Sciforce

Journal of Artificial Intelligence and Machine Learning

Journal homepage: www.sciforce.org
ISSN : 2995-2336

Open Access Research Article

Decentralized Trust: Consensus Mechanisms and Blockchain Integration In Distributed Systems

Divya Soundarapandian

Software Engineering Manager, The Home Depot., United States

Abstract

This research investigates the application of the Grey Relational Analysis (GRA) method to evaluate and rank various decentralized computing alternatives, including Decentralized Systems, Networked Systems, Distributed Computing Architectures, Multi-node Systems, and Federated Systems. These systems play a vital role in modern digital infrastructure by enhancing performance, reducing bottlenecks, and improving security. To determine the most suitable alternative, the study considers key evaluation parameters such as availability, response time, cost, speed, and storage capacity. The GRA methodology provides a robust multi-criteria decision-making framework, enabling objective comparison and selection among the alternatives. Research Significance: With the exponential growth in data generation and processing needs, centralized systems face challenges in scalability, reliability, and latency. Decentralized and distributed architectures offer viable solutions, supporting resilient and efficient computing environments. This research is significant as it applies a structured decision-making approach to compare these emerging architectures, helping stakeholders adopt the most suitable system for their needs based on performance and cost-efficiency. Methodology: Grey Relational Analysis (GRA) GRA is a powerful multi-criteria decision-making tool used to handle complex systems with incomplete or uncertain data. In this study, GRA is used to evaluate alternatives against normalized criteria values. The method involves calculating the grey relational coefficients and grades to rank the alternatives. GRA's ability to accommodate varying units and scales of parameters makes it particularly useful in technology evaluations.

Alternatives: Decentralized Systems, Networked Systems, Distributed Computing Architectures, Multi-node Systems, Federated Systems. Each of these architectures is assessed for its ability to support distributed computing environments with robust performance and operational flexibility.

Evaluation Parameters:

Availability: Uptime and reliability of the system, Response Time: Speed of the system, n reacting to user or node requests.

Cost: Overall implementation and maintenance expenses. Speed: Data processing and communication efficiency.

Storage Capacity: Ability to manage and store large volumes of data. Result: Based on the GRA analysis, Distributed Computing Architectures emerged as the most suitable alternative, offering balanced performance across all evaluation parameters. Federated Systems and Multi-node Systems followed closely, particularly excelling in response time and availability. The results highlight the practicality of GRA in systematically evaluating technological choices, helping organizations make data-driven decisions.

Keywords: Decentralized Systems, Grey Relational Analysis, Distributed Computing, System Evaluation, Multi-Criteria Decision Making, Federated Architecture

Introduction

A consensus mechanism was introduced that uses failure detection to identify and flag "detectable" instances of misconduct. This is through generated suspicions. Peer Review creates a secure log for each individual node, capturing the messages it sends and receives, as well as the application's inputs and outputs. Nodes in distributed systems can fail due to various reasons, such as hardware or software failures, or because an attacker has compromised the node. We demonstrate the practicality of Peer Review by implementing it in three distinct types of decentralized systems. [1] NetBet is a software platform that provides a shared environment across both time and space, designed to support research, education, and development in distributed systems and networking. This method very effectively manages the reliability challenges of large-scale distributed systems - often built with volatile commodity hardware - although additional improvements in reliability are still necessary. Having root access on local nodes is very valuable because it allows for experiments involving kernel changes or direct access to raw sockets. Batch processing enables an experimenter to systematically explore a large problem space without the need for manual intervention. A test run on a single node lasts 135 seconds, with most of this time dedicated to rebooting the node and waiting for it to complete the boot process. [2]

Received date: February 15, 2026 **Accepted date:** February 22 2026; **Published date:** March 10, 2026

***Corresponding Author:** Divya Soundarapandian, Software Engineering Manager, The Home Depot., United States; E- mail: divyasoundarapandian90@gmail.com

Copyright: © 2026 Divya Soundarapandian, This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

Developers often build distributed systems based on the message passing paradigm, where processes exchange data by sending messages across the network. Decentralized systems operate within a data center, a facility with a large number of well-connected machines. These data centers can contain tens of thousands of machines, supporting tens of thousands of applications. While this consistency is sufficient for Coda, which only provides weak consistency for file operations, many distributed applications benefit greatly from distributed transactions like those provided by Sinfonia.[3] The increasing complexity of large decentralized systems has led to a growing trend towards automating various management tasks through decentralized components. They also support positive and negative authentication policies, and include a tool to detect any ambiguities or discrepancies. Policy-based management forms the foundation for automating the management of large-scale distributed systems and networks. There are still many challenges to be solved. Generating low-level policies automatically from high-level policies is as complex as building programs directly from requirements specifications. [4] We have found that approaching operating system problems as distributed system problems can lead to innovative solutions to existing challenges or provide valuable new perspectives on established OS methods. In distributed systems, verbose RPCs are reduced by encoding high-level functionality concisely; ultimately, this leads to the transfer of a single RPC or executable code.

They have different consistency and deployment requirements, and distributed systems provide valuable insights into these challenges, especially as systems grow increasingly concurrent and heterogeneous. [5] Although decentralized systems share some behavioral similarities with aggregate systems, we show that, Our research on decentralized systems is algebraic because we establish a relationship between the smooth behaviors of free modules defined in polynomial rings and submodules. In fact, the components within the behavior of a total autonomous system are entire functions formed as linear combinations of exponential functions derived from the zeros of the characteristic polynomial. [6] Semantics are greatly influenced by the concept of physical time within decentralized systems. There are various approaches to address the unique characteristics of decentralized systems. Two distinct evaluation strategies are explored, each reflecting a different method for addressing the unique characteristics of decentralized systems. Differences from decentralized systems are highlighted, along with references to related research efforts. Ode provides robust event operators, although their type and syntax can be confusing, and their practical use is occasionally ambiguous. [7] Using this abstract protocol as a foundation, we examine several replication protocols from both database systems and distributed systems, highlighting their commonalities and differences. This highlights another key difference between distributed systems and database replication protocols. Research has shown that the specification of any problem can be broken down into two main components: security properties and liveness properties. Distributed systems distinguish between deterministic and non-deterministic behavior in replicas. Deterministic behavior implies that replicas will produce identical results when given identical operations. In contrast, decentralized systems typically rely on more strict ordering policies. [8] Nevertheless, the size and complexity of large decentralized systems has led to a shift towards automating a large number of management tasks through the use of decentralized components. In large-scale decentralized organizational environments, it is not possible to define unique policies for each individual target. The increasing size and complexity of large decentralized systems has led to a growing trend towards automating many management tasks through decentralized components. [9] As mentioned earlier, we believe that decentralized systems should balance language-specific simplicity and security with the flexibility of the operating system.

The flexible configuration options mentioned above are especially important for experimental systems. Koenig provides a language-based method for building distributed systems that separates the programming of individual software components from the assembly of these components into complete systems. Distributed systems are designed to use probabilistic, globally unique random bit strings to act as node identifiers. [10,11] Despite this, we strongly believe that future distributed systems must place greater emphasis on security to ensure their continued reliability. However, the primary way a workstation is used - and the style of use that best aligns with the Andrew paradigm - is by a user running directly at the console. [12] Further tests should be conducted to evaluate performance in a WAN environment. In addition, it is necessary to compare the differences between a unified information server and a single information server for the same data. We also want to explore a variety of user access methods. [13] Blockchain technology is a revolutionary system that enables secure and transparent digital transactions without the need for a central authority. At its core, blockchain is a decentralized ledger where data is stored on a network of computers called nodes. Each transaction is recorded in a block, and these blocks are linked together in a chronological chain, making the data immutable and tamper-proof. One of the most important aspects of blockchain is its security. [14] Transactions are verified through consensus mechanisms such as proof of work or proof of stake, which ensure that all participants agree on the validity of the data. This decentralized verification process eliminates the risk of fraud and hacking, which is common in traditional centralized systems. Blockchain has found applications beyond cryptocurrencies such as Bitcoin and Ethereum.[15] It is increasingly used in industries such as supply chain management, healthcare, and voting systems, where transparency and data integrity are paramount. For example, it can track the movement of goods to prevent fraud or securely store patient medical records for easy and secure sharing. Despite its benefits, blockchain faces challenges such as scalability issues, high energy consumption, and regulatory concerns.[16]

Materials and method

Alternatives: Decentralized Systems: Decentralized systems are computing architectures where processing and data storage are distributed across multiple independent nodes rather than being centralized. These systems enhance fault tolerance, scalability, and resource sharing by allowing each node to operate autonomously while coordinating with others. Decentralized systems are widely used in blockchain technology, peer-to-peer networks, and cloud computing infrastructures. Networked Systems: Networked systems refer to a collection of interconnected computers or devices that communicate and coordinate their actions by passing messages over a network. These systems support distributed processing and enable resource sharing across geographic locations. Networked systems are fundamental to modern computing environments such as the internet, corporate intranets, and IoT deployments. Distributed Computing Architectures: Distributed computing architectures describe frameworks in which multiple computers

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

work together to perform computational tasks. These architectures divide workloads into smaller parts processed concurrently on different machines, improving performance and reliability. Examples include grid computing, cluster computing, and cloud platforms that leverage distributed resources. Multi-node Systems: Multi-node systems consist of multiple interconnected processing units or nodes that collaboratively handle data processing and storage. Each node in the system can function independently or in cooperation with others, facilitating high availability and load balancing. Multi-node systems are common in large-scale data centers and scientific computing applications. Federated Systems: Federated systems are composed of autonomous subsystems that collaborate to achieve a common goal while maintaining their independence. This approach allows diverse and geographically dispersed entities to share resources and information securely. Federated systems are often applied in healthcare data exchange, multi-organizational collaborations, and federated learning models.

Evaluation parameter: Availability: Availability refers to the degree to which a system, service, or application remains operational and accessible when required. High availability ensures that users and processes can rely on continuous service with minimal disruptions, which is especially critical in sectors like finance, healthcare, and e-commerce. Downtime not only affects productivity but can also lead to financial losses and damage to reputation. Designing systems with redundancy, failover mechanisms, and robust infrastructure contributes significantly to enhancing availability. Response Time: Response time is the duration a system takes to respond to a user's request or an external input. It is a key performance indicator for user experience, particularly in interactive applications and online services. A system with quick response time allows for smooth and efficient interaction, while delays can frustrate users and reduce operational efficiency. Optimizing response time often involves improving software algorithms, network latency, and server performance. Cost: Cost encompasses all financial aspects related to acquiring, operating, and maintaining a system. This includes initial capital expenditure, licensing fees, energy consumption, maintenance costs, and potential upgrade expenses. It is essential for organizations to evaluate total cost of ownership (TCO) when planning technology investments to ensure that the system delivers value over time without becoming a financial burden. Speed: Speed in a technological context usually refers to the rate at which a system can perform tasks or process data. This includes CPU processing speed, data transfer rates, and execution of operations. High-speed systems are essential for applications requiring real-time analytics, rapid computation, or high-throughput data processing. Enhancing speed often involves upgrading hardware components, optimizing code, and utilizing parallel processing techniques. Storage Capacity: Storage capacity indicates the amount of data a system can store and manage effectively. As data generation increases rapidly across industries, sufficient storage is essential for handling large volumes of structured and unstructured data. Choosing the right type of storage—such as SSDs for speed or cloud storage for scalability—is crucial for meeting both current and future data demands while maintaining accessibility and security.

GRA

The Gray relational analysis (GRA) model has evolved considerably - from models that rely solely on proximity to assess similarity to more advanced frameworks that incorporate both proximity and similarity measures. To ensure a more comprehensive assessment, contemporary GRA models combine absolute and relative measures of Gray relational coefficients. A synthetic GRA model becomes particularly valuable in situations where it is necessary to consider not only the absolute differences in data sequences, but also the relative rate of change from the initial reference point. The synthetic coefficient (u) is calculated by assigning proportional importance to these two related perspectives. A significant improvement in the GRA framework is the extension of the sequence distance estimate from two to three ranks, which improves its discriminative power. Based on this improvement, Tang Yaoguo introduced a model that extends the traditional real number sequence GRA to handle interval number sequences, using the concept of interval distances. This adaptation simplifies the Gray relational analysis of high-dimensional data by moving from categorical integral approaches to multiple integral models, which reflects the concept of a general GRA model. Innovations such as group-wise rotation and group-wise attention modules have further refined GRA. The group-wise rotation component introduces a new mechanism for independently rotating the group of rotation kernels, which significantly reduces the number of parameters compared to state-of-the-art (SOTA) methods. This lightweight and flexible design makes the GRA module highly adaptable for integration into existing convolutional neural networks (CNNs), where it effectively reduces feature degradation and improves model performance. To investigate the effect of architectural choices, an elimination study was conducted on the number of groups used in the GRA module.

The results demonstrated that the group-wise rotation mechanism efficiently captures the dynamics of oriented objects by allowing the kernels to rotate independently, thereby improving the direction sensitivity. In the field of decision making under uncertainty, the SVPNS-MADM method - a multi-attribute decision making (MADM) technique based on GRA within the framework of single-valued Penta partitioning neuromorphic sets (SVPNS) - has been introduced. This method defines the Hamming distance for SVPNS and is based on the neutrosophic extension of GRA. Pentapartitioned neutrosophic sets serve as a robust mathematical framework for managing random, uncertain, and ambiguous information. For example, Mondal and Pramanik used a neutrosophic GRA-based decision model in the SVNS context to support material selection in construction, specifically to select optimal clay bricks. To support such decision models, Mallik and Pramanik further formalized the concept of pentapartitioned neutrosophic sets (PNS). A detailed description and numerical example of the SVPNS-MADM approach are provided in Section 5, while Section 6 outlines the conclusion insights and future research directions. In applied settings, the GRA approach proves to be versatile and practical for handling multi-factor uncertainty. It does not rely on extensive databases or strict classification rules. By determining the weight of each criterion, GRA helps to build effective optimization models for decision-making using intuitive fuzzy data. In addition, GRA has been used to improve used vehicle valuations, where models such as artificial neural networks help to estimate market value based on formal criteria. In industrial contexts such as selecting low-power engines, GRA helps to integrate and analyze standardized indicators to identify better-performing alternatives. The ease of GRA in managing multi-criteria uncertainty has led to its widespread adoption in industries such as electronics, textiles, food, and oil & gas. For example, combined models using fuzzy-

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems.. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

AHP and fuzzy-GRA have been developed to weight criteria and effectively rank alternatives. Future studies could integrate GRA with complex hierarchical structures and explore broader criteria such as social, environmental, and political factors to address complex decision-making challenges. Gray relation analysis (GRA) is a robust method in the field of multi-criteria decision making (MCDM), designed to evaluate and compare multiple alternatives that involve conflicting or interdependent attributes. It is particularly useful in technology evaluations, where it can be used to rank various systems or solutions based on factors such as availability, response time, cost, processing speed, and storage capacity. For example, in selecting a cloud service provider, GRA helps identify the most suitable option by comparing the relative performance of each alternative to a best-of-breed solution. This is done by normalizing each criterion and comparing it to the benchmark, ensuring a well-rounded and equal evaluation. The strength of GRA lies in its ability to work effectively even with incomplete or uncertain data, which is highly applicable to real-world situations. With this approach, decision makers are empowered to assess the trade-offs between key performance indicators and make informed, data-backed decisions.

Results and Discussion

Table 1: Distributed System

	availability	response time	cost	speed	storage capacity
Decentralized Systems	35.0	57.0	35.0	58.0	45.0
Networked Systems	14.0	67.0	75.0	54.0	49.0
Distributed Computing Architectures	25.0	42.0	84.0	57.0	29.0
Multi-node Systems	35.0	53.0	67.0	51.0	61.0
Federated Systems	47.0	64.0	81.0	35.0	85.0

Decentralized systems, networked systems, distributed computing architectures, multi-node systems, and federated systems – across five key attributes (availability, response time, cost, speed, and storage capacity), distinct performance patterns emerge. Federated systems score high on availability (47.0) and storage capacity (85.0), indicating strong reliability and data management capabilities. However, they lag behind on speed (35.0), indicating potential performance tradeoffs. Distributed computing architectures perform exceptionally well on cost-effectiveness (84.0), making them economically attractive while also maintaining competitive speed (57.0). Networked systems exhibit high response times (67.0), beneficial for real-time applications, although they incur higher costs (75.0). Multi-node systems offer a balanced profile across all criteria, with relatively high storage (61.0) and moderate availability and speed. Meanwhile, decentralized systems show stable but lower values overall, offering simplicity but with limited scalability. These insights guide the optimal architecture selection based on specific performance priorities.

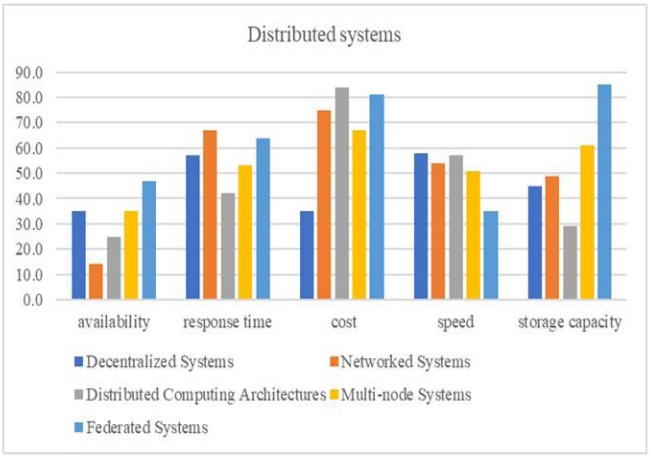

Figure 1: Disributed systems

Figure 1 provides a detailed multidimensional analysis of five system architecture types, as assessed on five key performance metrics: availability, response time, cost, speed, and storage capacity. The data reveals distinct performance profiles for each architecture type. Federated systems show exceptional performance in storage capacity (approximately 85) and cost-effectiveness (approximately 80), while maintaining competitive response time scores (65), making them ideal for large-scale data management scenarios. Networked systems show strong performance in response time (68) and cost-effectiveness (75), although they show moderate availability (15) and speed (55) attributes. Distributed systems show balanced performance on most metrics, with notable strengths in availability (35) and response time (55), but low cost scores (35). Multi-node systems show consistent mid-range performance across all categories, with particularly strong showings in cost (67) and storage capacity (62). Distributed computing architectures present a mixed profile, excelling in cost-effectiveness (85) but showing limitations in availability (25) and speed (55). The analysis indicates that no single architecture dominates

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems.. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

all performance dimensions, indicating that architecture selection should be driven by specific organizational priorities and use case requirements rather than overall superiority.

Table 2: Normalized Data					
	Normalized Data				
	availability	response time	cost	speed	storage capacity
Decentralized Systems	0.6364	0.6000	0.4132	0.0000	0.7143
Networked Systems	0.0000	1.0000	0.8922	0.1739	0.6429
Distributed Computing Architectures	0.3333	0.0000	1.0000	0.0435	1.0000
Multi-node Systems	0.6364	0.4400	0.7964	0.3043	0.4286
Federated Systems	1.0000	0.8800	0.9641	1.0000	0.0000

Federated systems demonstrate the highest performance in terms of availability, response time, cost-effectiveness, and speed, but are inferior in storage capacity. Decentralized systems score well in availability and storage capacity, but rank very low in speed. Networked systems perform well in response time and show moderate cost-effectiveness, although they score very low in availability. Distributed computing architectures are the most cost-effective and have the highest storage capacity, but their performance in availability and response time is weak. Multi-node systems show balanced performance with moderate values in all attributes, with the exception of federated systems that slightly outperform others in speed. This assessment highlights the strengths and limitations of each architecture, allowing decision makers to prioritize systems based on specific operational requirements such as speed, storage, or availability. The data suggests that federated systems have the most versatile overall performance.

Table 3: Deviation sequence					
	availability	response time	cost	speed	storage capacity
Decentralized Systems	0.3636	0.4000	0.5868	1.0000	0.2857
Networked Systems	1.0000	0.0000	0.1078	0.8261	0.3571
Distributed Computing Architectures	0.6667	1.0000	0.0000	0.9565	0.0000
Multi-node Systems	0.3636	0.5600	0.2036	0.6957	0.5714
Federated Systems	0.0000	0.1200	0.0359	0.0000	1.0000

The gray correlation coefficients reveal how closely each system matches the best solution on five performance attributes: availability, response time, cost, speed, and storage capacity. Federated systems demonstrate the highest correlations on availability, speed, and cost, making them the most favorable option overall despite their low score on storage capacity. Networked systems show the best performance on response time and moderate strength on cost and storage, but have low correlations on availability. Distributed computing architectures stand out for their cost-effectiveness and storage capacity, but lag behind on availability and response time. Decentralized systems consistently perform well on availability and storage, although their cost and speed coefficients are low. Multi-node systems exhibit balanced but moderate correlations on all attributes, with no single metric showing dominant strength. These coefficients help explain the trade-offs between systems and guide decision makers toward solutions that best align with specific operational goals.

Table 5: GRG	
	GRG
Decentralized Systems	0.4875
Networked Systems	0.6050
Distributed Computing Architectures	0.6210
Multi-node Systems	0.5052
Federated Systems	0.8061

Gray Relational Grade (GRG) scores provide a composite measure of overall performance for five system architectures, combining multiple criteria into a single score. Of the systems evaluated, federated systems clearly lead with the highest GRG score of 0.8061, indicating better overall performance across factors such as availability, response time, cost, speed, and storage capacity. This suggests that federated systems offer a better balance and meet performance requirements more comprehensively than others. Distributed computing architectures follow closely with a GRG of 0.6210, showing strong performance but falling slightly behind the federated approach. Networked systems come in third with a score of 0.6050, reflecting solid performance but with some trade-offs. Multi-node systems and decentralized systems have relatively low GRG values of 0.5052 and 0.4875, respectively, indicating very moderate overall performance. These scores help decision makers prioritize system choices by highlighting which architectures provide the most consistent and optimal performance for complex multi-criteria requirements.

Table 6: Rank	
Decentralized Systems	5
Networked Systems	3
Distributed Computing Architectures	2
Multi-node Systems	4
Federated Systems	1

The rankings of the five system architectures reflect their overall performance across several evaluation criteria. Federated systems take the top spot (rank 1), indicating excellent performance and

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems.. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

balance across key factors such as availability, response time, cost, speed, and storage capacity. This indicates that federated architectures are the most efficient and reliable choice of options. Distributed system architectures follow closely in second place (rank 2), showing particularly strong performance in cost and storage capacity, although there are some compromises in other areas. Networked systems take third place, reflecting their strengths in response time and cost, but with lower availability. Multi-node systems take fourth place, indicating moderate performance but overall less competitive compared to the leading options. Finally, decentralized systems take last place (rank 5), indicating that while they may excel in some criteria such as availability and storage, their low scores in speed and cost affect their overall suitability. These rankings provide valuable guidance for selecting the optimal system architecture based on specific needs.

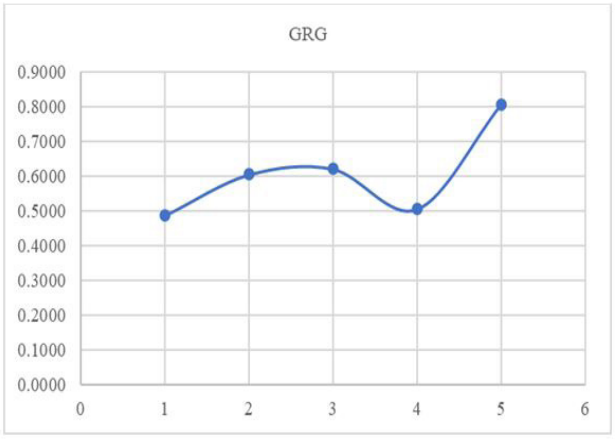

Figure 2: GRG

The data reveal a complex growth pattern characterized by three distinct phases. Initially, GRG demonstrates a steady upward trajectory from approximately 0.48 at time point 1 to 0.62 at time point 2, representing a 29% increase. This growth momentum continues at time point 3, where values plateau at approximately 0.63, indicating a stabilization period. However, the trend reverses in the middle phase, with GRG decreasing to 0.51 at time point 4, representing a 19% decrease from the peak. The most dramatic change occurs in the final phase, where GRG experiences substantial growth, reaching 0.80 at time point 5—the highest value recorded throughout the observation period. This represents a 57% increase from the previous measurement and indicates a potential exponential growth phase. Although the overall pattern GRG shows some fluctuations in the short term, the long-term trend shows significant positive growth, indicating that the final measurement exceeds the initial value by approximately 67%.

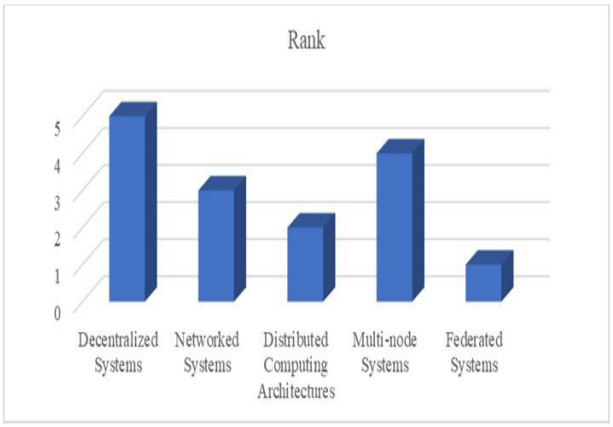

Figure 3: Rank

Decentralized systems emerge as the highest performing architecture with a ranking of approximately 5.5, indicating superior performance characteristics stemming from their autonomous operational capabilities and reduced single points of failure. Multi-node systems come in second with a ranking of 4.5, demonstrating strong performance benefits derived from their distributed processing power and redundancy mechanisms. Networked systems occupy the middle ground with a ranking of 3.5, reflecting a balanced approach that leverages connectivity while maintaining manageable complexity. Distributed computing architectures rank lower at approximately 2.5, suggesting potential challenges related to integration overhead and network latency despite their scalable benefits. Federated systems receive the lowest ranking of 1.5, likely due to their inherent complexity in management, standardization challenges, and potential performance bottlenecks arising from inter-organizational integration requirements. The ranking distribution reveals a clear preference hierarchy, with structures emphasizing autonomy and decentralization, working better than they require extensive coordination and federal management structures.

Conclusion

In conclusion, blockchain technology represents a significant advancement in the way data and transactions are managed, providing unparalleled security, transparency, and decentralization. While it originated as the backbone for cryptocurrencies, its potential has now expanded to a variety of sectors, including finance, healthcare, supply chain, and governance. Despite challenges such as scalability and regulatory hurdles, the continued innovation and growing adoption of blockchain represent a transformative future for digital trust and efficiency. As the technology matures, it is poised to reshape industries by enabling more secure, transparent, and trusted systems that empower users worldwide.

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems.. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

References

1. Wang, Jiangshan, Yifan Pu, Yizeng Han, Jiayi Guo, Yiru Wang, Xiu Li, and Gao Huang. "Gra: Detecting oriented objects through group-wise rotating and attention." In European Conference on Computer Vision, pp. 298-315. Springer, Cham, 2025.
2. Das, Suman, Bimal Shil, and Surapati Pramanik. "SVPNS-MADM strategy based on GRA in SVPNS Environment." Neutrosophic Sets and Systems 47, no. 1 (2021): 50-65.
3. Lenzen, Manfred, Richard Wood, and Blanca Gallego. "Some comments on the GRAS method." Economic systems research 19, no. 4 (2007): 461-465.
4. Lu, Haonan, Yongman Zhao, Xue Zhou, and Zikai Wei. "Selection of agricultural machinery based on improved CRITIC-entropy weight and GRA-TOPSIS method." Processes 10, no. 2 (2022): 266.
5. Alvarez□Lajonchere, L., N. H. Shoukry, B. Gra, Y. Amador□ Cañizares, F. Helle, N. Bedard, I. Guerra et al. "Immunogenicity of CIGB□230, a therapeutic DNA vaccine preparation, in HCV□chronically infected individuals in a Phase I clinical trial." Journal of viral hepatitis 16, no. 3 (2009): 156-167.
6. Gundala, Tirumala Rao. (2025). Predictive Analytics for SSO Performance: Improving Authentication Response Times in Oracle Enterprise Environments Using Linear Regression, Random Forest Regression. International Journal of Computer Science and Data Engineering, 2(4), 1–6. doi: https://dx.doi.org/10.55124/csdb.v2i4.265
7. Liu, Enci, Jie Li, Anni Zheng, Haoran Liu, and Tao Jiang. "Research on the prediction model of the used car price in view of the pso-gra-bp neural network." Sustainability 14, no. 15 (2022): 8993.
8. Solomon, Alan, Tina Richey, Charles L. Murphy, Deborah T. Weiss, Jonathan S. Wall, Gunilla T. Westermark, and Per Westermark. "Amyloidogenic potential of foie gras." Proceedings of the National Academy of Sciences 104, no. 26 (2007): 10998-11001.
9. Kuncan, Melih. "An intelligent approach for bearing fault diagnosis: combination of 1D-LBP and GRA." Ieee Access 8 (2020): 137517-137529.
10. Illing, Patricia T., Julian P. Vivian, Nadine L. Dudek, Lyudmila Kostenko, Zhenjun Chen, Mandvi Bharadwaj, John J. Miles et al. "Immune self-reactivity triggered by drug-modified HLA-peptide repertoire." Nature 486, no. 7404 (2012): 554-558.
11. Perikala. K (2025). Architecting MCP-Based Platforms for Enterprise-Scale Agentic Generative AI. Journal of Business Intelligence and Data Analytics, 2(3), 1-8. https://doi.org/10.55124/jbid.v2i3.264
12. Tse, Maggie, Haocun Yu, Nutsinee Kijbunchoo, A. Fernandez-Galiana, P. Dupej, L. Barsotti, C. D. Blair et al. "Quantum-enhanced advanced LIGO detectors in the era of gravitational-wave astronomy." Physical Review Letters 123, no. 23 (2019): 231107.
13. Nguyen, Phi-Hung, Jung-Fa Tsai, Venkata Ajay KUMAR G, and Yi-Chung Hu. "Stock investment of agriculture companies in the Vietnam stock exchange market: An AHP integrated with GRA-TOPSIS-MOORA approaches." The Journal of Asian Finance, Economics and Business 7, no. 7 (2020): 113-121.
14. Lumpkin, G. Thomas, Todd W. Moss, David M. Gras, Shoko Kato, and Alejandro S. Amezcua. "Entrepreneurial processes in social contexts: how are they different, if at all?." Small Business Economics 40 (2013): 761-783.
15. Mostafa, Ahmed M. "An MCDM approach for cloud computing service selection based on best-only method." IEEE Access 9 (2021): 155072-155086.
16. Haeberlen, Andreas, Petr Kouznetsov, and Peter Druschel. "PeerReview: Practical accountability for distributed systems." ACM SIGOPS operating systems review 41, no. 6 (2007): 175-188.
17. Naidu Paila. (2024). An Integrated AI-Driven Framework for Automating the Procure to Pay Process in SAP Ariba and Optimizing Human Resources. International Journal of Computer Engineering and Technology (IJCET), 15(3), 273–283
18. Rajendra Kattunga, Bilquis Ali and Prabha Patil (2025). Evaluation of Financial Performance Indicators Using Ensemble Machine Learning Techniques: A Case Study of a Diversified Industrial Enterprise. Journal of Business Intelligence and Data Analytics, 2(4), 1-8 https://doi.org/10.55124/jbid.v2i4.266
19. White, Brian, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, and Abhijeet Joglekar. "An integrated experimental environment for distributed systems and networks." ACM SIGOPS Operating Systems Review 36, no. SI (2002): 255-270.
20. Aguilera, Marcos K., Arif Merchant, Mehul Shah, Alistair Veitch, and Christos Karamanolis. "Sinfonia: a new paradigm for building scalable distributed systems." ACM SIGOPS Operating Systems Review 41, no. 6 (2007): 159-174.
21. Rajendra Kattunga, Bilquis Ali and Prabha Patil (2024). Data-Driven Performance Optimization in Healthcare Distribution Operations Using Ensemble and Kernel-Based Regression Models. International Journal of Artificial intelligence and Machine Learning, 2(3), 1-8. https://doi. org/10.55124/jaim.v2i3.296
22. Sloman, Morris. "Policy driven management for distributed systems." Journal of network and Systems Management 2 (1994): 333-360.
23. Baumann, Andrew, Simon Peter, Adrian Schüpbach, Akhilesh Singhania, Timothy Roscoe, Paul Barham, and Rebecca Isaacs. "Your computer is already a distributed system. Why isn't your OS?." In HotOS, vol. 9, pp. 12-12. 2009.
24. Krishnamoorthy Selvaraj, Satya Sukumar Makkapati, SeetaramRayarao, Surya Rao Rayarao and Dr. Suryakiran Navath, Ph.D. A Data Science Model for a study of Stars Supplemental Provider Rating in the Healthcare Domain Journal of Business Intelligence and Data Analytics 2024, 1, 12-16
25. Pillai, Harish K., and Shiva Shankar. "A behavioral approach to control of distributed systems." SIAM Journal on Control and Optimization 37, no. 2 (1999): 388-408.

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems.. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299

26. Schwiderski, Scarlet. Monitoring the behaviour of distributed systems. No. UCAM-CL-TR-400. University of Cambridge, Computer Laboratory, 1996.

27. Wiesmann, Matthias, Fernando Pedone, André Schiper, Bettina Kemme, and Gustavo Alonso. "Understanding replication in databases and distributed systems." In Proceedings 20th IEEE International Conference on Distributed Computing Systems, pp. 464-474. IEEE, 2000.

28. Moffett, Jonathan D., and Morris S. Sloman. "Policy hierarchies for distributed systems management." IEEE Journal on selected areas in communications 11, no. 9 (1993): 1404-1414.

29. Magee, Jeff, Jeff Kramer, and Morris Sloman. "Constructing distributed systems in Conic." IEEE transactions on software engineering 15, no. 6 (1989): 663-675.

30. Tarkoma, Sasu, Christian Esteve Rothenberg, and Eemil Lagerspetz. "Theory and practice of bloom filters for distributed systems." IEEE Communications Surveys & Tutorials 14, no. 1 (2011): 131-155.

31. Satyanarayanan, Mahadev. "Integrating security in a large distributed system." ACM Transactions on Computer Systems (TOCS) 7, no. 3 (1989): 247-280.

32. Krishnamoorthy Selvaraj, Satya Sukumar Makkapati, Surya Rao Rayarao and Dr. Suryakiran Navath, Ph.D. Computational Molecular Biology in Data Science Applications on Bioinformatics in Genetic Sequencing of COVID Mutations, Journal of Data Science and Information Technology 2024, 1, 5-9

33. Zheng, Zibin, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. "Blockchain challenges and opportunities: A survey." International journal of web and grid services 14, no. 4 (2018): 352-375.

Citation: Divya Soundarapandian. (2026) Decentralized Trust: Consensus Mechanisms and Blockchain Integration in Distributed Systems.. International Journal of Artificial intelligence and Machine Learning, 4(1), 1-8. https://doi.org/10.55124/jaim.v4i1.299