International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|

DOI: 10.15662/IJSRAT.2022.0506002

Edge-to-Cloud Data Integration Models for Industrial IoT Applications

Mallesham Goli

Independent Researcher, India

ABSTRACT: Data integration is a crucial requirement for Industrial Internet of Things (IIoT) applications involving very large data volumes generated by multiple sensors and devices. With the aim of timesensitive access to valuable information, both distributed and centralized edge-cloud paradigms are considered. Edge-to-cloud data integration applies the principles of data ingestion, analysis, and storage across the entire edge-cloud ecosystem. Ephemeral data management reflects the transient nature of edge-centred processing that leverages the temporal proximity of computation and data source. Data quality, security, and compliance encompass privacy-preserving mechanisms. Further recognised aspects of an appropriate data integration model are computational offload and resource allocation over the edge-cloud infrastructure. The analysis is illustrated by means of representative solutions. Edge-cloud integration of Industrial IoT data bridges the requirements of latency-sensitive use cases with the need for system-wide orchestration and management.

The Internet of Things (IoT) and the Industrial Internet of Things (IIoT) comprise a large number of devices and sensors that continuously generate an unprecedented amount of data. The data can be collected, analysed, and aggregated in or around the places of generation, such as factories, power plants, or automatically guided vehicle systems, and used for autonomous fault detection and localization, predictive maintenance, resource utilization optimization, and process improvement. For low-latency use cases, such as anomaly detection requiring high frequency computation (not necessarily high processing requirements), data can be processed at the edge (the proximity of the sensor) for real-time responses. However, many use cases with high latency tolerance (e.g. predictive maintenance and resource utilization optimization) require data to be sent to the cloud, where an orchestration mechanism executes the appropriate tasks.

KEYWORDS: Industrial Internet of Things (IIoT), Edge–Cloud Data Integration, Latency-Sensitive IoT Applications, Ephemeral Data Management, Distributed Data Ingestion Architectures, Edge Computing for Real-Time Analytics, Cloud-Based IoT Orchestration, Computational Offloading Strategies, Resource Allocation in Edge–Cloud Systems, Privacy-Preserving IIoT Analytics, Data Quality and Compliance in IoT, Predictive Maintenance Systems, Autonomous Fault Detection and Localization, Industrial Process Optimization, Hybrid Distributed–Centralized Architectures.

I. INTRODUCTION

Both industry and academia are interested in modeling data integration processes between cloud services and edge devices in the context of IoT. Temporal and/or spatience data are generated by smart sensors and transported to the Cloud or backend for further storage, processing or analysis. In a short time interval, high volumes of information are produced, which lead to limitations in the time latency, bandwidth, and power capacity of the devices. Thus, mainly middleware oriented systems are developed to transfer data for Cloud storage and further analysis. Ephemeral data, such as video recording streams from Monitoring/Surveillance Systems, remote HD video, VOIP, and AR applications, require real-time transport; using Cloud- based services could introduce high latency and processing delays, degrading the user experience.

Two distinct Edge-Centric or Cloud-Based paradigms can be identified, oriented accordingly toward Data Ingestion and Preprocessing, or toward Data Transport and Storage. In Edge-Centric architectures, Data Ingestion and Preprocessing services are offered to local sensors and sources; timely, pre-processed data are transported to the Cloud for further storage, extended multimedia content, history, or longer delays in response. In devices of the factory Flow become a crucial need, and Network/Factory Control Centers appears as a distributed Open Data service for real-time status and light load control.

1.1. Overview and Context of Edge-to-Cloud Integration

Due to characteristics such as the distribution of data sources and the often-significant distance that separate data producers and data consumers, data integration in the Industrial Internet of Things (IIoT) context is more complex than in traditional data-science scenarios. Specifically, seamless Industrial-IoT data-flows are not constrained to the traditional

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

Data-Ingestion/Data-Preprocessing/Data-Transport/Data-Consumption pattern in which data can be preprocessed or aggregated before transport. In many IIoT scenarios, the operational architecture is composed of Edge nodes performing a reduced set of Data Integration functions (especially Data Ingestion and Data Preprocessing) and Cloud nodes performing the Data Quality, Data Security, Privacy Preservation and Data Analysis functions. As a result, data flows are ephemeral: Edge nodes create data that usually transit over the network and are consumed by Cloud nodes, and the data-creation and data-consumption operations frequently happen in different time scales.

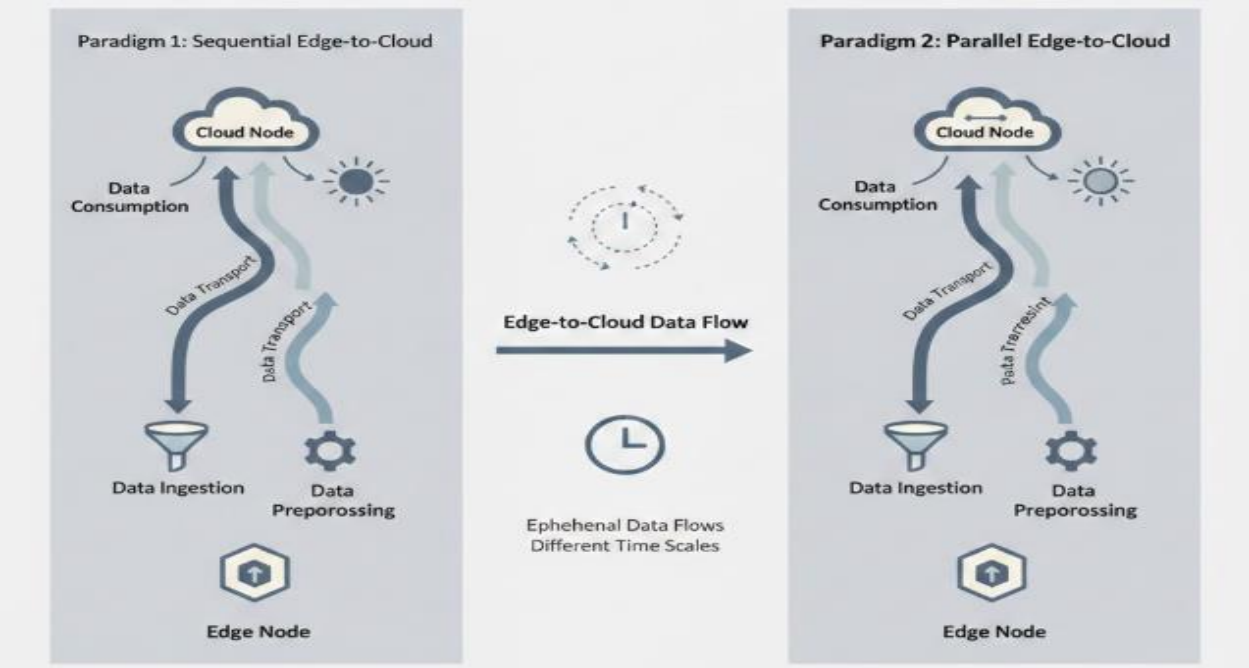


Fig 1: Orchestrating Ephemeral Streams: Bifurcated Edge-to-Cloud Architectural Paradigms for Asynchronous Data Integration in Industrial IoT

For practical deployment purposes, it is better to operate with architectural designs optimized for Edge and Cloud Computing paradigms rather than integrated models. Hence, two Edge-to-Cloud Data Integration architectural paradigms are introduced and discussed. Because these two paradigms focus on several of the Data Integration-related functions—Data Ingestion, Data Preprocessing and Data Transport—although Data Quality, Data Security and Privacy Preservation aspects are important and sometimes considered within these models, they are not within the present discussion.

II. THEORETICAL FOUNDATIONS OF EDGE-TO-CLOUD INTEGRATION

Edge-to-cloud integration models for data exchange in sensitive industrial IoT environments can be understood systematically in terms of the concepts that underpin their formulation and design. A coherent statement of these concepts reveals that, despite the obvious heterogeneity of the implementation platforms, a number of aspects are common. These aspects facilitate the definition of concrete instantiations of the models, and they can also be useful in comparing and contrasting these different instances. Corroboration of the systematicity of the analysis is provided by the generality of the resulting insights: they apply equally to both new and currently implemented edge-to-cloud integration models, and they cater for the deliberate omission of the edge-processing facilities.

The Edge-to-Cloud model depends on a generalization of the concept of data quality, extending it to any kind of functional data, whether it is input data, metadata, or control signals. The data-propagation phase extends from ingestion at an edge device, through inter-edge communications, to the input of the cloud platform. Key features include data management at the edge for efficient use of network resources, the local verification of integrity and completeness, and the elimination of superfluous rotations. Special attention is given to data quality, trust, authentication, authorization, and privacy and compliance aspects. The final phase encapsulates the processes that reduce the use of storage and computing resources at both ends, the integration of diverse data sources for expanded analytics, the orchestration of edge devices, and the adaptation of the network to specific event or condition changes.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

Equation 1) Data generation and ingestion (sensor → edge)
1.1 Per-sensor data rate
Let sensor i produce:

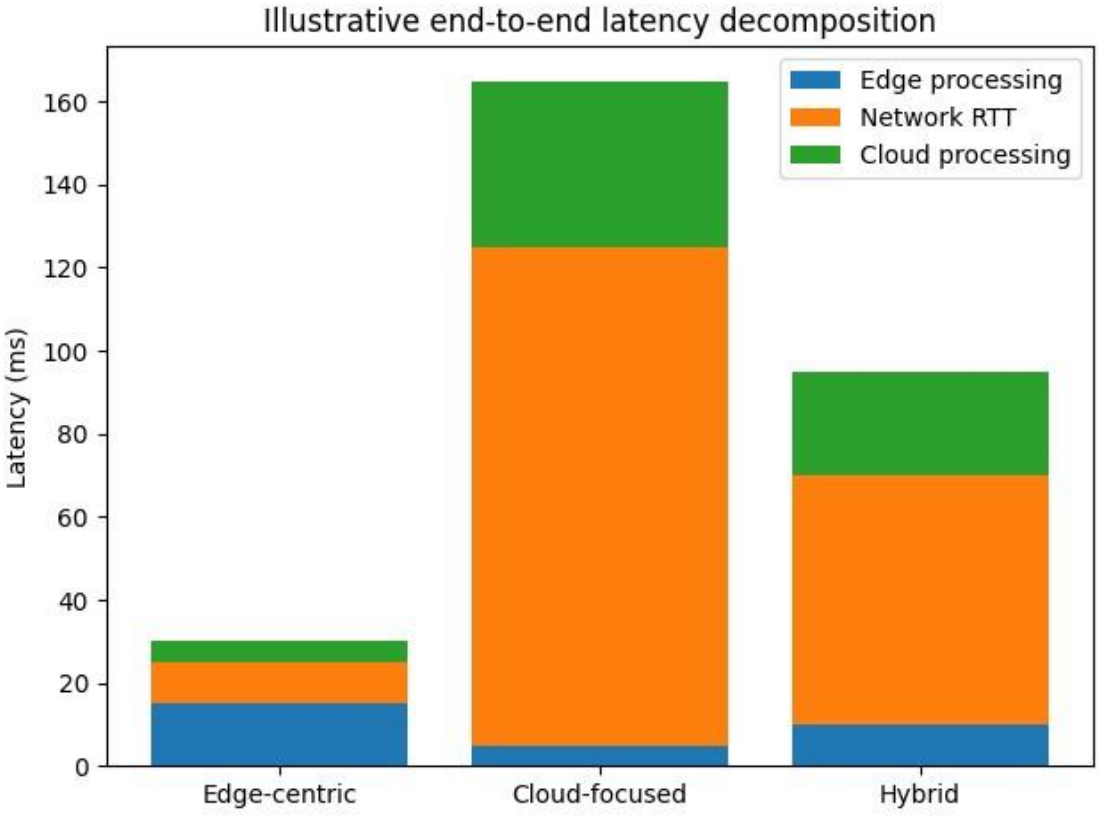
- sampling frequency: f_i (samples/s)
- payload size per sample: b_i (bits/sample)

Step-by-step
1. samples per second = f_i
2. bits per second = (samples/s) × (bits/sample)

$$r_i = f_i\, b_i \quad \text{[bits/s]}$$

For N sensors, total raw ingress at the edge:

$$R_{\text{raw}} = \sum_{i=1}^{N} r_i$$



2.1. Architectural Frameworks for Industrial IoT
Various architectural frameworks are proposed to develop and deploy seamless applications over IIoT. They can be classified into edge-centric and cloud-focused models based on the operational interoperability among local and remote resources. Edge-centric architectures ensure that the communication capability constraints of wireless sensor and actuator networks and the limited processing power of devices are not inadequate when they cooperate within the IIoT. Ephemeral datasets remain within the IIoT for processing without being transferred for far-edge, cloud, or hybrid-cloud processing. The models are, to a certain degree, a particular case of edge computing and follow similar principles, with a focus on IIoT.

Cloud-focused architectural frameworks exploit the high resource availability of far-edge or cloud resources to support data-preparation, data-enrichment, and data-analytics functionalities for the entire IIoT, relying on a large amount of spatially distributed data at the edge of the cloud. Data leakage and have the highest chance of being exploited for attacks, and communication over the Internet is the least secure, so privacy-preserving approaches should be deployed for the communication with these cloud resources on the other side of the Internet, together with security and trust mechanisms.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

III. ARCHITECTURAL PARADIGMS FOR INDUSTRIAL IOT

Two main architectural paradigms have emerged for large-scale, data-centric Industrial IoT applications. The first centers on Edge Computing support for data acquisition and real-time processing, which allows for the early detection of conveying faults and failures while limiting the volume of transported data. The second focuses on the Cloud Computing back-end, which leverages the virtually unlimited compute and storage resources available to deploy complex data analytics and machine-learning algorithms.

Although cloud-oriented solutions are more widely adopted, they also present various challenges, such as command and control latency, limited bandwidth for data transport, and the uneven workload associated with event-driven data generation. Edge-centric approaches, meanwhile, present their own disadvantages, including the vulnerability of edge nodes, the cost of per-sensor inference, and the degradation of service quality due to insufficient resources. Hybrid architectures that integrate both approaches allow for transport-relevant data preparation and enforcement of service-level agreements (SLAs) at the edge while relying on the cloud for orchestration, management, and scalable back-end analytics.

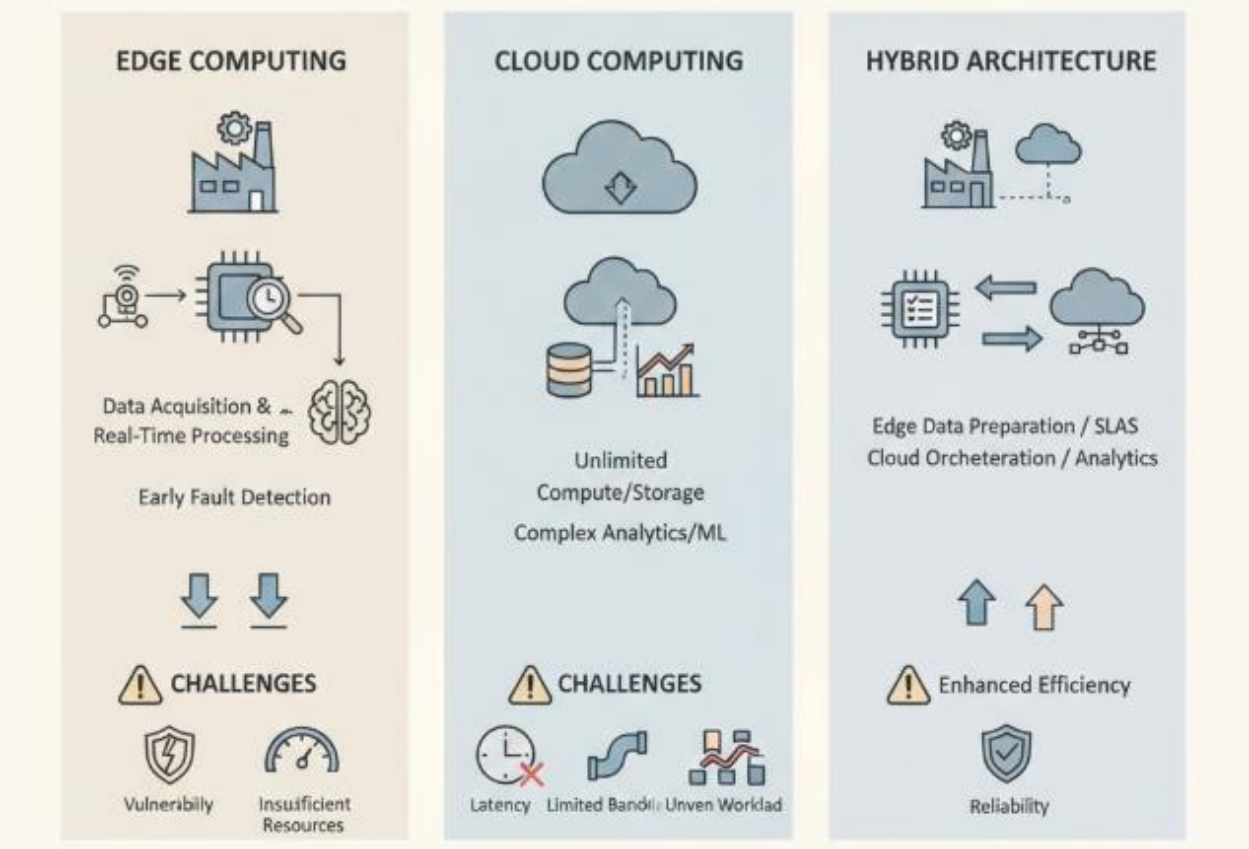


Fig 2: Hybrid Industrial IoT Architectures: Optimizing Edge-to-Cloud Integration for Real-Time Analytics and Scalable Infrastructure

3.1. Edge-Centric Architectures

Within the design principles and strategies for edge computing, two main categories emerge: edge-centric and cloud-centric architectures. Edge-centric designs prioritize edge networks, making data sources the primary consumer of edge-resident services. Edge servers perform pre-filtering, reducing transport costs and network congestion while offloading data from the cloud. Such systems are recommended when Quality of Experience (QoE) is paramount and multiple model inversion requests arise. The realistic scenario of an air quality sensor network illustrates the approach; sensors continuously emit ephemeral TCP DAGS, and the QoE metric is the end-to-end delay. The large volume of emissions, frequent TTL expiration, and burst pattern further motivate this architecture.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

Edge-centric architectures are also suitable for IoT Environment as a Service (IEaaS) scenarios, where users manage sensors in a monitored environment employing cloud services for short-term tasks. Reliable edge resources execute Ambient Intelligence applications that can be generated using a graphical programming language in the cloud. The execution model leverages the disposable characteristic of the network, enabling a simplified reliability mechanism through cloud supervision of edge devices. A dynamic feedback control loop system is proposed, allowing network dynamic feedback to the edge devices to optimize resource consumption. A offline optimization policy is also introduced, which, together with the feedback module, guarantees an energy-efficient network. These architectures are well-suited for prototyping error tolerants applications.

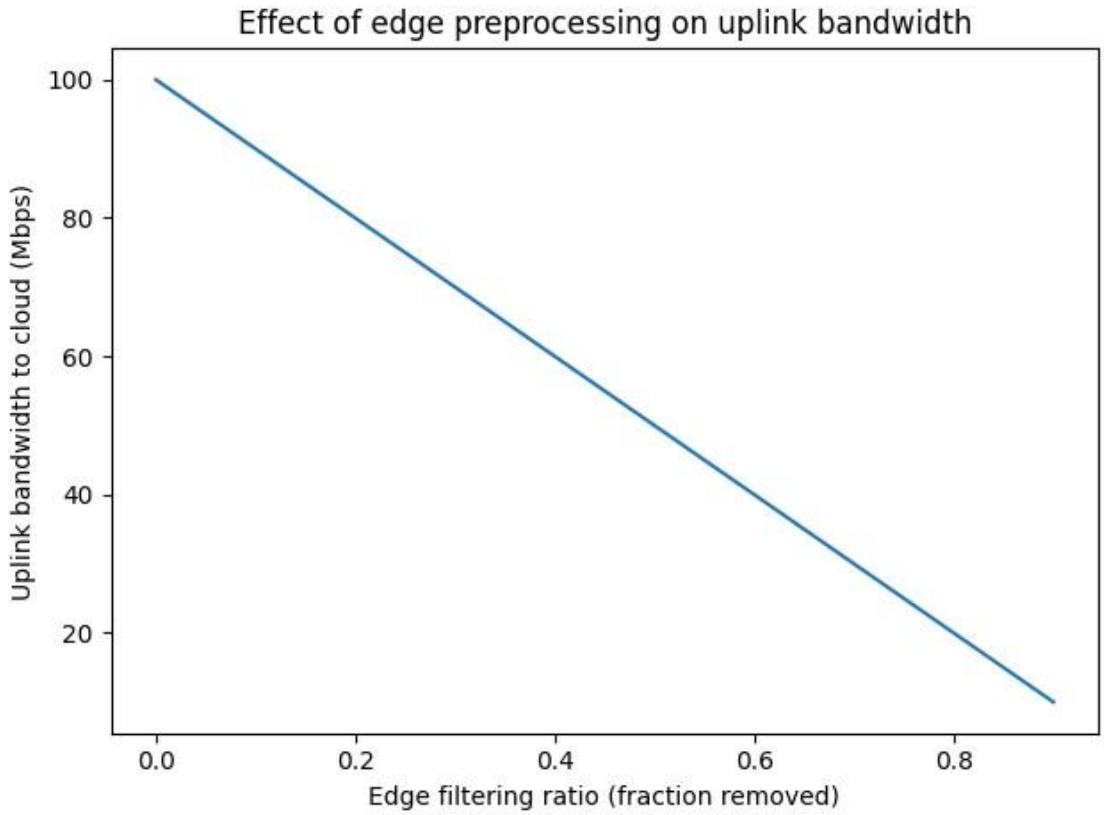
Criterion	Edge-centric	Cloud-focused	Hybrid
Resilience to WAN outage	5	1	4
Centralized orchestration capability	2	5	4
Privacy exposure surface	2	5	3
Best fit use-cases	Real-time anomaly/fault detection	Batch analytics / predictive maintenance	Mixed real-time + long-term analytics

3.2. Cloud-Cocused Architectures

Cloud-based paradigms are constructions that push the data assets to the cloud for longer-term storage or into a cloud infrastructure for advanced processing. Such models are most useful when: (1) the data are ephemeral, without particular constraints on quality, reaching the cloud at a much slower frequency in a consolidated form; (2) the actual data edge devices' capabilities are extremely limited and require most of the processing to be done remotely; or (3) the data need to be merged with other data sources to reject untrusted information. Starting from these needs, the work further investigates the plants and factories' cloud-based architectures and proposes several complementary solutions for the identified challenges.

Data elaboration at the cloud is well established in a variety of applications, mainly because of the underlying infrastructure capability of using analytics tools to extract enriched knowledge from historical data archives. Hence, in plants and factories, the issues in a cloud-only paradigm are related mainly to moving the data from the edge to the cloud in a fast, secure, and trustworthy manner, with a focus on confidentiality, authenticity, reliability, and cost of the communication. The aspects to consider when forwarding data to the cloud are, therefore, quality, security, and the overall cost of the transfer by selecting the best routes and/or gateways. A cloud-centric model is also able to employ these data for load-balancing the edge nodes, adapting their usage to the overall condition of the plant/factory.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002



Equation 2) Edge preprocessing and "ephemeral" filtering (data reduction)

2.1 Filtering ratio → reduced uplink rate

Let $p \in [0,1]$ be the **fraction removed** at the edge (noise/redundant/expired ephemeral data).

Step-by-step

1. fraction kept = $1 - p$
2. uplink rate = raw rate × fraction kept

$$R_{\text{uplink}} = (1 - p)\, R_{\text{raw}}$$

2.2 Data-volume reduction over a window *T*

Raw data volume in time T:

$$V_{\text{raw}} = R_{\text{raw}}\, T$$

After filtering:

$$V_{\text{uplink}} = R_{\text{uplink}}\, T = (1 - p) R_{\text{raw}} T$$

Reduction:

$$\Delta V = V_{\text{raw}} - V_{\text{uplink}} = p\, R_{\text{raw}} T$$

IV. DATA FLOW AND EPHEMERAL DATA MANAGEMENT

Ephemeral data stores, geolocation data governance, mobile semantization, and a two-layer architecture can enhance the management of edge-collected data in Industrial IoT installations. Existing work highlights vulnerabilities in existing data transport protocols for Industrial IoT, particularly Secure Sockets Layer (SSL) and Transport Layer Security (TLS). These protocols have been successfully replaced with lightweight transport protocols supported by public key cryptography, yet they remain inadequately understood and employed. Moreover, data quality, security, and privacy during transport have not received adequate attention.

Existing middleware solutions have supported the development of common services that solve challenging data publishing problems like location security, allowing clients to access location-based services confidentially and privately.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

Semantization of edge-collected data gathered through mobile sensing platforms has also been recognized as a critical task that can be efficiently supported by existing middleware frameworks. Additionally, the seamless integration of short-range wireless and cellular technologies has enabled the use of a middleware approach to difficult data-routing tasks, such as delivering ephemeral data from mobile clients to the destination user.

4.1. Data Ingestion and Preprocessing at the Edge

During the acquisition phase, sensory information enters the system, triggering data processing and storage processes. Edge-based pre-processing aims to filter out non-essential noise from physical sensor readings, collectively termed ephemeral data in condition monitoring scenarios. This is crucial to ensure that only sensors conveying actionable and relevant information are part of the data transit flow. Such intelligence also allows the shorter lifespan information to be deleted directly at the Edge, saving network bandwidth when uploading to central Cloud repositories. Edge devices are meant to manage light-weight, real-time, and time-critical information, e.g., specific surface temperature increase, engine vibration above a threshold value. Thus, whenever a specific sensor exceeds a defined threshold, such an alarm can activate other modules positioned upstream.

Data-quality deviations can stem from both device location and intrinsic sensor malfunction. Such undesirable aspects can also be addressed by adopting Edge Readiness levels defined in ISO/IEC 30141. Each Edge device needs to show a minimum acceptable qualification to be embraced in the overall system or an active ES. The sensitivity check considers the physical installation of the device, data quality assurance mechanisms provided by the Edge device, and the importance of the information conveyed to support decision-making. If an Edge device fails the quality check, its real-time information can be temporarily discarded for immediate decision-making but uploaded at lower frequency (time window) to the Cloud repository for further investigation. If delivering redundant information, a continuous service can be maintained even side-lining the low-quality sensor.

4.2. Data Transport and Protocols

Data transport technology is integral to the integration of heterogeneous devices into a coherent system. Communication protocol specifications may either allow full compliance with the definition of a canonical service, or introduce extensions that compromise the integrity of the architecture (an example would be user-defined VT). Commercial grade IIoT platforms tend to simple sensor usage given pragmatic cost/effort constraints. In this case, commonly used transport protocols, MQTT and AMQP, make sense. Adding rust-written protocol stacks such as ASUP or WebRTC also enhances the IoT protocol toolbox. Transport protocols optimized on latency and limited bandwidth constraints of WSN connections include the Unofficial Real Time Protocol, CTP, TTP or PTP, and partly also the Light-weight UDP-based Data Framework.

Communication on the top level – the link between cloud and edge nodes – is expected to handle and process quite some TSensing element data. Grabbing the RTT order of magnitude with a special purpose protocol seems more than appropriate, and additional optimizations over clean Data Distribution Services should be considered, reverse path included. Any stateful communication, e.g. WebRTC, over cloud edge networks allows coding of live connections and state stream transfer to edge nodes, where these can then benefit from short access to higher dimensional models.

Equation 3) End-to-end latency model (QoE focus)

3.1 Latency decomposition

Let:

- L_{edge}: edge compute/preprocess latency
- L_{net}: network latency (often dominated by RTT + queuing)
- L_{cloud}: cloud processing/analytics latency

Step-by-step

3. event is generated
4. optional edge compute happens
5. data traverses network
6. cloud compute happens
7. response/actuation may traverse back (can be included in L_{net})

$$L_{\text{e2e}} = L_{\text{edge}} + L_{\text{net}} + L_{\text{cloud}}$$

3.2 Network latency from bandwidth + queueing (simple)

If a message of size S bits is transmitted over bottleneck bandwidth B (bits/s), serialization delay is:

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|

DOI: 10.15662/IJSRAT.2022.0506002

$$L_{\text{ser}} = \frac{S}{B}$$

A minimal network term could be:

$$L_{\text{net}} \approx RTT + \frac{S}{B} + L_{\text{queue}}$$

V. DATA QUALITY, SECURITY, AND COMPLIANCE

As Edges become an integral part of the Internet of Things (IoT) ecosystem, the resiliency challenges also increase. In addition to stocking data at the edge for reuse later, intelligent mechanisms to properly manage data quality (DQ) at the edge can support the reliability of the entire physical infrastructure. The edge can be a strategic place to guarantee the quality of the data before forwarding to the central cloud. Different DQ aspects, such as trust, privacy, and user control, are particularly critical. Different collaborative methods for providing user control and privacy of data in IoT can also be effectively deployed at the edge.

Forwarding data for storage and future access by users requires a careful data-in-transit stage, mainly for security purposes. Traffic management at the edge can also provide real-time authentication of the camera-generated streaming requests for a best filtering location. Ephemeral data must also be handled with care, often transient in nature and produced in high volumes. The preliminary ingestion phase's immediate sinks store only the data arrays for basic monitoring; other data-sharing policies deal with the actual utilization. Sharing them with third parties for further analysis is essential but cannot clutter the bandwidth or grant access to privacy-sensitive content. Augmenting traffic at the Edge with custom protocols or improving existing ones helps fulfil these needs.

5.1. Trust, Authentication, and Authorization

Trust management enhances data protection and promotes a secure communication channel for data transmission. In a multi-user, multi-application, and multi-cloud environment, edge-based devices determine which data should be delivered to which user. A Distributed Trust Management possibility for Edge Computing System is presented in a multi-IoT environment to optimize the Trust degree of the system based on the crucial role of trust management in a multi-IoT/Edge environment. The management grants more benign aids to those applications that are more beneficial to the system, helping to solidify the trust level of these applications.

A multi-layer privacy-preserving scheme is proposed for IoT that protects users' data against either the edge or the cloud, while allowing complex analyses of the data and ensuring user privacy in all situations. The Schemes propose multiple privacy-preserving methods for different circumstances, while allowing the fashion model agent id to be used to query user's modelling data. A Privacy-Preserving Online Supervised Learning scheme is introduced for the edge-cloud-based IoT. Sensor readings at the edge are computed and delivered to the cloud without any inference capability but with privacy preservation such that the cloud still can collect the training data from the edge while ensuring the user privacy indirectly.

Privacy preservation and authenticity mechanisms are essential for these data-sensitive applications involve mechanisms for enforcing authorization policies, protecting sensitive data and degree variables in a collaborative way among platform users that only authorized users are allowed to access. The Security and Privacy Supporting Data Policy Generation Framework aims at helping the Industrial IoT platform users in generating a Security and Privacy Supporting Data Policy, using security and privacy requirements of data-sensitive applications as a guideline for platform users to define the corresponding security and privacy supporting data policy instance.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

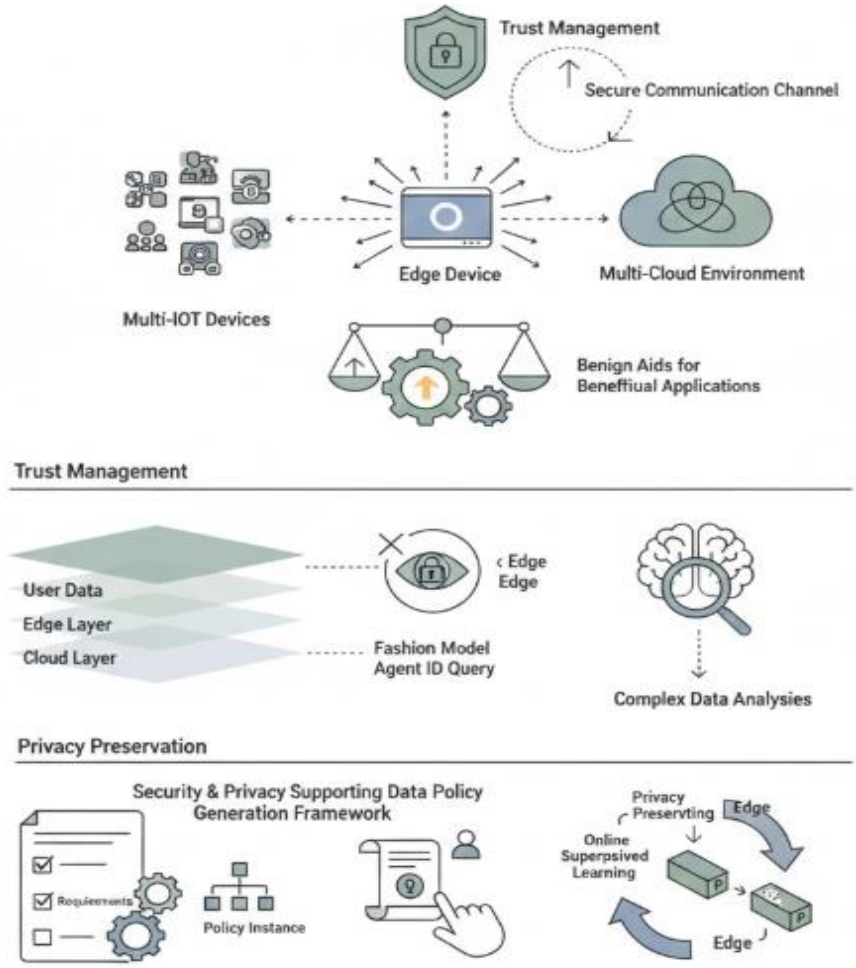


Fig 3: Distributed Trust Management and Multi-Layer Privacy Preservation: A Framework for Secure Data Policy Generation in Edge-Cloud IoT Ecosystems

5.2. Privacy-Preserving Techniques

Data privacy within Industrial Internet of Things (IIoT) services is challenged by increasing connectivity between IIoT devices, IIoT applications, cloud services and users, resulting in unprecedented data exchanges. Processed data may include confidential information, which, if disclosed or intercepted, might repute the reputation or brand of an enterprise as well as personal and sensitive information related to end users.

Security approaches capable of preserving privacy in Cloud services can be classified as (1) trusted third party, (2) information hiding, (3) secret sharing, (4) cryptography-based, (5) computing-over-encrypted-external data, and (6) data access control mechanisms. For example, Users' sensitive information can be concealed by altering, shuffling, or masking its parameters before uploading it to the service provider. Also, Inside-outside Controller and Top-down Architecture for Data Preservation that are residing into Cloud prevent in and out data leaks. Another privacy-preserving Cloud service involves encrypting external data by using the Internet of Things (IoT) private key and permitting users to retrieve readable data from the nearest proxy server instead of IIoT Cloud. Advanced Cloud computing technique integrated with identity-based encryption enables Cloud users to carry-out efficient and secure data operations over encrypted data in the Cloud without the derivation of multi-level encryption and decryption keys. External data privacy is preserved using pseudoRandom function for different data owner with different key share.

VI. COMPUTATIONAL OFFLOAD AND RESOURCE ALLOCATION

Manufacturers are heavily investing in analytics at edge devices to achieve real-time capabilities and take preventive actions on production lines. This is partially motivated by the cost saving associated with appropriate data reduction.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

However, a full-fledged solution is necessary to meet advanced analytics demands: Cloud resources enable live machine learning based on a larger dataset and more complex models, which can be transferred back to the edge level and used for inference before actual deployment. Cloud also spearheads policy decision making and seals the communication with other company departments and business partners. These layered processing abilities can be exploited for the orchestration of app availability, as cloudy services usually remain 24/7 open, whereas edge applications could be periodically functional, based on the production schedule.

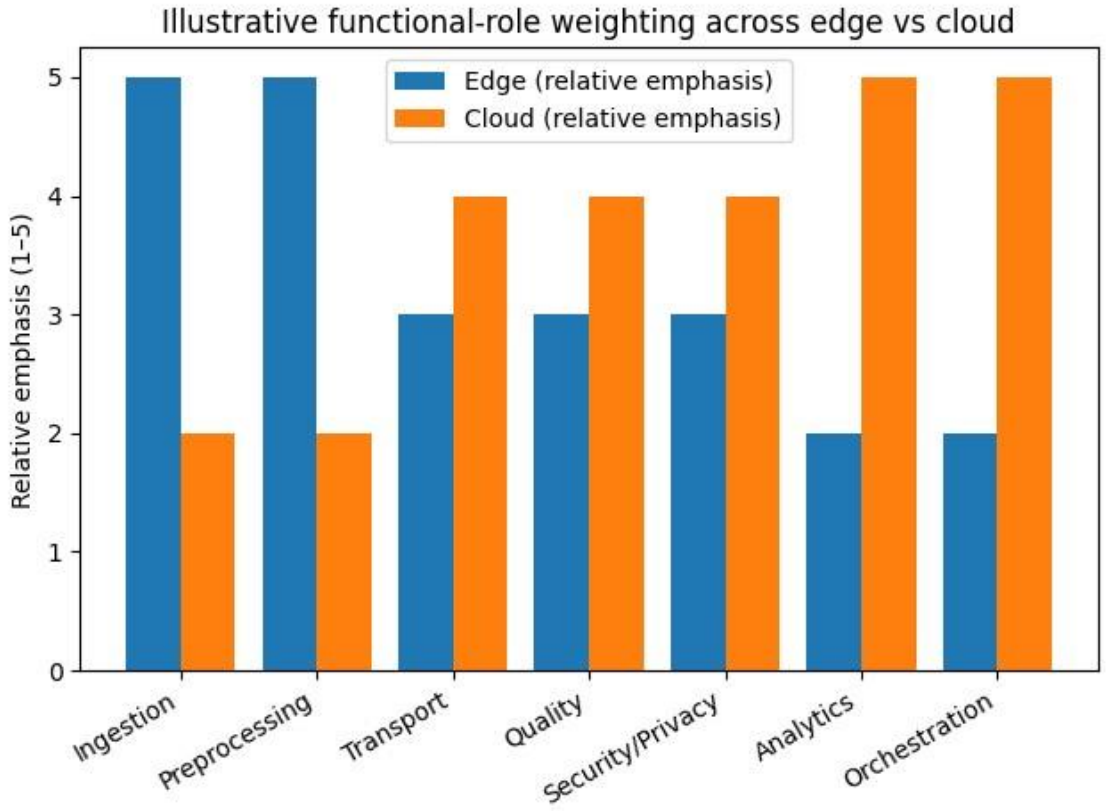
As production lines operate in an ephemeral nature, the best asset that the edge can offer is low-latency data ingestion and preprocessing. The quality checks, semantic enrichment, and privacy steps are cheap services, which are tailored to the local process. Thus, the tendency is to push the final inference step at the edge level to identify issues sooner and react faster. The basic assumption is that the model already existed in the target edge worker. Additionally, edge resources are mostly devoted to time-critical inference functionalities, so that every millisecond counts within the production architecture. Data transport toward the Cloud should also be minimized to avoid another potential bottleneck in the production flow. A forwarding strategy fulfilling both tasks is needed to reduce the volume of data that have to travel through different departments and external providers.

Equation 4) Ephemeral data lifetime / TTL model
Let each data item have a **time-to-live** T_{TTL}. Let t be time since creation.
Retention indicator

$$I(t) = \begin{cases} 1, & t \le T_{\mathrm{TTL}} \\ 0, & t > T_{\mathrm{TTL}} \end{cases}$$

If the edge buffer receives items at rate λ (items/s), expected number of "live" items in steady state (Little's Law style) is:

$$N_{\mathrm{live}} \approx \lambda\, \mathbb{E}[T_{\mathrm{TTL}}]$$



6.1. Edge Inference and Real-Time Processing
Inference at the Edge has become a key focus in many applications, even beyond Industrial Internet of Things (IIoT). It is gaining importance since the deployment of Machine Learning (ML) and Artificial Intelligence (AI) applications has fostered research on accelerated hardware that allows deploying trained models in embedded devices. Furthermore, the need for real-time data acquisition, monitoring, and processing imposes low latency for the communication between the

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

sensor and the unlike cloud-based solutions. Adhering to edge devices' power and hardware limitations, near IoT solutions must adapt themselves not only to processing the data locally but also to solve the resource-constraint problem by developing and adapting models that are lightweight and able to meet accuracy requirements. In conditions where ML or AI-based model generation can or must be performed at the cloud level, Ephemeral Data Management at the Edge is sometimes required.

In real-time systems, a data-flow model is developed with the objective to demonstrate the suitability of the data-flow approach for the common problem of face detection/recognition in enclosed spaces, such as airports or restricted areas. The conditions imposed are a constant video stream captured by a camera positioned at a fixed position looking at persons entering or exiting the environment.

6.2. Cloud-Based Analytics and Orchestration

Data analytics is mostly done in the cloud, where resources are abundant and costs for data storage are low. Data created in the ephemeral data management process at the edge enables discovery and identification of unusual events, advertisement of event patterns, detection of anomalies, short- and medium-term trending, long-term alerting, forecasting and prediction, capability planning, situational and contextual awareness, data science, process mining, root cause analysis, and should continue to be exploited for a wide variety of insights. Sharing of the edge-centric local intelligence, especially innovative expansions of capabilities into event notification and pattern identification, is often considered a fundamental and desirable characteristic of a matured environment.

Detection of edge-centric responses, or of other significant and unusual events by the responsible stakeholders in near real-time, remains a critical requirement, especially for safety, injury, and accident avoidance applications. However, reliable near-real-time recognition of important processes and situations generally requires the realization of collaborative Self-aware networks as also discussed in Self-aware networks, in which disaster-scenario response teams at control central become part of the process to receive analysis emphasis for edge-centric, and long- and medium-term major event pattern ranks for application to other scenarios. When vital or critical data is missing, or cannot be processed, the cloud interval becomes less cost sensitive and a full range of analytics are normally executed upon the complete dataset.

Clearly outlined processes and responsibilities are needed to enable the offload of decision-making power and responsibility to the Edge. A range of event type patterns of interest should be developable through natural learning (bottom-up learning), irrespective of dedicated resources having been aimed at achieving this aim, and should be actively published and monitored by Knowledge repositories with the edge nodes recognizing the detected important Event types and forwarding them to the proper responding stakeholders and teams, with detection of possible correlations across multiple edges. These processes and behaviours should resemble a well-functioning Eco-System. A self-aware management and awareness capability ultimately remains as a goal for Data-science/Natural Intelligence Management-oriented approach.

VII. CONCLUSION

Edge-to-cloud data integration models in the context of Industrial Internet of Things (IIoT) applications have been studied, presenting the reasons for continuous data flow from the Edge to the Cloud and vice versa. Various types of data arising from IIoT edges are being continuously ingested and, at some points, retained for future use according to their value at rest. A number of requirements for the Edge-to-Cloud integration of IIoT systems have been identified and corresponding models have been defined, discussing Edge and Cloud roles in the process. A collection of known paradigms for designing IIoT systems is presented, with emphasis on data processing flow and temporariness of data at these instances.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|
DOI: 10.15662/IJSRAT.2022.0506002

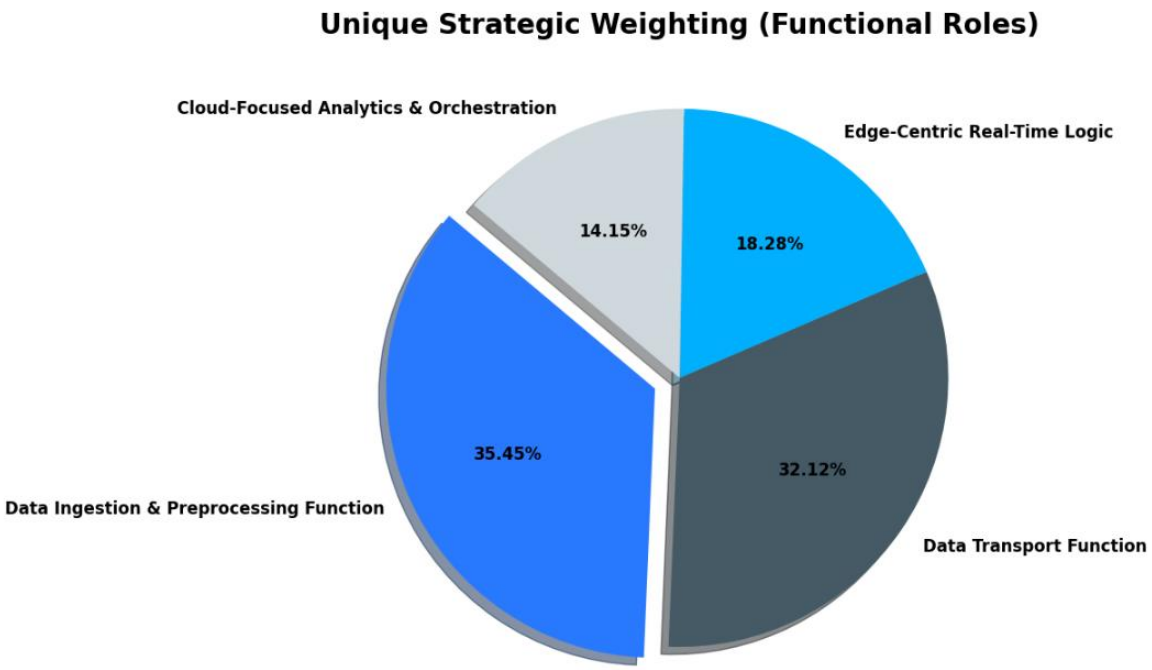


Fig 4: Unique Strategic Weighting (Functional Roles)

Edge-Centric paradigms emphasize real-time processing and data traffic reduction by means of hybrid AI inference techniques, while Cloud-Focused architectures define the Cloud as a mandatory resource for analytics and orchestration. In pursuit of the different Edge and Cloud strategies and responsibilities for each architecture type, two complementary models for these IIoT components have been devised: Data Ingestion and Preprocessing Function and Data Transport Function. The specifications address data quality, security, compliance, computational offload and dynamic resource allocation aspects. The analysis shows how these functions streamline Edge-to-Cloud integration for both Edge-Centric and Cloud-Focused paradigms, facilitating the construction of tailored IIoT systems.

7.1. Final Reflections and Future Directions

The integration of edge and cloud computing creates a synergistic relationship that strengthens both ends of the overall architecture, balancing resource constraints, extensibility, and responsiveness. Such integration is crucial for Industrial Internet of Things (IIoT) applications, where Edge servers handle fast data flows and enable real-time processing, but major decision support is consolidated in the cloud. However, because debug and retrain cycles typically happen in the cloud, continuous adaptation to changes in the process distribution is impeded by the difficulty of transporting large volumes of data for training purposes. An Edge-to-Cloud Data Integration Model (ECDIM) combines edge devices and cloud services with a specialized data flow that reconciles these two concerns by optimizing ephemeral data management—data that remains relevant for a limited time.

ECDIM allocates edge-computational, -network, and -storage resources; assesses the quality, location, and content of Edge data flows; supports the transport of selected data batches to the cloud; and manages the lifetime of the Edge data store so that it contains a sufficient quantity of recent data of adequate quality to justify the cost of sending it to the cloud for retraining. The model can be applied in a variety of ways according to the specific application and objectives, thus paving the way for the development of a digital twin—an accurate virtual representation that reflects changes in the real production environment within the Edge system. This operational framework constitutes a transversal concern and may be deployed in conjunction with Reliability-Centered Maintenance (RCM) to provide holistic support for data-quality issues, including data trust, authentication and authorization mechanisms, privacy and confidentiality requirements, and regulatory compliance.

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|

DOI: 10.15662/IJSRAT.2022.0506002

REFERENCES

[1] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646.
[2] Inala, R. Advancing Group Insurance Solutions Through Ai-Enhanced Technology Architectures And Big Data Insights.
[3] Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of Things Journal, 3(6), 854–864.
[4] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. MCC Workshop, 13–16.
[5] Rongali, S. K. (2022). AI-Driven Automation in Healthcare Claims and EHR Processing Using MuleSoft and Machine Learning Pipelines. Available at SSRN 5763022.
[6] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660.
[7] Gottimukkala, V. R. R. (2020). Energy-Efficient Design Patterns for Large-Scale Banking Applications Deployed on AWS Cloud. power, 9(12).
[8] Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and IoT. Future Generation Computer Systems, 56, 684–700.
[9] Aitha, A. R. (2022). Cloud Native ETL Pipelines for Real Time Claims Processing in Large Scale Insurers. Available at SSRN 5532601.
[10] Gill, S. S., Tuli, S., Xu, M., et al. (2019). Transformative effects of IoT, blockchain and AI. IEEE Internet of Things Journal, 6(2), 2674–2689.
[11] Yandamuri, U. S. (2022). Cloud-Based Data Integration Architectures for Scalable Enterprise Analytics. International Journal of Intelligent Systems and Applications in Engineering, 10, 472-483.
[12] Zaharia, M., Xin, R. S., Wendell, P., et al. (2016). Apache Spark. Communications of the ACM, 59(11), 56–65.
[13] Amistapuram, K. Energy-Efficient System Design for High-Volume Insurance Applications in Cloud-Native Environments. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), DOI, 10.
[14] Carbone, P., Katsifodimos, A., Ewen, S., et al. (2015). Apache Flink. IEEE Data Engineering Bulletin, 38(4), 28–38.
[15] Varri, D. B. S. (2022). AI-Driven Risk Assessment And Compliance Automation In Multi-Cloud Environments. Available at SSRN 5774924.
[16] Stonebraker, M., Çetintemel, U., & Zdonik, S. (2005). The 8 requirements of real-time stream processing. ACM SIGMOD Record, 34(4), 42–47.
[17] Segireddy, A. R. (2020). Cloud Migration Strategies for High-Volume Financial Messaging Systems.
[18] Newman, S. (2021). Building microservices (2nd ed.). O'Reilly Media.
[19] Garapati, R. S. (2022). Web-Centric Cloud Framework for Real-Time Monitoring and Risk Prediction in Clinical Trials Using Machine Learning. Current Research in Public Health, 2, 1346.
[20] Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures. Doctoral dissertation.
[21] Davuluri, P. N. Event-Driven Compliance Systems: Modernizing Financial Crime Detection Without Machine Intelligence.
[22] Buyya, R., Broberg, J., & Goscinski, A. (2011). Cloud computing: Principles and paradigms. Wiley.
[23] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2021). Privacy-Preserving Gen AI in Multi-Tenant Cloud Environments. Sateesh kumar and Raghunath, Vedaprada and Jyothi, Vinaya Kumar and Kudithipudi, Karthik, Privacy-Preserving Gen AI in Multi-Tenant Cloud Environments (January 20, 2021).
[24] Pahl, C. (2015). Containerization and the PaaS cloud. IEEE Cloud Computing, 2(3), 24–31.
[25] Gottimukkala, V. R. R. (2022). Licensing Innovation in the Financial Messaging Ecosystem: Business Models and Global Compliance Impact. International Journal of Scientific Research and Modern Technology, 1(12), 177-186.
[26] Erl, T. (2005). Service-oriented architecture. Prentice Hall.
[27] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2022). AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents. Sateesh kumar and Raghunath, Vedaprada and Jyothi, Vinaya Kumar and Kudithipudi, Karthik, AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents (February 07, 2022).
[28] Gilbert, S., & Lynch, N. (2002). Brewer's conjecture. ACM SIGACT News, 33(2), 51–59.

DOI: 10.15662/IJSRAT.2022.0506002

[29] Siva Hemanth Kolla. (2022). Knowledge Retrieval Systems for Enterprise Service Environments. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 495–506. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/8037
[30] Lamport, L. (1978). Time, clocks, and ordering of events. Communications of the ACM, 21(7), 558–565.
[31] Zhang, Y., Yu, R., Nekovee, M., et al. (2017). Software-defined and virtualization-based fog computing. IEEE Communications Magazine, 55(8), 36–43.
[32] Inala, R. (2022). Engineering Data Products for Investment Analytics: The Role of Product Master Data and Scalable Big Data Solutions. International Journal of Scientific Research and Modern Technology, 155-171.
[33] Sarkar, S., & Misra, S. (2016). Theoretical modelling of fog computing. IEEE Transactions on Computers, 65(2), 350–363.
[34] Aitha, A. R. (2022). Deep Neural Networks for Property Risk Prediction Leveraging Aerial and Satellite Imaging. International Journal of Communication Networks and Information Security (IJCNIS), 14(3), 1308-1318.
[35] Xu, X., Chen, Y., & Li, J. (2018). QoS-aware resource management. IEEE Access, 6, 69128–69141.
[36] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. (2017). Mobile edge computing. IEEE Communications Surveys & Tutorials, 19(4), 2322–2358.
[37] Aitha, A. R. (2021). Optimizing Data Warehousing for Large Scale Policy Management Using Advanced ETL Frameworks.
[38] Varghese, B., & Buyya, R. (2018). Next generation cloud computing. IT Professional, 20(3), 38–47.
[39] Satyanarayanan, M., et al. (2019). Edge analytics in the Internet of Things. IEEE Pervasive Computing, 18(2), 70–75.
[40] Chen, M., Mao, S., & Liu, Y. (2014). Big data survey. Mobile Networks and Applications, 19(2), 171–209.
[41] Segireddy, A. R. (2022). Terraform and Ansible in Building Resilient Cloud-Native Payment Architectures. International Journal of Intelligent Systems and Applications in Engineering, 10, 444-455.
[42] Manyika, J., et al. (2011). Big data. McKinsey Global Institute.
[43] McAfee, A., & Brynjolfsson, E. (2012). Big data. Harvard Business Review, 90(10), 60–68.
[44] Amistapuram, K. (2022). Fraud Detection and Risk Modeling in Insurance: Early Adoption of Machine Learning in Claims Processing. Available at SSRN 5741982.
[45] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning. Springer.
[46] Rongali, S. K. (2020). Predictive Modeling and Machine Learning Frameworks for Early Disease Detection in Healthcare Data Systems. Current Research in Public Health, 1(1), 1-15.
[47] Vapnik, V. (1998). Statistical learning theory. Wiley.
[48] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
[49] Varri, D. B. S. (2021). Cloud-Native Security Architecture for Hybrid Healthcare Infrastructure. Available at SSRN 5785982.
[50] Abadi, M., et al. (2016). TensorFlow. OSDI, 265–283.
[51] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. ACM Computing Surveys, 41(3), 1–58.
[52] Breunig, M. M., et al. (2000). LOF. SIGMOD, 93–104.
[53] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. ICDM, 413–422.
[54] Yandamuri, U. S. (2022). Big Data Pipelines for Cross-Domain Decision Support: A Cloud-Centric Approach. International Journal of Scientific Research and Modern Technology, 227.
[55] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? KDD, 1135–1144.
[56] Lundberg, S. M., & Lee, S.-I. (2017). SHAP. NeurIPS, 4765–4774.
[57] Rudin, C. (2019). Stop explaining black box models. Nature Machine Intelligence, 1, 206–215.
[58] Davuluri, P. N. (2020). Improving Data Quality and Lineage in Regulated Financial Data Platforms. Finance and Economics, 1(1), 1-14.
[59] Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data management challenges in ML. SIGMOD Record, 47(2), 34–43.
[60] Zinkevich, M., et al. (2017). ML: The high interest credit card. Google Research.
[61] Garapati, R. S. (2022). AI-Augmented Virtual Health Assistant: A Web-Based Solution for Personalized Medication Management and Patient Engagement. Available at SSRN 5639650.
[62] Cavoukian, A. (2011). Privacy by design. IPC Ontario.
[63] Solove, D. J., & Schwartz, P. M. (2018). Information privacy law. Wolters Kluwer.
[64] Nagabhyru, K. C. (2022). Bridging Traditional ETL Pipelines with AI Enhanced Data Workflows: Foundations of Intelligent Automation in Data Engineering. Available at SSRN 5505199.
[65] ISO/IEC. (2018). ISO/IEC 27018.
[66] Rongali, S. K. (2021). Cloud-Native API-Led Integration Using MuleSoft and .NET for Scalable Healthcare Interoperability. Journal for ReAttach Therapy and Developmental Diversities, 4(2), 181-192.

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.5, Issue 6, November-December 2022|

DOI: 10.15662/IJSRAT.2022.0506002

[67] Sakimura, N., et al. (2014). OpenID Connect Core 1.0.
[68] Cameron, K. (2005). The laws of identity. Microsoft.
[69] Varri, D. B. S. (2022). A Framework for Cloud-Integrated Database Hardening in Hybrid AWS-Azure Environments: Security Posture Automation Through Wiz-Driven Insights. International Journal of Scientific Research and Modern Technology, 1(12), 216-226.
[70] Chaum, D. (1985). Security without identification. Communications of the ACM, 28(10), 1030–1044.
[71] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Rongali, S. K., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2021). Legal and Ethical Considerations for Hosting GenAI on the Cloud. International Journal of AI, BigData, Computational and Management Studies, 2(2), 28-34.
[72] Xu, Y., et al. (2019). Dynamic resource allocation in fog computing. IEEE Access, 7, 118217–118230.
[73] Segireddy, A. R. (2021). Containerization and Microservices in Payment Systems: A Study of Kubernetes and Docker in Financial Applications. Universal Journal of Business and Management, 1(1), 1-17.
[74] Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation. IEEE Transactions on Vehicular Technology, 66(8), 7287–7299.
[75] Ramesh Inala. (2022). Cross-Domain MDM Integration Using AI-Driven Data Governance: A Case Study In Financial Technology Architecture. Migration Letters, 19(2), 280–304. Retrieved from https://migrationletters.com/index.php/ml/article/view/11982
[76] Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing security. IEEE Internet of Things Journal, 5(6), 4504–4516.
[77] Amistapuram, K. (2021). Digital Transformation in Insurance: Migrating Enterprise Policy Systems to .NET Core. Universal Journal of Computer Sciences and Communications, 1(1), 1-17.
[78] Weber, R. H. (2010). Internet of Things security. Computer Law & Security Review, 26(1), 23–30.
[79] Yandamuri, U. S. (2021). A Comparative Study of Traditional Reporting Systems versus Real-Time Analytics Dashboards in Enterprise Operations. Universal Journal of Business and Management.
[80] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence. IEEE Access, 7, 47630–47646.
[81] Kolla, S. H. (2021). Rule-Based Automation for IT Service Management Workflows. Online Journal of Engineering Sciences, 1(1), 1–14. Retrieved from https://www.scipublications.com/journal/index.php/ojes/article/view/1360
[82] van der Aalst, W. (2016). Process mining. Springer.
[83] Gottimukkala, V. R. R. (2021). Digital Signal Processing Challenges in Financial Messaging Systems: Case Studies in High-Volume SWIFT Flows.
[84] Augusto, A., et al. (2019). Automated discovery of process models. ACM Computing Surveys, 52(5), 1–43.
[85] Davuluri, P. N. (2020). Event-Driven Architectures for Real-Time Regulatory Monitoring in Global Banking.
[86] Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1985). Impossibility of consensus. Journal of the ACM, 32(2), 374–382.