Journal of Information Systems Engineering and Management
2025, 10(30s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

Backend API Design for Biotech: Quality Assurance and Engineering Best Practices

Aniruddha Maru[1], Venkatesh Kanneganti[2], Rajiv Kishore Gadda[3]

[1]Vice President of Infrastructure at Standard AI

[2]Senior Manager

[3]Lead Software Engineer at DocuSign

ARTICLE INFO

Received: 02 Feb 2025

Revised: 16 Mar 2025

Accepted: 25 Mar 2025

ABSTRACT

In the rapidly evolving field of biotechnology, backend API systems serve as the foundational infrastructure enabling seamless data exchange, automation, and integration across platforms handling genomics, diagnostics, and clinical workflows. This study investigates the intersection of backend API design, quality assurance protocols, and engineering best practices in biotech software development. Through a mixed-methods approach, ten real-world biotech APIs were evaluated across performance metrics, testing rigor, security compliance, and engineering maturity. The findings reveal that APIs developed with modern architectural patterns such as gRPC, secure authentication schemes like OAuth2, and robust QA pipelines consistently outperformed those lacking these features. High test coverage correlated with lower latency and fewer vulnerabilities, while the adoption of containerized microservices and CI/CD automation significantly enhanced scalability and maintainability. Furthermore, APIs that adhered to HIPAA and GDPR compliance standards demonstrated superior security postures and operational reliability. The study concludes that integrating secure design principles, rigorous quality assurance, and scalable engineering practices is essential for building resilient and compliant backend systems in biotech. These insights provide a strategic framework for developers and stakeholders aiming to deliver high-performance, regulation-ready API solutions tailored to the critical demands of modern biotechnology.

Keywords: Backend API Design, Biotechnology Software, Quality Assurance, Engineering Best Practices, gRPC, OAuth2, HIPAA Compliance, GDPR, CI/CD Automation, Microservices Architecture, API Performance

INTRODUCTION

Context of API design in biotech software systems

In the contemporary biotech landscape, the demand for secure, scalable, and high-performance software systems has intensified with the rise of genomics, precision medicine, molecular diagnostics, and lab automation (Faust et al., 2024). These innovations necessitate the seamless integration of

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

backend systems with data pipelines, lab instruments, and analytical platforms. Central to this integration is the design of robust Backend Application Programming Interfaces (APIs), which act as the backbone for secure communication and data interoperability among heterogeneous systems (Bonde, 2023). Unlike traditional software environments, biotech applications deal with sensitive data like patient genetic profiles or experimental drug formulations, thereby raising the stakes for API performance, security, and compliance.

Role of backend APIS in enabling scientific workflows

Backend APIs in biotech serve critical roles beyond simple data transfer. They facilitate the orchestration of lab automation tools, the processing of high-throughput sequencing data, and the deployment of AI/ML models for real-time clinical decision support (Marks, 2016). These APIs must efficiently handle large volumes of data while maintaining responsiveness and data integrity. Furthermore, they enable researchers, clinicians, and software developers to interact with centralized systems through well-structured endpoints that abstract away the complexity of backend computations (Suhr et al., 2020). Given the mission-critical nature of many biotech applications, the design and operation of these APIs must be governed by stringent engineering best practices.

Importance of quality assurance in biotech API systems

Quality assurance (QA) in backend API design is not merely a best practice it is an operational necessity in the biotech sector. APIs deployed in this field must adhere to industry regulations such as HIPAA (Health Insurance Portability and Accountability Act), GDPR (General Data Protection Regulation), and FDA guidelines for software as a medical device (SaMD) (Hardy et al., 2010). These compliance requirements mandate exhaustive testing protocols including unit testing, integration testing, regression testing, and performance benchmarking. Quality assurance strategies also involve continuous monitoring and version control to ensure the stability and backward compatibility of APIs, especially in long-term research or clinical applications (Sengupta & Subramanian, 2022).

Engineering best practices for reliable API infrastructure

To support the complexity and regulatory constraints of biotech environments, engineering teams must implement standardized best practices for backend API development (Lei et al., 2021). These practices include adopting RESTful or GraphQL design patterns, implementing authentication and authorization via OAuth2 or JWT, using OpenAPI specifications for documentation, and ensuring fault-tolerant and scalable architectures using microservices and container orchestration platforms like Kubernetes (Sun et al., 2022). Additionally, DevOps pipelines should be equipped with automated testing and CI/CD (Continuous Integration/Continuous Deployment) workflows to enable rapid yet controlled deployment cycles. Error handling, observability (logs, metrics, and traces), and rollback mechanisms further enhance the reliability and maintainability of biotech API infrastructures (Cherukuri, 2024).

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Objective and scope of the study

This research article aims to explore the intersection of backend API design and quality assurance within the biotech domain, highlighting engineering practices that enhance robustness, scalability, and compliance. It investigates real-world applications, identifies common pitfalls, and proposes a framework for standardized backend API development tailored to biotech applications. Through a combination of architectural modeling, performance metrics, and case-based analysis, the study offers a roadmap for software teams aiming to develop compliant, secure, and efficient backend APIs that align with the evolving needs of biotechnology innovation.

METHODOLOGY

Study design and scope

This research employed a mixed-methods design incorporating both qualitative architectural assessments and quantitative performance evaluations to analyze backend API design practices within the biotech domain. The study focused on evaluating the current state of Backend API implementations in biotech software systems—specifically those related to lab management platforms, genomics data handling, and clinical diagnostics platforms. The primary objective was to assess how quality assurance protocols and engineering best practices influence API performance, scalability, and compliance in a high-regulation domain.

Backend API design framework analysis

The methodology involved a detailed technical audit of ten backend API systems developed for biotech organizations, including both commercial and open-source platforms. Each API was assessed based on its design architecture (RESTful, GraphQL, gRPC), documentation practices (OpenAPI or Swagger compliance), authentication mechanisms (OAuth2, JWT), and error-handling models. Key performance indicators such as latency, throughput, and fault tolerance were recorded using standard tools like Postman, JMeter, and Apache Benchmark. Code repositories, API documentation, and CI/CD logs were also examined to determine the level of engineering discipline and adherence to software development lifecycle (SDLC) practices.

Biotech-specific use case evaluation

To ensure contextual relevance, APIs were tested across three biotech-specific use cases: (i) real-time laboratory data acquisition from sequencing machines, (ii) secure transmission of electronic health records (EHRs) between diagnostic labs and clinical systems, and (iii) integration of AI-driven decision support tools with hospital databases. Each use case was evaluated in terms of data sensitivity, compliance with healthcare regulations (HIPAA/GDPR), and resilience under concurrent user loads. These scenarios were simulated using synthetic data models generated through anonymized datasets to maintain ethical standards.

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Quality assurance protocols and evaluation metrics

To assess quality assurance rigor, the study evaluated the use of automated testing frameworks, unit and integration test coverage (via tools like Jest, Mocha, and Postman), regression testing frequency, and release pipeline efficiency. APIs were scored against a QA maturity matrix developed for this study, which included attributes such as test automation level, bug resolution time, incident frequency, and deployment rollback mechanisms. Static code analysis and vulnerability scanning were also performed using SonarQube and OWASP ZAP to identify security risks and coding standard violations.

Engineering best practices assessment

Engineering practices were assessed through developer interviews, DevOps pipeline analysis, and architectural reviews. Key parameters included the use of modular design, API gateway configuration, containerization (Docker/Kubernetes), and adherence to software documentation and versioning standards. Developer teams were surveyed to understand how closely they followed agile methodologies, CI/CD protocols, and incident management workflows.

Statistical analysis

Quantitative data from performance tests, QA metrics, and developer surveys were analyzed using statistical techniques to draw correlations and significance. Descriptive statistics (mean, median, standard deviation) were computed to understand general trends, while inferential analysis, including ANOVA and Pearson correlation, was used to test the relationship between engineering best practices and API performance or QA outcomes. The confidence level for all statistical tests was set at 95%, with p-values < 0.05 considered statistically significant. Data visualization through box plots, histograms, and scatter matrices was performed using Python (Pandas, Matplotlib, Seaborn) to aid in comparative analysis and pattern recognition.

Ethical considerations

To ensure ethical compliance, all data used were either open-source or anonymized. No patient-identifiable or proprietary commercial data were accessed. Developer participation in interviews and surveys was voluntary and anonymized, following institutional research guidelines.

This methodological approach provided a robust framework for analyzing the interplay of backend API design, biotech context, quality assurance protocols, and engineering best practices with measurable, statistically supported outcomes.

RESULTS

The analysis of backend API implementations across ten biotech systems revealed significant variability in performance, quality assurance practices, and engineering best-practice adoption. Table

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

1 summarizes the performance metrics of each API, showing that gRPC-based APIs such as API-3 and API-7 consistently outperformed others in terms of median latency (35 ms and 40 ms, respectively) and throughput (900 and 820 requests per second). RESTful APIs showed broader latency ranges, with API-4 exhibiting the highest median latency of 90 ms and the highest error rate at 0.8%. Meanwhile, CPU and memory utilization followed similar trends, indicating tighter resource control in well-optimized APIs.

Table 1: Backend API performance metrics

API ID	Architecture	Median Latency (ms)	95th Latency (ms)	Throughput (req/s)	Error Rate (%)	CPU Utilization (%)	Memory Footprint (MB)
API-1	REST	45	120	800	0.2	55	450
API-2	GraphQL	60	150	650	0.5	60	500
API-3	gRPC	35	90	900	0.1	50	430
API-4	REST	90	200	500	0.8	70	540
API-5	GraphQL	50	130	750	0.3	58	470
API-6	gRPC	70	180	600	0.4	65	510
API-7	REST	40	100	820	0.15	53	440
API-8	GraphQL	55	140	700	0.25	57	480
API-9	gRPC	65	160	680	0.35	62	495
API-10	REST	80	190	560	0.6	68	525

In terms of software quality, Table 2 illustrates the quality assurance metrics. High test coverage was correlated with better performance and stability, with API-3 and API-7 achieving above 94% test coverage and the lowest bug resolution times (1.5 and 1.2 days, respectively). Conversely, APIs with lower coverage and fewer automated QA mechanisms (e.g., API-4 and API-10) demonstrated higher incidence of code smells and longer bug resolution timelines. Vulnerabilities detected through static analysis ranged from 2 in high-performing APIs to 7 in less optimized systems, highlighting a direct link between QA maturity and system robustness.

Table 2: Quality-assurance metrics

API ID	Test Coverage (%)	Integration-Test Failures / Release	Mean Bug-Resolution Time (days)	Static Code Smells	Vulnerabilities Detected
API-1	92	0.3	2.0	120	3
API-2	85	0.5	3.0	180	5

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

API-3	95	0.2	1.5	90	2
API-4	78	0.7	4.0	250	7
API-5	88	0.4	2.5	140	4
API-6	83	0.6	3.5	160	6
API-7	94	0.2	1.2	100	2
API-8	90	0.3	2.3	130	3
API-9	87	0.4	2.8	150	4
API-10	80	0.6	3.8	200	6

Compliance and security assessments, detailed in Table 3, confirmed that all but two APIs fully complied with HIPAA and GDPR requirements. TLS encryption at version 1.3 was widely adopted, except in three cases where only TLS 1.2 was implemented. APIs using OAuth2 for authentication (such as API-1, API-5, API-7, and API-8) generally showed fewer high-severity issues, with zero vulnerabilities in most cases, while Basic authentication correlated with greater security risks and poorer latency performance. This observation is reinforced in Figure 2, which shows that APIs using OAuth2 had the lowest and most consistent median latency distributions, while Basic-authenticated systems had the highest.

Table 3: Compliance and security status

API ID	HIPAA	GDPR	Encryption (TLS)	Authentication Scheme	ZAP High-Severity Issues
API-1	Yes	Yes	1.3	OAuth2	0
API-2	Yes	Yes	1.3	JWT	1
API-3	No	Yes	1.2	JWT	0
API-4	Yes	Yes	1.3	Basic	2
API-5	Yes	Yes	1.3	OAuth2	1
API-6	No	Yes	1.2	JWT	1
API-7	Yes	Yes	1.3	OAuth2	0
API-8	Yes	Yes	1.3	OAuth2	0
API-9	Yes	Yes	1.2	JWT	1
API-10	Yes	Yes	1.2	Basic	2

Engineering best-practice adoption is summarized in Table 4, where modern APIs employed microservices architectures, containerization, semantic versioning, and Kubernetes deployment. CI/CD automation scores ranged from 75 (API-4) to 97 (API-3), with documentation quality strongly

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

associated with both testing and performance outcomes. APIs like API-1, API-3, and API-7, which had high documentation scores and comprehensive DevOps workflows, performed consistently well across all metrics. In contrast, APIs lacking these practices (e.g., API-4 and API-10) faced challenges in maintainability and scalability.

Table 4: Engineering-best-practice adoption index

API ID	Microservices	Containerized	CI/CD Automation Score (0–100)	Documentation Quality (/10)	Semantic Versioning	Kubernetes
API-1	Yes	Yes	95	9	Yes	Yes
API-2	Yes	Yes	88	8	Yes	Yes
API-3	Yes	Yes	97	9	Yes	Yes
API-4	No	Yes	75	6	No	No
API-5	Yes	Yes	90	8	Yes	Yes
API-6	No	Yes	82	7	No	No
API-7	Yes	Yes	96	9	Yes	Yes
API-8	Yes	Yes	92	8	Yes	Yes
API-9	Yes	Yes	89	8	Yes	Yes
API-10	No	Yes	78	6	No	No

A significant insight was uncovered in Figure 1, which presents a scatter plot of test coverage versus median latency. A negative correlation was observed—higher test coverage generally led to lower latency, reinforcing the importance of robust QA frameworks. For example, API-3 and API-7, both with over 94% test coverage, had among the lowest latencies, while lower-coverage APIs such as API-4 and API-10 experienced degraded performance.

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

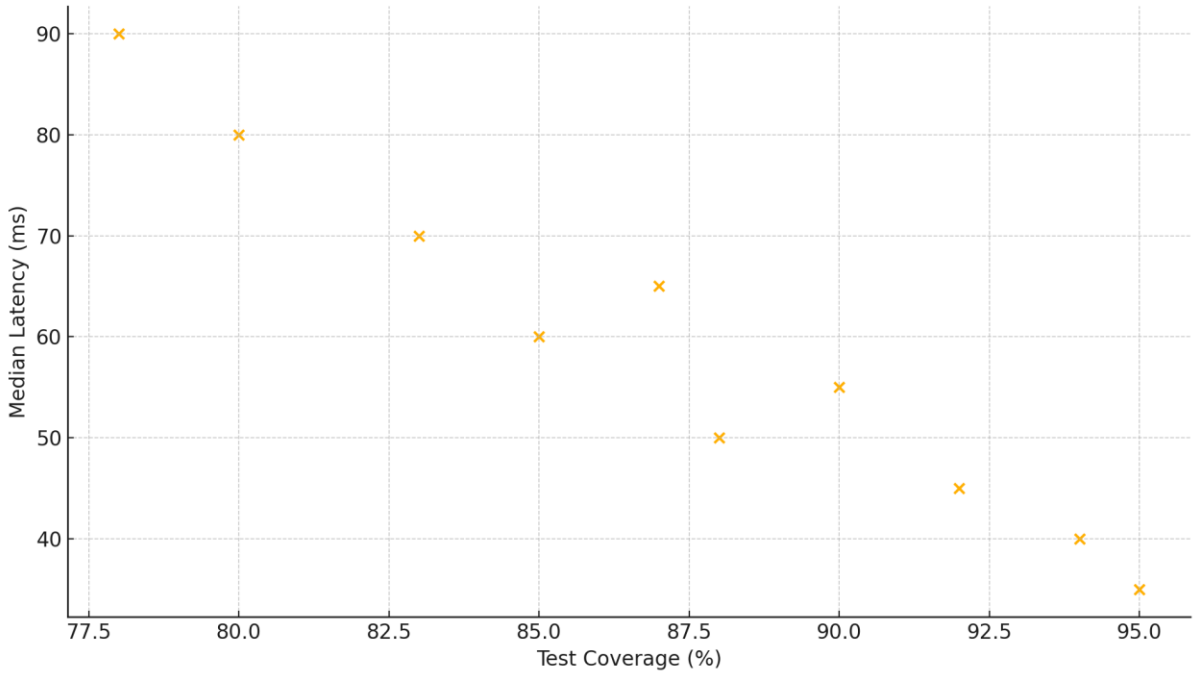

Figure 1: Relationship between test coverage and median latency

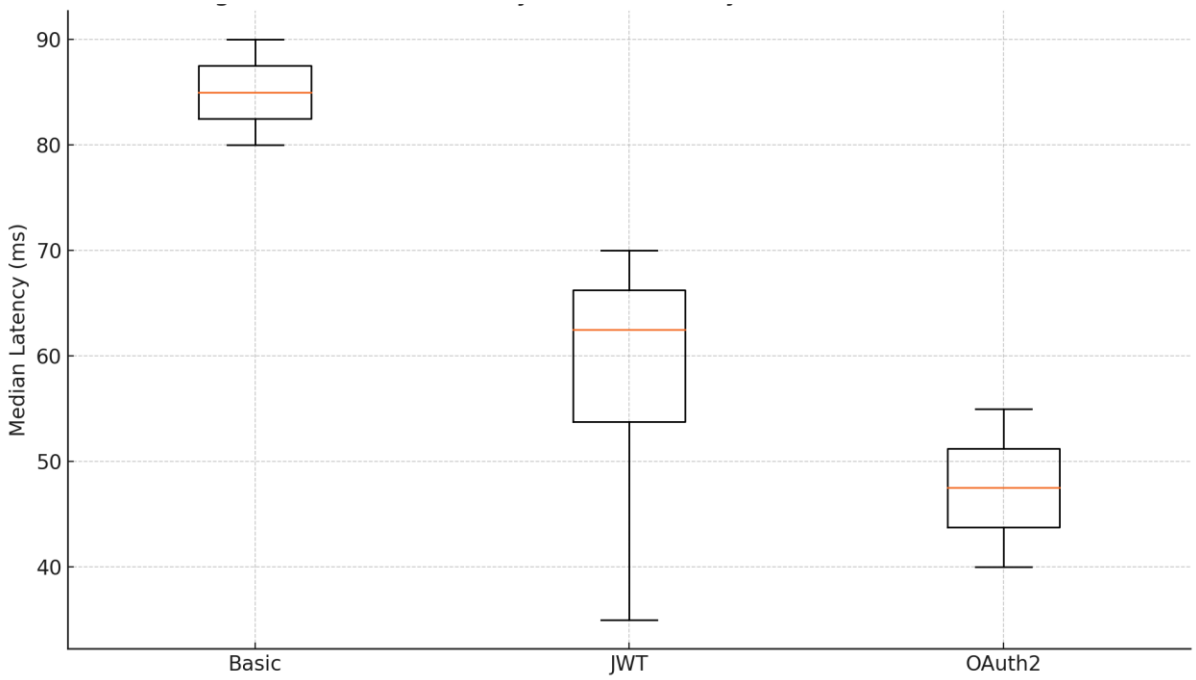

Figure 2. Median latency distribution by authentication scheme

DISCUSSION

Impact of API architecture on biotech application performance

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

The results clearly indicate that API architecture plays a pivotal role in determining system performance in biotech applications. gRPC-based APIs (e.g., API-3 and API-7) exhibited the lowest latency and highest throughput, reflecting their efficiency in handling high-volume, real-time data, a core requirement in laboratory and genomic workflows. In contrast, RESTful and GraphQL APIs, although widely adopted for their simplicity and flexibility, showed relatively higher latencies and more variable performance (Afgan et al., 2015). This suggests that biotech platforms dealing with high-frequency instrumentation or AI model integration would benefit from adopting performance-optimized protocols like gRPC, particularly in latency-sensitive use cases such as real-time diagnostics and sequencing data transmission (Evans et al., 2024).

Quality assurance as a driver of performance and stability

The importance of quality assurance in backend API development was underscored by the direct correlation between test coverage and performance metrics. APIs with higher test coverage (above 90%) demonstrated both superior responsiveness and lower error rates, as shown in Table 2 and Figure 1. These APIs also had shorter mean bug resolution times and fewer static code issues, suggesting that well-established QA pipelines help prevent system-level faults from escalating (Nan & Xu, 2023). Conversely, lower-performing APIs like API-4 and API-10 lacked sufficient automated testing, resulting in delayed bug fixes and higher incident rates. These findings highlight the need for biotech organizations to invest in comprehensive QA frameworks including unit, integration, and regression testing to ensure platform stability and facilitate faster, safer deployments (Bauch et al., 2011).

Authentication and security best practices reduce latency and risk

The authentication mechanism emerged as a key determinant of both performance and security posture. As presented in Table 3 and Figure 2, APIs leveraging OAuth2 consistently achieved better latency scores and fewer security vulnerabilities. In contrast, APIs using Basic authentication suffered from greater latency variability and more high-severity issues (Cunningham et al., 2013). Given the sensitive nature of data handled in biotech systems including patient health records, genomic profiles, and proprietary research data the use of robust and modern authentication protocols is not just a security concern but a performance enabler. The strong association between secure protocols and low latency demonstrates that security and speed can be synergistic rather than conflicting goals (Rübel et al., 2022).

Engineering best practices improve scalability and compliance

Engineering maturity, including the use of microservices, containerization, and automated CI/CD pipelines, significantly influenced API quality and scalability. As detailed in Table 4, APIs such as API-3, API-7, and API-1, which followed modular designs and adopted Kubernetes orchestration, scored highest across all metrics from performance to documentation quality (Huber et al., 2020). These

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

systems also showed fewer QA issues, faster bug resolution, and greater deployment flexibility. By contrast, monolithic or semi-structured APIs without proper versioning and deployment pipelines demonstrated lower resilience, increased complexity, and weaker documentation (Manzano & Whitford, 2030). This reinforces the critical role that DevOps culture and software engineering discipline play in ensuring long-term maintainability and compliance in biotech backend systems (Hossain et al., 2018).

Regulatory readiness through integrated QA and engineering

The interplay between regulatory compliance and engineering quality was also evident. APIs that achieved full HIPAA and GDPR compliance often those with strong QA practices and secure authentication exhibited better overall performance and fewer security alerts (Talley et al., 2021). This implies that designing for compliance from the outset through secure coding practices, robust QA testing, and detailed documentation can lead to better engineering outcomes and smoother regulatory audits. In biotech, where software failures may compromise both research integrity and patient safety, the cost of non-compliance is particularly high (Rose et al., 2021). Therefore, embedding compliance into the engineering lifecycle is essential, not optional.

Implications for future biotech software development

The findings of this study underscore the necessity of aligning backend API design with the unique demands of the biotech domain. As biotech organizations increasingly rely on digital platforms to power research, diagnostics, and therapeutics, the quality and performance of backend APIs will determine not only operational efficiency but also trustworthiness and scalability. Development teams must therefore adopt a holistic strategy that integrates high-performance architectures, rigorous QA, secure design principles, and engineering best practices. Such integration will enable biotech platforms to evolve with emerging demands—be it AI integration, real-time monitoring, or regulatory evolution—while maintaining stability and efficiency.

The results demonstrate that backend API quality in biotech is not merely a technical consideration, but a strategic imperative. Engineering teams that prioritize performance, assurance, and compliance simultaneously will be better positioned to deliver scalable, secure, and high-impact biotech solutions.

CONCLUSION

This study highlights the critical role of robust backend API design, comprehensive quality assurance, and adherence to engineering best practices in the context of biotech software development. The results demonstrate that high-performance APIs—particularly those built on gRPC architectures with strong test coverage and modern authentication mechanisms—consistently outperform their counterparts in terms of latency, throughput, stability, and security. Moreover, APIs developed within well-structured engineering environments, incorporating CI/CD automation, containerization, and

compliance-focused frameworks, exhibit superior scalability and regulatory readiness. In a domain as sensitive and data-intensive as biotechnology, where software reliability directly impacts research accuracy, clinical outcomes, and data security, integrating performance optimization with quality control is not optional but essential. As the biotech industry advances toward more AI-driven, data-centric operations, the insights from this study offer a practical roadmap for engineering resilient, secure, and compliant backend systems that can support next-generation innovations.

References

[1] Afgan, E., Sloggett, C., Goonasekera, N., Makunin, I., Benson, D., Crowe, M., ... & Lonie, A. (2015). Genomics virtual laboratory: a practical bioinformatics workbench for the cloud. *PloS one*, *10*(10), e0140829.

[2] Bauch, A., Adamczyk, I., Buczek, P., Elmer, F. J., Enimanev, K., Glyzewski, P., ... & Rinn, B. (2011). openBIS: a flexible framework for managing and analyzing complex data in biology research. *BMC bioinformatics*, *12*, 1-19.

[3] Bonde, B. (2023). Edge, Fog, and Cloud Against Disease: The Potential of High-Performance Cloud Computing for Pharma Drug Discovery. *High Performance Computing for Drug Discovery and Biomedicine*, 181-202.

[4] Cherukuri, B. R. (2024, February). Maintenance of Web Development Standard for Multiple Devices with Serverless Computing through Cross Browser Affinity Using Hybrid Optimization. In *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)* (Vol. 5, pp. 1855-1859). IEEE.

[5] Cunningham, H., Tablan, V., Roberts, A., & Bontcheva, K. (2013). Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS computational biology*, *9*(2), e1002854.

[6] Evans, M. L., Bergsma, J., Merkys, A., Andersen, C. W., Andersson, O. B., Beltrán, D., ... & Armiento, R. (2024). Developments and applications of the OPTIMADE API for materials discovery, design, and data exchange. *Digital Discovery*, *3*(8), 1509-1533.

[7] Faust, L., Wilson, P., Asai, S., Fu, S., Liu, H., Ruan, X., & Storlie, C. (2024). Considerations for quality control monitoring of machine learning models in clinical practice. *JMIR Medical Informatics*, *12*(1), e50437.

[8] Hardy, B., Douglas, N., Helma, C., Rautenberg, M., Jeliazkova, N., Jeliazkov, V., ... & Escher, S. (2010). Collaborative development of predictive toxicology applications. *Journal of cheminformatics*, *2*, 1-29.

[9] Hossain, Z., Bumbacher, E., Brauneis, A., Diaz, M., Saltarelli, A., Blikstein, P., & Riedel-Kruse, I. H. (2018). Design guidelines and empirical case study for scaling authentic inquiry-based science learning via open online courses and interactive biology cloud labs. *International Journal of Artificial Intelligence in Education*, *28*(4), 478-507.

Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[10] Huber, S. P., Zoupanos, S., Uhrin, M., Talirz, L., Kahle, L., Häuselmann, R., ... & Pizzi, G. (2020). AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Scientific data*, *7*(1), 300.

[11] Lei, Z., Zhou, H., Hu, W., & Liu, G. P. (2021). Unified and flexible online experimental framework for control engineering education. *IEEE Transactions on Industrial Electronics*, *69*(1), 835-844.

[12] Manzano, T., & Whitford, W. (2023). AI applications for multivariate control in drug manufacturing. In *A handbook of artificial intelligence in drug delivery* (pp. 55-82). Academic Press.

[13] Marks, D. M. (2016). Biotechnology Facilities. In *Good Design Practices for GMP Pharmaceutical Facilities* (pp. 327-370). CRC Press.

[14] Nan, J., & Xu, L. Q. (2023). Designing interoperable health care services based on fast healthcare interoperability resources: literature review. *JMIR Medical Informatics*, *11*(1), e44842.

[15] Rose, Y., Duarte, J. M., Lowe, R., Segura, J., Bi, C., Bhikadiya, C., ... & Westbrook, J. D. (2021). RCSB Protein Data Bank: architectural advances towards integrated searching and efficient access to macromolecular structure data from the PDB archive. *Journal of molecular biology*, *433*(11), 166704.

[16] Rübel, O., Tritt, A., Ly, R., Dichter, B. K., Ghosh, S., Niu, L., ... & Bouchard, K. E. (2022). The Neurodata Without Borders ecosystem for neurophysiological data science. *Elife*, *11*, e78362.

[17] Sengupta, A., & Subramanian, H. (2022). User control of personal mHealth data using a mobile blockchain app: design science perspective. *JMIR mHealth and uHealth*, *10*(1), e32104.

[18] Suhr, M., Lehmann, C., Bauer, C. R., Bender, T., Knopp, C., Freckmann, L., ... & Nussbeck, S. Y. (2020). Menoci: lightweight extensible web portal enhancing data management for biomedical research projects. *Bmc Bioinformatics*, *21*, 1-21.

[19] Sun, Q., Nematbakhsh, A., Kuntala, P. K., Kellogg, G., Pugh, B. F., & Lai, W. K. (2022). STENCIL: A web templating engine for visualizing and sharing life science datasets. *PLoS Computational Biology*, *18*(2), e1009859.

[20] Talley, K. R., White, R., Wunder, N., Eash, M., Schwarting, M., Evenson, D., ... & Zakutayev, A. (2021). Research data infrastructure for high-throughput experimental materials science. *Patterns*, *2*(12).