Global Academic Journal of Economics and Business

Available online at https://www.gajrc.com
DOI: https://doi.org/10.36348/gajeb.2026.v08i04.039

OPEN ACCESS

ISSN: 2706-9001 (P)
ISSN: 2707-2584 (O)

Original Research Article

A RAG and Agentic AI Architecture for Automated Fixed-Access Network Investigation

Rizwan Farooq Khan[1]*
[1]Independent Researcher

***Corresponding Author**
Rizwan Farooq Khan
Independent Researcher

Article History
Received: 17.06.2026
Accepted: 11.08.2026
Published: 12.08.2026

Abstract: Fixed-access network investigation remains labour-intensive because faults are expressed across heterogeneous evidence: customer reports, topology, alarms, optical measurements, device telemetry, configuration histories, tickets, software releases and operational knowledge. This review examines how retrieval-augmented generation (RAG) and agentic artificial intelligence can be combined to automate investigation without allowing a language model to substitute plausible narrative for verified engineering evidence. The study synthesises research published from 2020 to 2025 on retrieval, tool-using language agents, root-cause analysis, optical-access diagnostics, intent-based management and zero-touch operations, together with relevant fixed-access standards. It proposes an evidence-governed architecture in which a planning agent decomposes an incident, permission-aware retrievers assemble time-bounded context, specialised tools execute deterministic queries, a causal reasoning layer ranks hypotheses, and an independent verifier checks numerical results, provenance and policy compliance. The review argues that the most defensible role for generative models is orchestration and explanation, whereas calculations, state inspection and change execution should remain within typed, auditable tools. A staged workflow is developed for incident framing, evidence alignment, topology reconstruction, hypothesis testing, counterfactual checking, confidence calibration, human approval and organisational learning. Evaluation should therefore measure retrieval coverage, factual and numerical correctness, root-cause ranking, time-to-isolation, unsafe-action rate and operator trust rather than linguistic fluency alone. The resulting framework supports faster and more consistent diagnosis across fibre, copper, residential gateway and disaggregated access environments while preserving human authority over consequential network actions.

Keywords: Retrieval-Augmented Generation, Agentic AI, Fixed-Access Networks, Root-Cause Analysis, Aiops, Network Automation, Optical Access.

1. INTRODUCTION

Large fixed-access networks are difficult to investigate because the service experienced by a subscriber is produced by a chain of physical, logical and operational dependencies. A single symptom such as intermittent broadband may originate in optical distribution loss, an unstable optical network unit, copper impairment, access-node congestion, a misapplied service profile, residential Wi-Fi conditions, authentication failure, software regression or an upstream transport event. Evidence is distributed across systems owned by different teams, recorded with inconsistent clocks and identifiers, and retained at different granularities.

Investigators consequently spend substantial time locating data, reconciling terminology and reconstructing a service path before they can test a technical hypothesis. The operational problem is not merely anomaly detection; it is evidence integration under uncertainty.

RAG offers a useful foundation because it separates the language model's generative capability from an external, updateable body of knowledge. Foundational work showed that retrieval can ground generation in explicit non-parametric memory and provide a route to fresher, more attributable answers [1]. Dense retrieval improved the ability to find semantically relevant passages beyond exact keyword overlap [2], while heterogeneous benchmarks demonstrated that no retriever is uniformly reliable across domains and that robust systems need lexical, dense and re-ranking components [3]. Retrieval-enhanced models can also use very large external memories without encoding every fact in model parameters [4]. Later synthesis distinguishes naive, advanced and modular RAG, emphasising query transformation, hybrid retrieval, filtering, re-ranking and feedback loops [5]. These developments are directly relevant to network operations, where manuals, runbooks, ticket histories and software notes change continuously.

However, document retrieval alone cannot calculate a loss-of-signal rate, compare counters across a maintenance window, trace a subscriber through a changing topology or verify whether a proposed configuration is safe. Agentic methods extend RAG by allowing a model to plan and invoke tools. ReAct interleaves reasoning with actions and observations [6]. Toolformer demonstrates that language models can learn when and how to call external interfaces [7], and Gorilla shows that retrieval over changing API documentation can improve tool selection [8]. Self-RAG and corrective retrieval introduce mechanisms for deciding when retrieval is needed and for reacting when evidence quality is poor [9, 10]. These capabilities suggest an architecture in which the model coordinates deterministic network functions rather than inventing operational results.

Agency also increases risk. An agent may select the wrong customer, use stale topology, confuse event time with ingestion time, execute an over-broad query, accept malicious instructions embedded in a retrieved ticket, or present an unsupported hypothesis with excessive confidence. RAG evaluation research therefore separates retrieval quality, faithfulness and answer quality [11], while dedicated benchmarks show that models remain vulnerable to distracting, contradictory and counterfactual context [12]. Reflective and iterative methods can improve outputs [13, 14], but self-critique is not equivalent to independent verification. For fixed-access operations, the architecture must enforce permissions, typed tool schemas, evidence provenance, execution limits and human approval.

This review has one aim: to develop a defensible RAG and agentic AI architecture for automated fixed-access network investigation. Its objectives are to identify the evidence and reasoning requirements of investigation; compare RAG, causal analysis and tool-use patterns; define a reference architecture and workflow; establish safety and governance controls; and propose evaluation criteria suitable for operational deployment. The contribution is a synthesis that connects recent language-agent research with network root-cause analysis, optical-access diagnostics and fixed-access management standards, while maintaining a clear boundary between analytical assistance and autonomous control.

2. Fixed-Access Investigation as an Evidence Problem

A credible investigation begins with a service object, time interval and severity definition rather than an unconstrained natural-language question. The service object may be a subscriber circuit, access port, optical distribution branch, virtual access function, residential gateway, VLAN, address session or regional service. Each object has identifiers that differ across inventory, assurance and customer systems. An architecture must therefore resolve identity before retrieval: account references map to service identifiers; services map to access and aggregation resources; devices map to software, configuration and geographical context. Identity resolution should be deterministic and return ambiguity when mappings conflict.

The evidence can be organised into six layers. Customer and service evidence records the experienced symptom, product, contractual objective and affected population. Topology and inventory evidence represents physical and logical dependencies, including splitters, fibres, access nodes, line cards, ports, service profiles and upstream paths. State evidence includes alarms, counters, optical levels, error rates, session records, resource utilisation and device health. Change evidence captures configuration commits, software upgrades, planned work and automated actions. Knowledge evidence contains vendor documents, standards, runbooks and resolved incidents. Context evidence includes weather, power events, construction activity and other external factors. The architecture should retrieve across these layers but preserve each source's authority, timestamp semantics and retention limitations.

Root-cause methods show why topology and time matter. Hierarchical causal models can combine dependencies between entities with abrupt changes in individual metrics to rank likely causes [15]. AIOps libraries similarly integrate causal discovery, Bayesian scoring and graph analysis while permitting expert constraints [16]. Telecom alarm research demonstrates that causal propagation and graph influence can identify initiating alarms within large correlated streams [17]. Privacy-preserving graph models further show that network and software dependencies may be analysed without centralising every raw dataset [18]. These methods do not remove the need for engineering judgement; they provide structured priors and ranking signals that an agent can use when forming and testing hypotheses.

Fixed access introduces domain-specific observability. Machine-learning research in optical access highlights traffic heterogeneity, data scarcity, constrained computation and the difficulty of transferring a model between deployments [21]. Experimental PON studies show that optical time-domain reflectometry can support faulty-branch identification, but overlapping reflections and topology changes complicate generalisation [22]. More recent work combines measured and synthetic traces, classifiers and branch-assignment data to detect and classify optical events [23]. Accordingly, a generative agent should not infer fibre impairment from prose alone. It should call a validated optical-analysis function, retrieve the relevant topology version, expose the trace interval and model confidence, and preserve the measurement as evidence.

The evidence model must also recognise negative findings. Absence of an alarm does not prove absence of a fault; a counter may be unavailable because telemetry stopped; a normal aggregate may conceal a subset failure; and a successful device query may occur after service recovery. Every retrieved item should therefore carry source, event time, collection time, object scope, quality flag, authorisation label and transformation history. This makes it possible to distinguish direct observations, computed features, model predictions and human assertions. Table 1 summarises the evidence classes and the questions each class can answer.

Table 1: Evidence classes for fixed-access network investigation

Evidence class	Typical sources	Primary investigative use	Required controls
Customer and service	Trouble reports, product catalogue, SLA, affected-customer set	Define symptom, severity, scope and customer impact	Identity resolution; minimisation; consent and retention
Topology and inventory	Service graph, OLT/ONU/port, splitter, VLAN, BNG path, software and location	Reconstruct dependencies and shared blast radius	Historical version; authoritative source; conflict flag
State and performance	Alarms, optical power, error counters, sessions, utilisation, CPE telemetry	Establish observations, baselines, change points and peer comparisons	Event and collection time; units; sampling quality
Change and workflow	Configuration commits, upgrades, maintenance, automation logs, tickets	Test temporal precedence and identify operational triggers	Actor, approval, diff, rollback and completion state
Knowledge and precedent	Standards, vendor documents, runbooks, known errors, resolved incidents	Interpret parameters, select tests and compare patterns	Version, owner, effective date, source trust and expiry
External context	Power, weather, civil works, third-party outages and regional events	Explain correlated effects beyond managed network boundaries	Source reliability; geographical and temporal matching

3. REVIEW METHODOLOGY

A structured critical review was conducted over publications and technical specifications dated 2020-2025. Searches combined terms for retrieval-augmented generation, language agents, tool use, root-cause analysis, AIOps, telecom fault diagnosis, optical access, intent-based networking, closed-loop automation, access-network abstraction and fixed-access operations. Peer-reviewed conference papers, journal articles, authoritative preprints with reproducible technical detail, and specifications from recognised standards bodies were included. Sources were excluded when they described generic chatbots without retrieval or execution, reported only promotional claims, lacked relevance to investigation, or fell outside the date range.

The synthesis used five analytical lenses. First, functional fit examined whether a method supports retrieval, computation, causal ranking, explanation or action. Second, evidence integrity assessed provenance, temporal alignment and numerical verifiability. Third, operational fit

considered latency, scalability, deployment isolation and integration with existing operations support systems. Fourth, safety examined permissions, prompt injection, tool constraints and change governance. Fifth, evaluation considered whether reported metrics correspond to engineering outcomes. The review did not aggregate accuracy values across incomparable datasets; instead, it used thematic comparison to identify reusable architectural principles and unresolved gaps.

The literature was mapped into three interacting bodies. Retrieval and agent research explains how a model can locate knowledge, choose tools and revise a plan [1-14]. Root-cause and telecom studies explain how alarms, metrics, topology and causal dependencies can be analysed [15-23]. Standards and governance sources define the management objects, interfaces and controls needed for deployment [25-30]. This triangulation avoids treating the language model as the complete solution. It positions the model as one component within an evidence, analytics and control system.

4. Proposed Evidence-Governed Architecture

The proposed architecture contains seven planes: interaction, orchestration, retrieval, investigation tools, evidence and causality, verification, and controlled action. The interaction plane accepts an incident, operator question or machine-generated trigger. It requires the requester to select or confirm the affected service object, investigation window and intended outcome. A policy engine attaches the user's role, permitted data domains and action rights. This step prevents a broad prompt from becoming an unrestricted search across customer and network data.

The orchestration plane converts the request into a typed investigation plan. It identifies sub-questions such as whether the issue is individual or shared, physical or logical, persistent or intermittent, and correlated with a recent change. The planner may use reasoning-and-action patterns [6], but its internal narrative is not treated as evidence. Each step must specify a tool, object scope, time range, expected output type and stop condition. A budget constrains tool calls, runtime and data volume. When identity, scope or intent is ambiguous, the orchestrator asks for clarification rather than selecting a convenient interpretation.

The retrieval plane maintains separate indexes for normative knowledge and operational memory. Normative collections include standards, vendor documentation, configuration guides and approved runbooks. Operational collections include tickets, post-incident reviews, change records and known-error articles. Hybrid retrieval combines lexical matching for exact alarm codes and parameter names with dense retrieval for conceptual similarity, then applies metadata filters and re-ranking. The need for mixed retrieval is supported by cross-domain evidence that dense methods may generalise unevenly [3]. Retrieved passages are immutable excerpts with document version, section, owner and effective date. Instructions embedded inside retrieved content are treated as data, not executable commands, because indirect prompt injection can manipulate tool-using systems [24].

The investigation-tool plane performs deterministic work. A topology tool resolves service paths and shared dependencies from inventory. A telemetry tool executes parameterised time-series queries and calculates aligned features. An alarm tool groups, deduplicates and orders events. A configuration tool provides read-only diffs by default. A ticket tool retrieves comparable incidents. An optical tool analyses traces and power measurements. A sandboxed code tool supports approved calculations when a pre-built function is unavailable. Tools expose narrow schemas, validate units and identifiers, and return structured results with error codes. The model cannot directly construct unrestricted database queries, shell commands or device changes.

The evidence and causality plane converts tool outputs into an investigation graph. Nodes represent services, resources, observations, changes, hypotheses and tests; edges represent dependency, temporal precedence, correlation, contradiction and support. Learned RCA methods may suggest edges or root-cause rankings [15-18], but the graph distinguishes discovered relationships from inventory-defined relationships and operator assertions. A hypothesis record contains the proposed cause, affected scope, predicted observations, evidence for and against, tests performed, and residual uncertainty. This structure reduces confirmation bias because an agent must seek disconfirming evidence as well as supporting signals.

The verification plane independently checks the candidate conclusion. It re-executes critical calculations, verifies that cited evidence falls within the incident window, confirms topology version, checks unit conversions and compares the narrative with structured outputs. RAG evaluation frameworks encourage separate assessment of context and generated answers [11], while benchmarks reveal weakness when context is noisy or false [12]. Corrective retrieval can trigger a new search when evidence quality is low [10], and reflective methods can improve a draft [13,14]; nevertheless, final acceptance should depend on deterministic

assertions where possible. A verified brief labels every conclusion as observed, calculated, inferred or unresolved.

The controlled-action plane is deliberately separate. Read-only investigation may run automatically within authorised scopes. Low-risk actions, such as opening a ticket, attaching evidence or requesting a measurement, can be policy-approved. Configuration, reboot, profile change or traffic movement requires an explicit action proposal containing target, expected effect, pre-checks, rollback, blast radius and approval state. Closed-loop automation standards emphasise lifecycle, governance and coordination rather than unbounded action [26]. The architecture therefore supports increasing autonomy by evidence class and action risk, not by granting a general agent broad network credentials.

Fixed-access standards provide the integration substrate. Broadband Forum work on access and home-network operations frames automation and intelligence requirements [27]. Network-map and equipment-inventory YANG modules support a consistent topology representation [28]. Access-network abstraction provides common control and management views across traditional and disaggregated nodes [29], while functional disaggregation separates access-node hardware from cloud-hosted control and management functions [30]. The architecture should use these models and interfaces where available, reducing the number of vendor-specific assumptions embedded in prompts or agent code.

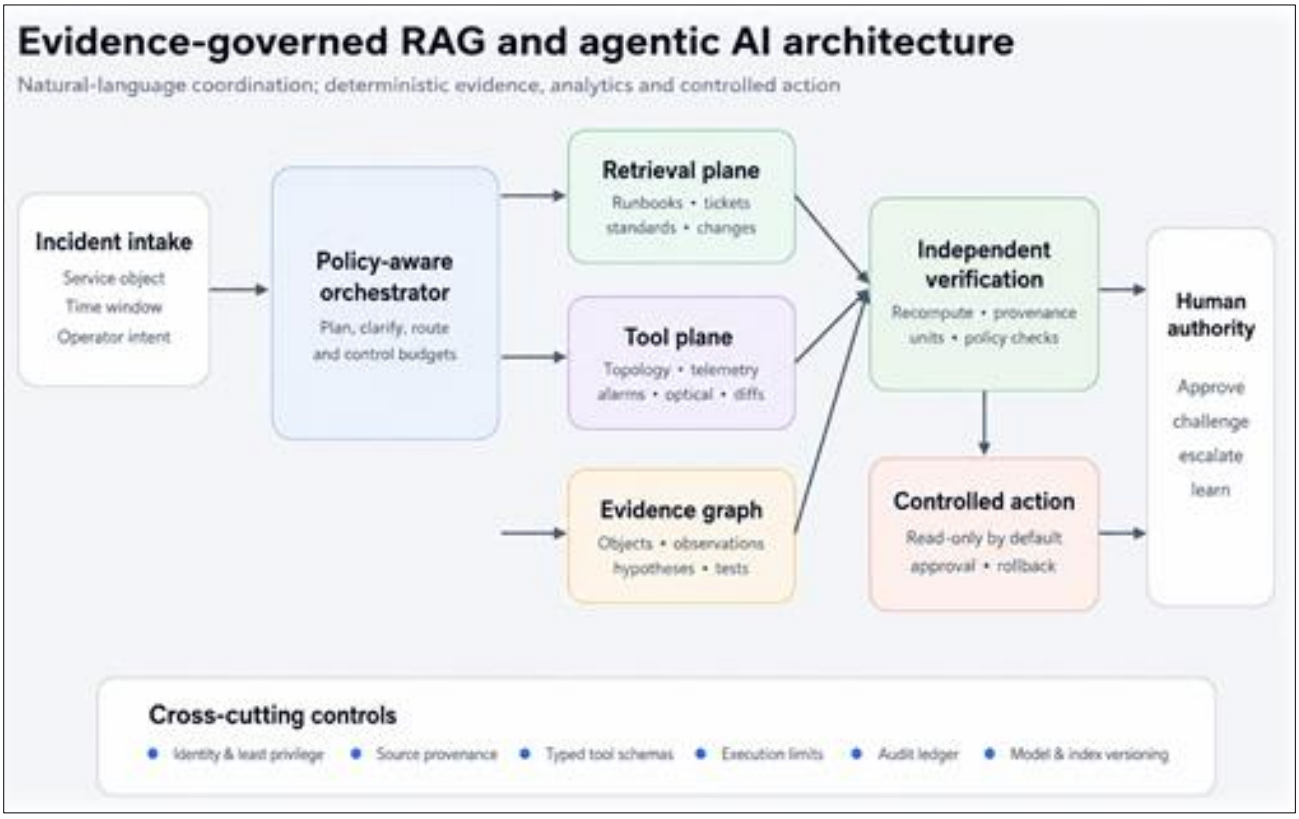

Figure 1: Evidence-governed architecture separating orchestration, retrieval, deterministic tools, verification and controlled action

5. Agentic Investigation Workflow

The workflow begins with incident framing. The agent normalises the symptom, determines affected products and identifies the smallest defensible service object. It checks whether the question requests diagnosis, impact assessment, evidence collection or remediation. It then fixes an incident window that includes a baseline period and notes clock uncertainty. A severe incident may justify broader evidence collection, but the same permission and minimisation rules apply.

The second stage aligns evidence. Identity mappings are resolved, topology is reconstructed for the relevant historical time, and source clocks are normalised. The agent retrieves customer contacts, alarms, counters, optical data, session state and recent changes in parallel, but it does not immediately narrate a cause. Instead, it creates an evidence timeline and marks gaps. This is important because apparent precedence may be an artefact of delayed ingestion, polling frequency or alarm suppression.

The third stage generates a limited set of hypotheses from domain rules, similar incidents and causal models. For example, widespread loss of signal beneath one splitter, simultaneous optical-power deterioration and an external-work record may support a distribution-fibre hypothesis. Authentication failures with stable optical levels and a recent policy deployment may support a logical-control hypothesis. The agent ranks hypotheses using scope match, temporal precedence, topology consistency, signal specificity, historical frequency and data quality. Frequency is never allowed to override contradictory direct evidence.

The fourth stage designs discriminating tests. Each test should have an expected result under the hypothesis and an alternative result that weakens it. The agent may compare affected and unaffected peers, calculate a change point, inspect a configuration difference, request a fresh optical measurement, trace a session, or query shared-resource utilisation. Tool outputs are added to the evidence graph, and hypothesis scores are updated. If a test cannot be executed because data is missing or permissions are insufficient, the limitation remains visible rather than being filled by model inference.

The fifth stage performs counterfactual and blast-radius checks. The investigator asks whether the proposed cause explains unaffected services, whether another event better accounts for the timeline, and whether recovery occurred without removing the suspected cause. Shared-dependency analysis estimates which other customers or nodes should be affected. This stage is essential for preventing a local symptom from being attributed to a regional event, or a coincidental change from being labelled causal.

The sixth stage produces a calibrated investigation brief. It states the incident scope, most likely cause, confidence, decisive evidence, contradictory evidence, unresolved questions, affected population and recommended next step. Confidence is derived from evidence completeness and test results rather than the model's verbal certainty. Numerical statements are linked to tool outputs. Document claims are linked to exact excerpts. The brief uses engineering language and distinguishes root cause from trigger, contributing condition and observable symptom.

The final stage governs action and learning. An operator approves, rejects or revises the conclusion. Approved remediation follows pre-check, execution, verification and rollback gates. The outcome is captured as an incident memory containing the evidence pattern, successful tests, false leads, final cause and verified fix. Retrieval of this memory in future incidents can improve consistency, but its scope, software context and expiry must be recorded. Figure 2 shows the gated loop, and Table 2 compares agent patterns suitable for each stage.

Figure 2: Gated investigation workflow with continuous assurance and outcome learning

Table 2: Agent patterns and their defensible use in the investigation lifecycle

Pattern	Best use	Principal failure mode	Required safeguard
Planner-orchestrator	Decompose incident and sequence evidence collection	Over-broad scope or unnecessary tool calls	Typed plan, budgets, clarification and stop conditions
Retrieval agent	Locate standards, runbooks, tickets and change records	Stale, irrelevant or malicious context	Hybrid retrieval, metadata filters, re-ranking and trust labels
Topology/telemetry agent	Resolve dependencies and perform deterministic calculations	Wrong object, time window, unit or aggregation	Read-only typed tools, historical topology and schema validation
Causal hypothesis agent	Rank candidate causes and propose discriminating tests	Correlation mistaken for causation; confirmation bias	Evidence graph, negative tests and counterfactual checks
Verifier agent	Recompute claims and check provenance and policy	Same-model agreement mistaken for independence	Separate code paths, deterministic assertions and thresholded abstention
Action agent	Prepare approved remediation and workflow updates	Excessive agency or uncontrolled blast radius	Risk tier, approval token, pre-check, rollback and post-check
Learning agent	Curate verified incident memories and regression cases	Encoding a false or obsolete lesson	Human sign-off, context metadata, expiry and revalidation

6. Assurance, Security and Governance

Security must be designed around the assumption that prompts, retrieved text and tool outputs are untrusted. A ticket may contain copied command lines, vendor documents may include examples that resemble instructions, and external advisories may be maliciously modified. Indirect prompt-injection research demonstrates that models can confuse contextual data with executable directions [24]. The system should isolate policy instructions from retrieved content, strip active markup, label source trust, scan for injection patterns and prohibit retrieved text from changing permissions or tool policies.

Least privilege applies to both data and actions. The agent receives short-lived credentials for explicitly approved tools and objects. Read access is filtered by customer, region, technology and role. Write tools are unavailable during analysis unless a separate approval token is issued. Every tool call records requester, model version, prompt template, arguments, returned rows or artefacts, and result hash. Sensitive fields are masked before they enter the model context. The evidence store retains enough detail for audit while applying contractual and regulatory retention requirements.

Model and retrieval governance should follow lifecycle risk management. The NIST AI Risk Management Framework organises practice around governing, mapping, measuring and managing AI risk [25]. Applied here, governance defines ownership and acceptable autonomy; mapping identifies operational harms and affected stakeholders; measurement tests retrieval, calculations, bias and failure modes; and management sets thresholds, escalation and monitoring. Model upgrades, embedding changes, index rebuilds and prompt revisions require versioned evaluation because apparently minor changes can alter retrieval and tool selection.

Operational safety requires explicit failure behaviour. When evidence is insufficient, the system must abstain, narrow the claim or request a human decision. When tools disagree, it should preserve both results and identify the discrepancy. When a query exceeds resource or time limits, it should return partial evidence rather than a fabricated completion. When confidence is below threshold, no remediation proposal should be promoted to automatic execution. These behaviours are more important than conversational smoothness.

Governance also covers organisational accountability. Network engineering owns diagnostic rules and action constraints; data owners validate source semantics; security teams approve credential and sandbox designs; service assurance owns operational metrics; and an AI governance body reviews model risk. Investigators require training to interpret confidence, inspect evidence and recognise automation bias. The agent should reduce clerical work while keeping operators actively engaged in hypothesis selection and approval.

7. Evaluation and Operationalisation

Evaluation should use an incident corpus with verified root causes, timelines, topology

snapshots, tool outputs and remediation outcomes. The corpus must include common faults, rare events, multi-fault cases, missing data, misleading alarms, stale inventory and benign changes. Synthetic cases are useful for coverage but should not replace real operational incidents. Splits should prevent near-duplicate tickets or repeated network events from appearing in both training and test sets.

Metrics should be reported at component and end-to-end levels. Retrieval metrics include evidence recall, precision, version correctness and citation coverage. Tool metrics include argument validity, execution success, numerical error and unit correctness. Investigation metrics include top-k root-cause accuracy, mean reciprocal rank, hypothesis calibration, time-to-isolation and number of unnecessary queries. Safety metrics include unauthorised access attempts, prompt-injection success, unsupported claims, unsafe-action proposals and abstention quality. Human factors include operator agreement, correction rate, workload, trust calibration and time saved. RAGAS-type measures can support context and answer assessment [11], but network cases require exact comparison of counters, timestamps and identifiers.

A phased deployment reduces risk. Phase one provides search and evidence summarisation with no tool execution. Phase two adds read-only topology, telemetry and ticket tools. Phase three introduces hypothesis ranking and deterministic verification. Phase four allows low-risk workflow actions, such as evidence attachment and measurement requests. Only after stable performance and governance maturity should selected remediation actions enter supervised closed loops. Each phase requires shadow operation, comparison with expert investigations and rollback to manual procedures.

Performance architecture matters. Fixed-access estates can contain millions of services and high-volume telemetry, so the agent should not retrieve raw data into a prompt. It should push filtering, aggregation and windowing into data platforms, cache stable topology and document results, and stream only compact evidence objects to the model. Long-running analyses should be asynchronous and resumable. Regional or on-premises deployment may be required where operational data cannot leave the operator environment.

Economic evaluation should include more than model cost. Benefits arise from reduced mean time to isolate, fewer dispatches, improved first-time resolution, consistent evidence packs and faster onboarding of investigators. Costs include data integration, source-quality remediation, security controls, evaluation, model hosting and ongoing knowledge maintenance. A narrow, well-instrumented use case may deliver more value than a broad assistant that cannot be audited.

A practical acceptance test should also include deliberate ambiguity. Investigators often ask compressed questions such as whether a line is unstable, although instability may refer to optical power, synchronisation, packet loss, session churn or customer-perceived interruption. The system should identify the missing definition, propose measurable alternatives and wait for confirmation when the distinction changes the investigation path. This behaviour tests whether the agent manages uncertainty responsibly instead of producing a fluent answer to an underspecified request.

A representative fixed-access benchmark should preserve the sequence in which evidence became available, because a retrospective package can make diagnosis unrealistically easy. Each case should therefore provide successive snapshots: the initial customer symptom, the first alarm set, the topology known at that moment, later telemetry, subsequent change records and the final verified cause. During evaluation, the agent receives only the snapshot available at each stage. This design measures whether it asks productive questions, updates beliefs when new observations arrive and avoids using information that would not yet have been accessible to an investigator. It also distinguishes genuine diagnostic reasoning from simple matching against a completed incident report. Scenario construction should cover the principal access technologies and operational boundaries. Fibre cases may include gradual optical degradation, rogue optical network units, unstable ranging, overloaded passive branches, failing line-card optics and mis-associated reflectometry traces. Copper cases should include impulse noise, crosstalk, deteriorating joints, profile instability and vectoring interactions. Service-layer cases should include address-session churn, authentication delay, incorrect quality-of-service mapping, multicast defects and residential gateway software regressions. Cross-domain cases are particularly important because the apparent access symptom may originate in aggregation, power, timing, cloud-hosted control or customer premises. The benchmark should also include healthy services with unusual but acceptable behaviour so that the system is penalised for over-diagnosis. Expert adjudication needs a documented protocol. At least two experienced engineers should independently identify the supported cause, plausible alternatives, decisive evidence, missing evidence and actions that would have been unsafe. Disagreement should be resolved through a recorded

review rather than hidden by a single gold label. Some incidents legitimately retain uncertainty; these cases can use graded labels that distinguish confirmed, probable, possible and excluded causes. Agent confidence should be evaluated against those grades, and evidence citations should be checked at claim level. A correct cause accompanied by invented measurements or irrelevant sources should not receive a passing score. Operational trials should compare the architecture with current practice rather than with an artificial baseline. Useful comparators include manual investigation, existing rule-based correlation, search-only assistance and a language model without deterministic tools. A crossover design can assign comparable incidents to different investigators and conditions while controlling for expertise and technology. Alongside accuracy and speed, reviewers should record how often the system changes an engineer's initial view, whether that change is beneficial, and whether operators can detect incorrect recommendations. Post-trial interviews can expose hidden workload, such as time spent checking citations or correcting object mappings. Deployment is justified only when measurable gains persist after these verification costs are included.

Finally, longitudinal monitoring should detect drift in sources, terminology and tool behaviour. Retrieval coverage may decline after documentation restructures; topology mappings may change after migrations; and software releases can alter counter semantics. A standing regression suite should run after every model, prompt, index, schema or connector change. Results should be segmented by access technology, region, fault class and incident complexity. Degradation beyond agreed tolerance should block release, trigger source-owner review and preserve the previous production version until the discrepancy is understood. This discipline converts evaluation from a one-off demonstration into a continuing operational control that protects reliability as the environment evolves.

8. DISCUSSION AND CONCLUSION

The review indicates that RAG and agentic AI can materially improve fixed-access investigation when their roles are constrained. Retrieval is valuable for locating current documentation and comparable incidents; language models are valuable for decomposing questions, coordinating tools and presenting evidence; causal and graph methods are valuable for ranking propagation paths; and deterministic tools are essential for numerical and state-based claims. The architecture fails when these functions are collapsed into one unconstrained prompt.

Several research gaps remain. Public fixed-access datasets rarely combine topology, telemetry, alarms, tickets and verified causes, limiting reproducible evaluation. Temporal topology and configuration history are often absent even though they are essential for causal interpretation. Confidence calibration for multi-tool agents is immature, and self-reflection may reinforce an incorrect premise. Future work should develop benchmark incidents, typed evidence standards, counterfactual test generation, safe multi-agent coordination and methods that quantify the marginal value of each evidence source.

In conclusion, automated investigation should be designed as an evidence-governed engineering workflow rather than a conversational shortcut. A defensible system retrieves permissioned knowledge, executes bounded tools, represents hypotheses explicitly, verifies decisive claims and separates analysis from action. It should accelerate the path from symptom to tested explanation while making uncertainty and provenance more visible. Under these conditions, RAG and agentic AI can strengthen operational resilience across modern fixed-access networks without displacing the technical judgement and accountability of network professionals.

REFERENCES

1. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S. and Kiela, D. (2020), 'Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks', Advances in Neural Information Processing Systems, 33, 9459-9474.
2. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W.-t. (2020), 'Dense Passage Retrieval for Open-Domain Question Answering', Proceedings of EMNLP 2020, 6769-6781.
3. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. and Gurevych, I. (2021), 'BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models', NeurIPS Datasets and Benchmarks, 1.
4. Borgeaud, S., Mensch, A., Hoffmann, J. et al. (2022), 'Improving Language Models by Retrieving from Trillions of Tokens', Proceedings of the 39th International Conference on Machine Learning, 162, 2206-2240.
5. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M. and Wang, H. (2023), 'Retrieval-Augmented Generation for Large Language Models: A Survey', arXiv:2312.10997.
6. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y. (2023), 'ReAct: Synergizing Reasoning and Acting in Language

Models', International Conference on Learning Representations.
7. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N. and Scialom, T. (2023), 'Toolformer: Language Models Can Teach Themselves to Use Tools', Advances in Neural Information Processing Systems, 36.
8. Patil, S.G., Zhang, T., Wang, X. and Gonzalez, J.E. (2024), 'Gorilla: Large Language Model Connected with Massive APIs', Advances in Neural Information Processing Systems, 37.
9. Asai, A., Wu, Z., Wang, Y., Sil, A. and Hajishirzi, H. (2024), 'Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection', International Conference on Learning Representations.
10. Yan, S.-Q., Gu, J.-C., Zhu, Y. and Ling, Z.-H. (2024), 'Corrective Retrieval Augmented Generation', arXiv:2401.15884.
11. Es, S., James, J., Espinosa-Anke, L. and Schockaert, S. (2024), 'RAGAS: Automated Evaluation of Retrieval Augmented Generation', Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, 150-158.
12. Chen, J., Lin, H., Han, X. and Sun, L. (2024), 'Benchmarking Large Language Models in Retrieval-Augmented Generation', Proceedings of the AAAI Conference on Artificial Intelligence, 38(16), 17754-17762.
13. Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K. and Yao, S. (2023), 'Reflexion: Language Agents with Verbal Reinforcement Learning', Advances in Neural Information Processing Systems, 36.
14. Madaan, A., Tandon, N., Gupta, P. et al. (2023), 'Self-Refine: Iterative Refinement with Self-Feedback', Advances in Neural Information Processing Systems, 36.
15. Wang, D., Chen, Z., Ni, J., Tong, L., Wang, Z., Fu, Y. and Chen, H. (2023), 'Hierarchical Graph Neural Networks for Causal Discovery and Root Cause Localization', arXiv:2302.01987.
16. Liu, C., Yang, W., Mittal, H., Singh, M., Sahoo, D. and Hoi, S.C.H. (2023), 'PyRCA: A Library for Metric-based Root Cause Analysis', arXiv:2306.11417.
17. Zhang, K., Kalander, M., Zhou, M., Zhang, X. and Ye, J. (2021), 'An Influence-based Approach for Root Cause Alarm Discovery in Telecom Networks', arXiv:2105.03092.
18. Bourgerie, R. and Zanouda, T. (2023), 'Fault Detection in Telecom Networks Using Bi-level Federated Graph Neural Networks', IEEE International Conference on Data Mining.
19. Njah, Y., Leivadeas, A., Venugopal, N. and Farahmand, F. (2023), 'Toward Intent-Based Network Automation for Smart Environments: A Healthcare 4.0 Use Case', IEEE Access, 11, 136565-136576.
20. Mekrache, A., Ksentini, A. and Verikoukis, C. (2024), 'Intent-Based Management of Next-Generation Networks: An LLM-Centric Approach', IEEE Network, 38(5), 29-36.
21. Wong, E., Mondal, S. and Ruan, L. (2023), 'Machine Learning Enhanced Next-Generation Optical Access Networks—Challenges and Emerging Solutions', Journal of Optical Communications and Networking, 15, A49-A62.
22. Abdelli, K., Tropschug, C., Griesser, H. and Pachnicke, S. (2023), 'Faulty Branch Identification in Passive Optical Networks Using Machine Learning', Journal of Optical Communications and Networking, 15(4), 187-196.
23. Straub, M., Reber, J., Saier, T., Borkowski, R., Li, S., Khomchenko, D., Richter, A., Färber, M., Käfer, T. and Bonk, R. (2024), 'ML Approaches for OTDR Diagnoses in Passive Optical Networks—Event Detection and Classification: Ways for ODN Branch Assignment', Journal of Optical Communications and Networking, 16(7), C43-C50.
24. Yi, J., Xie, Y., Zhu, B., Kiciman, E., Sun, G., Xie, X. and Wu, F. (2023), 'Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models', arXiv:2312.14197.
25. National Institute of Standards and Technology (2023), Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, doi:10.6028/NIST.AI.100-1.
26. ETSI (2021), Zero-touch Network and Service Management (ZSM); Closed-Loop Automation; Part 1: Enablers, ETSI GS ZSM 009-1 V1.1.1.
27. Broadband Forum (2021), Access and Home Network O&M Automation/Intelligence, Technical Report TR-436.
28. Broadband Forum (2021), YANG Modules for Network Map and Equipment Inventory, Technical Report TR-454.
29. Broadband Forum (2022), Access Network Abstraction, Technical Report TR-484.
30. Broadband Forum (2024), CloudCO Enhancement—Access Node Functional Disaggregation, Technical Report TR-477.