans , and as we claim in IV.4.1 this is not a vector of attack that concerns us. Another form trade-based manipulation could, in principle, take is if exchange Π was already infected with false beliefs an agent A could swap a high volume of asset $\$_i$ into asset $\$_j$ at Π then go over to exchange Θ with a high volume of asset $\$_j$ swapping it for $\$_i$, skewing Θ's beliefs about relative supply and demand of $\$_i$ and $\$_j$. It seems like A would net a profit by playing Π and Θ against eachother in this way. However, we will argue that this is not a concern for the `Danaswap` exchange.

Belief IV.2.1 (No rational trade-based price manipulation)**.** *Trade-base price manipulation on `Danaswap` costs more than it's worth*

The reasons for IV.2.1 are twofold:

[2] https://bzx.network/
[3] https://blog.peckshield.com/2020/02/15/bZx/

IV.2.1.1. Invariant-driven beliefs

The prices of assets in `Danaswap` are driven by **the invariant equation** a la (Egorov 2019), seeing also (Thomas 2021a) for details. If "value" V is a reasonably well-behaved (i.e. something like "continuous") map from the set of assets to $\mathbb{R}$, then a price manipulation attack would be some way of siphoning out $\Sigma_{i=1}^{n} V(\$_i)$ into the pocket of an attacker. It is the case that, due to transition system semantics and the invariant equation, any starting state $(x : \$_i, y : \$_j)$ **must** by construction transition to $(x - \delta : \$_i, y + \delta : \$_j)$ under the suitable swapping transaction; i.e. *must not* transition to $(x - \delta : \$_i, y : \$_j)$ with the amount δ of $\$_i$ deposited into the pocket of the attacker, by construction.

There are two intuition pumps you can use to sympathize with this argument.

1. There is a sort of *conservation law* point of view, the statement $I(x) = 0$ from (Thomas 2021a) can be interpreted as saying "balances of assets are conserved" across the exchange.

2. We observe an absence of prince manipulation scandals in Curve, the exchange based on (Egorov 2019).

IV.2.1.2. Incentive alignment

Belief IV.2.2 (Arbitrage makes belief more true)**.** *Arbitrageurs make* `Danaswap`*'s beliefs more true.*

In light of rudimentary definitions of arbitrage (Wikipedia[4], Investopedia[5]), this is trivially equivalent to believing that `Danaswap` forms a market at all. Consequently, if some agent puts pressure on making `Danaswap`'s beliefs less true, the community of arbitrageurs will step in and apply counterpressure because it is in their interest to do so, following the definition of arbitrage.

IV.2.2. Information-based manipulation

Another flavor in (Pirrong 2017)'s taxonomy is information-based manipulation. In information-based manipulation, the manipulator leverages **disinformation** to knock prices in a direction favorable to them (Pirrong 2017, 4). This attack vector is rather broad, we provide just two scenarios but one can readily imagine more.

[4] https://en.wikipedia.org/wiki/Arbitrage#Price_convergence
[5] https://www.investopedia.com/terms/a/arbitrage.asp

IV.2.2.1. Scenario: a false partnership

Alice has a large `DANA` position. She compromises the account of a discord mod of a non-Ardana ecosystem product Π and announces a new partnership between that product and Ardana. Since the false announcement came directly from a Π discord mod, the Π community believes it is true and there is a frenzy for `DANA`, driving up the price.

IV.2.2.1.1. A false partnership: branch one Influx of Π community investors changes the balance of an upcoming governance decision.

IV.2.2.1.2. A false partnership: branch two When the hack is discovered it's a PR problem for Ardana. A redditor speculates that Alice is an Ardana insider, gaining a significant portion of the subreddit's total upvote volume that week.

IV.2.2.2. Scenario: shorting

At some point, there exists a protocol Φ that empowers speculators of Cardano to take short positions on Cardano-native tokens, perhaps this taking the form of contracts that pay out if a particular asset isn't trading for a forecasted price on some exchange. Bob gains a large short position against `DANA`. If Bob's short position isn't looking good, Bob might reach out to a journalist and fabricate a story about being a former Ardana employee who is now whistleblowing about something. The story is published, and people pull out of `DANA`, driving the price down.

IV.2.3. Conclusion

`Danaswap`, Ardana stablecoins, and any mechanism relating to the `DANA` governance token each seem more resilient to price manipulation than other defi projects. Trade-based price manipulation is a nonissue. There exist ecosystem activities that remain a plausible opening to information-based price manipulation, where their plausibility is a function of personal security norms (see (Shapira 2021a)) and integrity norms not simply of the Ardana Community but of the ambient defi space, especially the Cardano defi space.

IV.3. Vampire attack

Definition IV.3.1 (Vampire attack)**.** *Let Π and Θ be similar protocols, but Π launched and attracted investors and customers earlier, and Θ is somehow derivative of Π. Suppose Θ competes with Π such that Θ makes parameter choices or other measures to become more attractive to investors or customers than Π. A* ***vampire attack*** *is defined as the migration of value (liquidity or other assets) out of Π into Θ.*

IV.3.1. The literature

Consult a selection of stories about vampire attacks.

- $\Theta =$ `SushiSwap`; $\Pi =$ `UniSwap`[6]. SushiSwap was in fact a fork of UniSwap's code, and they provided incentives that directly targeted UniSwap investors and liquidity providers. This is the canonical notion of a vampire attack, with what appears to be the most written about it because of it's scale of impact and how early on the defi scene it was found. Our present definition is generalized for analysis that applies outside of the specific conditions here.

- $\Theta =$ `Swerve`; $\Pi =$ `Curve`[7]. The term "vampire" does not occur in this article, but blaize.tech[8] considers it to be a vampire attack. By forking Curve, Swerve offered a platform very similar to Curve's, and became competitive in total value locked (TVL) in a matter of days while people pulled out of Curve. There doesn't appear to be anything unique about Curve and Swerve being stablecoin exchange protocols.

- $\Theta =$ `Artion`; $\Pi =$ `Opensea`[9]. At current writing it's too early to tell, but it's possible that Artion by providing a platform competitive with Opensea will be considered to have vampire attacked it. Unfolding events for this to be the case would have to be that Artion is successful at the expense of Opensea. My choice to be influenced by a CoinDesk writer's choice to call this a vampire attack is up for debate, but my intention is to be consistent with the ecosystem and the literature and I don't see grounds to exclude this writer from either.

- See extended notes on forks in (Lee 2020).

IV.3.1.1. Major takeaways

- Lack of vampire attack stories in the Cardano ecosystem is, according to my analysis, not a by-construction property of Cardano. I.e. it's a matter of time.

[6] https://youtu.be/UFjXwrCGuog
[7] https://finance.yahoo.com/news/swerve-finance-total-value-locked-075020390.html
[8] https://blaize.tech/services/how-to-prevent-liquidity-vampire-attacks-in-defi/
[9] https://www.coindesk.com/tech/2021/09/24/andre-cronjes-new-nft-marketplace-is-a-vampire-attack-su

- Forking a codebase is often used as evidence in favor of the accusation that a given Θ conducted a vampire attack, though forking is not an intrinsic property of the attack.

IV.3.2. Scenario: reputational damage to Ardana if it is considered Θ

If you imagine a competing exchange protocol beat Ardana to market, then in principle Ardana could be (weakly) accused of vampire attacking them.

Imagine if a bunch of Curve investors pull out their liquidity, exchange it for `ADA` on Coinbase, and start playing `Danaswap`. Would Curve think of that like a vampire attack? The literature has not seen a vampire attack across a distance as great as that between Ethereum and Cardano, but that's only because it is early.

If the literature or ecosystem chooses to view Ardana as a vampire attacker, the project could suffer reputational damage. However, even in this event, it needs to be shown that the public relations problem is actually significant. I.e., is Θ's public relations challenges impacted severely by a vampire attack accusation? It might warrant further research, but we do not conduct that research in the current document.

IV.3.3. Scenario: value siphoned out if we become Π

Suppose another exchange protocol for stablecoins launches with an incentive structure more attractive to our Community than our own. Then, everyone could choose to migrate to this other protocol. According to the literature, we would be justified in considering this a vampire attack.

Suppose further that, having open sourced the `Danaswap` repo, this competitor's product is a fork of our own, making supplement components for any aspects that were closed source. If we follow the literature, we would be even more justified in considering this a vampire attack.

IV.3.3.1. Mitigations

- **Keeping the `Danaswap` code closed source**. This is a minor payout in risk reduction. We can also make a custom license, make it source available but proprietary, etc.

- **Peg fee parameters to democracy via `DANA` governance token holders**. The Community's preferences are a part of the competitive landscape; if a derivative competitor is closer to our Community's wants and needs than we are, then the Ardana Community will be siphoned out of Ardana. An automatic mechanism to decrease this possibility would look simply like setting policies such as the fee structure with vote inputs from the Community, however, automation shouldn't be the last word; attention and care will have to be paid to make sure people are actually using the mechanism. We see this having a stronger payout in risk reduction.

IV.3.4. Conclusion

In any kind of market, participants take on the unavoidable risk of competitors showing up with better rates. The factor of code forking presented by the open source software context doesn't change this much, and the factor of fee structure parameters presented by the cryptoeconomic context doesn't either.

Vampire attack is a loose mirage of competitive phenomena: ultimately judged by the ecosystem literature, often coming down to individual journalists or researchers. It is in principle possible to be accidentally accused of vampire attacking. We do not want value siphoned out: there exist some practices to decrease this possibility that mostly amount to community engagement and giving the community a real voice in the system.

IV.4. Flashloans

Flashloans are associated with something like `$136M`[10] in[11] losses[12].

Definition IV.4.1 (Flashloan attack)**.** *Let a **flashloan** be some multi-step transaction that begins with an uncollateralized loan and ends with repayment of that loan, with arbitrary logic in between. Then, a **flashloan attack** is some method of manipulating prices during such a transaction and profiting.*

Ethereum offers flash loans because they have **multi-step atomic transactions**. There is no such mechanism in Cardano.

Belief IV.4.1 (No flashloans)**.** *Flashloans will not be entering the Cardano ecosystem.*

[10] https://peckshield.medium.com/bzx-hack-full-disclosure-with-detailed-profit-analysis-e6b1fa9b18fc
[11] https://news.bitcoin.com/defi-protocol-harvest-finance-hacked-for-24-million-attacker-returns-2-5
[12] https://www.coindesk.com/markets/2021/05/20/flash-loan-attack-causes-defi-token-bunny-to-crash-ov

As such, ‘Danaswap,’ Ardana stablecoins, and mechanisms related to `DANA` governance tokens are not vulnerable to flashloan attacks.

IV.4.1. Action: monitoring Cardano for developments in multistep atomic transactions

Project Ardana will be monitoring the evolution of Cardano, because we believe that if multistep atomic transactions are introduced flashloans will be shortly around the corner.

IV.4.1.1. In this event, the following mitigation strategy sketches will become urgent

- Onchain code only allow interop from one platform and users, not arbitrary platform.
- Lending products ought to require to collateralize in one whole transaction ahead of time before.
- Block price manipulation by disallowing mid-transaction information from updating prices.

IV.5. Reentrancy

Definition IV.5.1 (Reentrant). *A procedure is* ***reentrant*** *if it can be initiated while it's already running or a prior initiation has been interrupted and both runs can terminate, failing to raise an error.*

The infamous “DAO Hack” of 2016 occurred because Solidity allows the programmer to write reentrant smart contracts (Ma et al. 2019, 59–63).

Belief IV.5.1 (No reentrancy). *Plutus does not afford the freedom to write reentrant contracts.*

We can make a blanket statement that smart contracts in Cardano are invulnerable by construction to reentrancy. This is true because no transaction can be validated by (and it follows can require validation from) two different contracts. If you imagine Alice writes contract A and invokes it (executing the program $\texttt{Alice}_A$) to validate transaction T, then Bob invokes A ($\texttt{Bob}_A$) before Alice's invocation terminates, T will be validated by **at most** one of $\texttt{Alice}_A$, $\texttt{Bob}_A$.

As such, reentrancy attacks are not a threat to `Danaswap`, Ardana stablecoins, or any mechanism relating to the `DANA` governance token.

IV.6. A whale buys a high volume of `DANA` when the float is exhausted, forcing the protocol to buy it back at an arbitrary price

This subchapter was cowritten with Bassam Awad

An attack vector was internally raised and it led to some changes (IV.6.1.1) in the management of the protocol. For the current document to discuss it, we will provide some detail about Ardana's **pegging strategy** and consequences of `DANA`'s **deflationary** disposition.

Definition IV.6.1 (Pegging strategy)**.** *A mechanism that monitors and somehow "controls" or guides the price of an asset is a* ***pegging strategy*** *when it's intent is to keep the asset's price within a tight ϵ of some (stable) reference currency.*

Definition IV.6.2 (Float)**.** *We will call a pegging strategy that consists of maintaining a sum of assets (such as `DANA`) intended for buying and selling at time and volume such that a targeted stablecoin asset's (say `dUSD`'s) price can be guided to within ϵ of it's reference currency (say, `USD`) a* ***floating strategy****. Such a sum of assets is called a* ***float****, the term borrowed from in i.e. a coffee shop the sum of cash in the cash register for daily operation.*

Ardana **must** use a floating strategy because `DANA` is *deflationary*, or subject to no ad-hoc minting. This is discussed further below. Consequently, there is a question of **how much wealth should be tied up in the float**.

IV.6.1. Scenario: `DANA`'s price gets really high and the float is low at the same time

Consider the case when the float consists strictly of a `DANA` balance. Recall that the stablecoin protocol is **obliged** by the logic of it's smart contract to buy and sell stablecoins (like `dUSD`). We will see that **if the float consists only of `DANA`**, then the ability to do this is massively impacted by the price of `DANA`.

1. Balance of the float dips under some threshold θ.
2. Some **whale** (agent with lots of capital) buys up a massive `DANA` position.
3. To fulfill the pegging strategy, the Ardana stablecoin protocol is obliged to keep it's float above θ.
4. The whale can charge the protocol whatever it likes for `DANA`.

This scenario is not just a problem for the protocol, but a Community problem because **there's no guarantee that the protocol is liquid enough to fulfill it's obligations**. Thus, this scenario weakens the very notion of what a stablecoin is: an asset who's value is pegged to a reference asset.

IV.6.1.1. Mitigation: diversify

Because of the above exploit, Ardana has decided to diversify the float, introducing `ADA` to it. An **administrator** is now required to manually decide what asset the floating strategy operates in from time to time. Additionally, the float will be maintained as a **war chest**, and act as an investment fund during peacetime. Furthermore, debt auctions (see ("Ardana" 2021)) will be allowed to use `ADA` instead of `DANA`.

IV.6.2. Analysis

IV.6.2.1. Other protocols

One approach to this problem is asking the reasonable question *why doesn't MakerDAO (MakerDAO 2015) have this problem*? However, the answer (MakerDAO's pegging strategy is to mint and burn `MKR` as needed whereas `DANA` supply is fixed/deflationary) is too straightforward to be of any help.

Curve's governance token `CRV` is on an inflation schedule, so while it's not deflationary like `DANA` it's not ad-hoc like `MKR`. In the case of this attack vector, we think `CRV` is closer to `DANA` than to `MKR`, so we are obliged to consider why hasn't Curve been subject to this attack? While an analysis of Curve's pegging strategy is out of scope for the current document, we weakly believe that this is something that simply hasn't happened *yet* but there's no in-principle reason why Curve is resilient to this attack.

IV.6.2.2. Best we can do

Belief IV.6.1 (No float exploit mitigation strategy better than centralization)**.** *Obliging an administrative war chest to secure the protocol by diversifying the float is the best idea we have to mitigate the above scenario.*

IV.6.2.3. Centralization

The authorship of the current document is assured that it is Ardana's intentions to decentralize this mechanism as quickly as possible after launch.

V. Postamble

V.1. Toward formal verification

Formal verification (FV) was considered too time-intensive to do before launch. The authorship of the current document was seen as a sort of warmup for FV. Now that we're launched, we will begin a formal verification stage of the project.

`Danaswap` validators are the first priority for formal verification.

Other properties we'd like to prove

1. **Non-indebted pools are never liquidated**.
2. **"No money for nothing"**: no one can arbitrarily withdraw assets from the protocol without depositing something else or paying some fee.
3. **Modular resilience**. In the language of (Genovese 2018), **modular risk** of a composite contract is risk that is greater than the sum of the risks of the individual lego blocks. We would like not just for the Ardana project to be **compositional** (i.e. the sum risk is *no more* than the sum of the risks of the individual lego blocks), but for it to be compositional with respect to actors that may interact with it, arbitrarily.
4. Proofs of results from Idealized `Danaswap` vs. an attacker with infinite money .

V.2. Future work

Numerous research opportunities are out of scope for the current document.

V.2.1. What is the general delta in incentives off ethereum via no-multistep-atomic-transactions?

As mentioned in IV.4.1, there are no multi-step atomic transactions in Cardano.

V.2.2. Compare/contrast transaction-ordering dependence risks across ethereum to cardano and other blockchains.

For the current document, we lack any confidence level regarding the frontrunning question because we are waiting on publications or code from IOHK to determine network fee resolution. A comment in (Guillemont 2021) suggests that miner-type frontrunning will not be an issue, but our confidence is not high in everything shaking out that way.

V.2.3. Liquidity arbitrage

Liquidity arbitrage is when an arbitrageur attempts to play `Danaswap` pools against eachother. Liquidity tokens may be priced one way in subpool P_i and priced another way in P_j. As an intuition, we can block this by making a different type of liquidity token per subpool; but we don't want to do this for user experience reasons (a zoo of liquidity tokens may be prohibitively difficult to reason about for all but a select few power users).

There is also the extent to which liquidity arbitrage arises due to parallelization, with synchronization questions across pools complexifying the problem.

Further research pressure ought to be applied to see if liquidity arbitrage aligns with the function of arbitrage in general (IV.2.2) or forms a kind of threat.

V.2.4. Idealized `Danaswap` vs. an attacker with infinite money

We take the limit case of an idealized `Danaswap` which has arbitrary facility to add subpools and consider how an attacker with infinite money would approach making `Danaswap`'s beliefs less true. This thought experiment could provide confidence in beliefs along the lines of IV.2, and instruct us to for instance impose minimum trade requirements or implement automatic addition and subtraction of pools such that the protocol becomes more resilient to "economic DoS."

V.2.5. Resilience of `Danaswap` to action-based Price manipulation

In IV.2 we considered trade-based manipulation and briefly mentioned centralization's opening to information-based manipulation, but (Pirrong 2017) also defines action-based manipulation. As Ardana is one component in a chaotic system consisting of many different kinds of agents and assets both onchain and offchain, there is *some* degree to which action-based (like when you short-sell your own stock, close down your physical capital centers, cover your short position, then reopen the physical capital centers) manipulation could play a role.

V.2.6. Understand IV.6.1.1 better to come up with a *minimal* intervention

Ardana's intervention to mitigate the problem of a whale sitting on `DANA` is a little heavy handed. While it is in scope for the post-launch evolution of the platform to work on decentralizing the mitigation is installed at launch, it also may warrant application of research pressure to come up with different solutions.

VI. Bibliography

10 "Ardana." 2021. WEBSITE IS IN FLUX / INSERT URL HERE WHEN IT COOLS DOWN[1].

Egorov, Michael. 2019. "StableSwap - Efficient Mechanism for Stablecoin Liquidity." https://curve.fi/files/stableswap-paper.pdf.

Genovese, Fabrizio Romano. 2018. "Modularity Vs Compositionality: A History of Misunderstandings." https://blog.statebox.org/modularity-vs-compositionality-a-history-of-

Guillemont, Sebastien. 2021. "[FR] - Increase Network Throughput." GitHub issue. 2021. https://github.com/input-output-hk/cardano-node/issues/3247.

IOHK. 2020. "FAQ: Native Tokens (Cardano's Multi-Asset Support Feature)." 2020. https://cardano-ledger.readthedocs.io/en/latest/explanations/faq.html.

Kerckhove, Tom Sydney. 2021. "JSON Vulnerability in Haskell's Aeson Library." 2021. https://cs-syd.eu/posts/2021-09-11-json-vulnerability.

Kiayias, Aggelos, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol." *CRYPTO 2017*. https://iohk.io/en/research/library/papers/ouroborosa-provably-secure-proof-of-stake-

Lee, Ian. 2020. "Fork Defense Strategies in DeFi." Bankless Newsletter. 2020. https://newsletter.banklesshq.com/p/fork-defense-strategies-in-defi.

Ma, Richard, Jan Gorzny, Edward Zulkoski, Kacper Back, and Olga V. Mack. 2019. *Fundamentals of Smart Contract Security*. Momentum Press.

MakerDAO. 2015. "The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System." https://makerdao.com/whitepaper/White%20Paper%20-The%20Maker%20Protocol_%20MakerDAO%E2%80%99s%20Multi-Collateral%20Dai%20(MCD)%20System-FINAL-%20021720.pdf.

MLabs, Team. 2021. "Common Plutus Vulnerabilities." 2021. https://mlabs.slab.com/public/posts/j8pjrj5y.

[1] https://WEBSITE%20IS%20IN%20FLUX%20/%20INSERT%20URL%20HERE%20WHEN%20IT%20COOLS%20DOWN

Obimbo, Charlie, Yong Zhou, and Randy Nguyen. 2018. "Analysis of Vulnerabilities of Web Browser Extensions." In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 116–19. https://doi.org/10.1109/CSCI46756.2018.00029.

Pirrong, Craig. 2017. "The Economics of Commodity Market Manipulation: A Survey." https://www.bauer.uh.edu/spirrong/manipulation_review_JCM_Pirrong.pdf.

Shapira, Isaac. 2021a. "Ensuring Your Personal Digital Security." *Journal.Platonic.Systems*. https://journal.platonic.systems/p/ensuring-your-personal-digital-

———. 2021b. "Securing Ardana Swap." INTERNALLY CIRCULATING / ADD HYPERLINK HERE WHEN PUBLISHED[2].

Thomas, Morgan. 2021a. "Danaswap: A Scalable Decentralized Exchange for the Cardano Blockchain." INTERNALLY CIRCULATING / ADD HYPERLINK HERE WHEN PUBLISHED[3].

———. 2021b. "Transaction Throughput Scalability Strategies for Plutus Smart Contracts." *Journal.Platonic.Systems*. https://platonic.systems/papers/utxo-models-public.pdf.

Thomson, Edward. 2021. "GitHub Advisory Database Now Powers Npm Audit." 2021. https://github.blog/2021-10-07-github-advisory-database-now-powers-npm-audit/.

[2] https://INTERNALLY%20CIRCULATING%20/%20ADD%20HYPERLINK%20HERE%20WHEN%20PUBLISHED
[3] https://INTERNALLY%20CIRCULATING%20/%20ADD%20HYPERLINK%20HERE%20WHEN%20PUBLISHED