Polaris: north-star governance

Aditya Kashyap (aditya@getpolaris.io), R. Alex Stokes (alex@getpolaris.io)

Version 0.0.2 (November 28, 2018)

Contents

1 Overview of Polaris

Polaris is the governance layer for the next generation of the decentralized web. We present a protocol that allows for any blockchain, present and future, to trustlessly use a secure set of governance mechanisms hosted on the network. Polaris is organized in a "hub and spoke" model that allows any other blockchain or dApp to efficiently connect. These users of Polaris connect to a governance module on the platform that offers a focus of coordination for the challenge under consideration.

The flagship module of Polaris will be the futarchy mechanism under development where prediction markets are used to drive outcomes that improve a system-wide metric dictating the welfare of an organization. We strongly believe futarchy offers the best form of decentralized governance to date as we can hold participants accountable to their activity. As we will see later, existing forms of governance are susceptible to various attacks as they fail to punish powerful actors who use their influence to extract undue value from the rest of the community.

After offering some motivations for our vision of governance, we will explore in more detail the "hub and spoke" model of the Polaris protocol and the design of the futarchy module. We describe the token model behind the protocol and examine a series of use cases where Polaris presents a way to prevent the destruction of massive amounts of value. We conclude with some risks and describe possible mitigations.

1.1 Challenges of governance

Governance is a big topic and consequently can be an overloaded term for many. Many hear the word governance and think about times when heated politics stood in the way of good governance. We want to examine the particular issues seen over and over across blockchain communities and explain why the design choices made in Polaris are best positioned to help. To begin, let's define governance as the processes and norms that the members of a community use to help them move past challenging questions that affect the community as a whole. In crypto, we have developed lots of methods for coming to consensus. A way to think about governance is that it is the set of things we need to consider when we need to change the existing method of consensus.

There are many different forms of governance given how general the setting is. We can talk about "softer" forms of governance – how do we decide what our values are and how does that reflect the tools our communities use – and also talk about "harder" forms of governance – what do those tools do and how can they change? Softer governance necessarily requires a subtler touch for relatively subjective issues and is not as easy to capture with proofs and algorithms. Harder forms of governance are easier to quantify and are trivially amenable to improvement under various metrics of success. Most real world issues usually have a mix of both and there is a rich middle ground between the two poles.

The situation intensifies with the Cambrian explosion of decentralized networks we have seen in recent years. Many participants are excited by their potential but are not used to thinking about distributed systems in decentralized contexts where there may be no central authority to turn to when things get difficult. An unfortunate example is found with the attacks on The DAO, an Ethereum-based project that aimed to create a venture fund directed by its investors. The DAO was structured as a series of smart contracts on the blockchain; after its deployment, it became clear that there was a re-entrancy bug in the code which allowed an attacker to start draining the invested funds. Given the immutability of the Ethereum blockchain, there was little investors could do to retrieve the stolen funds. A group of white hat hackers came to the rescue to save what funds they could, but ultimately the attacker made off with $50M worth of ether. The only viable recourse at the time was to execute a hard fork to restore account balances to a recovery contract. This hard fork was ultimately highly contentious and led to the creation of Ethereum Classic and calls out yet another instance of blockchain governance we could speak to. The point to make here, however, is that given the autonomy of the blockchain's smart contracts, there was little recourse to be had; an investor in The DAO could not simply call customer service and request a chargeback.

Polaris grew from our experiences with these communities and their various governance successes and failures. We believe those methods of governance referred to as "on-chain" governance will be most successful in the long run. On-chain governance refers to the notion that a blockchain's community use the chain itself as a place for coordination. Concretely, this could look like a miner using values in the block header to signal a change in the gas limit or it could look like a dApp using a smart contract to tally token votes for a particular issue. This notion contrasts with "off-chain" governance which uses existing platforms for users to coordinate around protocol usage, maintenance and change. Examples of this type of governance could look like the emoji reactions used to gauge support for a particular proposal or it could look like a mailing list where "core devs" of some software repo discuss the best way to enhance the codebase further.

Regardless of its form, the governance of these organizations should happen in as decentralized a manner as possible. The notion of decentralization is also a vague one, but for our purposes we can simply require the property that no user or users should have an advantage over other groups of users that was not previously agreed upon in good faith. In this light, Polaris is a protocol for facilitating the aspects of decentralized governance that are inherently tricky. We will now enumerate the benefits of on-chain mechanisms used in decentralized governance to motivate the design choices in our protocol.

The key benefit of on-chain governance is the fact that, ignoring externalities, it strictly lowers the transaction costs implicit in some organization using an off-chain method to coordinate around some change. The game-theoretic structure of governance is rich: first, the organization needs to signal their preferences in a way so that some decision can be made and consequently, the organization needs a way to then act on this decision. Moreover, each step of this process has to be viewed by each participant as legitimate in the sense of respecting the established norms for the organization as a whole. It should be apparent that these functions are harder to perform when the organization at hand is decentralized – how do you get a bunch of strangers on the internet to act contemporaneously, much less to cooperate in one direction or another? By using the blockchain as a common focal point, an organization can streamline much of the overall means of governance.

On top of the coordination gains, on-chain governance makes stakeholder participation more transparent than most off-chain solutions which may happen behind closed doors. With on-chain governance, users can directly inspect the methods used in some community and additionally see the actions taken in prior events as they are all recorded on the blockchain. This ability to engage directly with governance empowers users to participate. Users faced with less accessible forms of governance feel confused with unexpected decisions at best and disenfranchised at worst.

On-chain governance allows decisions to be enforceable in a way that off-chain governance only suggests by using smart contracts to execute binding outcomes. Governance is strongest when participants can place trust in the fact that decisions that emerge from a governance process will be widely accepted and implemented by a project's community. This gives meaning and value to the time invested in participating in a governance process, and encourages wider participation. One major deterrence that discourages stakeholders from participating in governance processes today is that many perceive it as a waste of time since decisions aren't truly enforceable.

A final point to make concerns longevity. A common mode of blockchain organization today is some foundation employing the founding team who works to improve the protocol and foster its adoption.

While this model makes sense during a bootstrapping phase, we have to think about what happens to these protocols after the foundation is no longer around, either due to lack of funding or turnover in headcount. On-chain governance helps migrate stewardship away from the early centralized team to the active users and stakeholders who want to see the network prosper.

This being said, on-chain governance represents a very powerful meta-feature that should be used very carefully. Opponents of on-chain governance point to how powerful this capability is and how it can easily be abused. As a brief example, consider a binding protocol upgrade to Ethereum that increases fees for all users who hold less than 100 ETH; this upgrade is deployed after some on-chain vote was gamed by the miners of the network. While this example may be contrived, it should suggest the type of thing we want to guard against with on-chain governance. The current debate mainly centers around the infeasibility of preventing these types of attacks with examples of systems that are live today; our position is that early failures should not discourage pursuing ways of achieving the aforementioned benefits.

2 How does it work?

Before taking a closer look at the futarchy mechanism, let's examine the foundation of Polaris to better understand the choices made to support more collaborative and efficient governance across the ecosystem.

2.1 Hub and spoke model

Polaris is organized in a "hub and spoke" model. This phrase refers to the architecture of the system where a single blockchain acting as the hub provides a base layer for any other blockchain that wishes to connect to it acting as a spoke. We refer to this base layer as the Core and each connecting blockchain as a Ray. The main idea is that the Core provides the base security for the platform and orchestrates any activity across the Rays. Integration with the Core avoids the quadratic communication cost implied by every community who seeks a collaborative governance solution and otherwise finds they have to integrate with each separate platform on a case-by-case basis. This strong emphasis on interoperability for the governance use case is a primary value proposition of Polaris.

A Ray can be considered its own blockchain, specific to the parties who want to participate in some singular governance process. The unifying aspect of a Ray is the logic, termed the *state transition function* that determines the meaning of the data on that chain (cf. "transactions") and what that data means in aggregate (cf. "the state"). The Core contains a collection of governance modules which serve as the state transition functions that instantiate a given Ray. This flexibility allows for users of Polaris to bring their own governance tailored to the particular situation at hand. The state transition functions are written in WebAssembly following the direction of many other projects in the space like Polkadot, Ethereum's ewasm, Dfinity and others. This decision allows for a host of popular programming languages like Go or Rust to be used to implement the governance modules along with smart contracting languages like Solidity or Vyper (pending a wasm backend).

Let's now examine how the Core provides the base security for the entire system. We take inspiration from the interoperability designs of Polkadot and the sharding architecture of Ethereum's Serenity protocol. The Core is a proof-of-stake blockchain where the system's validators are elected to produce new blocks in proportion to some "stake" they hold. To operationalize this scheme, we introduce a token called the **Polaris Coin** that serves as a unit of account for the validator's stake. Following the Casper design methodology, a validator's stake is locked as a bonded deposit and subject to rewards and penalties given proof of the validator's behavior on the network. For instance, if a validator attempts to create two conflicting blocks then proof of this attempt can be used to delete their deposit (cf. Casper's "slashing"). In the event that a given validator follows the protocol by successfully building the Core chain along with the system's other validators, they receive additional Polaris Coins as interest on their deposit as a reward.

In addition to their duties to the Core chain, validators are periodically randomly shuffled across the Rays. A validator always has duties to the Core chain and one Ray at a given time. The validator is expected to help build the Ray by collecting that Ray's data and assembling valid blocks according to the specified state transition function for that Ray. With sufficiently strong randomness, it will be difficult for a subset of the validators to collude for the purpose of carrying out an attack against a given Ray. To generate this randomness, the Core chain will involve a game like RanDAO where validators submit entropy to the chain via a commit-reveal scheme in order to drive the random beacon. We are closely following the efforts by the Ethereum Foundation and the Chia Network to investigate the applicability of verifiable delay functions (VDFs) to strengthen a RanDAO-style random beacon.

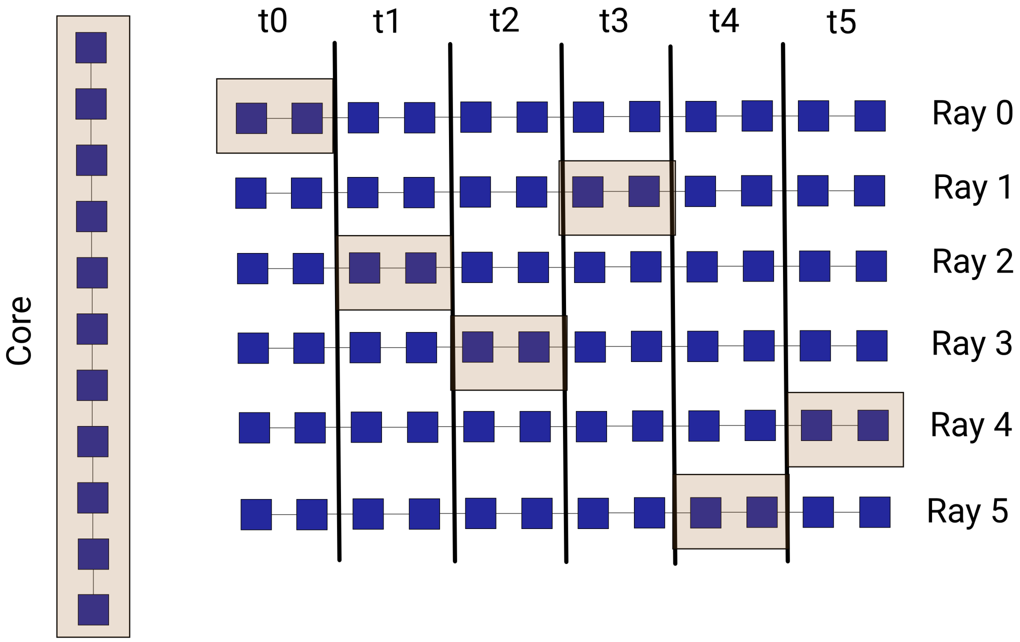

Figure 1: One validator's view of the network. Each blue square is a block. The orange windows show which portion of a Ray a single validator is responsible for at any given time. While not to scale, this drawing shows how at each time slice t_i, a validator needs to have a view of the Core and part of one of the Rays. At the latest time t_5, this validator is building Ray 4.

In addition to providing the system's security, the other main task of the Core chain is to facilitate orchestration between the various Rays in the system. In accordance with the Ray's state transition function, a validator could process a message type that logs some data into a queue on the Core chain. This message contains a destination address (e.g. the hash of the genesis block for that Ray) for some other Ray with the intent being that validators on the destination Ray will be able to move that message from the Core's queue into the state of the Ray.

This queue functionality is the main vehicle for foreign chains to enter the Polaris system. The mechanism here issues the creation of a Ray which serves as a light client of the foreign chain. To move tokens from the foreign chain into the Polaris system, a Merkle proof is presented to the effect that the foreign tokens are locked on the foreign chain. This proof generates a corresponding "phantom token" on this Ray which can then be routed to the appropriate Ray for participation in the foreign chain's governance.

2.2 An example

Let's look at an example to trace how a user of Polaris would successfully use the system to complete some governance objective.

Consider the stablecoin project Maker running on Ethereum. Maker is in part a series of smart contracts that facilitate a decentralized peg of their stablecoin called Dai to an external reference rate, e.g. the US dollar. To create some Dai, a user puts down some underlying collateral to secure the Dai in an instrument called a collateralized debt position (CDP). A series of cryptoeconomic games play out between creators of CDPs and those who aim to liquidate risky CDPs with the outcome being a rich set of feedback mechanisms that help Dai target its reference price. There are many parameters in the system describing things like the collateralization ratio sufficient to maintain a CDP and things like the total number of Dai in the system (the "debt ceiling"). Maker commits to decentralized governance by conferring holders of their governance token MKR the privilege of setting these parameters in real time. For this example, let's say that there an increasing demand for Dai and the system is near the debt ceiling. In this circumstance, holders of MKR can vote to raise the debt ceiling to allow for the creation of more Dai. Note that this scenario has already happened in the beta version of Maker's stablecoin, the single-collateral Dai.

Realizing the high-quality nature of the governance modules on Polaris, MKR holders decide through various means (forums, twitter, reddit) to use a majority coin-vote module hosted on Polaris. There is widespread knowledge across the community that the outcome of this decision will be respected (n.b. see the section later that discusses the nature of "bindingness" in decentralized governance). To participate, MKR holders send their tokens to a bridging smart contract on Ethereum. The purpose of this bridging smart contract is to generate the light client proofs necessary to send tokens from the Ethereum chain to Polaris. This bridging smart contract also acts as a light client of some smart contract on a Polaris Ray so that tokens can return from Polaris to the Ethereum chain. The tokens deposited in this bridging smart contract are locked here until a proof from this Polaris smart contract is presented that the Polaris tokens have been destroyed.

Given the proof of locked deposit, MKR holders can submit a transaction to the Polaris bridge to generate an equal number of Polaris Coins. It is worth pointing out that this bridge transaction can in general be facilitated by a decentralized exchange with a spot price for the conversion or according to some token bonding curve which greatly widens the design space for users of Polaris. The Polaris Coins can then be moved from the bridge Ray to the specific Ray hosting the instance of the majority vote module that MKR holders intend to use. To do this, the MKR holder sends a transaction moving the Polaris Coins from the bridge Ray to the Maker Ray via the queue in the Core described earlier. In the simplest case, once the Polaris Coins have made its way to the Maker Ray, users send the Polaris Coins to some smart contract on this Ray which signals their vote to raise the debt ceiling.

Let's assume that the majority of MKR holders signal their preference to raise the debt ceiling. Anyone

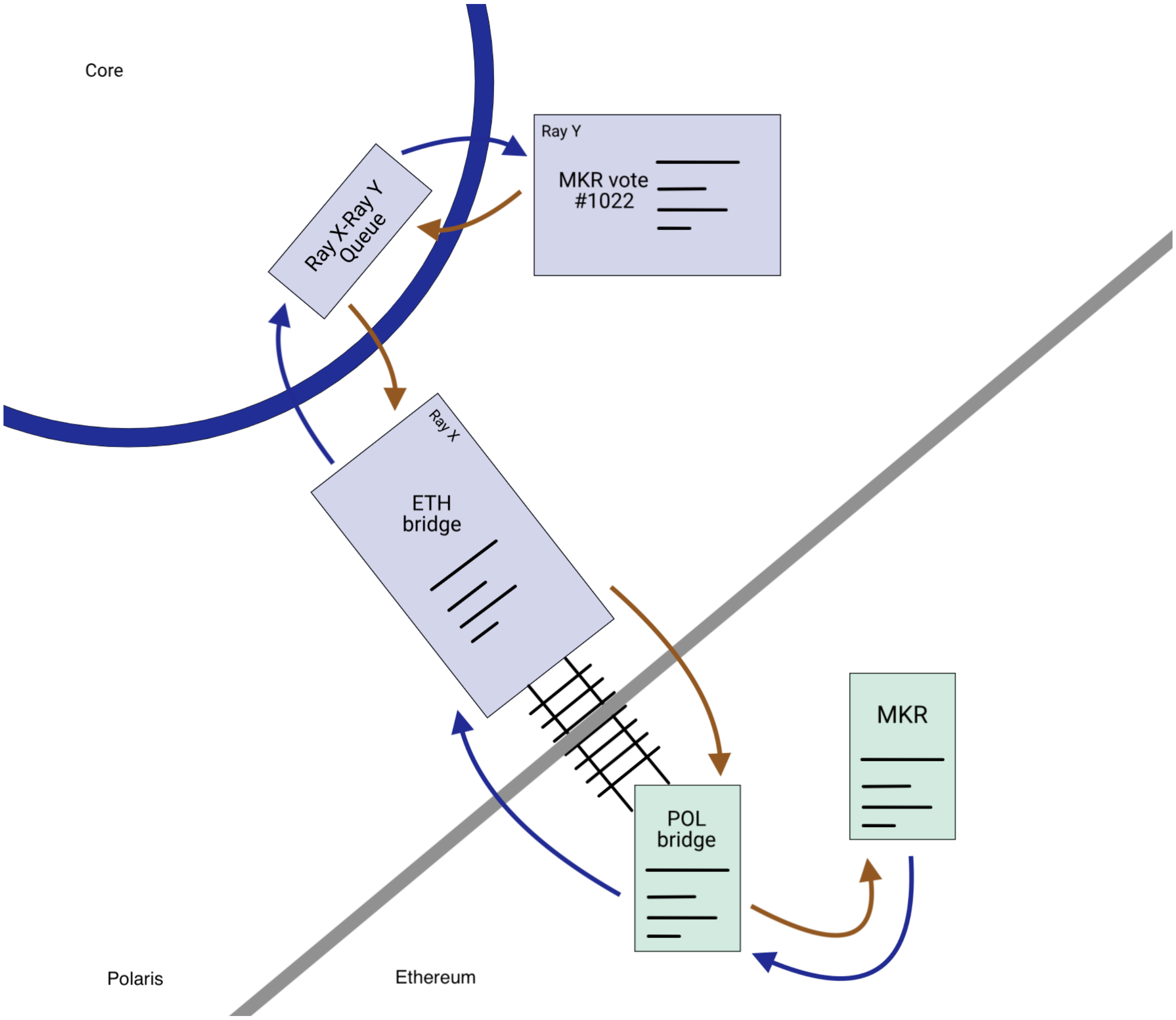

Figure 2: Example showing how a Maker holder would participate in a token vote held on Polaris. The blue arrows show the movement of assets to the vote; the orange arrows show the movement of assets back to the MKR contract where they can again be used freely. The green boxes are smart contracts on the Ethereum chain. The light blue boxes are Rays in the Polaris system. The Core is shown as a central disk for illustration purposes.

will be able to see this outcome play out in real-time in accordance with the logic of the governance module. From this point, MKR holders would go ahead and author the necessary transactions to raise the debt ceiling. In general, the outcome of this governance can even be automated itself: a light client proof of the outcome can be communicated back to the Ethereum chain which then triggers an update in the Maker smart contract machinery. We refer to this functionality as the “bindingness” of the governance process and will speak to it in greater detail later. For now, just recognize it is possible with sufficient smart contract infrastructure.

After the token vote has ended, users can follow the reverse process to unlock their MKR on the Ethereum chain. They move their Polaris Coins back to the Ethereum bridge Ray and burn them. This activity generates a light client proof that the Ethereum smart contract can recognize to unlock the original MKR.

2.3 Why this design?

At this point, you may be wondering why we require all of this complexity for a simple token vote. The example just given was meant to illustrate the basics and glossed over the points in the process that allow for greater collaboration and interaction. We provide some points elaborating on the design here:

2.3.1 Secure governance

The first design goal is to leverage the security of the Core chain for the efficient operation of the Ray chains. This requirement is readily satisfied by the "hub and spoke" model. The system's validators provide security – the users of Polaris just enjoy secure governance as a by-product.

2.3.2 Governance liquidity

We envision a future where all blockchains and dApps present and future use Polaris for their governance. The main benefit to using a shared platform for governance is to allow for the trustless cooperation and coordination of multiple decentralized communities. A major argument for building a dApp on top of Ethereum at the moment is that you immediately get interoperability with all the other dApps on top of Ethereum. This situation gives us constellations of communities like the "decentralized finance" initiative De.Fi which uses 0x for exchange, Dharma for lending, Set for derivatives, Abacus for compliance and WalletConnect for mobile usage. Strong governance will be required as each of these protocols grows and adapts to their user's needs. You can easily imagine situations where the change of these protocols in lockstep can lead to features that help users but would otherwise be tricky to coordinate (either due to real technical or organizational challenges or merely the perception thereof). By providing a common arena for the governance of all protocols, everyone benefits from a reduction in the friction that accompanies change.

2.3.3 Sovereign governance

The full realization of this vision necessitates a system independent from any existing system. We wish to provide a "neutral grounds" for governance to take place in, isolated from the technical limitations or political limitations of some underlying platform. To be concrete, do you really want CryptoKitties to stop your block size debate? By designing Polaris as a separate system, we aim to provide this "decentralized commons" to help any and all communities coordinate around mutually beneficial outcomes.

2.3.4 Shared governance

Governance expertise in a particular niche vertical shouldn't be isolated to a single crypto project. It should ideally be shared across the entire crypto universe across multiple projects to increase the size of the pie for everybody. The Polaris on-chain governance solution allows for this to happen frictionlessly, something not possible off-chain.

2.3.5 Efficient governance

By allowing for coordination across Rays, the interoperability benefits that apply to dApps on Ethereum today apply to the governance efforts of those same dApps tomorrow. The modular and generic nature of the Ray allows for a larger design space in terms of governance initiatives that can occur between multiple parties. This large design space gives users a vast sandbox in which to experiment with methods of governance that we cannot imagine today giving organizations a chance to find even better ways to collaborate at scale.

2.3.6 Diverse governance

While many of the examples in this whitepaper focus on the governance of protocol assets, Polaris can be used for governance in a variety of settings. For example, some corporation could organize in a "distributed" manner (i.e. everyone is remote) and use Polaris to carry out some corporate governance use case (e.g. determining the next quarter's priorities) while incorporating employees all around the globe.

3 Modular governance

While we are excited to see the various governance experiments carried out on Polaris, we are personally most excited by the potential of futarchy. Let's now look at futarchy in detail to understand its benefits and see how the flagship module of Polaris will work.

3.1 Futarchy

Let's recall the basic setting of a futarchy. The stakeholders of an organization decide on a metric. The way to decide on a metric can be any of a number of processes including something like a representative democracy. Once the metric is set, proposals are given with the aim of introducing policy changes in a way as to maximize the metric. For each proposal, a pair of prediction markets are created with assets that pay out conditional on their mandate being carried out. Some time passes to allow speculators to trade each asset and then the relative prices are compared. If one market clearly demonstrates one preference over the other as demonstrated by their price data, then the resolution of the proposal is clear. For example, let's say there is some proposal with an approval share price five times greater than the rejection price. The change as given in the proposal would be implemented and then the metric is assessed. If the metric has indeed risen in response to the change, then the traders who purchased rejection assets would lose their funds in a transfer to the traders who correctly predicted the outcome of the proposal. If instead, the metric has fallen in response to the change, then the traders who purchased acceptance assets would lose their funds in a transfer to those who purchased rejection assets and correctly forewarned the harmful effects of the change. The high-level idea here is to use the metric to determine "what we want," and use prediction markets to determine "how we get it."

This construction has a variety of benefits when applied to the process of governance. Prediction markets are known to be highly efficient aggregators of public and private information in a simple, quantifiable number – the market price. The idea of futarchy is to use this tool to solve "information failures" in policy deliberation. If a trader has some private information that gives them an advantage

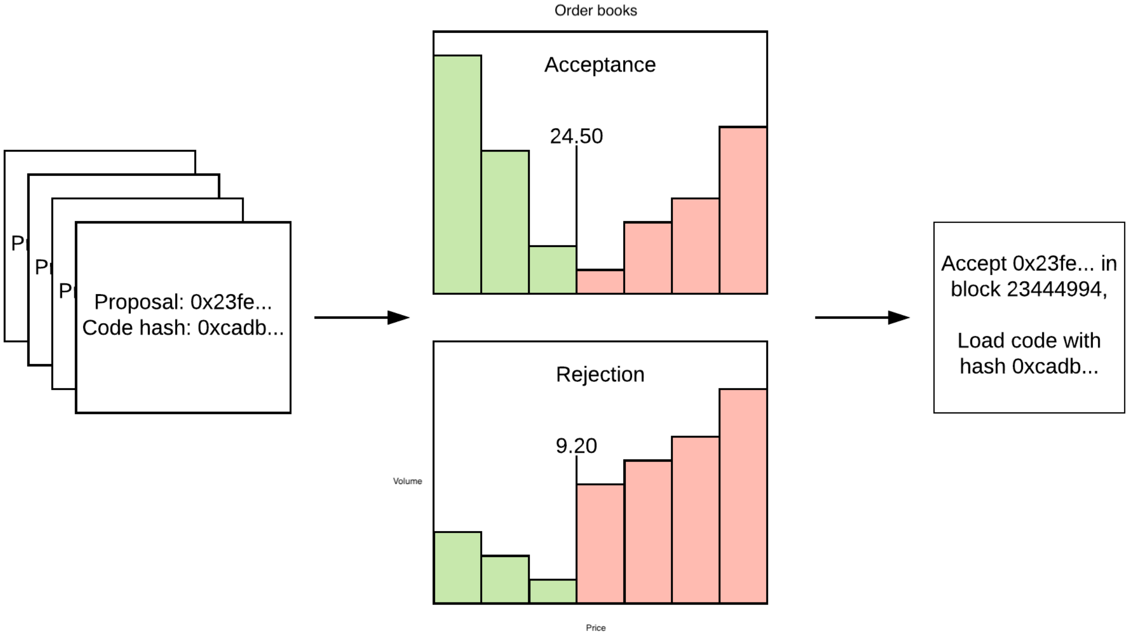

Figure 3: A schematic of futarchy where prediction markets determine the acceptance or rejection of protocol upgrades.

in predicting the outcome of an event, they now have a direct incentive to use this information as a bet against the opposite occurrence. Summed across all market participants, prediction markets integrate the opinion of the crowd which has obvious appeal to the process of governance where we wish to integrate every stakeholder's preferences.

Futarchy directly addresses some of the failures we examined earlier in the various governance models. Prediction markets are directly emulated by smart contracts so that they provide an accessible medium for the expression of opinion in a blockchain by a user's coin resource. This mechanism improves on the ad-hoc nature of governance in Bitcoin where centralized groups used outsized influence from their positions and resources outside the protocol to affect the direction of the blockchain. Moreover, the range and depth of a given asset allows for a rich medium with which to express fractional opinion. This ability yields a higher fidelity signal of preference within the protocol and presents a greater opportunity for discourse more refined than "fork or not." A participant may back a portfolio of various proposals to express their unique and nuanced world view, which is a more natural form of expression for humans than binary voting.

The benefits continue: by providing for a reward in the form of a conditional payout given the correct bet, the prediction market aspect of this system directly incentivizes "voters" with pertinent information to become "traders." This property solves a well-known problem with voting known as voter apathy where any specific individual is not properly motivated to actually cast a vote. We expect the DAO voter turnout would have told a much more representative story if participants had stronger incentives to signal their preference. Similarly, votes cannot be cast without some conviction – traders on the losing side forfeit their bets to fund the winning market. This fact implies that there is an explicit cost to voting in a way that doesn't reflect what you truly believe will improve the overall health of the system; this cost directly translates to an increased cost of a bribe. More expensive bribes mean votes in the system are harder to buy which limits the formation of cartels by making it more and more expensive for cartel operators to influence the outcome of a given decision. In this way, futarchy introduces some risk to the delegated voting protocols that are popular in "liquid democracy systems." An otherwise uninterested voter who just wants to collect a bribe has to think twice before joining some delegative body.

We can't help but point out the very nice cryptoeconomic instance of "checks and balances" we see aspired to in ad-hoc systems like Bitcoin governance with futarchy's separation of the metric from the individual prediction markets. As long as the metric is defined in a way as to maximize the user's utility, then the markets are necessarily constrained to realize those outcomes; anything else arises at some large cost.

3.2 How sound is sound?

This model doesn't come without potential downsides. Failures in prediction markets can occur where pricing data doesn't reflect the participants' valuation of some outcome. This failure arises in a prediction market when the relevant information is not equally accessible to all market participants. Given the open source nature of this endeavor, we expect to minimize instances of this problem and can build tooling on top of Polaris to facilitate the spread of information.

There are no real precedents for what the futarchy metric that the prediction markets are tied to should be, or how it should be defined. Thus, the ecosystem could be susceptible to "voting attacks" early on, where bad actors discover a weakness in the metric definition and push malignant proposals that seek to erode the organization's longevity or functionality.

Another general cause for concern could be market manipulation. In this regard, futarchy is actually anti-fragile - participants with more accurate knowledge are incentivized to bet against market manipulators, who are subsequently punished. This follows as there is a fundamental information asymmetry between the informed participant who has knowledge they can use to assess the misvaluation of an asset and the manipulator who is forced to move in one direction regardless of other facts. An example manipulative attack would be like the "bear raids" seen in the equities market. This strategy has the traders try to drive down the price of a stock in an effort to cover a short position. If another trader has information relevant to the policy outcome (and sufficient liquidity), they can assume a long position to counter the effects of such a raid. In the scenario in which the raiding traders were wrong and the informed trader is correct, the markets will ultimately resolve in the informed traders' favor.

You may notice that there is nothing in this protocol stopping the formation of voting pools, where users aim to pool their capital in an effort to amplify their preferences. This process seems analogous to the occurence of delegates in the delegative models discussed previously with the bribing attacker example. The key difference is that in the futarchy model users who delegate have their capital at risk. This fact follows from the property that market participants who elect the outcome that does not contribute to the public good (as ascertained by the metric) lose their funds. In the worst case scenario, the damage that can be done by swaying a vote is eliminated due to the nature of the metric. We don't aim to begrudge the nature of economies of scale; what we do require is that it remain accountable.

3.3 Applications of futarchy towards efficient governance

The market-based nature of futarchy lends itself to a novel number of applications in governance that are simply not feasible today. An interesting application that highlights the benefits of on-chain governance is the creation of research firms who specialize in futarchy operations. These firms will position themselves as governance experts; given the potential for financial upside in correctly predicting which things will improve the futarchy's metric an individual firm can develop "trading strategies" that correspond to governance outcomes. Allowing for competition between all such firms should produce healthy markets and correspondingly, healthy governance.

Moreover, a record of each firm's performance (perhaps pseudonymized) will exist in public on the blockchain. This data can be used by any participant to follow the efforts of those who specialize in researching decisions, whether for the purposes of lending support or to hold purported experts accountable when discussing the various merits of this or that proposal.

4 Token model and key actors

Polaris employs a dual token model. The first token, referred to as the Polaris Coin, is the primary token in the system and serves many of the same functions that Ether does in Ethereum. The second token, the Polaris Share, is a token given to early supporters of the project and entitles them to a pro rata share of revenue on the network.

4.1 Polaris Coins, a token to coordinate users around governance

Polaris Coins allow stakeholders to participate in their platform's governance. In general, Polaris Coins are the instrument that let a participant enter into usage of one of the governance modules. Taking the futarchy module as an example, Polaris Coins are used to purchase accept or reject shares of a particular proposal's prediction market. A given community does not need a specific token on Polaris (although it will be possible) to carry out their governance on the platform. The linkage from the activity on some Ray to the interested communities will be clear from some other mechanism, e.g. some community's foundation has stated they will respect the outcome of governance on a particular Ray.

Polaris Coins also serve a critical function as the means to compensate validators for their efforts in maintaining the Polaris system. In order to receive this compensation, validators must stake a deposit of Polaris Coin to enter the validator set. The Coins also act as the "reserve" token, in the sense that all other activity in the system can move through it. This flexibility allows for multiple groups to carry out joint initiatives that may otherwise be too costly to carry out independently. Any fees incurred on the Core chain are payable in Polaris Coins and it is straightforward to require payment in Polaris Coin for any fees incurred on the Rays.

4.2 Polaris Shares, a token for early supporters

Polaris Shares will be a finite-supply token that will allow holders to collect some fraction of the revenue on the network. Each governance module will have an associated "governance fee" that dictates the cost associated with using the service. Let's consider an example with the futarchy module where

Share holders will collect a portion of the proceeds from every settled market. Let's say there is a futarchy instance resolved for a referendum concerning what to do with the funds in the `eos.savings` account on the EOS blockchain. This account collects $\frac{4}{5}$ of inflation on the network and is intended to fund initiatives meant to improve the community. However, the blockchain launched without a way to spend the funds themselves and there is now a debate about how to do this. EOS stakeholders decide to use futarchy on Polaris to determine if the funds should be burned or held aside for some future distribution mechanism. The futarchy sees a lot of activity and the total payout for winning participants is 10,000,000 Polaris Coins. For sake of argument, the governance fee set for this module is 10% which implies that 1,000,000 Coins are distributed pro rata to Share holders.

The purpose of the Polaris Share is to build support for the network before it is launched. The nature of the share is analogous to the founder's reward found in ZCash.

4.3 Key actors

The key actors in the system are the proof-of-stake validators and the communities using Polaris to facilitate their governance. The validators are rewarded for building the system and in turn providing its security. The users of Polaris use the governance modules for their own purposes and in general you can have many different actors involved in particular types of governance. Given the large design space enabled by the modularity of the system, we look forward to seeing the types of governance and unique cryptoeconomic mechanisms employed on the network.

5 Use cases

Given the wide nature of the scope of governance, we can imagine many use cases for Polaris. This scope includes the governance decisions of any blockchain, present and future. It can even include future permissioned chains deployed in the enterprise setting. To stimulate your thinking about specific examples consider:

- The Bitcoin community wants to determine if the blocksize should be increased. They implement a form of liquid democracy to allow a wide variety of users to delegate to technical members of the community who have special insight into this question.
- The Gnosis community (and Ethereum by extension) wants to determine if an assassination market hosted with their prediction market tooling should be deleted from the global state. They use futarchy themselves to discover the outcome.
- The Bitcoin Cash community wants to decide if the protocol should be changed to incorporate new opcodes including OP_CHECKDATASIG. They use a token vote to decide.
- The Ethereum community wants to use futarchy to determine if a hard fork to unfreeze the Parity funds increases the protocol's utility.
- Members of the Ethereum Enterprise Alliance want to add a new member. Existing members implement a permissioned vote on some Ray that uses zero-knowledge proofs to convey the vote result without revealing any individual vote.

We'll now turn to a more thorough example illustrating the unique benefits of Polaris with a real-world example borrowed from Scott Bigelow's presentation at devcon iv in October 2018 (starts at

1:59:00). The presentation focused on Scott's experience building the Augur platform and bringing it from an idea that existed even before the Ethereum mainnet launch to the popular prediction markets platform it is today. The pertinent part of this talk describes a problem the Augur team ran into due to their smart contract infrastructure and the limitations around gas on the Ethereum platform. Several contract types on Augur take quite a bit of gas; for example, the main market contract takes 5.8M gas to deploy. In order to hide this cost from the end user of Augur, the team instead has the user deploy a delegation contract that uses the heavy contract for market logic while only demanding gas costs on the order of the 'DELEGATECALL' opcode (which the talk says is around 2000 gas).

A similar delegation architecture is used for the REP ERC-20 token itself which alone costs 2.4M gas to deploy. The problem arose when it was time for Augur to integrate the REP token into the 0x decentralized exchange infrastructure. The smart contracts implementing the v1 of the 0x protocol had a hard-coded upper limit on the amount of gas it would take to do a standard 'getBalanceOf' call from the ERC-20 interface. This limit was set to 4999 gas which as Scott points out should be more than enough gas for a simple storage read. It turns out that this limit was too low when the originating token (i.e. REP) is accessed inside a `DELEGATECALL` context. The 0x contract would not allocate enough gas for the call to the REP contract and so the token was incompatible with 0x. This oversight resulted in an inability to list REP on the Paradex relayer acquired by Coinbase. Moreover, it limits the ability of REP to enter into the 0x ecosystem and all of the financial applications built on top of it. This kind of thing slows the growth of the prediction markets platform and all those who wish to build on top of it by integrating into some new use case.

Polaris presents a great solution to this type of problem. Let's say that both communities are sufficiently motivated to fix this issue and it just takes a smart contract upgrade that has already been vetted. Members from each community can decide to use Polaris futarchy to determine community consensus. They would follow the process previously described, generating some Polaris Coins in response to locked REP or ZRX. These coins are used to purchase shares in the futarchy markets to signal the crowd's preferences. Whether the outcome drives a smart contract upgrade autonomously or instead the stewards of the 0x protocol agree to upgrade the smart contract in question manually (see section 6.1.2 later discussing "binding"), this use of Polaris demonstrates the value in having a place for multiple decentralized communities to come together to work on multi-party governance initiatives.

Without Polaris, the best these communities can do is essentially what happened: the core devs for each protocol had a chat about the state of things and they collaborate to fix the issue in 0x v2. While this worked in this specific case, readers should realize that this type of solution will not work as every application in the ecosystem realizes their claims to full decentralization where no single organization or team has full control.

To demonstrate even further the power of a shared space for governance that Polaris offers, we can extend the prior example to a more hypothetical one (although still following the team's announced road map). In this case, Augur wants to integrate Dai as the primary payment token used in the prediction markets in their v2. Doing so will allow users of Augur to avoid volatility risk while holding funds in some market. While this is an exciting use of Maker's stablecoin and adds tangible benefits to users of Augur, you can easily imagine a scenario where there is yet another smart contract bug that needs to be fixed. The story only gets richer when mixing in more protocols, like adding in a new feature of 0x to this example. It is the vision of Polaris to usher in a world where each of these protocols can grow and change with their users while still retaining autonomy, in an attempt to fully realize the goals of the decentralization movement we find ourselves in.

6 Risks and mitigations

6.1 Dangers of on-chain governance

Conversations around how to implement decentralized governance can quickly lead to discussing the dangers around some types of on-chain governance. The phrase "on-chain governance" refers to using the blockchain as an integral part of the governance process. Blockchains are great at coordinating users so it seems good at first; however, the catch comes when we think about how we will map the types of governance we are used to (i.e. some kind of voting process) in a Sybil-resistant fashion. To achieve Sybil resistance, we usually require proof of some kind of scarce resource with obvious choices being hashpower (e.g. from a mining farm) or coins (e.g. in a token vote). The immediate consequence of this fact is that the distribution of the scarce asset may not match our intuitions around who should be able to participate in governance and to what extent. The nightmare scenario involves a cryptocurrency "whale" – someone who holds a lot of the token used for a vote – to express outsize influence on the governance outcome. To learn more about this concern, we refer readers to the writing of Vlad Zamfir (1) or Vitalik Buterin (1, 2).

Our position is that we recognize the power behind existing forms of on-chain governance and the Polaris project is committed to finding new forms or improving existing forms of governance that limit the downsides of on-chain governance while maximizing the upside. Given the importance of governance to any project in the space, we do not think we should shy away from working on these methods today. It is a core to the project's ethos to take an iterative approach to governance in a way that can build a working process with a community over time. We believe that we can do better, together.

6.1.1 On forking

Hard forks are the de-facto method of decentralized governance used today. Our position is that while effective, they are blunt tools for most initiatives. Hard forks always carry a high coordination cost whether you are a developer of some blockchain or simply a user in the community who depends on the underlying chain for your business. A great example of the cost of a hard fork can be seen in the recent testnet deployment of Ethereum's Constantinople release. A hard fork was deployed to a testnet and it became apparent that there was not enough hashpower on the testnet to successfully mine the new chain across all clients. This caused a chain split. As the problem was being sorted, the fork lengthened and then a new set of bugs kicked in concerning synchronizing older blocks which only made the problem of the fork worse. While we admire and applaud the efforts of the Ethereum Core Devs to fix the fork, this incident should underscore the riskiness of this methodology. We envision a future where hard forks are a means of last resort.

6.1.2 On the question of binding

One general function of governance is to incorporate the preferences of the various stakeholders and produce some signal that reflects some combination of these preferences. There is usually some follow-on process that implements an outcome according to the signal. For example, we can take the case of a local election for a city council with voting. Every citizen gets to register one vote for the candidate they think will do the best job. The votes are tallied and the result is revealed; this is the signaling. Everyone who took part in the process probably did so with the understanding that the candidate with the most votes is selected to take on the actual role; this is the "follow-on process."

We have the same general framework when working with decentralized governance. However, in this context there is a substantial change due to the fact that many of the resources under consideration are operationalized by software. With resources rendered by code, we can imagine writing more code that concerns the change of the original code. In practice, we can find instances of systems that aspire to have as much of the governance as possible driven by the action of smart contracts.

This functionality could be readily realized even today for basic examples: think about the Uniswap decentralized exchange. Let's say the community wants to tweak the model underlying how liquidity is pooled for a given trading pair. This change could be implemented in a way so that the market making contract points to some other smart contract for the pooling logic. With this reference to the logic, the system can now be upgraded by deploying a new smart contract and updating the pointer. The question of how this happens in a way that respects the community's desires is precisely a question of governance. One possibility that illustrates the notion of "binding governance" is as follows: the smart contract machinery to upgrade the market making logic is extended so that it incorporates some data from yet another smart contract. This new smart contract could be (for example) a futarchy module on Polaris. The key idea here is that a closed loop can be formed where the upgrade process automatically incorporates the outcome of the module's governance.

It should be clear that such a system has reduced the coordination costs to a minimum. Especially if the underlying protocol provides for a monetary policy that subsidizes all of the various smart contracts involved, the economic cost to such a system becomes marginal. The power possible with this type of governance is a tool like any other and the point to realize is that this power can be used to help the users of a system but also to hurt them. We expect that as the state of decentralized governance matures, more and more users will be comfortable with the exact methods used and this will allow for increasingly efficient designs.

In the case of futarchy with a bad metric, one can imagine a scenario where the majority tyrannizes the minority by pushing through a proposal that could devalue or seize the assets of the minority. Because the proposals that are accepted by a prediction market are binding, the minority that is being unjustly punished would have no recourse.

In the meantime, we recognize that it is probably in some organization's best interest to not drive binding changes from governance on Polaris, although in theory it will be readily available. Events on Polaris will act as efficient signaling that the community at hand will all agree to respect. Under this regime, these organizations will then turn to off-chain mechanisms to follow through the remaining stages of any particular initiative, whether that is the release of some funds or the upgrade of some smart contracts. We fully expect that over time more and more systems will elect to have more and more of the governance process automated on-chain and the Polaris team will work to realize this goal.

6.2 Attacks against futarchy

Critics of on-chain governance such as Vlad Zamfir have rightly raised concerns of plutocracies emerging, with whales controlling or manipulating governance in major ways. A correctly designed prediction market with a properly chosen health metric mitigates this risk because it will penalize whales for manipulating the market in ways that are detrimental to the communities involved.

6.3 Attacks against the Core chain

There are a variety of attacks that can be attempted against the underlying public infrastructure that hosts each organization's governance. We will discuss some of them here. The key defense is the cryptoeconomic security generated by the system's validators. Their behavior can be constrained in such a way that they have strong incentive to follow the protocol honestly.

The primary attack involves an economic majority of validators colluding to fork the Core chain. Under this scenario, it will be unclear to a user which series of events they should trust when determining governance for their own organization. The ways to prevent malicious forking are covered in some detail in the Casper FFG paper which we will adapt to our system. The main defenses are strong slashing conditions which allow for punishment of a validator upon evidence of bad behavior and a long withdrawal period to allow full nodes who have gone offline to synchronize to the current head of the chain that the rest of the validator set currently sees.

The next major attack vector will be the randomness beacon generated on the Core chain. The idea here is that an attacker could manipulate their RandDAO entry so that they are called to produce more blocks than seen in expectation. With more control over the future execution of the protocol, the attacker could start to gain exclusive access to a particular Ray where they could then attack the initiatives of some organization's governance. The problem is exacerbated by the fact that at any given time, a specific Ray will have a fraction of the system's economic security and that this single attacker could in fact be several colluding validators. We follow the research lead of the Ethereum Foundation and the Chia Network into how to strengthen these beacons with specialized hardware to compute VDF proofs. The strategy with these proofs is that it takes a certain amount of time before a validator can determine their next RanDAO update at which point they will lose the opportunity to "grind" several different options in an attempt to bias the randomness.

Given the dynamic nature of the number of Rays that may be in the system at any given time, there could be a scenario where there is a mismatch between the number of Rays in the system and number of validators in the system. We again rely on economic incentives to target a nice equilibrium. Let's say that there are N validators in the system and that a given validator provides V amount of security. In the case that there are R Rays, we aim to distribute validators as equally as possible to target a security threshold of V * N / R. If N is much larger than R (in the case where there are many more validators per Ray than Rays) then the security of a given Ray increases. However, the converse situation can be problematic: if N is small relative to R, then the overall security per Ray has decreased from our ideal and any given Ray becomes easier to attack. The fix here is to offer an attractive reward to validators for performing their service while not inflating the coin supply unnecessarily. The equilibrium reward rate should target a value such that a given Ray has at least a minimum number of validators per unit time. Economic analysis of particular parameters for the genesis protocol are forthcoming.

6.4 Attacks against the Ray chains

We can now consider attacks against the Rays themselves. In general they are all subsets of attacks against the Core so everything said there applies here. One particular attack to note is the case when a Ray's validators fail to extend the Ray chain. This failure could result from either a careless validator or a malicious validator and the protocol should act to punish either misstep as a liveness fault (see this article for more context). The fix for this type of attack is to tie some part of the validator reward to including proofs of updated Ray chains. If a validator cannot produce a proof of new Ray blocks then they forgo some rewards.

While it may be sufficient to simply enshrine Ray liveness penalties into the genesis protocol, a particular Ray can include some notion of transaction fee paid to validators in accordance with its state transition function. This fee would serve as an extra incentive for validators to follow the protocol honestly and increase the cost of an attacker.

Another attack that has been raised in the course of discussing the design of Polaris is the following: let's say a dApp on Ethereum uses Polaris and locks their token in the bridging smart contract. They produce an equal number of phantom token on Polaris. The attack is that the holders of the phantom token then try to move the phantom token back to the Ethereum chain across some different bridge, in effect executing a double spend. Putting aside the issue of chain reorgs, this attack can be mitigated by having one canonical bridge between the two platforms. More precisely, the origination dApp has some clause in its own smart contract infrastructure that only respects tokens from the canonical bridge. Any sort of token transfer from some other incoming source would not be viewed as valid and at that point the dApp's smart contract could reject the transfer.

6.5 Concerns around user experience

Careful readers may have noticed many components to the system involving several transactions and token types. You may be concerned about the user experience for any particular instance of Polaris governance. We are firmly committed to producing high-quality user interfaces that abstract the low-level details around transactions and tokens so that end users can focus on the issues that matter to them: their community. Particularly in the case of tokens, we can take advantage of services like 0x and Kyber to source liquidity and frictionless exchange as required in-protocol. Analogues to these services can exist as special purpose functions on Polaris to facilitate the movement of some organization's governance tokens from their native platform to Polaris.

6.6 Comparison to other projects

Given our mission of providing quality governance tools for any and all projects we can compare ourselves to a number of existing projects, many of which are in development or have recently launched some mainnet product.

6.6.1 Interoperability

While interoperability is key to the Polaris vision, our focus is on governance. Interoperability is a means to that end. In this category we could be compared to the projects Polkadot or Cosmos. From our current understanding, it seems that the emphasis for Polkadot is on interoperability between the parachains (as facilitated by the relay chain) with less emphasis on any bridge parachains. While this may simply be a reflection of order of operations, Polaris emphasizes the bridge parachains as the top priority to provide the governance modules to existing platforms and dApps. Cosmos seems to have a stronger focus on communication between existing chains with their positioning as the "internet of blockchains" although they seem to focus on facilitating general communication and not focus as much on facilitating the governance of the protocols that move through it.

6.6.2 Governance

There are a number of dApps on the Ethereum chain committed to providing governance for organizations in that ecosystem with a similar focus on modules for on-chain governance. We are aware of: Aragon, DAOStack, and GovBlocks in this category. We applaud the efforts of each of these projects and hope we can work productively with them as we all explore the broad design space of decentralized governance. As discussed earlier, it is our position that we want a space for any community's governance that is independent of the underlying technical or political considerations of some other platform. For this reason, a core tenet of the Polaris project is to remain neutral to realize this space. This mission also requires a smart contracts system independent from any existing chain like Ethereum or EOS.

6.6.3 Futarchy

Given that futarchy is our flagship governance module, we will list some exciting efforts in the space: we have seen interest from Gnosis into the feasibility of applying futarchy for governance, we have seen some activity from the team LevelK via an Aragon Nest Grant, and are aware that the project Amoveo is interested in it. The Tezos project is interested in developing futarchy for their own version of on-chain governance in their protocol.

We have little data from live cryptoeconomic experiments to support or reject the expectations of futarchy-based governance. We hope to work with these teams to drive the state of knowledge about futarchy forward. If you are interested in helping us with a futarchy pilot program, please get in touch!

7 Conclusion

We have presented Polaris, the governance layer for the decentralized web. The main goal of the system is to provide public infrastructure to facilitate the decentralized governance of any blockchain projects, present and future. We only see the role of decentralized governance growing over time, especially as more and more of the fledgling projects in our ecosystem move past the bootstrapping phase. We hope you are also excited by this vision and want to join us! To use an oft-quoted passage: if you want to go fast, go alone; if you want to go far, go together.

We want to hear your questions, comments and concerns. Please email us at info@getpolaris.io or join our Telegram group @polaris_network to get in touch.