UDC 004.75:004.4

THE EVOLUTION OF WEB ARCHITECTURES: FROM MONOLITHS TO EDGE COMPUTING

Roilian M.
Engineering Manager, LeanDNA (Austin, USA)

ЭВОЛЮЦИЯ ВЕБ-АРХИТЕКТУР: ОТ МОНОЛИТОВ К EDGE COMPUTING

Ройлян М.
Engineering Manager, LeanDNA (Остин, США)

Abstract
The article explores the stages in the evolution of web architectures within the context of industrial IT systems – from monolithic and centralized solutions to distributed models, including edge computing. It analyzes architectural and network transformations associated with the shift toward edge-based computation, as well as the impact of these changes on fault tolerance, processing latency, and node autonomy. It is emphasized that the implementation of edge infrastructures requires new approaches to containerization, communication protocols, and security. Special attention is given to the use of edge computing for inventory management, production processes, and integration with ERP/SCADA systems. Practical examples from high-precision manufacturing and logistics sectors are presented. The article underscores the need to rethink architectural principles in the era of industrial digitalization.

Keywords: web architectures, edge computing, containerization, industry, protocols, microservices.

Аннотация
В статье рассматриваются этапы эволюции веб-архитектур в контексте промышленных ИТ-систем – от монолитных и централизованных решений к распределенным моделям, включая edge computing. Анализируются архитектурные и сетевые трансформации, связанные с переходом к вычислениям на периферии, а также влияние этих изменений на отказоустойчивость, задержки обработки и автономность узлов. Подчеркивается, что внедрение edge-инфраструктур требует новых подходов к контейнеризации, протоколам взаимодействия и безопасности. Особое внимание уделено применению edge computing для управления запасами, производственными процессами и интеграции с ERP/SCADA-системами. Представлены практические примеры из отраслей высокоточных производств и логистики. Статья актуализирует необходимость переосмысления архитектурных принципов в условиях цифровизации промышленности.

Ключевые слова: web-архитектуры, edge computing, контейнеризация, промышленность, протоколы, микросервисы.

Introduction

Modern manufacturing companies, especially in high-tech sectors, require fault-tolerant information systems capable of rapid response amid supply chain uncertainty and increasing data volumes from production and warehouse environments. This demand has driven a shift in web architectures – from centralized monoliths to distributed and hybrid models that process data closer to its source. Transitioning from cloud-based systems to edge computing addresses limitations in network bandwidth, communication latency, and the need for uninterrupted operation during connectivity loss. Edge computing enables local processing, decentralizes decision-making, and enhances the autonomy of digital production platforms, making it especially effective for inventory management, equipment monitoring, and real-time traceability.

The current article aims to explore the evolution of web architectures in the context of their application to industrial IT systems for controlling material flow and inventories, with a focus on the technological and network-specific nature of edge computing. It explores infrastructural details of the solutions like network topology design, interaction among edge nodes and the cloud service, and the potentiality of integrating these along supply chains.

Main part. Technological evolution of web systems

The creation of web applications for the management of industrial and logistics processes represents fundamental shifts in the methods of data processing and transmission, as well as the architecture of software solutions. In the early days of IT infrastructure development, **centralized architectures** based on mainframes and monolithic applications were common. With the advent of **distributed computing**, this was superseded by the use of Service-Oriented Architecture (SOA), followed by **microservice-based solutions**. SOA advocated modularity and reusability of business logic with standardized interfaces, most commonly achieved through SOAP and XML. The evolution towards microservices was a natural continuation of the decomposition trend: applications were refactored into separate components, each doing a specific task and communicating with others using lightweight REST API or message queues.

The next step in the evolution has been **cloud computing**, adopted by many organizations to address scalability, optimize infrastructure costs, and manage centralized services. However, in industrial settings, the above solutions were limited by several critical factors, including latency and external network dependency. The need to balance centralized control with local responsiveness initially led to the development of **hybrid architectures**, which combined cloud-based orchestration with localized processing capabilities. As system demands for real-time decision-making, bandwidth efficiency, and operational autonomy continued to grow, **edge computing** emerged as a logical next step. It advanced the hybrid model by shifting more computational logic directly to edge nodes – closer to sensors, machines, and production lines – thereby enhancing responsiveness, reducing reliance on central connectivity, and enabling more resilient, decentralized industrial systems (table 1).

Table 1

Evolution of web architectures [1, 2]

Architectural model / period of adoption	**Key features**	**Advantages**	**Disadvantages**
Centralized (mainframe / monolith), 1970 – 1990s	Single point of processing; high component coupling; low flexibility.	Centralized control and security.	Low scalability; single point of failure.
Distributed (SOA / microservices) 2000 – 2015	Modularity; reusability of components; standardized interfaces.	Flexibility; component reusability.	Complexity of service coordination; high coupling in ESB
Cloud / hybrid 2010 – present	Scalability; centralized management; resource usage based on demand.	Rapid scaling; cost optimization.	Network latency; dependence on network and cloud providers.
Edge computing 2015 – present	Data processing near the source. Autonomy; low latency.	Minimization of latency. Resilience to connection loss.	Local infrastructure requirements. Complex security assurance infrastructure.

As shown in the table, the evolution of web architectures in the industrial environment represents a gradual transition from centralized, poorly scalable solutions to flexible, fault-tolerant, and adaptive models capable of efficiently processing data near their generation sources. Each architectural paradigm reflected the technological maturity of its time and addressed the current challenges within the existing constraints.

Edge computing as a response to low-latency requirements

Modern manufacturing and logistics processes are increasingly demanding strict requirements for the response time of information systems. This is due to the need for immediate response to changes in equipment status, real-time processing of telemetry from sensors, as well as ensuring synchronization of actions at various levels of the production chain [3]. It is against this landscape that the edge computing model is being developed – an architectural pattern in which computations and decision-making occur close to the source of data, in contrast to remote cloud data centers.

According to Precedence Research, the global edge computing market is estimated to be worth $432,94 billion in 2024 and is expected to reach approximately $5132,29 billion by 2034, growing at an average rate of 28% from 2025 to 2034. North America has remained the market leader in recent years, accounting for about 40%.

The edge computing model is structured to address the growing need for low-latency data processing and real-time decision-making. Three major levels constitute the edge model architecture (fig. 1).

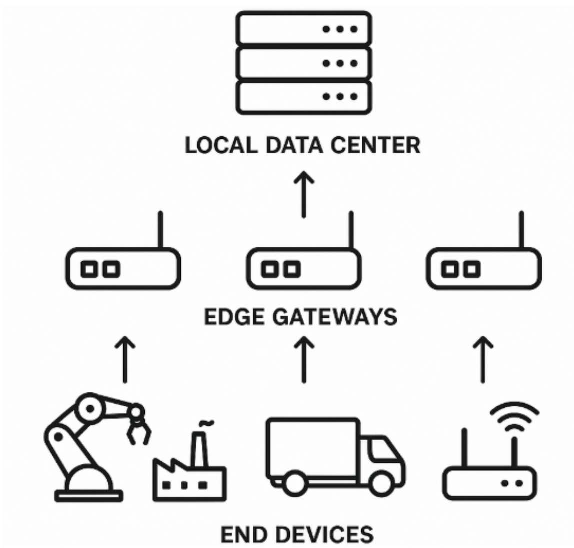

Figure 1. Architecture of edge models

Devices consist of sensors, actuators, and embedded computers that produce raw data streams. **Gateways** act as an intermediate layer, carrying out aggregation, preprocessing, and preliminary data analysis. They can utilize local AI models for making independent decisions and send only aggregated or vital data to the cloud. **Local data centers**, which are on-site or proximate, do more sophisticated tasks, synchronize between nodes, and offer temporary data storage whenever global networks are not accessible [4]. This design minimizes the network load, lowers the response time, and increases the overall system reliability.

The implementation of edge computing as infrastructure demands tight control over network connection properties, particularly in industrial settings where low latency and high availability are paramount. These solutions rely on developing an infrastructure distributed network with **Quality of Service (QoS)** support – a feature through which differentiation of traffic and bandwidth guaranteeing priority data flows are provided (fig. 2).

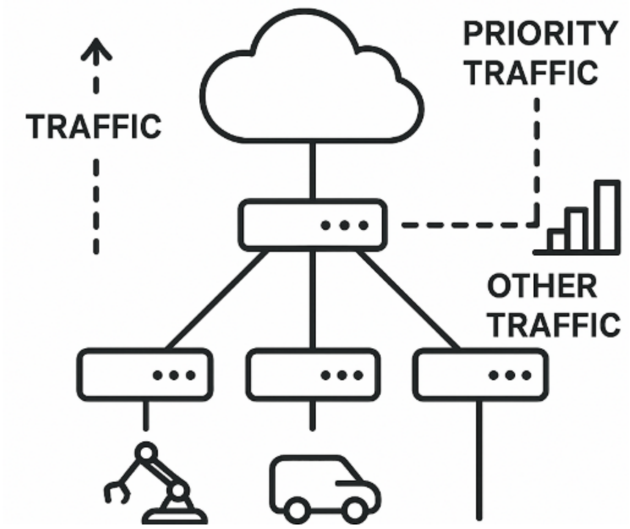

Figure 2. QoS scheme

To ensure scalability and centralized management, the **Software-Defined WAN (SD-WAN)** technology is used. It abstracts the network's logical model from physical transmission channels, implementing routing and policy management at the software level. The global SD-WAN market demonstrates steady annual growth, and according to QKS Group, it is expected to reach $7,04 billion by 2030, with an annual growth rate of 19,43%.

In edge-oriented systems, SD-WAN enables SLA-aware routing, where traffic with low latency tolerance (such as video analytics, data from PLC, and RFID systems) is directed through channels with the lowest RTT (Round-Trip Time) and minimal packet loss. Meanwhile, less critical traffic (such as telemetry for archiving) can be rerouted through channels with higher latency but greater bandwidth (fig. 3).

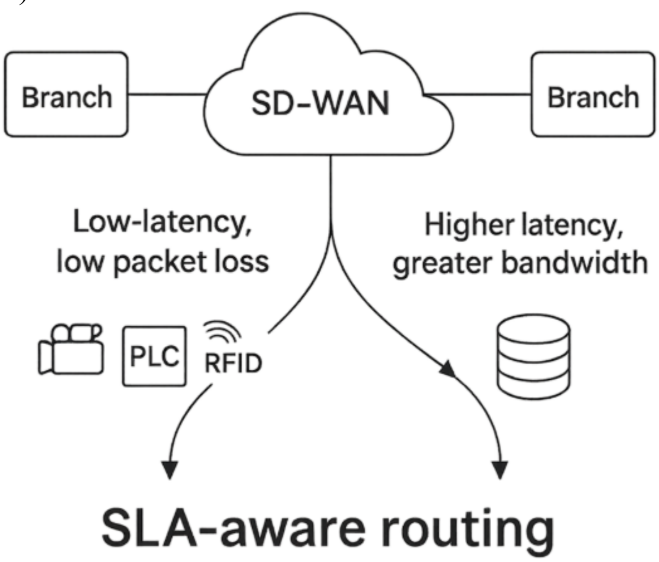

Figure 3. SD-WAN architecture with SLA-aware routing

In addition, the edge infrastructure includes mechanisms such as link redundancy, local breakout, and policy-based forwarding, which are critical in industrial scenarios with high downtime costs. This network architecture not only ensures low data transmission latency but also provides resilience against network degradation or individual link failures, maintaining production continuity and minimizing the risks of disrupting technological cycles.

Technology stack and interaction protocols in edge systems

Containerization and the choice of operating environment are critical in designing edge solutions, as they affect platform stability, scalability, and maintainability. Edge systems typically run on resource-constrained devices, requiring isolated applications with centralized control and secure updates. Lightweight operating systems like Yocto Linux, Ubuntu Core, and BalenaOS, combined with container tools such as Docker and containerd, support efficient, portable deployments. Orchestration solutions like K3s, MicroK8s, and Portainer enable low-overhead, autonomous edge operations. While containerization ensures modularity and application isolation, it does not fully address lifecycle management, configuration, or system integration. Therefore, an additional runtime layer is needed to handle I/O interaction, event processing, synchronization with central systems, and secure device management.

For example, **Azure IoT Edge Runtime** is a modular runtime environment that enables running Docker containers directly on edge devices. It is integrated with Azure IoT Hub and supports the deployment of both custom logic and built-in modules, including a message router (Edge Hub), modules for streaming analytics, and machine learning models based on ONNX or TensorFlow. Edge Runtime ensures automatic module updates, configuration management, and bidirectional synchronization with the cloud.

Another example is **AWS Greengrass Core**, which offers local execution of AWS Lambda functions, stream processing support, data caching, and offline operation with subsequent synchronization. Within the Greengrass Group, edge devices can exchange messages via a local bus, use MQTT, and perform data transformation before sending it to AWS IoT Core. Special attention is given to security, utilizing X.509 authentication, function isolation, and IAM policies.

The messaging between components of the edge architecture plays a crucial role in **ensuring data consistency**, service manageability, and system resilience in conditions of unstable networks [5]. In the edge environment, priority is given to lightweight and event-driven protocols that can operate with intermittent connectivity and support QoS mechanisms (table 2).

Table 2

Comparison of message exchange protocols in edge architectures [6, 7]

Protocol	Core model	QoS / reliability	Typical applications
MQTT	Publish/subscribe	Three-level QoS (0, 1, 2); resilient to failures.	IoT devices, sensors, edge gateways.
OPC-UA	Client/server + pub/sub	Session support, addressability, built-in PKI.	SCADA, industrial controllers, MES.
REST	Request/response	No QoS, sensitive to latency and disruptions.	API calls, cloud synchronization.
AMQP	Message queue	Delivery confirmation, queues, reliable connection.	Telemetry, corporate data buses.

The choice of data transfer protocol in the edge architecture is determined not only by the characteristics of the network infrastructure but also by the requirements for delivery reliability, the volume of transmitted information, and the type of interacting components. In addition to ensuring reliable data delivery, edge platforms must have the ability to perform local analytics and autonomously respond to events – this requires the inclusion of data processing modules and support for ML inference at the edge.

Event processing on the edge node includes data preprocessing (aggregation, filtering, metric calculation), implementation of business rules, and, if necessary, local decision-making without cloud interaction. Containerized microservices, streaming analytics modules (such as Azure Stream Analytics, Apache Edgent), and ML inference using optimized engines – ONNX Runtime, TensorFlow Lite, OpenVINO – are used to execute models on CPU, GPU, and VPU. In scenarios with high event density, CEP frameworks such as Esper or Apache Flink (in an edge configuration) are employed to detect correlations and respond to complex sequences. When network connectivity is restored, data is synchronized with the central platform through REST, gRPC, or message brokers, considering QoS and retransmission logic, ensuring data consistency and integrity across the system.

In distributed architectures with autonomous edge nodes, security and access control are essential [8]. Communication is secured using mTLS and X.509 certificates, ensuring mutual authentication and data protection. Device identity and onboarding are managed via hardware-based TPM and services like Azure DPS or AWS IoT Device Defender, enabling centralized certificate and policy control. Access to services and data is regulated through RBAC, with permissions set at node, container, or API levels. Additionally, context-aware policies (e.g., geolocation, time, device type) enhance flexible and reliable access control in dynamic, distributed edge environments.

Thus, the technology stack of edge systems covers all levels – from the operating system and container environment to high-level analytics and access control – providing a fully functional and secure implementation of edge computing.

Advantages of using edge computing

Integrating edge computing into production chains significantly reduces information processing delays, increases the autonomy of control nodes, and ensures resilience to network infrastructure failures. This is especially crucial for highly automated enterprises. For example, **Amazon** uses edge computing in its warehouse operations for real-time data processing and inventory management. The company has implemented real-time tracking systems, enabling quick responses to demand changes and optimizing stock levels, thus reducing stockouts and improving inventory accuracy. Additionally, edge computing processes data directly at warehouses, speeding up decision-making and enhancing operational efficiency.

In the aerospace industry, companies also implement edge computing to optimize production and inventory management. For instance, **Boeing** uses edge computing to monitor equipment status and streamline manufacturing processes. This technology improves system autonomy, reduces dependence on cloud services, optimizes inventory management, and enables faster reactions to production changes, all critical for aerospace's high precision and reliability demands.

In aerospace edge infrastructure also provides a reliable platform for local monitoring. For example, an edge node on the assembly line processes data from RFID tags at each workstation. If the assembly sequence is incorrect or a required part is missing, the system can halt the process or alert the central quality system.

Overall, managing inventory at the workshop and warehouse levels is a key application of the edge approach. Using edge devices with connected sensors (RFID, IoT tags, scales, cameras), material movement, status, and availability can be tracked without relying on the cloud.

The integration of edge layers with industrial information systems occurs through API and industrial protocols. ERP systems (e.g., SAP, 1C: ERP), APS (Advanced Planning & Scheduling), and SCADA interact with edge nodes via REST/gRPC interfaces or message brokers (MQTT, AMQP). In practice, edge devices transmit aggregated metadata, preprocessing results, or event signals. SCADA systems use this data for visualization and real-time control, while ERP systems make strategic decisions, including procurement and logistics. Combining edge with APS, for example, allows dynamic production schedule adjustments based on real equipment status and local warehouse material availability.

Thus, using edge computing in production chains not only reduces the load on central resources and improves operational resilience but also enables flexible, locally adapted management scenarios for production, inventory, and quality in the context of digital industrial enterprises.

Conclusion

The evolution of web architectures demonstrates a systemic shift from centralized, rigid solutions to distributed, cloud-based, and ultimately edge-oriented models capable of effectively addressing the challenges of modern industrial digitalization. Integrating edge computing into the information layers of production, inventory, and supply chain management enables not only reduced data processing latency and increased network resilience, but also the implementation of localized analytics and autonomous decision-making. The architectural principles of edge infrastructures – combining containerization, lightweight communication protocols, and secure interaction with enterprise systems – form the technical foundation for high-tech industries, where process continuity and operational precision are critical. Thus, the transition to edge models represents not merely a technological evolution, but a paradigm shift in the design of industrial IT systems.

References

1. Shethiya A.S. Building Scalable and Secure Web Applications Using. NET and Microservices // Academia Nexus Journal. 2025. Vol. 4. No. 1.

2. Asghari A., Sohrabi M.K. Server placement in mobile cloud computing: A comprehensive survey for edge computing, fog computing and cloudlet // Computer Science Review. 2024. Vol. 51. P. 100616.
3. Umarov A. Using technologies to optimize logistics and sales // Universum: economics and jurisprudence: electron. scientific journal. 2025. No. 2(124). P. 51-55.
4. Kumar A., Singh D. Securing IoT devices in edge computing through reinforcement learning // Computers & Security. 2025. Vol. 155. P. 104474.
5. Bolgov S. Automation of business processes using integration platforms and backend technologies // International Research Journal of Modernization in Engineering Technology and Science. 2024. Vol. 6(12). P. 3847-3851.
6. Swain M., Tripathi N., Sethi K. Identifying communication sequence anomalies to detect DoS attacks against MQTT // Computers & Security. 2025. P. 104526.
7. Garifullin R. Development and implementation of high-speed frontend architectures for complex enterprise systems // Cold Science. 2024. No. 12. P. 56-63
8. Bhattacharya T., Peddi A.V., Ponaganti S., Veeramalla S.T. A survey on various security protocols of edge computing // The Journal of Supercomputing. 2025. Vol. 81. No. 1. P. 310.