

ENHANCING PROMETHEUS:

Mentor: Dr. Hadar Binsky
Project Number 15006422



PROMETRIC

By Rotem Zahavi and Romi Polaczek

What is Prometheus?



Prometheus is an open-source tool used to monitor systems and send alerts.

It features:

- **Tracks Data Over Time:** It records and stores data, like system performance, over time.
- **PromQL:** You can use its special language (PromQL) to find and analyze this data.
- **Built-In Alerting:** Supports alerts and integrates with notification systems.

Prometheus is popular for monitoring systems and applications due to its performance and scalability.



The Problem

Prometheus currently keeps all metrics for the same amount of time, no matter how important or how often they're used. This causes a few problems:

- **Wasted resources:** Unimportant or rarely used data takes up space for no good reason.
- **Higher costs:** Storing data that isn't very useful for a long time costs more money.
- **Limited control:** Users can't choose how long to keep different types of data based on how important they are.

prometheus / prometheus

Q

Type / to search

<> Code

Issues 537

Pull requests 192

Discussions

Actions

Projects 1

Wiki

Security 2

Insights

Retention time configurable per series (metric, rule, ...). #1381

New issue

Open

taviLaies opened on Feb 10, 2016

...

Hello,

I'm evaluating prometheus as our telemetry platform and I'm looking to see if there's a way to set up graphite-like retention. Let's assume I have a retention period of 15d in prometheus and I define aggregation rules that collapse the samples to 1h aggregates. Is there a way to keep this new metric around for more than 15 days?
If this is not possible, could you provide some insight on how you approach historical data in your systems?

Thank you

👍 102

matthiasr on Feb 10, 2016

Contributor ...

This is not something Prometheus supports *directly* at the moment and for the foreseeable future. The focus right now is on operational monitoring, i.e. the "here and now".

You can get something like this by using a tiered system. The first-level Prometheus would scrape all the targets and compute the rules. A second-level Prometheus would [federate](#) from it, only fetching the result of these rules.

Assignees

No one assigned

Labels

kind/enhancement

priority/P3

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

prometheus / prometheus

Q

Type ↵ to search

+

▼

<> Code

Issues 561

Pull requests 197

Discussions

Actions

Projects 1

Wiki

Security 2

Insights

Add Feature to Set Retention Time per Metric #15350

New issue

Open

KONE-XAD opened this issue last month · 0 comments

KONE-XAD commented last month

...

Proposal

Currently, Prometheus applies the same retention time configuration across all metrics, which results in metrics of varying importance and usage being stored for the same duration. This unified approach can lead to storage inefficiencies, as less critical or infrequently used metrics may occupy significant storage space, despite not requiring long-term retention.

We would like to request the following feature for Prometheus:

The ability to set different retention times for specific metrics or groups of metrics.
Critical or high-priority metrics could be configured with longer retention times.
Less important metrics could have shorter retention periods, helping to save storage resources.
This more granular approach would allow users to more effectively manage storage usage, reduce costs, and improve overall system efficiency.

Background:
Currently, all metrics in Prometheus follow the same retention policy, which may cause storage inefficiencies for low-frequency or less important metrics. By allowing different retention times for different categories of metrics, we could optimize storage usage.

Desired Feature:

The ability to configure different retention times for specific sets of metrics, namespaces, or labels via the configuration file or Prometheus API.
Provide flexibility to allow users to adjust the retention period for each metric according to their needs.
This should clearly convey the request for a feature that would allow different retention policies for different metrics.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

Notifications

Customize

Subscribe

You're not receiving notifications from this thread.

1 participant

...

The Solution - Prometric

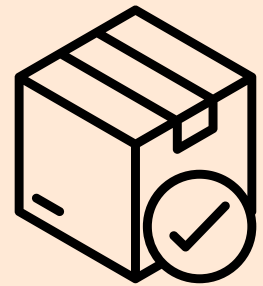
Prometric introduces an external API layer that enables per-metric retention policies. This solution allows system administrators and developers to define different retention times for different metrics, without modifying Prometheus' core.

The API integrates with Prometheus' query interface, manages policies in a database, and automatically deletes outdated metrics. A web-based interface further provides visibility, control, and visualization of retention policies in action.

Demo

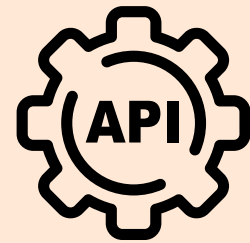
<https://youtu.be/XSM9cNxP8YE>

Advantages



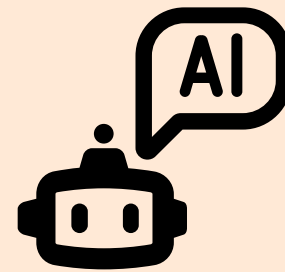
Optimized Storage Usage

Prevents wasteful storage of unnecessary data while keeping critical metrics longer.



Seamless Prometheus Integration

Works as an external API layer without modifying Prometheus core.



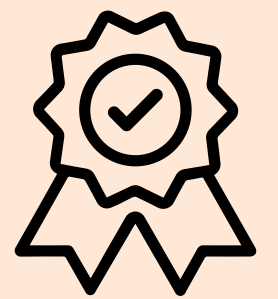
AI-Powered Query Assistance

Natural language queries automatically translated into PromQL.



Full-Featured Prometheus Interface

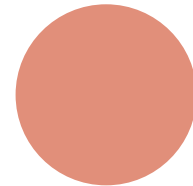
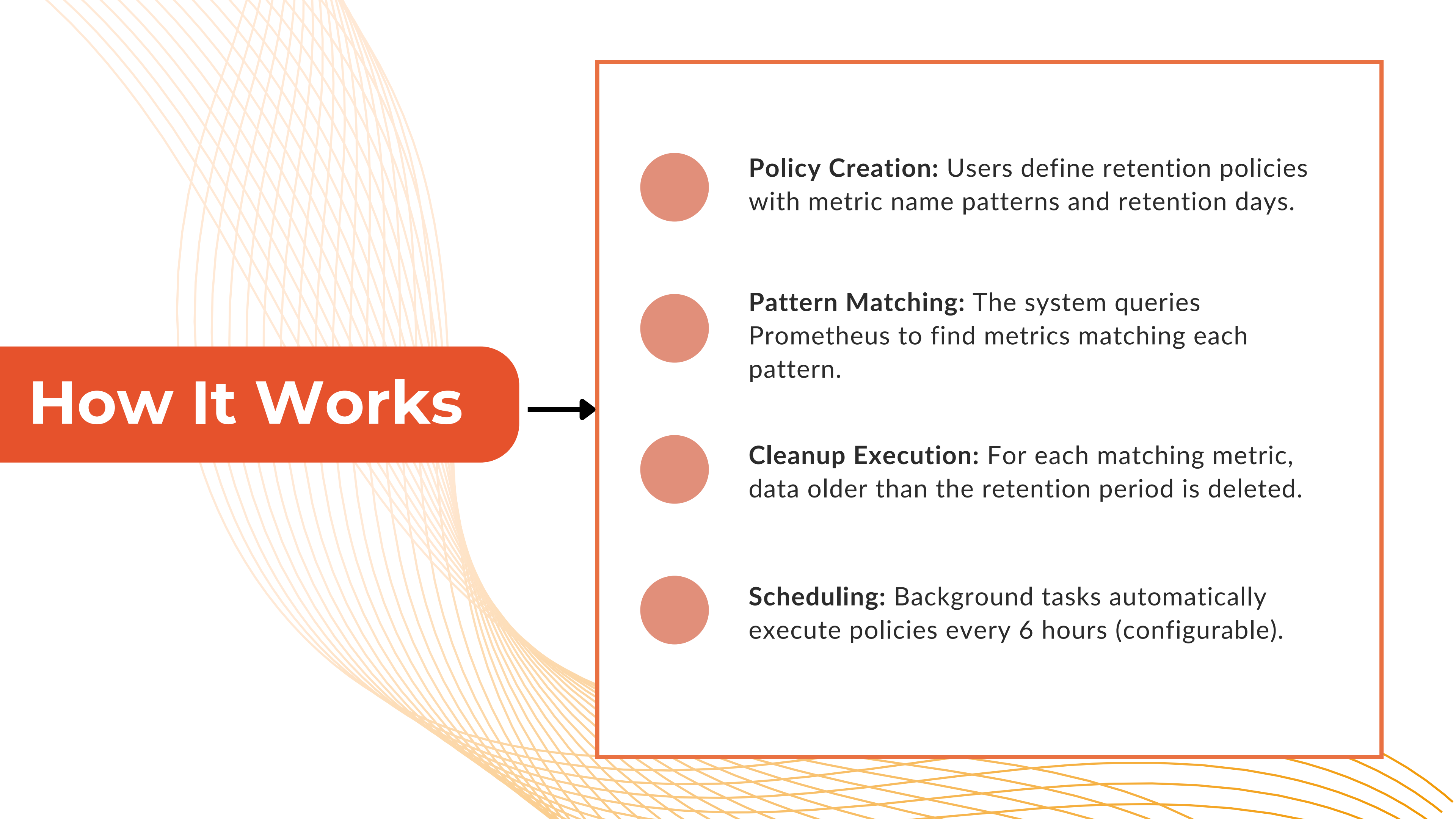
Web interface that provides all Prometheus features plus additional enhancements.



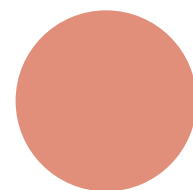
Reliability & Transparency

Automated background process ensures consistent enforcement of policies with full logging.

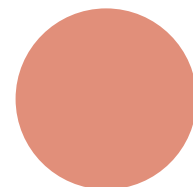
How It Works



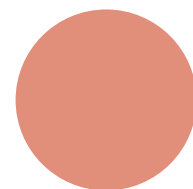
Policy Creation: Users define retention policies with metric name patterns and retention days.



Pattern Matching: The system queries Prometheus to find metrics matching each pattern.



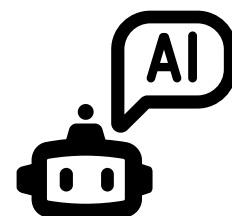
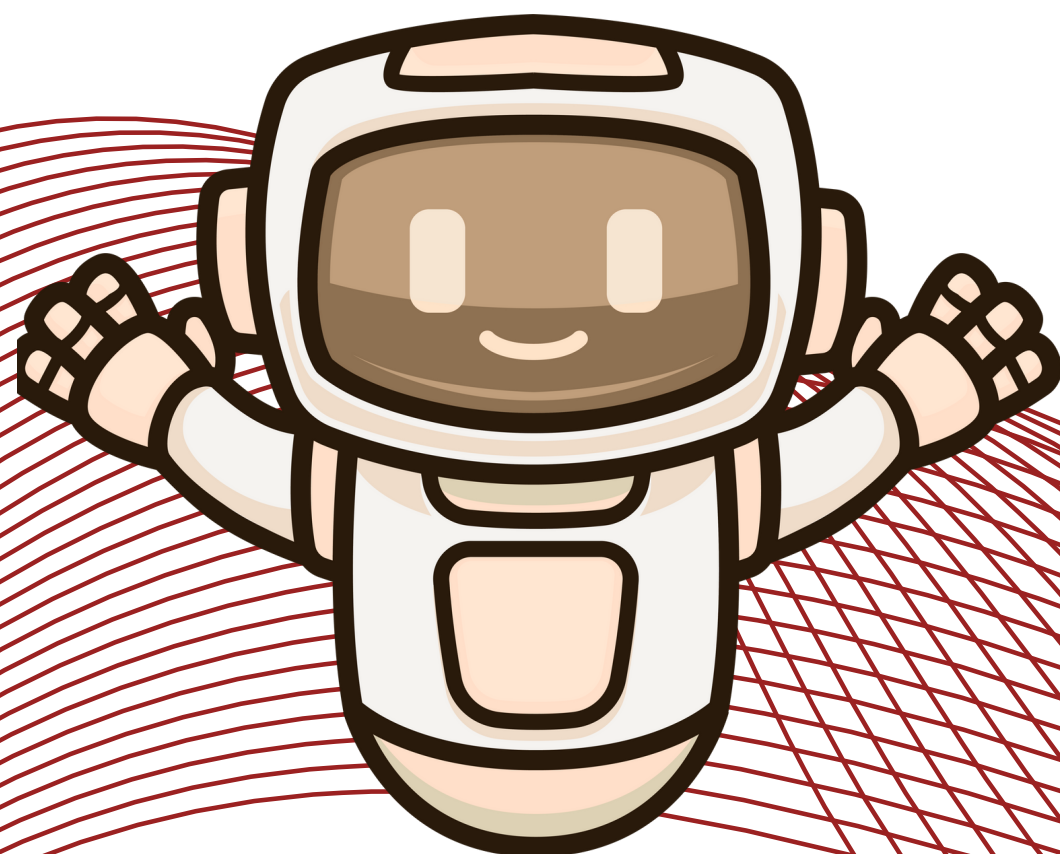
Cleanup Execution: For each matching metric, data older than the retention period is deleted.



Scheduling: Background tasks automatically execute policies every 6 hours (configurable).



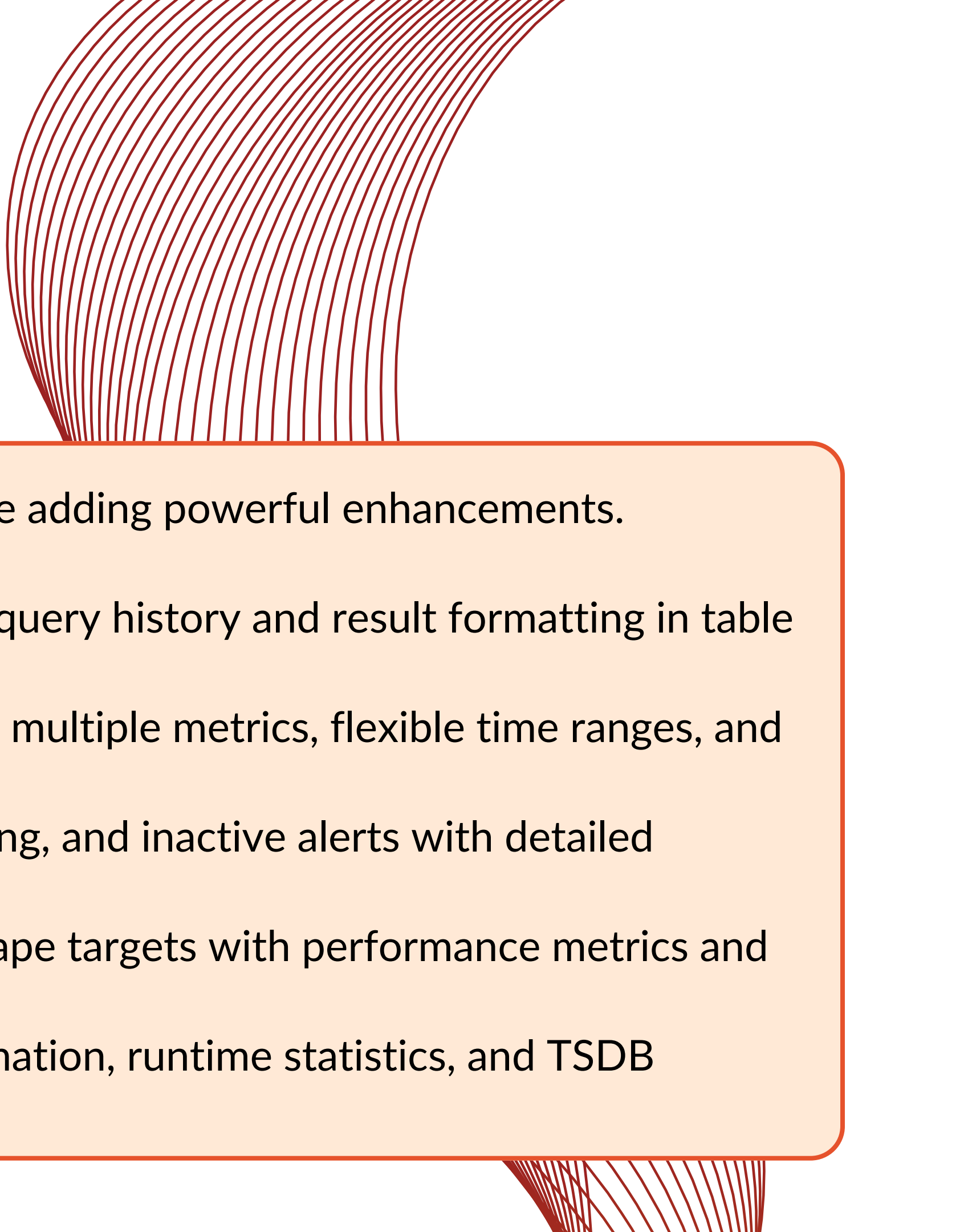
AI-Powered Query Assistant



Prometric integrates OpenAI GPT-3.5 Turbo to revolutionize how users interact with Prometheus data.

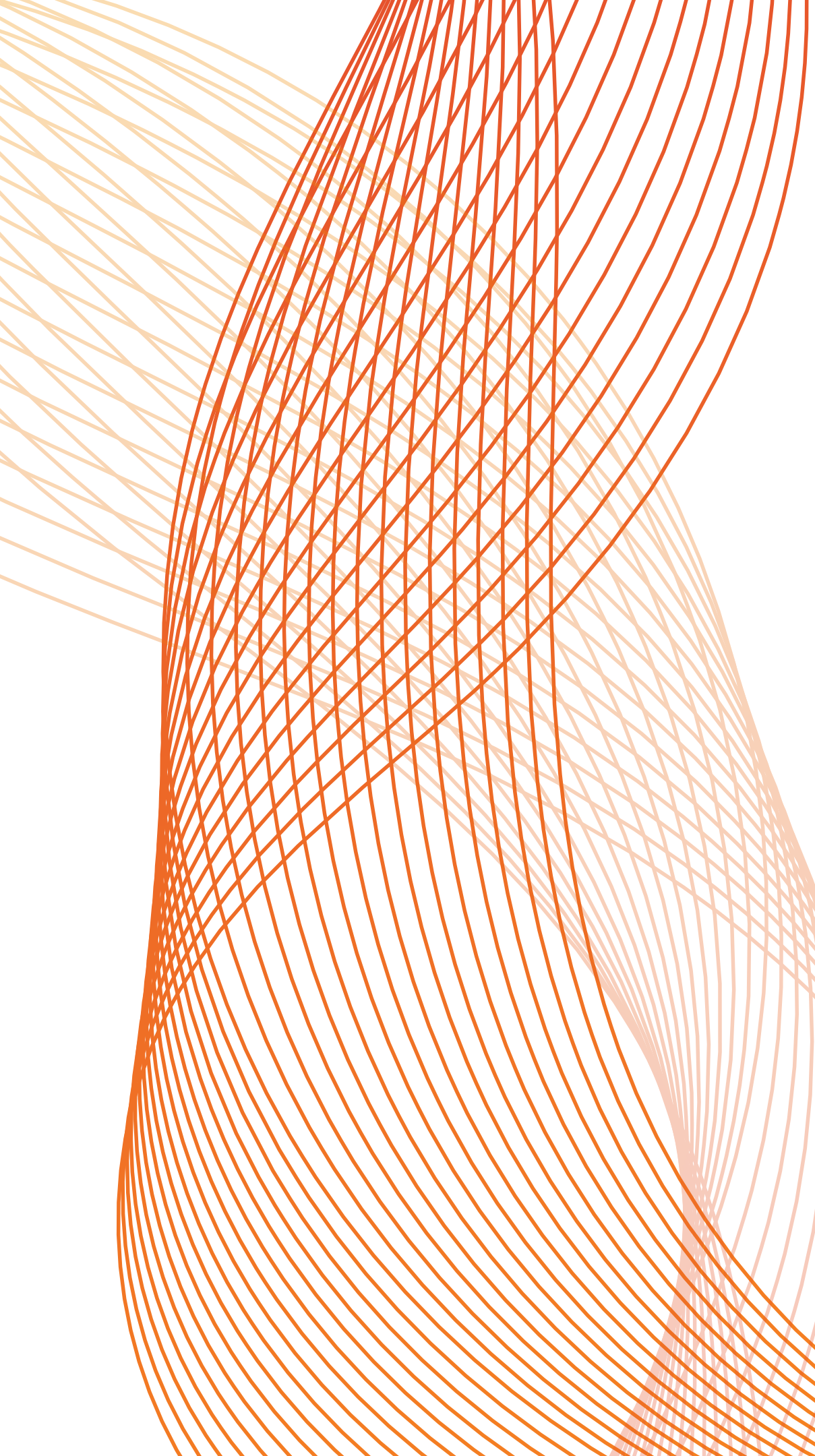
- **Natural Language Translation** – Converts plain English questions into valid PromQL queries automatically.
- **Real-time Chat Interface** – Persistent conversation with query history and explanations for each generated query.
- **Zero Learning Curve** – Non-experts can query Prometheus instantly without memorizing PromQL syntax.

Complete Prometheus Integration



Prometric implements all core Prometheus features while adding powerful enhancements.

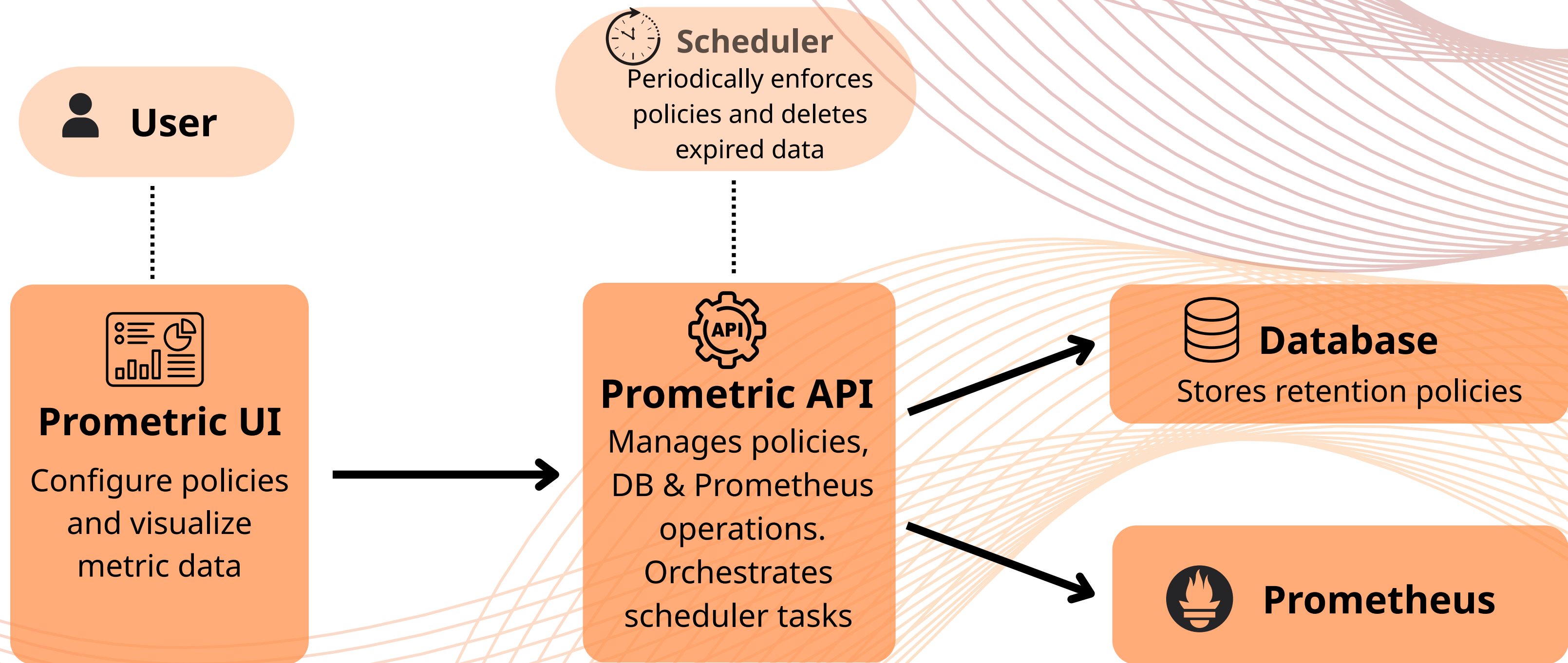
- **Query Console** – Real-time PromQL execution with query history and result formatting in table and JSON views.
- **Interactive Graphing** – Time series visualization with multiple metrics, flexible time ranges, and CSV export capabilities.
- **Alert Management** – Live monitoring of firing, pending, and inactive alerts with detailed metadata display.
- **Target Monitoring** – Health status tracking of all scrape targets with performance metrics and error reporting.
- **System Status** – Comprehensive view of build information, runtime statistics, and TSDB performance data.



SYSTEM Architecture

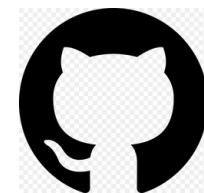
Prometric follows a modern microservices architecture with clear separation of concerns and scalable design patterns.

- **Frontend Layer** - Next.js application with server-side rendering, automatic code splitting, and optimized performance.
- **Backend API** - FastAPI-based REST service handling Prometheus integration, retention policies, and AI query processing.
- **External Integrations** - Seamless communication with Prometheus server and OpenAI API for enhanced functionality.
- **Data Persistence** - SQLAlchemy-managed database with automated policy execution and comprehensive logging.



Thank you for listening!

We would love to hear your
thoughts and answer any question



Prometric