

## OPPORTUNITIES AND CHALLENGES OF LEVERAGING APIs FOR CDN-BASED MEDIA SYSTEMS

HENDRI <sup>1\*</sup>, RUKMI SARI HARTATI <sup>2</sup>, LINAWATI <sup>3</sup> and DEWA MADE WIHARTA <sup>4</sup>

<sup>1,2,3,4</sup> Udayana University.

Email: <sup>1</sup>hendriabubakar@mercubuana.ac.id (\*Corresponding Author), <sup>2</sup>rukmisari@unud.ac.id,

<sup>3</sup>linawati@unud.ac.id, <sup>4</sup>wiharta@unud.ac.id

ORCID ID: <sup>1</sup>0000-0000-0000-0000, <sup>2</sup>0000-0002-1268-9111, <sup>3</sup>0000-0001-8303-2822, <sup>4</sup>0009-0002-5034-9334

### Abstract

In today's digital era, Content Delivery Networks (CDNs) are crucial for efficient content delivery. Leveraging Application Programming Interfaces (APIs) in CDN-based media systems offers many opportunities, such as enhanced content delivery and improved user experiences. This research aims to identify the opportunities and challenges of leveraging APIs for CDN-based media systems. A qualitative approach is used by reviewing relevant literature and industry case studies. The results show that APIs can increase productivity and monetize digital assets but also face challenges such as the complexity of the CDN ecosystem and API management. In conclusion, while there are many opportunities in leveraging APIs for CDNs, media organizations need to address technical and operational challenges to maximize their benefits.

**Keywords:** Content Delivery Networks, API, Content Delivery, Media Systems, User Experience, Technology Integration.

**JEL classifications:** O32, O33.

### INTRODUCTION

The proliferation of diverse digital content, from video streaming to web-based applications, has been a driving force behind the remarkable growth of the internet as a critical infrastructure for global access to information and entertainment. To meet the increasing demand for efficient and high-performance content delivery, Content Delivery Networks have been widely adopted. As the CDN ecosystem continues to evolve, it has become increasingly complex, introducing both opportunities and challenges for media systems. The utilization of APIs can play a significant role in navigating this complex landscape, enabling media organizations to streamline content delivery, enhance user experiences, and explore new revenue streams.

One of the key advantages of leveraging APIs in CDN-based media systems is the ability to provide a stable and controlled interface for service provisioning, customer access, and third-party development (Lindman et al., 2020). APIs allow media organizations to expose their digital assets to internal teams or external parties, enabling the development of value-adding applications that can enhance the overall user experience (Lindman et al., 2020). Additionally, APIs can provide media organizations with a level of control and stability, encouraging wider adoption and reducing the risk of abuse (Lindman et al., 2020).

Furthermore, the use of APIs can enhance the productivity of media development teams by promoting reusability and enforcing consistency in new applications (Patni, 2023). Public APIs can also add value to media businesses by allowing third-party developers to integrate with

their services, potentially attracting new customers or expanding their reach (Patni, 2023).

However, the growing complexity of the CDN ecosystem also presents challenges for media systems leveraging APIs. The diverse range of CDN architectures and the evolving interconnection markets can create complexities in API management and integration (Stocker et al., 2017). Media organizations must carefully navigate these challenges to ensure seamless API-driven content delivery and a positive user experience.

As more scientific research and media content move towards distributed and cloud-based computing, the need for robust and trustworthy APIs has become increasingly important (Cleveland et al., 2020). Media systems must prioritize the development of well-designed, scalable, and secure APIs to support the complex technology stacks that underpin modern content delivery (Cleveland et al., 2020).

In the ever-evolving digital landscape, the integration of Content Delivery Networks and Application Programming Interfaces has become a crucial consideration for organizations seeking to optimize their online presence and deliver seamless user experiences. Digital platforms have long recognized the value of well-designed and effectively managed APIs, which often outperform traditional modes of service distribution (Wulf & Blohm, 2020). However, the challenges associated with designing successful APIs extend beyond technical considerations, as they must align with the overall business objectives and the demands of third-party developers and end customers (Wulf & Blohm, 2020).

The strategic significance of APIs in transforming industries through agile service development, specialization, scalability, and leveraging network effects has led many firms to overlook the importance of aligning API design with their business goals (Wulf & Blohm, 2020). This misalignment can result in suboptimal outcomes, such as the adoption of transaction-based pricing models that may not be well-suited to the provider's objectives (Wulf & Blohm, 2020).

In the context of Content Delivery Networks, the integration of APIs presents its own set of unique challenges. The rise of diverse content, from video to web applications, has been a major driver behind the phenomenal growth of the Internet and the emergence of CDNs as essential infrastructure (Stocker et al., 2017). The complexities of the CDN ecosystem, including the role of location and the evolving interconnection markets, further underscore the need for a strategic approach to API integration (Stocker et al., 2017).

To address these challenges, organizations must adopt a holistic approach that considers the technical, strategic, and operational aspects of API integration within their CDN infrastructure.

## LITERATURE REVIEW

### Data Transfer Theory

Data transfer is a fundamental aspect of information technology, enabling the exchange of information between various devices, systems, and networks. The term "technology transfer" can be used to describe the movement of new concepts, processes, or products from one area

of science or engineering to another, which is particularly relevant in the context of data transfer.

The process of technology transfer, including data transfer, involves various strategies, management approaches, and potential inhibiting factors. Technology transfer can occur in both vertical and horizontal dimensions. Vertical technology transfer involves the transfer of technical information and results from the research and development phase to the development and production phases, ultimately leading to the commercialization of the technology (Soltanzadeh & Soltanzadeh, 2018). On the other hand, horizontal technology transfer refers to the transfer of technology between different organizations, industries, or economies (Walker & Ellis, 2000).

The transfer of technology, including data transfer, is crucial in various industries, such as the pharmaceutical industry, where it is fundamental to the drug development process (Pavithra et al., 2021). The success of technology transfer, including data transfer, depends on the understanding of the process and the ability to predict its future prospects. However, the literature on technology transfer indicates a lack of empirically-based field studies, with the majority of the literature consisting of "wisdom" literature and case studies that provide insights into various aspects of the process (Douds, 1971).

Despite the importance of data transfer in information technology, there is a need for more comprehensive research and theory-building in this area to better understand the underlying mechanisms, strategies, and challenges involved in the technology transfer process.

### **The Importance and Benefits of Application Programming Interfaces**

Application Programming Interfaces have become an integral part of the modern digital landscape, providing a vital link between different software applications and enabling seamless data exchange and integration.

At its core, an API is a set of protocols, tools, and standards that governs how software components should interact with one another. APIs have evolved from being a technical solution to a strategic business tool, allowing organizations to unlock the value of their digital assets, extend their reach, and collaborate with partners more effectively (Sturm et al., 2017).

One of the primary benefits of APIs is their ability to enhance productivity and efficiency within an organization. By exposing certain functionalities or data through a well-designed API, development teams can leverage existing resources to build new applications more quickly, without having to reinvent the wheel (Patni, 2023). This promotes code reuse, ensures consistency across the organization, and ultimately streamlines the software development process.

Furthermore, APIs have become an essential mechanism for connecting businesses and driving innovation. Innovative companies are now using APIs as an interface to their business, allowing them to monetize their digital assets, expand their value proposition through partner-delivered capabilities, and engage with customers across multiple channels and devices (Patni, 2017, 2023).

This shift in the role of APIs has been driven by the growing need for interconnectivity and collaboration in the digital age. As businesses become increasingly interdependent, APIs have emerged as the de facto industry standard for integrating data and functionality across diverse application ecosystems (Sturm et al., 2017).

The technical benefits of APIs, such as providing a stable and controlled interface for service provision and third-party development, have been well-documented in the literature (Lindman et al., 2020). However, the strategic implications of API adoption are now being recognized, with APIs seen as an artifact that must be continuously enhanced to maintain their relevance and value (Lindman et al., 2020).

### **The Benefits of Content Delivery Networks**

Content Delivery Networks have become an integral part of modern web infrastructure, offering a wide range of benefits that have revolutionized the way content is delivered to end-users. One of the primary advantages of CDNs is their ability to improve content availability and reliability. By distributing content across a network of geographically dispersed servers, CDNs ensure that content is readily available to users, reducing the impact of server outages or network congestion at the origin server.

Another key benefit of CDNs is their ability to enhance content delivery performance. By serving content from servers located closer to the end-user, CDNs can significantly reduce latency and improve download speeds, providing a faster and more seamless user experience (Buyya et al., 2008). This is particularly important for content-heavy applications, such as video streaming, where even minor delays can have a significant impact on the user experience.

CDNs also play a crucial role in optimizing content delivery by employing various techniques, such as caching, compression, and load balancing (Buyya et al., 2008). Caching involves storing frequently accessed content on the CDN's edge servers, reducing the load on the origin server and improving response times. Compression techniques can further reduce the size of the content, leading to faster downloads, while load balancing distributes traffic across multiple servers, ensuring that no single server becomes overloaded. In addition to performance and availability benefits, CDNs can also contribute to cost savings for content providers. By offloading the majority of the traffic to the CDN infrastructure, content providers can avoid the need to invest in and maintain high-capacity origin servers, which can be both costly and resource-intensive (Stocker et al., 2017).

Furthermore, CDNs can enhance the security of content delivery by providing protection against certain types of attacks, such as Distributed Denial of Service attacks (Li et al., 2020). By absorbing and mitigating these attacks, CDNs can help ensure the continuity of content delivery, even in the face of malicious activity. In conclusion, Content Delivery Networks offer a wide range of benefits that have made them an essential component of modern web infrastructure. These benefits include improved content availability, enhanced delivery performance, optimized content delivery, cost savings, and increased security, all of which have contributed to the widespread adoption of CDNs by content providers and web applications (Ngaffo et al., 2020).

## METHODS

The field of Information Systems has witnessed a shift from a predominant focus on technological aspects to an increasing interest in managerial and organizational issues, leading to a growing emphasis on qualitative research methods (A Jabar et al., 2009). Commonly utilized approaches include action research, case study, ethnography, and grounded theory (A Jabar et al., 2009).

Case study research, in particular, has emerged as the most widely employed qualitative method in Information Systems research, as it is well-suited for understanding the complex interactions between information technology-related innovations and organizational contexts (Darke et al., 1998). The case research strategy offers a valuable approach for investigating emerging socio-technical phenomena in Information Technology (Al-Adwan, 2017). By combining the rigorous theoretical sampling and coding mechanisms of grounded theory with the interpretive nature of case study research, researchers can generate relevant and rigorous theories that facilitate a deeper understanding of these phenomena (Adwan, 2017).

Qualitative research methods, such as case study research, enable researchers to gain a nuanced understanding of the organizational and social issues associated with the development, implementation, and use of computer-based information systems. However, the analysis of data generated through these methods can be problematic, and researchers must carefully consider the appropriate application of such approaches (Adwan, 2017). To ensure the successful completion of case study research in Information Systems, researchers must strive to balance rigor, relevance, and pragmatism (Darke et al., 1998). By adopting a thoughtful and well-designed approach, scholars can contribute valuable insights into the complex interplay between information technology and organizational contexts (Darke et al., 1998).

This research uses descriptive and narrative methods related to situations and conditions examined by previous studies as well as direct experiences through experimentation. The descriptive method in this research aims to provide a clear and systematic depiction of the phenomenon being studied. This approach allows researchers to gather detailed and comprehensive data about various aspects of the observed situation or condition (Edwards, 2016).

The narrative approach is used to examine the direct experiences of individuals or groups involved in specific situations. Through narratives, researchers can understand the subjective perspectives of participants, offering deep insights into how they experience and interpret the phenomenon being studied. This approach is particularly useful in qualitative research contexts, where the nuances and details of human experiences can be more fully revealed.

The use of descriptive and narrative methods also includes analyzing data from relevant previous studies. By reviewing the literature and findings from prior research, researchers can understand the broader context and compare new findings with existing data. This helps identify common patterns and significant differences, providing a strong theoretical foundation for the current research.

Additionally, direct experiences obtained through experiments and field observations provide crucial empirical data. These experiences allow researchers to test hypotheses and evaluate theories in real-world contexts. Through experimentation, researchers can control certain variables and observe their effects directly, adding validity to the research findings. Overall, the combination of descriptive and narrative methods enables the research to cover various aspects of the phenomenon being studied. By integrating data analysis from previous research and direct field experiences, this study aims to provide a comprehensive and in-depth overview, contributing significantly to our understanding of the situations and conditions being examined.

## RESULT

The rapid growth of digital content and the increasing demand for seamless user experiences have placed significant emphasis on the performance and efficiency of content delivery networks. CDNs have emerged as a crucial infrastructure to accelerate the delivery of diverse content across the Internet, providing commercial-grade performance for video delivery and web browsing. (Stocker et al., 2017). However, the complexities of modern CDN systems and the evolving landscape of content distribution have presented new challenges in ensuring optimal data transfer speeds. One of the critical aspects of CDN performance is the speed at which application programming interfaces can transfer data. Dynamic web pages, in particular, require servers to generate the response content per-user request before delivering it back to the user, which introduces network traffic, server workload, and results in extra latency. To address this challenge, researchers and industry have focused on developing innovative solutions to accelerate the dynamic content generation and distribution process, with caching playing a vital role. The HTTP range request mechanism, a feature designed to allow clients to request only a part of a web resource, has been a significant consideration in the context of CDN performance (Li et al., 2020). By enabling the efficient retrieval of partial content, this mechanism can potentially improve the overall data transfer speed and reduce the impact of partially failed transfers. Moreover, the heterogeneity, burstiness, and non-stationary nature of real-world content requests pose significant challenges in designing caching algorithms that can automatically adapt to these changing conditions.

### API Code Implementation

Add New Content API (POST) Request:

Endpoint: /api/content

Description: Allows administrators to add new content to the CDN.

Request:

```
{
  "content_url": "https://example.com/images/image1.jpg",
  "origin_server": "https://origin-server.com",
  "cache_duration": 3600 // Cache duration in seconds
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "content_id": "c12345",
  "content_url": "https://example.com/images/image1.jpg",
  "origin_server": "https://origin-server.com",
  "cache_duration": 3600,
  "last_modified": "2023-03-15T08:30:00Z",
  "popularity": 1200 // Number of times accessed
}
```

**Retrieve Content Details API (GET) Request:**

Endpoint: /api/content/c12345

Description: Retrieves details of a specific content item from the CDN.

Request:

```
GET /api/content/c12345
Host: your-cdn-api.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "content_id": "c12345",
  "content_url": "https://example.com/images/image1.jpg",
  "origin_server": "https://origin-server.com",
  "cache_duration": 3600,
  "last_modified": "2023-03-15T08:30:00Z",
  "popularity": 1200 // Number of times accessed
}
```

**Update Content API (PUT) Request:**

Endpoint: /api/content/c12345

Description: Allows administrators to update details of existing content in the CDN.

## Request

```
PUT /api/content/c12345
Host: your-cdn-api.com
Content-Type: application/json
{
  "cache_duration": 7200 // Updated cache duration in seconds
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "message": "Content updated successfully",
  "content_id": "c12345"
}
```

Code API implementation plays a crucial role in the accuracy and efficiency of CDN operations. Accurate implementation of code APIs ensures that CDN servers can effectively cache, serve and manage content requests (Li et al., 2020).

Inaccurate or suboptimal API implementation can lead to various issues, such as inefficient content caching, delayed delivery, and even security vulnerabilities like amplification attacks (Li et al., 2020). To maintain the reliability and performance of CDNs, it is essential to ensure the accuracy of code API implementation at every stage of the CDN ecosystem (Buyya et al., 2008).

This research paper explore the importance of accurate code API implementation in Content Delivery Networks, analyze the potential challenges and implications, and provide recommendations for best practices to enhance the overall effectiveness of CDN operations.

## CONCLUSION

The proliferation of digital content and the increasing demand for seamless user experiences have made Content Delivery Networks (CDNs) a crucial component of modern web infrastructure. CDNs are widely used by popular websites to efficiently deliver content to users worldwide.

The evolution of CDNs has been driven by the need to accelerate the delivery of diverse content and provide commercial-grade performance, particularly for video streaming and web applications. Designing caching algorithms that can automatically adapt to the heterogeneity, burstiness, and non-stationary nature of real-world content requests is a major challenge in CDN optimization.



Code API implementation plays a crucial role in the accuracy and efficiency of CDN operations. Accurate code API implementation ensures that CDN servers can effectively cache, serve, and manage content requests. Inaccurate or suboptimal API implementation can lead to various issues, such as inefficient content caching, delayed delivery, and even security vulnerabilities like amplification attacks. To maintain the reliability and performance of CDNs, it is essential to ensure the accuracy of code API implementation at every stage of the CDN ecosystem.

## References

- 1) A Jabar, M., Sidi, F., Selamat, M. H., Abd Ghani, A. A., & Ibrahim, H. (2009). An Investigation into Methods and Concepts of Qualitative Research in Information System Research. *Computer and Information Science*, 2(4), p47. <https://doi.org/10.5539/cis.v2n4p47>
- 2) Adwan, A. S. A. (2017). Case study and grounded theory: A happy marriage? An exemplary application from healthcare informatics adoption research. *International Journal of Electronic Healthcare*, 9(4), 294. <https://doi.org/10.1504/IJEH.2017.085821>
- 3) Buyya, R., Pathan, M., & Vakali, A. (Eds.). (2008). *Content Delivery Networks* (Vol. 9). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-77887-5>
- 4) Cleveland, S. B., Jamthe, A., Padhy, S., Stubbs, J., Packard, M., Looney, J., Terry, S., Cardone, R., Dahan, M., & Jacobs, G. A. (2020). Tapis API Development with Python: Best Practices In Scientific REST API Implementation: Experience implementing a distributed Stream API. *Practice and Experience in Advanced Research Computing*, 181–187. <https://doi.org/10.1145/3311790.3396647>
- 5) Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: Combining rigour, relevance and pragmatism. *Information Systems Journal*, 8(4), 273–289. <https://doi.org/10.1046/j.1365-2575.1998.00040.x>
- 6) Douds, C. F. (1971). The state of the art in the study of technology transfer—A brief survey. *R&D Management*, 1(3), 125–131. <https://doi.org/10.1111/j.1467-9310.1971.tb00070.x>
- 7) Edwards, S. L. (2016). Narrative analysis: How students learn from stories of practice. *Nurse Researcher*, 23(3), 18–25. <https://doi.org/10.7748/nr.23.3.18.s5>
- 8) Li, W., Shen, K., Guo, R., Liu, B., Zhang, J., Duan, H., Hao, S., Chen, X., & Wang, Y. (2020). CDN Backfired: Amplification Attacks Based on HTTP Range Requests. *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 14–25. <https://doi.org/10.1109/DSN48063.2020.00022>
- 9) Lindman, J., Horkoff, J., Hammouda, I., & Knauss, E. (2020). Emerging Perspectives of Application Programming Interface Strategy: A Framework to Respond to Business Concerns. *IEEE Software*, 37(2), 52–59. <https://doi.org/10.1109/MS.2018.2875964>
- 10) Ngaffo, A. N., El Ayeb, W., & Choukair, Z. (2020). Information-Centric Networking Challenges and Opportunities in Service Discovery: A Survey. *2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*, 1–8. <https://doi.org/10.1109/ComNet47917.2020.9306088>
- 11) Patni, S. (2017). *Pro RESTful APIs*. Apress. <https://doi.org/10.1007/978-1-4842-2665-0>
- 12) Patni, S. (2023). Fundamentals of RESTful APIs. In S. Patni, *Pro RESTful APIs with Micronaut* (pp. 1–15). Apress. [https://doi.org/10.1007/978-1-4842-9200-6\\_1](https://doi.org/10.1007/978-1-4842-9200-6_1)

- 13) Pavithra, G. M., Manoranjith, & Nagalakshmi, S. (2021). An Overview Of Technology Transfer As A Regulatory Aspect. *International Journal of Applied Pharmaceutics*, 14–19. <https://doi.org/10.22159/ijap.2021v13i2.40067>
- 14) Stocker, V., Smaragdakis, G., Lehr, W., & Bauer, S. (2017). The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem. *Telecommunications Policy*, 41(10), 1003–1016. <https://doi.org/10.1016/j.telpol.2017.02.004>
- 15) Sturm, R., Pollard, C., & Craig, J. (2017). Application Programming Interfaces and Connected Systems. In *Application Performance Management (APM) in the Digital Enterprise* (pp. 137–150). Elsevier. <https://doi.org/10.1016/B978-0-12-804018-8.00011-5>
- 16) Walker, A., & Ellis, H. (2000). Technology Transfer: Strategy, Management, Process And Inhibiting Factors. A Study Relating To The Technology Transfer Of Intelligent Systems. *International Journal of Innovation Management*, 04(01), 97–122. <https://doi.org/10.1142/S1363919600000068>
- 17) Wulf, J., & Blohm, I. (2020). Fostering Value Creation with Digital Platforms: A Unified Theory of the Application Programming Interface Design. *Journal of Management Information Systems*, 37(1), 251–281. <https://doi.org/10.1080/07421222.2019.1705514>