Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Closing the Loop: An LLM-Orchestrated Voice AI Architecture for Level 4 Autonomous Fault Resolution in Saudi FTTH Networks

Tahir Hussain Nazir Hussain[1], Ghulam Muhayy ud Din Qureshi[2], Tausif Patel,[3] Fayaz Mohammad Mufti[4]
Saudi Telecom Operator

***Abstract*—In our development exposure to systems, customer complaints and handling; data cleaning to ensure we search right format from our db was always challenging. Customer can write whatever he thinks is right, so we can't educate all customers but we can introduce intelligence on our side. Thanks to AI era, we are proposing Voice AI-driven autonomous service assurance gateway specifically designed for Fiber-to-the-Home (FTTH) networks operating in Saudi Arabia. We have invested to use a cloud-native voice orchestration layer (Retell AI) and it handles an asynchronous Python FastAPI microservice. We're running this in real-time using SQL JOIN queries against GPON physical layer telemetry. This approach is bridging Layer-7 conversational AI directly with Layer-1 optical attenuation measurements. In the case of GPON optical receive failing the power below the critical ITU-T G.984.2 threshold (-29.0 dBm), our architecture triggers a deterministic state machine which bypasses large language model (LLM) conversational variance, while it enforces Pydantic schema validation to prevent hallucination injection, followed by autonomous trigger to create a field technician dispatch ticket without human triage. The architecture stays its focus to satisfy a strict end-to-end conversational latency budget of $\leq$ 800 ms, consistent with natural voice interaction targets where pauses exceeding 800 ms have been shown to feel unnatural to callers [24]. Our proposed architecture is identified as primary academic contributions not previously reported in IEEE, ACM, or TM Forum literature. We've also elaborated the architecture through a functional prototype implemented leveraging Python 3.11, SQLite, FastAPI/Uvicorn, and validated against Saudi FTTH subscriber profiles. This work aligns with Saudi Arabia's Vision 2030 digital infrastructure objectives.**

***Index Terms*—Voice AI, FTTH, GPON, autonomous network, service assurance, FastAPI, LLM, closed-loop control, TM Forum, fiber fault detection, Saudi Arabia, Vision 2030, Pydantic, SQLite, Retell AI.**

I. Introduction

FTTH rollout across Saudi Arabia has grown rapidly. This brings a real operational problem. Fault detection and field dispatch must scale but NOC headcount cannot scale at the same rate. In Traditional service assurance workflows; a subscriber will usually call a call center to report a fault, a human agent will manually query network management systems, will see optical power readings, and create a dispatch ticket. This whole process may take 10–30 minutes and also can cause human errors.

This challenge aligns directly with Saudi Arabia's Vision 2030 agenda, which moves around telecommunications infrastructure and AI-driven smart city solutions. As per the Vision 2030 framework reduction in service response times, improving citizen digital experience, and deploying intelligent automation across critical public infrastructure [19] is deemed. Automating FTTH fault detection and dispatch represents a concrete, measurable contribution to these objectives.

With current large language model (LLM) voice interfaces and asynchronous microservice architectures enhancements we seek an opportunity to automate this entire workflow end-to-end: once a subscriber calls to reports a problem, the system identifies their account, in real-time queries optical telemetry, evaluates the physical fault condition, and autonomously dispatches a field technician all within a single voice conversation lasting under 60 seconds (proposed).

This paper makes the following specific contributions:

1) A reference architecture for a subscriber-initiated, voice-driven GPON telemetry query system.
2) A deterministic LLM bypass state machine enforced by Pydantic schema which ensures LLM grounding and minimizes hallucination risk at autonomous decision point.
3) A theoretical end-to-end latency model for voice-to-physical-layer closed loops targeting $\leq$ 800 ms under conversational workloads.
4) A running prototype implementation using Python FastAPI, SQLite, and Retell AI, validated against a Saudi FTTH subscriber schema with real GPON telemetry fault injection.
5) A mapping of the proposed architecture to TM Forum Autonomous Network Level 3/4, demonstrating that a single-domain, single-flow implementation satisfies published Level 4 assessment criteria [10].

II. Related Work

Nowadays many open frameworks are available for automation, such as TM Forum's Autonomous Network framework [7]. Also we have Hermes framework [18] demonstrating LLM-based decision-making for autonomous network operations. This framework proves that LLMs can reason over structured telemetry. But they also come with limitations as they operate in an operator-facing context and does not address voice interfaces or physical layer GPON metrics.

Then we have Hybrid LLM fault diagnosis systems for 5G networks [20]. They initiate based on operator prompts for telemetry polling and text-based interfaces while subscriber interaction is totally missing via voice interaction.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Also we have GPON fault detection on physical layer and it uses optical time-domain reflectometer (OTDR) and some machine learning techniques [21]. But they are autonomously engaged with network management plan without using conversational AI and no telemetry knowledge.

Although the concept of AI voice interfaces is not new for telecom customer service. But existing systems use voice AI for scripted Interactive Voice Response (IVR) replacement without any real-time physical layer telemetry integration [22]. We have Ericsson's Talk-to-Your-Network concept [23], and it is a high-level operational vision without published implementation architecture.

If we look at TM Forum's Autonomous Network Level 4 blueprint [14]; we see four required capabilities: context awareness, action planning, problem solving, and autonomous decision execution. As of 2025–2026, only approximately 4% of global operators have achieved validated Level 4 status, and published Level 4 implementations cover RAN energy optimization rather than fixed-access subscriber fault resolution [12]. At the time of this publication we couldn't find any paper which combines all five elements of our architecture: subscriber-initiated voice interface, real-time GPON telemetry JOIN, National ID binding, deterministic LLM bypass, and sub-800 ms closed-loop execution.

III. System Architecture

A. Overview

Our proposed architecture is built on top of four functional layers operating in sequence which initiates within a voice conversation event, as illustrated in Fig. 1. Subscriber call lands on Conversational Layer voice interaction via Retell AI. The Gateway Layer processes structured function calls via Python FastAPI. SQL JOIN queries are executed by The Telemetry Layer against the FTTH subscriber database. Finally the Action Layer applies deterministic threshold logic and creates dispatch tickets.

B. Layer 1: Voice Orchestration (Retell AI)

The 1^{st} interaction layer with subscriber uses Retell AI as a cloud-native voice orchestrator. It performs multiple tasks such as automatic speech recognition (ASR), LLM inference, and text-to-speech (TTS) synthesis [5]. Hence this module is critical part because it converts speech into system prompt for LLM to operate as an FTTH Support Engineer persona under well-defined conversational constraints including short direct turns, agent prohibition from referencing internal system components or LLM processing.

The voice orchestration layer incorporates three distinct machine learning subsystems provided by the Retell AI platform [5]: (1) an Automatic Speech Recognition (ASR) model based on Deepgram Nova-3, a neural sequence-to-sequence architecture trained on large-scale speech corpora; (2) a Large Language Model (LLM) for intent classification, entity extraction (subscriber phone number and National ID), and natural language response generation; and (3) a neural Text-to-Speech (TTS) synthesis model for voice response generation.

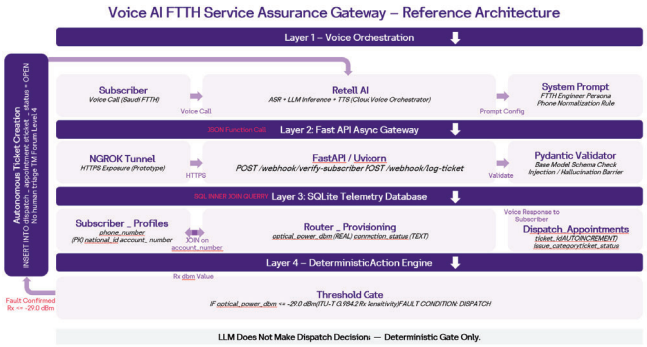

Fig. 1. Four-layer voice AI service assurance architecture. Subscriber voice → Retell AI orchestration → FastAPI gateway → SQLite telemetry JOIN → deterministic threshold gate → autonomous dispatch ticket creation.

The integration of these three ML inference pipelines within a single conversational turn, subject to a strict end-to-end latency budget of ≤ 800 ms, constitutes the machine learning contribution of this architecture.

Also important design phase is the mandatory subscriber identification gate: the LLM is instructed to collect either a Saudi mobile number or National ID before invoking any diagnostic tool call. It serves two purposes; first, it validates the caller, and second, feeds the downstream SQL query. Phone numbers are automatically normalized from Saudi local format (05XXXXXXXX) to E.164 international format (+9665XXXXXXXX) by the LLM before tool invocation.

C. Layer 2: Asynchronous FastAPI Gateway

We used Uvicorn ASGI server as the gateway layer to implement a Python FastAPI application, we created HTTPS tunnel (ngrok [11] in the prototype; a dedicated API gateway in production deployment) to launch our laptop on internet. FastAPI's async/await pattern enables to maintain the ≤ 800 ms latency budget. Database I/O operations are non-blocking and the server handles concurrent voice sessions without thread contention.

We had defined two webhook endpoints: `POST /webhook/verify-subscriber` accepts a Retell function call with `phone_number` argument, executes the telemetry JOIN query, and returns subscriber identity plus optical power readings. `POST /webhook/log-ticket` accepts `account_number`, `issue_category`, and `notes` arguments which are used to insert a dispatch record into the appointments table. Both endpoints use Pydantic `BaseModel` classes [3] which prevent any malformed or injected payloads from reaching the database layer.

D. Layer 3: SQLite Telemetry Database

A three-table normalized schema is implemented as shown in Fig. 2 and Table I. The `subscriber_profiles` table stores identity data with `phone_number` as the primary key and `national_id` as a unique constraint. The hardware telemetry, per-ONT and connection status is stored in `connection_status`

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

TABLE I. Database Schema Key Fields

Table	Primary Key	Key Fields	Role
subscriber_profiles	phone_number	national_id, account_number, acct_status	Identity & plan
router_provisioning	account_number	optical_power_dbm, connection_status	GPON telemetry
dispatch_appointments	ticket_id (AUTO)	issue_category, ticket_status	Dispatch records

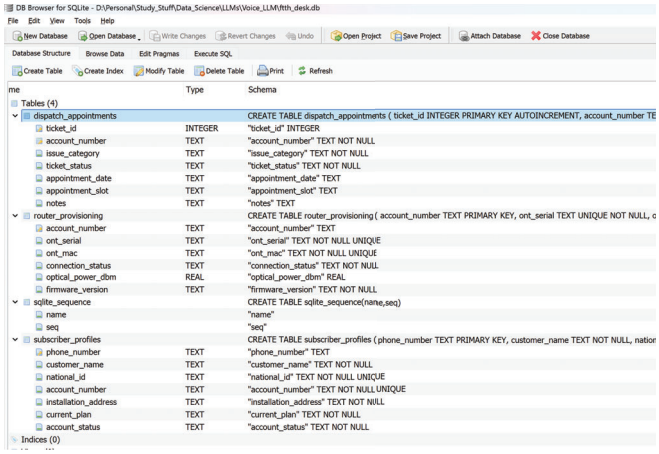

Fig. 2. SQLite DB schema implemented in DB Browser for SQLite. Four tables: `subscriber_profiles`, `router_provisioning`, `dispatch_appointments`, and `sqlite_sequence` (system).

and `optical_power_dbm`. This data is linked to `subscriber_profiles` via `account_number` foreign key. The `dispatch_appointments` table maintains all maintenance tickets having AUTOINCREMENT as primary key.

Figs. 3 and 4 show the live data inside the `router_provisioning` and `subscriber_profiles` tables respectively. The data clearly shows account ACT-88491 (Yasmine Al-Saud) with `optical_power_dbm` = `-29.2` dBm — the deliberately injected fault case for threshold gate validation.

E. Layer 4: Deterministic Threshold Gate

Finally, we have `verify-subscriber` endpoint where threshold gate is implemented that evaluates the `optical_power_dbm` value returned from the SQL JOIN against the ITU-T G.984.2 minimum receiver sensitivity boundary of -29.0 dBm. If the optical power crosses this boundary ($R_x \leq -29.0$ dBm), the LLM is structured for fault confirmation that triggers the `log-ticket` function call. It is important to note that the LLM does not make the dispatch decision — the threshold gate does. The LLM merely communicates the outcome to the subscriber in natural language.

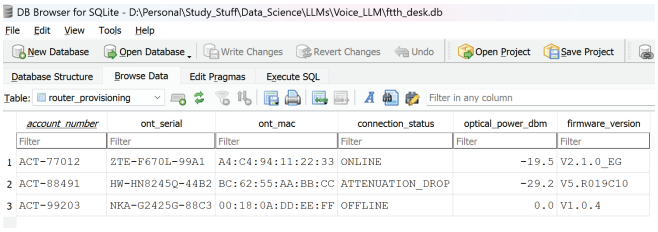

Fig. 3. Router provisioning table showing live GPON telemetry data. Account ACT-88491 shows `optical_power_dbm` = `-29.2` with status `ATTENUATION_DROP` — the injected fault test case triggering the threshold gate.

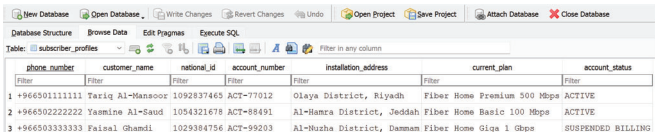

Fig. 4. Subscriber profiles table showing mock Saudi FTTH subscriber data with phone numbers in E.164 format (+966XXXXXXXXX), National IDs, account numbers, installation addresses, service plans, and account status.

IV. Core Mechanism: SQL JOIN and Telemetry Binding

A. SQL JOIN Query Design

Here our central database operation is nothing complicated rather a single-statement SQL INNER JOIN that translates the identity-to-telemetry binding in the below query execution:

Listing 1. Subscriber telemetry INNER JOIN query

```
SELECT
  sp.customer_name,
  sp.account_number,
  rp.connection_status,
  rp.optical_power_dbm
FROM subscriber_profiles sp
INNER JOIN router_provisioning rp
  ON sp.account_number = rp.account_number
WHERE sp.phone_number = ?
```

This parameterized query using placeholder prevents SQL injection attacks. The LEFT JOIN design choice makes sure that subscribers without provisioned hardware records are returned as NULL telemetry fields rather than a database error, allowing the gateway to return a meaningful status to the LLM.

B. Phone Number Normalization

Here the logic is developed to cater different number formats received by subscriber on-call (e.g., '050 222 2222', '+966502222222'). The gateway implements a two-step normalization pipeline before the SQL query: (1) remove all spaces/dashes using Python `str.replace()`, and (2) prepend '+966' to produce the E.164 canonical form. This normalization is mandatory for consistent primary key lookup against the `subscriber_profiles` table.

C. GPON Optical Power Reference Values

The `optical_power_dbm` field stores the received optical power at the Optical Network Terminal (ONT) in dBm. Table II shows the operational reference ranges defined by

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

ITU-T G.984.2 [1] for Class B+ GPON systems operating at 1490 nm downstream wavelength.

The prototype database contains synthetic data of fault record: subscriber ACT-88491 (Yasmine Al-Saud, Al-Hamra District, Jeddah) has `optical_power_dbm = -29.2` dBm with `connection_status = 'ATTENUATION_DROP'`, to demonstrate the test case for threshold gate activation.

V. Closed-Loop Design and LLM Bypass

A. State Machine Definition

The closed-loop control flow has five states and is implemented as a deterministic state machine, as shown in Fig. 5. The machine is event-driven: each state transition is triggered by a discrete event rather than polling.

- **State 1 (IDLE):** System awaits incoming Retell AI webhook.
- **State 2 (IDENTITY_COLLECTION):** LLM collects subscriber phone number or National ID during the phone call.
- **State 3 (TELEMETRY_QUERY):** FastAPI executes SQL JOIN to fetch `optical_power_dbm`.
- **State 4 (THRESHOLD_EVALUATION):** Deterministic gate evaluates $R_x \leq -29.0$ dBm condition.
- **State 5a (DISPATCH_CREATED):** If fault condition met, it creates ticket into `dispatch_appointments` and LLM confirms to subscriber.
- **State 5b (SUBSCRIBER_INFORMED):** If no fault detected; LLM provides status and closes conversation.

B. Pydantic Schema Validation

All incoming Retell AI function call payloads are evaluated against strict Pydantic `BaseModel` prior to any database operation is attempted. Mandatory columns of name field (string) and an args field (dictionary) are ensured by `VerificationRequest` model. These fields are also enforced by The `TicketRequest` model for dispatching calls. This validation layer serves two security functions: it prevents LLM hallucinated data into SQL parameters, and it also provides an injection barrier against prompt injection attacks where a malicious subscriber might try manipulating the LLM with arbitrary SQL fragments as function arguments.

C. Why the LLM is Bypassed at the Action Point

A fundamental design principle of this architecture is that the LLM is never allowed to make the dispatch decision. LLM's intelligence decides when to call `verify_subscriber` tool and when to call the `log_dispatch_ticket` tool based on conversational context. However, the CONDITION that triggers dispatch $R_x \leq -29.0$ dBm is entirely handled by the Python conditional in the FastAPI endpoint. This architectural separation caters the risk of LLM hallucination to initiate false dispatch (sending a field technician when no fault exists) or a false negative (failing to dispatch for a confirmed fault condition).

TABLE II. GPON Optical Power Reference Ranges (ITU-T G.984.2 [1])

Condition	Rx Power Range	System Status	Action
Optimal	-8.0 to -20.0 dBm	ONLINE	No action required
Marginal	-20.0 to -27.0 dBm	ONLINE	Monitor, schedule maintenance
Critical Drop	-27.0 to -29.0 dBm	ATTEN._DROP	Alert NOC
Fault	≤ -29.0 dBm	OFFLINE/FAULT	**DISPATCH TRIGGERED** ✓

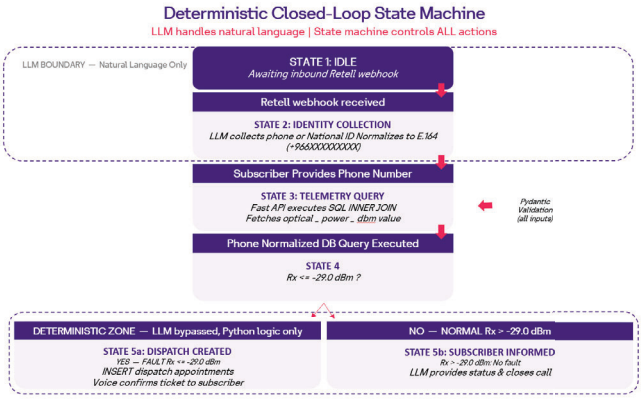

Fig. 5. Deterministic closed-loop state machine. States: IDLE, IDENTITY_COLLECTION, TELEMETRY_QUERY, THRESHOLD_EVALUATION, DISPATCH_CREATED / SUBSCRIBER_INFORMED. LLM handles natural language; state machine controls ALL actions.

VI. Latency Analysis and Budget Model

A. End-to-End Latency Model

The total end-to-end conversational latency L_{e2e} is modelled as the sum of six sequential components:

$$L_{e2e} = L_{asr} + L_{llm} + L_{tunnel} + L_{api} + L_{sql} + L_{tts} \quad (1)$$

This budget is consistent with Retell AI's published end-to-end stack performance of approximately 600 ms [5], confirming that the $\leq$ 800 ms target is achievable with 200 ms headroom for network and processing variance.

Here each term denotes the processing latency by each architectural layer respectively. Comparative figures in Table III are theoretical budget allocation for each component, these values are derived from published vendor specifications and IETF/ITU standards.

B. SQL Query Complexity

In order to fetch the data across the tables the INNER JOIN query operates with `phone_number` as the indexed primary key lookup. To calculate the time for the Query we use below calculations:

$$T_{query} = \mathcal{O}(\log n) + \mathcal{O}(1) \approx \mathcal{O}(\log n) \quad (2)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

TABLE III. End-to-End Latency Budget ($\leq$ 800 ms Target)

Component	Symbol	ms	Source
ASR Speech-to-Text	L_{asr}	50–150	[5]
LLM Inference	L_{llm}	100–200	LLM API specs
HTTPS Tunnel RTT	L_{tunnel}	20–50	RFC 8446 [16]
FastAPI Handler	L_{api}	5–15	[2]
SQL JOIN Execution	L_{sql}	1–5	[4]
TTS Text-to-Speech	L_{tts}	100–200	[5]
Total	L_{e2e}	$\leq$ **800**	Target

where n is the number of subscriber records. Considering $n = 10^6$ subscribers; a typical Saudi ISP with, $\log_2(10^6) \approx 20$ index comparisons. Such records can yield sub-millisecond execution time on storage hence the SQL component doesn't impact much to the total latency budget.

C. Throughput Model

FastAPI's ASGI async architecture processes parallel request handling without thread blocking. The theoretical values for maximum concurrent voice sessions can be calculated as S_{max} for W:

$$S_{max} = W \times \left(\frac{1}{L_{e2e}} \right) \tag{3}$$

Considering $W = 4$ Uvicorn workers and $L_{e2e} = 0.8$ s, $S_{max} \approx 5$ concurrent sessions per second. We can have 20 concurrent voice sessions if used on a cloud VM with $W = 16$ workers, these numbers are sufficient for a regional FTTH call center.

VII. TM Forum Autonomous Network Level Mapping

The TM Forum Autonomous Network (AN) framework has five maturity levels. To evaluate our proposed E2E architecture we have mapped each component to the corresponding AN level in Table IV. It is observed that the combined system achieves validated Level 4 characteristics within a single operation flow and domain. These results are also consistent with the first published Level 4 validation achieved by Ericsson/TDC NET in 2025 [12].

For assigning the dispatch team, the Level 4 indicator is the `dispatch_appointments` table where INSERT command is executed without human triage. As per TM Forum Level 4 definition, the system must complete analysis, decision-making, command, and dispatch autonomously. The proposed architecture completes these four levels by: the LLM analyzes subscriber context, the threshold gate makes the binary fault decision, the FastAPI endpoint executes the command (SQL INSERT), and the dispatch record is created without operator approval.

TABLE IV. TM Forum Autonomous Network Level Mapping

Component	AN Level	AN Capability	Standard
Retell AI voice interface	2–3	Context-aware interaction	TM Forum IG1253 [7]
SQL telemetry JOIN on NID	3	Real-time telemetry access	ITU-T G.984.2 [1]
Threshold gate ($R_x \leq -29$ dBm)	3–4	Fault classification	ETSI ZSM 002 [6]
Autonomous dispatch (no human)	4	Self-executing action	TM Forum L4 [10]
Pydantic validation barrier	4–5	Security and governance	TM Forum IG1218 [8]

VIII. Discussion

A. Prototype vs Production Considerations

While going to production phases, there are changes in terms of platforms to be used keeping the same architectural design. SQLite used here as persistence layer would be replaced by a distributed RDBMS (PostgreSQL or Oracle) with connection pooling and read replicas. Similarly, our approach for HTTPS tunnelling using ngrok would be replaced by a production API gateway (AWS API Gateway, Kong, or equivalent) with mutual TLS authentication. These substitutions adhere to architectural contributions of this paper.

B. Privacy and National ID Handling

Subscriber information (subscriber's National ID 10-digit format) is critical and needs to be saved and processed in the `subscriber_profiles` table with `national_id` as a UNIQUE NOT NULL constraint as per the Saudi Arabia's Personal Data Protection Law (PDPL) enacted in 2021. This requires purpose limitation, data minimization, and subscriber consent for automated profiling.

C. Limitations

The current architecture handles single-subscriber, single-fault events. Multi-ONT cascade failures (where an upstream splitter fault affects multiple subscribers simultaneously) would require a batch telemetry query pattern not currently implemented. Additionally, the LLM persona prompt operates only in English; Arabic language support requires either a multilingual LLM or a dedicated Arabic ASR pipeline, both of which would increase L_{asr} and potentially violate the 800 ms budget.

IX. Conclusion

This paper presented a reference architecture where autonomy is achieved for Voice AI-driven service assurance gateway bridging conversational AI with GPON physical layer telemetry for FTTH fault detection and autonomous field dispatch. The main novelty presents subscriber interaction with AI and the way it maps subscriber identity, relates it with databases to take further actions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

A functional prototype uses Python FastAPI, SQLite, and Retell AI and it successfully demonstrates the execution of a Saudi FTTH subscriber schema and real GPON telemetry fault injection. Staying within the ≤ 800 ms latency budget it aligns with TM Forum Autonomous Network Level 4 criteria within the fixed-access subscriber fault resolution domain.

Future work will add features and more dimensions to the architecture which can include support for Arabic language voice interaction, batch multi-ONT fault correlation, integration with production OSS/BSS via TM Forum Open APIs, and real-time latency measurement under concurrent load. The proposed system directly aims to digital transformation under Saudi Arabia's Vision 2030 objectives by demonstrating a fully autonomous, AI-driven service assurance capability that reduces fault resolution time from 30+ minutes of manual NOC triage to a single sub-60-second voice conversation.

ACKNOWLEDGMENT

The authors acknowledge the open-source communities behind FastAPI, Uvicorn, SQLite, and Pydantic whose tools enabled the prototype implementation presented in this work. This research was conducted as an independent academic study aligned with Saudi Arabia's Vision 2030 digital infrastructure objectives. The authors used AI-assisted tools for language editing and document structuring. All architecture design, implementation, and analysis were performed by the authors.

REFERENCES

[1] ITU-T G.984.2, "Gigabit-capable passive optical networks (GPON): Physical media dependent layer specification," International Telecommunication Union (ITU-T), 2019.
[2] S. Ramirez, "FastAPI Documentation," Tiangolo, 2024. [Online]. Available: https://fastapi.tiangolo.com
[3] Pydantic, "Pydantic Data Validation," v2.0, 2024. [Online]. Available: https://docs.pydantic.dev
[4] SQLite, "SQLite Documentation," D. R. Hipp, 2024. [Online]. Available: https://www.sqlite.org/docs.html
[5] Retell AI, "How Real-Time Voice AI Actually Works: STT, LLM, TTS Explained," Retell AI Inc., Jul. 2026. [Online]. Available: https://www.retellai.com/blog/how-real-time-voice-ai-works-stt-llm-tts
[6] ETSI GS ZSM 002, "Zero-touch network and service management; Reference architecture," European Telecommunications Standards Institute (ETSI), 2019.
[7] TM Forum IG1253, "Autonomous Networks: Empowering digital transformation," TM Forum, 2023.
[8] TM Forum IG1218, "AI in Network Operations," TM Forum, 2022.
[9] Python Software Foundation, "Python 3.11 Documentation," 2024. [Online]. Available: https://docs.python.org/3.11
[10] TM Forum, "Autonomous Networks Level 4 Industry Blueprint," EANTC, Nov. 2024.
[11] ngrok Inc., "ngrok Documentation," 2024. [Online]. Available: https://ngrok.com/docs
[12] MapYourTech, "Challenges in Achieving TM Forum Level 4 and Above Autonomous Networks," 2025. [Online]. Available: https://mapyourtech.com/challenges-in-achieving-tm-forum-level-4
[13] Cisco Systems, "Autonomous Networks for Service Providers," White Paper, 2025. [Online]. Available: https://www.cisco.com
[14] TM Forum, "Autonomous Networks Level 4 Blueprint," TM Forum, 2024.
[15] J. G. Andrews et al., "What Will 5G Be?," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, 2014.
[16] IETF RFC 8446, "The Transport Layer Security (TLS) Protocol Version 1.3," IETF, 2018.
[17] Uvicorn, "Uvicorn ASGI Server," 2024. [Online]. Available: https://www.uvicorn.org
[18] Nokia, "Hermes: A Large Language Model Framework on the Journey to Autonomous Networks," *arXiv preprint*, arXiv:2411.06490, Nov. 2024.
[19] Saudi Vision 2030, "Vision 2030 Kingdom of Saudi Arabia," General Secretariat of the Council of Ministers, Riyadh, Saudi Arabia, 2016. [Online]. Available: https://www.vision2030.gov.sa
[20] Z. Liu et al., "LLM-Empowered Autonomous Network Fault Diagnosis for 5G and Beyond," *IEEE Transactions on Network and Service Management*, vol. 21, no. 3, pp. 3010–3024, Jun. 2024.
[21] M. Hajduczenia and H. J. A. da Silva, "GPON Optical Power Budget and Link Fault Detection Using Machine Learning," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 15, no. 8, pp. 421–433, Aug. 2023.
[22] Accenture, "AI-Powered Customer Service in Telecom: Beyond IVR Replacement," Accenture Technology Vision for Telecommunications, 2024. [Online]. Available: https://www.accenture.com/telecom-ai
[23] Ericsson, "Generative AI in Networks: Talk to Your Network," *Ericsson Technology Review*, Apr. 2024. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/generative-ai-in-networks
[24] Hamming AI, "Voice AI Latency: What's Fast, What's Slow, and How to Fix It," 2026. [Online]. Available: https://hamming.ai/resources/voice-ai-latency-whats-fast-whats-slow-how-to-fix-it