Commercially-Focused Ethereum Scaling

Mark Briscombe, mark@hubii.com
Jens Ivar Jørdre, jensivar@hubii.com
Morten Fjeldstad, morten@hubii.com
John Derbyshire, john@hubii.com
Jacobo Toll-Messia, jacobo@hubii.com

September 20, 2018

Contents

1 Preface

At the time of writing, hubii's content business handles millions of API calls per day, covering our partners' needs and delivering content to over 50 million people globally. Blockchain provides some unique features that can help us to enhance our current value proposition. Unfortunately, it is not yet capable of meeting basic commercial needs. It is for this reason why everyone working in this space is so focused on scaling.

We chose to build upon Ethereum for many reasons, but it does not scale to meet our current business needs. hubii's business needs a protocol that can potentially process millions of payments and trades per second. Yet, we need it to be completely trustless. The need for trustlessness is born from the nature of blockchain itself; trusting third-party products built upon blockchain would come with increased counterparty and, therefore, commercial risk.

After careful evaluation of the current maturity level of other scaling initiatives, we decided to pause our ongoing content-based work and embark ourselves on a short, yet intense, journey of contributing to this amazing wider ecosystem. 6 months later we give you **nahmii**.

Our track record working with leading companies places us in a unique position to promote the mass adoption of nahmii through our existing partnerships and deployment of our upcoming ecosystem of products.

The intention of this document is to be an approachable white paper, where we set out the design and operating principles of nahmii in non-technical language. Alongside this white paper we will be releasing the first deployment to the public Ethereum testnet for review, comment, testing and bug-finding. The product itself will constitute the detailed technical documentation.

Please note that we do not discuss every security provision and attack vector prevented by the architecture in this document, instead some very basic examples are provided which illustrate the key concepts.

2 Partner Words

> “The nahmii system addresses a real-world use case: incentivization of producing information goods. For example, the Bugmark project addresses market failures in software incentivization by trading futures contracts on the status of issues in a software issue tracker. This enables not only developers, but also code reviewers, testers, and managers, to overcome market failures that have historically resulted in software being produced at a quality level below that desired by either developers or users.
>
> In order for futures contracts to fulfill their promise as a new form of software incentivization, we require
>
> - low transaction costs, especially across jurisdictions (futures contracts are a way to trade information goods, not compensate labor, so can have extremely low transaction costs if the platform supports it)
> - ease of deposit and withdrawal, even of small amounts (such as would be generated by bug triage, code reviews, or other low-overhead 'meta' tasks)
> - quick results for typical requests, in order to facilitate what might be multiple trades per issue for large numbers of related issues.
>
> Where both conventional payment platforms and existing cryptocurrencies can satisfy some of our requirements, it appears that nahmii can address all of them and help facilitate a new type of market for software quality incentivization.”

Don Marti, Strategist for Mozilla and advisory board member for Incentives Research

> “Bugmark needs scalable payment-processing for digital currency. We're 100% confident in the nahmii team and their solution fixes Ethereum's show-stopping scalability issues. Any business who needs to transact in digital currencies should take a serious look at nahmii.”

Andrew Leak, Lead Developer, Bugmark

3 Introduction to nahmii

3.1 What is nahmii?

nahmii is a second layer scaling solution for the Ethereum blockchain. Second layer solutions are designed to handle much greater transaction volumes than the main Ethereum network; they do this by moving the majority of the processing off-chain. These off-chain transactions are then enforced by the security constructions on the Ethereum main chain, which acts as the arbitrator for all disputes.

All transactions using nahmii will be processed initially by hubii, who will act as the operator of the network. Importantly, the security of the nahmii network does not rely on users trusting hubii; the system has been designed to work in a trustless manner.

3.1.1 driips

Upon release there will be two forms of transactions that are possible on the nahmii protocol, payments and trades, supporting Ether and ERC20 tokens. All forms of off-chain transactions that can result in a change of state on the Ethereum main chain will be termed *driips* in this paper, for ease of differentiation from Ethereum transactions.

3.1.2 nahmii Tokens (NII)

nahmii will be a tokenised protocol by necessity, yet there will be no token sale. 100% of revenue generated by fees from using the protocol will fund the security mechanisms, which includes rewarding the token holders who play an essential role in monitoring, validating and actively protecting the protocol.

3.1.3 hubii Tokens (HBT)

HBT is the native token of the hubii ecosystem, which was created in September 2017. It will be the liquid currency for hubii's content business and its function remains unchanged by the introduction of nahmii

3.2 Foundation Model

nahmii will be governed in accordance with a foundation model, whereby members are responsible for the efficient and reasonable management of the protocol. All members, hubii included, will be equal partners with the same associated rights, privileges and responsibilities. Members will be geographically distributed and leading companies across diverse industries, ensuring that the Foundation itself will be decentralised in nature.

Members are required to play an active role in monitoring, validating and protecting the network. Their role also includes a commitment to building nahmii-based solutions and the provision of nodes to assist protocol operation. The Foundation will be required to ensure the responsible divestment of hubii's large holding at an appropriate point in time. This divestment is part of a broader plan to guarantee a plurality of nahmii token holders, thus making the network more resistant to a 51% attack.

4 nahmii Fundamentals

4.1 State Channels

nahmii can be thought of as a multi-party state channel. A state channel is the name given to one general purpose off-chain scaling solution for Ethereum; it is essentially the exchanging of signed messages between users off-chain, which allows for eventual on-chain settling of the final state at a later point.

This is best explained by way of an example: Alice wants to pay Bob 1 token per tweet for $1{,}000$ tweets. Using Ethereum, she could theoretically send $1{,}000$ on-chain transactions to Bob of 1 token each time he tweeted. However, if everyone did this, the Ethereum network would rapidly become congested, transaction fees would rise and Alice's total cost to send those tokens would be punitively high. We can improve this situation by using state channels. In the simple case whereby Alice and Bob both trust each other, Alice could send an email containing a signed Ethereum transaction for 1 token after tweet 1 and a second email with a 2 token transaction after tweet 2. This would go on until she sent a $1{,}000$ token signed transaction after tweet $1{,}000$, at which point Bob could then send the transaction from the last email to the network and receive $1{,}000$ tokens. This example is less relevant for nahmii as Alice and Bob trust each other; here, Alice could simply send the payment in advance of the tweets, trusting that Bob would deliver on his side of the bargain. Regardless, this example serves to highlight some of the major issues which would need to be solved in the case of Alice and Bob mistakenly trusting one another, they include:

- Bob sending all of the transactions he received from Alice to the network. In this case, he can potentially steal $1{,}000 + 999 + 998 + \ldots + 3 + 2 + 1 = 500{,}500$ tokens from Alice, assuming her balance was high enough
- Alice waiting until Bob has sent $1{,}000$ tweets before moving her tokens to another address. When Bob tries to claim his $1{,}000$ tokens by sending the final signed transaction to the network, there are no funds in Alice's account to pay him. Alice is effectively getting $1{,}000$ tweets for free

State channel constructions become progressively more complex as additional security provisions are added to protect users. Similarly, the move from unidirectional payments to bidirectional payments and, ultimately, more general state transitions increases the complexity of the system significantly.

4.2 Benefits of nahmii

There are a number of benefits to off-chain constructions or so-called 'second layer scaling solutions', described below. Understandably, there is a huge amount of excite-

ment and anticipation surrounding these kinds of projects.

4.2.1 Security

Whilst nahmii is an off-chain solution, it is architected such that it maintains security which can theoretically approach that of the Ethereum base layer. All attempts to defraud the network should lead to failure. The security model of nahmii is discussed in detail later in this document.

4.2.2 Transactional Throughput

Performance and throughput has been a major priority in the design and implementation of the nahmii backend. Best practices allowing massively scalable/web-scale deployments have been followed and compromises in performance have only been made where we feel security would otherwise be jeopardised. The resulting backend will conservatively be able to process 15 driips per second per user (address), *with no practical restriction on the number of potential addresses*. For driips that require serial processing across addresses (such as trades, which require the maintenance of an order book) we will be more limited, but equally as performant as centralised exchanges. nahmii can therefore accommodate millions of users, handling millions of driips per second as required.

driips Per Second Per User

Note, we prefer to specify the transactional throughput of nahmii in terms of driips per second per user. We believe this is the ultimate metric for assessing a protocol, particularly when considering the capability of nahmii to handle trading, microtransactions or devices connected via the Internet of Things. As an example, consider microtransactions related to content, a user could not have a positive experience of browsing or viewing content if there were noticeable delays of seconds between actions.

Base Layer Scaling

Scaling of the Ethereum network base layer is complementary to constructions such as nahmii. Ethereum transactions are needed for deposits, disputes and withdrawals for the nahmii protocol. As such, as Ethereum scales, the on-ramps and off-ramps for nahmii gain capacity with the associated benefits of quicker and cheaper transactions.

4.2.3 Transactional Volume and Gas

In ordinary operation, nahmii will only require transactions to be submitted to the Ethereum network for deposits and withdrawals from the system. Once a user has

deposited into nahmii that user may make essentially infinite driips within the platform, only needing to send further transactions to the Ethereum network to either top up their account balance or to withdraw some funds back to the main Ethereum network. Despite the potential additional gas overhead for depositing and withdrawing on-chain, nahmii users will benefit from significant cost savings after just a few payments or trades.

The on-chain dispute mechanisms within nahmii also incur additional transaction volume and gas costs. This is a small price to pay for the associated security benefits that these mechanisms provide; nahmii is designed to be a robust system where attackers are dissuaded by maximally significant penalties for failure and similarly high chances of detection. As such, any dispute overhead is minimal for the network relative to other security options.

4.2.4 Finality

Transaction finality is an essential component of any economic system. In an ideal scenario, the process of finalising transactions would be limited only by latency. In blockchain-based systems however, transaction finality is generally probabilistic; a Bitcoin transaction is considered final after 6 block confirmations, which are usually completed in around 60 minutes. For Ethereum the equivalent might be considered to be 12 block confirmations or just under 3 minutes.

The finality of driips within nahmii is *immediate*, i.e. transactions are final as soon as a signed execution receipt is published by the operator.

Finality is critical for an exchange which seeks to have tight spreads on currency pairs. Arbitrageurs need assurance that both sides of their trades are executed across whichever two exchanges they are using. If an exchange provides rapid guarantees that a trade has taken place and cannot be reversed, arbitrageurs are exposed to significantly reduced risks. The lack of these guarantees goes some way to explaining the significantly wider spreads on cryptocurrency exchanges compared with traditional forex markets. These risks are not fully mitigated by using a centralised exchange, which provides a receipt of trade execution, as execution guarantees are contingent upon successfully withdrawing from the exchange due to counterparty risk. The nahmii protocol is designed to avoid this counterparty risk.

Latency

For nahmii, driip finality is determined by latency in the system. nahmii's architecture ensures that a user's connection latency will have no impact on the execution of an or-

der and its driip finality once a signed order has been registered with the system. The main source of any delays in processing would therefore be the security and fraud detection checks that are performed as the order is registered and subsequently executed. It is important to ensure that the user's experience is optimised for instant feedback in any products built upon nahmii. This requires a robust backend architecture.

Of particular importance for users of an exchange is transparent order processing and management of their orders. As such, the user interface must give near real-time feedback about order placement and cancellation. Our off-chain order book must add minimal latency, as discussed earlier. We expect that 95% or more of all placed orders will eventually be cancelled by traders using nahmii; latency adds risk to traders, in particular arbitrageurs.

Storage and Bandwidth Requirements

There may be a concern about the possible storage requirements of nahmii once it achieves a high throughput. However, it is possible to trim settled driips from the live driip database. Once driips have fully settled they are effectively checkpointed.

The bandwidth requirements of nahmii, notably its data publishing element, will be in keeping with similarly scalable web applications. The architecture can also leverage enterprise grade cloud infrastructure if needed.

The architecture of nahmii ensures effective decentralisation, as opposed to requiring a set of decentralised nodes which are relied upon to maintain consensus about data. This approach is far better suited to real world use cases.

4.2.5 driip Volume Limits

The nahmii architecture places no arbitrary limits upon the amount of tokens in a driip. As such, payments and trades will not normally be limited in size.

From Micropayments to Picotransactions

Micropayments have long been hailed as the solution to the problem of content monetisation, however the underlying technology has failed to facilitate this vision. Payments made in fiat currencies are limited either by the minimum transaction size or the requirement for a third-party to aggregate smaller amounts. In contrast, true micropayments are made using tiny fractions of dollar, with minimal transaction fees. The nahmii protocol is able to handle extremely small driips such as *pico*transactions, depending on the token. Due to our highly efficient architecture, nahmii can provide the ideal platform for genuine micropayments: very small transactions with very low

fees, delivered trustlessly to content providers.

Fees are the determining factor for minimum driip amounts. For a fee of 0.1% and a typical ERC20 token with 15 decimal places, this minimum driip size is 1×10^{-12} tokens, or a picotoken. For Ether which has 18 decimal places the minimum driip size is 1×10^{-15} tokens or a femtotoken.

The ability to have value flow in this way opens up numerous new business models, many of which hubii will be taking advantage of immediately by leveraging our existing content business. Aside from content monetisation, genuine micropayments and the countless financial applications, this also means that nahmii can function as a protocol upon which IoT (Internet of Things) systems can be developed. Note, the minimum driip size may be modified in future to ensure the protocol is resistant to abuse.

4.2.6 Network Fees

There will be fees for transacting any type of driip on nahmii. These fees are essential for the security of the network and for building the ecosystem around the nahmii protocol. The fee structure has been designed so that the costs of using nahmii should be competitive with, if not considerably lower than, the fee for performing the equivalent transaction on the Ethereum base network. Our architecture is able to benefit from extremely cheap computation and should therefore be substantially more efficient in processing transactions. Furthermore, unlike with the Ethereum network, fees on nahmii will be predictable and consistent.

4.2.7 Account Flexibility

Ordinarily state channel constructions can introduce undesirable properties, such as restrictions on top-ups, movement of arbitrary token denominations or partial withdrawals. nahmii will have no such restrictions. Users may top-up with ease, transfer or trade any arbitrary amount of funds and make partial withdrawals as needed. Users are not required to close their nahmii account to settle their balance and withdraw.

4.2.8 Hot, Warm and Cold Wallets

With clever adaptation of the user interface, users will have the option to store funds in 'hot', 'warm' and 'cold' wallets within nahmii:

- A cold wallet is the most secure method of storing funds within nahmii and is the recommended solution for users with large balances. Such a wallet would usually be controlled by a hardware device, such as Trezor or Ledger Nano S

- Warm wallets require a password, code or similar passphrase in order to sign driips, with user's key pairs stored securely in an encrypted format on the device they are using. This model offers strong security, however it carries a higher risk than the cold wallet option
- A hot wallet can be used within the nahmii ecosystem and remain 'unlocked' upon entry to our products. Typically this not would be considered secure, however it can be acceptable for small sums of money (such as a micropayments wallet)

Due to nahmii's low latency, instant finality and ease of use, we can enable these features in a user friendly way which is not currently possible on the Ethereum network.

4.2.9 Upgrades

One of the key architectural principles behind nahmii is that the development of the protocol should keep pace with the needs of commercial use cases. This has been a particular problem for the blockchain ecosystem as a whole, leading to many competing 'blockchains' of dubious quality and purpose. It is our strong belief that that any governance of a blockchain should be introduced at the second layer. The base layer, in nahmii's case Ethereum, should remain untouched as a bastion for the key principles of blockchain technology: decentralisation, immutability and permissionless innovation.

hubii and the nahmii Foundation will ultimately be responsible for ensuring the security of nahmii during any future upgrades. In general, nahmii has been designed to be easily upgradable, however users may choose to opt-out of upgrades in a trustless fashion. Users will also have the subsequent option to opt-in again later.

5 nahmii Architecture

5.1 Security Construction

The nahmii security architecture divides into three levels, each containing a set of nodes. This separation of concerns is best understood as:

1. **Operator**

   The nahmii network will be operated initially by hubii, who will provide the first point of validation on the network. This arrangement is subject to the governance and approval of the nahmii Foundation. Much of nahmii's security construction is designed to ensure the operator is unable to commit fraud, even if compromised.

2. **User Monitoring**

   Any user of the Ethereum network can submit fraud challenges to the smart contract, with the reward for a successful challenge being the fraudulent user's balance. These challenges are explained in more detail in the 'Continuous Fraud Challenge' and 'driip Settlement Challenge' sections.

3. **Data Availability Validation**

   The fraud challenges detailed above require users to submit proof in the form of driip records. These records are published by the operator and nahmii token holders are responsible for validating that this data is available at all times. The 'Data Availability Validation' section provides more detail on these points.

The security provisions within the nahmii protocol are designed to protect against three possible fraudulent attacks, characterised in terms of data availability. First, users are protected against the possibility of a compromised network operator through the 'Continuous Fraud Challenge' mechanism. Second, users are protected against illegitimate driip rollbacks through the 'driip Settlement Challenge' mechanism. Both of these challenges require network data to be both available and accurate, hence the need for a third security provision for when data is not available. The fully decentralised 'Data Availability Oracle' is designed to continually test data availability, this is the third security provision.

This section sets out the three security provisions in detail, explaining the rationale behind each and how they work together to ensure the safe operation of the nahmii protocol.

5.1.1 Operator

The operator of the nahmii network will be hubii, with the option to decentralise this processing later with the support of the nahmii Foundation. Additionally, it is theoretically possible for other entities to run their own nahmii-based systems. As an example, an online gaming company may wish to be the operator for an in-game item distribution system powered by nahmii.

The security constructions within nahmii have been designed to protect user's funds in the event of a rogue or compromised operator. In the event that the network has been compromised, nahmii will simply close down gracefully and within minutes restart under a different set of suitably air-gapped smart contracts. Users can opt-out of this migration.

The security of the nahmii network is further enhanced by the foundation model of governance, which makes the possibility of a malevolent operator gaining overall control substantially less likely.

5.1.2 User Monitoring

Continuous Fraud Challenge

In order for nahmii to approve a potential driip, it must first be signed by both the user initiating the driip and the network operator using their respective private keys. The integrity of the network depends on the network operator only signing valid driips, hence the requirement for a security check to ensure that this is the case. This 'Continuous Fraud Challenge' is therefore designed to protect users from the possibility of a compromised or rogue network operator. Note, below we give one simple example of fraud, but the nahmii protocol must be able to handle all forms of fraudulent driip.

The twin sign off process is best illustrated by way of a simple example, a request by Alice (A) to make a payment of 100 tokens to Bob (B). First, Alice initiates the payment request and signs it with her private key to verify the driip . Next, the network operator (O) checks Alice's balance to ensure that the appropriate funds are present. Provided that Alice has a sufficient balance, the network operator signs the driip using their own private key. After performing this second check, the network operator decrements Alice's balance by 100 tokens and increases Bob's balance by the same amount. Finally, the network operator publishes the driip details on a publicly accessible website. The process can therefore be represented as:

1. A initiates a payment request to send 100 tokens to B
2. A signs the driip using her private key

3. O checks that A has sufficient balance for the payment (A does)
4. O signs the driip using their private key
5. O decrements A's balance by 100 tokens
6. O increases B's balance by 100 tokens
7. O publishes the driip details to a publicly accessible website

In this example, we have presumed that both A and O hold valid private keys and that O has not been compromised.. If O has been compromised however, this raises the prospect of invalid driips being signed off by the network. Alice in this case is assumed to be complicit in the fraud, as she has also signed the driip. Once again, this is best illustrated by an example:

1. A initiates a payment request to send 100 tokens to B
2. A signs the driip using her private key
3. O checks that A has sufficient balance for the payment (A does)
4. O signs the driip using their private key
5. **O increases A's balance by** 100 **tokens**
6. O increases B's balance by 100 tokens
7. O publishes the driip details to a publicly accessible website

Clearly something has gone wrong; A's balance should decrease by 100 tokens, not increase by the same amount. Unless the operator was compromised, A's invalid driip request would normally be rejected at step 3. If the operator incorrectly approves the fraudulent request, the only possible explanation is that O has been compromised and is working with A.

The key to identifying fraudulent driip of this type lies in step 7, where O publishes all driip data to a publicly accessible website. Based on this information, users of the nahmii network can challenge any driip by calling the appropriate function of the smart contract. All of the necessary information is publicly available to prove the fraudulent driip. Once a successful proof is submitted to the smart contract, the network is halted and the user who raised the challenge is awarded A's balance as a reward. Again, A is assumed to be colluding with O on the attack and so all users should take care to never sign an invalid driip in case the operator is also compromised at that time. Open

source tools, such as hubii's reference software to interact with nahmii, will ensure that the operator cannot deliberately trick users into signing invalid driips.

An example of a successful challenge by Carol, C, is:

1. A initiates a payment request to send 100 tokens to B
2. A signs the driip using her private key
3. O checks that A has sufficient balance for the payment (A does)
4. O signs the driip using their private key
5. **O increases A's balance by** 100 **tokens**
6. O increases B's balance by 100 tokens
7. O publishes the driip details to a publicly accessible website
8. C raises a challenge against the driip by calling the smart contract
9. C provides evidence of fraud using the data from step 7
10. Smart contract confirms fraud and stops the exchange
11. C is awarded A's funds as a reward

This challenge is not free, as calling the smart contract incurs a gas cost. This mitigates against an effective DoS (Denial of Service) attack against nahmii, as to do so the attacker must DoS attack the entire Ethereum network. If many challenges are raised in a short period of time, the cost of calling the smart contract would rise quickly to the point where a sustained attack would be uneconomical. Importantly, there is no cost for the network operator to permit these fraud challenges. This feature eliminates a further vulnerability whereby an attacker could repeatedly spam the network with challenges, each of which had an associated cost for the operator in an attempt to 'bankrupt' the network.

We anticipate that many nodes will be monitoring the published network data for anomalies, including those nodes run by the nahmii Foundation partners. Please refer to the earlier 'Foundation Model' section of this document for more information regarding these points.

driip Settlement Challenge

The second security provision within the nahmii network is designed to ensure that driips are settled such that a user's balance is brought up to date. This is critical to ensure users are unable to perform effective personal rollbacks which would otherwise undermine trust in the system. Note, this is a simplistic way to describe nahmii, as some flexibility has been added to the protocol without compromising security. Again, one simple example of an illegitimate driip settlement is provided below, but there will be a number of more complex possibilities that the nahmii protocol will be able to handle.

The need for an 'driip Settlement Challenge' is best understood in terms of the nahmii withdrawal process. nahmii requires that users 'settle' their account before withdrawing and the settlement process may include an intent to withdraw only a certain portion of a user's available funds.

Once a user has initiated the settlement process, the smart contract starts the 'dispute period' during which the request can be challenged. It is at this point that any nahmii user can challenge the request through the 'driip Settlement Challenge'. As with the explanation of the 'Continuous Fraud Challenge', this process is best understood by way of an example:

nahmii user Alice (A) has completed ten driips on the exchange and is yet to settle her account. The ninth driip shows that Alice had a balance of 100 HBT tokens and Alice requests to settle her account to this driip nonce. She also requests to withdraw 50 HBT tokens of her balance once her account is settled. Alice makes the appropriate request by calling the smart contract and providing the details of driip nine. The smart contract then begins the dispute period, during which an 'driip Settlement Challenge' is possible. A successful withdrawal request without a successful challenge therefore looks like this:

1. A begins the nahmii withdrawal process by requesting to settle her account
2. A calls the smart contract and provides both the details of the driip she wishes to settle up to and the balance she wishes to withdraw
3. The smart contract checks that the request is valid and begins the dispute period
4. No successful 'driip Settlement Challenge' is made during the dispute period
5. A's request is approved and the appropriate funds are moved to her withdrawable balance

6. A may now withdraw funds from this balance at any point

The alternative scenario is one in which A's withdrawal request is successfully challenged by C. In our example, A wanted to settle her account up to driip nine of ten at which point her balance was 100 HBT. If A's tenth driip was a payment sending 100 HBT tokens to B, her available balance *taking all ten driips into consideration* is 0 HBT. A's request to withdraw 50 HBT tokens is therefore fraudulent, as she does not have these funds available in her account. C may challenge this fraudulent request by providing the appropriate evidence to the smart contract as proof, namely the details of driip ten. A successful 'driip Settlement Challenge' rewards the challenger with the fraudulent user's balance. This is detailed below:

1. A begins the nahmii withdrawal process by requesting to settle her account
2. A calls the smart contract and provides both the details of the driip she wishes to settle to and the balance she wishes to withdraw
3. The smart contract checks that the request is valid and begins the dispute period
4. C raises an 'driip Settlement Challenge' by calling the appropriate smart contract function
5. C provides evidence of A's fraudulent request, namely the later driip showing the discrepancy in A's available balance
6. The 'driip Settlement Challenge' is successful and C receives A's balance as a reward

Users are required to maintain a minimum balance in order to use the nahmii network, but this can be implemented operator-side without adding any risk to users. In the unlikely event that the operator chose to set the minimum balance requirement at an excessively high level, users would still be able to exit and withdraw from nahmii.

The minimum balance requirement is designed to mitigate against the 'nothing at stake' problem, whereby users can effectively test the security of the network for free. If users can still transact with a nominal balance, the deterrent for attacking the network (namely, losing that balance) is insufficiently small to outweigh the potential gains from a successful attack. Similarly, the reward for challenging a fraudulent driip settlement (namely, claiming that balance) is an insufficiently great incentive for other users of the system who might otherwise raise a settlement challenge. Raising a settlement challenge is not free, it incurs a gas cost; the system therefore requires that the cost to raise a settlement challenge should always be lower than the potential reward, as otherwise there is no incentive for a rational actor to do so.

We recognise that the minimum balance requirement represents a minor inconvenience for users of nahmii, however the costs of imposing this requirement are more than outweighed by the associated security benefits. This highlights one of the fundamental security principles underpinning the nahmii architecture: any attempt to defraud the network must always be maximally risky for the attacker and accompanied by a sufficiently great cost if they fail. In this way, we can design a system which makes prolonged attacks unsustainably unaffordable and opportunistic attacks unattractive.

5.1.3 Data Availability Validation

The Data Availability Problem

The fraud and settlement challenges set out in this section rest on the principle of data availability, which requires that the network operator publish accurate and complete driip records at all times. This data is crucially important for ensuring confidence in the nahmii network and the network operator. Without the relevant data, users cannot challenge fraudulent driips as there is no evidence to send to the smart contract. Similarly, the off-chain elements of nahmii cannot be verified without the driip records; users cannot therefore be confident that their driips have been processed correctly by the network operator without access to the appropriate data.

nahmii users cannot simply presume that the network operator is trustworthy, they must be able to inspect the published data to be sure. As such, nahmii requires a decentralised method by which the smart contract can check that data is available. Furthermore, this must be secured against external manipulation.

The challenge described above is known as the 'data availability problem'. If users of a network need the network operator to publish driip data to check whether the operator is trustworthy, how can users ensure that the network operator is publishing the appropriate data? Our solution is the 'Data Availability Oracle', a fully decentralised mechanism by which distributed nahmii token holders are incentivised to monitor data availability.

Data Availability Oracle

The 'Data Availability Oracle' is designed to protect nahmii users from a potentially compromised network operator who is withholding data. As discussed, the 'Continuous Fraud Challenge' and 'driip Settlement Challenge' processes rely on users submitting proof of a fraudulent driip. This proof is taken from data published by the network operator, without which the challenges cannot function. In the unlikely event

that a compromised network operator chooses to publish fraudulent driip data, this attempted fraud is easily identifiable through the driip signatures, sums and values. Far more likely is the alternative scenario, whereby a compromised operator refuses to publish the relevant data (either by selectively excluding certain driips or simply withholding all data). In this case, the proof of the attempted fraud is the *absence* of data rather than the presence of tangible evidence. This requires a different kind of solution.

The Oracle is best understood as a function of the account settlement process within nahmii. Before a user can withdraw funds, they must first request to settle their account to a particular driip. The first step in the withdrawal process is therefore to check whether the user's driips up to the driip in question are all valid, this requires the user to call the smart contract and start the dispute timer. During this period, other nahmii users may challenge the request through either the 'Continuous Fraud Challenge' or 'driip Settlement Challenge' mechanisms. If the request is proven to be fraudulent, the user loses their funds. If the dispute period ends without a successful challenge, the user can then effectively reactivate the smart contract (which has been dormant during the dispute period). As the settlement request has not been proven fraudulent, the smart contract performs one final check: *is data available?* This is done by querying the Oracle.

In order for the Oracle to function, it must return a binary response when the smart contract checks whether data is available (yes/no, true/false etc.). The Oracle is a game theory-based distributed intelligence tool, adapted from the *hubii mind* product, which was first discussed in the hubii ICO white paper. It operates on the principle of a small reward for being a good actor and a severe punishment for being a bad actor. The purpose of the Oracle is to continually test several statements relating to data availability. These statements have only two truth conditions, true or false (i.e. no indeterminacy is possible), where the combination of these statements provides a similarly clear answer to the question 'is data available?'. The Oracle will only have the ability to effectively pause user settlement whilst it returns 'false'. This should also account for the possibility of a legitimate, temporary problem with data availability which is fixed later. If the Oracle subsequently returns the response 'true' again, the nahmii settlement process will be reactivated.

The questions around data availability necessarily include a temporal component, as data availability can change over time. This requirement relates to the staking mechanism at the heart of the nahmii Oracle, whereby nahmii token holders stake their tokens against 'true' or 'false' for each of the data availability statements. It is trivial to see why the monitoring market must include a temporal component. Without this time restriction, two users may stake their tokens on opposite outcomes and both be

correct. Consider the simple statement 'data is available', this can be both true at time t_1 and false at time t_2. It is therefore essential that the crucial monitoring questions are formulated correctly. In order for the status of a question to change between 'true' and 'false' then the staking of users must achieve a variety of criteria. It is insufficient for this system to be based on a simple honest majority assumption.

Users are incentivised to participate in this monitoring market by the promise of payment for being correct, with the optional introduction of additional incentives for staking early in the process if required. This is known as the 'Data Availability Bond'. This bond is accrued from nahmii network revenues and a portion is available as a reward for staking correctly. By only awarding a fraction of the bond as a reward for identifying when data availability changes, we ensure that there is always a reward for reversing any status change. Tokens of users who staked in opposition are seized and provide an additional reward; this is an essential punishment for being a bad actor.

The Oracle will function entirely on-chain as a true decentralised process, with no possibility of centralised interference. Therefore, the Oracle must be optimised over time and will still be in testing when initial testnet nahmii deployment takes place. It must be able to quickly and accurately resolve whether data is available, yet be Sybil and 51% attack resistant. This is a non-trivial requirement.

As part of the foundation model, discussed earlier in this paper, nahmii Foundation members will be responsible for overseeing the divestment of hubii's nahmii token holding at an appropriate time. This will ensure a plurality of token holders, minimising the risk of attack on the Oracle. Similarly, key Foundation members will be required to host replicate driip data. This will help mitigate some data availability false alarms.

5.2 Liquidity of Funds

When a user deposits into nahmii, their funds will ordinarily fund the liquidity pool. Upon settlement, withdrawals will be made from the same pool. The nahmii network has been designed so that users can only ever access the appropriate amount of funds in the liquidity pool; as such, nahmii is non-custodial and all funds are fully backed by user's deposits.

A reserve fund can be added to the nahmii architecture, however this is not strictly necessary for the network to function. The reserve fund would permit users to deposit into an individual account, rather than the liquidity pool. This fund would effectively sit between user accounts, with balances being settled between user accounts and the reserve fund.

The option to add a reserve fund to the nahmii architecture introduces both additional complexity and potential benefits to the design. While, strictly speaking, nahmii with a reserve fund would be no more secure than the alternative without one, users would gain the additional option to deposit into the reserve fund directly. A share of the revenue generated through the reserve fund can then be allocated to these users as an incentive for holding their funds in this way, effectively offering an interest rate on their deposit. This additional feature is attractive to certain groups and could act as a valuable incentive to recruit new nahmii users.

5.3 Temporary Rollback

The finality of driips on the nahmii network is determined by publication of data. In addition we also utilise the concept of checkpointing; each time a new higher global driip nonce is settled, nahmii records this value and all valid driips with a lower nonce are considered checkpointed.

In the event that operator fraud is detected, users can still settle their account up to the nonce value of the 'last known good' driip. Any driips beyond the point at which fraud was detected must never be settled, as to do so would risk compounding the prior fraud. Importantly, the smart contract cannot trust that the global driip nonce of a fraudulent driip is correct and therefore cannot take this value as the limit for further settlement. This global driip nonce can itself be fraudulent and, if this was used, the operator could deliberately commit fraud and effectively roll the protocol back to any previously advantageous point in time.

It is also possible for a driip to have a nonce value between the last valid settlement and the supposedly identified fraud. If the last known good driip is at global nonce 100 and the driip at global nonce 200 was shown to be fraudulent, we cannot yet say whether driips 101 to 199 are valid or not. At this stage, the smart contract will only begin the settlement process for driips up to global nonce 100 and will reject anything beyond this point.

In effect, this means that in the unlikely scenario that the operator commits fraud, any valid driips with a higher global driip nonce than that previously settled cannot be settled immediately. What we require is a mechanism for testing whether any driips between nonce 101 and 199 are fraudulent.

The mechanism for advancing the checkpoint takes the form of an interactive game. Users will be able to take part in an incentivised on-chain game where they can increase the checkpoint nonce up to the one previous to where the operator committed

fraud. 50% of users (those who were sent funds or made a good trade) with valid driip chains between the highest settled driip nonce and the one where the operator committed fraud will be incentivised to take part in this game to free up their most up to date funds for settlement. This interactive game will not be implemented upon initial test release, so users are recommended to monitor the progress of the rolling checkpoint.

6 Tokens and Airdriip

nahmii will be a tokenised protocol, however there will be no token sale. 120 billion tokens will be created and airdriipped monthly over a 10 year period. 100% of nahmii's revenue will be distributed to the token holders and the community of protocol facilitators. The revenue will be obtained from driips happening within hubii core's exchange and payment system. Though this project was not described in the original white paper and was sponsored by hubii AS, we recognise our existing hubii community and as such the token split for each monthly airdriip will be as follows:

- 50% proportionally to HBT token holders (including any HBT on deposit within nahmii)
- 20% proportionally to Ethereum holders that register or deposit in hubii core
- 20% to a strategic growth fund
- 10% to key partners in developing nahmii

As stated, all revenue will go to the token holders and facilitators of the protocol. It is essential that a significant fraction of this revenue is shared with token holders. In a similar fashion to many projects in this space, the security of the protocol is strongly related to the value of the token itself. As such, the token value acts as a bond for participants to ensure the correct operation of the security mechanisms. It is therefore critical to note that this token holder revenue share is not passive income; token holders must monitor, validate and secure the protocol. This is is an incentive mechanism for participation, just as Bitcoin miners receive a mining subsidy and transaction fees.

The majority of the remainder of the generated revenue contributes additionally to the 'Data Availability Bond' described in this paper. This bond, which increases over time, provides specific incentives for data availability validation and staking. This represents an additional reward for further active participation in the protocol for token holders, as only NII will be accepted for staking on data availability questions. Again, protocol value contributes directly to the difficulty of a 51% attack. As the value grows, a 51% attack becomes more expensive and the risk for an attacker increases.

Additionally there will be a minor security bond. This bond incentivise users to identify a number of operator-only attacks, where there are no colluding user's funds to be seized.

The exact share of revenue between token holders, data availability bond and the other minor security bond will be optimised over time.

6.1 Balance-Blocks

In order to calculate the appropriate airdriip share for each address holding HBT and ETH, we introduce the concept of *balance-blocks*. Balance-blocks are designed to measure both the number of tokens held at an address and how that balance has changed over time, where balance is measured in tokens and time in blocks. More formally, the balance-block is defined as the integral under the balance versus block height chart for a given address across a specified period. One balance-block is therefore equivalent to holding one token for the period of one block.

The balance-block concept is sensitive to how a user's balance changes over time, ensuring that all token holders are treated fairly during the airdriip. This approach compares favourably with the traditional method of simply recording address balances at a fixed point in time and allocating the airdriip tokens accordingly. In the traditional case, there is a strong incentive for a user to only hold HBT tokens around the time of the nahmii airdriip; a user who transfers 100 HBT into their wallet one day before the airdriip assessment will receive the same share as another user who has held 100 HBT for the entire month. This would cause undesirable liquidity squeezes on any HBT trading. Under the balance-block model, the second user would receive a much greater of share of the airdriip relative to the first. This additional share is proportional to the duration and magnitude of their holdings, thirty times more in this case, as they would have held 100 HBT for thirty days compared to the 100 HBT held for one day by the first user.

Airdriipped NII will be distributed to users in accordance with their balance-block holdings over the qualifying period. While this method of distribution will serve to minimise monthly liquidity squeezes on HBT trading due to the periodic nature of the airdriip, we have also chosen to utilise balance-blocks as we strongly believe that this form of distribution is the fairest and safest possible way to organise an airdriip. There must be a minimum level of balance-blocks in order to receive any airdriip, this is an essential anti-spam measure. The exact minimum will be communicated prior to any airdriip and may be subject to change over time.

The distribution of tokens during the nahmii airdriip cannot be a trustless process due to the limitations of the Ethereum network. It might also be necessary to stagger the airdriip to avoid unnecessarily high transaction fees. The intention is to eventually migrate the airdriip itself to occur within nahmii.

6.2 Revenue

All forms of driips will generate revenue within the nahmii protocol. Unlike their equivalent on the Ethereum network or in some other scaling solutions, fees within nahmii are designed to be predictable and transparent.

Fee levels within nahmii are not fixed and can be adjusted to meet the needs of the market. Fees will be initially set by hubii, however the nahmii Foundation will ultimately be responsible for future fee decisions.

6.2.1 Payment driip Fees

Payment fees are accrued trustlessly on a percentage basis, with discounts based on individual volume of each payment. The discount mapping can be set per currency, but there are also default amounts. Fees will be extremely competitive with Ethereum transactions, even at high individual payment volume. There may be a minimum fee requirement added to mitigate spam. Fees are paid in the currency of the payment.

6.2.2 Trade driip Fees

Exchange fees are based on a user's rolling 30 day volume, decreasing as transactional volumes rise. 30 day volume figures will be calculated in a trusted fashion and will most likely be calculated in USD equivalent initially. This will only be a minor trust issue as the fraud checks will not allow driips with fees that exceed the lower or upper boundaries. In the worst case, the operator can only apply discounts incorrectly. If it is observed that the operator is not consistently applying the right volume-based discounts then users can always safely exit and the operators reputation will be damaged. Fees are paid in the two currencies of the trade.

6.2.3 Trustless Generation and Claims

Fees in nahmii are incurred upon settlement, rather than at the time of the driip taking place. Every subsequent driip a user makes tracks the fees that they owe. Upon settlement, this fee is transferred automatically to a revenue fund. Revenue will accrue periodically and token holders will be able to trustlessly claim their share of the revenue after each period is closed by the operator.

A token holder's claim on their revenue share will be determined by the NII balance-blocks that they have accrued in the previous qualifying period. For more information on balance-blocks, please see the 'Balance-Blocks' section. In order to make this claim process trustless, a minor modification to the nahmii ERC20 token was required in order to keep track of balance-blocks during transfers.

If NII are deposited to an address where the user does not have control of their private keys, there is no guarantee that the user will be able to claim their revenue share. This would ultimately be at the discretion of the third-party service that the user has deposited their NII into. Token holders are therefore strongly incentivised to keep their tokens in addresses that they control at all times, which also ensures that those tokens are always available for staking into the nahmii Data Availability Oracle. It should be noted that when tokens are staked into the Oracle itself, revenue will continue to be trustlessly accrued and can be claimed back from the Oracle contract later. This ensures that there is no disincentive to stake into the Data Availability Oracle.

7 hubii core

hubii core is the first product built upon nahmii. It should be considered the reference implementation and will be open source. We expect and actively encourage many user interfaces to be created to interact with nahmii. Users will find not only a method of making payments and trading using nahmii, but also a growing set of features to interact in general with Ethereum. hubii core already includes hardware wallet integration for maximal security.

8 The Future

8.1 Immediate Withdrawal

Ordinarily, users will have to complete a successful settlement process prior to withdrawal. This process includes a dispute period, the duration of which will be optimised for safety of user's funds. However, we can enable the possibility of immediate withdrawal. This option requires that other nahmii users can review the settlement request, before offering immediate liquidity to the settling user, in exchange for a fee. This fee will be market driven and agreed by the two parties in advance of the withdrawal. The reviewing user, providing liquidity, will receive the settled funds after the dispute period instead of the original settling user. As such, the reviewer will accept the risk that the settlement process will not be successful. This risk should be very low, provided that the reviewing user checked that the relevant data was both available and correct. The effective interest rate charged by the reviewing user should therefore approach the settling user's perception of the time value of their funds. This feature will be added as soon as it is sufficiently tested.

8.2 Derivatives

The implementation of trustless derivatives is one example of additional forms of driips. nahmii will natively handle payments and trades, but other forms of driips will become available over time. Margin trading is well understood by the wider cryptocurrency community and this is likely to be the next driip that is added to nahmii. Although it is not particularly challenging to implement this function within nahmii, there are potential regulatory compliance issues which must be addressed before this form of trading is publicly released.

In addition to the benefits for traders, this feature is very important for hubii itself; trustless margin trading is our proposed method to provide the option to remove the exchange rate risk of hubiits (HBT) for users of our platform. Given sufficient liquidity, it is possible to hedge against price fluctuations and therefore mitigate this issue. This was discussed in our previous content platform white paper.

8.3 Fiat-Backed Tokens

The simplest way to integrate fiat in nahmii is by using fiat-backed ERC20 tokens. We are already exploring this possibility with various partners and we hope to add this soon. Inevitably, a fiat-backed token will be a trusted construction as fiat in itself is inherently trusted. It is therefore critical that we work with leading industry partners who are regulated by e-money or banking licenses. In many ways, there is no

functional difference between a fiat-backed ERC20 token and an account balance in an e-money service. However, there are regulated restrictions which must be adhered to.

8.3.1 Know Your Customer/Anti-Money Laundering

In order to comply with regulations, particularly when users are interacting with fiat currencies, it will at some point be necessary to identify those users. nahmii has been constructed to have this functionality natively without impacting those users who choose not to use fiat.

8.4 Cross-Chain Swaps

nahmii is designed for Ethereum and ERC20 tokens. We are working with partners on cross-chain implementations in a trustless fashion, although it is possible to implement a trusted model immediately.

8.5 Privacy

Transactional privacy has long been a primary concern of the blockchain community. While most blockchains offer pseudo-anonymity, there has always been an interest in moving to absolute privacy. We are currently undertaking research into nahmii driips with similarly high levels of privacy, which can maintain the same security guarantees.

8.6 Data Availability Redundancy

Data availability is a critical element of the nahmii protocol and it is necessary that we avoid 'data-withholding' false alarms. One way to maximise redundancy and minimise reliance on a single API is to use Distributed Hash Tables (DHTs).

8.7 Multisignature Wallets

An essential tool for security over funds within the Ethereum ecosystem has been multisignature wallets. nahmii will be multisignature compatible and this feature will be added to the platform at the earliest possible date.

8.8 Bond Limits

nahmii has a number of minor security bonds which are funded by a small share of nahmii's revenue. Currently a fixed percentage of nahmii revenue is diverted to increase these bonds, however there may be a sensible limit applied to how much these bonds can grow. Under the assumption that the operator is a good actor these bonds

will continue to grow and should never need to be paid out, as such they might be viewed as burned funds. As nahmii achieves mainstream success, this ever increasing bond size may become needlessly large. We may therefore make provisions for limiting this fund if required.

8.9 Other Token Standards

Adding other forms of tokens, such as ERC721, will be relatively simple once a standard has been finalised.

8.10 Patent

Certain elements of the nahmii protocol are patent pending. Patents elicit a mixed response from people, especially across the wider blockchain community. In this instance, the decision to seek a patent is driven by the need to keep the nahmii protocol both open and democratic on a perpetual basis. There are two principles behind this decision: in the first instance, that the patent application will ensure that no vested interest can interfere with or prevent the protocol from being deployed and used; second, by vesting the patent in the hands of the Foundation we are demonstrating our faith in the nahmii community to manage its availability and build upon it for the future.

When granted, this patent will be given over to the Foundation in perpetuity, conditional on the Foundation adhering to certain key principles. It will then be up to the members of the Foundation to determine a strategy for the nahmii patent. It may be decided that the best strategy is to give the patent away; however, this does not undermine the logic of applying for the patent in the first instance as to do so gives the Foundation the choice of how to proceed.

9 Appendix 1 - Comparison Between nahmii, Raiden and Plasma

There are countless scaling solutions proposed across the blockchain ecosystem. For Ethereum, the two most prominent constructions until now have been Raiden and Plasma. A high level comparison between nahmii, Raiden and Plasma is provided here. It is important to note that this was based on hubii's understanding at a particular time (Q1 2018), it may not reflect the current situation as these protocols continue to develop.

9.1 Raiden

Raiden is a payment-focused project, closely related to the Lightning Network from Bitcoin. It is a system designed around payment channels between individuals. The current status is that these payment channels are unidirectional and many-to-one. However, research is ongoing for eventual many-to-many and bidirectional payments, mainly for small payments. Once implemented in full, payments can be routed through a network of nodes, making multi-hop transactions feasible and allowing any user to pay another user of the network, who is connected through a chain of nodes.

A high level comparison between Raiden and nahmii:

+ Raiden is posited as a fully decentralised system. However, its design can certainly create some centralising forces the extent to which remains somewhat unknown

+ Raiden payments can offer somewhat improved privacy over the Ethereum base layer

= Raiden is low latency, similar to nahmii

= Raiden is highly scalable

- The system is best suited and targeted for small transactions due to the architecture, but it is difficult to quantify the scale until the system is established

- There is a high capital inefficiency inherent within Raiden's design

- Fees will be determined by the nodes through which payments are routed and therefore might be unpredictable

- Raiden's release timeframe is unclear

9.2 Plasma

Plasma consists of a blockchain type construction or 'externalised multiparty channels'. It consists of potentially nested child chains all reporting to their parent chain and, ultimately, the root Ethereum chain. Plasma child chains are similar to a blockchain, in that they work by bundling transactions into blocks which are submitted to their parents for later evidence. These transactions are merkleised and so only the merkle root of the block needs to be submitted to the parent chain. Recent research has moved to 'Plasma Cash', which offers some enhancements over Plasma at the cost of introducing additional downsides.

A high level comparison between Plasma/Plasma Cash and nahmii:

+ Plasma/Plasma Cash potentially has natively increased privacy

- Plasma/Plasma Cash will introduce latency, which is a problem for many applications. Of particular concern is the ability to build a liquid exchange on a protocol with high latency

- Plasma as originally specified requires multiple interactions by users to perform a single action. This is mitigated by Plasma Cash, but not without impact elsewhere

- Plasma/Plasma Cash has an unclear route to decentralisation

- Plasma/Plasma Cash has an unclear route to trustlessness

- Plasma/Plasma Cash has an unclear release timeframe

- Fees on Plasma/Plasma Cash will be unpredictable

- Due to it's high latency and periodic (potentially computationally intensive) commitments to parent chains, Plasma/Plasma Cash has scaling limitations overall and at an individual account level. An individual account might only be able to perform 1 transaction every 7.5s at best