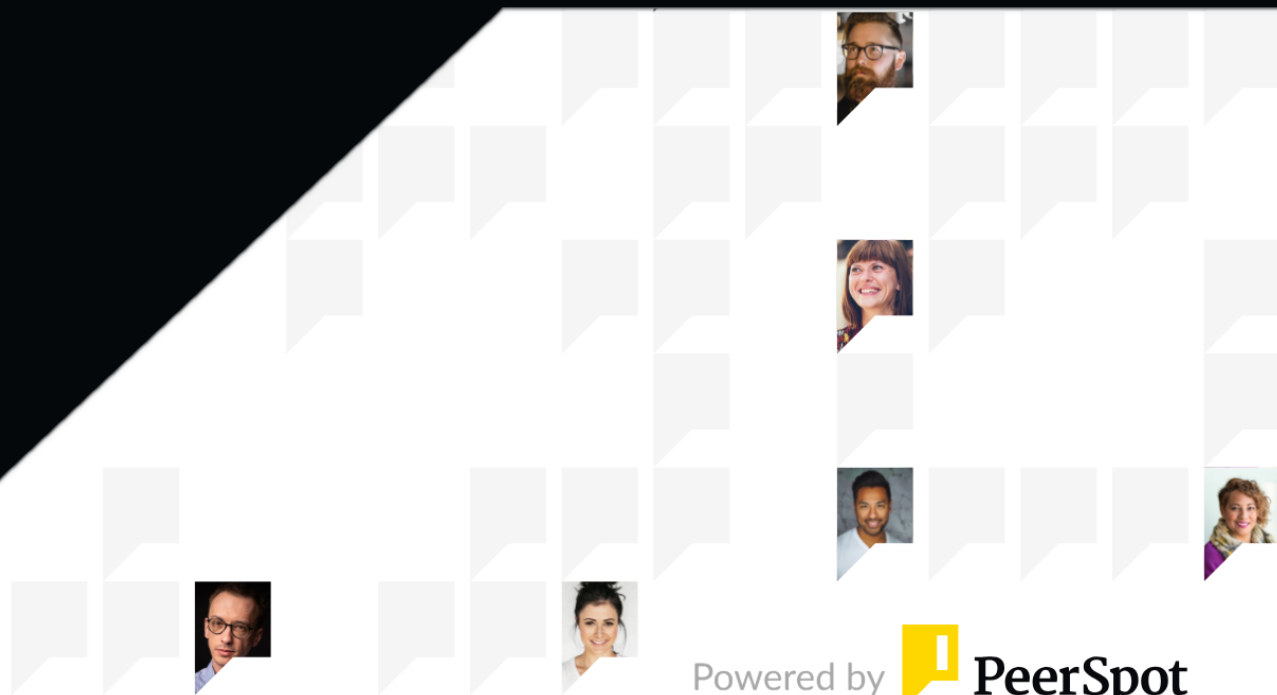




Apollo GraphOS

Reviews, tips, and advice from real users



Powered by  **PeerSpot**

Contents

Product Recap..... 3 - 4

Valuable Features..... 5 - 12

ROI..... 13

Use Case..... 14 - 17

Setup..... 18

Other Advice..... 19 - 22

Trends..... 23 - 24

About PeerSpot..... 25 - 26

Product Recap



Apollo GraphOS

Apollo GraphOS Recap

Apollo GraphOS is a comprehensive platform designed for managing and enhancing GraphQL development. It streamlines the process of creating, testing, and maintaining GraphQL APIs effectively, offering a robust environment for developers.

Apollo GraphOS offers developers a powerful toolkit for streamlining GraphQL API lifecycle management. Its robust structure supports efficient schema design, observability, and collaboration, making it a valuable asset for development teams. These features allow for enhanced control and oversight, ensuring a seamless workflow. However, there is still room for improvement in integration capabilities with other tools and platforms, which users have noted in reviews.

What are Apollo GraphOS's key features?

- **Schema Federation:** Allows seamless integration of multiple services under a unified graph.
- **GraphQL Insights:** Provides in-depth analytics to optimize API performance and usage.
- **Version Control:** Ensures precise and organized tracking of schema changes over time.
- **Collaboration Tools:** Facilitates teamwork through shared access and communication features.
- **Security Management:** Offers advanced protection mechanisms for your API.

What benefits should users expect?

- **Improved Efficiency:** Streamlines development processes, saving time and resources.
- **Enhanced Collaboration:** Promotes teamwork and efficient communication with shared resources.
- **Scalability:** Supports growth with robust infrastructure adaptable to increasing demands.
- **Data-Driven Decisions:** Enables informed choices through comprehensive analytics tools.

In industries like e-commerce and fintech, Apollo GraphOS enables teams to rapidly implement scalable GraphQL solutions, facilitating real-time data interactions and improving customer experiences. Healthcare sectors utilize it for secure data management, while media industries benefit from dynamic content delivery capabilities.

Valuable Features

Excerpts from real customer reviews on PeerSpot:



“Apollo GraphOS has had a positive impact on our organization by making our API development process more structured, predictable, and collaborative.”



Mitali Bhosle

Business Development Manager at Digitaltrack



“Apollo GraphOS has had a positive impact by improving the way we manage and evolve our GraphQL APIs.”



Swapnil Dhabewar

Software at Tester



“Apollo GraphOS has significantly enhanced both the developer experience and product delivery timelines by offering a centralized approach to managing GraphQL schemas and services, helping reduce overhead and improve collaboration between frontend and backend teams.”



Nazeh Abel

Artificial Intelligence Consultant at Medallion Technologies



“The most valuable features of Apollo GraphOS OS for me are its security and authentication processes.”



Sagor Rana

Front End Engineer at Upwork



“Apollo GraphOS has positively impacted our organization by reducing unwanted code and helping us combine multiple services.”



Amar-Kumar

Technical Lead at a tech services company with 501-1,000 employees

What users had to say about valuable features:

The most valuable features of Apollo GraphQL are its security and authentication process. These roles are crucial, especially when working with APIs. Its ability to filter and streamline API responses for performance efficiency is also highly appreciated.

Sagor Rana

Front End Engineer at Upwork

[Read full review](#) 

“The best features I find most valuable are schema registry, schema checks, API observability, and Apollo Federation.

“It provides visibility into API usage, performance, and query patterns. This helps us to identify bottlenecks and understand how clients are using GraphQL APIs..”

Swapnil Dhabewar

Software at Tester

[Read full review](#) 

“One of the standout features is the intuitive support for mutations and the flexibility they provide. They make state updates seamless and reduce the boilerplate often associated with RESTful services. I also value the Schema Registry and Federation capabilities, which enable us to build scalable, modular APIs while avoiding tight coupling. Community support and documentation have also been excellent, helping us get up to speed quickly and troubleshoot when needed. Overall, the ability to efficiently fetch only the data needed with minimal friction makes development much faster and cleaner..”

Nazeh Abel

Artificial Intelligence Consultant at Medallion Technologies

[Read full review](#) 

“The best feature Apollo GraphOS offers is unifying multiple APIs and endpoints into a single point.

“The unified endpoint feature helps my team on a daily basis by combining different services together. For example, when fetching a menu and pricing information, the backend would previously require multiple REST APIs. Now with Apollo GraphOS, we can easily combine the menu service, pricing, and inventory together. Our frontend mobile team, kiosk team, and web ordering team can access this data through a single GraphQL endpoint.

“Apollo GraphOS has positively impacted our organization by reducing unwanted code and helping us combine multiple things. It unifies the API layer, which is straightforward to understand and saves our team time.

“It has reduced both complexity and time. Previously, we had to manage all services in parallel and individually. After unifying to a single API layer, the process became much easier and more time-saving..”

Amar-Kumar

Technical Lead at a tech services company with 501-1,000 employees

[Read full review](#) 

“From my experience, the most valuable features of Apollo GraphOS are its schema management and governance capabilities. The schema registry combined with automated schema checks has been extremely useful because it helps identify breaking changes before they reach production, which has made our release process much more reliable. I also appreciate the federation support as it allows different teams to own and maintain their individual GraphQL services while still exposing a single unified API to client applications. Another feature we have found particularly helpful is the usage insights, which provide visibility into how fields and operations are being consumed by applications. This information makes it easier to retire unused fields, assess the impact of proposed changes, and make informed decisions about API evolution. The integrations with CI/CD pipelines have also streamlined our development workflow by automating schema validations as part of every deployment. On the operational side, the platform offers good observability into the GraphQL layer, making it easier to troubleshoot issues and understand API behavior. While these features work together effectively, there is a learning curve around federation and schema governance. Organizations tend to get the most value once they have established clear ownerships and deployment standards across their teams.

“The feature that has had the biggest impact for our team is the automated schema validation and schema checks. Before we introduced Apollo GraphOS into our deployment workflow, there was always some uncertainty when making changes to shared GraphQL schemas because it was not straightforward to know whether a seemingly small modification would affect existing applications. We relied heavily on manual reviews and testing, which worked up to a point but became difficult as more teams started contributing to the API. With automated schema checks integrated into our CI/CD pipeline, every schema change is validated before it can be merged or deployed, and any potential breaking changes are flagged early in the development cycle. This has significantly reduced deployment risk and eliminated many of the issues that previously surfaced only after release. It has also improved collaboration between front-end and back-end teams because developers can make changes with greater confidence knowing they have immediate feedback on compatibility. While features such as federation and usage insights are also valuable, the automated schema validations have had

the most noticeable day-to-day impact because it has made our release process more predictable, reduced production incidents related to API changes, and allowed the team to move faster without compromising stability.

“The biggest value we have seen from Apollo GraphOS is not just the technology itself, but the way it has improved collaboration across teams. As our environment grew, it became increasingly difficult to coordinate API changes between front-end and back-end developers, especially when multiple teams were working on different services at the same time. Having a centralized platform for schema management, validation, and usage visibility gave everyone a shared view of the API and made ownership much clearer. It also made onboarding new developers easier because they could explore the schema and understand how services were connected without relying heavily on documentation or tribal knowledge. From an operational standpoint, we have been able to standardize our GraphQL development process and reduce the number of production issues caused by uncoordinated schema changes. To get the most value from it, organizations need well-defined governance, clear ownerships of subgraphs, and disciplined schema design. Once those practices are in place, Apollo GraphOS becomes an effective part of the development workflow and scales effectively as the number of services and teams increase.

“We have been using Apollo GraphOS for a little over three years. It initially started as a pilot in a small group of endpoints within our IT and security teams so we could evaluate how it fits into our existing environment and understand the operational impact before rolling it out more broadly. The pilot lasted a few months, during which we tested policy creations, application control, compatibility with our business-critical software, and the day-to-day management experience. After we were satisfied with the results and addressed a few application exceptions, we expanded the deployment across the organization in phases. Over time, the product has become part of our standard endpoint security strategy, and we have gained enough operational experience to understand both its strengths and its limitations. The longer we have used it, the more refined our policies have become, which has helped reduce administrative efforts while maintaining a consistent security posture. The real strength of Apollo GraphOS comes from how its features work together rather than any single

capability in isolation. Schema management, federation, usage insights, and CI/CD integrations complement each other to create a more controlled and efficient API development process. As our environment grew and more teams started contributing to the GraphQL platform, having centralized visibility into schema changes and API usage made coordination much easier and reduced the amount of manual communication required between teams. We also found that the platform encourages better development practices because developers think more carefully about schema evolution and backward compatibility before making changes. Organizations should be prepared to invest some time in defining governance, naming conventions, and ownership of subgraphs. Without those processes, it is harder to get the full benefit from the platform. Once those practices are established, these features come together effectively and make managing a large GraphQL environment much more straightforward..”

Mitali Bhosle

Business Development Manager at Digitaltrack

[Read full review](#) 

ROI

Real user quotes about their ROI:

“I think we have seen a return on investment. We have improved our development efficiency. I would estimate we saved 10% to 15%, or perhaps 20% of the time previously spent on manually checking schema, schema validation, or API change reviews..”

Swapnil Dhabewar

Software at Tester

[Read full review](#) 

Use Case

I primarily use Apollo GraphQL to work with APIs in conjunction with Node.js. When I make API calls, I receive a lot of information, but I only need specific details like names, emails, and photos. Using Apollo GraphQL, I can filter out unnecessary information, which improves performance. I have deployed it on my GitHub account.

Sagor Rana

Front End Engineer at Upwork

[Read full review](#) 

“I’ve been using Apollo GraphOS across several mobile and web projects. On the mobile side, I’ve integrated it with React Native and Flutter. For web applications, the usage has been primarily within React-based projects. My development environment typically includes CI/CD pipelines, modular architecture, and cloud deployments, and Apollo GraphOS fits smoothly into this ecosystem. It helps streamline the way we manage our GraphQL layer and aligns well with modern frontend workflows which is a very nice to have experience.”

Nazeh Abel

Artificial Intelligence Consultant at Medallion Technologies

[Read full review](#) 

“Apollo GraphOS's main use case is combining multiple backend services into a unified single GraphQL query. In our kiosk implementation, we display a launch screen with a dynamic banner, store-specific banners, and a default banner. Apollo GraphOS allows the backend to retrieve data from multiple services and provide it through a single GraphQL query.

“For the backend, this represents a major implementation advantage. Using REST APIs would require calling multiple APIs to fetch data. However, Apollo GraphOS provides an endpoint that allows us to hit a single endpoint to retrieve all the data..”

Amar-Kumar

[Read full review](#) 

Technical Lead at a tech services company with 501-1,000 employees

“My main use case for Apollo GraphOS is managing and monitoring GraphQL APIs across the development life cycle. I use it to manage our GraphQL schema, validate schema changes before deployment, and monitor API usage and performance. It also helps us collaborate across the teams by providing a centralized view of a GraphQL schema, and it ensures that changes don't introduce breaking issues for existing clients.

“For example, when we introduce a new field or modify an existing schema, we use the schema validation and checks to verify that the change won't break existing front-end applications. That gives us more confidence when deploying the new update and reduces the risk of API compatibility issues.

“One additional benefit is that Apollo GraphOS provides a centralized platform for our GraphQL API, instead of managing a schema and API changes separately in different services..”

Swapnil Dhabewar

Software at Tester

[Read full review](#) 

“Our main use case for Apollo GraphOS is to provide a unified GraphQL API across multiple back-end services while allowing different development teams to work independently. We have several applications that consume data from different microservices, and before using Apollo GraphOS, each front-end team had to integrate with multiple APIs, which increased development effort and made it harder to maintain consistency. Apollo GraphOS gave us a single GraphQL endpoint backed by a federated architecture. Each team could own its own schema without affecting the overall API experience. We also use the platform schema registry and schema validation feature as part of our CI/CD pipeline to catch breaking changes before they reach production. The usage insights and scale schema checks have been particularly valuable because they show us which fields are actively being consumed, allowing us to deprecate or modify APIs with much more confidence. From a day-to-day operations perspective, it has improved collaboration between front-end and back-end teams, reduced duplicate API development, and made it easier to evolve our services as the environment has grown. That said, designing and maintaining a federated schema require good governance and clear ownership across teams. There is some initial learning and coordination involved, especially for organizations adopting GraphQL Federation for the first time..”

Mitali Bhosle

Business Development Manager at Digitaltrack

[Read full review](#) 

Setup

The setup process involves configuring and preparing the product or service for use, which may include tasks such as installation, account creation, initial configuration, and troubleshooting any issues that may arise. Below you can find real user quotes about the setup process.

“From a developer's perspective, the setup and onboarding into the development environment was straightforward. I felt it was good value for the capabilities that we used..”

Swapnil Dhabewar

Software at Tester

[Read full review](#) 

Other Advice

From one to ten, I would rate Apollo GraphQL six or six point five. Despite its benefits, it can sometimes break down badly, and the need for continuous improvements might involve handling heavy loads, especially for smaller to medium apps. I rate the overall solution at 6.5.

Sagor Rana

Front End Engineer at Upwork

[Read full review](#) 

“For my use case, I would rate Apollo GraphOS an eight out of ten because it is a very large platform and we have not used everything from it. I would recommend Apollo GraphOS if you have multiple services and want to combine them into a single unified platform, which will make your ecosystem faster compared to REST APIs. My overall review rating is eight out of ten..”

Amar-Kumar

Technical Lead at a tech services company with 501-1,000 employees

[Read full review](#) 

“A good example was when we needed to update a part of our GraphQL schema to support a new feature. Before deploying the change, we used the schema registry and schema checks to verify whether existing client applications would be affected.

“Regarding AI from a governance and security perspective, I think Apollo GraphOS does a good job. Features such as the schema-related checks reduce the risk of breaking changes.

“Based on my experience, the platform's core features have been reliable. I don't have sufficient experience with its AI capabilities to rate the accuracy of AI-generated output.

“I would recommend that organizations have a clear understanding of their GraphQL architecture and use cases before adopting Apollo GraphOS. They can take advantage of features such as schema management, schema checks, and observability from the beginning, and at the same time, invest time in learning the platform's advanced features. I rated this product six out of ten overall..”

Swapnil Dhabewar
Software at Tester

[Read full review](#) 

“My advice would be to start with a clear GraphQL strategy rather than focusing on the tool itself. Apollo GraphOS delivers the most value when you have multiple services, multiple development teams, or a growing API ecosystem that needs consistent governance. I would recommend beginning with a small pilot, establishing clear ownership for each subgraph, and defining schema design standards and review processes before scaling across the organization. It is also worth integrating schema validation into your CI/CD pipeline early, as that is one of the features that provides immediate operational benefits by preventing breaking changes from reaching production. Both front-end and back-end teams should be involved from the beginning because Apollo GraphOS works best when API design is treated as a shared responsibility rather than something owned by a single team. Invest some time in training developers on GraphQL Federation and schema governance. The platform itself is straightforward to use once it is set up, but having consistent development practices is what really determines long-term success. Organizations that take the time to establish those foundations will get much more value from the platform than those that treat it as just another infrastructure tool.

“As GraphQL environments become larger and involve more teams, having more

built-in guidance for schema design, best practices, and migration planning would be helpful, especially for organizations adopting federation for the first time. I would also like to see richer operational dashboards that combine schema health, performance metrics, and governance insights into a single view, making it easier for platform teams to monitor the overall GraphQL ecosystem. The platform already addresses our primary requirements effectively. The areas I have mentioned are more about enhancing usability and reducing operational overhead rather than fixing major shortcomings.

“While we did not track every important improvement with formal KPIs, we did notice several measurable operational benefits after adopting Apollo GraphOS. The most noticeable was a reduction in production issues caused by GraphQL schema changes because the automated schema validation spots compatibility problems much earlier in the development cycle. We also saw the release process become more predictable with fewer last-minute rollbacks or hotfixes related to API changes. From a developer productivity standpoint, teams spent less time coordinating schema updates and manually reviewing potential downstream impacts, allowing them to focus more on feature development. Onboarding new developers also became easier because they had a centralized view of the GraphQL schema and clear ownership of different services. Overall, we estimate that API-related deployment issues dropped by around thirty percent, and the time spent validating schema changes before release was reduced significantly through automation. The day-to-day operational improvements are noticeable, especially as the number of services and development teams continues to grow.

“From our perspective, AI capabilities have not been a primary reason for using the platform. Our focus has been on its GraphQL Federation, schema governance, API management, and CI/CD integration rather than AI functionality. Based on our experience, the platform has been reliable in those four areas with consistent schema validation and governance features that have helped reduce deployment issues. If Apollo expands AI capabilities in the future, we would want to evaluate them over time in a production environment before commenting on their accuracy or reliability. We prefer to assess AI features based on real operational experience rather than initial impressions.

“I give Apollo GraphOS a rating of nine out of ten because in day-to-day use, it has

been a reliable and well-designed platform that solves the core challenges of managing GraphQL APIs at scale. The schema registry, federation capabilities, automated schema validation, and CI/CD integrations have all delivered tangible value and made the release process much more predictable. The reason I held back from giving it a perfect score is that some enterprise scenarios still require additional effort. For example, troubleshooting requests that span multiple subgraphs can be more involved than we would prefer, and new teams need time to become comfortable with federation concepts and governance practices. I would also welcome more built-in administrative controls, richer reporting, and broader out-of-the-box integrations with enterprise monitoring and development tools to reduce the amount of customization needed. These are more about making an already strong platform easier to adopt and operate as environments become larger and more complex rather than major weaknesses or reasons to avoid the product..”

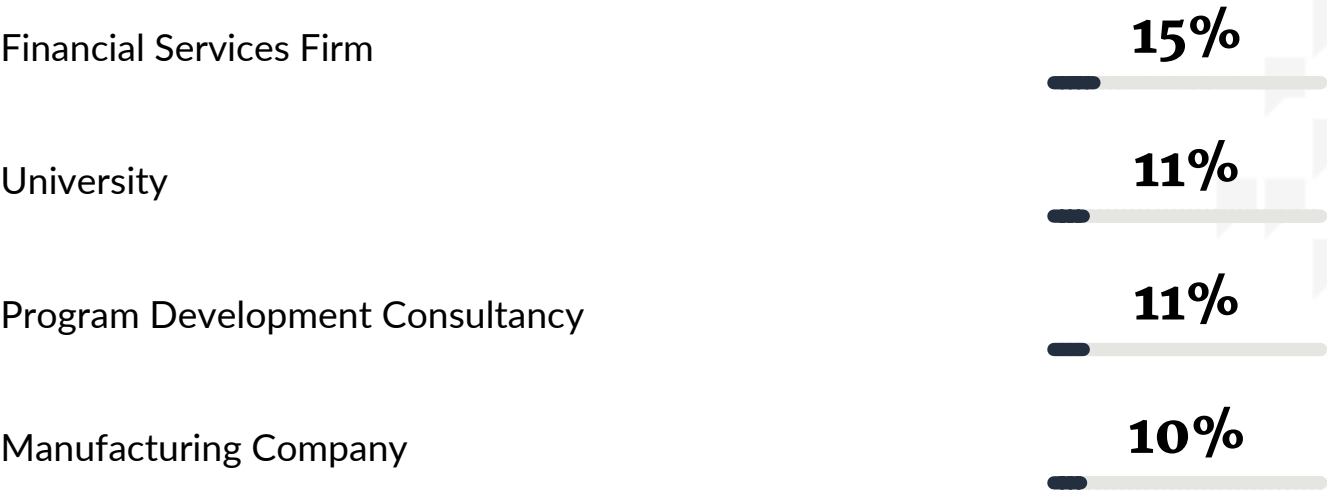
Mitali Bhosle

Business Development Manager at Digitaltrack

[Read full review](#) 

Top Industries

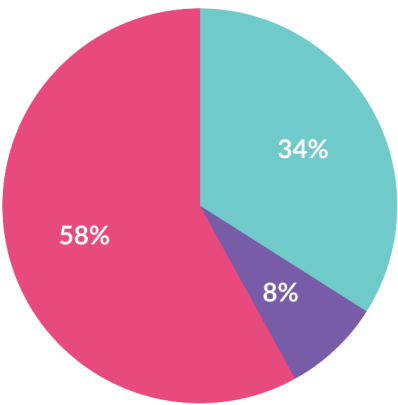
by visitors reading reviews



Company Size

by reviewers

by visitors reading reviews



 Large Enterprise  Midsized Enterprise  Small Business

About this buyer's guide

Thanks for downloading this PeerSpot report.

The summaries, overviews and recaps in this report are all based on real user feedback and reviews collected by PeerSpot's team. Every reviewer on PeerSpot has been authenticated with our triple authentication process. This is done to ensure that every review provided is an unbiased review from a real user.

Get a custom version of this report... Personalized for you!

Please note that this is a generic report based on reviews and opinions from the collective PeerSpot community. We offer a [customized report](#) of solutions recommended for you based on:

- Your industry
- Company size
- Which solutions you're already considering

The customized report will include recommendations for you based on what other people like you are using and researching.

Answer a few questions in our short wizard to get your customized report.

[Get your personalized report here](#)

About PeerSpot

PeerSpot is the leading review site for cloud, AI, and business software. We created PeerSpot to provide a trusted platform to share information about software, applications, and services. Since 2012, over 22 million people have used PeerSpot to choose the right software for their business.

PeerSpot helps tech professionals by providing:

- A list of products recommended by real users
- In-depth reviews, including pros and cons
- Specific information to help you choose the best vendor for your needs

Use PeerSpot to:

- Read and post reviews of products
- Access over 30,000 buyer's guides and comparison reports
- Request or share information about functionality, quality, and pricing

Join PeerSpot to connect with peers to help you:

- Get immediate answers to questions
- Validate vendor claims
- Exchange tips for getting the best deals with vendor

Visit PeerSpot: www.peerspot.com

PeerSpot

244 5th Avenue, Suite R-230 • New York, NY 10001

reports@peerspot.com

+1 646.328.1944