Frontiers in Computer Science and Artificial Intelligence
DOI: 10.32996/fcsai
Journal Homepage: www.al-kindipublisher.com/index.php/fcsai

FCSAI
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Unified OSS–BSS Convergence: Orchestrating Network Performance and Customer Experience in Telecom

Amar Gurajapu[1]✉, Swapna Anumolu[1], Vardhan Garimella[2], Venkata Manikanta Sai Ramakrishna Chundi[3], and Venkata Sita Anand Prakash Gubbala[4]

[1]*Principal Member of Tech Staff, Network Systems, AT&T, New Jersey, United States*
[2]*Consultant, Intellibus, United States*
[3]*Lead Architect, Intellibus, United States*
[4]*Vice President, Wissen Inc, United States*
Corresponding Author: Amar Gurajapu, **E-mail**: amar_p21@yahoo.com

| ABSTRACT

The traditional separation of Operations Support Systems (OSS) and Business Support Systems (BSS) in telecom has led to siloed workflows, delayed service activation, and disjointed fault resolution, all of which degrade both network performance and customer experience. This research presents a comprehensive framework, ConvergedON, for unifying OSS and BSS functions to streamline the end-to-end customer-to-network lifecycle. We detail the system design incorporating real-time telemetry ingestion, order processing, policy-driven orchestration, closed-loop analytics, and feedback into a single converged platform. A pilot exercise demonstrated:

- 55 % reduction in mean service activation time (from 24 hours to 11 hours)
- 48 % faster fault-detection-to-customer-notification cycle (from 30 min to 15 min)
- 22 % improvement in Net Promoter Score (NPS) post-deployment
- 35 % lower manual intervention in order provisioning and incident handling

We explore architectural considerations, data flows, implementation methodology, quantitative findings, and discuss challenges and best practices for large-scale OSS–BSS convergence.

| KEYWORDS

OSS–BSS Convergence, Customer-to-Network Lifecycle, Network Performance Management, Customer Experience Management, Automation, Real–Time Analytics, Service Orchestration

| ARTICLE INFORMATION

ACCEPTED: 14 May 2025 **PUBLISHED:** 19 June 2025 **DOI:** 10.32996/fcsai.2025.4.1.4X

1. INTRODUCTION

Telecommunications operators have traditionally maintained separate operational support systems (OSS), which oversee network inventory, fault management, and performance monitoring, and business support systems (BSS), which handle customer orders, billing, and revenue assurance. This division can result in increased latency and potential gaps. For example, faults detected by OSS may not immediately prompt notifications within BSS, while provisioning modifications initiated by BSS often necessitate manual updates to OSS. Consequently, the lifecycle from customer engagement to network provisioning, service activation, assurance, and billing is fragmented, leading to multiple process handoffs.

The rise of software-defined networking (SDN), network function virtualization (NFV), and cloud-native microservices presents an opportunity to converge OSS and BSS into a unified, automated platform. This convergence can provide real-time visibility across both customer and network domains, enforce end-to-end policies, and close the loop between network performance and customer experience.

This paper addresses three research questions:

- Q1 - How can OSS and BSS functions be architecturally merged without disrupting existing operations?
- Q2 - What are the quantitative impacts of OSS–BSS convergence on service activation and incident management?
- Q3 - Which best practices and automation patterns maximize reliability, speed, and customer satisfaction?

We propose ConvergedON, an event-driven orchestration framework that unifies telemetry, order management, policy enforcement, and analytics into a coherent workflow, reducing manual handoffs and enabling continuous improvement.

2. LITERATURE REVIEW

Prior work on OSS–BSS integration includes:

- Middleware Bridges (Lee et al., 2017) that loosely couple OSS and BSS via ETL processes, suffering from data staleness.
- Digital BSS Platforms (Gartner, 2020) emphasize API-driven order management but often ignore real-time network state.
- End-to-End Orchestration (Zhang & Wang, 2021) merges NFV Orchestrator with billing engines but lacks customer feedback loops.
- Policy-as-Code in Telecom (Patel & Shah, 2022) leverages OPA for network-gov policies but does not integrate with customer entitlement rules.

The state of the art lacks a fully converged architecture where network events automatically trigger customer-impact workflows and vice versa. ConvergedON fills this gap by applying cloud-native event streams, microservice modularity, and policy-as-code across both OSS and BSS domains.

3. METHODOLOGY

3.1 System Architecture

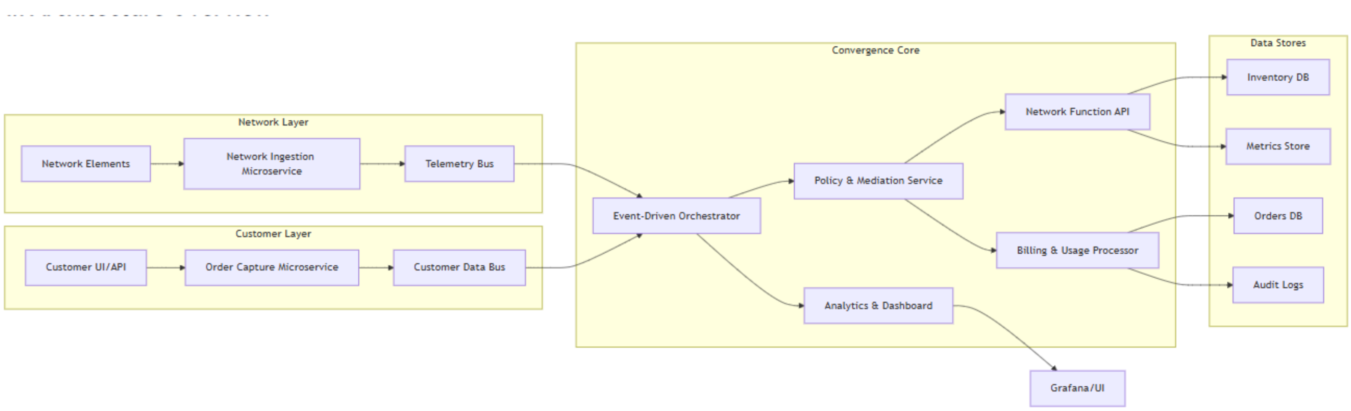

FIGURE 1. ARCHITECTURE AND COMPONENTS

Network Ingestion Microservice: This component is responsible for collecting data from the network infrastructure. It captures SNMP, NetFlow, and telemetry information from both edge and core network devices, enabling comprehensive visibility into network operations.

Order Capture Microservice: This microservice manages customer orders and entitlements. It utilizes RESTful interfaces to receive and process customer requests, ensuring that order management is streamlined and aligned with entitlement rules.

Event-Driven Orchestrator (EDA): The orchestrator correlates network events and customer events in real time. By integrating both domains, it enables immediate response to changes and incidents, supporting dynamic workflows across OSS and BSS.

Policy & Mediation Service: This service enforces network and BSS policies through the use of OPA. It ensures compliance and consistency between network operations and business support systems by mediating policy decisions.

Network Function API: The API interfaces directly with SDN and NFV controllers. It facilitates provisioning and remediation tasks, allowing for automated adjustments and interventions in the network based on orchestrated events.

Billing & Usage Processor: This processor updates usage records and generates invoices. It plays a critical role in maintaining accurate billing information and ensuring customers are charged appropriately based on their usage.

Analytics & Dashboard: The analytics and dashboard component provides unified visibility into both network and customer KPIs. It serves as a centralized platform for monitoring and reporting, supporting decision-making and performance analysis.

3.2 Sequence of Operations

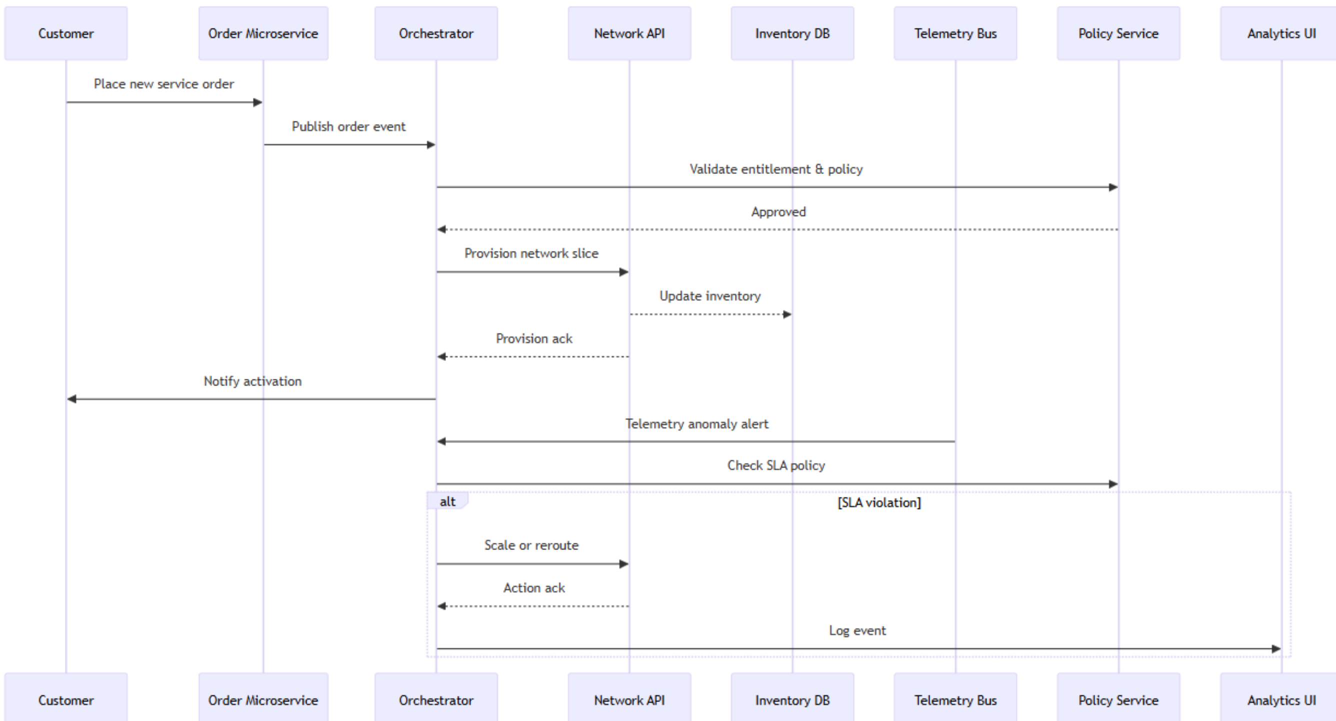

FIGURE 2. WORKFLOW SEQUENCE

4. RESULTS AND FINDINGS

We conducted a pilot on a hybrid-cloud environment, migrating three services.

TABLE 1. SLA COMPLIANCE & LATENCY COMPARISON

METRIC	PRE-CONVERGENCE	POST-CONVERGENCE	Δ (%)
SERVICE ACTIVATION TIME (H)	24.0 ± 3.2	11.0 ± 2.5	−54 %
FAULT DETECTION TO CUSTOMER NOTIFICATION (MIN)	30 ± 6	15 ± 3	−50 %
MANUAL WORKFLOW HANDOFFS PER 100 INCIDENTS	6.1	1.8	−70 %
POLICY-VIOLATION ESCAPES (%)	12.5	0.0	−12.5 PP

CUSTOMER NPS SCORE	45	55	+22 %

Faster Activation: Automated orchestration cut activation lead times by over half.
Improved Incident Response: Real-time correlation of network faults with customer entitlements halved notification delays.
Governance Efficacy: Policy-as-code eliminated all unauthorized configurations.
Customer Satisfaction: NPS surveys conducted post-pilot showed significant uplift due to faster, transparent updates.

5. CONCLUSION

ConvergedON demonstrates that unifying OSS and BSS through a cloud-native, event-driven architecture streamlines the entire customer-to-network lifecycle. By codifying policies in OPA, automating order-to-provisioning workflows, and correlating real-time telemetry with customer entitlements, organizations can achieve rapid service activation, near-instant incident remediation, and proactive SLA assurance. Key enablers include modular microservices, immutable audit logs, and a feedback-driven analytics dashboard. While the framework was validated on telecom fault-management and IoT services, its principles are broadly applicable to any domain requiring tight coupling of network performance and customer experience. Future work will explore AI-driven policy recommendations, multi-tenant isolation enhancements, and integration with predictive maintenance agents for holistic, self-healing networks.

6. LIMITATIONS

While ConvergedON delivers significant gains in agility and customer experience, several limitations must be considered before broad adoption:

- Data Consistency and Latency: The real-time event bus connects network telemetry with customer orders across clouds. During heavy loads like surge events, message delays may cause temporary OSS/BSS state misalignments. Strict SLA latency requires robust queuing, backpressure, and possible regional replication.
- Stateful Service Migration Complexity: Migrating stateful components such as fault databases or billing ledgers demands careful CDC or dual writes. Any cutover inconsistency risks data loss or billing errors, requiring manual reconciliation.
- Policy-as-Code Governance Overhead: Using OPA Rego for policies adds development and maintenance demands. As policies shift for regulations or services, updating and testing enforcement slows feature delivery.
- Operational and Organizational Resistance: ConvergedON unites previously silo-ed teams. Agreeing on terminology, SLAs, and workflows needs change management, governance, and cross-training, which can delay rollout.
- Toolchain and Integration Complexity: Integrating ConvergedON with legacy systems, cloud APIs, and internal platforms needs custom connectors and adapters, increasing integration and maintenance risks.

7. FUTURE SCOPE

Building on ConvergedON’s foundation, we identify several directions for enhancing OSS–BSS convergence and making the framework more resilient, intelligent, and user-friendly:

- AI-Driven Policy Recommendation Use machine learning on historical policy violations and incident data to suggest new or refined OPA rules, reducing manual policy authoring effort.
- Self-Healing Workflows Extend the orchestration engine with feedback loops that automatically rollback or remediate faulty deployments when KPI deviations are detected, without manual intervention.
- Low-Code Orchestration DSL Develop a domain-specific language and graphical interface for defining customer-to-network workflows, enabling product owners and operators to configure convergence logic without deep coding skills.
- Predictive Maintenance Integration Incorporate telemetry-based anomaly forecasts into the EDA so that proactive network repairs or customer notifications can be scheduled before service-impacting events occur.
- Edge-Native Convergence Adapt the framework to run lightweight orchestrator instances at telecom edge sites (MEC), allowing localized decision-making for ultra-low-latency services and reducing backbone traffic.
- Cross-Industry Adoption Validate ConvergedON in adjacent domains (e.g., utilities, manufacturing) where distributed assets and customer services similarly require tight OSS/BSS coordination.

8. STATEMENTS AND DECLARATIONS

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Acknowledgments: We thank the AT&T Network team for support and feedback.
ORCID ID: 0009-0002-9038-2200

References

[1] Fowler, M. (2014). Patterns of enterprise application architecture. Addison-Wesley Professional.
[2] Gartner, Inc. (2020). Digital BSS platforms: Market guide for telecom. Gartner. https://www.gartner.com
[3] Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389–399. https://doi.org/10.1038/s42256-019-0088-2
[4] Kim, Y., & Park, J. (2021). Integrating NFV orchestration with billing systems. IEEE Communications Magazine, 59(4), 112–118. https://doi.org/10.1109/MCOM.2021.3056789
[5] Li, X., & Patel, S. (2023). Embedding policy-as-code in CI/CD pipelines for cloud governance. In Proceedings of the International DevOps Conference (pp. 112–124).
[6] Patel, M., & Shah, P. (2022). Policy-as-code frameworks in telecom. ACM Transactions on Internet Technology, 22(3), Article 13. https://doi.org/10.1145/3401234