NGAPI and NGMonitor: Bridging NovaGenesis and the Current Internet

William S. Mamede, Antônio M. Alberti, José M. C. Brito
Instituto Nacional de Telecomunicações
Santa Rita do Sapucaí - MG, Brazil
e-mails: william.silva@mtel.inatel.br, antonioalberti@gmail.com, brito@inatel.br

***Abstract*—As novel Internet structures have been proposed over the past decade, the task of monitoring data transmissions and providing interoperability among different architectures has become increasingly complex. Consequently, the development of backward compatible solutions which also facilitate network traffic analysis has become a pivotal element for enabling the visualization and interconnection of Future Internet (FI) data within the current WEB infrastructure. Moreover, Artificial Intelligence (AI) has emerged as an invaluable tool for enhancing data analysis and visualization, especially within complex environments. This paper introduces an innovative data visualization tool, specifically engineered to oversee data flows within the NovaGenesis (NG) architecture, by addressing the critical challenge of bridging the gap between the prospective capabilities of the FI and the existing Internet, which also enables the incorporation of generative AI to assess network-related content.**

Keywords-NovaGenesis, Future Internet, ICN, ID-based Networking, Browser, Interoperability, Overlay Networks, Artificial Intelligence, Data Analysis using AI.

I. Introduction

The Internet, which is a fundamental pillar of contemporary society, is facing an unprecedented increase in the number of interconnected devices and data exchange. This exponential expansion, in addition to the user requirements and the emergence of novel technologies, has become a substantial challenge on the existing Internet architecture, developed on the host-centric Transmission Control Protocol/Internet Protocol (TCP/IP) paradigm. This architecture, developed several decades ago [1], contains significant limitations in effectively addressing modern challenges related to scalability, security, mobility, and flexibility [2], which presents an impediment to address the demands of modern applications and other Internet solutions.

In response to these constraints, the academic community is conducting studies aiming to demonstrate that the advances from Future Internet Architectures (FIAs) [3] provide a more resilient, secure and user-oriented digital environment. FIAs adopt content-centric paradigms, prioritizing the information itself over host addresses, and integrate cutting-edge technologies that exploit the capabilities of data-driven networking. However, the possibility of running existing applications in different Internet environments became a challenge, which, according to Zali *et al.* [4], could be mitigated by developing a backward compatible architecture, aiming to promote adherence to industry standards and protocols, ensuring compatibility and seamless communication between WEB applications. Such a solution could create a cross-platform infrastructure composed of multiple FIAs, and enhance content delivery, since the existence of an environment which allows interoperability among different networks will enable simultaneous use of various applications and protocols, resulting in smooth progression of the network. Also, as stated by Siddiqui and Mueller [5], this type of solution could expand the user base and services, as it would offer a flexible and adjustable approach to controlling network resources. Therefore, we concluded that interoperability provides the capability of having applications created with different architectures to be executed and distributed without relying on the Internet's structure.

NovaGenesis (NG) [6] represents a pioneering FIA, designed with the goal of transforming the Internet through the synthesis of principles of Information-Centric Networking (ICN), Software-Defined Networking (SDN), and Service-Oriented Design. Its architecture aims to provide a solution that provides features such as self-certifying identifiers, a publish/subscribe communication paradigm, and a distributed hash table mechanism, thereby enabling efficient content dissemination, service discovery, and autonomous network management.

With the goal of providing a tool to facilitate monitoring and analysis in a NG environment, we have designed and implemented NGMonitor, an innovative data visualization instrument that integrates the NG architecture with existing WEB technologies by furnishing a user-centric WEB-based dashboard for real-time visualization of network activity, and NGAPI, a module that provides a set of RESTful APIs that expose NG data and enables its interoperation with the current Internet. As a result, it facilitates the integration of pre-existing WEB solutions, including generative AI. Considering that, we developed a solution to consume Google Gemini API [7], aiming to perform network data analysis with the purpose of generating and present hierarchical visualization of network components, such as interrelations between domains, hosts, operating systems, processes, and transferred files, offering a comprehensive overview of network architecture and data exchange.

This paper presents the design, implementation and evaluation of NGAPI and NGMonitor, highlighting the practical applications of the introduced features and demonstrating how researchers and developers of NovaGenesis could benefit from this feature. This article is organized as follows. Section II reviews the related work as well as the methodologies adopted by other FIAs. Section III offers a summary of the architecture of the system that was developed. Section IV seeks to provide an in-depth overview of the NGAPI module, which facilitates IP communication within NovaGenesis. Section V presents NGMonitor, a WEB-based dashboard that presents valuable

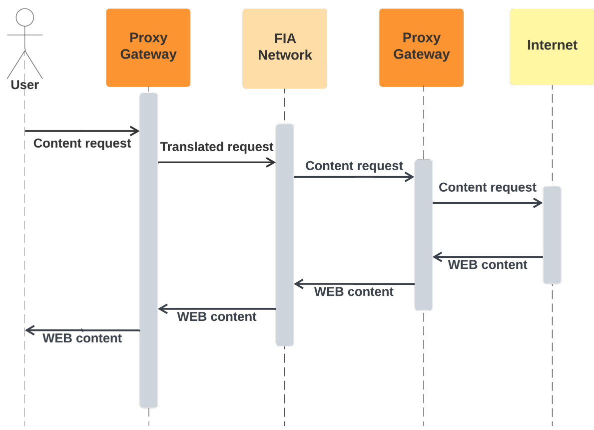

Figure 1. Tunneling approach for WEB interoperability used by FIA.

information from network operations. Section VI provides an analysis of the system's performance results from the experimental evaluation. Finally, Section VII concludes the article and suggests future research directions.

II. Related work

Numerous studies focus on proposing an innovative Internet architecture that serves as an alternative to the current TCP/IP network, originally developed several decades ago, in order to address present-day requirements, which, according to Conti *et al.* [8], include scalability, security, privacy, quality of service, interoperability and flexibility. Various solutions have been proposed to reach the desired goal, such as NovaGenesis [6], XIA [9], RINA [10], NDN [11], and MobilityFirst [12]. Notwithstanding, these approaches are well suited to address the limitations related to the network infrastructure and its design. However, there are other issues to be addressed, including the interoperability with already established WEB systems, which have been designed for the current Internet and often need to run over FIAs.

Our study has identified a potential approach to achieve interoperability, by adding an intermediate proxy between the existing Internet infrastructure and the novel network stack (post-IP) on the access side, as well as on the far-end side between the novel network and an IP network, which is presented in Figure 1. Imagine a legacy application that performs HTTP requests for a server through a TCP/IP socket. Following this tunneling approach, the application request and the server response will remain the same, but the FIA must be enhanced to accept the application request and then deliver the requested data from the responsible server. This could be reached by implementing a new layer for translating HTTP requests to FIA requests (subscriptions or other primitives) and vice versa. Thus, this kind of implementation can allow the network core to evolve and, in the same way, make it backward compatible with existing WEB-based applications. This concept was discussed in [5], where the authors proposed a socket to perform an API translation for the future Internet.

Following the approach presented before, a solution named Content-oriented interoperability framework for current and future Internet architectures (COIN) [13], for example, was implemented providing an interoperability tool, similar to that presented previously, in which a proxy was implemented responsible for translating HTTP requests for NDN [11] at the network entry point and vice versa. Therefore, this solution does not imply any change at the application and server level. A similar solution can also be found in Performance Enhancing Proxy for Deploying Network Architectures (PEP-DNA) [14], in which a proxy was used to translate messages from TCP to RINA [10], as well as reverse translation. The main difference between their approaches is that the proposed proxy has been implemented as part of the Linux operating system kernel [15].

Taking the previous information into account, we developed an intermediate layer (NGAPI) which aims to allow the interoperability of NovaGenesis with the current Internet, going beyond the approaches described before, since it enables the consumption of existing WEB applications deployed in a real-world scenario. Furthermore, our solution includes the addition of a WEB application (NGMonitor), which aims to monitor the traffic in the network, thereby introducing a unique characteristic that distinguishes it from other FIAs. In the subsequent sections, we will detail the architecture of the solution that has been implemented, along with use case examples that aim to demonstrate the practicality and effectiveness of the new feature introduced.

III. System Architecture: Modular and extensible design

NGAPI and NGMonitor were developed following the microservices pattern, to build independent components that, when deployed together, provide a complete end-to-end solution. This modular architecture offers several advantages, including:

- Scalability: The system can be easily scaled by adding or removing modules as needed, ensuring that in the future it is capable of increasing data volumes in complex network environments.
- Flexibility: Each module can be developed and maintained independently, which makes it simple to perform updates and enhancements without impacting other system parts.
- Generality: The microservices design allows for potential adaptation and extension to support other features, including solutions developed for other Internet architectures.

The architecture of the NGMonitor is composed of three main services (as presented in Figure 2):

1) **NG Network Core**: This is the main component of the solution, which contains all the business logic related to NovaGenesis [16], [17]. It operates following the publish/subscribe behavior purpose instead of consuming from the conventional TCP/IP protocol stack. The essential services that make the system operate are the Proxy/Gateway Controller Service (PGCS) and the Name Resolution and Network Caching System (NRNCS), which establish the infrastructure for data exchange. These services enable some of the NG functionalities keys:

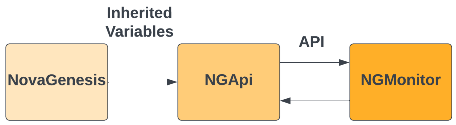

Figure 2. System architecture overview.

- **Name Resolution**: NRNCS offers a mechanism capable of retrieving previously published bindings, ensuring their direct delivery to authorized subscribers.
- **Content Distribution**: The combined operation of PGCS and NRNCS enables efficient publication and subscription of content, ensuring data distribution on the network.
- **Service Discovery**: PGCS is responsible for performing an orchestration in order to discovery of the Publish/Subscribe Service (PSS), Generic Indirection Resolution Service(GIRS), and Hash Table Servic (HTS) during the initialization phase of the system. In addition, it provides a proxy service that allows the representation of other NG services within an operating system.

2) **NGAPI: NovaGenesis Application Programming Interface**: This module aims to provide a set of APIs to enable communication between the NG network and applications developed under the current Internet. This is achieved by implementing a set of endpoints using restCppSdk [18]. Therefore, NGAPI makes it possible to interoperate NovaGenesis with the current Internet, enabling its communication using the HTTP protocol. This interoperability is imperative for NGMonitor, facilitating the consumption and expose of data from external APIs.

3) **NGMonitor: A WEB-Based Dashboard**: This module is an HTTP-based application, developed using the Angular framework [19], aiming to provide a dynamic and intuitive interface to enable visualization of NG data. This module consumes the data provided by NGAPI, processes, and then provides refined and comprehensive data to the end user, including the consumption of Gemini AI [7], which is provided by NGAPI, with the aim of analyzing a set of NG data and generating insightful reports. These reports, presented to the user within the dashboard, provide an understanding of network relationships and data flow patterns, providing insights into the NG network structure, which enhances the analytical capabilities of NGMonitor, empowering researchers and developers to gain deeper knowledge and insights from what the network is executing.

In order to understand how NGMonitor works, it is needed to make clear how the communication happens between the network elements. Firstly, NGMonitor periodically consumes the interfaces provided by NGAPI, obtaining consequently the most updated data related to services that are offered, files that are transmitted and received, which aims to ensure that dashboard presents a real-time overview of the network activities. In addition to that, when information related to the connection between network elements and its processes is requested, it consumes the NGAPI endpoint that enables

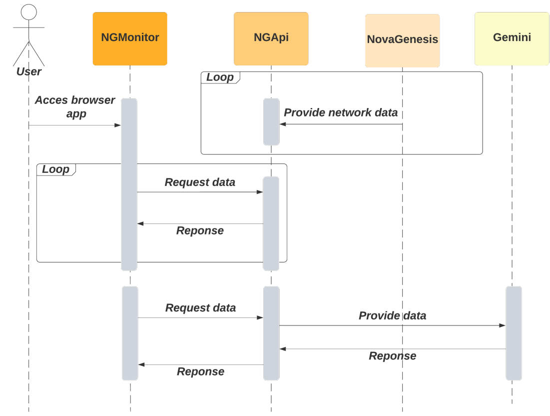

Figure 3. System diagram illustrating the integration between components.

communication with Gemini API, making it possible to perform an evaluation of network data and producing insightful reports for visualization on the dashboard. This synergistic integration of components is presented in Figure 3.

Therefore, NGAPI and NGMonitor emerge as a promising solution that aims to mitigate the lack of an interoperable data visualization and monitoring tool implemented over a FIA. Considering that, we can consider that the proposed solution has the following strengths:

- **Interoperability**
- **Real-Time Insights**
- **Insights Enhanced using AI**
- **User-friendly interface**

In the following sections, we will describe the introduced modules, NGAPI and NGMonitor, by providing a comprehensive analysis of the technologies used at this development, a detailed explanation regarding the implemented functionalities, and provide details about how the modules interact with each other.

IV. NGAPI: Interoperating and Exposing NovaGenesis Data through a RESTful Interface

This section presents how NGAPI was developed, which is the module responsible for providing NovaGenesis data to an application developed under an IP based network, by serving as an intermediary layer between the core of the NG network and the current Internet, enabling seamless communication and interoperability via the HTTP protocol. This enables the integration of a wide number of existing WEB applications with NovaGenesis and introduces the possibility of providing information from FIA usage for other architectures. Thus, the solutions we propose are designed to promote seamless data exchange, thereby enabling the creation of innovative applications that can fully utilize the potentials offered by the advanced architecture of next-generation networks. This is accomplished without necessitating bespoke integration efforts for each specific application intended to be targeted.

1) NGAPI Architecture and Implementation: NGAPI was engineered using the C++ programming language [20] and incorporates the *cpprestsdk* library, which provides a set of tools that enables the development and consumption of HTTP-based endpoints [18]. This choice was guided by the native implementation of NovaGenesis, which is mostly developed in C++, making then it easy to access network variables and data in real time, which avoids the need of creation of additional layers.

2) Exposing Data through RESTful Endpoints: NGAPI, as presented at a high level in Figure 4, provides a set of RESTful endpoints designed to provide NovaGenesis data to external applications. Then, during the operation of the network, it is stored information related to its operation and the content that was transmitted, with the objective of delivering such data as a JSON response from an API request. This feature aims to facilitate the understanding of activities carried out in NovaGenesis. The endpoints provided include the following:

- serviceOffers: This endpoint retrieves and provides information about all services offered in a given NG domain, allowing researchers and developers to understand the services that are exposed and explore potential interactions or collaborations.
- transmittedMessages: This endpoint returns a list of all transmitted data files, providing details such as file names, timestamps, and the source application that sent the data. This endpoint enables monitoring of data flow patterns.
- receivedMessages: Similarly to the ‘transmittedMessages‘ endpoint, this endpoint retrieves information about data files received by hosts, providing insight into the flow of data from different sources.
- transmittedImage: This endpoint allows retrieval of specific transferred images.
- getDataAnalysedByAI: This endpoint is responsible for collecting NG bindings [17] and offers these resources for analysis by Gemini AI. Bindings are key-value pairs elements stored in the HTS in a categorized way, which describe the relationships between several elements, including connections between physical and virtual entities, associations between services and processes, and links between physical devices. This data is sent to Gemini AI, which processes this relational information and generates insights into domains, hosts, operating systems, processes, and transferred files, detailing their complex connections and dependencies. The resulting analytical insights are then used by NGMonitor to build a hierarchical visualization of the network topology.

3) Interoperating with current Internet: Furthermore, NGAPI has the ability to act as an intermediate layer, allowing data to be exchanged within the network in a format compatible with standard Internet protocols, particularly HTTP. Therefore, in an NG environment, in which a host has access to the current Internet, it can act as a bridge, which allows to:

- Leverage Existing Applications: Make it possible to consume existing WEB applications, provided through the HTTP protocol in NovaGenesis. For instance, a network user might utilize any data supplied by an API (accessed through HTTP), within an NG terminal, without TCP/IP Internet connection, if any host along the network infrastructure has the NGAPI module connected to the Internet. Therefore, this solution is useful in a range of applications, including the Internet of Things (IoT). As detailed in the findings by Alberti *et al.* [16], there is a noticeable improvement in performance associated with Internet of Things (IoT) use cases, particularly when evaluating parameters such as data transfer, memory utilization, and CPU resource consumption. Therefore, the use of NGPAPI provides a significant advantage in terms of backward compatibility, facilitating effective communication among devices that operate within varied network architectures. This interoperability presents a critical benefit, as it allows the smooth integration of current IoT devices and applications into the NovaGenesis network, while maintaining uninterrupted access to previously established Internet-based services.
- Integrate with Cloud Services: Enable the consumption of popular cloud services and platforms in NovaGenesis, extending its capabilities, and providing access to a wider range of tools and resources.
- Share Data Across the Internet: Facilitate communication between hosts connected in different networks.

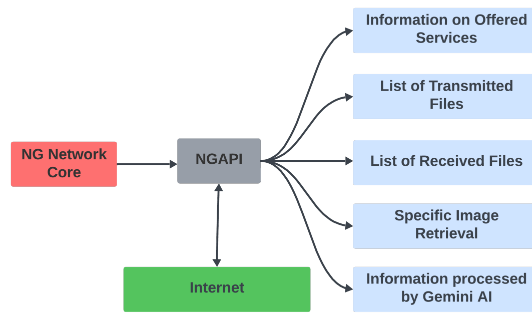

Figure 4. A simplified representation of the NGAPI module.

Figure 5 illustrates the steps taken when attempting to consume content present on the current Internet from a NovaGenesis node, also described as follows:

1) *Content request:* If a user would like to access some data provided via a HTTP API in the current Internet, such as the consumption of information provided by an IoT device or some WEB page, it is possible to consume that by performing a request for desired content through NGAPI, without requiring a direct Internet connection on their device, by providing the path of interest at the end of the request URL (e.g. http://ngapi/wantedContentPath).
2) *Process request:* After receiving the request, the desired path is converted using a hash algorithm and then checked if the information is already available on the network. This hash value represents a unique identifier for the data within the NovaGenesis network. The NGAPI then checks if the hash value exists in the network:
   - *Content Available:* If the resource exists, the NGAPI retrieves the content from the NovaGenesis network and returns it to the user.
   - *Content Not Available:* If the resource does not exist, the

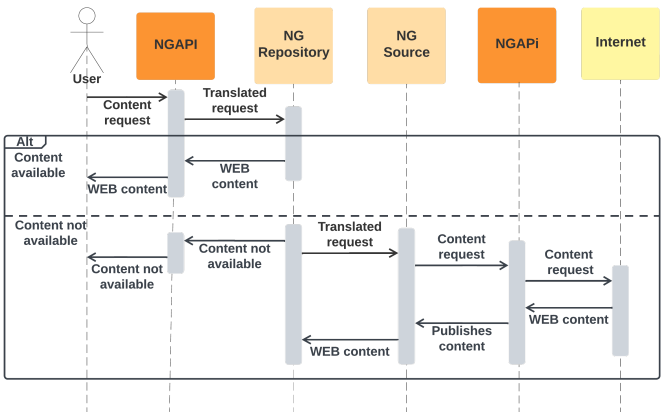

Figure 5. NGAPI sequence diagram.

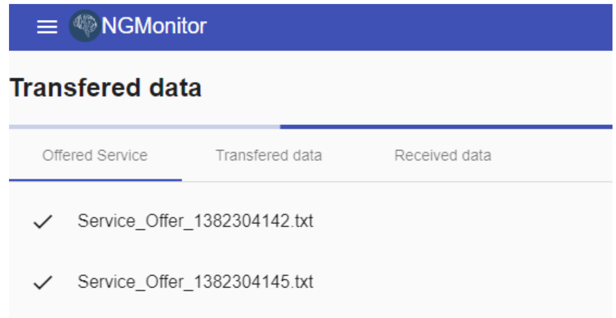

Figure 6. Assigned offered services.

NGAPI proceeds to the next step, initiating a request to the source of the content.

3) *Request to Source:* The node which received the request publishes a content request to the source node in the NG network, containing the desired URL.
4) *Source Fetching:* The source node, upon receiving the request, fetches the content from the Internet through the NGAPI external API.
5) *Source Publishing:* After receiving the content from the Internet, the source publishes the received content on NG networks, making it available to all consumers who seek to access it.

Hence, NGAPI's capability to expose and consume HTTP API facilitates the interaction between existing Internet applications and NovaGenesis, enhancing its accessibility and real-world solutions, allowing users and developers to take the advantages and befits provided by a content-oriented architecture in applications developed under the current Internet. Therefore, this solution elucidates the critical role of interoperability in the advancement of FIAs and enables a smooth evolution of the network, which subsequently allows the coexistence of IP and ICN networks, as outlined by Conti *et al.* [8]. Its ability to seamlessly integrate with existing Internet infrastructure, utilizing the prevailing technologies and tools, ensures a fluid transition toward a more interconnected and versatile digital ecosystem. By providing a comprehensible and accessible interface for external applications to interact with NovaGenesis, NGAPI provides researchers, developers, and network administrators with the tools necessary to fully investigate the potentialities inherent in FIAs.

V. NGMonitor - An Angular Dashboard

This section presents NGMonitor, the WEB-based dashboard developed with the aim of providing a tool that enables visualization and analysis of data extracted from NovaGenesis. This solution was developed using an Angular framework [19], and provides a robust and interactive user interface, allowing researchers and developers to receive insights related to the operation of the network. This solution, in contrast to existing FIAs, presents an innovative tool designed to provide user-friendly insights related to network behavior. Therefore, NovaGenesis network administrators and researchers will benefit from our solution, as it provides a simplified overview of the network, consequently leading to an improved resource management and an improved process of decision making.

1) Capabilities of NGMonitor: Visualization and analytical evaluation: NGMonitor contains a set of functionalities, designed and developed to enable users to perform monitoring and in-depth analysis of NovaGenesis, by providing:

- Real-time visualization: NGMonitor presents dynamic and interactive visualizations of NG data, updated in real time in order to provide an updated representation of the network activity. This enables users to observe and analyze the progression of network events and data transfers as they happen, thereby facilitating the comprehension of network behavior. This feature incorporates the following aspects:
  - Visualization of the Service Offer: The dashboard provides a list of available services, as presented in Figure 6, based on data retrieved from the ‘serviceOffers‘ endpoint. This enables researchers and developers to efficiently identify the set of services available in the network.
  - Data Flow Monitoring: It enables the user to monitor the transmission of files, providing file names, timestamps, and the respective source applications, as demonstrated in Figure 7. This information is obtained by the endpoints ‘transmittedMessages‘ and ‘receivedMessages‘.
  - Image Visualization: NGMonitor provides an image viewer for ‘transmittedImage‘ data. After user interaction with an image entry within the ‘transmittedMessages‘ or ‘receivedMessages‘, the dashboard is able to retrieve and show transmitted images in an emergent pop-up, as presented in Figure 8.
- Hierarchical Visualization powered by AI: It provides integration with Google Gemini AI [7], a state-of-the-art AI model. This integration enables the analysis of NG data and consequently the generation of a hierarchical visualization of network components. This visualization presents the interrelationships between domains, hosts, operating systems (OSIDs), processes (PIDs), and files transferred within the NG network, as presented in Figure 9. This report can

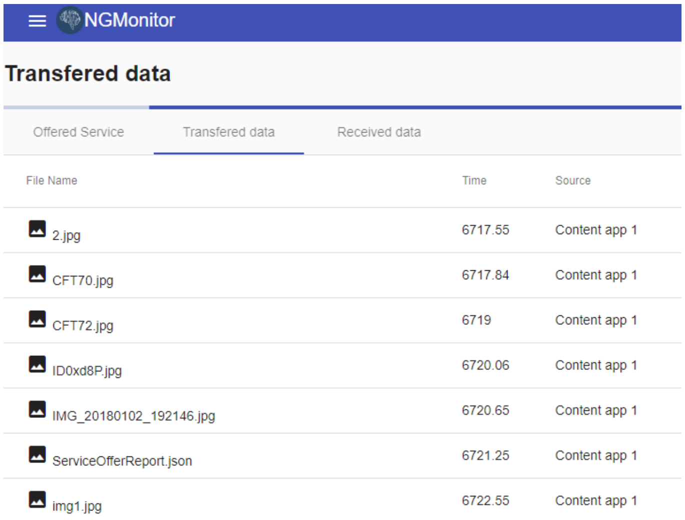

Figure 7. Transferred files from a NG source content distribution application.

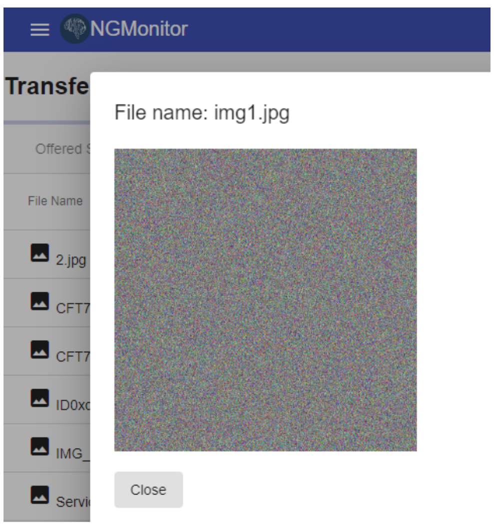

Figure 8. Visualization of a transmitted image.

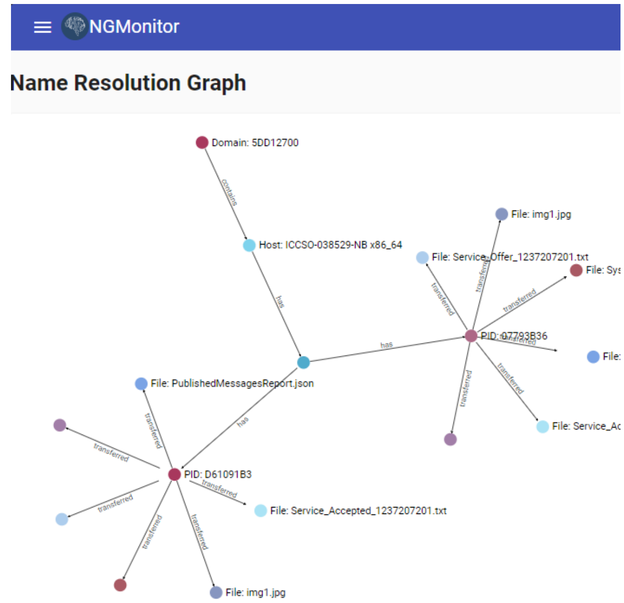

Figure 9. AI-Driven Visualization of Data Flow Patterns in NovaGenesis.

provide comprehensive information related to the network's architecture, enabling users to identify potential bottlenecks or issues.

- User-friendly: NGMonitor boasts an intuitive user interface that prioritizes the user experience. It has been constructed using the Angular Material library [21], a decision that enhances its visual appeal and interactive features, making it more engaging and accessible for users.

VI. NGAPI Performance Metrics

This section presents an analysis of the NGAPI performance, which constitutes the cornerstone of the introduced monitoring enhancement, allowing seamless communication between NovaGenesis, NGMonitor and the current Internet. In order to analyze the responsiveness of the API, we perform extensive tests using JMeter [22], which is a widely recognized open source tool for load testing and performance evaluation. To do so, we develop a test case that tries to simulate a scenario in which multiple users generate a high number of requests in a certain time slot. This scenario is designed in order to stress the implemented APIs and the interoperability feature, then producing critical insights related to system performance.

To perform these analyses, we define the following essential metrics: Mean response time, which provides the typical duration required for the API to process a request and subsequently provide the requisite information to the front-end application; Minimum response time, representing the swiftest duration recorded during API interactions; Maximum response time, conversely, this metric indicates the most prolonged duration encountered, thereby reflecting the worst-case scenario.

The results of our analysis for the internal interfaces, which provides the network information data directly, are encapsulated in the graph presented in Figure 10, which shows the response times distribution for a variety of requests. In this representation, the x-axis denotes individual APIs, while the y-axis quantifies the average, minimum, and maximum response times, measured in milliseconds, corresponding to each API.

Moreover, we perform an experiment designed to assess the interoperability of APIs between NovaGenesis and the existing Internet. In order to achieve this, our experimental setup involves executing multiple requests via an external interface accessible over the current Internet, together with the execution of an identical request using NGAPI, which is aligned with the process illustrated in Figure 5. Therefore, every initial call to NGAPI will initiate the process to have the requested content available on the NG network, which subsequently triggers a request to the original API, followed by the release and availability of the data to the user. Consequently, a delay is anticipated before the content becomes accessible. Observations from our experimental execution indicate that the duration

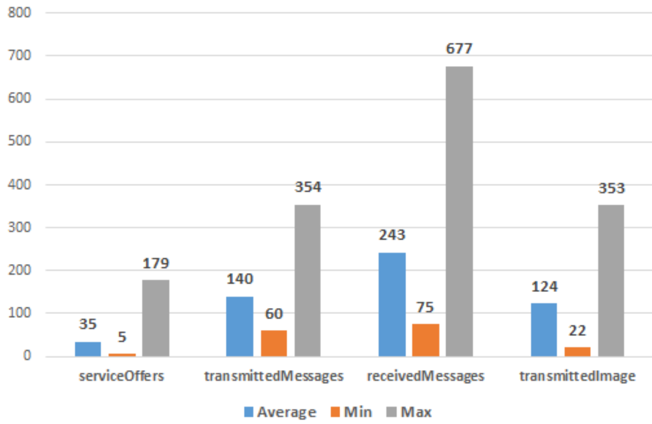

Figure 10. NGAPI Response Time Metrics.

necessary for the content to become accessible initially reached a maximum of 6 seconds. However, subsequent requests for identical content demonstrated markedly reduced response times, attributable to the NovaGenesis routing mechanism. This experiment underscored an important performance aspect: the first retrieval of information, necessitating publication, presents a slower pace compared to later requests, which make use of the content that is already available within the NovaGenesis network. The results presented in Table I show a significant difference in efficiency between accessing data from a standard Internet API and NGAPI, when comparing its response time in milliseconds.

TABLE I
RESPONSE COMPARISON IN MILLISECONDS BETWEEN INTERNET API AND NOVAGENESIS NGAPI

Requested from	Average	Median	Min	Max
Internet	686	645	637	1630
NovaGenesis	10	6	4	464

It is important to mention that the performance evaluation was not executed for interactions with the Google Gemini API. Since this data processing occurs outside NovaGenesis, the response time and performance related to this interface are defined by external factors, such as the availability of computational resources and the prevailing network conditions. Due to that, it was decided to omit this endpoint in the performance tests, since our analysis is based on the evaluation of the efficiency of NGAPI and obtain information within NovaGenesis.

VII. CONCLUSION

This paper introduces a solution composed of NGAPI and NGMonitor, which are pioneering tools designed for interoperation, visualization, and analysis of data within the NovaGenesis future Internet architecture. The proposed method offers an efficient approach to tackle the problem of incorporating FIA advances with existing Internet features and infrastructure, providing an intuitive platform for researchers, developers, and network administrators who seek to systematically monitor and interpret intricate data flows.

To accomplish this objective, a solution was proposed that takes advantage of a synthesis of key fundamental attributes, specifically enumerated as interoperability, real-time visualization, and AI-driven analysis. Therefore, the user benefits from our solution through NGAPI by enabling the communication for TCP/IP designed solutions over NovaGenesis, which means that any existing application could benefit from NovaGenesis strengths, also allowing a smooth evolution of the network, since it became possible to have backward compatible applications. Furthermore, NGMonitor provides a novel feature compared to other FIAs, in which a network administrator could monitor, analyze and receive insights from an external generative AI platform, allowing a proactive identification of network issues during data transfer and the generation of customized reports with the aim of facilitating a decision on network operations and resource management.

In summary, this paper effectively elucidates the potential of interoperability and AI-driven analysis to advance the research, development, and practical applications of FIAs. By providing an intuitive platform endowed with robust visualization and analytical capabilities, the combination of NGAPI and NGMonitor substantiates its significant contribution to the progression of FIA technologies. As the Internet continues to advance and merge with a wider range of devices and services, it became crucial to allow coexistence of IP-based applications with FIAs, therefore, instruments similar to the solution implemented by this study will become indispensable for the surveillance, comprehension, and governance of the intricacies inherent in the future Internet.

ACKNOWLEDGMENT

This work was funded by RNP, with resources from MCTIC, Grant No. 01245.020548/2021-07, under the Brazil 6G project of the *Centro de Referência em Radiocomunicações* (CRR) of the *Instituto Nacional de Telecomunicações* (Inatel), Brazil. The work has also been partially funded by *Fundação de Amparo à Pesquisa do Estado de Minas Gerais* (FAPEMIG RED-00194-23). This work was carried out with support from CNPq, National Council for Scientific and Technological Development - Brazil (303382/2021-0). It was also supported by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) - Finance Code 001. The authors also thank CNPq, and Vivavox Telecom.

REFERENCES

[1] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, “A survey of naming and routing in information-centric networks”, *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–53, 2012. DOI: 10.1109/MCOM.2012.6384450.

[2] G. Xylomenos *et al.*, “A survey of information-centric networking research”, *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014. DOI: 10.1109/SURV.2013.070813.00063.

[3] P. Mueller, “Towards the future internet: A review”, in *From Multimedia Communications to the Future Internet: Essays Dedicated to Ralf Steinmetz on the Occasion of His Retirement*, S. Schulte and B. Koldehofe, Eds. Cham: Springer Nature Switzerland, 2024, pp. 74–89, ISBN: 978-3-031-71874-8. DOI: 10.1007/978-3-031-71874-8_6.

[4] Z. Zali, E. Aslanian, M. H. Manshaei, M. R. Hashemi, and T. Turletti, “Peer-assisted information-centric network (picn): A backward compatible solution”, *IEEE Access*, vol. 5, pp. 25 005–25 020, 2017. DOI: 10.1109/ACCESS.2017.2762697.

[5] A. A. Siddiqui and P. Mueller, “A requirement-based socket api for a transition to future internet architectures”, in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2012, pp. 340–345. DOI: 10.1109/IMIS.2012.117.

[6] A. M. Alberti and D. Singh, “Developing a novagenesis architecture model for service oriented future internet and iot: An advanced transportation system scenario”, in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 359–364. DOI: 10.1109/WF-IoT.2014.6803188.

[7] *Gemini API Docs and Reference*, Online. Available: https://ai.google.dev/gemini-api/docs [retrieved September 2024].

[8] M. Conti, A. Gangwal, M. Hassan, C. Lal, and E. Losiouk, “The road ahead for networking: A survey on icn-ip coexistence solutions”, *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2104–2129, 2020. DOI: 10.1109/COMST.2020.2994526.

[9] D. Naylor *et al.*, “Xia: Architecting a more trustworthy and evolvable internet”, *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 50–57, Jul. 2014, ISSN: 0146-4833. DOI: 10.1145/2656877.2656885.

[10] J. D. Day, *Patterns in network architecture: A return to fundamentals*. Prentice Hall, 2010.

[11] L. Zhang *et al.*, “Named data networking”, *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[12] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, “Mobilityfirst: A robust and trustworthy mobility-centric architecture for the future internet”, *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, pp. 2–13, Dec. 2012. DOI: 10.1145/2412096.2412098.

[13] M. Jahanian, J. Chen, and K. K. Ramakrishnan, “Managing the evolution to future internet architectures and seamless interoperation”, in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1–11. DOI: 10.1109/ICCCN49398.2020.9209599.

[14] K. Ciko, M. Welzl, and P. Teymoori, “Pep-dna: A performance enhancing proxy for deploying network architectures”, in *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, 2021, pp. 1–6. DOI: 10.1109/ICNP52444.2021.9651953.

[15] *What is the Linux kernel?*, Online. Available: https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel [retrieved September 2024].

[16] A. M. Alberti *et al.*, “Advancing novagenesis architecture towards future internet of things”, *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 215–229, 2019. DOI: 10.1109/JIOT.2017.2723953.

[17] A. M. Alberti, M. A. F. Casaroli, D. Singh, and R. da Rosa Righi, “Naming and name resolution in the future internet: Introducing the novagenesis approach”, *Future Generation Computer Systems*, vol. 67, pp. 163–179, 2017, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2016.07.015.

[18] Microsoft Corporation, *C++ rest sdk*, Online. Available: https://microsoft.github.io/cpprestsdk/index.html [retrieved September 2024].

[19] N. Jain, A. Bhansali, and D. Mehta, “Angularjs: A modern mvc framework in javascript”, *Journal of Global Research in Computer Science*, vol. 5, no. 12, pp. 17–23, 2014.

[20] B. Stroustrup, *The C++ Programming Language*, 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2013, ISBN: 978-0-321-56384-2.

[21] V. K. Kotaru, “Introduction to angular material”, in *Material Design implementation with AngularJS: UI Component Framework*. Berkeley, CA: Apress, 2016, pp. 1–5, ISBN: 978-1-4842-2190-7. DOI: 10.1007/978-1-4842-2190-7_1.

[22] *Apache jmeter*, Online. Available: https://jmeter.apache.org/ [retrieved October 2024].