Machine Translated by Google

Revista Ibero-Americana Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year II, v.2 2022. | Submission: 12/19/2022 | Accepted: 12/21/2022 | Publication: 12/23/2022

FROM D-7 LATENCY TO REAL-TIME: THE CONVERGENCE OF DATA MESH AND Event-Oriented Architectures (EOA) in Orchestration Scalable Financial Ecosystems

FROM D -7 LATENCY TO REAL -TIME: THE CONVERGENCE OF DATA MESH AND EVENT -DRIVEN ARCHITECTURES (EDA) IN THE ORCHESTRATION OF Scalable Financial Ecosystems

Author: Willian Gouveia de Aguiar

Bachelor's degree in Information Systems from Bandeirante University.

Product Management – PM3

SUMMARY

This article proposes a robust architectural model aimed at large financial corporations facing the challenge of migrating from monolithic data warehouses and batch processes to federated ecosystems based on the Data Mesh paradigm. The central focus of the analysis lies in the drastic reduction of latency in data ingestion and consumption, addressing the transition from seven-day (D-7) processing windows to real-time (Online). Through an analytical approach grounded in software engineering and distributed systems theory, it investigates how the implementation of Event-Driven Architectures (EDA) acts as the necessary connective tissue to enable the decentralization proposed by the Data Mesh. The case study presents the reverse engineering of a real digital transformation, discussing the technical challenges of eventual consistency, computational governance in data meshes, and the strategic centralization of event consumption as a catalyst for the Customer 360º vision.

The research concludes that the integration between autonomous data domains and continuous information flows is the only sustainable path to competitiveness in the modern financial sector.

Keywords: Data Mesh. Event-Driven Architecture. Financial Systems. Data Latency. Data Engineering. Real-Time.

ABSTRACT

This article proposes a robust architectural model for large financial corporations facing the challenge of migrating from monolithic Data Warehouses and batch processes to federated ecosystems based on the Data Mesh paradigm. The central focus of the analysis lies in the drastic reduction of latency in data ingestion and consumption, addressing the transition from seven -day processing windows (D -7) to real -time (Online). Through an analytical approach grounded in software engineering and distributed systems theory, it investigates how the implementation of Event-Driven Architectures (EDA) acts as the necessary connective tissue to enable the decentralization proposed by Data Mesh. The case study presents the reverse engineering of a real digital tr ansformation, discussing the technical challenges of eventual consistency, computational governance in data meshes, and the strategic centralization of event consumption as a catalyst for the Customer 360º vision. The research concludes that the integration between autonomous data domains and continuous information flows is the only sustainable path for competitiveness in the modern financial sector.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Keywords: Data Mesh. Event-Driven Architecture. Financial Systems. Data Latency. Date Engineering. Real-Time.

1. INTRODUCTION

The contemporary financial industry is going through a technological inflection point without Precedents, driven by accelerated digitization and the demand for hyper-personalized user experiences. The traditional data management model, historically based on large monolithic repositories and batch processing, shows clear signs of functional exhaustion. The latency inherent in these systems, which often operate with data updated in windows of days or weeks (D-1, D-7), has become an unacceptable bottleneck in a market where fraud must be detected in milliseconds and credit offerings must be instantaneous. The inability to process and react to business events as they occur represents not only an operational disadvantage but a systemic risk to the financial institution's relevance to the modern consumer.

In this scenario, the emergence of the Data Mesh concept, introduced by Zhamak Dehghani, arises as a structural response to the limitations of centralized Data Lakes, proposing a socio-technical paradigm shift where data is treated as a product and ownership is...
Decentralized for business domains. However, decentralization alone does not solve the latency problem; a transport and integration mechanism that supports the required speed and volume is necessary. This is where Event-Driven Architectures (EDA) come in.
Driven Architecture plays a crucial role, enabling the temporal and spatial decoupling between producers and consumers of information, facilitating the continuous flow of data throughout the organization.

This work seeks to explore the convergence of these two powerful architectural currents.
— Data Mesh and EDA — applying them specifically to the context of financial ecosystems.
Scalable. The justification for this approach lies in the inherent complexity of the data.
banking, which require rigorous transactional consistency (ACID) at their origins, but which
They require flexibility and high availability for analytical and customer engagement purposes.
The transition from a D-7 latency model to real-time processing is not just a...
Not just an infrastructure upgrade, but a complete re-engineering of information value streams.
within the corporation.

Through the analysis of a practical case study, we will detail the engineering challenges.
The challenges faced in reducing the information lifecycle, moving from a reactive and delayed model to a proactive and immediate one, will be discussed. Aspects of data modeling, the choice of streaming technologies, and the trade-offs involved in the CAP theorem (Consistency, Availability and Partition Tolerance in distributed systems. The ultimate goal is to provide a
A theoretical and practical framework for solution architects, data engineers, and technology managers seeking to navigate the complexities of modernizing data platforms in highly critical enterprise environments.

2. The Obssolencive of Analytical Monoliths and the ETL Crisis

The traditional Business Intelligence (BI) and Data Warehousing (DW) model, consolidated in Decades ago, authors like Ralph Kimball and Bill Inmon based their work on the premise that...
The data should be extracted from transactional systems, transformed, and loaded (ETL) into a central repository for further analysis. In financial institutions, this cycle
It frequently followed nightly or weekly operating windows, resulting in latency.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

The information ranged from D-1 to D-7 or more. This architecture, while effective for static regulatory reports and monthly accounting closings, proves inadequate for the dynamics of Open Finance and Instant Payments (such as PIX). The centralized bottleneck of the data engineering team, responsible for maintaining complex ETL pipelines, creates a disconnect between those who understand the data (the business domain) and those who implement it technically, generating technical debt and slowing down the delivery of value.

Maintaining monolithic pipelines in batches to feed a "360º Customer" view in D-7 results in an outdated snapshot of consumer reality. In a financial context, the A customer's credit situation, spending behavior, and security risks change. Minute by minute. Try to personalize an offer or prevent fraud based on data from Last week's processing is ineffective. Furthermore, the computational cost of processing large volumes of... Recurring data processing (total reprocessing or large sliding windows) increases The volume of data increases exponentially, making the financial operation unsustainable.

The rigidity of the schema-on-write data schemas typical of traditional data warehouses also hinders the rapid evolution of data models needed to support new digital products, creating a scenario of "ossification" of the data architecture.

The fragility of batch processes is also evident in their fault tolerance. In a pipeline In a long sequential process that processes weekly data, a failure at an intermediate step can... compromising the entire processing chain, delaying the availability of information for days and requiring complex and risky manual interventions. In contrast to the resilience of In modern microservices, monolithic ETL jobs are single points of failure that affect... across the organization. The dependence on windows and night processing also This conflicts with the global and 24/7 nature of current financial operations, where there is no more "off-peak hours" for performing heavy tasks without impacting performance. transactional systems of origin.

From an infrastructure standpoint, the vertical scalability required by large servers of Analytical databases (MPP - Massively Parallel Processing) have a prohibitive cost and limited elasticity. As data volumes grow to the petabyte scale, the The strategy of "moving the data to processing" becomes unfeasible due to network bandwidth and disk I/O limitations. The architecture needs to evolve towards models where...

Processing occurs close to the data or where the data flows continuously through topologies. distributed processing, eliminating the need for large batch loads that saturate the corporate network and degrade the performance of core banking systems during periods of extraction.

Additionally, data quality suffers in the centralized batch ingestion model. This is because the data engineers responsible for the ETL often lack knowledge of the business rules. Deeply rooted in the source systems, the transformations applied can introduce semantic errors or lose important nuances of the original context. Correcting these errors is slow and bureaucratic, as it involves multiple teams and development cycles. This leads to a phenomenon of distrust in the data in the Data Lake or Data Warehouse, where business users end up creating their own parallel spreadsheets ("Shadow IT") to obtain faster answers, undermining corporate governance and the single source of truth.

Finally, the inability to react to real-time events prevents the implementation of logics of complex business transactions based on triggers. In the D-7 model, the bank only discovers that the The customer had a problem with their credit card days later, when the retention opportunity

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Proactive support is a thing of the past. Batch-based architecture is inherently passive and retrospective, always looking in the rearview mirror, while modern financial product management demands a predictive and prescriptive approach, anticipating customer needs even before they explicitly manifest. Overcoming this model is not just a technological update, but a necessity for strategic survival.

3. DATA MESH: DOMAIN FEDERATION AS A RESPONSE TO COMPLEXITY

The Data Mesh paradigm emerges to address the organizational and technical scalability limitations of centralized models. Based on Eric Evans' Domain-Driven Design (DDD) principles, Data Mesh proposes decentralizing data ownership, distributing it to the teams closest to the source of the information—the domains.

In a financial institution, this means that data for "Cards," "Investments," "Checking Accounts," and "Insurance" are managed, maintained, and served by their respective product teams, and not by a central data area. This change transfers responsibility for data quality and availability to those who possess business knowledge, eliminating the bottleneck of the central engineering team and allowing the organization to scale linearly with increasing complexity.

One of the fundamental pillars of Data Mesh is the concept of "Data as a Product." This implies that domains not only expose raw data, but offer curated, documented, reliable, and easy-to-consume data products. To enable the transition from D-7 to Real-Time, these data products cannot be just tables and static or CSV files; they must be event streams or low-latency APIs. The Data Product must encapsulate the complexity of the transformation and guarantee Service Level Agreements (SLAs) for availability and freshness of information. This requires domain teams to incorporate data engineering competencies, promoting an integrated DevOps and DataOps culture.

Self-serve data infrastructure is another critical component for the success of this architecture. To prevent decentralization from resulting in duplication of effort and technological fragmentation, the organization must provide a common platform that abstracts the complexity of provisioning storage, processing, and orchestration resources. In the context of latency reduction, this platform should offer "as a service" streaming capabilities (such as managed Kafka or Pulsar), Change Data Capture (CDC) tools, and stream processing frameworks (such as Apache Flink or Spark Streaming), allowing domains to publish and consume events in real time without having to reinvent the infrastructure wheel with each new project.

Federated computational governance is the mechanism that maintains cohesion in this distributed ecosystem. Unlike traditional centralized governance, based on control and bureaucracy,

Governance in Data Mesh focuses on interoperability and policy automation. To ensure that data from different domains can be correlated to form a 360º view of the customer, it is necessary to establish global standards for identification, metadata schemas, and access control. The use of data contracts and evolutionary schemas (such as Avro or Protobuf) ensures that changes in the data structure of a producer domain do not break the processes of downstream consumers, allowing for the independent evolution of services.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

The application of Data Mesh in finance faces specific challenges related to security and Regulatory compliance. Decentralization cannot mean loss of control over data. sensitive information (PII) or banking secrecy. The architecture must incorporate security policies such as "Policy as Code," ensuring that access control, encryption, and auditing are... automatically applied across all data products. This allows for agility. The data network provided does not compromise the institutional robustness required by regulatory bodies, transforming compliance from a manual obstacle into an intrinsic quality attribute of the platform.

The cultural shift required by Data Mesh is as profound as the technical one. It requires abandoning the Moving away from a "data silo" mentality and adopting a collaborative and sharing approach. Product teams are now evaluated not only by the functionalities of their applications, but also by the quality and usefulness of the data they provide to the rest of the organization. This shift in incentives is fundamental to enabling real-time architecture, as it motivates producing domains to emit quality events at the source, instead of delegating data cleaning and organization to subsequent ETL processes that add latency and reduce data fidelity.

4. Event-Oriented Architectures (EA) as a Connective Fabric

Event-Driven Architecture (EDA) represents the technological backbone necessary to operationalize the real-time view in a Data Mesh environment. In contrast to... Request-response (REST/HTTP) based architectures, which are synchronous and create Due to strong temporal coupling between services, EDA allows systems to communicate from Asynchronous form through the emission and consumption of immutable facts (events). In the context In financial terms, an event can be "Authorized Transaction," "Customer Updated Address," or... "Credit Proposal Rejected". These events are immediately propagated to a log. distributed, making them available to any interested consumer with latency of milliseconds, definitively breaking with the D-7 cycle.

Using high-performance distributed logs, such as Apache Kafka, allows data to be treated as continuous and infinite streams (unbounded streams), rather than finite and static sets. This enables the Kappa architecture pattern, where the data processing layer serves both real-time analytics and historical reprocessing, simplifying the technology stack by eliminating the need to maintain a complex Lambda architecture (with Separate paths for batch and speed). The persistence of events in the log allows new Data services or products are "plugged into" the architecture and replay the event history. to build their local state, ensuring complete decoupling between producers and new consumers.

To reduce latency from days to real-time, it is imperative to capture state changes. directly at the source. The Change Data Capture (CDC) pattern works by reading the transaction logs from the operational databases (WAL - Write Ahead Log in PostgreSQL, for example) and converting each insertion, update, or deletion into a flow event. This decouples the Generating application logic events, ensuring that any changes to the database are recorded. be captured with minimal latency and without impacting transactional system performance.

This technique is fundamental for migrating legacy systems (Mainframe/Cobol) to the architecture. Modern without the need for deep refactoring of old code.

Complex Event Processing (CEP) allows you to identify patterns, trends, and anomalies in sliding time windows while the data is in

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

movement. Instead of storing the data and then retrieving it (passive approach of the DW), the The system evaluates the data the moment it arrives. This is crucial for fraud detection, where anomalous behavior patterns must be identified and blocked before the transaction is completed. The ability to perform joins between different data streams (e.g., transaction stream + geolocation stream) in real time instantly enriches the context of the information, something impossible in D-7 models.

Consistency in event-driven distributed systems follows the Consistency model. Eventual. According to the CAP Theorem, in a distributed system subject to network partitions, One must choose between availability and strong, immediate consistency. In data architectures In real-time mesh, availability and partition tolerance are generally prioritized, accepting This means there will be a small delay (on the order of milliseconds or seconds) for all nodes in the system to converge to the same state. The design of financial systems from this perspective requires dealing with idempotence (the ability to process the same event multiple times without duplicate side effects) and event ordering, ensuring financial integrity even in failure scenarios.

EDA's scalability is horizontal and massive. Event topics can be partitioned and... distributed among multiple brokers and consumers, allowing for parallel processing of Millions of events per second. This solves the scalability problem of monolithic data warehouses. As the transaction volume grows, simply add more nodes to the streaming cluster and... More instances for consumer groups, while maintaining low and predictable latency.

This elasticity is essential to support peak loads typical of financial retail (such as Black Friday or payroll payment dates), where traditional systems often suffer. performance degradation.

5. CASE STUDY: REVERSE ENGINEERING OF LATENCY (FROM D-7 TO (ONLINE)

Analyzing the migration trajectory of a large financial conglomerate, it was observed that the The initial state (D-7) was characterized by a file transfer matrix (MFT) and SQL scripts scheduled via Control-M. Customer data consolidation involved extracting data from mainframes, card systems, and digital channels, which were then stored in a Staging Area. Data cleansing and conformation processing occurred weekly due to the massive volume of data and the restricted maintenance window. This resulted in inefficient marketing campaigns and outdated risk analyses, where a defaulting customer on Monday would only be identified by the CRM system the following Monday.

The reverse engineering process to achieve real-time began with the identification of critical domain events. Instead of trying to replicate all tables from the legacy database, The focus was on business entities: "Customer", "Account", "Transaction". A [system/method] was implemented. CDC (Change Data Capture) layer connected to the DB2 logs of the Mainframe, publishing Granular changes in Kafka topics. To handle the complexity of consuming these raw events, the "Event Consumption Hub" was created—an architectural artifact within the Client domain responsible for ingesting, deduplicating, and ordering events from multiple disparate sources, building a unified and up-to-date view of the client state in memory or in low-latency (key-value) NoSQL databases.

The reduction in latency has exposed data quality challenges that were previously masked by... Batch processing. Registration inconsistencies, duplicate records, and formatting errors. They became visible instantly. This forced the implementation of schema validation in

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

production time (using Schema Registry and Avro) and the adoption of Dead Letter standards Queues (DLQ) are used to segregate invalid events without interrupting the processing flow. Governance has ceased to be a post-mortem audit process and has become a gatekeeper. active in the ingestion pipeline, ensuring that only compliant data enters the mesh. real-time data.

The challenge of eventual consistency was mitigated through design patterns such as the Saga Pattern and Command Query Responsibility Segregation (CQRS). The read model (used by digital channels and analysts) was separated from the write model, being updated asynchronously by events. For the end user, the perception is real-time: when making a transfer, the balance is updated on the application screen immediately, as the interface consumes data from the read projection that is fed by the event stream in milliseconds. Behind the scenes, robust accounting reconciliation can occur asynchronously, but the user experience and data availability for other systems are immediate.

The network and processing topology was restructured to support high throughput. Centralized processing on large Hadoop clusters was replaced with stream processing microservices (using Kafka Streams or KSQL) distributed at the edges of the domains. This reduced "east-west" network traffic (between internal servers) of large files and prioritized traffic of small, compressed binary messages. The result was a drastic reduction in bandwidth and storage usage, as data no longer needed to be duplicated across multiple layers of staging and temporary tables, existing instead as an immutable log accessible on demand.

The impact on the business was measurable and transformative. The ability to launch offers Contextualized offers (e.g., offering travel insurance at the moment a customer buys an airline ticket with their card) significantly increased product conversion. Credit risk analysis began to consider transactional behavior in the last few minutes, and not just the previous month's history, allowing for the approval of more credit with less risk. The architecture validated the thesis that centralizing the logic of event consumption, combined with decentralizing data production in the Mesh model, is the key to agility at scale.

6. Governance, Security, and the Future of Federated Ecosystems

Migrating to a federated, real-time ecosystem requires a reinvention of information security models. Traditional perimeter control (firewalls) is insufficient in a...
Microservices architecture and data mesh where traffic is internal and intense. The Zero Trust model is adopted, where every interaction between event producers and consumers must be...
Authenticated and authorized. The use of granular Access Control Lists (ACLs) in Kafka topics ensures that only authorized services can read or write to specific data streams, preventing the leakage of sensitive information within the corporate infrastructure itself.

The General Data Protection Law (LGPD) and similar regulations pose challenges. additional elements in immutable log architectures. The "right to be forgotten" seems to contradict the The principle of immutability of events. To solve this, cryptographic techniques are used. Deleted keys (crypto-shredding). Sensitive data within the event is encrypted. with a user-specific key; to "forget" the user, simply delete the key. encryption, making historical data mathematically inaccessible, even if the bytes remain in the log. This demonstrates how the technical architecture must evolve in symbiosis with the legal requirements.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Data cataloging and lineage become complex, yet essential, in distributed environments. It's necessary to trace the flow of information from the original transaction on the mainframe, through event threads, transformations in microservices, to consumption on the executive dashboard or mobile application. Data observability tools and automated catalogs are indispensable for mapping these dynamic dependencies.
Without this, Mesh architecture can degenerate into unmanageable chaos, where no one knows the origin or reliability of a given piece of data.

The evolution of metadata standards is what enables interoperability without coupling. The adoption of common business ontologies and semantic data contracts facilitates mutual understanding between distinct domains. When the "Cards" domain publishes a "Purchase" event, the "Loyalty Points" domain should be able to interpret that.

unequivocally. Governance therefore focuses on defining these controlled vocabularies and ensuring that interface contracts are respected, acting as an "urban planner" for the data city, defining zones and roads, but leaving the construction of buildings (data products) to local architects (domain teams).

The future of these architectures points to a greater convergence between the operational and analytical worlds, a concept sometimes called Translytical or HTAP (Hybrid Transactional/Analytical Processing). Zero latency is not just about speed, but about eliminating the barrier between "doing the business" and "analyzing the business." Artificial intelligence and machine learning models will be deployed directly into the event stream (In-stream Analytics), making autonomous decisions in real time, without human intervention, closing the feedback loop instantly.

Finally, the sustainability of these architectures depends on the continuous training of the teams. The complexity of distributed systems, eventual consistency, and stream processing demands a professional profile with a strong foundation in computer science and information systems. Technology is not magic; it requires rigorous engineering. The transition from D-7 to Real-Time is, ultimately, a journey of technical maturation for the organization, moving from manual and fragile processes to robust and resilient automation, enabled by an architecture that reflects the dynamic and interconnected nature of the modern financial market.

7. CONCLUSION

The research conducted throughout this article demonstrates that the convergence between the Data Mesh paradigm and Event-Driven Architectures (EDA) is not merely a passing technological trend, but rather an imperative structural evolution for financial institutions that aspire to leadership in the digital age. The analysis of the obsolescence of batch processing models and monolithic Data Warehouses revealed that D-7 (or even D-1) latency is incompatible with the expectations of immediacy of the modern consumer and the need for real-time risk management. The inability to process data at the speed of business results in loss of competitiveness, operational inefficiency, and strategic blindness.

The proposed architectural model, validated by the reverse engineering case study, showed that reducing latency to real-time levels requires more than simply adopting streaming tools like Kafka. It demands a restructuring of the organizational topology and data flows, decentralizing responsibility to business domains while centralizing governance of the infrastructure and interoperability standards. The "event consumption centralizer" model proved to be an effective pattern for reconciling the nature of data.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

Distributed data generation necessitates unified and coherent customer views.
(Customer 360º), overcoming the theoretical challenges of the CAP theorem through software design.
intelligent.

It was also observed that eventual consistency, often feared in financial environments,
For conservatives, it's a manageable and necessary trade-off to achieve high availability and
global scalability. Through techniques such as Event Sourcing, CQRS and validation of
Even with rigorous schemes, it is possible to maintain the integrity of financial data while enjoying the benefits.
agility of decoupled systems. Federated computational governance then emerges as the
The fulcrum, ensuring that team autonomy does not result in data anarchy, but rather
in a harmonious ecosystem of interoperable data products.

The transition from a batch process culture to an event-driven culture represents
A profound change in the way the organization "thinks." It stops looking to the past.
(static reports) to react to the present and predict the future (predictive analytics in flow). This
This change enables new business models based on hyper-personalization and proactive detection.
Fraud prevention and contextualized financial services, generating tangible value for the institution.
For both the end customer and the data itself, the data ceases to be a byproduct of the process and becomes an asset.
central to the strategy.

It can therefore be concluded that the architecture discussed here provides the technical basis for the next one.
Generation of banking services. Horizontal scalability, fault resilience, and flexibility.
The evolutionary advancements provided by the union of Mesh and EDA prepare corporations for a future.
where the volume of data will continue to grow exponentially and acceptable latency will tend to
Zero. The technical achievement of reducing the availability of information from seven days to real time.
It is, above all, a business achievement, freeing the organization from the shackles of cycles of
It is streamlining legacy processing and allowing it to operate at the speed of digital life.

With all this in mind, it is recommended that future work explore the integration of data meshes with emerging technologies in Generative Artificial Intelligence and Data Fabric, investigating how semantic automation can further reduce friction in cross-domain integration. The journey of data modernization is ongoing, and mastery of the fundamentals of distributed systems and network topologies, as discussed from an Information Systems perspective, will remain the critical differentiator for professionals who will shape the future of the global financial sector.

REFERENCES

BREWER, Eric. CAP years twelve later: How the "rules" have changed. Computer, vol. 45, no. 2,

pp. 23-29, 2012.

DEHGHANI, Zhamak. Data Mesh: Delivering Data-Driven Value at Scale. Sevastopol:

O'Reilly Media, 2020.

EVANS, Eric. Domain-Driven Design: Tackling Complexity in the Heart of Software.

Boston: Addison-Wesley Professional, 2003.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google

HOHPE, Gregor; WOOLF, Bobby. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Boston: Addison-Wesley Professional, 2003.

INMON, William H. Building the Data Warehouse. 4th ed. Indianapolis: Wiley, 2005.

KIMBALL, Ralph; ROSS, Margy. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Indianapolis: Wiley, 2013.

KLEPPMANN, Martin. Designing Data -Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol: O'Reilly Media, 2017.

Kreps, Jay. I Heart Logs: Event Data, Stream Processing, and Data Integration. Sebastopol: O'Reilly Media, 2014.

NEWMAN, Sam. Building Microservices: Designing Fine-Grained Systems. 2nd ed. Sebastopol: O'Reilly Media, 2021.

STOPFORD, Ben. Designing Event-Driven Systems: Concepts and Patterns for Streaming Services with Apache Kafka. Sebastopol: O'Reilly Media, 2018.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.