IIP 0001

ION Improvement Proposal

IIP: 0001
Title: IIP Purpose and Guidelines
Author:
Status: Draft
Type: Process
Created: 01-06-2018

What is an IIP?

IIP stands for ION Improvement Proposal. An IIP is a design document providing information to the ION community, or describing a new feature for ION or its processes or environment. The IIP should provide a concise technical specification of the feature and a rationale for the feature.

We intend IIPs to be the primary mechanisms for proposing new features, for collecting community input on an issue, and for documenting the design decisions that have gone into ION. The IIP author is responsible for building consensus within the community and documenting dissenting opinions.

Because the IIPs are maintained as text files in a versioned repository, their revision history is the historical record of the feature proposal.

IIP Types

There are three kinds of IIP:

- A **Standards Track** IIP describes any change that affects most or all ION implementations, such as a change to the network protocol, a change in block or transaction validity rules, or any change or addition that affects the interoperability of applications using ION.

- An **Informational** IIP describes a ION design issue, or provides general guidelines or information to the ION community, but does not propose a new feature. Informational IIPs do not necessarily represent a ION community consensus or recommendation, so users and implementers are free to ignore Informational IIPs or follow their advice.

- A **Process** IIP describes a process surrounding ION, or proposes a change to (or an event in) a process. Process IIPs are like Standards Track IIPs but apply to areas other than the ION protocol itself. They may propose an implementation, but not to ION's codebase; they often require community consensus; unlike Informational IIPs, they are more than recommendations, and users are typically not free to ignore them. Examples

include procedures, guidelines, changes to the decision-making process, and changes to the tools or environment used in ION development. Any meta-IIP is also considered a Process IIP.

IIP Workflow

The IIP process begins with a new idea for ION. Each potential IIP must have a champion -- someone who writes the IIP using the style and format described below, shepherds the discussions in the appropriate forums, and attempts to build community consensus around the idea. The IIP champion (a.k.a. Author) should first attempt to ascertain whether the idea is IIP-able. Posting to the ION-dev@lists.linuxfoundation.org mailing list (and maybe the Development & Technical Discussion forum) is the best way to go about this.

Vetting an idea publicly before going as far as writing an IIP is meant to save both the potential author and the wider community time. Many ideas have been brought forward for changing ION that have been rejected for various reasons. Asking the ION community first if an idea is original helps prevent too much time being spent on something that is guaranteed to be rejected based on prior discussions (searching the internet does not always do the trick). It also helps to make sure the idea is applicable to the entire community and not just the author. Just because an idea sounds good to the author does not mean it will work for most people in most areas where ION is used. Small enhancements or patches often don't need standardisation between multiple projects; these don't need an IIP and should be injected into the relevant ION development workflow with a patch submission to the applicable ION issue tracker.

Once the champion has asked the ION community as to whether an idea has any chance of acceptance, a draft IIP should be presented to the ION-dev mailing list. This gives the author a chance to flesh out the draft IIP to make it properly formatted, of high quality, and to address additional concerns about the proposal. Following a discussion, the proposal should be sent to the ION-dev list and the IIP editor with the draft IIP. This draft must be written in IIP style as described below, else it will be sent back without further regard until proper formatting rules are followed.

IIP authors are responsible for collecting community feedback on both the initial idea and the IIP before submitting it for review. However, wherever possible, long open-ended discussions on public mailing lists should be avoided. Strategies to keep the discussions efficient include: setting up a separate SIG mailing list for the topic, having the IIP author accept private comments in the early design phases, setting up a wiki page or git repository, etc. IIP authors should use their discretion here.

It is highly recommended that a single IIP contain a single key proposal or new idea. The more focused the IIP, the more successful it tends to be. If in doubt, split your IIP into several well-focused ones.

The IIP editors assign IIP numbers and change their status. Please send all IIP-related email to the IIP editor, which is listed under IIP Editors below. Also see IIP Editor Responsibilities & Workflow. The IIP editor reserves the right to reject IIP proposals if they appear too unfocused or too broad.

Authors MUST NOT self assign IIP numbers, but should use an alias such as "IIP-johndoe-tokens" which includes the author's name/nick and the IIP subject.

If the IIP editor approves, he will assign the IIP a number, label it as Standards Track, Informational, or Process, give it status "Draft", and add it to the IIPs git repository. The IIP editor will not unreasonably deny an IIP. Reasons for denying IIP status include duplication of effort, disregard for formatting rules, being too unfocused or too broad, being technically unsound, not providing proper motivation or addressing backwards compatibility, or not in keeping with the ION philosophy. For an IIP to be accepted it must meet certain minimum criteria. It must be a clear and complete description of the proposed enhancement. The enhancement must represent a net improvement. The proposed implementation, if applicable, must be solid and must not complicate the protocol unduly.

The IIP author may update the Draft as necessary in the git repository. Updates to drafts may also be submitted by the author as pull requests.

Standards Track IIPs consist of two parts, a design document and a reference implementation. The IIP should be reviewed and accepted before a reference implementation is begun, unless a reference implementation will aid people in studying the IIP. Standards Track IIPs must include an implementation -- in the form of code, a patch, or a URL to same -- before it can be considered Final.

Once an IIP has been accepted, the reference implementation must be completed. When the reference implementation is complete and accepted by the community, the status will be changed to "Final".

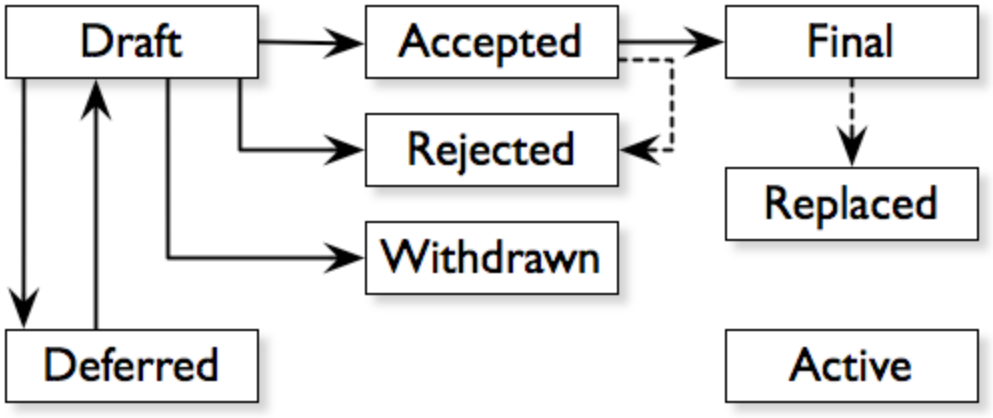
A IIP can also be assigned status "Deferred". The IIP author or editor can assign the IIP this status when no progress is being made on the IIP. Once an IIP is deferred, the IIP editor can re-assign it to draft status.

A IIP can also be "Rejected". Perhaps after all is said and done it was not a good idea. It is still important to have a record of this fact.

IIPs can also be superseded by a different IIP, rendering the original obsolete. This is intended for Informational IIPs, where version 2 of an API can replace version 1.

The possible paths of the status of IIPs are as follows:

ION Improvement Proposal



Some Informational and Process IIPs may also have a status of "Active" if they are never meant to be completed. E.g. IIP 1 (this IIP).

What belongs in a successful IIP?

Each IIP should have the following parts:

- **Abstract** — a short (~200 word) description of the technical issue being addressed.
- Copyright/public domain — Each IIP must either be explicitly labelled as placed in the public domain (see this IIP as an example) or licensed under the Open Publication License.
- **Specification** — The technical specification should describe the syntax and semantics of any new feature. The specification should be detailed enough to allow competing, interoperable implementations for any of the current ION platforms (IONJ, ION-js).
- **Motivation** — The motivation is critical for IIPs that want to change the ION protocol. It should clearly explain why the existing protocol specification is inadequate to address the problem that the IIP solves. IIP submissions without sufficient motivation may be rejected outright.
- **Rationale** — The rationale fleshes out the specification by describing what motivated the design and why particular design decisions were made. It should describe alternate designs that were considered and related work, e.g. how the feature is supported in other languages.
    - The rationale should provide evidence of consensus within the community and discuss important objections or concerns raised during discussion.
- **Backwards Compatibility** — All IIPs that introduce backwards incompatibilities must include a section describing these incompatibilities and their severity. The IIP must explain how the author proposes to deal with these incompatibilities. IIP submissions without a sufficient backwards compatibility treatise may be rejected outright.
- **Reference Implementation** — The reference implementation must be completed before any IIP is given status "Final", but it need not be completed before the IIP is accepted. It is better to finish the specification and rationale first and reach consensus on it before writing code. The final implementation must include test code and documentation appropriate for the ION protocol.

ION Improvement Proposal

Motivation

###

Changelog

IIP 0002

ION Improvement Proposal

IIP: 0002
Title: Secondary tokens
Author: Adam Mattlack and Michael Pfeiffer
Status: Draft
Type: Process
Created: 01-06-2018

Abstract

This IIP introduces the capacity to add **secondary tokens** to the ION blockchain.

Plain language tl;dr

“Add secondary tokens to the ION blockchain” — Introduce the capacity to create secondary tokens on the ION blockchain.

- ***Effect*****:** Token creation functionality facilitates new use cases for ION.
- ***Effect*****:** ionomy.com assets and gaming assets can be transacted and accounted for on the blockchain.
- ***Effect*****:** Masternodes facilitate network scaling by offloading token management from the core ION blockchain. Fees for token transactions performed by the masternodes can be distributed to ION masternode addresses and/or secondary token addresses.

Motivation

- *Tokens*
    - Introducing secondary tokens on the ION blockchain would add functionality to the blockchain that increase the utility and scalability of the ION network, including an increased functional role for masternodes.

Specification

Tokens

This proposal would allow for secondary tokens to be built on the ION blockchain and managed in the ION wallet.

IIP 0003

ION Improvement Proposal

IIP: 0003
Title: Revised block reward schedule
Author: Adam Mattlack and Michael Pfeiffer
Status: Draft
Type: Standards Track
Created: 01-06-2018

Abstract

This IIP introduces a **revised the block reward schedule**.

Plain language tl;dr

“Revised block reward schedule” — The ION blockchain will issue higher block rewards over a longer period of time.

- ***Effect*:** Higher block rewards are designed to incentivize the direction of computing power to maintain secure blockchain protocols. This is an improvement over a protocol that drops off precipitously leaving insufficient predictable motivation to maintain a growing ecosystem that includes not only ION, but all secondary tokens on the core blockchain.

Motivation

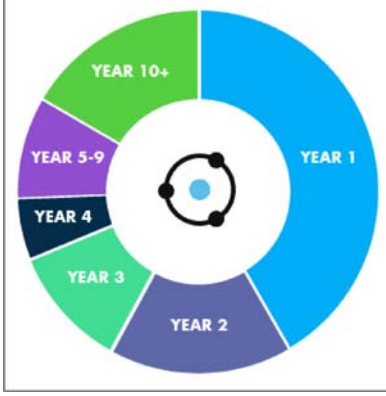
- *Revised block reward schedule*
    - This proposal adjusts the block rewards schedule so that the decline in rewards would be more gradual. Retaining higher rewards in the coming years increases the likelihood of adequate incentivization for maintaining network security, not only for ION, but also for all secondary tokens, thus protecting the viability of the ecosystems that depend upon the ION blockchain.

Specification

Block Reward Schedule

ION Improvement Proposal

The ION White Paper (2016) specifies the following block reward schedule:



Year	IONs/Block	IONs//Year	Total
1	23	12,000,000	22,900,000
2	17	9,000,000	31,900,000
3	11.5	6,000,000	37,900,000
4	5.75	3,000,000	40,900,000
5-9	1.85	1,000,000	45,900,000
10-100	0.2	100,000	55,000,000

White Paper Specification

The following chart shows a problem with the white paper specification. Notably, the Block reward is scheduled to drop off precipitously from Year 4 (5.75 ION/block) to Year 5 (1.85 ION/block). The effect potentially discourages masternode operation. Decay of the masternode network would erode the utility of the ION network, and disrupt economies built around secondary tokens.

Year	ION/Block	Issuance/Year	Total Projected (WP spec)	Inflation Rate (%) (WP spec)	Total Realized + Projected Realized (4/25)	Amount behind spec
1	23	12,000,000	22,900,000	177%	19,300,000	3,600,000
2	17	9,000,000	31,900,000	39.3%	27,000,000	4,900,000
3	11.5	6,000,000	37,900,000	18.8%		
4	5.75	3,000,000	40,900,000	7.9%		
5	1.85	1,000,000	41,900,000	2.44%		Too steep
6	1.85	1,000,000	42,900,000	2.38%		
7	1.85	1,000,000	43,900,000	2.33%		
8	1.85	1,000,000	44,900,00	2.27%		
9	1.85	1,000,000	45,900,000	2.22%		
10	0.2	100,000	46,000,000	0.22%		Too steep
11-100	0.2			0.22 - 0.18%		

ION Improvement Proposal

Proposed Block Reward Schedule

The proposed block reward schedule holds steady the number of new ION generated from Years 4-9. During this time period, while the block reward is static, the Inflation Rate nevertheless declines because the proportion of new coins relative to the total supply diminishes.

For the purposes of this document, “inflation” is treated as “supply inflation” and is calculated as the number of new ION generated in a year relative to the total existing supply at the beginning of the year.

Proposed			White Paper Reference		Realized + Proposed		
Year	ION/ Block	Issuance /Year	Total Projected (White Paper spec)	Inflation Rate (%) (WP spec)	Total Realized + Proposed Realized	Issuance /Year Realized + Projected Realized	Realized/Proposed Inflation Rate
1	23	12,000,000	22,900,000	110%	19,300,000	7,400,000	67.89% Realized
2	17	9,000,000	31,900,000	39.3%	27,000,000	8,300,000	43.01% Realized
3	11.5	6,000,000	37,900,000	18.8%	33,000,000	6,000,000	22.22% Projected
4	5.75	3,000,000*	40,900,000	7.9%	36,000,000	3,000,000*	9.09% Projected
5	5.75	3,000,000*	41,900,000	2.44%	39,000,000	3,000,000*	8.33% Projected
6	5.75	3,000,000*	42,900,000	2.38%	42,000,000	3,000,000*	7.69% Projected
7	5.75	3,000,000*	43,900,000	2.33%	45,000,000	3,000,000*	7.14% Projected
8	5.75	3,000,000*	44,900,00	2.27%	48,000,000	3,000,000*	6.66% Projected
9	5.75	3,000,000*	45,900,000	2.22%	51,000,000	3,000,000*	6.25% Projected
10	1.90	1,000,000	56,000,000**	1.8%	52,000,000	1,000,000*	1.96% Projected
11-100	0.2	100,000	65,000,000*	1.8%	61,000,000	100,000	0.019% Projected to 0.016% Projected

This proposal calls for the issuance of an additional 6 million ION over the currently realized supply. The currently realized supply is about 5 million ION lower than the white paper specification. The proposed adjustment achieves the goal of incentivizing a robust masternode network without substantially increasing supply inflation.

This chart does not address the potential earnings from fees generated from the ION network and from the proposed token network. Significant transactional velocity could mean that masternode operation would be sufficiently profitable from fees alone that network generation of new ION may no longer be necessary to incentivize masternode operation.

While this proposal addresses the drop off in block rewards from Year 4 to Year 5, it does not address the drop off from Year 9 (1.85 ION/block) to Year 10 (0.2 ION/block) which is also precipitous. The proposal defers these decisions to a future vote by the ION governance once there is sufficient information about the transactional velocity of the network to estimate the incentives necessary to maintain network security and utility.

IIP 0004

ION Improvement Proposal 0004

IIP: 0004
Title: Timelock-based block reward distribution
Author: Adam Mattlack and Michael Pfeiffer
Status: Draft
Type: Standards Track
Created: 01-06-2018

Abstract

This IIP introduces **timelock-based block reward structure**.

Plain language tl;dr

“Timelock-based block reward structure” — The ION blockchain will issue higher block rewards for a longer time commitment of masternode operation.

- ***Effect*:** Higher block rewards are designed to incentivize the direction of computing power to maintain secure blockchain protocols. This is an improvement over a protocol that drops off precipitously leaving insufficient predictable motivation to maintain a growing ecosystem that includes not only ION, but all secondary tokens on the core blockchain.

Motivation

- *Timelock-based block reward structure*
    - A proposal to distribute a progressively higher percentage of block rewards to those masternode operators that commit to locking the collateral output transaction for relatively longer periods of time.

ION Improvement Proposal

Specification

Timelock-based masternode reward structure

Days Locked	0	30	90	120	240	360
Masternode block reward share	50%	55%	60%	65%	70%	75%
Staking wallet block reward share	50%	45%	40%	35%	30%	25%

Masternode collateral inputs will be timelock transactions. Masternode operators will be able to select the timelock period according to the schedule above. Masternode operators committing to operate masternodes for longer time periods will receive larger percentages of the block reward.

IIP 0005

ION Improvement Proposal 0005

IIP: 0005
Title: Masternode collateral halving
Author: Adam Mattlack and Michael Pfeiffer
Status: Draft
Type: Standards track
Created: 01-06-2018

Abstract

This IIP **halves the masternode collateral requirement**.

Plain language tl;dr

“Halve masternode collateral requirement” — 10,000 ION required to operate a masternode instead of 20,000 ION.

- ***Effect:*** By lowering the barrier to network participation through a reduction in cost of masternode acquisition: ION adoption, distribution, governance, ecosystem diversification, and most important, the strength of the ION network, are advanced.

Motivation

- *Halve masternode collateral requirement*
    - By halving the masternode requirement, the barrier to network participation is reduced.
    - A robust, extensive, and scalable masternode network is required to sustain the projected growth in demand for secondary token transactions. Lowering the collateral requirement increases the number of available masternodes in the network.
    - Permeable network participation fuels the generation of new use cases for the ION blockchain, and decentralized governance has more checks and balances when distributed among more — and more diverse — membership.
    - ION was developed not simply to be a protocol, but a protocol with uses and users. ION was envision not just as a cryptocurrency, but as a component in a thriving economy. Masternodes function as an incentive system for people and businesses who are interested in investing in the ION ecosystem by increasing its utility and thereby its value to other users.

- To this end, a gradually declining barrier to participation, structurally pegged to the rate of gradually declining rate of coin generation, should be considered in the future and could be designed to motivate new network participation and innovation dynamically. Such a proposal may synchronize collateral requirements with the block reward schedule. As block rewards decline according to the graduated schedule, the collateral requirement to operate a masternode could also decline. Until such a proposal can be specified, this halving proposal makes an important temporary adjustment.
- The proposed design would be an improvement over the white paper specification because a rising market price of ION would put network participation and masternode operation out of reach of most.

Specification

Masternode collateral requirement halving

Year	Collateral base	adjustment	Adjusted Collateral Requirement
3	20,000	-50%	10,000

This proposal will change the masternode collateral requirement from 20,000 to 10,000 ION. Implementation will require a hard fork.