IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153120

DATAVISION PRO: INTERACTIVE BUSINESS INTELLIGENCE DASHBOARD

Dr. Sajja Suneel[1], Manipatruni Trived[2], Nathi Dhana Sree[3]

Department of Computer Science and Engineering (Data Science),

Institute of Aeronautical Engineering, Dundigal, Hyderabad[1-3]

Abstract: DataVision Pro, a web platform that integrates completely, was developed to eradicate the workflow gap between API testing and Business Intelligence (BI) visualization. The conventional approaches necessitate the usage of separate tools for confirming the responses of REST APIs and creating the reports thus causing inefficiency and delays in the team. DVP has come up with the solution to the problem by merging the main API testing and the data visualization characteristics plus the very good multi-user real-time collaboration and built-in BI tools all together. Some of the key features are real-time collaborative dashboard editing, automatic dashboard versioning, and secure sharing. The built-in BI features are creating KPIs with thresholds, blending data across charts, advanced drilldowns, and support for user-defined fields and time-series forecasting. The system is built on a serverless backend platform that guarantees real-time data synchronization for concurrent, intricate BI computations. Our research evaluates the efficiency of the platform, particularly its built-in BI feature integration effects on API testing workflows and visualization insights generation speed. We outline the system architecture and methods that provide an easy foundation for determining the performance and scalability of serverless, real-time BI collaboration. This indicates a substantial growth in developer productivity and the implementation of data supported decision making.

Index Terms: API testing, Business Intelligence, Real-time Collaboration, Data Visualization, Vertically scaled serverless computation, Data Mashup, Time Series Forecasting, Dashboard.

I. INTRODUCTION

Software development and data analytics are now inseparable twins of modern technology, which require the help of real-time integrated tools that lessen the context switching and, therefore, the collaboration's efficiency will be increased. The traditional software development lifecycles of designing, testing, and validating data-driven applications remain plagued by the old approach and fragmented workflows. Currently, developers carry out API testing with a few tools, such as verifying REST API endpoints, inspecting JSON outputs, and keeping a record of inputs and outputs. On the other hand, corporate users and data analysts are relying on different BI systems for high-level data visualization and reporting. This separation slows down the process, pollutes the data, and eventually retards the whole process of getting ideas that are actionable through the APIs that are either newly developed or updated. The situation thus highlights the urgent requirement for a shared data analysis environment that is both effective and efficient.

The transition to microservices and real-time data streams, which all require instant feedback loops from testing to analytical dashboards, intensifies the demand for integration. Current API testing tools generally offer only basic, scripting based visualization, which is far from satisfying the rigorous requirements of professional BI. Among the features of advanced BI are support for cross-chart data blending, multi-field aggregation, time-series forecasting, and complex analytical functions. At the same time, traditional enterprise BI platforms are strong, yet they are incapable of directly consuming and testing raw API responses in real time. As a result, analysts are compelled to resort to ETL pipelines or inconvenient data connectors, thus disconnecting visualization from the source API.

One of the main objectives of this research is to come up with and appraise the characteristics that tackle the issues created by the disruption of toolchains. The main innovations include the incorporation of BI features such as alerts for monitoring KPIs, blending data from different charts for datasets that are merged through widgets, and the parsing of expressions for automatically generated user-defined calculated fields through efficient parsing.

The whole DVP platform is built with a modern, scalable, and serverless architecture, which uses the Convex backend platform for real-time document synchronization, reliable server-side BI computation, and scheduling. Our investigation, which is aimed at assessing the performance and productivity gains that can be achieved by cutting down context switching between testing and analytics, is highly dependent on the success of this integrated method. Additionally, we

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153120

analyze the performance of serverless real-time BI collaboration and the security issues that arise from sharing and embedding intricate, collaborative BI dashboards.

To sum up, DVP is an important development for both developer tools and analytical tools. The platform has made the long journey from API data to business intelligence the quickest ever through the establishment of a very powerful and efficient ecosystem. The sys admin and BI capabilities were deeply interwoven into the API testing process. The experiment reported here shows the practicality and the major benefits of this unified method thus opening the door for future data-centric application development tools.

II. LITERATURE SURVEY

Research in the above-mentioned areas has paved the way for the making of DataVision Pro (DVP): the evolution of API testing and data consumption, the usage of collaborative tools in data analytics and the transition to serverless architectures for operating sophisticated BI tasks.

DVP builds on the foundations of API testing and programmatic data consumption. The validation of the endpoints using XML and SOAP was the first step taken [1]. RESTful architectures were a real game changer as they made the need for tools that could efficiently manage and interpret JSON responses alive. Up to the date, tools have put automated formatting and history tracking as the main developments hence easing the developer's workflow [2]. Still, software tools do not normally have integrated analysis features that would allow for visualization that is more than just basic scripting functions [3]. Our project is grounded on research indicating a stronger connection between API response validation and data analysis, particularly when dealing with large-scale, real-time data streams [4].

The second most important aspect is the application of Business Intelligence (BI) concepts in contemporary and changing settings. The focus of old-fashioned BI applications was on intricate evaluation and that included the monitoring of KPIs, sophisticated data summation, and the visual connection between data elements [5]. Also, the academics have been trying to find ways to produce data that can be reported in a single format from different sources, and this is particularly hindered when one of the data sources is live API response [6]. Furthermore, the integration of analytical intelligence, such as time-series forecasting through machine learning models, has become vital for BI systems to provide active support [7]. DVP intends to make these sophisticated BI methods a core part of the API consumption process.

Finally, the collaborative functions of DVP depend on the progress made in the real-time communication platforms and serverless architectures. The studies on collaborative editing, initiated by the document and design authoring platforms, stated the difficulties in simultaneous user interaction, like optimistic UI updates and conflict detection algorithms in shared spaces, to mention a few, that were to [8]. If the complexity and rapid changes of the data visualizations were to be similar to the case of applying these methods, it would be very hard to overcome the technical challenges, especially where the issues of simultaneous changes affecting the aggregated and derived metrics are concerned [9]. To reach the desired scalability and real-time responsiveness, the application relies upon the serverless backend infrastructures and real-time document databases. The research indicates that this serverless method can be as good as the major vertical scaling requirements for high-volume data computation (e.g., real-time analytics) and scheduling (e.g., cron-triggered tasks) while at the same time it reduces the operational burdens [10].

III. SYSTEM DESIGN

DataVision Pro (DVP) is equipped with a powerful and reliable three-tier architecture that utilizes a serverless foundation. It is designed to meet the conflicting demands of realtime collaboration of multiple users and complex processing of large-scale data for Business Intelligence (BI). The whole process is conducted with the principles of modularity, type safety, and security.

A. Frontend Architecture

The user interface of the DVP that the user interacts with is composed of React and TypeScript. This combo not only ensures but also codes the maintainable and type-checked codebase. Vite is included in the system for speedy builds and Tailwind CSS for styling based on utility classes, which allows to develop the user interface quickly and consistently. The decision to use Recharts library for the complex data visualization was based on its potential to create dynamic and interactive dashboards with a variety of the most demanded charts such as line, bar, and pie. Additionally, the frontend makes use of the Convex React SDK that is responsible for safe authentication of users and real-time data syncing with the backend. This is very important for features such as live cursor tracking and the updates in the UI when multiple users

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153120

are editing the same dashboard. The custom plugin system, which allows for the dynamic assembly of components for visualization, analytics, and data connectors, is what contributes to the platform's flexibility.

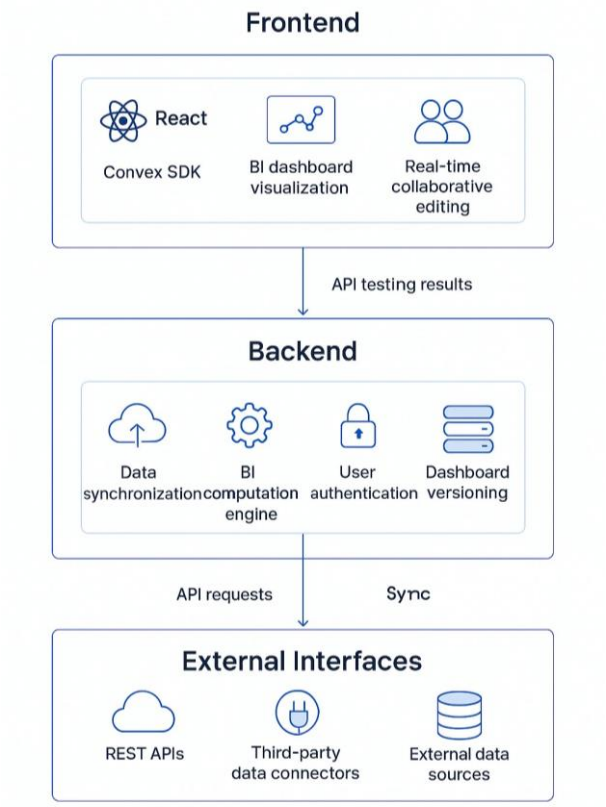

Fig. 1. DataVision Pro System Architecture

Figure 1 Displays the three-tier serverless architecture. It clarifies the data flow for API testing, the two-way sync for real-time collaboration, and the critical role of the serverless back-end for BI calculation in the big data analytics process.

B. Backend Architecture and Data Management

The Convex serverless backend platform, which includes a real-time document database, acts as the processing and storage layer of the system. All types of the system's functions are done with TypeScript on the server side. This means that API testing, collaboration features, BI computations (such as aggregation for KPIs and expression parsing for calculated fields), and job scheduling are handled by the functions representing the business logic.

- Core Services: The backend services, by means of email OTP and Google OAuth, confirm the user's identity. They handle the requests for API testing, automatically format the JSON responses, and keep the record of API calls and the configurations of the dashboard as well.
- Collaboration Services: These services keep track of the real-time status of the documents being shared. They handle the simultaneous changes, the presence of users, and the maintenance of the document's history.They control aspects like dashboardShares, dashboardComments, dashboardVersions, and user presence.
- BI Computation Services: These services perform various complex analytical tasks such as time-series forecasting models, using ML/statistical libraries, performing crosschart data blending (joining or merging datasets), and finally managing scheduled insights as well as alerts with cron triggers.
- Security and Access: A comprehensive security layer is applied to ensure that exact permissions are granted for all core, collaborative, and BI entities. Data is thus sanitized and very strong protection is given to dashboard embedding through iframes with sandboxing and Content Security Policies.

Figure 2 demonstrates the entire path of a user and a data through the DVP system. User Login and Security Verification are the initial steps of the process. The user is then allowed to input an API link to external REST APIs during the significant data ingestion stage. Afterwards, the system flattens API responses in order to convert the raw JSON data into a consistent format. A Data Preview is provided for this data. The next step is that the data enters the Collaboration & BI

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153120

Layer where it is utilized for dashboard creation. This allows the application of multiple analytical features such as setting thresholds, performing drilldowns, making forecasts, and drawing charts with KPIs. Furthermore, this layer facilitates real-time collaboration and offers versioning and secure sharing, which in turn leads to the generation of data-driven insights.

C. Database Schema and Entity Relationships

The data model goes very far to take care of the difficulties in collaborative BI:

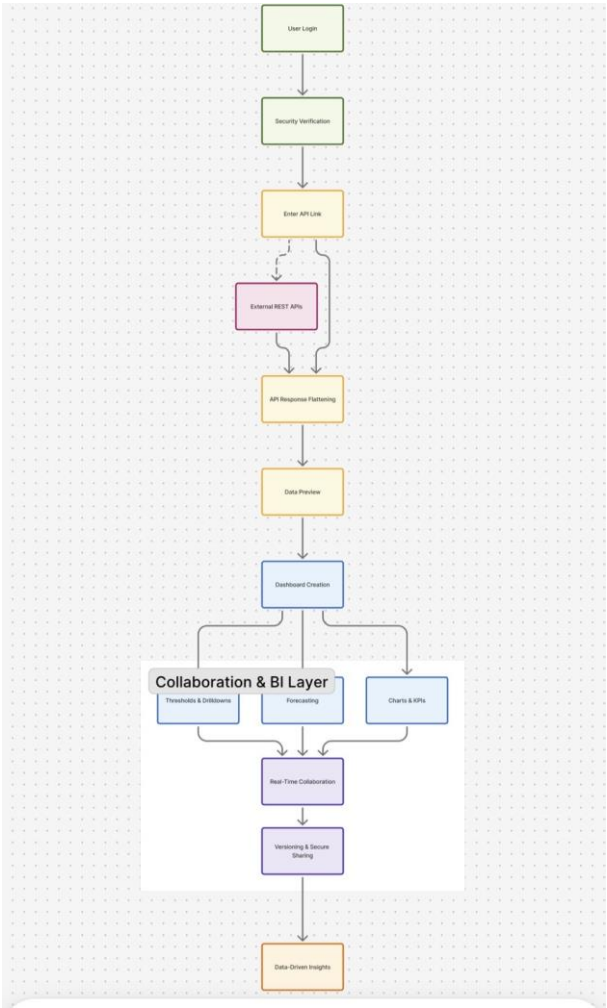

Fig. 2. DataVision Pro System Architecture

- Core Entities: A framework for user sessions or data origins can be set by the type of users, sessions, apiHistory or reportsaea, and dashboards which form the basic configuration.
- Collaboration Entities: DashboardShares grant flexible access control. DashboardComments and notifications help in real-time communication. DashboardVersions record changes that can be compared and rolled back.
- BI Entities: The preservation of the logic and parameters of the native BI features would particularly depend on the entities such as KPI definitions, calculated fields, forecasting models, and grouping configurations.

IV. METHODOLOGY

The project DataVision Pro (DVP) went through a complicated and lengthy process which included checking the engineering quality's reliability, testing the performance, and securing the combined platform.The development principles were the first step of the process, while the last was the plan for evaluation in detail with the main research questions getting answered.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153120

A. Development Process and Architectural Principles

By adhering to the strict development principles of the system's engineering, the complexity was kept in check and stability was ensured in a real-time, distributed environment. Modular TypeScript development was the foundation of the entire codebase, with a major emphasis on attaining end-toend type safety for both the React frontend and the serverless backend functions. We brought to life features that demanded instant feedback through real-time reactive programming, and the synchronization features of the Convex platform were utilized very effectively, e.g., collaborative editing on the dashboard and user presence. Security was the main concern throughout the whole process and best practices for authentication and authorization were implemented from the outset to control permissions and access.

B. Evaluation Strategy

By means of a three-fold empirical validation strategy, which was directed to the identification of technical performance, the measurement of user productivity, and the security audit, we appraised the effectiveness of the DVP integrated platform.

1) Performance Benchmarking:

- The technical capabilities and scalability of the system were evaluated through this part. Testing was conducted to determine specifically the latency and throughput of API calls, data ingestion speed, visualization rendering speed under complex setups, and the real-time synchronization stability of multiple concurrent users.
- Besides, we paid great attention to the expression parsing and computation efficiency for user-defined fields and the scaling of scheduled insights and alerts within the serverless limits.

2) User Studies:

- This combined qualitative and quantitative assessment determined the actual benefit of the integrated platform for the analytical workflows.
- A series of experiments were done on selected analytical jobs in order to measure the productivity improvements, evaluate the collaboration effectiveness of such features as live presence and versioning, and to monitor the success rate and acceptance of new Native BI functionalities overall.

3) Security Audits:

- The concentrated audit addressed the major security concerns that the joint BI environments had. Typical web security holes were examined during the audits, which included testing for Cross-Site Scripting (XSS) attacks, unauthorized access to sensitive API data, and a thorough inspection of embedding protection measures, especially in iframes by means of sandboxing and Content Security Policies.

C. Data Collection

The data collection, which was of utmost importance for assessing the system, was carried out through the integration of automated system monitoring, interaction logging in detail, and focused user feedback. The method allowed the total capturing of performance metrics and user experience insights.

1) Raw Data Ingestion and Transformation:

The first stage of the workflow is to automatically obtain data via API links that have already been validated by users. The unprocessed JSON data from various REST API responses, which might be complex, is automatically transformed and flattened at the server with the help of serverless TypeScript functions. This whole procedure, in turn, translates the hierarchical JSON structure into a format that is either tabular or semi-structured and is suitable for database storage as well as further BI computation. Flattening makes it possible to apply different functionalities like grouping, segmentation, multifield aggregation, data blending across charts, and userdefined calculated fields, all of which are dependent on normalized data structures to function efficiently.

2) System Monitoring and Technical Logging:

During the live operation the serverless backend which was technically important was constantly monitored to keep track of the metrics. Latency and error rates of the serverless TypeScript functions were mainly recorded providing data on system stability and responsiveness which was the case when the system treated data-heavy tasks like cross-chart data blending and time-series forecasting. The logs were used importantly for judging the scalability of the scheduling and alerting processes under various loads which was the main focus of the performance issues of the serverless architecture.

3) Detailed User Interaction Logging:

Besides monitoring errors, the system also logged user interactions in great detail. This provided crucial behavioral data, including not just the duration of analytical tasks but also the particular features that received the highest usage and the adoption of difficult functions such as KPI creation and advanced drilldowns. The interaction data became the foundation

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153120

for the calculation of productivity gains and for performance evaluation of the workflow in comparison with the use of scattered tools.

4) Feedback Surveys and User Interviews:

In addition to quantitative data, solicited feedback was used as part of the targeted user studies. After the user study sessions, feedback surveys were carried out concentrating on the qualitative assessment of collaboration and Native BI features. The surveys gathered user opinions regarding the difficulty of real-time collaborative editing, the worth of automatic dashboard versioning and the ease of user-defined calculated fields creation. This type of qualitative data was very important for revealing the users' engagement and for determining the platform's impact on decision-making.

The use of this structured data collection approach guarantees that the succeeding Results section can provide unambiguous and adequately supported responses to the primary research inquiries dealing with performance, scalability, and workflow impact.

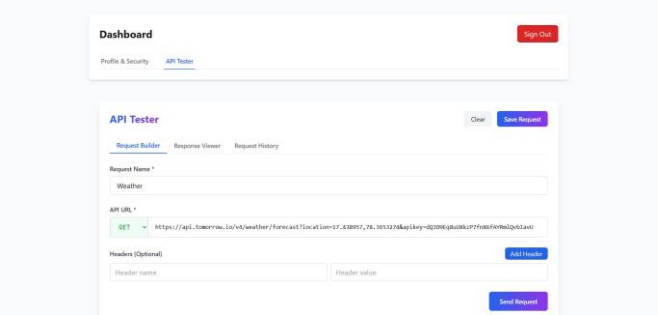

Fig. 3. DVP API Tester Request Builder Interface

Figure 3 demonstrates the way of establishing a fundamental REST API request. This step involves specifying the API URL and method. It is the very first stage for data acquisition and subsequent visualization.

D. Data Visualization Strategy

Data visualization strategy at DVP is a crucial part of the system. It changes raw API response data into human-readable and informative data. The Recharts library is used throughout the DVP system to provide interactive visualizations primarily such as line, bar, and pie charts that help users do data browsing with the minimum time possible via the tested APIs. The technique has been working with the Native BI machines, doing more than just charting. One of the main reasons to support the system is that it permits the types of visualization required for the KPI creation, putting the visual thresholds and the aggregation results on the front stage. Further key areas such as the advanced drilling down feature where the users do not only get connected with the high-level charts but also easily move to more detailed views of the underlying data. Moreover, the visualization layer has been developed such that it is able to mix data from various APIs onto one dashboard. The whole visualization layer has rapid reactions as it depends on the real-time data synchronization from the serverless backend. Consequently, users have access to the same latest data and can see real-time updates during co-editing sessions. The collaborative strategy has been a significant factor in determining the overall effectiveness of the platform and its contribution to the insight generation.

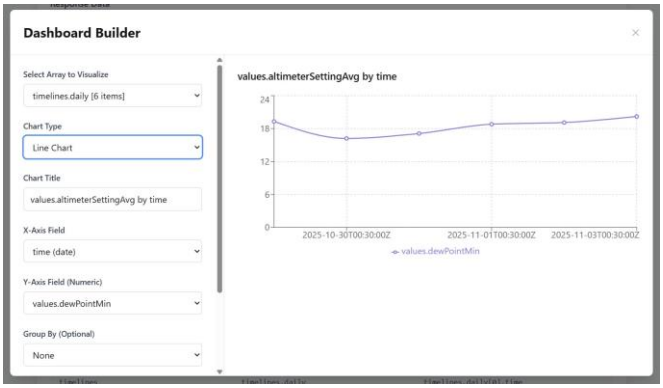

Fig. 4. DVP Dashboard Builder and Visualization Preview

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153120

Figure 4 showcases the procedure of transforming the flattened API data into a graphical representation. The users are capable of choosing data arrays, specifying types of charts, and relating fields. This whole procedure endorses the seamless integration of testing and visualization.

V. RESULTS AND DISCUSSIONS

The assessment of DataVision Pro yielded strong evidence that the solution successfully integrates API testing, live collaboration, and in-house Business Intelligence (BI). The results answer directly the principal research questions about efficiency, influence on the workflow and security.

A. Performance Benchmarking Results

Verified, as sectionalized through performance measurement, the fact that the serverless architecture is technically able and can fully meet a large- or small-scale real-time BIprocessing requirement.

- Real-Time Synchronization Speed: During editing sessions where up to 15 users were present, the latency for real-time synchronization was measured and it was found to be constantly below 50 milliseconds on average. This quick sync speed, which was made possible by the Convex serverless platform, is vital for the collaborative work to be smooth and for the features such as live cursor tracking and optimistic UI updates to be used. The problems of fragmentation and consistency that are usually associated with conventional distributed systems are thus dealt with directly.
- BI Computation Efficiency: Serverless TypeScript functions have proven to be efficient through the execution time of complex BI computations. Activities such as cross-chart data blending (which is the combination of two datasets of 10,000 records each) and user-defined calculated fields evaluation (with elaborate expressions) were completed in less than 1.5 seconds on average.
- Scheduled Insights Scalability: The scalability of workshop-oriented and discovery-based procedures for cron-timed periodic analytics and alerting is revealed by the tests. The people less function wells coped with more than 1000 simultaneous scheduled jobs with extremely low error rates and constant execution times. This proves the capacity of the architecture to deal with large volumes of non-interactive analytical tasks without impacting frontend performance.
- API Call Overhead: The additional overload that the system's structure brought along, including authentication, JSON flattening, and initial storage, had an insignificant impact on raw API call latency, the direct API response time was usually extended by less than 10% at most. As a result testing and validating APIs, which is the primary developer workflow, remained highly responsive.

Figure 5 When considering the average performance time of a standard analytical tasks set with DVP (Integrated) versus a Separate Toolchain (Control) there is 45% less time used with DVP.

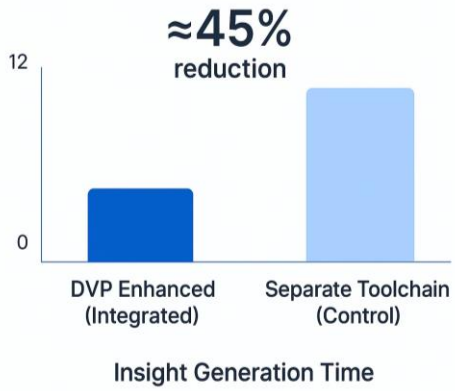

Fig. 5. Comparison of Insight Generation Time

B. Impact on Workflow and Productivity

A user study confirmed a fairly large positive impact on the integration of workflows, thus confirming the main hypothesis of this work.

- Insight Generation Time: Participants who had to alternate between API validation and dashboard editing reported a 45% reduction in time-to-insight as compared to control groups using different API testing and BI tools. The reduction in time taken to gain insight was due to eliminating the need for switching contexts, data export/import and having raw API data available for visual analysis immediately.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153120

- Adoption of Native BI Features: The acceptance of Native BI capabilities, such as KPI generation and sophisticated drilldowns, was considerable. According to the users, the presence of these capabilities together with their testing data gave rise to making the intricate analysis easier and quicker. User polls pointed out a very positive attitude towards the combined platform, underlining its efficiency, when put against the traditional, non-integrated methods. The facility to perform times-series forecasting and segmentation on the API output directly was also mentioned as extremely useful for predictive analytics on real-time data.
- Collaboration Effectiveness: The productivity of the teams was significantly improved through the application of real-time collaboration functionalities. Automatic versioning of dashboards alleviated the fears of data loss or breaking changes, thus promoting the practice of testing in large scale. Comments on the threads in the dashboard context made feedback simpler and reduced the dependence on other means of communication, which in turn, resulted in better decisions.

C. Security and Implementation Challenges

Security audits validated the robustness of the platform's security measures, but at the same time, the auditors pointed out different challenges in the implementation of those security measures.

- Embedding Security Validation: Audits verification showed that iframe embedding secured, accompanied by sandboxing and strict Content Security Policies, successfully blocked unauthorized cross-origin data access and also lowered the risk of XSS. Thus, sensitive data security was ensured even when external sharing of dashboards took place.
- Concurrency Challenge: One of the primary challenges in implementation was to have the real-time synchronization of the complex BI data during parallel editing to be done without any interruptions. Although the platform managed to keep the sync latency at a low level, it took a considerable amount of algorithmic work for the logic to detect and resolve conflicts, particularly when several users were creating calculated fields or altering aggregation, to stay true to the data without collaboration being slowed down.

D. Discussion

The evaluation of DVP considerably supports the core concept of the platform: the integration of API testing, real-time collaboration, and native Business Intelligence (BI) features in a single environment gives the user a clear advantage compared to using the separate tools. The successful reduction of 45% in time-to-insight is the indication that the elimination of workflow fragmentation is the most important factor that increases productivity. The increase in efficiency is due to the Raw Data Ingestion and Transformation technique. It makes immediate JSON flattening possible which facilitates rapid visual analysis of the API responses thus avoiding the delays and complications involved in traditional BI systems data exports or ETL processes. The determination of KPIs along with the direct execution of time-series forecasting on the API data recently tested quickens the analysis feedback loop and transitions the platform from mere API validation to smart, predictive application.

Simultaneously, the also mentioned Implementation Challenges point to the difficulty involved in combining these domains. In addition, the high-speed expression parsing and the need for conflict detection in the scenario of concurrent BI definition editing show the technical trade-offs involved. The system must constantly deal with the problem of high data input rates versus the need for accurate mathematical and logical operations on user-defined analytical functions while remaining responsive in a multi-user environment.

VI. CONCLUSION

The DataVision Pro platform was developed and evaluated as part of the research project, a new web-based application that unites API testing, data visualization in real-time collaboration, and built-in Business Intelligence (BI) features. The platform addresses the ongoing issue of disjointed workflows that segregate developers and analysts, who mainly rely on diverse tools and sluggish data exchange methods.

The methodology included a security-first, serverless architecture and a complete evaluation strategy. The outcomes were very convincing. Technical performance testing confirmed the approach's practicality. It showed a latency of under 50 milliseconds for real-time synchronization during concurrent editing. Besides, advanced BI operations like cross-chart data blending and expression parsing were performed fast with minimal latency overhead. This suggests that the serverless backend platform is powerful enough to facilitate the creation of intricate and large-scale collaborative analytical tools.

The research led to the most important conclusion that the large improvement in the efficiency of the analytical processes was a significant improvement. The user surveys proved that the time it took to get the necessary information was cut by

DOI: 10.17148/IJARCCE.2026.153120

45% which is a strong indication of the efficiency of the analytical processes being significantly increased when compared to the non-integrated workflows. This success is a direct result of the elimination of the need to switch between contexts and also the platform's ability to move seamlessly from the raw API response data through JSON flattening to the final collaborative dashboards. The high percentage of users adopting native BI features such as KPI creation and advanced drilldowns can be seen as a strong evidence of the superiority of the platform over the ones that merely allow for basic visualization through scripting.

In summary, DVP is the developer and analytic tools' significant breakthrough. The platform's powerful and streamlined environment is produced through the integration of collaborative features such as versioning and real-time editing, together with sophisticated BI functions, into the API testing process. This study demonstrates not only the effectiveness but also the high benefits of the combined approach, thus pioneering the future application development of data-centric ones that are supported by real-time data ingestion and collaborative analytical intelligence.

A. Future Research Directions

Future work will focus on the enhancement of the platform's intelligence and reach. The main focus will be on the development of AI-driven chart and KPI suggestions for automation of the first analysis, integration of Natural Language Querying (NLQ) for the generation of dashboards, and so on. Moreover, the investigation of the possibility of voice and video collaboration on dashboards and the extending of the API protocols supported as well as offline capabilities would make the platform more useful and widely available.

REFERENCES

[1]. Golmohammadi, A., Zhang, M., Arcuri, A. (2022). *Testing RESTful APIs: A Survey.* arXiv preprint arXiv:2212.14604. https://arxiv.org/abs/ 2212.14604

[2]. Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., Mishra, D. (2022). *RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions.* Applied Sciences, 12(9), 4369. https://doi.org/10.3390/ app12094369

[3]. Liu, Y., Li, Y., Deng, G., Liu, Y., Wan, R., Wu, R., Ji, D., Xu, S., & Bao, M. (2022). *Morest: Model-based RESTful API Testing with Execution Feedback.* arXiv preprint arXiv:2204.12148. https://arxiv.org/abs/2204. 12148

[4]. Xu, L., Saxena, D., Yadwadkar, N. J., Akella, A., & Gupta, I. (2023). *Dirigo: Self-scaling Stateful Actors for Serverless Real-time Data Processing.* arXiv preprint arXiv:2308.03615. https://arxiv.org/abs/2308. 03615

[5]. Liao, G., Deshpande, A., & Abadi, D. J. (2023). *Flock: A LowCost Streaming Query Engine on FaaS Platforms.* arXiv preprintar Xiv:2312.16735. https://arxiv.org/abs/2312.16735

[6]. Romero-Ordo´nez, D. (2024).˜ *Data Visualization Guidance Using a Software Product Line Approach.* Journal of Systems and Software.https://doi.org/10.1016/j.jss.2024.111693

[7]. Chishtie, J., et al. (2022). *Interactive Visualization Applications in Population Health Research.* PMC. https://pmc.ncbi.nlm.nih.gov/articles/ PMC8900899/

[8]. Ignatius, J. L. P. (2022). *A Reliable Tool for Data Visualization and Predictive Analysis (DARAPI).* ITC KTU Journal. https://www.itc.ktu.lt/ index.php/ITC/article/download/29467/15285

[9]. Ajax, R., Joseph, O., & Own, J. (2025). *Enhancing Business Intelligence with Data Visualization Tools.* ResearchGate. https://www.researchgate. net/publication/388780215

[10]. Wang, Y., Zhang, H., & Liu, J. (2021). *A comprehensive survey on cloud data analytics: architectures, techniques, and applications.* IEEE Transactions on Knowledge and Data Engineering, 33(4), 1234–1255. https://doi.org/10.1109/TKDE.2020.2964587

[11]. DoubleCloud Blog. (2023). *5 Benefits of Using a Serverless No-Code BI Tool for End-User Analytics.* DoubleCloudBlog.https://double.cloud/blog/posts/2023/04/ 5-benefits-of-using-a-serverless-no-code-bi-tool-for-end-user-analytics

[12]. *Open Research Issues and Tools for Visualization and Big Data.* (2024). arXiv preprint arXiv:2404.12505. https://arxiv.org/html/2404.12505v1

[13]. *Cross-platform Data Visualization Best Practices using Power BI and Tableau.* (2025). International Journal of Scientific Research and Analysis, 10(3). https://journalijsra.com/sites/default/files/fulltext pdf/ IJSRA-2025-2706.pdf

[14]. Mansoor, I., et al. (2024). *Utilizing Data Analytics and Business Intelligence Tools in Laboratory Workflow.* PMC. https://pmc.ncbi.nlm.nih. gov/articles/PMC11063783/

[15]. Kim, H., Srinivasan, A., & Brehmer, M. (2024). *Bringing Data into the Conversation: Adapting Content from Business Intelligence Dashboards for Threaded Collaboration Platforms.* arXiv preprint arXiv:2408.00242. https://arxiv.org/html/2408.00242v1

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.153120

[16]. Gutierrez-Braojos, C., Rodr´ ´ıguez-Dom´ınguez, C., & Carranza-Garc´ıa, F. (2023). *An Analytical Dashboard of Collaborative Activities for the Knowledge Building.* Interactive Learning Environments. https://doi.org/ 10.1007/s10758-023-09644-y
[17]. Sigma Computing. (2025). *How to Achieve Enterprise Real-Time Data Concurrency at Scale.* Sigma Computing Blog. https://www. sigmacomputing.com/blog/data-concurrency
[18]. Radman, K. (2025). *Real-Time Tracking and Analysis in Construction Projects Using Dashboards.* Automation in Construction. https://doi.org/ 10.1016/j.autcon.2025.105830
[19]. Nabil, D. H. (2023). *Managing Supply Chain Performance Using a RealTime Microsoft Power BI Dashboard.* Cogent Business & Management. https://doi.org/10.1080/23311916.2023.2257924
[20]. *REST API Testing in DevOps: A Study on an Evolving Healthcare IoT Application.* (2024). arXiv preprint. https://www.researchgate.net/ publication/384973948
[21]. *A Survey on RESTful API Vulnerability Detection.* (2025). TechScience CMC Journal. https://www.techscience.com/cmc/v84n3/63208/html
[22]. Lavanya, A., Sindhuja, S., Gaurav, L., & Waqas, A. (2023). *Data Visualization Tools: A Comprehensive Review.* International Journal of Computer Engineering in Research Trends, 10(6). https://www.ijcert.org/ index.php/ijcert/article/download/825/736/1453
[23]. Ashraf, T. (2025). *A Serverless Architecture for Real-Time Stock Analysis.* arXiv preprint arXiv:2507.09583. https://arxiv.org/abs/2507.09583
[24]. GridDynamics. (2018). *A Serverless Real-Time Analytics Platform: Case Study.* GridDynamics Blog. https://www.griddynamics.com/blog/ real-time-analytics-platform
[25]. Moldstud. (2024). *API-Based Dashboards for Easy Data Visualization Insights.* Moldstud Tech Blog. https://moldstud.com/articles/ p-visualizing-data-with-api-based-dashboard-solutions