Cardano Guardrails on Governance Actions

Technical Rationale

Intersect Parameters Committee

September 27th 2024

Table of contents

Revision history

Version	Date	Change
0.01	2024-07-09	Created document; drafted rationale for general parameters
0.02	2024-07-11	Added rationale for terminology and transaction fees; added initial rationale for some economic parameters
0.03	2024-07-12	Added rationale for additional economic parameters
0.04	2024-07-15	Completed rationale for economic parameters; added rationale for governance parameters; added rationale for treasury withdrawals, constitution and committee actions; added monitoring and reversion of changes
0.05	2024-07-16	Added initial rationale for some network parameters
0.06	2024-07-17	Added rationale for additional network parameters; provided rationale for the interim period; started glossary
0.07	2024-07-18	Added initial rationale for some technical/security parameters
0.08	2024-07-19	Added introduction and summary; completed rationale for technical/security parameters; added discussion on parameter groups; completed glossary.
0.09	2024-07-23	Checked against constitution, fixed issues, first draft of hard fork guardrails.
0.10	2024-07-24	Checked CDDL/ledger rules and added Appendix B, Added changes section to general parameter guardrails. Checked guardrails against constitution. Added list of protocol parameter groups (Appendix C).
0.11	2024-08-05	Read through, fix various issues prior to technical review; added Appendix D. Started to share for first review.
0.12	2024-09-09	Review and fix small changes. Shared for wider review.
0.13	2024-09-17	Added Appendix E plus links to appendices.
0.14	2024-09-27	Incorporated feedback from the IOG Formal Methods Group.

1. Introduction

As part of the interim Cardano constitution, a number of guardrails have been defined on Cardano governance actions. The guardrails restrict governance actions, helping to avoid undesirable outcomes and assisting voters to decide whether a proposed action complies with the intentions of the Cardano constitution. These include guardrails on all of Cardano's updatable protocol parameters, as well as on hard forks, treasury withdrawals, constitutional changes, and the interim period during which the governance system will be bootstrapped. Wherever possible, these guardrails have been automated in the form of an on-chain script that prevents non-compliant governance actions from being submitted prior to voting, the "guardrails script". This provides a simple form of programmable governance. In other cases, for example where conditions are less clear cut or require off-chain knowledge/information, the guardrails cannot be automated, and so are subject to human interpretation. Some of these more general guardrails may be automatable in future, subject to implementation improvements, use of trusted oracles, or other techniques.

This document provides the technical rationale for each of the guardrails that are specified in the interim constitution, giving an explanation for why the guardrail has been specified, what the primary concerns are that motivated the guardrail, whether the guardrail is automatable, how and whether the guardrail could sensibly be changed in future, and if so what the implications of that change might be. It distills technical discussion by Intersect's Parameter Committee and other technical groups over a period of approximately two years leading up to the Chang hard fork.

The majority of the guardrails concern on-chain parameter changes, defining limits and ranges on how those parameters can be set. The limits have been designed conservatively so that they prevent clearly undesirable outcomes. Many of them could be altered in future if underlying technical concerns were properly met, either to relax them if new information was provided that supported a relaxation or implementation improvements were made, or to tighten them if it turned out that the ranges were too lax. The guardrails governing the Interim governance period ([INTERIM–01] etc) are, of course, temporary and will cease to apply once the final constitution is ratified.

In this rationale, text in italics is taken verbatim from the interim constitution. Normal text represents the technical rationale etc.

The guardrails and associated rationale described here apply to Cardano mainnet. Alternative guardrails are possible in other situations, e.g. testnets or alternative deployments of the Cardano node, e.g. within Hydra heads or as partner chains.

2. Contextualisation

2.1. Terminology and Guidance

Should/Should not.
Where this Appendix says that a value "should not" be set below or above some value, this means that the guardrail is a recommendation or guideline, and the specific value could be open to discussion or alteration by a suitably expert group recognized by the Cardano community in light of experience with the Cardano Blockchain governance system or the operation of the Cardano Blockchain.

Rationale:
Not all conditions can be fully automated. Human judgment and discretion may be necessary in some cases, or the conditions may be external to the blockchain. For example, it may be necessary to evaluate benchmark performance data for the blockchain, and the current governance mechanism has no way to determine this automatically.

Must/Must not.
Where this Appendix says that a value "must not" be set below or above some value, this means that the guardrail is a requirement that will be enforced by Cardano Blockchain ledger rules, types or other built-in mechanisms where possible, and that if not followed could cause a protocol failure, security breach or other undesirable outcome.

Rationale:
Some conditions would potentially result in damage to the blockchain, security violations or other undesired outcomes. These must be prevented. Not all such conditions can be detected automatically, but where it is possible to do so, then a guardrails script or builtin mechanism should be used to prevent the outcome. In cases where automation is not possible, the guardrails provide the necessary strong guidance to enable safe governance. Some conditions are absolute, but in other cases, conservative thresholds have been set. The latter might be relaxed as experience is gained with the system operation. The scope for such changes has been highlighted in this rationale.

Benchmarking.

Benchmarking refers to careful system level performance evaluation that is designed to show a-priori that, for example, 95% of blocks will be diffused across a global network of Cardano Blockchain nodes within the required 5s time interval in all cases. This may require construction of specific test workflows and execution on a large test network of Cardano nodes, simulating a global Cardano Blockchain network.

Performance analysis.

Performance analysis refers to projecting theoretical performance, empirical benchmarking or simulation results to predict actual system behavior. For example, performance results obtained from tests in a controlled test environment (such as a collection of data centers with known networking properties) may be extrapolated to inform likely performance behavior in a real Cardano Blockchain network environment.

Simulation.

Simulation refers to synthetic execution that is designed to inform performance/functionality decisions in a repeatable way. For example, the IOSim Cardano Blockchain module allows the operation of the networking stack to be simulated in a controlled and repeatable way, allowing issues to be detected before code deployment.

Performance Monitoring.

Performance monitoring involves measuring the actual behavior of the Cardano Blockchain network, for example, by using timing probes to evaluate round-trip times, or test blocks to assess overall network health. It complements benchmarking and performance analysis by providing information about actual system behavior that cannot be obtained using simulated workloads or theoretical analysis.

Rationale:
Governance decisions should be informed by evidence, and not taken arbitrarily. This is especially important where security issues could arise. Timing is an essential part of maintaining the integrity of the Ouroboros Praos consensus algorithm and the security/dependability of the Cardano mainnet.

Reverting Changes.

Where performance monitoring shows that actual network behavior following a change is inconsistent with the performance requirements for the Cardano Blockchain, then the change must be reverted to its previous state if that is possible. For example, if the block size is increased from 100KB to 120KB and 95% of blocks are no longer diffused within 5s, then a change must be made to revert the block size to 100KB. If this is not possible, then one or more alternative changes must be made that will ensure that the performance requirements are met.

Rationale:
Changes should always consider the possibility of unexpected issues, and be prepared to deal with the consequences. While technical input will be essential, this will require DReps and SPOs to collaborate to deal with problems.

Severity Levels

Issues that affect the Cardano Blockchain network are classified by severity level, where:

- Severity 1 is a critical incident or issue with very high impact to the security, performance or functionality of the Cardano Blockchain network

- Severity 2 is a major incident or issue with significant impact to the security, performance or functionality of the Cardano Blockchain network

- Severity 3 is a minor incident or issue with low impact to the security, performance or functionality of the Cardano Blockchain network

Rationale:
This is a standard and widely-used classification, designed to ensure that appropriate and measured responses are taken to on-chain incidents.

Future Performance Requirements.

Planned development such as new mechanisms for out-of-memory storage may impact block diffusion or other times. When changing parameters, it is necessary to consider these future performance requirements as well as the current operation of the Cardano Blockchain. Until development is complete, the requirements will be conservative; they may then be relaxed to account for actual timing behavior.

Rationale:

Changes may need to be backed out if the system design or implementation changes. It is better to allow for future changes that may impact performance and then relax constraints on changes as actual system performance becomes known rather than to stretch performance envelopes in the short term and then retract those. For example, block sizes or transaction sizes might need to be reduced in future if sufficient allowance is not made for out-of-memory storage costs, a solution for which is currently in implementation.

2.2. Automated Checking ("Guardrails Script")

Guardrails may overlap: in this case, the most restrictive set of guardrails will apply.

Rationale:
This ensures that all guardrails are enforced. Overlapping guardrails generally arise where the more restrictive guardrail could conceivably be relaxed or eliminated through the governance process, but the less restrictive one represents an absolute value that cannot be violated (e.g. zero or 100%).

*Where a parameter is not explicitly listed in this document, then the script **must not** permit any changes to the parameter.*

Rationale:
This ensures that the guardrails script is consistent with its handling of any omitted parameters, taking a conservative approach to omissions. It also indirectly ensures that guardrails are maintained and updated as new parameters are added. The alternative would be not to restrict any changes to unlisted parameters, that is to take a permissive approach to omitted parameters.

Conversely, where a parameter is explicitly listed in this document but no checkable guardrails are specified, the script **must not** impose any constraints on changes to the parameter.

Rationale:
This ensures that the guardrails script doesn't apply additional conditions that have not been identified in the guardrails.

2.3 Monitoring and Reversion of Parameter Changes

Monitoring Parameter Changes

All network parameter changes must be monitored carefully for no less than 2 epochs (10 days)

*- Changes **must** be reverted as soon as possible if block propagation delays exceed 4.5s for more than 5% of blocks over any 6 hour rolling window*

Rationale:
This ensures that the real-world impact of the parameter change does not impact the guarantees that are required by the Ouroboros Praos protocol for Cardano mainnet.

*All other parameter changes **should** be monitored*

*- The reversion plan **should** be implemented if the overall effect on performance, security or functionality is unacceptable.*

Rationale:
The monitoring requirement ensures that information is available to inform decisions about reverting changes to parameter changes, and to inform future parameter change proposals. The reversion requirement is stated as “should” since reversion may not be possible or desirable in all circumstances, or an alternative reversion may give better outcomes in specific circumstances.

Reversion Plans

*A specific reversion/recovery plan **must** be produced for each parameter change. This plan **must** include:*

- Which parameters need to change and in which ways in order to return to the previous state (or a similar state)
- How to recover the network in the event of disastrous failure

*This plan **should** be followed if problems are observed following the parameter change. Note that not all changes can be reverted. Additional care must be taken when making changes to these parameters.*

Rationale:
The reversion plan is available to the community and can be acted upon at short notice if necessary. SPOs, the constitutional committee and other parties that are needed for the reversion can be identified and prepared appropriately. As described above, the reversion plan may not foresee the exact situation that the blockchain is in, or the suggested reversion may be impractical or unacceptable. In this case, a revised plan that was appropriate to the actual situation would need to be formulated and acted on.

2.4 Changes to Guardrails

The guardrails are designed to be conservative, to support open governance by restricting actions that could be harmful to the blockchain. It may be necessary or desirable to make changes to these guardrails as experience is gained with governance, or as changes are made to the system or new behavioral information is obtained. The rationale indicates where and how changes can be made to each of the guardrails if this is required. Some thresholds are set axiomatically (e.g. voting thresholds requiring > 50% of the stake are necessary for security reasons, or negative/zero values might not be sensible), and so cannot be changed. Other thresholds might be changed if detailed system modeling shows that broader ranges are acceptable, or that the guardrail allows too broad a range to be set. This document gives a general rationale, but does not give a detailed report for each threshold setting.

Intersect's Parameter Committee is designated as the technical custodian of the parameter guardrails, which form part of the constitution that is overseen by the Intersect Civics Committee. Changes to guardrails can be requested by instituting a Parameter Change Request. Supporting information can be found in various places. General benchmarking reports are published on Cardano Updates, but these generally cover overall node performance. Additional benchmarking/analysis/modeling reports are commissioned and published on the Cardano Forum, as part of the regular work of Intersect's Parameter Committee. These reports will generally be included as part of responses to Parameter Change Proposals, and will gradually form a reference library that can be used to inform future changes both to the parameters and to the guardrails. Where e.g. security concerns arise, some reports may be held on file by Intersect and made available on a need-to-know basis, or abstracted for public consumption.

2.5 Limit Settings and CDDL/Implementation Restrictions

Some limit settings are enforced by types in the ledger CDDL specification (and others are enforced in the underlying ledger implementation). These limits are identified in the text, and summarized in Appendix B.

3. Overall Guardrails on Protocol Parameters

3.1 General Guardrails

Guardrails

PARAM-01 (y)

Concerns: Integrity

*Any protocol parameter that is not explicitly named in this document **must not** be changed by a Parameter update governance action*

Rationale:

By requiring at least one guardrail for each changeable parameter (which may or may not be automatically checkable by the guardrails script), this meta-guardrail ensures that no parameter is overlooked when constructing the guardrails document. It also allows a straightforward catch-all for the guardrails script: no parameter can be changed unless there is at least one guardrail specified in the constitution. Finally, it implicitly captures non-updatable protocol parameters - ones that cannot, anyway, be changed by a Parameter update governance action.

Changes:

This is a conservative guardrail. The alternative would be permissive, allowing arbitrary changes unless the parameter was provided with some guardrails.

PARAM-02 (y)

Concerns: Integrity

*Where a protocol parameter is explicitly listed in this document but no checkable guardrails are specified, the guardrails script **must not** impose any constraints on changes to the parameter. Checkable guardrails are shown by a (y)*

Rationale:

This meta-guardrail complements [PARAM-01]. It ensures that where guardrails cannot be automatically checked, the guardrails script will always allow a change action.

Changes:

No sensible change is possible to this meta-guardrail. If a checkable guardrail is not specified, no check should be made by the script.

PARAM-03 (y)

Concerns: Security

*Critical protocol parameters require an SPO vote in addition to a DRep vote: SPOs **must** say "yes" with a collective support of more than 50% of all active block production stake. This is enforced by the guardrails on the stake pool voting thresholds.*

Rationale:

This meta-guardrail is designed to ensure that changes to parameters that are critical to the blockchain operation meet the security assumptions for Ouroboros Praos. That is, such changes are always supported by a majority of the active stake. It complements [PARAM-05]. Appendix D lists the critical parameters.

Changes:

The threshold cannot be reduced below 50% without endangering the security of the blockchain. The threshold could be increased above 50% without weakening the security requirement. However, if the threshold was set too high, then effective governance might not be possible. The meta-guardrail has been built-in to all relevant voting guardrails so does not need to be considered separately unless they are changed.

PARAM-04 (x - “should”)

Concerns: Security

*At least 3 months **should** normally pass between the publication of an off-chain proposal to change a critical protocol parameter and the submission of the corresponding on-chain governance action. This guardrail may be relaxed in the event of a Severity 1 or Severity 2 network issue following careful technical discussion and evaluation.*

Rationale:

This meta-guardrail is designed to ensure that changes to parameters are properly considered and debated by the Cardano community before they are voted and enacted on-chain. The means of publication is deliberately not specified, but should be a recognised public forum that is widely used by ada holders and DReps, and that is easily accessible (e.g. it should not be behind a paywall). It may be that the community decides to use **Info** actions for this purpose, for example.

The exception for high severity network issues is to allow rapid responses that require changes to parameters if a serious event occurs on-chain. As described in the guardrails, Severity 1 is a critical incident, and Severity 2 is a major incident. This allows the right balance to be struck between community involvement and speed of reaction.

The list of critical parameters is given in Appendix D.

Changes:

It is not sensible to change this guardrail.

PARAM-05 (y)

Concerns: Security

*DReps **must** vote "yes" with a collective support of more than 50% of all active voting stake. This is enforced by the guardrails on the DRep voting thresholds.*

Rationale:

This meta-guardrail ensures that DReps have provided analogous support to that required for other on-chain changes. It complements [PARAM-03].

Changes:

The threshold cannot be reduced below 50% without endangering the security of the blockchain. The threshold could be increased above 50% without weakening the security requirement. However, if the threshold was set too high, then effective governance might not be possible. The meta-guardrail has been built-in to all relevant voting guardrails so does not need to be considered separately unless they are changed.

PARAM-06 (x - “should”)

Concerns: Security

*At least 3 months **should** normally pass between the publication of an off-chain proposal to change a parameter that is critical to the governance system and the submission of the corresponding on-chain governance action. This guardrail may be relaxed in the event of a Severity 1 or Severity 2 network issue following careful technical discussion and evaluation*.

Rationale:

The rationale is the same as for [PARAM-04], but for governance parameters rather than critical system parameters. It is less likely, though still possible, that a Severity 1 or Severity 2 incident could require some critical governance parameter to be changed in order to mitigate the issue.

The list of critical governance parameters is given in Appendix D.

Changes:

It is not sensible to change this guardrail.

3.1 Non-Updatable Protocol Parameters

Some fundamental protocol parameters cannot be changed by the Protocol Parameter Update governance action. These parameters can only be changed in a new Genesis file as part of a hard fork. It is not necessary to provide specific guardrails on updating these parameters.

Rationale:

The document does not set restrictions on the values of non-updatable parameters, since these cannot be changed by an on-chain Parameter Update action. They can only be changed via a hard fork. Appendix E lists the non-updatable parameters.

Changes:

For completeness, guardrails could be defined for each of the non-updatable parameters, so that sensible new settings could be defined in future hard forks. These would need to be discussed with technical experts, including those familiar with the security and operation of the Cardano blockchain. The guardrails would apply to hard fork actions.

3.2 Parameter Groups

Parameters are grouped into four distinct groups: economic, network, technical/security, and governance. Each parameter group has different voting thresholds for change. The groups are listed in Appendix C.

Rationale:

The grouping allows different technical expertise to be applied to different kinds of parameters. Different cadences apply to each of the groups: network parameter settings need to be continually monitored on-chain; economic parameter settings typically need to be reviewed on a regular periodic basis; security/technical parameter settings typically need to be considered in response to on-chain events or prior to a hard fork; governance parameter settings typically follow human timescales.

The parameter groups are aligned with Advisory Groups in Intersect's Parameter Committee. Each parameter group contains a similar number of parameters, so that each expert group of the Parameter Committee has similar work levels. In several cases a single parameter may have multiple concerns, e.g. ***stakePoolTargetNum*** concerns security, network and economic matters. In this case, the group that has the most significant concerns over the parameter has been chosen. In debating a change, all relevant concerns should be considered.

The change thresholds are different for each parameter group so that different standards of governance can be applied. For example, governance parameter changes should require strong community consent, which may not be required in other cases.

In addition, some parameters have been identified as security concerns. These require SPO votes in addition to DRep votes.

Rationale:

The guardrails ensure that at least 50% of actively delegated stake consents to the security concerns. This reinforces the security constraints from the original Praos research papers.

4. Guardrails on Economic Parameters

Triggers for Change

- Significant changes in the fiat value of Ada resulting in potential problems with security, performance or functionality
- Changes in transaction volumes or types
- Community requests or suggestions
- Emergency situations that require changes to economic parameters

Counter-indicators

Changes to the economic parameters should not be made in isolation. They need to account for:

- External economic factors
- Network security concerns

Core Metrics

- Fiat value of Ada
- Transaction volumes and types
- Number and health of stake pools
- External economic factors

Rationale:

The need for economic changes may be triggered by a variety of events/system conditions, including ones that are not internal to the blockchain. Although economic considerations are important, they must not jeopardize the security or operation of the blockchain as a whole. Community input on changes is important, as is the balance between different classes of users (normal Ada holders, DeFi users, NFT holders, DApp developers, etc).

Changes:

Additional triggers or metrics may be added in the light of experience. Additional counter-indicators may also be identified. Existing triggers, metrics, or counter-indicators may also be adapted to meet the needs and goals of the Cardano ecosystem.

4.1 Transaction fee per byte (txFeePerByte) and Fixed transaction fee (txFeeFixed)

Defines the cost for basic transactions in Lovelace:

fee(tx) = txFeeFixed + txFeePerByte x nBytes(tx)

Guardrails

TFPB-01 (y)

Concerns: Security, Economic

***txFeePerByte must not** be lower than 30 (0.000030 ada). This protects against low-cost denial of service attacks.*

Rationale:

This guardrail provides a baseline to protect against resource-exhaustion attacks. It ensures that SPOs are paid for the resources (compute, storage, network, labour, etc.) that are used to process and store transactions on-chain.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [TFPB-02] and [TFPB-03]. Setting the limit too low raises the possibility that an attacker could flood the system with low-cost transactions. Setting it too high raises the possibility that users would be priced out, thereby rendering Cardano unusable. It is recommended to model the overall economic effect if ***txFeePerByte*** was set to the new limit. While this is not absolutely prohibited, it is not recommended to set the lower bound on ***txFeePerByte*** to zero, since then it would be possible to process transactions at a fixed price (***txFeeFixed)*** which would not account for all the computational and other costs associated with the transaction.

TFPB-02 (y)

Concerns: Security, Economic

txFeePerByte must not *exceed 1,000 (0.001 ada). This ensures that transactions can be paid for.*

Rationale:

This guardrail complements [TFPB-01]. It ensures that transactions are never excessively priced. The security concern is that lock-out could theoretically occur if the fees are too high, for example preventing some necessary on-chain action from taking place. The practical effect of excessive pricing would, however, be seen well before this point, with possible loss of users and perhaps operators. An unused blockchain is, by definition, insecure.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [TFPB-01] and [TFPB-03]. As with [TFPB-01], setting the limit too high raises the possibility that users would be priced out, thereby rendering Cardano unusable. Setting it too close to the lower bound specified by [TFPB-01] reduces choice in setting ***txFeePerByte***. As with [TFPB-01], it is recommended to model the overall economic effect if ***txFeePerByte*** was set to the new limit.

TFPB-03 (y)

Concerns: Limit Setting

txFeePerByte must not *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***txFeePerByte***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format. The guardrail overlaps [TFPB-02], but is retained to ensure that the hard lower bound is respected, even if [TFPB-02] is changed.

TFF-01 (y)

Concerns: Security, Economic

txFeeFixed ***must not*** *be lower than 100,000 (0.1 ada) This protects against low-cost denial of service attacks*

Rationale:

This guardrail provides a baseline to protect against resource-exhaustion attacks, supplementing [TFPB-01]. It ensures that SPOs are paid for the resources (compute, storage, network, labour etc.) that are used to process and store transactions on-chain. This cost is paid for all transactions and reflects the cost of the computational time and other resources that are needed to build a standard transaction.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [TFF-02] and [TFF-03]. As with [TFPB-01] setting too low a lower bound on ***txFeeFixed*** raises the possibility that an attacker could flood the system with low-cost transactions, while setting too high a lower bound on ***txFeeFixed*** raises the possibility that users would be priced out, so rendering Cardano unusable. It is recommended to model the overall economic effect if ***txFeeFixed*** was set to the new limit. While this is not prohibited, it is not recommended to set the lower bound on ***txFeeFixed*** to zero, since some overhead costs would then not be paid for.

TFF-02 (y)

Concerns: Security, Economic

txFeeFixed must not *exceed 10,000,000 (10 ada) This ensures that transactions can be paid for.*

Rationale:

This guardrail complements [TFF-01]. As with [TFPB-02], it ensures that transactions are never excessively priced. The security concern is that lock-out could theoretically occur if the fees are too high, for example preventing some necessary on-chain action. The practical effect of excessive pricing would, however, be seen well before this point, with possible loss of users and perhaps operators. An unused blockchain is, by definition, insecure.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [TFF-01] and [TFF-03]. As with [TFF-01], setting the limit too high raises the possibility that users would be priced out, thereby rendering Cardano unusable. Setting it too close to the lower bound specified by [TFF-01] reduces choice in setting ***txFeeFixed***. It is recommended to model the overall economic effect if ***txFeeFixed*** was set to the new limit.

TFF-03 (y)

Concerns: Limit Setting

txFeeFixed must not *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***txFeeFixed***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format. The guardrail overlaps [TFF-02], but is retained to ensure that the hard lower bound is respected, even if [TFF-02] is lowered.

TFGEN-01 (x - "should")

Concerns: Security

To maintain a consistent level of protection against denial-of-service attacks, ***txFeeFixed*** *and* ***txFeePerByte*** ***should*** *be adjusted whenever Plutus Execution prices are adjusted (****executionUnitPrices[steps/memory]****).*

Rationale:

This guardrail ensures consistency in pricing Plutus and normal transactions, avoiding asymmetric denial-of-service attacks.

Changes:

No change to this guardrail is possible.

TFGEN-02 (x - unquantifiable)

Concerns: Security, Economic

Any changes to ***txFeeFixed*** *or* ***txFeePerByte*** ***must*** *consider the implications of reducing the cost of a denial-of-service attack or increasing the maximum transaction fee so that it becomes impossible to construct a transaction.*

Rationale:

This guardrail ensures that security as well as economic concerns are addressed when either ***txFeeFixed*** *or* ***txFeePerByte*** is changed.

Changes:

No change to this guardrail is possible.

4.2. UTxO cost per byte (utxoCostPerByte)

Defines the cost for storage in UTxOs

- Sets a minimum threshold on ada that is held within a single UTxO (~1 ada minimum, could be >= 50 ada in the worst case)
- Provides protection against low-cost denial of service attack on UTxO storage. This attack has been executed on other chains - it is not theoretical. DoS protection decreases in line with the free node memory (proportional to UTxO growth)
- Helps reduce long term storage costs
- Provides an incentive to return UTxOs when no longer needed. Should significantly exceed minimum tx cost (~ 0.15 ada)

Guardrails

UCPB-01 (y)

Concerns: Security, Economic

***utxoCostPerByte must not** be lower than 3,000 (0.003 ada)*

Rationale:

This guardrail provides protection against resource-exhaustion attacks. It ensures that SPOs are paid for the cost that is incurred to keep the UTxO in memory and in storage. It imposes a cost on each UTxO that is designed to encourage UTxOs to be either amalgamated or burnt when they are no longer required, since there is a minimum cost on each UTxO.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [UCPB-02], [UCPB-03] and [UCPB-04]. Setting the limit too low could enable a low-cost denial-of-service attack. Setting the limit too high eliminates some uses of the blockchain, by making them economically infeasible.

It will be possible to reduce this limit, but not eliminate it, once out-of-memory storage becomes available for the UTxO set, enabling a bound to be set on memory usage. A calculation needs to be done on the expected cost of long-term out-of-memory storage. It cannot be eliminated, or set too low, since otherwise a user would be able to use the blockchain as a cheap, permanent storage solution for large amounts of data, thus reducing its utility for general users.

The overall economic and security effects should be properly modeled for any change to ***utxoCostPerByte.***

UCPB-02 (y)

Concerns: Security, Economic

utxoCostPerByte must not exceed 6,500 (0.0065 ada)

Rationale:

This guardrail ensures that the deposit for each UTxO is set at a level that allows reasonable economic use of the blockchain. It also ensures that the system can continue to function, since too high a value would mean that no changes could be made to the blockchain.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [UCPB-01], [UCPB-03] and [UCPB-04]. As with [UCPB-01], setting the limit too low could enable a low-cost denial-of-service attack. Setting the limit too high eliminates some uses of the blockchain, by making them economically infeasible.

The overall economic and security effects should be properly modeled for any change to ***utxoCostPerByte.***

UCPB-03 (y)

Concerns: Limit Setting

utxoCostPerByte must not *be zero*

Rationale:

Taken together with [UCPB-04], this guardrail protects against unacceptable settings of ***utxoCostPerByte.*** Two guardrails are used rather than one to simplify/make it easier to track the consistency of the guardrails script.

Changes:

No change to this guardrail is possible. The guardrail overlaps [UCPB-02]/[UCPB-04], but is retained to ensure that the hard lower bound is respected, even if [UCPB-02] is lowered.

UCPB-04 (y)

Concerns: Limit Setting

utxoCostPerByte must not be negative

Rationale:

Taken together with [UCPB-03], this guardrail protects against unacceptable settings of ***utxoCostPerByte.*** Two guardrails are used rather than one to simplify/make it easier to track the consistency of the guardrails script.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format. The guardrail overlaps [UCPB-02]/[UCPB-03], but is retained to ensure that the hard lower bound is respected, even if [UCPB-02] is lowered.

UCPB-05 (x - "should")

Concerns: Security

Changes need to account for:
i) The acceptable cost of attack
ii) The acceptable time for an attack (at least one epoch is assumed)
iii) The acceptable memory configuration for full node users (assumed to be 16GB for wallets or 24GB for stake pools)
iv) The sizes of UTxOs (~200B per UTxO minimum, up to about 10KB) and
v) The current total node memory usage

Rationale:

This covers known security concerns that might impact the setting of ***utxoCostPerByte***.

Changes:

Changes to this guardrail can be made as and when new attack vectors become apparent, as out-of-memory storage solutions are rolled out, or as acceptable memory configurations change.

4.3. Stake address deposit (stakeAddressDeposit)

Ensures that stake addresses are retired when no longer needed

- Helps reduce long term storage costs
- Helps limit CPU and memory costs in the ledger

*The rationale for the deposit is to incentivize that scarce memory resources are returned when they are no longer required. Reducing the number of active stake addresses also reduces processing and memory costs at the epoch boundary when calculating stake snapsho*ts.

Guardrails

SAD-01 (y)

Concerns: Security, Economic

***stakeAddressDeposit must not** be lower than 1,000,000 (1 ada)*

Rationale:

This guardrail ensures that a deposit is paid for long-term storage of stake addresses that is large enough to encourage stake addresses to be returned when they are no longer needed. It also ensures that the deposit is large enough to prevent a denial-of-service attack by using large numbers of stake addresses that are never returned.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [SAD-02] and [SAD-03]. Setting the limit too low could enable a low-cost denial-of-service attack through claiming unneeded stake addresses. Setting the limit excessively high will mean that it is infeasible for Ada holders to stake their Ada holdings, thereby reducing overall network security.

SAD-02 (y)

Concerns: Security, Economic

stakeAddressDeposit must not *exceed 5,000,000 (5 ada)*

Rationale:

This guardrail ensures that a deposit is paid for long-term storage of stake addresses that is large enough to encourage stake addresses to be returned when they are no longer needed. It also ensures that the deposit is large enough to prevent a denial-of-service attack by using large numbers of stake addresses.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [SAD-01] and [SAD-03]. As with [SAD-01], setting the limit too low could enable a low-cost denial-of-service attack through claiming and holding unneeded stake addresses.

SAD-03 (y)

Concerns: Limit Setting

stakeAddressDeposit must not be negative

Rationale:

This guardrail protects against unacceptable settings of ***stakeAddressDeposit.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format. The guardrail overlaps [SAD-01], but is retained to ensure that the hard lower bound is respected, even if [SAD-01] is lowered.

4.4. Stake pool deposit (stakePoolDeposit)

Ensures that stake pools are retired by the stake pool operator when no longer needed by them

- Helps reduce long term storage costs

The rationale for the deposit is to incentivize that scarce memory resources are returned when they are no longer required. Rewards and stake snapshot calculations are also impacted by the number of active stake pools.

Guardrails

SPD-01 (y)

Concerns: Security, Economic, Network, Decentralization

stakePoolDeposit ***must not*** *be lower than 250,000,000 (250 ada)*

Rationale:

This guardrail ensures that a deposit is paid for long-term storage of stake pool records that is large enough to encourage them to be returned when they are no longer needed. It also ensures that the deposit is large enough to prevent a denial-of-service attack by using large numbers of stake pools, and helps limit the number of pools that Ada holders can choose from.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [SPD-02] and [SPD-03]. Setting the limit too low could enable a low-cost denial-of-service attack through creating unneeded stake pools. Setting the limit excessively high will mean that it becomes economically infeasible to create the number of stake pools that are needed to ensure adequate decentralisation.

SPD-02 (y)

Concerns: Security, Economic, Network, Decentralization

***stakePoolDeposit must not** exceed 500,000,000 (500 ada)*

Rationale:

This guardrail ensures that the deposit on stake pool records is not set too high. This avoids the possibility that new stake pool operators can be priced out, so ensuring that adequate decentralisation/competition is possible.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [SPD-01] and [SPD-03]. As with [SPD-01], setting the limit too low could enable a low-cost denial-of-service attack through making it easy to create unneeded stake pools.

SPD-03 (y)

Concerns: Limit Setting

***stakePoolDeposit must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***stakePoolDeposit.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format. The guardrail overlaps [SPD-01], but is retained to ensure that the hard lower bound is respected, even if [SPD-01] is lowered.

4.5. Minimum Pool Cost (minPoolCost)

Part of the rewards mechanism

- The minimum pool cost is transferred to the pool rewards address before any delegator rewards are paid

Guardrails

MPC-01 (y)

Concerns: Limit Setting

***minPoolCost must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***minPoolCost.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MPC-02 (y)

Concerns: Economic, Network

***minPoolCost must not** exceed 500,000,000 (500 ada)*

Rationale:

This guardrail ensures that the minimum pool cost is not set so high that new stake pools cannot economically compete with existing stake pools, or that existing pools are priced out of the Cardano ecosystem through offering minimal or now rewards to delegators for low.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down. Setting it too high could allow **minPoolCost** to be set to a value that priced out some stake pools.

MPC-03 (y)

Concerns: Economic, Security

***minPoolCost should** be set in line with the economic cost for operating a pool*

Rationale:

This guardrail is designed to ensure that ***minPoolCost*** is stable against fiat values and relates to actual operating costs.

Changes:

The guardrail could be changed or eliminated if the Cardano community agreed that this was desirable. However, this would negate one of the assumptions that is made by the Cardano incentives scheme, and so have some security implication.

4.6. Treasury Cut (treasuryCut)

Part of the rewards mechanism

- The treasury cut portion of the monetary expansion is transferred to the treasury before any pool rewards are paid

- Can be set in the range 0.0-1.0 (0%-100%)

Guardrails

TC-01 (y)

Concerns: Economic, Sustainability

***treasuryCut must not** be lower than 0.1 (10%)*

Rationale:

This guardrail ensures that the treasury continues to collect income, enabling it to fund long term sustainability of Cardano (both for maintenance and for new development).

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, while respecting [TC-02], [TC-03] and [TC-04]. Setting the limit too high could require ***treasuryCut*** to be set to a value that made Cardano economically unviable. Setting the limit too low could allow ***treasuryCut*** to be set to a value that restricted or prevented further development or maintenance.

TC-02 (y)

Concerns: Economic

***treasuryCut must not** exceed 0.3 (30%)*

Rationale:

This guardrail ensures that the treasury cut does not dominate the economic use of Cardano, enabling delegators and stake pool operators to receive a fair return from fees and monetary expansion.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, while respecting [TC-01], [TC-03] and [TC-04]. Setting the limit too high could allow ***treasuryCut*** to be set to a value that made Cardano economically unviable. Setting the limit too low could force ***treasuryCut*** to be set to a value that restricted or prevented further development or maintenance.

TC-03 (y)

Concerns: Limit Setting

***treasuryCut must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***treasuryCut.*** Note that a setting of zero is technically allowed by this guardrail (but limited by [TC-01]).

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

TC-04 (y)

Concerns: Limit Setting

treasuryCut must not exceed 1.0 (100%)

Rationale:

This guardrail protects against unacceptable settings of ***treasuryCut.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

TC-05 (~ - no access to change history)

Concerns: Governance, Economic

treasuryCut must not *be changed more than once in any 36 epoch period (approximately 6 months)*

Rationale:

This guardrail is designed to ensure that ***treasuryCut*** remains stable and predictable over time. This affects the staking returns that are obtained by Ada holders and stake pools. It also affects future treasury income.

Changes:

The period determined by the guardrail could be changed, or the guardrail could be eliminated if the Cardano community agreed that this was desirable. However, this could make treasury and staking returns more volatile. Increasing the period excessively could make it impossible to make adjustments if these proved to be necessary.

4.7. Monetary Expansion Rate (monetaryExpansion)

Part of the rewards mechanism

- The monetary expansion controls the amount of reserves that is used for rewards each epoch

Governs the long-term sustainability of Cardano

- The reserves are gradually depleted until no rewards are supplied

Guardrails

ME-01 (y)

Concerns: Economic, Sustainability

***monetaryExpansion must not** exceed 0.005*

Rationale:

This guardrail ensures that the supply of funds in the Cardano reserves is not depleted too rapidly, thereby ensuring the long-term sustainability of Cardano.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, while respecting [ME-02] and [ME-03]. Setting the limit too high could allow ***monetaryExpansion*** to be set to a value that made Cardano unviable in the long-term. Setting the limit too low could force ***monetaryExpansion*** to be set to a value that made Cardano unattractive to delegators or SPOs.

ME-02 (y)

Concerns: Economic

***monetaryExpansion must not** be lower than 0.001*

Rationale:

This guardrail ensures that there is a minimum flow of funds from the reserves to the treasury and delegators. This means that there will be a guaranteed minimum economic return to stake pool operators and delegators to compensate for the cost of running the Cardano network.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, while respecting [ME-01] and [ME-02]. Setting the limit too high could allow ***monetaryExpansion*** to be set to a value that made Cardano unviable in the long-term. Setting the limit too low could require ***monetaryExpansion*** to be set to a value that made Cardano unattractive to delegators or SPOs, so reducing the economic viability of the blockchain.

ME-03 (y)

Concerns: Limit Setting

***monetaryExpansion must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***monetaryExpansion.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

ME-04 (x - "should")

Concerns: Governance, Economic

monetaryExpansion ***should not*** *be varied by more than +/- 10% in any 73-epoch period (approximately 12 months)*

Rationale:

This guardrail is designed to ensure that ***monetaryExpansion*** remains stable and predictable over time. As with [TC-04], this affects the staking returns that are obtained by Ada holders and stake pools. It also affects future treasury income. This guardrail is expressed as “should not” rather than “must not” in order to allow scope for changes in the economic environment to influence the setting of ***monetaryExpansion.***

Changes:

The period or amount that is determined by the guardrail could be changed, or the guardrail could be eliminated if the Cardano community agreed that this was desirable. However, this could make staking and treasury returns more volatile.

ME-05 (x - "should")

Concerns: Governance, Economic

monetaryExpansion ***should not*** *be be changed more than once in any 36-epoch period (approximately 6 months)*

Rationale:

This guardrail is designed to ensure that ***monetaryExpansion*** remains stable and predictable over time. As with [TC-04], this affects the staking returns that are obtained by Ada holders and stake pools. It also affects future treasury income. This guardrail is expressed as “should not” rather than “must not” in order to allow scope for changes in the economic environment to influence the setting of ***monetaryExpansion.***

Changes:

The period determined by the guardrail could be changed, or the guardrail could be eliminated if the Cardano community agreed that this was desirable. However, this could make staking and treasury returns more volatile.

4.8. Plutus Script Execution Prices (executionUnitPrices[priceSteps/priceMemory])

Define the fees for executing Plutus scripts

Gives an economic return for Plutus script execution

Provides security against low-cost DoS attacks

Guardrails

EIUP-PS-01 (y)

Concerns: Economic, Security

***executionUnitPrices[priceSteps] must not** exceed 2,000 / 10,000,000*

Rationale:

This guardrail ensures that Plutus script step execution prices are set to an economically sensible level, enabling a fair return to stake pool operators while not unnecessarily restricting Plutus script users.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, while respecting [EUIP-PS-02]. Setting the limit too high could allow ***executionUnitPrices [priceSteps]*** to be set to a value that made Plutus scripts economically unviable. Setting the limit too low could require ***executionUnitPrices[priceSteps]*** to be set to a value that allowed denial-of-service attacks through the use of large numbers of inexpensive Plutus scripts. Changes should consider the relative pricing of both execution steps and memory units for Plutus scripts.

EIUP-PS-02 (y)

Concerns: Security, Economic

***executionUnitPrices[priceSteps]* must not** *be lower than 500 / 10,000,000*

Rationale:

This guardrail ensures that script step execution prices are not reduced to a level that encourages denial-of-service attacks, and ensures that there is an economically sensible return to SPOs.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, while respecting [EUIP-PS-01]. Setting the limit too high could require ***executionUnitPrices [priceSteps]*** to be set to a value that made Plutus scripts economically unviable. Setting the limit too low could allow ***executionUnitPrices[priceSteps]*** to be set to a value that allowed denial-of-service attacks through the use of large numbers of inexpensive Plutus scripts. Changes should consider the relative pricing of both execution steps and memory units for Plutus scripts.

EIUP-PM-01 (y)

Concerns: Security, Economic

***executionUnitPrices[priceMemory]* must not** *exceed 2,000 / 10,000*

Rationale:

This guardrail ensures that Plutus script memory execution prices are set to an economically sensible level, as with Plutus script step prices. The memory cost is less significant than step cost, but the limit is set higher since far fewer memory units will be used in a Plutus script execution, and the overall fees should be set similarly for both execution steps and memory units.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, while respecting [EUIP-PM-02]. Setting the limit too high could allow ***executionUnitPrices [priceMemory]*** to be set to a value that made Plutus scripts economically unviable. Setting the limit too low could require ***executionUnitPrices[priceMemory]*** to be set to a value that allowed denial-of-service attacks through the use of large numbers of inexpensive Plutus scripts. Changes to ***executionUnitPrices[priceMemory]*** should consider the relative pricing of both execution steps and memory units for Plutus scripts.

EIUP-PM-02 (y)

Concerns: Security, Economic

executionUnitPrices[priceMemory] ***must not*** *be lower than* 400 / 10,000

Rationale:

This guardrail complements [EIUP-PM-01] to ensure that Plutus script memory execution prices are set to an economically sensible level. The memory cost is less significant than step cost, but the limit is set higher since benchmarking has shown that far fewer memory units will be used in a Plutus script execution, and the overall fees should be set similarly for both execution steps and memory units.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, while respecting [EUIP-PM-01]. Setting the limit too high could require ***executionUnitPrices [priceMemory]*** to be set to a value that made Plutus scripts economically unviable. Setting the limit too low could allow ***executionUnitPrices[priceMemory]*** to be set to a value that allowed denial-of-service attacks through the use of large numbers of inexpensive Plutus scripts. Changes to ***executionUnitPrices[priceMemory]*** should consider the relative pricing of both execution steps and memory units for Plutus scripts.

EUIP-GEN-01 (x - "similar to")

Concerns: Security

*The execution prices **must** be set so that: i) the cost of executing a transaction with maximum CPU steps is similar to the cost of a maximum sized non-script transaction and ii) the cost of executing a transaction with maximum memory units is similar to the cost of a maximum sized non-script transaction.*

Rationale:

This guardrail is designed to ensure that costs for different kinds of denial-of-service attacks are similar. The word “similar” is used to avoid completely rigid settings.

Changes:

No sensible changes are possible to this guardrail.

EUIP-GEN-02 (x - "should")

Concerns: Security

*The execution prices **should** be adjusted whenever transaction fees are adjusted (**txFeeFixed/txFeePerByte**). The goal is to ensure that the processing delay is similar for "full" transactions, regardless of their type. This helps ensure that the requirements on block diffusion/propagation times are met.*

Rationale:

As stated in the text above, this guardrail is designed to ensure that Ouroboros Praos security requirements on block diffusion/propagation times are maintained.

Changes:

No sensible changes are possible to this guardrail.

4.9. Transaction fee per byte for a reference script (minFeeRefScriptCoinsPerByte)

Defines the cost for using Plutus reference scripts in Lovelace

Guardrails

MFRS-01 (y)

Concerns: Economic, Security

minFeeRefScriptCoinsPerByte must not *exceed 1,000 (0.001 ada)*

Rationale:

This guardrail ensures that the cost of reference scripts is not set to an excessive level that would prevent them from being used/encourage the use of in-transaction scripts over reference scripts. Since the size of reference scripts is otherwise limited by **txSize,** excessive pricing could affect the economic viability of some transactions.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, while respecting [MFRS-02]. Setting the limit too high could allow ***minFeeRefScriptCoinsPerByte*** to be set to a value that made reference scripts economically unviable, or encouraged in-transaction scripts instead. Setting the limit too low could require ***minFeeRefScriptCoinsPerByte*** to be set to a value that allowed denial-of-service attacks through the use of excessively large reference scripts.

MFRS-02 (y)

Concerns: Limit Setting

***minFeeRefScriptCoinsPerByte must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***minFeeRefScriptCoinsPerByte.*** Note that a setting of zero is allowed by this guardrail. This is possible, but could lead to low cost denial of service attacks.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MFRS-03 (x - "should")

Concerns: Security

To maintain a consistent level of protection against denial-of-service attacks, ***minFeeRefScriptCoinsPerByte should*** be adjusted whenever Plutus Execution prices are adjusted (**executionUnitPrices[steps/memory**]) and whenever **txFeeFixed** is adjusted.

Rationale:

This guardrail is designed to ensure that costs for different kinds of denial-of-service attacks are similar.

Changes:

No sensible changes are possible to this guardrail.

MFRS-04 (x - "unquantifiable")

Concerns: Security, Economic

Any changes to ***minFeeRefScriptCoinsPerByte*** ***must*** consider the implications of reducing the cost of a denial-of-service attack or increasing the maximum transaction fee.

Rationale:

This guardrail is designed to reflect security and economic concerns in the choice of setting.

Changes:

No sensible changes are possible to this guardrail.

5. Guardrails on Network Parameters

Triggers for Change

Changes to network parameters may be triggered by:

1. Measured changes in traffic demands over a 2-epoch period (10 days)
2. Anticipated changes in traffic demands
3. Community requests

Counter-indicators

Changes may need to be reversed and/or should not be enacted in the event of:

- Excessive block propagation delays
- Stake pools being unable to handle traffic volume
- Scripts being unable to complete execution

Core Metrics

All decisions on parameter changes should be informed by:

- Block propagation delay profile
- Traffic volume (block size over time)
- Script volume (size of scripts and execution units)
- Script execution cost benchmarks
- Block propagation delay/diffusion benchmarks

Detailed benchmarking results are required to confirm the effect of any changes on mainnet performance or behavior prior to enactment. The effects of different transaction mixes must be analyzed, including normal transactions, Plutus scripts, and governance actions.

Rationale:

The need for network parameter changes may be triggered by a variety of events/system conditions, including ones that are not internal to the blockchain. These will generally be revealed through monitoring live or historic network traffic. More strategic changes, such as block size increases, will be informed by understanding competing user demands, and by the topology and configuration of the Cardano network as a whole.

Changes:

Additional triggers or metrics may be added in the light of experience. Additional counter-indicators may also be identified. Existing triggers, metrics, or counter-indicators may also be adapted to meet the needs and goals of the Cardano ecosystem.

5.1. General Network Guardrails

NETWORK-01 (x - "should")

Concerns: Monitoring

*No individual network parameter **should** change more than once per two epochs*

Rationale:

This guardrail ensures that the effect of each change on the network can be monitored.

Changes:

The number of epochs specified by the guardrail could be changed. Two epochs is generally the least time that allows for all network conditions to be monitored and compared with previous states.

NETWORK-01 (x - "should")

Concerns: Monitoring

Only one network parameter ***should*** be changed per epoch unless they are directly correlated, e.g., per-transaction and per-block memory unit limits

Rationale:

This guardrail ensures that the effects of individual network changes can be easily determined.

Changes:

No sensible change is possible to this guardrail.

5.2. Block Size (maxBlockBodySize)

The maximum size of a block, in Bytes.

Guardrails

MBBS-01 (y)

Concerns: Network, Security, Performance

***maxBlockBodySize must not** exceed 122,880 Bytes (120KB)*

Rationale:

This guardrail ensures that the block size is not too large. The limit has been determined conservatively by the timeliness requirements for Ouroboros Praos and known network and system performance characteristics.

Changes:

The exact upper bound specified by this guardrail could be increased, provided that this was supported by benchmarking and performance analysis to show that the worst case impact on block diffusion/propagation times and other metrics was consistent with the Ouroboros Praos security requirements. Increases in ***maxBlockBodySize*** may allow more or larger transactions to be processed. However, they may restrict Plutus script execution limits, or planned enhancements, such as the UTxO-HD out-of-memory storage solution.

MBBS-02 (y)

Concerns: Network, Performance

***maxBlockBodySize must not** be lower than 24,576 Bytes (24KB)*

Rationale:

This guardrail ensures that the block size cannot be too small. It is set to allow at least 1.5 full-sized transactions in a block.

Changes:

The lower bound on ***maxBlockBodySize*** must allow at least one full-sized transaction to be included in a block, and should normally be set to allow at least 1.5 full-sized transactions to be included in a block.

MBBS-03 (x - "exceptional circumstances")

Concerns: Network, Performance

***maxBlockBodySize must not** be decreased, other than in exceptional circumstances where there are potential problems with security, performance or functionality*

Rationale:

This guardrail ensures that the network operates consistently and predictably for end users, who rely on its expected performance and throughput.

Changes:

No sensible changes are possible to this guardrail.

MBBS-04 (~ - no access to existing parameter values)

Concerns: Network, Performance, System Integrity

maxBlockBodySize ***must*** *be large enough to include at least one transaction (that is,* ***maxBlockBodySize*** *must be at least* ***maxTxSize****)*

Rationale:

This guardrail imposes a strict lower bound on ***maxBlockBodySize***, ensuring that the network can always process at least one full-sized transaction per block.

Changes:

No sensible changes are possible to this guardrail.

MBBS-05 (x - "should")

Concerns: Network, Performance, Monitoring

maxBlockBodySize ***should*** *be changed by at most 10,240 Bytes (10KB) per epoch (5 days), and preferably by 8,192 Bytes (8KB) or less per epoch*

Rationale:

This guardrail ensures that the network adjusts gradually to changes in the block size, and that changes can be easily monitored and reverted if necessary.

Changes:

The exact numbers specified here could be varied in the light of experience with the network operation.

MBBS-06 (x - "should")

Concerns: Network, Performance, Security

*The block size **should not** induce an additional Transmission Control Protocol (TCP) round trip. Any increase beyond this must be backed by performance analysis, simulation and benchmarking*

Rationale:

This guardrail ensures that there is not a sudden increase in network round-trip times, which could affect the stability of the network as a whole.

Changes:

No sensible change is possible for this guardrail.

MBBS-07 (x - "unquantifiable")

Concerns: Network, Performance, Security

*The impact of any change to **maxBlockBodySize** must be confirmed by detailed benchmarking/simulation and not exceed the requirements of the block diffusion/propagation time budgets, as described below. Any increase to **maxBlockBodySize** must also consider future requirements for Plutus script execution (**maxBlockExecutionUnits[steps]**) against the total block diffusion target of 3s with 95% block propagation within 5s. The limit on maximum block size may be increased in the future if this is supported by benchmarking and monitoring results epoch*

Rationale:

This guardrail ensures that proper consideration is given to network performance characteristics, including future requirements.

Changes:

No sensible change is possible to this guardrail.

5.3. Transaction Size (maxTxSize)

The maximum size of a transaction, in Bytes.

Guardrails

MTS-01 (y)

Concerns: Performance

***maxTxSize must not** exceed 32,768 Bytes (32KB)*

Rationale:

This guardrail ensures that the transaction size is not too large. This limit is set conservatively to allow at least three full-sized transactions in a maximally sized block.

Changes:

The exact upper bound specified by this guardrail could be varied in line with changes to ***maxBlockBodySize.*** Increasing the limit allows larger transactions to be processed, but may restrict the total number of transactions that can be included in a block. While this is possible, it is not recommended to reduce the upper bound below the largest value that has ever been set for ***maxTxSize****,* since some previously successful transactions might then no longer be accepted by the network in future.

MTS-02 (y)

Concerns: Limit Setting

maxTxSize* *must not* *be negative

Rationale:

This guardrail protects against unacceptable settings of ***maxTxSize.*** A setting of zero is technically allowed by this guardrail, but would mean that no transactions could be accepted by the network.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MTS-03 (~ - no access to existing parameter values)

Concerns: System Integrity

maxTxSize* *must not* *be decreased

Rationale:

This guardrail ensures that all previously accepted transactions will still be accepted on chain.

Changes:

No sensible changes are possible to this guardrail.

MTS-04 (~ - no access to existing parameter values)

Concerns: System Integrity

maxTxSize must not** exceed **maxBlockBodySize

Rationale:

This guardrail ensures that a fully-sized transaction can be accepted on chain.

Changes:

No sensible changes are possible to this guardrail.

MTS-05 (x - "should")

Concerns: Performance, Monitoring

***maxTxSize should not** be increased by more than 2,560 Bytes (2.5KB) in any epoch, and preferably **should** be increased by 2,048 Bytes (2KB) or less per epoch*

Rationale:

This guardrail ensures that the network adjusts gradually to changes in the transaction size, and that changes can be easily monitored and reverted if necessary.

Changes:

The exact numbers specified here could be varied in the light of experience with the network operation.

MTS-06 (x - "should")

Concerns: Performance, Security

maxTxSize should not *exceed 1/4 of the block size*

Rationale:

This guardrail ensures that a block always contains multiple transactions. This helps with system load balancing under high load conditions, but also increases the difficulty of executing a successful denial-of-service attack: there is always an opportunity for non-malicious transactions to be processed in a block.

Changes:

No sensible change is possible for this guardrail.

5.4. Memory Unit Limits (maxBlockExecutionUnits[memory], maxTxExecutionUnits[memory])

The limit on the maximum number of memory units that can be used by Plutus scripts, either per-transaction or per-block.

Guardrails

MTEU-M-01 (y)

Concerns: Performance

maxTxExecutionUnits[memory] ***must not*** *exceed 40,000,000 units*

Rationale:

This guardrail ensures that the memory allocated by Plutus scripts in a transaction is not too large. Each memory unit represents 1 Byte of heap memory allocation, so the limit allows transactions to allocate 40MB. This limit is set conservatively against ***maxBlockExecutionUnits[memory]*** to allow at least three transactions in a block that each use the maximal memory allocation.

Changes:

The exact upper bound specified by this guardrail could be varied in line with changes to ***maxBlockExecutionUnits[memory].*** Increasing the limit allows Plutus scripts that allocate more memory to be processed, but may restrict the total number of transactions that can be included in a block. As specified by [MTEU-M-03], it is not recommended to reduce the upper bound below the largest value that has ever been set for ***maxTxExecutionUnits [memory]***, since some previously successful transactions might then no longer be accepted by the network.

MTEU-M-02 (y)

Concerns: Limit Setting

maxTxExecutionUnits[memory] ***must not*** *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxTxExecutionUnits[memory].*** A setting of zero is technically allowed by this guardrail, but would mean that no Plutus scripts could be executed. Such a setting might be needed if there was a temporary vulnerability affecting Plutus scripts, for example, but is not recommended under usual operating conditions. While [MTEU-M-01] and [MTEU-M-03] together make this guardrail technically redundant, it is retained to provide additional protection if either or both of those guardrails were to be changed.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MTEU-M-03 (~ - no access to existing parameter values)

Concerns: System Integrity

maxTxExecutionUnits[memory] ***must not*** *be decreased*

Rationale:

This guardrail ensures that all previously accepted Plutus scripts will still be accepted on chain.

Changes:

The guardrail could be removed if the consequences of rejecting previously allowable Plutus scripts were accepted.

MTEU-M-04 (x - "should")

Concerns: Performance, Monitoring

maxTxExecutionUnits[memory]* *should not *be increased by more than 2,500,000 units in any epoch*

Rationale:

This guardrail ensures that the network adjusts gradually to changes in Plutus script memory usage, and that changes can be easily monitored and reverted if necessary.

Changes:

The exact numbers specified here could be varied in the light of experience with the network operation.

MBEU-M-01 (y)

Concerns: Performance, Security

***maxBlockExecutionUnits[memory]* must not** *exceed 120,000,000 units*

Rationale:

This guardrail ensures that the memory allocated by Plutus scripts in a block is not too large. Each memory unit represents 1 Byte of heap memory allocation, so the limit allows a maximum of 120MB to be allocated. In the worst case, if there is no garbage collection, this means that Plutus scripts will impose 120MB of overhead on the node.

Changes:

The exact upper bound specified by this guardrail could be varied if this is supported by benchmarking and performance analysis**.** Excessive values could enable denial-of-service by requiring nodes to provision unsustainable amounts of memory. Lower values could restrict the total number of transactions that can be included in a block if ***maxBlockExecutionUnits[memory]***. Since execution times are roughly proportional to memory allocation rates, this limit works in tandem with ***maxBlockExecutionUnits[steps]***. Changes must ensure that the maximum script execution time for a block falls within the agreed Praos limits, as specified by [MBEU-M-04].

MBEU-M-02 (y)

Concerns: Limit Setting

***maxBlockExecutionUnits[memory]* must not** *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxBlockExecutionUnits [memory].*** As with [MTEU-M-02], a setting of zero is technically allowed by this guardrail, but would mean that no Plutus scripts could be executed. Such a setting might be needed if there was a temporary vulnerability affecting Plutus scripts, for example, but is not recommended for normal operations.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MBEU-M-03 (x - "should")

Concerns: Performance, Monitoring

***maxBlockExecutionUnits[memory]* should not** *be changed (increased or decreased) by more than 10,000,000 units in any epoch*

Rationale:

This guardrail ensures that the network adjusts gradually to changes in Plutus script memory usage, and that changes can be easily monitored and reverted if necessary.

Changes:

The exact numbers specified here could be varied in the light of experience with the network operation.

MBEU-M-04 (x - "unquantifiable")

Concerns: Performance, Security

*The impact of any change to **maxBlockExecutionUnits[memory]** **must** be confirmed by detailed benchmarking/simulation and not exceed the requirements of the diffusion/propagation time budgets, as also impacted by **maxBlockExecutionUnits[steps]**. Any increase **must** also consider previously agreed future requirements for the total block size (**maxBlockBodySize**) measured against the total block diffusion target of 3s with 95% block propagation within 5s. Future Plutus performance improvements may allow the per-block limit to be increased, but must be balanced against the overall diffusion limits as specified in the previous sentence, and future requirements.*

Rationale:

This guardrail ensures that the Praos timing guarantees are met.

Changes:

No sensible change is possible to this guardrail.

MEU-M-01 (~ - no access to existing parameter values)

Concerns: System Integrity

maxBlockExecutionUnits[memory] ***must not*** be less than ***maxTxExecutionUnits[memory]***

Rationale:

This guardrail ensures that at least one transaction that uses the maximum number of memory units can be accepted on chain.

Changes:

No sensible change is possible to this guardrail.

5.5. CPU Unit Limits (maxBlockExecutionUnits[steps], maxTxExecutionUnits[steps])

The limit on the maximum number of CPU steps that can be used by Plutus scripts, either per-transaction or per-block.

Guardrails

MTEU-S-01 (y)

Concerns: Performance

maxTxExecutionUnits[steps] ***must not*** *exceed 15,000,000,000 (15Bn) units*

Rationale:

This guardrail ensures that the CPU time that is used by Plutus scripts in a transaction is not too large. Each CPU unit represents 1 picosecond of execution, so the limit allows transactions to use 15ms of CPU time. This limit is set conservatively against ***maxBlockExecutionUnits[steps]*** to allow at least two transactions in a block that each use the maximum CPU execution time.

Changes:

The exact upper bound specified by this guardrail could be varied in line with changes to ***maxBlockExecutionUnits[steps].*** Increasing the limit allows Plutus scripts that use more CPU time to be processed, but may restrict the total number of transactions that can be included in a block. As specified by [MTEU-S-03], it is not recommended to reduce the upper bound below the largest value that has ever been set for ***maxTxExecutionUnits[steps]***, since some previously successful transactions might then no longer be accepted by the network.

MTEU-S-02 (y)

Concerns: Limit Setting

maxTxExecutionUnits[steps] ***must not*** *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxTxExecutionUnits[steps].*** A setting of zero is technically allowed by this guardrail, but would mean that no Plutus scripts could be executed. Such a setting might be needed if there was a temporary vulnerability affecting Plutus scripts, for example, but is not recommended under normal operating conditions. While [MTEU-S-01] and [MTEU-M-03] together make this guardrail technically redundant, it is retained to provide additional protection if either or both of those guardrails were to be changed.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MTEU-S-03 (~ - no access to existing parameter values)

Concerns: System Integrity

maxTxExecutionUnits[steps] ***must not*** *be decreased*

Rationale:

This guardrail ensures that all previous Plutus scripts will still be accepted on chain.

Changes:

The guardrail could be removed if the consequences of rejecting previous Plutus scripts were accepted.

MTEU-S-04 (x - "should")

Concerns: Performance, Monitoring

maxTxExecutionUnits[steps] ***should not*** *be increased by more than 500,000,000 (500M) units in any epoch (5 days)*

Rationale:

This guardrail ensures that the network adjusts gradually to changes in Plutus script CPU usage, and that changes can be easily monitored and reverted if necessary.

Changes:

The exact numbers specified here could be varied in the light of experience with the network operation.

MBEU-S-01 (y)

Concerns: Performance, Security

***maxBlockExecutionUnits[steps]* must not** *exceed 40,000,000,000 (40Bn) units*

Rationale:

This guardrail ensures that the CPU time that is used by Plutus scripts in a block is not too large. Each CPU unit represents 1 picosecond of execution, so the limit allows transactions to use 40ms of CPU time.

Changes:

The exact upper bound specified by this guardrail could be varied if this is supported by benchmarking and performance analysis. Higher values could enable denial-of-service attacks or reduce performance in an unacceptable way. Lower values could restrict the total number of transactions that can be included in a block. Since execution times are roughly proportional to memory allocation rates, this limit works in tandem with ***maxBlockExecutionUnits[memory]***. Changes must ensure that the maximum script execution time for a block falls within the agreed Praos limits as specified by [MBEU-S-04].

MBEU-S-02 (y)

Concerns: Limit Setting

***maxBlockExecutionUnits[steps]* must not** *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxBlockExecutionUnits[steps].*** As with [MTEU-M-02], a setting of zero is technically allowed by this guardrail, but would mean that no Plutus scripts could be executed. Such a setting might be needed if there was a temporary vulnerability affecting Plutus scripts, for example.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MBEU-S-03 (x - "should")

Concerns: Performance, Monitoring

maxBlockExecutionUnits[steps] ***should not*** *be changed (increased or decreased) by more than 2,000,000,000 (2Bn) units in any epoch (5 days)*

Rationale:

This guardrail ensures that the network adjusts gradually to changes in Plutus CPU usage, and that changes can be easily monitored and reverted if necessary.

Changes:

The exact numbers specified here could be varied in the light of experience with the network operation.

MBEU-S-04 (x - "unquantifiable")

Concerns: Performance, Security

The impact of any change to ***maxBlockExecutionUnits[steps]*** ***must*** *be confirmed by detailed benchmarking/simulation and not exceed the requirements of the diffusion/ propagation time budgets, as also impacted by* ***maxBlockExecutionUnits [memory]****. Any increase* ***must*** *also consider previously agreed future requirements for the total block size (****maxBlockBodySize****) measured against the total block diffusion target of 3s with 95% block propagation within 5s. Future Plutus performance improvements may allow the per-block limit to be increased, but must be balanced against the overall diffusion limits as specified in the previous sentence, and future requirements.*

Rationale:

This guardrail ensures that the Praos timing guarantees are met.

Changes:

No sensible change is possible to this guardrail.

MEU-S-01 (~ - no access to existing parameter values)

Concerns: System Integrity

maxBlockExecutionUnits[steps] ***must not*** *be less than* ***maxTxExecutionUnits[steps]***

Rationale:

This guardrail ensures that at least one transaction that uses the maximum number of Plutus CPU units can be accepted on chain.

Changes:

No sensible change is possible to this guardrail.

5.6. Block Header Size (maxBlockHeaderSize)

The size of the block header.

*Note that increasing the block header size may affect the overall block size (**maxBlockBodySize**).*

Guardrails

MBHS-01 (y)

Concerns: System Integrity, Performance

***maxBlockHeaderSize must not** exceed 5,000 Bytes*

Rationale:

This guardrail puts a reasonable limit on the header size. Together with [MBBS-02], it ensures that the majority of the block is devoted to transaction payload. It also helps ensure that [MBHS-05] is met.

Changes:

The exact upper bound specified by this guardrail could be varied if the protocol changed or ***maxBlockBodySize*** was altered.

MBHS-02 (y)

Concerns: Limit Setting

***maxBlockHeaderSize must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxBlockHeaderSize.*** A setting of zero is technically allowed by this guardrail, but would mean that no header could be included in the block.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MBHS-03 (x - "largest valid header" is subject to change)

Concerns: System Integrity

***maxBlockHeaderSize** must be large enough for the largest valid header*

Rationale:

This guardrail ensures that ***maxBlockHeaderSize*** is always set to a sensible value, so that a valid block can always be constructed.

Changes:

No sensible change is possible to this guardrail.

MBHS-04 (x - "should")

Concerns: System Integrity

maxBlockHeaderSize should *only normally be increased if the protocol changes*

Rationale:

This guardrail ensures that ***maxBlockHeaderSize*** is not changed arbitrarily.

Changes:

No sensible change is possible to this guardrail.

MBHS-05 (x - "should")

Concerns: Performance

maxBlockHeaderSize should *be within TCP's initial congestion window (3 or 10 MTUs*)

Rationale:

This guardrail ensures that the block header can be transmitted quickly and efficiently, within the time window that is needed for an explicit acknowledgement to be sent.

Changes:

No sensible change is possible to this guardrail, unless the underlying network stack is changed to use a protocol other than TCP.

6. Technical/Security Parameters

Triggers for Change

- Changes in the number of active SPOs
- Changes to the Plutus language
- Security threats
- Community requests

Counter-indicators

- Economic concerns, e.g. when changing the number of stake pools

Core Metrics

- Number of stake pools
- Level of decentralization

Rationale:

The need for technical/security parameter changes may be triggered by a variety of events/system conditions, including e.g. security threats that are not internal to the blockchain. These will generally be revealed through monitoring live or historic network traffic. More strategic changes, such as block size increases, will be informed by understanding competing user demands, and by the topology and configuration of the Cardano network as a whole.

Changes:

Additional triggers or metrics may be added in the light of experience. Additional counter-indicators may also be identified. Existing triggers, metrics, or counter-indicators may also be adapted to meet the needs and goals of the Cardano ecosystem.

6.1. Target Number of Stake Pools (stakePoolTargetNum)

Sets the target number of stake pools

- The expected number of pools when the network is in the equilibrium state
- Primarily a security parameter, ensuring decentralization by pool division/replication
- Has an economic effect as well as a security affect - economic advice is also required when changing this parameter
- Large changes in this parameter will trigger mass redelegation events

Guardrails

SPTN-01 (y)

Concerns: Decentralization, Economic, Security, Network

stakePoolTargetNum must not *be lower than 250*

Rationale:

This guardrail sets a lower bound on the target number of stake pools, ensuring that there will be at least 250 stake pools in an equilibrium state. The goal is to ensure sufficient network decentralisation to maintain good security guarantees and to provide economic incentives to maintain the network effectively.

Changes:

The exact lower bound specified by this guardrail could be varied, if the economic and security consequences were accepted. Lower values of ***stakePoolTargetNum*** potentially provide more rewards to fewer stake pools in the equilibrium state. Higher values would conversely allocate less rewards to more stake pools in the equilibrium state. Too high a lower limit on ***stakePoolTargetNum*** would force an unsustainable number of stake pools to be created, rendering some stake pools uneconomic and encouraging more "multi-pools"/"pool splitting". This could, counter-intuitively, **reduce** decentralisation, and increase the risk of disguised "Sybil" attacks. Experience with the Cardano mainnet shows that about twice as many pools as ***stakePoolTargetNum*** are created in practice. This reflects some real-world issues that are not covered by the theoretical incentives model, including "stake stickiness", "private pools", and "multi-pools". These real-world issues must be taken into account when changing this guardrail.

SPTN-02 (y)

Concerns: Decentralization, Economic, Security, Network

***stakePoolTargetNum must not** exceed 2,000*

Rationale:

This guardrail sets an upper bound on the target number of stake pools, targeting 2,000 stake pools in an equilibrium state. The goal is to ensure economic viability by providing a reasonable limit on the number of fully incentivised stake pools, so that the network can be properly maintained.

Changes:

The exact upper bound specified by this guardrail could be varied, if the consequences were accepted. As described for [SPTN–01], lower values of ***stakePoolTargetNum*** potentially provide more rewards to fewer stake pools in the equilibrium state. Higher values would conversely allocate less rewards to more stake pools in the equilibrium state. Too high an upper limit on ***stakePoolTargetNum*** would potentially allow an unsustainable number of stake pools to be created. Too low a limit would tend to concentrate stake into a few large pools, potentially opening opportunities for “Sybil” attacks. As also described for [SPTN-01], experience with the Cardano mainnet shows that about twice as many pools as ***stakePoolTargetNum*** are created in practice. This reflects some real-world issues that are not covered by the theoretical incentives model, including “stake stickiness”, “private pools”, and “multi-pools”. These real-world issues must be taken into account when changing this guardrail.

SPTN-03 (y)

Concerns: Limit Setting

***stakePoolTargetNum must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxBlockHeaderSize.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

SPTN-04 (y)

Concerns: Limit Setting

***stakePoolTargetNum must not** be zero*

Rationale:

This guardrail protects against unacceptable settings of ***stakePoolTargetNum.***

Changes:

No change to this guardrail is possible.

6.2. Pledge Influence Factor (poolPledgeInfluence)

Enables the pledge protection mechanism

Provides protection against Sybil attack

- Higher values reward pools that have more pledge and penalize pools that have less pledge

Has an economic effect as well as technical effect - economic advice is also required

- Can be set in the range 0.0-infinity

Guardrails

PPI-01 (y)

Concerns: Security, Economic

poolPledgeInfluence must not *be lower than 0.1*

Rationale:

This guardrail sets a reasonable lower bound on the pledge influence factor. It is designed to ensure that the security guarantees provided by the pledge cannot be eliminated.

Changes:

The exact lower bound specified by this guardrail could be varied, if the consequences were accepted. Lower values of ***poolPledgeInfluence*** reduce the incentive for stake pool owners to pledge their stake, thus making it easier to launch a "Sybil" attack where a single operator acquires a majority of stake via delegation. Conversely, higher values of ***poolPledgeInfluence*** increase the reward to pool owners, encouraging them to pledge their stake and accept more risk if their pool fails to perform adequately. An excessively high lower bound on ***poolPledgeInfluence*** would mean that stake pool owners would need to raise significant capital in order to finance their pool operation. This would reduce the number of pools that could be operated economically, thereby making "Sybil" attacks easier to launch.

PPI-02 (y)

Concerns: Security, Economic

***poolPledgeInfluence must not** exceed 1.0*

Rationale:

This guardrail sets a reasonable upper bound on the pledge influence factor. It is designed to ensure that pools can be set up and operated in an economically viable way.

Changes:

The exact upper bound specified by this guardrail could be varied, if the consequences were accepted. As described for [PPI-01], lower values of ***poolPledgeInfluence*** reduce the incentive for stake pool owners to pledge their stake, thus making it easier to launch a "Sybil" attack where a single operator acquires a majority of stake via delegation. Conversely, higher values of ***poolPledgeInfluence*** increase the reward to pool owners, encouraging them to pledge their stake and accept more risk if their pool fails to perform adequately. An excessively high lower bound would allow ***poolPledgeInfluence*** to be set to levels that would reduce the number of pools that could be operated economically, thereby making "Sybil" attacks easier to launch. Too low an upper bound would mean that stake pool pledges could not be used effectively to counter emerging "Sybil" attacks by attracting delegators to pools with higher levels of pledge.

PPI-03 (y)

Concerns: Limit Setting

***poolPledgeInfluence must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***poolPledgeInfluence.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

PPI-04 (x - “should”)

Concerns: Economic, Network

poolPledgeInfluence should not vary by more than +/- 10% in any 18-epoch period (approximately 3 months)

Rationale:

This guardrail ensures stability and certainty to stake pool operators and owners, who may need to raise funding for their pledge. By using the word “should”, the guardrail allows for ***poolPledgeInfluence*** to be raised quickly in the event that a “Sybil” attack was to emerge.

Changes:

The period or percentage could be changed if the Cardano community agreed.

6.3. Stake Pool Retirement Window (poolRetireMaxEpoch)

Defines the maximum number of epochs notice that a pool can give when planning to retire.

Guardrails

PRME-01 (y)

Concerns: Limit Setting

***poolRetireMaxEpoch must not** be negative*

Rationale:

This guardrail protects against unacceptable settings of ***poolRetireMaxEpoch.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

PRME-02 (x - “should”)

Concerns: Social, Network, Economic

***poolRetireMaxEpoch should not** be lower than 1*

Rationale:

This guardrail ensures that stake pools give delegators sufficient notice before the pool retires, allowing an opportunity to re-delegate their stake to other pools.

Changes:

The lower bound could be varied. Setting the limit to zero means that pools could retire at the end of the epoch in which they announced their retirement. Setting the limit too high could result in “zombie” pools that simply stopped operating without properly retiring. Delegators might not then be aware that their stake was no longer being used for block production.

6.4. Collateral Percentage (collateralPercentage)

Defines how much collateral must be provided when executing a Plutus script as a percentage of the normal execution cost

- Collateral is additional to fee payments
- If a script fails to execute, then the collateral is lost
- The collateral is never lost if a script executes successfully

Provides security against low-cost attacks by making it more expensive rather than less expensive to execute failed scripts

Guardrails

CP-01 (y)

Concerns: Security, Economic

collateralPercentage must not *be lower than 100*

Rationale:

This guardrail ensures that transactions that are deliberately submitted, despite having failed Plutus script Phase 1 checks, pay a reasonable penalty in fees. This discourages the use of failed Plutus scripts as an attack vector.

Changes:

The exact lower bound specified by this guardrail could be varied, if the consequences were accepted, while respecting [CP-02], [CP-03] and [CP-04]. Increasing the limit from 100 would have limited or no significant economic impact. Reducing the limit below 100 might make it profitable to exploit failed Plutus scripts as an attack.

CP-02 (y)

Concerns: Security, Economic

***collateralPercentage must not** exceed 200*

Rationale:

This guardrail ensures that the cost for failed Plutus scripts is not excessive, preventing the loss of significant collateral in the event of a user error.

Changes:

The exact upper bound specified by this guardrail could be varied, if the consequences were accepted, while respecting [CP-01], [CP-03] and [CP-04]. Increasing the upper bound excessively could force severe penalties on accidental errors. Reducing the limit might not allow sufficient penalty for deliberate exploits.

CP-03 (y)

Concerns: Limit Setting

***collateralPercentage must not** be negative*

Rationale:

Together with [CP-04], this guardrail protects against unacceptable settings of ***collateralPercentage.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

CP-04 (y)

Concerns: Limit Setting

***collateralPercentage must not** be zero*

Rationale:

Together with [CP-03], this guardrail protects against unacceptable settings of ***collateralPercentage.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules in the ledger.

6.5. Maximum number of collateral inputs (maxCollateralInputs)

Defines the maximum number of inputs that can be used for collateral when executing a Plutus script.

Guardrails

MCI-01 (y)

Concerns: Limit Setting

***maxCollateralInputs must not** be lower than 1*

Rationale:

This guardrail ensures that at least one collateral input can always be provided for a transaction involving Plutus scripts, so avoiding lock out.

Changes:

The exact lower bound specified by this guardrail could be increased from 1. The limit should not be reduced to zero.

6.6. Maximum Value Size (maxValueSize)

The limit on the serialized size of the Value in each output.

Guardrails

MVS-01 (y)

Concerns: System Integrity

maxValueSize must not *exceed 12,288 Bytes (12KB)*

Rationale:

This guardrail sets a reasonable upper bound on the maximum value size.

Changes:

The exact bound specified by this guardrail could be varied, if the implementation supported this.

MVS-02 (y)

Concerns: Limit Setting

maxValueSize must not *be negative*

Rationale:

This guardrail protects against unacceptable settings of ***maxValueSize.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

MVS-03 (~ - no access to existing parameter values)

Concerns: System Integrity

maxValueSize must** be less than **maxTxSize

Rationale:

This guardrail ensures that a maximally sized value can fit within a transaction.

Changes:

No sensible change is possible to this guardrail.

MVS-04 (~ - no access to existing parameter values)

Concerns: System Integrity

***maxValueSize must not** be reduced*

Rationale:

This guardrail ensures that previous values can continue to be accepted on-chain.

Changes:

No sensible change is possible to this guardrail.

MVS-05 (x - "sensible output" is subject to interpretation)

Concerns: System Integrity

***maxValueSize must** be large enough to allow sensible outputs (e.g. any existing on-chain output or anticipated outputs that could be produced by new ledger rules)*

Rationale:

This guardrail ensures that ledger changes that affect the value change can be accommodated on chain.

Changes:

No sensible change is possible to this guardrail.

6.7. Plutus Cost Models (costModels)

Define the base costs for each Plutus primitive in terms of CPU and memory unit

- There are about 150 distinct micro-parameters in total

Cost models are defined for each Plutus language version. A new language version may introduce additional micro-parameters or remove existing micro-parameters.

Guardrails

PCM-01 (x - unquantifiable)

Concerns: Performance, System Integrity, Security, Economic

***Cost model** values **must** be set by benchmarking on a reference architecture*

Rationale:

This guardrail ensures that Plutus cost model values are set rationally, by benchmarking, and that the results are consistent. This ensures stability and accuracy of pricing. It also ensures that Plutus scripts respect the required Praos-imposed timing guarantees.

Changes:

No sensible change is possible to this guardrail.

PCM-02 (x - primitives and language versions aren't introduced in transactions)

Concerns: System Integrity, Security, Economic

*The **cost model** **must** be updated if new primitives are introduced or a new Plutus language version is added*

Rationale:

This guardrail ensures that there is a cost model defined for each Plutus primitive, and that new Plutus language versions always have a corresponding cost model. This avoids the situation where new Plutus features are introduced, but cannot be used because no cost model is available.

Changes:

The guardrail could be changed if it was determined that a new Plutus language version should be added without it being available for use.

PCM-03 (x - "should")

Concerns: System Integrity, Security, Economic

***Cost model** values **should not** be negative*

Rationale:

This guardrail ensures that reasonable values are used for Plutus cost model values. In most cases, negative values are not permissible, since they could lead to negative costs that could permit low-priced denial-of-service attacks. Any exceptions to the general rule will need to be justified against the specific cost model for the primitives in question.

Changes:

No sensible change is possible to this guardrail.

PCM-04 (~ - no access to Plutus Cost Model parameters)

Concerns: System Integrity, Economic

*A **cost model** **must** be supplied for each Plutus language version that the protocol supports*

Rationale:

This guardrail ensures that scripts for new Plutus versions can be executed on chain, which would be impossible if no cost model was supplied. It complements [PCM-02] by requiring a new cost model, in addition to simply updating the overall set of cost models.

Changes:

As with [PCM-02], the guardrail could be changed if it was determined that a new Plutus language version should be added without it being available for use.

7. Governance Parameters

Triggers for Change

Changes to governance parameters may be triggered by:

1. Community requests
2. Regulatory requirements
3. Unexpected or unwanted governance outcomes
4. Entering a state of no confidence

Counter-indicators

Changes may need to be reversed and/or should not be enacted in the event of:

- Unexpected effects on governance
- Excessive Layer 1 load due to on-chain voting or excessive numbers of governance actions

Core Metrics

All decisions on parameter changes should be informed by:

- Governance participation levels
- Governance behaviors and patterns
- Regulatory considerations
- Confidence in the governance system
- The effectiveness of the governance system in managing necessary change

Rationale: The need for governance parameter changes may be triggered by a variety of events/system conditions, including e.g. failures to deal effectively with unanticipated governance events, or an inability to process the required volume of governance transactions on chain. Changes: Additional triggers or metrics may be added in the light of experience. Additional counter-indicators may also be identified. Existing triggers, metrics, or counter-indicators may also be adapted to meet the needs and goals of the Cardano ecosystem.

7.1. Deposit for Governance Actions (govDeposit)

The deposit that is charged when submitting a governance action.

- Helps to limit the number of actions that are submitted

Guardrails

GD-01 (y)

Concerns: Limit Setting

govDeposit must not be negative

Rationale:

This guardrail protects against unacceptable settings of ***govDeposit.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

GD-02 (y)

Concerns: Governance

***govDeposit must not** be lower than 1,000,000 (1 ada)*

Rationale:

This guardrail ensures that the deposit for each governance action is not too low. Since norms have not yet been established, the limit was chosen to allow the minimum value that was deemed to be acceptable by the community via the open governance workshops. A low value of ***govDeposit*** removes economic disincentives to submit governance actions on chain, potentially making governance more representative. However, this increases load on the chain and on governance participants (voters, DReps, SPOs, Constitutional Committee members and observers), which may make governance ineffective.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [GD-01] and [GD-03]. It should be reconsidered as governance practices become established.

GD-03 (y)

Concerns: Governance

***govDeposit must not** exceed 10,000,000,000,000 (10 Million ada)*

Rationale:

This guardrail ensures that the deposit for each governance action is not excessive. Since norms have not yet been established, it was chosen to allow the maximum value that was deemed to be acceptable by the community via the open governance workshops. A high value of ***govDeposit*** restricts the number of governance actions that will be submitted on chain, so reducing load on the chain and on governance participants (voters, DReps, SPOs, Constitutional Committee members and observers) and making governance more efficient and effective. However, it may limit participation in governance by Ada holders, perhaps requiring coalitions to be formed in order to submit on-chain governance actions, including **Info** actions.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [GD-01] and [GD-02]. It should be reconsidered as governance practices become established.

GD-04 (x - "should")

Concerns: Economic, Governance

***govDeposit should** be adjusted in line with fiat changes*

Rationale:

This guardrail ensures that governance deposits are pegged in line with the external value of Ada. This creates consistency in the cost of participating in Cardano governance.

Changes:

The guardrail could be removed if the community decided that external mapping was not necessary.

7.2. Deposit for DReps (dRepDeposit)

The deposit that is charged when registering a DRep.

- Helps to limit the number of active DReps

Guardrails

DRD-01 (y)

Concerns: Limit Setting

dRepDeposit must not be negative

Rationale:

This guardrail protects against unacceptable settings of ***dRepDeposit.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

DRD-02 (y)

Concerns: Governance

***dRepDeposit must not** be lower than 1,000,000 (1 ada)*

Rationale:

This guardrail ensures that the deposit for dReps is not too low. Since norms have not yet been established, it was chosen to allow the minimum value that was deemed to be acceptable by the community via the open governance workshops. A low value of ***dRepDeposit*** removes economic disincentives to register as a DRep, making governance more representative by reducing economic barriers for DReps and also supporting self-registrations. However, it may increase voting load on the chain and create confusion of choice in voters.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [DRD-01] and [DRD-03]**.** It should be reconsidered as governance practices become established.

DRD-03 (y)

Concerns: Governance

***dRepDeposit must not** exceed 100,000,000,000 (100,000 ada)*

Rationale:

This guardrail ensures that the deposit for DReps is not excessive. Since norms have not yet been established, it was chosen to allow the maximum value that was deemed to be acceptable by the community. A high value for ***dRepDeposit*** restricts the number of DReps that will register, so reducing voter choice, and limiting self-delegations. This reduces on-chain load, and reduces voter confusion through excessive choice. However, it makes it harder/impossible for delegators to self-delegate their voting power, reduces the choice of alternative DReps, and may concentrate political power and incentives.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [GD-01] and [GD-02]. It should be reconsidered as governance practices become established.

DRD-04 (x - "should")

Concerns: Economic, Governance

***dRepDeposit should** be adjusted in line with fiat changes*

Rationale:

This guardrail ensures that deposits for DReps are pegged in line with the external value of Ada. This creates consistency in the cost of participating in Cardano governance.

Changes:

The guardrail could be removed if the community decided that external mapping was not necessary.

7.3. DRep Activity Period (dRepActivity)

The period (as a whole number of epochs) after which a DRep is considered to be inactive for vote calculation purposes, if they do not vote on any proposal.

Guardrails

DRA-01 (y)

Concerns: Governance

dRepActivity must not *be lower than* 13 epochs (2 months)

Rationale:

This guardrail ensures that DReps are not recorded as inactive prematurely, which could result in the loss of vote delegators to other DReps, or a reduction in the count of active stake (so changing voting outcomes).

Changes:

The exact lower bound specified by this guardrail could be varied either up or down, respecting [DRA-02] and [DRA-03]**.**

DRA-02 (y)

Concerns: Governance

***dRepActivity must not** exceed 37 epochs (6 months)*

Rationale:

This guardrail ensures that DReps must participate in some governance activity if they are to avoid being recorded as inactive.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down, respecting [GD-01] and [GD-02].

DRA-03 (y)

Concerns: Limit Setting

dRepActivity must not be negative

Rationale:

This guardrail protects against unacceptable settings of ***dRepActivity.***

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

DRA-04 (~ - no access to existing parameter values)

Concerns: Governance

dRepActivity must** be greater than **govActionLifetime

Rationale:

This guardrail ensures that at least one action could be voted on before a DRep is considered to be inactive.

Changes:

No sensible changes are possible.

DRA-05 (x - "should")

Concerns: Governance

***dRepActivity should** be calculated in human terms* (2 months etc)

Rationale:

This guardrail ensures that on-chain DRep activity periods correspond to human activity. This was requested by the community.

Changes:

The guardrail could be removed if the community decided that it was acceptable to use internal time measurements only (Cardano epochs). In that case, the activity period could change if the epoch length was altered (currently set to 5 days).

7.4. DRep and SPO Governance Action Thresholds (dRepVotingThresholds[...], poolVotingThresholds[...])

Thresholds on the active voting stake that is required to ratify a specific type of governance action by either DReps or SPOs.

- Ensures legitimacy of the action

Guardrails

VT-GEN-01 (y)

Concerns: Governance

All thresholds must be in the range 50%-100%

Rationale:

This guardrail ensures that security requirements on SPO and DRep voting thresholds are always met. It also ensures that reasonable expectations on governance majorities are always met. The upper limit ensures that it is always theoretically possible to meet a voting threshold. The guardrail applies both to SPO and to DRep votes.

Changes:

No changes are possible to this guardrail. The upper bound is also enforced by rules in the ledger.

VT-GEN-02 (y)

Concerns: Governance, Security

*Economic, network and technical parameter thresholds **must** be in the range 51%-75%*

Rationale:

This guardrail ensures that economic, network and technical parameter changes have a clear majority of the relevant voters. It also ensures that changes are not too difficult to make. The guardrail applies to both DRep and SPO votes. It implicitly supports [PARAM-03] that applies a strict majority threshold to SPO votes that are needed for security reasons on critical parameters and [PARAM-05] that applies a strict majority threshold to DRep votes that are needed for security reasons on critical governance parameter changes.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [VT-GEN-01], [PARAM-03] and [PARAM-05].

VT-GEN-03 (y)

Concerns: Governance, Security

*Governance parameter thresholds **must** be in the range 75%-90%*

Rationale:

This guardrail ensures that governance parameter changes have a strong super-majority of the relevant voters. It also ensures that changes are not effectively impossible to make. The guardrail applies only to DRep votes. It implicitly supports [PARAM-03] that applies a strict majority threshold to SPO votes that are needed for security reasons on critical parameters and [PARAM-05] that applies a strict majority threshold to DRep votes that are needed for security reasons on critical governance parameter changes.

Changes:

The exact bounds specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [VT-GEN-01],[PARAM-03] and [PARAM-05].

VT-HF-01 (y)

Concerns: Governance, Security, Network

***Hard fork** action thresholds **must** be in the range 51%-80%*

Rationale:

This guardrail ensures that hard forks have sufficient support by SPOs and DReps, but are not required to be overly difficult to achieve. The guardrail applies both to SPO and to DRep votes.

Changes:

The exact bounds specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [VT-GEN-01], [PARAM-03] and [PARAM-05], and considering the security/network implications if too little delegated stake supports a hard fork.

VT-CON-01 (y)

Concerns: Governance

Update Constitution or proposal policy *action thresholds* ***must*** *be in the range 65%-90%*

Rationale:

This guardrail ensures that changes to the constitution (and/or proposal policy) have strong DRep support, but are not effectively impossible as could happen if excessive bounds were set.

Changes:

The exact bounds specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [VT-GEN-01].

VT-CC-01 (y)

Concerns: Governance

Update Constitutional Committee *action thresholds* ***must*** *be in the range 51%-90%*

Rationale:

This guardrail ensures that changes to the constitutional committee (and/or size and threshold) have strong DRep support, but are not effectively impossible. The guardrail applies both to SPO and to DRep votes.

Changes:

The exact bounds specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [VT-GEN-01].

VT-NC-01 (y)

Concerns: Governance

***No Confidence** action thresholds **must** be in the range 51%-75%*

Rationale:

This guardrail ensures that a vote to enter a state of no confidence is always possible. The upper limit is chosen so that No Confidence can never be excessively hard to achieve. Since SPOs do not vote on No Confidence actions, the guardrail applies only to DRep votes.

Changes:

The exact bounds specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [VT-GEN-01].

7.5. Governance Action Lifetime (govActionLifetime)

The period after which a governance action will expire if it is not enacted

- As a whole number of epochs

Guardrails

GAL-01 (y)

Concerns: Limit Setting

***govActionLifetime must not** be lower than 1 epoch (5 days)*

Rationale:

This guardrail ensures that governance actions have sufficient time to be enacted. The minimum period of one epoch has been chosen to be the same as the period that is available for ratifying actions pre-CIP1694.

Changes:

The lower bound on the governance action lifetime could be increased if this was determined to be necessary. However, this might prevent very rapid action in the event that a parameter needed to be changed urgently, for example, or if a hard fork was necessary to avoid some undesirable outcome. The lower bound should not be reduced to zero, since it might then be impossible to submit and vote on governance actions.

GAL-02 (y)

Concerns: Governance

***govActionLifetime must not** exceed 15 epochs (75 days)*

Rationale:

This guardrail ensures that governance actions do not persist indefinitely on chain, creating resource issues. 75 days is considered by the community to be a reasonable time to allow voting on a governance action, even if this is contentious.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, respecting [GAL-01] and [GAL-03]**.**

GAL-03 (x - "should")

Concerns: Governance

***govActionLifetime should not** be lower than 2 epochs (10 days)*

Rationale:

This guardrail gives stronger guidance on ***govActionLifetime.*** It is likely that DReps, SPOs and the Constitutional Committee will need more than a few days to consider and ratify a vote, as could be the case if only [GAL-01] were enforced.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, respecting [GAL-01] and [GAL-02]**.**

GAL-04 (x - "should")

Concerns: Governance

govActionLifetime should *be calibrated in human terms (eg 30 days, two weeks), to allow sufficient time for voting etc. to take place*

Rationale:

This guardrail ensures that governance action periods correspond to human activity rather than on-chain. This was requested by the community. .

Changes:

The guardrail could be removed if the community decided that it was acceptable to use internal time measurements only (Cardano epochs). In that case, the activity period could change if the epoch length was altered (currently set to 5 days).

GAL-05 (~ - no access to existing parameter values)

Concerns: Governance

govActionLifetime must *be less than* ***dRepActivity***

Rationale:

This guardrail complement [DRA-04]. As with [DRA-04], this guardrail ensures that at least one action could be voted on before a DRep is considered to be inactive.

Changes:

No sensible change is possible to this guardrail.

7.6. Maximum Constitutional Committee Term (committeeMaxTermLength)

The length of the maximum term that a committee member may serve

Guardrails

CMTL-01 (y)

Concerns: Limit Setting

***committeeMaxTermLength must not** be zero*

Rationale:

This guardrail sets a required lower limit on the committee maximum term limit. It supplements [CMTL-02].

Changes:

No change to this guardrail is possible. The limit is also enforced by rules in the ledger.

CMTL-02 (y)

Concerns: Limit Setting

***committeeMaxTermLength must not** be negative*

Rationale:

This guardrail sets a required lower limit on the committee maximum term limit. It complements [CMTL-01]. Two guardrails are used rather than one to simplify/make it easier to track the consistency of the guardrails script.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

CMTL-03 (y)

Concerns: Governance

***committeeMaxTermLength must not** be lower than 18 epochs*
(90 days, or approximately 3 months)

Rationale:

This guardrail ensures that the Constitutional Committee does not churn too rapidly.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, respecting [CMTL-01], [CMTL-02] and [CMTL-04].

CMTL-04 (y)

Concerns: Governance

committeeMaxTermLength* *must not *exceed 293 epochs (approximately 4 years)*

Rationale:

This guardrail ensures that the Constitutional Committee will not persist beyond a reasonable limit (similar to political terms). A re-election is required at least every 4 years.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, respecting [CMTL-01], [CMTL-02] and [CMTL-03].

CMTL-05 (x - "should")

Concerns: Governance

committeeMaxTermLength* *should not *exceed 220 epochs (approximately 3 years)*

Rationale:

This guardrail gives stronger guidance on ***committeeMaxTermLength.*** 3 years is a reasonable term for a committee member to serve without being re-elected.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, respecting [CMTL-01], [CMTL-02], [CMTL-03] and [CMTL-04].

7.7. The minimum size of the Constitutional Committee (committeeMinSize)

The least number of members that can be included in a Constitutional Committee following a governance action to change the Constitutional Committee.

Guardrails

CMS-01 (y)

Concerns: Limit Setting

***committeeMinSize must not** be negative*

Rationale:

This guardrail sets a required lower limit on the committee minimum size.

Changes:

No change to this guardrail is possible. The limit is also enforced by rules on the CDDL format.

CMS-02 (y)

Concerns: Governance, Security

***committeeMinSize must not** be lower than 3*

Rationale:

This guardrail ensures that the Constitutional Committee cannot be subverted by having too few members.

Changes:

The exact lower bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, respecting [CMS-01] and [CMS-03].

CMS-03 (y)

Concerns: Governance, Security

***committeeMinSize must not** exceed 10*

Rationale:

This guardrail ensures that the Constitutional Committee will be a manageable size. Coordination between members may be needed when voting on specific governance actions, for example, and good security procedures may need to be followed.

Changes:

The exact upper bound specified by this guardrail could be varied either up or down in the light of experience with the governance system, while respecting [CMS-01] and [CMS-02].

8. Treasury Withdrawal Actions

Treasury withdrawal *actions specify the destination and amount of a number of withdrawals from the Cardano treasury.*

Guardrails

TREASURY-01 (x)

Concerns: Governance, Economic

*DReps **must** define a net change limit for the Cardano Treasury's balance per period of time*

Rationale:

This guardrail defines the process for deciding how fast the treasury can be depleted. It cannot be implemented on chain via the guardrails script since the current treasury balance is not exposed to the script, and the guardrails script applies only on action submission, rather than on enactment. The change limit could be defined via an Info action, for example.

Changes:

This guardrail could be changed if the community agrees on alternative financial controls for the treasury.

TREASURY-02 (x)

Concerns: Governance, Economic

*The budget for the Cardano Treasury **must not** exceed the net change limit for the Cardano Treasury's balance per period of time*

Rationale:

This guardrail ensures consistency of the budget against the spending limit of [TREASURY-01]. Since there is no on-chain representation of the budget or the net change limit, this guardrail cannot currently be implemented via the guardrails script.

Changes:

This guardrail could be changed if the community agrees on alternative financial controls for the treasury.

TREASURY-03 (x)

Concerns: Governance, Economic

*The budget for the Cardano Treasury **must** be denominated in ada.*

Rationale:

This ensures that the budget for treasury withdrawals is known ahead of time against the value that is held in the treasury. This simplifies financial accounting, and means that spending guarantees can be made without the need for external borrowing, for example.

Changes:

This guardrail could be changed if the community agrees to use some other denomination for the treasury budget.

TREASURY-04 (x)

Concerns: Governance

*Treasury withdrawals **must not** be ratified until there is a community-approved Cardano budget then in effect pursuant to a previous on-chain governance action agreed by the DReps with a threshold of greater than 50% of the active voting stake.*

Rationale:

This ensures that the community has agreed the budget allocation before any withdrawal is made from the treasury. This cannot be enforced by the guardrails script since there is no on-chain representation of the budget, and there is no specific governance action to record the DRep vote. Unlike [INTERIM-02], which specifies a similar condition, the guardrail applies even after the interim governance period has completed. The nature of the governance action needs to be agreed by the Cardano community.

Changes:

This guardrail could be changed if the community agrees on alternative financial controls for the treasury.

9. Hard Fork Initiation Actions

*The **hard fork initiation** action requires both a new major and a new minor protocol version to be specified.*

- As positive integers

As the result of a hard fork, new updatable protocol parameters may be introduced. Guardrails may be defined for these parameters, which will take effect following the hard fork. Existing updatable protocol parameters may also be deprecated by the hard fork, in which case the guardrails become obsolete for all future changes.

Guardrails

HARDFORK-01 (~ - no access to existing parameter values)

Concerns: System Integrity

*The major protocol version **must** be the same as or one greater than the major version that will be enacted immediately prior to this change. If the major protocol version is one greater, then the minor protocol version **must** be zero.*

Rationale:

This guardrail ensures that the protocol version changes correctly and consistently as a result of a hard-fork initiation, mandating that major versions are not skipped. Generally, only the major protocol version will be incremented via a hard fork. The guardrail then requires the minor version to be reset to zero, so reinitialising the minor protocol version sequence.

Changes:

No sensible changes are possible to this guardrail.

HARDFORK-02 (~ - no access to existing parameter values)

Concerns: System Integrity

*The minor protocol version **must** be no less than the minor version that will be enacted immediately prior to this change*

Rationale:

This guardrail ensures that the minor protocol version also changes consistently as a result of a hard-fork initiation, mandating that major versions are not skipped. This guardrail applies only when the minor protocol version is changed. Normally [HARDFORK-01] will take precedence.

Changes:

No sensible changes are possible to this guardrail.

HARDFORK-03 (~ - no access to existing parameter values)

Concerns: System Integrity

*At least one of the protocol versions (major or minor or both) **must** change.*

Rationale:

This guardrail ensures that the hard fork impacts one or both of the major and minor protocol versions, thus ensuring that the hard fork action has an on-chain effect.

Changes:

No sensible changes are possible to this guardrail.

HARDFORK-04 (x)

Concerns: System Integrity, Network, Security

*At least 85% of stake pools by active stake **should** have upgraded to a Cardano node version that is capable of processing the rules associated with the new protocol version.*

Rationale:

This guardrail ensures that the network does not fork, by requiring the vast majority of the network to have upgraded to a correct version of the Cardano node prior to a hard fork event. The guardrail leaves scope for alternative implementations of the node to be used, as long as they can be shown to be correct. It does not mandate any specific way of determining whether the required upgrade percentage has been met. The guardrail allows flexibility and judgment to be exercised over the exact upgrade percentage, avoiding situations where, for example, 84.9% of stake pools have upgraded. The requirement is deliberately made on the delegated stake rather than the number of stake pools in order to ensure that the vast majority of the stake will support the upgrade. This also helps meet security requirements.

Changes:

The exact percentage could be changed, if the Cardano community agreed that a different upgrade percentage was sufficient.

HARDFORK-05 (x)

Concerns: System Integrity

*Any new updatable protocol parameters that are introduced with a hard fork **must** be included in this Appendix and suitable guardrails defined for those parameters.*

Rationale:

This meta-guardrail ensures that guardrails are not neglected when system upgrades are performed.

Changes:

No sensible changes are possible to this guardrail.

HARDFORK-06 (x)

Concerns: System Integrity

*Settings for any new protocol parameters that are introduced with a hard fork **must** be included in the appropriate Genesis file.*

Rationale:

This guardrail ensures the consistency of the Genesis file, requiring that parameter changes can be traced back to their origins

Changes:

No sensible changes are possible to this guardrail.

HARDFORK-07 (x)

Concerns: Documentation, Consistency

*Any deprecated protocol parameters **must** be indicated in this Appendix.*

Rationale:

This guardrail complements [HARDFORK-05], ensuring that parameters have a known end-of-life, and that guardrails that are no longer required can be eliminated or suitably adjusted.

Changes:

No sensible changes are possible to this guardrail.

HARDFORK-08 (~ - no access to existing parameter values)

Concerns: System Integrity

*New Plutus versions **must** be supported by a version-specific **Plutus cost model** that covers each primitive that is available in the new Plutus version.*

Rationale:

This guardrail ensures that Plutus cost models are correctly updated when new primitives are added.

Changes:

No sensible changes are possible to this guardrail.

10. Update Constitutional Committee Actions

Update Constitutional Committee or Threshold *governance actions may change the size, composition or required voting thresholds for the Constitutional Committee*

Guardrails

UPDATE-CC-01 (x)

Concerns: Governance

Update Constitutional Committee and/or threshold and/or term governance actions must not *be ratified until Ada holders have ratified through an on-chain governance action the Final Constitution.*

Rationale:

This guardrail limits how fast the treasury can be depleted. It cannot be implemented on chain via the guardrails script since there is no access to Constitutional Committee data, or to the Constitution.

Changes:

This guardrail can be eliminated following the Interim period.

11. New Constitution Actions

New constitution or proposal policy *actions change the hash of the on-chain constitution and the associated governance proposal policy script.*

Guardrails

NEW-CONSTITUTION-01 (x)

Concerns: Governance

A New Constitution and/or proposal policy governance action must *be submitted to define any required guardrails for new parameters that are introduced via a* ***Hard Fork*** *governance action.*

Rationale:

This guardrail ensures that new guardrails are defined and the guardrails script updated on chain for any new parameters that are introduced by a hard fork.

Changes:

This guardrail cannot sensibly be changed.

12. No Confidence Actions

No confidence *actions signal a state of no confidence in the governance system. No guardrails are imposed on* ***No Confidence*** *actions.*

Guardrails

- None

13. Info Actions

***Info** actions are not enacted on chain. No guardrails are imposed on **Info** actions.*

Guardrails

- None

14. Guardrails during the Interim Period

The Interim Period begins with the Chang Hard-Fork and ends after a community-ratified Final Constitution is enacted on-chain. Throughout the Interim Period, technical and constitution-enforced triggers will progressively turn on the features of CIP-1694.

Guardrails

INTERIM-01 (x)

Concerns: Governance

*To provide sufficient time for DReps to register and campaign and for Ada holders to choose their initial voting delegations, at least 18 epochs (90 days, or approximately 3 months) **must** elapse after the Chang hard fork before the subsequent hard fork can be ratified. Once the subsequent hard fork is enacted, DRep voting can occur as described in CIP-1694.*

Rationale:

This guardrail ensures that human political dynamics are respected as CIP-1694 is bootstrapped via the two hard forks.

Changes:

This guardrail is no longer relevant after the interim period has completed.

INTERIM-02 (x)

Concerns: Governance, Economic

*Treasury withdrawals **must not** be ratified until there is a community-approved Cardano Blockchain Ecosystem budget then in effect pursuant to a previous on-chain governance action.*

Rationale:

This guardrail enforces proper financial controls on the Cardano treasury as the governance system is bootstrapped. As with [TREASURY-04], this ensures that the community has agreed the budget allocation before any withdrawal is made from the treasury. This guardrail cannot be enforced by the guardrails script since there is no on-chain representation of the budget, and there is no specific governance action to record the DRep vote. The nature of the governance action needs to be agreed by the Cardano community.

Changes:

This guardrail is no longer relevant after the interim period has completed. It will be replaced by [TREASURY-04].

INTERIM-03 (x)

Concerns: Governance, Economic

*Treasury withdrawals **must** be consistent with the community-approved Cardano Blockchain ecosystem budget(s).*

Rationale:

This guardrail extends [INTERIM-02]. It ensures proper financial controls on the Cardano treasury as the governance system is bootstrapped. This guardrail cannot be enforced by the guardrails script since there is no on-chain representation of the budget

Changes:

This guardrail applies only to the interim period. An equivalent permanent guardrail might be desirable.

INTERIM-04 (x)

Concerns: Governance

*Ada holders **must** have ratified the Final Constitution as specified in Appendix II before ratifying any other proposed **new constitution**, **update constitutional committee and/or threshold and/or terms**, and **motion of no-confidence** governance actions.*

Rationale:

This guardrail ensures that the interim period ends in an orderly way with the ratification of the Final Constitution. The Interim Constitutional Committee can only be changed after the interim period has completed, including by a motion of No Confidence.

Changes:

This guardrail is no longer relevant after the interim period has completed.

INTERIM-05 (x)

Concerns: Governance

***New guardrails script** actions that are consistent with the Interim Constitution may be ratified during the interim period, provided the Interim Constitution itself is not changed.*

Rationale:

This guardrail allows for the possibility that the guardrails script (proposal policy) needs to be updated during the interim period, for example, because it has issues that need to be fixed. The guardrail applies to a **New constitution action** where either only a guardrails script is supplied, or both the guardrails script and an identical interim constitution is supplied.

Changes:

This guardrail is no longer relevant after the interim period has completed.

15. Summary and Conclusions

This document has provided the rationale underlying the guardrails on the governance actions that were defined in the interim constitution, including how and when changes to the guardrails can sensibly be made. In total, there are 165 guardrails governing 7 different types of governance action. 91 of the guardrails (over half) can be checked automatically using the on-chain proposal submission script (guardrails script), while a further 19 could be checked in future, if e.g. it was possible to access existing parameter values. The remainder generally require off-chain action (e.g. determining time scales or off-chain actions). 5 of these guardrails apply only to the interim period. Overall, this represents a major step towards robust programmable governance for Cardano.

Appendix A: Glossary

Term	Description
SPO	Stake Pool Operator, who runs a Cardano block-producing node
DRep	Delegated Representative, who votes on governance actions on behalf of themselves and/or other Ada holders
Delegator	An Ada holder, who has delegated their stake to a stake pool (for block production), and/or to a DRep (for governance)
TCP	Transmission Control Protocol - standard and widely used network protocol for internet communication
MTU	Maximum Transmission Unit - the size of the largest item that can be transmitted in a single network transaction
Interim Period	The period of time during which the interim Cardano constitution is in force.
Guardrail	A condition that is used to determine whether an on-chain governance action is/is not constitutional.
Guardrails Script	The Plutus script that is recorded on chain with the constitution, and that enforces the automatable guardrails when a governance action is submitted on chain
Low Cost Attack	An attack that is deemed to require too little Ada, effort, or other resource to execute. Acceptable values for these metrics need to be determined.

Appendix B: Rules Enforced by CDDL/Ledger

This appendix shows where the limit settings that are enforced by the CDDL format or in the ledger overlap or extend the guardrails Where guardrails exist, they are automatically enforced by the guardrails script, which may be additional to CDDL/ledger enforcement.

Guard-rail	CDDL Rule	Ledger Rule	Description
Y	Y		txFeePerByte must not be negative
Y	Y		txFeeFixed must not be negative
Y			utxoCostPerByte must not be 0
Y	Y		utxoCostPerByte must not be negative
Y	Y		stakeAddressDeposit must not be negative
Y	Y		stakePoolDeposit must not be negative
Y	Y		minPoolCost must not be negative
Y	Y		treasuryCut must not be negative
Y	Y		treasuryCut must not exceed 1.0 (100%)
Y	Y		monetaryExpansion must not be negative
Y	Y		minFeeRefScriptCoinsPerByte must not be negative
Y	Y		maxTxSize must not be negative
Y	Y		maxTxExecutionUnits[memory] must not be negative
Y	Y		maxBlockExecutionUnits[memory] must not be negative
Y	Y		maxTxExecutionUnits[steps] must not be negative
Y	Y		maxBlockExecutionUnits[steps] must not be negative
Y	Y		maxBlockHeaderSize must not be negative
Y	Y		stakePoolTargetNum must not be negative
Y			stakePoolTargetNum must not be zero

Guard-rail	CDDL Rule	Ledger Rule	Description
Y	Y		poolPledgeInfluence must not be negative
Y	Y		poolRetireMaxEpoch must not be negative
Y	Y		collateralPercentage must not be negative
Y		Y	collateralPercentage must not be zero
Y	Y		maxValueSize must not be negative
Y	Y		govDeposit must not be negative
Y	Y		dRepDeposit must not be negative
Y	Y		dRepActivity must not be negative
Y		Y	committeeMaxTermLength must not be zero
Y	Y		committeeMaxTermLength must not be negative
Y	Y		committeeMinSize must not be negative
	Y		maxBlockBodySize must not be negative
		Y	maxBlockBodySize must not be zero
		Y	maxBlockHeaderSize must not be zero
		Y	maxTxSize must not be zero
	Y		stakePoolDeposit must not be negative
	Y		stakePoolDeposit must not be zero
	Y		maxValueSize must not be zero
		Y	govDeposit must not be zero
		Y	dRepDeposit must not be zero
	Y		monetaryExpansion must not exceed 1.0 (100%)
Y		Y	All governance thresholds must not exceed 100%

Appendix C: List of Protocol Parameter Groups

Economic	***txFeePerByte***	*minimum fee coefficient*
	txFeeFixed	*minimum fee constant*
	utxoCostPerByte	*minimum Lovelace deposit per byte of serialized UTxO*
	minFeeRefScriptCoins PerByte	*minimum fee per byte for reference scripts*
	stakeAddressDeposit	*delegation key Lovelace deposit*
	stakePoolDeposit	*pool registration Lovelace deposit*
	minPoolCost	*minimum fixed rewards cut for pools*
	treasuryCut	*percentage of fees and monetary expansion taken by the treasury*
	monetaryExpansion	*monetary expansion rate*
	executionUnitPrices [priceSteps/priceMemory]	*Plutus Script Execution Prices*
	minFeeRefScriptCoins PerByte	*minimum fee per byte for reference scripts*

Network	***maxBlockBodySize***	*maximum block body size*
	maxTxSize	*maximum transaction size*
	maxTxExecutionUnits [steps/memory]	*maximum script execution units in a single* ***transaction***
	maxBlockExecutionUnits [steps/memory]	*maximum script execution units in a single block*
	maxBlockHeaderSize	*maximum block header size*

Technical/ Security	***stakePoolTargetNum***	*desired number of pools*
	poolPledgeInfluence	*pool pledge influence*
	poolRetireMaxEpoch	*pool retirement maximum epoch*
	collateralPercentage	*proportion of collateral needed for scripts*
	maxCollateralInputs	*maximum number of collateral inputs*
	maxValueSize	*maximum size of a serialized asset value*
	costModels	*Plutus Cost Models*

Governance	***govActionDeposit***	*governance action deposit*
	dRepDeposit	*DRep deposit amount*
	dRepActivity	*DRep activity period in epochs*
	dRepVotingThresholds[...], poolVotingThresholds[...]*	*governance voting thresholds*
	govActionLifetime	*governance action maximum lifetime in epochs*
	committeeMaxTermLength	*maximum term length (in epochs) for the constitutional committee members*
	committeeMinSize	*minimum constitutional committee size*

Appendix D: List of Critical Protocol Parameters

Critical to Blockchain Operation	***maxBlockBodySize***	*maximum block body size*
	maxTxSize	*maximum transaction size*
	maxBlockHeaderSize	*maximum block header size*
	maxValueSize	*maximum size of a serialized asset value*
	maxBlockExecutionUnits [steps/memory]	*maximum script execution units in a single block*
	txFeePerByte	*minimum fee coefficient*
	txFeeFixed	*minimum fee constant*
	minFeeRefScriptCoinsPerByte	*minimum fee per byte for reference scripts*
	utxoCostPerByte	*minimum Lovelace deposit per byte of serialized UTxO*
	govActionDeposit	*governance action deposit*

Critical to Governance	***stakeAddressDeposit***	*delegation key Lovelace deposit*
	stakePoolDeposit	*pool registration Lovelace deposit*
	minPoolCost	*minimum fixed rewards cut for pools*
	dRepDeposit	*DRep deposit amount*
	committeeMinSize	*minimum constitutional committee size*
	committeeMaxTermLength	*maximum term length (in epochs) for the constitutional committee members*

Appendix E: List of Non-Updatable Protocol Parameters

This list is taken from CIP-9. Non-updatable parameters cannot be changed by governance actions.

Parameter	Initial Value	Description
activeSlotsCoeff	0.05	The fraction of the total number of slots that will, on average, be selected to include a block in the chain. Smaller numbers increase security, but reduce efficiency.
genDelegs	...	Details of the public keys that have been selected by each of the genesis keys to act as a delegate for signing protocol updates etc. Superseded by CIP-1694.
updateQuorum	5	How many of the genesis delegate keys must endorse an update proposal. Superseded by CIP-1694.
networkId	"Mainnet"	Is this a testnet or mainnet
initialFunds	{}	initial distribution of funds to addresses.
maxLovelaceSupply	45000000000000000	The limit on the maximum number of lovelace that can be in circulation.
networkMagic	764824073	A magic number used to distinguish different networks.
epochLength	432000	The number of slots in an epoch.
SystemStart	"2017-09-23T21:44:51Z"	When did the system originally start operation.
slotsPerKESPeriod	129600	After how many slots will a pool's operational key pair evolve (Key Evolving Signatures).

slotLength	1	The length of each slot (in seconds).
maxKESEvolutions	62	What is the maximum number of times a KES key pair can evolve before a new KES key pair must be generated from the master keys.
securityParam	2160	After how many blocks is the blockchain considered to be final, and thus can no longer be rolled back (i.e. what is the maximum allowable length of any chain fork).