The Graph Tokeneconomics Notes/Research

1. How should query services be priced (at the network level or through free market mechanisms)?

Overview

The goal of pricing query services is to ensure that graph nodes remain agnostic to the subgraphs they power to maintain a liquid compute infrastructure, and that the allocation of compute resources across all subgraphs is distributed proportionally to the amount of queries they need.

Price-per-query payments allow for query providers to optimize business margins around service volume. There are two options: stable, where the price/query is set at the network level and enforced by all participating graph nodes, and free market, where the price/query is variable and set by query providers competitively. When deciding on a payment mechanism, we must consider the impact on optimizations regarding both query providers and end users alike.

Decision Factors

Value capture for query providers.

In free market pricing, query providers are responsible for pricing their own queries. For them, this exposes a host of new concerns and opportunities compared to stable pricing. Operators have the opportunity to make higher margins in more “in-demand” subgraphs assuming the query demand curves in networks remain stable. In other words, operators should only switch subgraphs if the query throughput on the new subgraph is comparable to the original one, relative to query price increase.

However, free market pricing mechanisms also require query providers to generate pricing models algorithmically (or manually with traders) that actively manage the price efficiency of the network. The impetus is now on the providers to create pricing mechanisms that allow them to capture additional value through market efficiencies created by the free market nature of query pricing. As a result, query providers are expected to build and maintain tools and invest human capital costs if they want to take advantage of this pricing model.

A free market pricing mechanism also creates a new risk vector for query providers. Now, they not only have to speculate on the demand volume of a given subgraph, they also have to consider how many other providers will want to switch their compute infrastructure over to the new subgraph and factor that into their pricing models.

In reality, most query providers are dapps that need hosting/indexing infrastructure to power their projects and are less price sensitive to opportunities in the market. A dapp will likely not switch their hosting infrastructure for a 30% increase in earnings in the early/intermediate stages of the network simply because they believe the value proposition for available queries on their applications will be worth more to them. Until query providers serve queries at or near bandwidth saturation (they are serving as many queries as possible given the resources they own), earnings opportunities are primarily determined by query demand within certain subgraphs. As a result, for the foreseeable future (until query providers are serving at or near capacity), earnings opportunities for graph nodes will be primarily in serving high-volume dapps over finding query price efficiencies, and the end impact on operator behavior will remain the same for free market and stable query pricing. As a side note, we are not sure if dapps will ever even want to run compute at saturation since browsing volume itself varies, and they would be losing significant margins if they don't have excess capacity to serve queries in a surge.

Smooth end user experiences.

For users, their main concern will probably be availability of access to dapps and predictable cost curves. While it's possible to achieve this in free market query pricing through query derivative and futures products, the network must now account for derivative liquidity and additional market complexities offloading query risk away from the end users.

Ultimately, free market pricing also creates adverse incentives and worse user experiences. Because compute costs track subgraph demand relative to compute resources available, end users may see pricing spikes during normal browsing due to other subgraphs gaining popularity. End users may now have to speculate on their own usage of dapps and at what time given the state of the network, since browsing at night may be cheaper than browsing during the day.

Conclusion

We believe queries should be priced at the network level to start for a few reasons. First, the development overhead is significantly lower as a network level-setting can be set and adjusted within the protocol to account for overall compute supply/demand, and the allocation of resources will still track query volume demand within the subgraphs. Second, we remove a major cost overhead (effective pricing and risk models for query providers) in order for them to enter the market. Last, but certainly not least, users have more predictable cost models for their browsing and don't have to worry about price spikes based on other users' usage. Ultimately, we do not believe the cost overhead of free market query pricing will be accounted for with the increased market efficiency—especially given that most query providers to start will be the dapp developers themselves.

Additional Notes

1) Network risk from market manipulation in low volatility/liquidity situations—query providers may collude to charge much higher in subgraphs that they own until competitors come in to offer more competitive pricing. While theoretically this shouldn't happen, the altcoin market is an example of a few parties manipulating the markets because there isn't enough liquidity to fill the economic inefficiencies.

2) If free market mechanisms still seem appealing to implement, one option is to create a price ceiling on resources. Bluezelle is a project that is doing just that.

3) Keep Network is another project that identifies the problem of price discovery and offers a marketplace as a potential solution, but highlights the different risk vectors (sybil attacks and price manipulation) in implementing such a market, and has opted for a randomized selection mechanism to match service providers and customers.

4) Vitalik covers a lot of the cost overhead issues in a github issue, "Evaluate alternative transaction fee market mechanisms".

2/3. Should there be a protocol tax? Why? How, if at all, should query services be taxed by the system?

<u>Overview</u>

In order to fund ongoing development and maintenance of the Graph infrastructure, the foundation must develop strategies to maintain consistent cash flow to pay for engineering improvements and other developments to the ecosystem (e.g. community grants, bonuses). The most common solution is to impose a tax or fee for ongoing transactions or usage. We believe that imposing taxes on an ecosystem is important and necessary to its success because infrastructure will never be perfect and because there will always be maintenance needs. However, the method of taxation and how its spent is critical to ensuring that it is a useful mechanism for the ecosystem. The absence of a tax is better than having one and allocating the funds poorly.

To create a sustainable tax on the ecosystem, value added through efficiency improvements to the infrastructure and additional sources of market liquidity from allocation of tax expenditure should be greater than the overall value extracted from the system via tax. If more value is taken from the system then generated from it, the system will collapse over the long run.

<u>Decision Factors:</u>

Use of funds.

The justification of tax is dependent entirely on the use of funds relative to its impact on network value. Therefore, I think it’s important to know how funds will be used, at what rates, for what improvements, and the estimated impacts of those improvements on the ecosystem before determining a tax rate. It should also be determined which augmentations to the network should be left to businesses to capture value through out-of-protocol development (private sector) and which aspects of the system lack profit incentives but benefit the ecosystem as a whole (public sector). Most importantly, use of funds should remain relatively riskless because risk is systemic when speculating with tax funds. For example, speculating on various subgraphs with tax funds is ineffective spending because the entire ecosystem pays for unpopular subgraphs.

We have identified three potential opportunities to use tax funds. The first is to maintain the open source codebase and make infrastructure improvements to increase the protocol efficiency. This will reduce systemic costs when running subgraphs or query nodes and give all supply-side network participants equal increase in value relative to the resources they dedicate.

The second is to reduce barriers to entry to the market. For example, building extensions for hosted query node infrastructure that allows less-technical parties to spin up, pay for, and manage query node infrastructure. By giving more parties access to the market, the overall market liquidity increases significantly, helping compute resources be allocated more effectively across the entire network. This will also create upward pressure on the token value as participation in the network becomes more competitive.

The third is to invest in better market standards. These can be things like basic query node operator requirements, minimum hardware specs, or query provider matching mechanisms (improvements to the gateway infrastructure). Reducing the overall performance volatility between the query providers may help them price more efficiently (if using the free market pricing mechanism) or manage their infrastructure better.

Tax pricing and denomination.

When considering tax implementations, it's important to understand how the rate is determined. For example, one solution is to have on-chain governance to set tax at the protocol level. Another is to just have a standard imposed by the foundation. A third is to remain as an optional fee similar to the fee mechanism for 0x relayers.

Another factor to consider is the denomination and structure that taxes are levied in because they determine who pays the network and how much they pay. If taxes are collected as transaction fees of queries (made in stablecoins through payment channels), the taxpayers would be the query nodes and taxed relative to the volume they serve on the network. Some projects collect taxes via a portion of the block rewards (DASH collects 10%), making the miners the effective taxpayers in this case, paying taxes proportional to the ownership they have in the infrastructure.

Conclusion

As mentioned previously, the justification of taxes in a system is dependent on the effectiveness of their expenditure. As such, we believe that the Graph team should decide on a policy (or lack thereof) based on the philosophical principles and beliefs of the team regarding effective use of funds.

Additional Notes

1) Grants are a great use of tax funds as well. Many projects have great grant programs, like Aragon's Nest.

2) Another project, Cosmos, applies a 2% reserve tax on their network for their reserve pool.

4. How much inflation should the system have and where should it flow to?

Overview

Usually the goal of inflation in a monetary system is to incentivize movement of assets within its ecosystem and discourage hoarding. By increasing the monetary supply, holders are encouraged to proactively spend their money and facilitate the flow of capital throughout an economy. In Ethereum and Bitcoin, the overall token supply is inflated to reward the nodes powering the infrastructure. Inflation is just another tax policy where the taxpayers are the token holders at rates proportional to their ownership of the token.

In the Graph ecosystem, however, an open question remains: are Graph tokens considered monetary instruments? The decision to impose an inflationary mechanism on the system is dependent on the philosophical interpretation of the token's meaning within the ecosystem.

Decision Factors

The utility of the token.

Given that Graph tokens exist primarily as a means to establish query nodes on the network, they are not a monetary instrument. Monetary instruments are generally used for exchange of goods or resources. This seems true: the value of the network is encapsulated by the total service volume (network-aggregate query rewards), rather than the market size of all the purchasable goods and services using the Graph token.

The desired relationship of the value of token to other cash instruments.

The intended behavior of the Graph token is to appreciate relative to the dollar over time as a reflection of the growth in service value from the Graph ecosystem. As a result, there is no attributable gain to the ecosystem using an inflationary mechanism because there is no direct need to induce velocity in the token. In monetary systems, an even distribution of money is important because money is meant to be spent. Graph tokens, however, are being used as collateral in exchange for rights to provide a service (more like a license). Distributions of Graph tokens should not affect the network unless the distribution is so significantly skewed that the entire market is dominated by very few players.

Conclusion

The Graph Protocol should not include inflation, especially if are already other forms of tax. Ultimately, the effect of inflation — creating sell pressure to induce higher velocity and more even distribution of tokens—has no direct causal relationship with the high-level goals of the network, which is increased output of query services to the dapp ecosystem. One of the justifications for inflation is as reward for block producers. However, the concept of block rewards is decoupled fundamentally from token inflation—inflation is just a form of tax, and there are other forms of tax previously mentioned that are suitable for incentivizing block production and general maintenance of the node infrastructure.

One possible implementation is for a tax to be levied on query providers as a fee per query. The tax can be calculated when providers exit their state channels, and is then pooled and released to index nodes as block rewards. Even more specifically, each subgraph can have their own tax pool, allowing block rewards for indexing nodes to stay proportional to overall query revenue.

Additional Notes

1) If inflation is present in the system, there are many types of inflation curves to consider that each have their own effect on the value distribution to the network. Linear inflation (constant per block), for example, incentivizes early adoption as the inflation rate decreases over time towards zero and should flow to those who contribute to the value proposition of the network (serving queries). Percentage-based inflation allows for more even long-term distribution (this should be used where even distribution is useful, like monetary instruments—not tokens). Inflation can also decrease over time until it settles at zero like Bitcoin, which carries extremely strong adoption incentives. However, this may disproportionately skew the distribution of wealth towards the early adopters and create fragile and volatile markets as the supply is generated dominated by a few parties.

5. How is staking used in the system? Who should stake tokens? How much should they stake?

Overview

Staking tokens generally reflects the opportunity cost of receiving rewards from the network. In general, staking mechanisms are used to align upside incentives through ownership ("skin in the game") and to discourage bad behavior with slashing conditions. As a function of being the recipient of network rewards (payments for queries), query nodes are the most natural use case for staking tokens in the Graph ecosystem—they represent a provision of services on behalf of the network and should demonstrate ownership and be held accountable for the services they provide.

Decision Factors

Relationship of token price to demand for services in the network.

Fundamentally speaking, there are a few token price mechanisms that should be fulfilled when considering the design. First, the token price should correlate to the demand for services on the network—that it essentially represents opportunity cost to be a service provider at a given volume. This is important from a security perspective: the more query volume a given provider serves, the more accountability that provider should have on the network. Currently, a query node that serves queries on the network can freely serve at any load under any token staking amount. This design is flawed as a query providers with few tokens staked can serve orders of magnitude more queries than providers with a significantly larger amount of tokens staked. As a result, the tokens do not represent cost of access to services in the network well and may become problematic later when the node operators' service ecosystem matures.

Relationship of token price of quality of services in the network.

The value of the token must properly reflect the value in the quality of service within the network. For example, if the Graph network's aggregate query volume does not change, but each query is served with significantly better latency, the price of the token should be higher in the latter network case. In order for this to be the case, rights to provide query services must be attached with an opportunity cost denominated in Graph tokens. If a query provider believes that he can provide a better service (lower latency, higher uptime/infrastructure reliability, lower hosting/compute costs), they would purchase tokens at a higher-than-market price to capture additional margins and profits from the existing query demand and increase the overall standards of the network as a result.

Conclusion

We believe that staking in the system should be used by Graph node operators who want to participate in serving queries. Each token can be set initially to represent some bandwidth of queries served (1 Graph token = 10,000 queries/1024 blocks), which should also create a base value peg for Graph tokens relative to expected demand for efficient market pricing. This mechanism also aligns a lot of network level economics: subgraphs with lower expected query volume, for example, will require less staking overall to serve the market demand of the particular subgraph.

In the case that the market hits a saturation point (more demand for queries than the amount of Graph tokens in existence), the token capacity can be adjusted upwards either through on-chain governance mechanisms or some aspect of foundation management. For example, as a way to increase market liquidity, each token may serve 12,000 queries/1024 blocks, increasing the utility value for all shareholders and increasing the network capacity as a whole. This is one possible fair implementation as all the token holders are rewarded for their early participation and adoption in the network prior to saturation. Token holders are rewarded either by increasing their service capacity and earning opportunities, or by being able to sell off excess capacity they no longer need in exchange for a nice payday.

6. How much should be slashed for various forms of malicious behavior in the system?

Overview

For the overall security and health of the network, slashing conditions are imposed on token stakers in the ecosystem to discourage bad or damaging behavior in the system. Slashing conditions were first created by Vitalik to solve the “nothing-at-stake” problem by punishing validators trying to compromise network security by signing on multiple chains. In the case of the Graph, the quality of the network is based on the quality and quantity of the query services ecosystem, and slashing conditions should be designed to optimize around those properties.

Slashing conditions should be fair and comparable to the damage of the associated behavior. Ideal slashing rates should be representative of the expected value attrition due to the associated behavior. Excessive slashing conditions, for example, create unnecessary risk for operators and may discourage participation due to the skewed expected value from participation in the network. Lenient slashing conditions, however, expose the network to malicious attacks and bad actors because of the relatively low cost to attempt an attack.

Decision Factors

Expected value loss from bad behavior on the network.

The amount of tokens slashed should be dependent on the effect of the behavior on the network and the level of harm. For example, minor service interruptions may be caused by behaviors such as missing votes, not relaying data, node downtime, etc. that affect the overall quality of the network. Slashing conditions for these service interruptions should reflect their overall impact on the network, such as incurring some form of penalty—but not enough to discourage query providers from being willing to take the risk of unexpected service interruptions. Most importantly, as mentioned in the previous question regarding token staking, the decrease in quality should translate to some value decrease in the network.

We see two primary classifications for network behavior regarding slashing conditions: “crash tolerant” and “fault tolerant” behavior. Crash tolerant behavior define actions that cause network-related service issues such as the ones previously mentioned (missing votes, not relaying queries, and server downtime). These slashing conditions should be adjusted based on what your team deems to be the acceptable standard for node operations: what is a tolerable, or rational expected downtime of a node? High expectations regarding infrastructure uptime probably results in higher slashing conditions for going down, and vice versa.

Fault tolerant behavior defines actively malicious behavior that attempts to sabotage the efforts of the network. Examples of this behavior are things such as double-signing on separate chains, attempting to collude with other block producers, etc. We believe that fault tolerant behavior should be severely punished and that all malicious actors should be removed from the ecosystem entirely. Malicious actors create systemic risks, costs, and other issues for the network with no possible upside, warranting high slashing conditions.

Distribution of slashed tokens.

Once tokens are slashed, it's important to consider the distribution/disposal mechanism of slashed tokens. Observing who is impacted after a slash and in what way is crucial to implementing effective slashing conditions.

One option is for slashed tokens to go to a community pool as a taxation scheme. Assuming the taxes are reinvested in protocol enhancements, low quality service providers in this situation are essentially required to invest an additional portion of their profits into the development of network infrastructure. As a result, query provider in the ecosystem benefits from efficiency increases, which may even help improve the quality of service of the slashed parties. This seems to be a fair redistribution of value within the ecosystem.

Another option is to burn the slashed tokens. This benefits token holders directly and proportionally to the amount slashed. Philosophically, this makes sense if the ones most at risk from poor service infrastructure are the token holders. However, we believe poor service infrastructure may damage the value attribution of end users as well, who may or may not hold tokens, and most likely will not be rewarded from direct burning of slashed tokens.

The third option is for the slashed tokens to go to the receiving end of poor service. Service customers may be entitled to a credit of tokens, for example, if a query provider fails to provide requested query services for the associated subgraph. However, this implementation may also enhance adverse incentives, like customers intentionally trying to scam the network to receive credits by misreporting query services or intentionally trying to damage query infrastructure (DDOSing their query providers, for example, is one way they can rack up easy credits and benefit personally while creating a net negative effect for everyone else on the network). There is also the question of how the credits are denominated—if Graph tokens are the rewarded unit of credit for consumers and they don't use Graph tokens to host query infrastructure, it does nothing but increase the velocity of the token as customers generate sell pressure from their 'refunds'.

<u>Conclusion</u>

Rather than impose nominal slashing conditions for the Graph network, we think it's best for the Graph team to evaluate different actions on the network and estimate value attrition for each one. Because it's difficult to model intrinsic value, we believe that the determination of slashing costs should be set by the Graph foundation or through some other form of governance. However, we do believe it's better to err on the side of slashing conditions that are too harsh than too lenient, since overly harsh slashing conditions may exclude risk-sensitive query providers and the network would miss out on improvements to the service infrastructure, whereas overly lenient slashing conditions may include risk-sensitive malicious actors and the network would be more susceptible to attacks or other compromising behavior that would otherwise have been discouraged.

Dissent

There were few dissenting opinions on this topic when conducting internal review of this document. An opposing viewpoint is that a bigger threat to the success of the Graph network is not getting enough adoption at the outset. The belief is that not all attacks will be effectively mitigated through harsh slashing conditions, as certain attack vectors (e.g. competitive networks, economically irrational actors) will be agnostic to the slashing conditions, and a single attack on the network will have the same economic impact. As a result, lowering slashing conditions is more prudent as it may encourage initial adoption through decreased risk for query providers to enter the market. Over time, slashing rates could then be increased to match the actual value loss of a slashable condition.

7. How should Graph tokens be distributed for maximum participation in the network?

Overview

The goal of distribution regarding resources is to have them end up in the hands of those who can most effectively utilize them to generate value. As a result, strategies should be mostly geared to people who can effectively stake Graph tokens to be query providers on the network. Tokens can be distributed through a variety of strategies, including token sales, airdrops, grants, etc., but the means of distribution is not as important as the mechanisms behind them, so attention should be placed in the incentives behind those who are receiving the token and how they will react upon receiving the grant.

Decision Factors

Incentivizing usage on the network

The first point of consideration is regarding the definition of “participation” within the network. Users and query providers are both participants, and both aspects can be incentivized with Graph tokens. However, given the existing token model, users don’t have a need to hold Graph tokens (unless they are paying queries for them, which is a mechanic that I think should be avoided at all costs), and should not be directly incentivized through token distribution mechanisms.

Creating distribution channels based around usage (queries served by a provider, etc.) may still generate strong incentives for service providers and stakeholders to direct end users to the network. Referral networks, user onboarding bonuses, and other strategies have been used by a variety of blockchain and non-blockchain startups to great effect.

One possible option is to create grants or airdrops that are directly correlated to actions performed on the network. For example, during a specified incubation phase of the network, or at launch, the Graph foundation can offer token grants relative to the total queries served to the end users for applicable service providers. Eligible receivers may need to be applicants that specify intent ahead of time to reduce risk of sybil attacks or attribute legal risk to bad actors.

Another possible option is to create a “mining program” similar to Livepeer’s merkle-mining where individuals are incentivized to sign up and claim account space to receive a grant of tokens. In the case of query providers on the Graph network, nodes can be eligible potential bonuses for incentivizing new account signups and rewarded when those users are then verified through consistent querying. For example, an allocation of tokens can be granted to a query provider once a user has reached 100 queries on a given subgraph.

Bootstrapping query infrastructure

The other aspect of consideration is incentivizing infrastructure stability. Depending on which pieces are determined to be essential to the network, it’s possible for additional grants or alternative grants to be designed around infrastructure hosting to ensure high availability when users reach the network. While usage bootstrapping mechanisms may run into sybil attack issues, infrastructure bootstrapping can just be allocations of tokens set aside. Ideal targets may be rackspace companies, cloud providers like Digital Ocean, or even decentralized protocols like Akash so that users have reliable and consistent experiences from the very start. Filecoin, for example, has an early miner program where they ask potential providers to apply and specify the volume of storage they expect to provide to the network. Graph network can create a similar “early query provider program” that may even have the same addressable audience as Filecoin’s program, but as a grant application.

There are many incentives to consider even within creating distribution grants for bootstrapping infrastructure. Coverage, quality, and length are the three key aspects. Coverage is the amount of indexing/possible query range that providers can serve, and grants can factor in the amount of indexed data that a query provider is serving to ensure that all subgraphs receive relatively even attention from the various query providers receiving grants. Quality is the uptime & latency of the query servers, and length is the amount of time an outside party must serve to receive the grant.

Conclusion

Besides the sale, we believe tokens should be reserved for a variety of participation incentives in the form of grants or airdrops. These grants should incentivize usage increases, infrastructure improvements, and service provider adoption. Given that the economy of the Graph network is centered around the activities of query providers, they should be the primary driving force behind the growth of the network and the main recipient of the grants. As such, we believe that distribution should be maxed over the total possible query infrastructure market, rather than general distribution to end users or similar ideas.

We suggest an early query provider program similar to Filecoin, with the Graph foundation allocating grants ahead of time where early nodes providing graph queries can “attest” the queries they serve on the platform. Given that the majority of the service providers are initially dapp developers looking for an indexing solution, this grant can be offered or tailored to high-volume dapps that are already looking into running private subgraphs as a way to import the dapp’s query volume to the public query infrastructure rather than a private instance (e.g. Airswap runs a public query node on their subgraph and is rewarded relative to the usage of their dapp).

Additional Notes:

1) The only difference we recognize between grants and airdrops is the verification/distribution of tokens: grants are manual distributions enforced through legal frameworks and airdrops are automatic distributions enforced through cryptographic proofs.

8. How would you estimate token valuation given your answers to the above?

Disclaimer: Please see our long-form disclaimer in this document prior to reviewing our answer to this question. Keep in mind that we are not broker-dealers so we do not price securities or engage in broker-dealer-related activities.

In general, estimating token valuation depends on a number of metrics. First, given the utility of a token, what business activities can be accomplished from it? What is total estimated size of those business activities? In general, utility token models like the Graph token represent access to providing services on the the network, and as such, should be valued from a network-level perspective. Our suggestion is to research market assumptions (total addressable market, average service volume/provider, expected margins, etc.) to craft a profitability model for any given provider. From there, there are various traditional valuation strategies that can be applied to estimate token value.

Taxi medallions, real estate, and other cash-flow driven businesses often use DCF (discounted-cash flow) models to value the underlying assets. This strategy works by accumulating the discounted value of future cash flows into a single, net present value of the token. In our opinion, this strategy best describes the value captured by the token because it not only represents the current value of the network, but the potential future growth and value of services provided by the network as well.

The information set forth herein is proprietary and shall be maintained in strict confidence. Each recipient hereof acknowledges and agrees that the contents of this document (i) constitute proprietary and confidential information that Fraktal Group, LLC and its affiliates (collectively, "Fraktal") derive independent economic value from not being generally known and (ii) are the subject of reasonable efforts to maintain their secrecy. The recipient further agrees that the contents of this document are a trade secret, the disclosure of which is likely to cause substantial and irreparable competitive harm to Fraktal. Any reproduction or distribution of this document, in whole or in part, or the disclosure of its contents, without the prior written consent of Fraktal, is prohibited. This document will be returned to Fraktal upon its request.

Certain factual statements made herein are based on information from various sources prepared by other parties. While such sources are believed by Fraktal to be reliable, Fraktal does not assume any responsibility for the accuracy or completeness of such information.

Certain statements in this document constitute forward-looking statements. When used herein, the words "project," "anticipate," "believe," "estimate," "expect," and similar expressions are generally intended to identify forward-looking statements. Such forward-looking statements, including the intended actions and performance objectives of the relevant party referenced herein, involve known and unknown risks, uncertainties, and other important factors that could cause the actual results, performance, or achievements of such party to differ materially from any future results, performance, or achievements expressed or implied by such forward-looking statements. All forward-looking

statements in this document speak only as of the date hereof. Fraktal and its affiliates expressly disclaim any obligation or undertaking to disseminate any updates or revisions to any forward-looking statement contained herein to reflect any change in its expectation with regard thereto or any change in events, conditions, or circumstances on which any such statement is based. Furthermore, nothing contained herein is, or should be relied upon as, a promise or representation as to the future performance of Fraktal or any of its affiliates.