Pyth Network 101

A curriculum made by a passionate community member. She also said, “seriously, just read the whitepaper”.

Summary

The Pyth network is **a first-party financial oracle network** delivering 250+ real-time price feeds across cryptocurrencies, FX rates, ETF’s, equities, and commodities to 20+ blockchains.

The network incentivizes market participants — exchanges, market makers, and financial services providers — to share directly on-chain the price data collected as part of their existing operations. The network then aggregates this price data on-chain and makes it available for blockchain and off-chain applications. Pyth sends this data to multiple blockchains through Wormhole, the cross-chain messaging protocol. Applications then “pull” Pyth prices onto their native blockchain when needed thanks to Pyth’s on-demand (or “pull-based”) oracle design.

Pyth provides unique features including confidence intervals (a plus-minus confidence band around reported asset prices), access to historical prices (for uses like options settlement), and a liquidity oracle (market impact information).

Major users of Pyth include Synthetix on Optimism, CAP Finance on Arbitrum, Ribbon Finance on Ethereum, and even TradingView.

▼ **Further reading:**

- TL;DR
- What is Pyth Network?
- Pyth is fast
- What is Pyth (video)

- Why is Synthetix using Pyth?

The Press Release (PR) Version

The Pyth Network is the largest first-party financial oracle network designed to publish continuous real-world data on-chain in a tamper-resistant, decentralized, and self-sustainable environment.

The network incentivizes some of the world's biggest institutional market participants — exchanges, market makers, and financial services providers — to share their proprietary price data on-chain. Pyth continuously aggregates this first-party data on-chain for use by smart contract applications. Applications "pull" Pyth prices onto their native blockchain when needed thanks to Pyth's on-demand (or "pull-based") oracle design.

You can learn more about Pyth from their website and documentation.

History

Started: April 7, 2021

Devnet: May 15, 2021

Mainnet: August 26, 2021

Whitepaper: January 18, 2022

On April 7, 2021, Dave Olsen, President of Jump Trading, announced on The Jump Off Point podcast that Jump Trading is "collaborating on a world oracle project called Pyth". Pyth's live devnet prices were revealed in mid-May, showcasing Pyth's live streaming and sub-second price update capabilities. The Pyth network officially launched on Solana mainnet on August 2021.

Pyth finished **2021** with over $1B in Total Value Secured (TVS), and facilitated over $7B in trading (including perpetual and synthetics platforms), supporting 38 applications and welcoming 41 data providers. By March 2022, Pyth reached $2B in TVS.

On **January 2022**, the Pyth Data Association released the Pyth network whitepaper. The whitepaper outlines the planned design and mechanisms around the PYTH token and its role in making the network self-sustaining and decentralized.

By May 2022, Pyth had secured 90% of the Solana addressable market and was ready to go **cross-chain**.

On August 2022, the Pyth network officially announced **Pythnet**, a network built on the Solana codebase that enables the Pyth network to deliver pricing data to other chains via Wormhole.

On October 2022, Pyth went live on BNB Chain and Aptos, with teams like Venus and Wombat excited to use Pyth data. On November, Pyth data went live on Ethereum and Optimism: Synthetix decided via public governance to integrate Pyth data into their Perps V2.

By the **end of 2022**, the Pyth network supported over $35B in total trading volume, and welcomed nearly 80 data providers (1 new data provider *every week*), offered 185 price feeds, and supported more than 100 applications across 8 blockchains.

On average, Pyth data goes live on a new chain every month. (Compare this to legacy oracles such as Chainlink, which took more than 2 years to deploy on Solana mainnet).

As of January 31, 2023, Pyth accumulated over 100,000 cross-chain price updates—that's a lot of demand for Pyth's 200+ price feeds!

By **Q2 2023,** the Pyth Network had more than 80 data providers supporting 250+ price feeds and 150+ integration partners, facilitated nearly $50B in total trading volume and serves 22 blockchains.

▼ **Further reading:**

- Pyth in Q1 2023
- Pyth in FY 2022
- Charts: How fast Pyth has grown

Pyth Network Products

Pyth offers real-time **price feeds** for crypto, US equities, FX pairs, and commodities (metals). Let's take the Pyth BTC/USD feed as an example.

The Pyth feed may say the price is $20000 ± $50.

The feed is published on-chain, where it is read by data users (”consumers”), who are either blockchain-enabled programs or off-chain applications.

Pyth’s on-chain program produces this price feed by aggregating prices from multiple data providers (”publishers”).

The Pyth program is built on an appchain called Pythnet to store and update the state of each price feed. Pyth prices are broadcast from Pythnet to other target chains using Wormhole, a cross-chain messaging protocol.

There’s many benefits to using Pyth over other oracles. Each price feed has the following features:

- ▼ **Confidence Intervals**

  Remember how the BTC/USD feed example said the price was $20000 ± $50?

  $20000 is the price. The $50 band is the confidence interval which represents the estimated uncertainty of the price. A tighter or smaller band implies greater certainty. No other oracles offer confidence intervals.

  The confidence interval represents the width around their price estimate in which the data providers (in aggregate) believe the true price probably lies. Different data providers have access to different types of data and may have different methods of calculating their price estimate and confidence.

  This confidence value informs data users about the perceived strength of the price output. Pyth empowers data users by continuously publishing a consolidated estimate of these reported price dislocations: Data providers can acknowledge a low liquidity environment on the trading venue and adjust their confidence accordingly. Applications can respond to this information for greater platform security.

  - ▼ **Examples:**

    Synthetic asset platforms can choose to scale liquidity at a price with the confidence reported rather than allow infinite liquidity for mints/redeems on every price (and confidence level).

    Borrow-lending platforms can choose to withhold or adjust liquidations when collateral prices are reported to have wide confidence bands. Watch how Jet uses confidence intervals.

▼ **Further reading:**

- What is confidence?
- Pyth primer: be Pyth confident!

▼ **Price and Confidence EMA**

Pyth offers an exponentially-weighted moving average (EMA) price and exponentially-weighted moving average confidence (EMAC). These values are time-weighted averages of the aggregate price and confidence. Some applications need a rolling, time-weighted average of the price for safe operations.

▼ **Let's get technical:**

In an EMA, the most recent samples receive the most weight, and samples further back in time get exponentially less weight the farther in the past they are.

While conceptually not as simple as an SMA (Simple Moving Average), the EMA has a particularly simple implementation for streaming applications such as Pyth. The exponential weighting method allows the entire history of prices and weights to be represented by a single number.

▼ **Further reading:**

- What is the Pyth EMA?
- EMA in the GitHub

▼ **Low-Latency, High-Frequency Updates**

Speed matters. Price feeds on Pythnet update faster than once every second, which is faster than the block time of most blockchains.

Fast updates and low-latencies (time it takes for updates to reach final user) allow apps to track the market more closely for more accurate and secure smart contract operations.

▼ **Availability Everywhere**

Pyth's catalog of 250+ price feeds is substantial. When Pyth deploys a price feed, it is immediately available on all 20+ supported chains. This is not the case with other legacy solutions, which must deploy that feed one-by-one on every destination chain.

For example, when Pyth released its ARB/USD price feed on the same date as Arbitrum's mainnet launch, 20+ blockchains had exposure to ARB/USD and perpetual trading protocols from Optimism, Arbitrum, Solana and Aptos began plugging in. Compare this to Chainlink, who had to expedite their roadmap for ARB/USD and released theirs for Optimism only.

▼ **First-Party Data**

Pyth's market data is contributed by over 80 first-party data providers (meaning they create and own the data), including some of the biggest exchanges, market makers, and trading firms—institutional players who truly know the prices of assets.

Third-party data means data that is purchased or scraped from another first-party provider: e.g. Chainlink sources include CoinGecko and CoinMarketCap.

▼ **On-Demand Updates**

The Pyth oracle network utilizes an on-demand price update model or "pull" update model. Unlike legacy oracles which continuously "push" prices on-chain for consumption, Pyth lets data users from any blockchain "pull" the price when they need it.

Each Pyth-supported blockchain has a Pyth contract which stores the price for each price feed Pyth offers. Anyone on that blockchain can submit a valid Wormhole message to the Pyth contract to update the price. Data users typically update the price and use it in a downstream application.

You can read about the advantages of the Pyth pull model below.

▼ **Further reading:**

- A great explanation in the docs
- A new model by Pyth. Legacy oracles are copying Pyth
- A summary thread

▼ **Transparent Aggregation**

Pyth's price aggregation mechanism combines the prices and confidences submitted by individual data providers into a single aggregate price and confidence.

Aggregation is *extremely* important: the aggregation mechanism guards against the price manipulation attacks by limiting each data provider's influence on the aggregate price.

Because the aggregation happens on-chain, the provenance of every data point can be traced: the output is verifiable and the process is secure and transparent.

▼ **The math behind it:**

The first step of the algorithm computes the aggregate price by giving each publisher three votes — one vote at their price and one vote at each of their price +/- their confidence interval — and by then taking the median of all the votes.

The second step computes the distance from the aggregate price to the 25th and 75th percentiles of the votes, then selects the larger of the two as the aggregate confidence interval.

Overall, the aggregate price will always lie within the 25th-75th percentile of the publisher's prices. The docs can into more detail.

▼ **Staking in the whitepaper:**

The Pyth whitepaper describes a staking system for publishers which incentivizes them to provide accurate data. In this system, each publisher would have a varying amount of stake. All of the results also hold for stake weights if Pyth replaces the percentage of publishers with the percentage of stake controlled.

In the future, the weight calculation could be extended to include other non-price factors such as publisher's stake, historical publisher's performance, and other relevant metrics.

▼ **Further reading:**

- A great explanation in the docs
- Pyth price and confidence aggregation

▼ **Where are Pyth Price Feeds Available?**

- Aptos
- Arbitrum
- Aurora
- Base

- BNB Chain
- Celo
- Conflux eSpace
- Cronos
- Ethereum
- KuCoin Community Chain
- Meter
- Injective
- Optimism
- Osmosis
- Polygon
- Polygon zkEVM
- ShimmerEVM
- Solana
- Sui
- zkSync Era

How Pyth Works

Pyth is a protocol that allows market participants to publish pricing information on-chain for others to use. The protocol is an interaction between:

Data Providers (”Publishers”)

Data providers submit pricing information to Pyth's oracle program. Pyth has multiple data providers for every product to improve the accuracy and robustness of the system. More is better!

These data providers include some of the biggest exchanges, market makers, and trading firms in the world. These institutions actively participate in price discovery and thus have access to accurate, timely price information. They *really* know the prices of different assets. As creators and owners of their own data, they are *first-party* data sources.

By having a diversity of data providers (US accredited exchanges, crypto exchanges, trading firms, financial services providers), Pyth creates a completely new composite market data stream at quality levels previously inaccessible to most people.

▼ **Further reading:**

- See the full list of data providers
- See how well each data provider is doing
- Law of big numbers: more data providers is better

Pyth On-Chain Program

Data providers publish their prices on Pythnet. Pyth's on-chain aggregation program then aggregates these prices (for each specific feed) to obtain the aggregate price and confidence. This program runs on Pythnet.

Pythnet is an application-specific blockchain operated by Pyth's data providers. This blockchain is a computation substrate to securely combine the data provider's prices into a single aggregate price for each Pyth price feed.

Thanks to Pythnet, Pyth is able to stream data at a sub-second latency and at affordable costs. The network's data providers can update prices at every Pythnet slot—currently once every 400ms.

▼ **A small note:**

Pyth's oracle program runs simultaneously on both Solana mainnet and Pythnet. Each instance of the program is responsible for its own set of price feeds.

Solana Price Feeds are available for use by Solana protocols. In this case, since the oracle program itself runs on Solana, the resulting prices are immediately available to consumers without requiring any additional work. On Solana, Pyth prices update every 400ms. On other blockchains, the delay between the Pythnet price update and that price point

Pythnet Price Feeds are available on multiple blockchains. The prices constructed on Pythnet are transferred cross-chain to reach consumers on these blockchains.

Pythnet forms the core of Pyth's off-chain price feeds that serve all blockchains (except Solana mainnet).

▼ **Further reading:**

- The docs explain Pythnet
- Pythnet blog

Data Users (”Consumers)

Data users read the price information produced by the oracle program.

Pyth needs a cross-chain component to ferry prices on Pythnet to target chains. Pyth uses a “pull” update model for target chain prices: instead of continually pushing updates to each target chain, users pull the prices on-chain when they are needed.

There are over 100 identified Pyth integrations on Ethereum, Optimism, BNB Chain, Solana, Aptos, and Aurora, as well as off-chain apps. Pyth users include borrow-lending platforms, synthetics, derivatives trading, decentralized options vaults, stablecoin protocols, and more.

▼ **Let’s get technical:**

The Pythnet attester program regularly attests to the most recently observed Pyth prices and creates a Wormhole message to be sent to the Wormhole contract on Pythnet. The Wormhole guardians observe the attestation message and create a signed VAA (wraps the message in a structure combining the message with the guardian signatures to form a proof).

A price service API continually listens to Wormhole for Pyth price update messages. It stores the latest update message in memory and exposes HTTP and websocket APIs for retrieving the latest update. (Anyone can run an instance of this webservice; the Pyth Data Association runs a public instance for convenience.)

When a user wants to use a Pyth price in a transaction, they retrieve the latest update message (signed VAA) from the price service and submit it in their transaction. The target chain Pyth contract will verify the validity of the price update message and, if it is valid, store the new price in its on-chain storage.

▼ **Further reading:**

- The docs explain Pyth’s cross-chain design

Pyth Network Participants

Data Providers (”Publishers”)

Data providers publish price feeds and earn a share of fees in exchange. As explored in the whitepaper, data users will reward data providers in proportion to the quantity of new pricing information that they share (and reduce the likelihood that uninformed publishers will earn rewards).

The whitepaper presents a staking and slashing system wherein publishers will be required to stake a minimum quantity of PYTH tokens per price feed (”product”) they support. These PYTH tokens will form a portion of the assets available for payouts but will only be slashed if the publisher publishes an inaccurate price. In other words, the protocol will only punish publishers for failures if the on-chain mechanism determines that they are at fault.

Users (”Consumers”)

Data users read price feeds and incorporate data into smart contracts or dApps.

As explored in the whitepaper, consumers will have the ability to pay data fees to become eligible for a payout in case of proven oracle failure.

The Pyth protocol has also been designed to allow for the optional enablement of data fees for updating the state of an on-chain price feed. The ongoing existence of and size of the fee will be determined by governance on a per-blockchain basis. Until governance is live, the fee will be 1 of the smallest denomination of the blockchain's native token (e.g., 1 wei on Ethereum). The fees collected by the protocol would go toward compensating data providers and possibly other uses as determined by governance.

Delegators

As explored in the whitepaper, Delegators will bet incentivized to participate in the protocol to earn data fees (coming from consumer data fees). Delegators stake tokens on a specific price feed (”product”) and publisher to (1) earn a share of the data fees and (2) increase the price feed robustness but may potentially lose their stake if the oracle is inaccurate. Staking increases that publisher's weight in the aggregation. The planned slashing mechanism would incentivize Delegators to spread stake more evenly to diversify risk.

Some Notable Pyth Data Users

▼ **BNB Chain**

- Venus V4
- Alpaca Finance
- LEVEL Finance
- Uniwhale
- Deri Protocol
- DeriTrade

▼ **Optimism**

- Synthetix
- Pika Protocol
- Lyra Finance
- Kwenta
- Polynomial
- Polynomial Protocol
- Synthex

▼ **Arbitrum**

- CAP Finance
- Vela Exchange
- Deri Protocol
- Duet Protocol

▼ **Ethereum**

- Ribbon Finance (DeFi Options Vault)

▼ **Solana**

- Solend
- Drift
- Mango Markets
- Tulip
- Lifinity
- Zeta Markets
- Bonfida (Solana Name Service)

▼ **Off-Chain**

- TradingView
- QCP
- Geneva Trading

Some Notable Pyth Data Providers

- CBOE (Chicago Board Options Exchange)
- Jane Street
- Optiver
- Binance
- OKX
- QCP Capital
- Akuna Capital
- CMS
- IMC
- Tower Research

- Wintermute

General Concepts (Glossary)

Oracles

Blockchain oracles act as the bridge between smart contracts living on a blockchain and real-world data. They are designed to relay information from off-chain sources (data outside a blockchain) to an on-chain network. In most cases, the off-chain data is used to trigger smart contracts that need external data to execute.

Borrow-Lending protocols, such as Solend, rely on oracle price feeds to determine the value of the underlying collaterals supplied and/or borrowed. Whenever the collateral provided is worth less than 1 + x% (usually x = 30) of the borrowed collateral, borrow-lending platforms must liquidate a user's position to keep the overall platform healthy (enough collateral to repay the outstanding debt).

Total Value Locked (TVL)

TVL represents the number of assets that are being staked in a specific protocol at the moment: this value is not meant to show the number of outstanding loans, but rather the total amount of underlying supply that is being secured by a specific protocol or application by DeFi completely.

Total Value Secured (TVS)

TVS represents the aggregate amount of Total Value Locked (TVL) within all protocols and platforms that depend on the proper operation of an oracle network to operate.

High Fidelity (HiFi) Data

Financial applications requires high-fidelity, time-sensitive, real-world data, which has historically been inaccessible on-chain. High-fidelity data refers to produced data that closely matches the operational context of interest. High-fidelity in this context of the financial markets means accurately reporting data in an accurate, timely fashion.

Contemporary music science introduced the concept of high-fidelity to audiophiles by demonstrating the importance of minimizing all distortions in sound reproduction. Simply put, sound engineers focused on reproducing music accurately (with more granularity) on new digital mediums.

First Party (vs Third Party) Data

First party data refers to information an entity collects directly and owns. In the context of oracles, a first party means the data source itself (e.g. the API provider itself) publishes data directly to the blockchain.

Third party data refers to data collected from another (third party) source. A third party oracle calls on another entity (e.g. calls an API) to retrieve the data and then publish it on the blockchain.

▼ **Further reading:**

- Pyth as a first party oracle
- API3 joins the discussion

“Pull” (vs “Push”) model

Pyth uses an on-demand price update model that is slightly different from other oracles you may be more familiar with. Most oracles today use a push model, where the oracle runs an off-chain process that continuously sends transactions to update an on-chain price. In contrast, Pyth network does not operate an off-chain process that pushes prices on-chain. Instead, it delegates this work to Pyth data users.

Pyth price updates are created on Pythnet and streamed off-chain via the Wormhole Network, a cross-chain messaging protocol. These updates are signed such that the Pyth on-chain program can verify their authenticity. Updating the on-chain price is a *permissionless* operation: anyone can submit a valid Wormhole message to the Pyth contract to update the price. Typically, users of Pyth prices will submit a single transaction that simultaneously updates the price and uses it in a downstream application.

This pull model is highly scalable and allows Pyth to deliver high-frequency price updates for a large number of products without overwhelming the transaction capacity of target chains (or incurring excessive gas fees).

▼ **Further reading:**

- The docs explain the pull model

Latency

The classical definition of latency is the delay between an instruction to transfer data and the transfer itself. In the context of blockchain, latency refers to the delay between a transaction submission and confirmation (or finality).

Low-latency data is strictly superior than higher latency data. However, the decentralized nature of blockchains (reaching consensus between myriads of nodes simultaneously) means processing is slower than e.g. AWS.

A blockchain oracle can only update the on-chain price as fast as new blocks are produced.

▼ **Further reading:**

- Pyth Intern explains latency

Financial Market Pricing Data

In finance, market data is price and related data for a financial instrument reported by a trading venue (e.g. a stock exchange). Market data allows traders and investors to know the latest price and see historical trends for assets/instruments such as equities, fixed-income products, derivatives, and currencies.

Financial market data can be simply defined as a combination of:

1. Orders at which prices traders wish to trade
2. The prices at which trades have just occurred

Both 1 and 2 are dependent on traders, and since traders typically match with one another at exchanges, first party data is generated by both traders and exchanges.

▼ **Further reading:**

- Pyth specializes in financial data
- Pyth supports the Dow Jones 30 equities

EMA

The EMA is a moving average that places a greater weight and significance on the most recent data points.

Like all moving averages, this technical indicator is used to produce buy and sell signals based on crossovers and divergences from the historical average.

Traders often use several different EMA lengths, such as 10-day, 50-day, and 200-day moving averages.

See the Pyth tool-tip in any price feed page: `The EMA (exponentially-weighted moving average) is a time-weighted average of the aggregate price. In an EMA, the most recent samples receive the most weight, and samples further back in time get exponentially less weight the farther in the past they are.`

▼ **Further reading:**

- Investopedia's explanation

TWAP/SMA

In finance, **time-weighted average price (TWAP)** is the average price of a security over a specified time.

TWAP can also refer to an algorithmic trade execution strategy that aims to achieve an average execution price close to the time-weighted average price of the user-specified period. A TWAP strategy is often used to minimize a large order's impact on the market by dispersing the large order into smaller quantities and executing them at regular intervals over time.

▼ **Further reading:**

- A guide by one among many financial firms
- Binance has a nice tutorial

The Price of an Asset

Consider a single stock, say TSLA.

“What is the price of TSLA?” is a seemingly simple question, but there are subtle complications.

Overall, there is no “one single price” for TSLA at any moment in time.

There is the best bid price (the price a person could currently sell TSLA at) and the best ask price (the price a person could now buy TSLA at). They are typically close together, but the difference between them (the so-called bid-ask spread) could indeed be viewed as the uncertainty of what the “price” of TSLA currently is, with the midpoint (halfway between the best bid and best ask) being an estimate of the price itself.

Another “price” that could be considered the proper price is actual transaction prices. “Last traded price” is a price that is often reported, and especially in a thinly traded stock, might be even more representative of the true price than the current best bid and best ask prices (where the bid-ask spread might be very wide).

▼ **Further reading:**

- A classic article on uncertainty and price theory by Pyth

Confidence (or “Uncertainty”) in a Price

In scientific and engineering fields, observation or measurement is almost always accompanied by a measurement of uncertainty. The distance between two points might be measured to be 10.12m +/- 0.01m. The time it took for a car to travel that distance might be 1.23s +/- 0.05s. The measurement of uncertainty is the observer’s best estimate of how far off from the “true” value their measurement is likely to be, given the precision of their measuring equipment, the difficulty in making the measurement, and potentially other factors the observer deems important.

Pyth extends the concept of measurement uncertainty to asset prices with its **Confidence Interval**.

▼ **Further reading:**

- A classic article on uncertainty and price theory by Pyth
- Pyth and confidence aggregation
- Pyth Intern takes a stab

Publisher Oracle Network versus Reporter Oracle Network

Pyth is a first-of-its-kind data oracle network that introduces a new design to maximize access to information in a safe way on-chain that strips out middlemen costs. This design enables maximum scalability and growth of on-chain financial market data with exciting potential for off-chain as well.

Pyth is the largest ***publisher oracle network*** in the world with over 75+ original data sources publishing directly to the network. In this network, the nodes own and publish data directly on-chain.

Most legacy oracles are ***reporter oracle network*** where nodes carry data over the last mile from API endpoint to on-chain consumption. In this network, the nodes must purchase the data from first-party sources or other middlemen.

One of the advantages of **reporter networks** is that they are much more nimble. Given the adaptability of the independent nodes to fetch unstructured information and then normalize, reporter networks should be faster in expanding to a wider variety of data including generalized queries. This works really well with information that is in the public domain. But there are disadvantages in data quality, legality, and the scalability of this model.

▼ **A deeper comparative explanation:**

- While they are more generalized and adaptable, **reporter networks** are always constrained by the cost and output format of the data they purchase to report. These constraints are economic (the data costs something — middlemen aggregators charge additional fees, and the nodes themselves need to earn something) and non-economic, such that purchasers do not have control of update speeds, or have limited ability to request new features. In fact, many existing data supply chains will not permit their data to be reported to the blockchain because it is not possible to limit the distribution to only paying subscribers.
- The biggest challenge with a **publisher network** is overcoming the initial cold start of onboarding data providers to fill the shelves of the marketplace. Being the first person in a network carries quite a bit of risk both reputationally as well as technologically. If the network fails to pick up traction, early participants will have spent political capital and development resources that they could have used elsewhere.

- Once the supply side is sufficiently robust such that it can comprehensively cover necessary markets, the network needs sufficient demand to encourage further development and incentivize the suppliers. At some point, a critical mass is reached whereby the risk profile switches such that it is more reputationally or economically damaging to not be a part.

▼ **Further reading:**

- A classic article on Publisher versus Reporter networks by Pyth

Pythnet

Pythnet is powered by Solana technology: it runs the same validator software, but is a separate network that is specially configured to be a proof-of-authority chain. The network depends on a tightly controlled supply of the chain's native token, called PGAS, which is currently controlled by the Pyth Data Association. Operating a validator on the network requires a large stake of PGAS tokens.

The Pyth Data Association allows each data provider to operate one validator by delegating them the necessary stake. Each data provider is then given a sufficient quantity of PGAS tokens to publish prices to the network. The network is configured such that account creation is very expensive, preventing anyone without a substantial quantity of PGAS from deploying programs to the network. Once governance is live, it will take over management of the PGAS token from the Pyth Data Association.

Pythnet allows the network to scale with tremendous efficiency and a highly performant uptime.

Thanks to Pythnet, Pyth Network is able to expand to other blockchains with minimum friction. Users in other ecosystems can enjoy high-frequency and high-fidelity price updates on many types of assets.

▼ **Further reading:**

- Introducing Pythnet
- Pythnet docs explanation

Publisher Metrics

Pyth Publishers Metrics is a feature that provides insights that will empower developers, publishers, and delegators by providing the historical performance of the network's data sources. This powerful tool reflects our commitment to transparency and delivering timely, accurate, and valuable first-party data for everyone.

▼ **Further reading:**

- Read more about publisher metrics here
- You can see the metrics live if you click on the pubkey listed in a price feed page.

Pyth Data Association

The Pyth Data Association was created to support the Pyth network. A board of directors oversees the Association.

Frequently Asked Questions

▼ **How does Pyth work?**

- Pyth is a protocol that allows market participants to publish pricing information on-chain for others to use. The protocol is an interaction between three parties:
- 1. Data providers submit pricing information to Pyth's oracle program. Pyth has multiple data providers for every price feed to improve the accuracy and robustness of the system.
- 2. Pyth's on-chain oracle program on Pythnet combines providers' submitted data to produce a single aggregate price and confidence interval.
- 3. Applications read the price information produced by the oracle program. More specifically, Pyth allows users to “pull” prices onto the blockchain when needed. That price becomes publicly available for everyone on that chain.

▼ **What is an on-demand or “pull-based” oracle?**

- Most oracles today use a “push” model, where the oracle runs an off-chain process that continuously sends transactions to update an on-chain price. In contrast, Pyth does not operate an off-chain process that pushes prices on-chain. Instead, it delegates this work to Pyth users.

- Pyth price updates are created on Pythnet and streamed off-chain via the Wormhole Network, a cross-chain messaging protocol. These updates are signed such that the Pyth on-chain program can verify their authenticity.
- Updating the on-chain price is a *permissionless* operation: anyone can submit a valid Wormhole message to the Pyth contract to update the price. Typically, users of Pyth Network prices will submit a single transaction that simultaneously updates the price and uses it in a downstream application. Read more here.

▼ **What is Pythnet? Is it different than Solana?**

- Pythnet is an application blockchain network built on the Solana codebase that enables the Pyth network to aggregate first-party data at sub-second speeds and deliver pricing to other chains via the Wormhole cross-chain messaging protocol.
- The Pyth network originally chose to build on Solana because of the network's demonstrated ability to process thousands of transactions per second with fast finality. Furthermore, Solana's 400ms slot time allows Pyth's prices to update faster than most other Layer 1's. The Pythnet appchain was then launched in August 2022.
- Note that Pyth's oracle program runs simultaneously on both Solana mainnet and Pythnet. Each instance of the program is responsible for its own set of price feeds. Solana Price Feeds are available for use by Solana protocols.
- Pythnet is a separate blockchain from Solana. Accordingly, the liveness of Pythnet Price Feeds are unrelated to that of Solana Price Feeds. Read more here.

▼ **How does Pyth use Wormhole?**

- Wormhole allows Pyth to scale easily to multiple blockchains while keeping data sourcing and on-chain aggregation on the Pythnet appchain. As a generic message-passing protocol, Wormhole can bring Pyth price outputs as transactions to EVM chains, Aptos, Cosmos Hub, and more.

- Pyth price updates are created on Pythnet and streamed off-chain via the Wormhole Network, a cross-chain messaging protocol. These updates are signed such that the Pyth on-chain program can verify their authenticity. Updating the on-chain price is a permissionless operation: anyone can submit a valid Wormhole message to the Pyth contract to update the price. Typically, users of Pyth Network prices will submit a single transaction that simultaneously updates the price and uses it in a downstream application. Read more here.

▼ **What is the latency between Pythnet and Wormhole?**

- The latency between Pythnet to Wormhole is about 1-2 seconds excluding your native block time. You can use Pyth Benchmarks to facilitate delayed settlement or backfilling of orders by requesting any arbitrarily stale price.

▼ **What is the cost structure of using Pyth Price Feeds?**

- The Pyth protocol is designed to allow for the optional enablement of data fees in order to update the state of on-chain price feeds.
- The ongoing existence of and size of the fee will be determined by governance on a per-blockchain basis. Until governance is live, the fee will be 1 of the smallest denomination of the blockchain's native token (e.g. 1 wei on Ethereum).
- The fees collected by the protocol will go toward compensating data providers and possibly other uses as determined by governance. Read more here.

Supplementary Reading

I want to understand the context of oracles

- A brief primer on oracles by CMS Peary (Jul 2022)
- *Pyth Network Deep Dive - A fun mini-documentary.*
- A good podcast episode on oracles feat. Steve Kaminsky from Jump

- Forbes explores the landscape of big data
- Fortune peers into the future of crypto

I want to *feel* how much Pyth has grown since Day 1

- Animated explainer from the early days
- Our newest animated explainer
- See how far Pyth has come in July of 2022, Mike Cahill in Binance Blockchain Week (Paris)
- Mainnet anniversary milestones
- A lot happened in 2022 alone
- A lot happened in Q1 2023 alone

I want to learn about Pyth as a self-sufficient ecosystem

- Mike Cahill's lecture: you only need to watch the first 13 minutes. The content after is still good, but from 2021…
- The Pyth Whitepaper. Focus on what you can understand.
- Jayant breaks down the whitepaper into a very easy to understand lecture. This thread covers the Pyth grading model too, and brings in some more recent features, like cross-chain.
- Pyth Intern explainer thread to strengthen your sense of the Pyth "system".
- A very short thread by the Intern about what makes Pyth special. It's a bit outdated, but it will give you a sense of what to focus on.
- A past Ambassador's thread on what Pyth is setting out to do.
- If you like watching Jayant speak, here's one of his latest videos

I want to read EVERYTHING about Pyth

Whoa, slow down. OK fine, you can start with this series and read it in order.

- Some articles will be very quick, since you’re smart and you already learned this information.
- Some articles will be new and exotic – don’t worry, just take in what you understand!