Network Innovation

Kindly supported by:

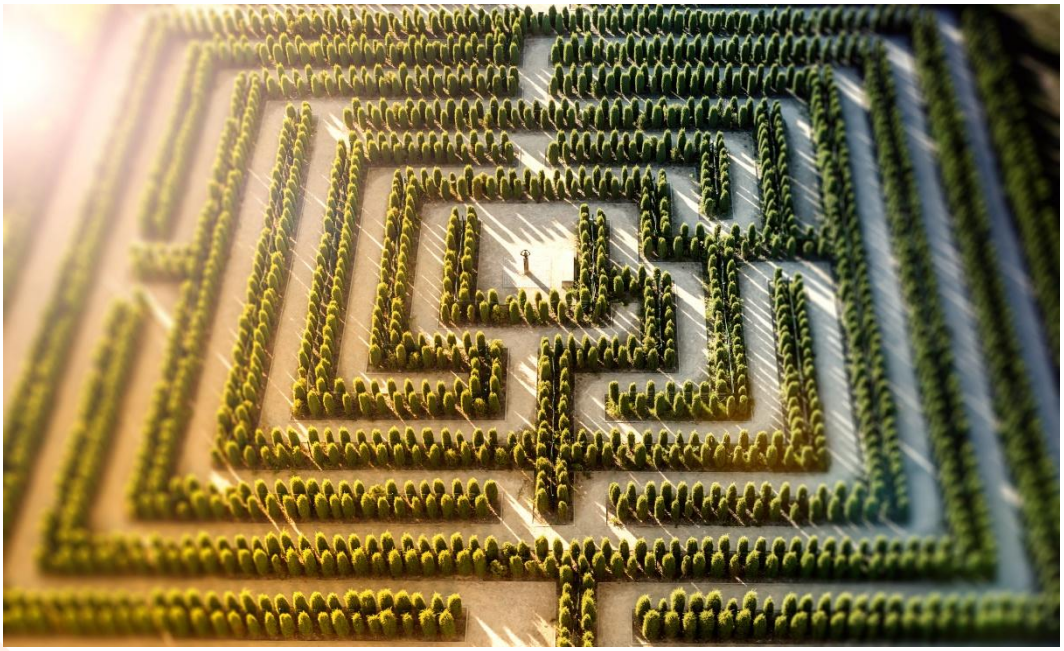
VITRIA VIA AI

STL PARTNERS

Executive Briefing

THE JOURNEY TO A SELF-HEALING NETWORK: INTELLIGENCE, AGENTS AND COMPLEXITY

As the deployment of AI/ML ramps up, telcos need to develop an 'intelligence architecture', supporting the move towards Level 4 and 5 in the TM Forum's Autonomous Network Framework. How will such an intelligent architecture support the self-healing network?



Charlotte Patrick, Associate Analyst | February 2025

Foreword

Methodology

This report presents analysis and insights from an interview programme carried out between October 2024 and January 2025 with 14 senior decision-makers in telcos and vendors worldwide. The aim was to understand the current plan for development of intelligence in the network to enable delivery of Level 4 and Level 5 autonomy according to the TM Forum's Autonomous Network Framework model.

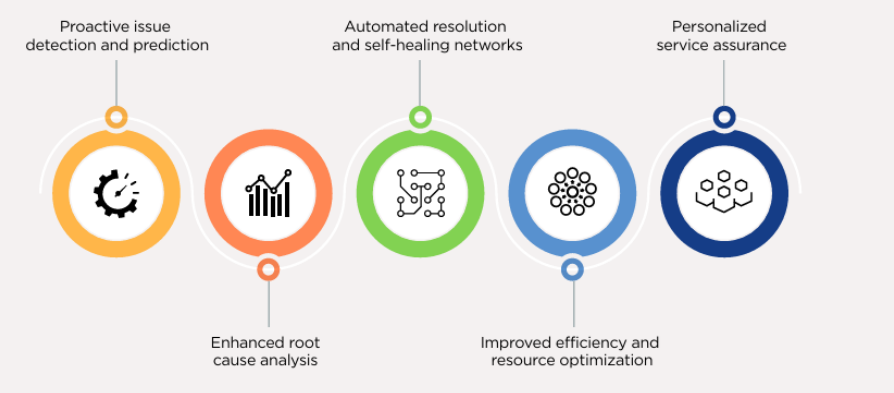
Editorial independence

This report has been prepared by Charlotte Patrick Research in collaboration with STL Partners and was commissioned by Vitria Technology. STL Partners maintains strict editorial independence. Mentions or allusions to companies or products in this document are intended as illustrations of market evolution and are not included as endorsements or product/service recommendations.

Our sponsor

Vitria has a long history of success in supporting telcos with streaming analytics, operational intelligence, internet of things (IoT) analytics and artificial intelligence for IT operations (AIOps). In service assurance, it leverages AI as illustrated in Figure 1.

Figure 1: Transforming service assurance with AI



Source: AI: Transforming Service Assurance - Vitria

Vitria is well-placed to support telcos as they start building an intelligence architecture, by providing:

- **Ecosystem observability**; using ML/AI-based incident cross-correlation across technology stacks and network domains;
- **Intelligent automation** – relying on ML-generated incident analysis and probable cause;
- ML-based **automation of service recovery** tasks based on recommended fix.

Its intelligent solutions can be quickly deployed thanks to ease of onboarding data and provide fast time to value with quantifiable ROI.

Executive Summary

Assurance is one of the most data-intensive activities on the telco network, and requires both intelligence and data federation. There has already been a strong focus on machine learning (ML) deployment by telcos – these models are required to take the telco on a journey from observability (the ability to see the performance of the network or faults occurring and their root cause) towards a self-healing network where remediation of issues is performed in an autonomous manner. In addition to this focus on ML deployment, telcos are also implementing generative AI (GenAI) where copilots are used to understand issues and provide network and service operations teams with suggested resolutions. During 2024, assurance has also seen the first telco deployments of simple agent and multi-agent systems (MAS), which look set to become some of the underpinning requirements towards the self-healing network.

The journey towards an intelligence architecture

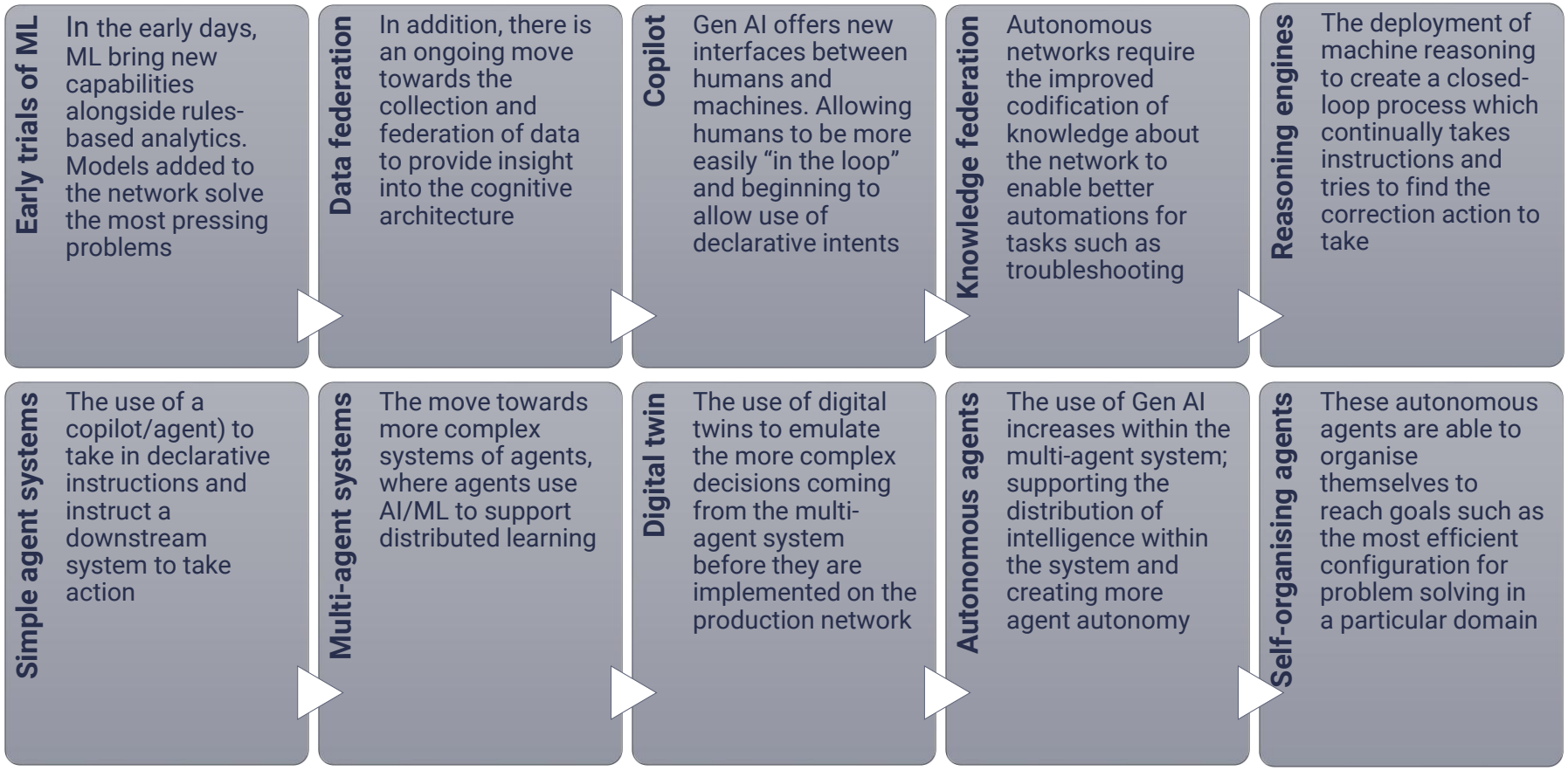
We use the term 'intelligence architecture' in this research to describe the move from the ad lib deployment of ML models towards a more considered architecture for intelligence and data collection, in order to support the complex requirements of a network at Levels 4 and 5 of the TM Forum's Autonomous Networks Framework model. Some of these key requirements include the need to distribute intelligence at appropriate points in the network (potentially, close to the location of the data that will feed it), the development of new solutions to capture and manage knowledge, scalability to minimise resources needed to support it and lifecycle management of the models.

At a top level, telcos need to take action across four dimensions:

- **Data** – the gathering, federation, transportation and use of data for training models and its use in the models;
- **Intelligence** – development of appropriate models to perform good-quality decision-making;
- **Agents** – deployment of more simple agent systems and more intelligent multi-agent systems (MAS) which bring together multiple agents to solve problems or achieve specific goals;
- **Knowledge** – the federation and the organisation of knowledge about the network, services, devices and other elements to enable competent decision-making.

A telco intelligence architecture may develop several ways, depending on factors such as the speed of improvement of AI/ML, so that the quality of its decision-making makes it useful in an autonomous network. The general direction of travel is shown in Figure 2 below.

Figure 2: The journey towards an intelligence architecture for assurance



Source: Charlotte Patrick Research

Creating a set of solid first steps towards building the intelligence for a self-healing network

Overcoming the significant barriers to creating anything more than a simple agent system (getting the telco to the bottom left of Figure 2) is likely to be a five to ten-year project.

To be pragmatic, telcos can approach this in the following ways:

- **Deployment of a simple agent** to support teams in a few assurance processes, allowing the development of in-house AI skill sets (in particular, the encoding of human knowledge into the agent and the training of humans in how to best interact with the agent).
- **Early focus on building data mesh**: This is a decentralised data architecture approach where data is treated as a product, managed by domain-specific teams which can support the data needs of both existing intelligence and any future agent deployments. Where data will need to be centralised to support longer-term plans operators will need to consider alternative options.
- **Testing of knowledge graphs or other types of knowledge base**: These are graphs that hold information on the network elements of a production network, such as cell towers, network devices and customers. During our research, one interviewee was trialling the population of their first graph with documents and tickets created by their copilot.
- **Bringing any existing analytics/ML models** already in place to do diagnostics and decisioning for a particular use case **into a simple agent system**. This will allow the telco to gain some experience of the basic requirements of an agent system.
- **Development of a basic blueprint for a multi-agent system** to use in discussion between interested parties in a telco and potential vendors. This will force some initial decisions on the best use cases to tackle, what elements of Figure 2 to trial immediately and what other elements to put on the watch list, whether there is a business case to build the first large language model (LLM)-based agents, etc.

Planning an intelligence architecture using agents will bring it in line with other future-focussed projects such as open RAN, which require similar network characteristics, including the distribution of intelligence, virtualisation and disaggregation. It also creates a modular system of models where models can be easily swapped in and out, as required and when new technologies appear. This is particularly useful since rapid technology change in this area is very likely within the next few years.

Other recommended actions

In the words of one interviewee, “achieving Level 4 is going to be very, very tricky”. This means that:

- The initial business case will be at its strongest if a value can be put against the delivery of a self-healing network, with room to define the exact architecture as technologies mature. In our recent research (Finding value from AI, analytics and automation in the telco: Part 2), we calculated that

an average telco (with annual revenues of USD15.6 billion) could generate benefits of USD62 million per year thanks to a self-healing network.

- Only very few people within telcos fully understand the topics in this research note. Significant education needs to take place to combat the confusion and the potential for hype of the self-healing network. This will allow good-quality discussions on the telco's aspirations and will better equip the senior team to understand whether a MAS is needed and what ROI it can bring.
- If there is relative uncertainty as to where an agent network might be more beneficial (it was certainly not clear from our research which use cases might benefit from a more agentic solution), telcos should then focus on testing the most likely beneficiaries of increased intelligence (those throwing more complex problems such as running slices or some long-term planning). This should be the focus of early agentic trials.

Current opinions on GenAI within a telco are rather muted, with deployed solutions often taking time to bring suitable results and frustration voiced about outcomes from new copilots:

- Where possible, work on improving trust by ensuring there is training on easy wins such as prompt generation. There is a good chance that today's GenAI limitations will be overcome in the next five years – given the billions of dollars being poured into the area – and the organisation needs to be ready to increase its use in that timeframe.

Telcos must remember that human involvement in network problem-solving does not create 100% accurate resolutions – so, any trialled automations should be measured against a realistic benchmark of what human remediation would achieve

Table of Contents

Table of Figures

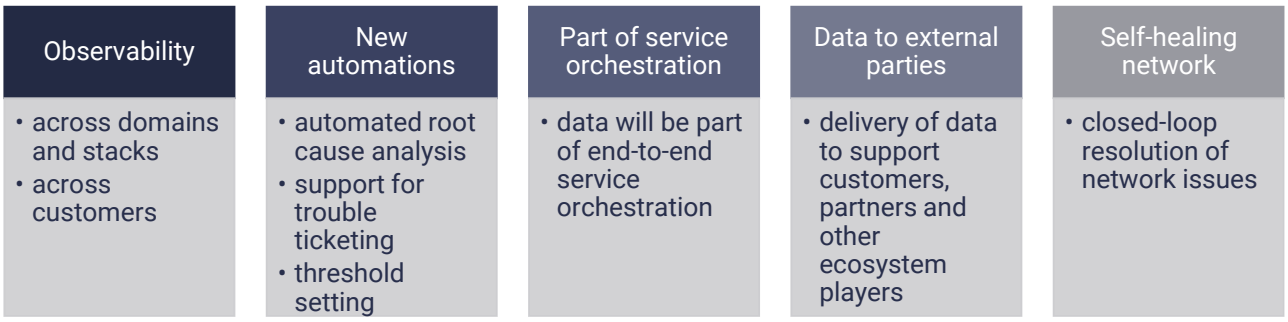
Introduction

Among the trends impacting the deployment of new AI/ML in assurance, one area that is top of mind for the telcos and the vendors that we recently interviewed was the need to bring together data from across domains and up/down stacks, along with the necessary intelligence, to support decision-making for more complex root cause analysis.

Telcos have already significantly invested in data gathering but data federation (the 'stitching' together of multiple data sources to allow them to be used by assurance solutions) is still an ongoing task. However, it is important because some decisions may require information from various data siloes on the network (to answer a question such as: "Is the problem being experienced by this group of customers caused by an issue on the RAN or in the core?"). Data from systems such as billing and customer relationship management (CRM) may be needed to provide customer-specific insight ("How much will this customer-affecting network problem cost the company?").

Figure 3 sets out a range of uses for this federated intelligence and data:

Figure 3: The need for federated intelligence and data



Source: Charlotte Patrick Research

Federated data will be necessary across all these uses because of several reasons.

Observability

5G standalone (SA) creates significant demand for additional reporting, visualisation and clustering for triage and decision-making – first across domains and stacks; and then across customers for service assurance:

- **Across domains and vertical stacks:** Apart from the already noted need to collect and understand data from across domains and vendors, the move from dedicated hardware to virtual machines and containers requires new data sets from IT stacks. These more complex views will require data federation, including third-party data from other systems gathered to a suitable point in the network for analysis.

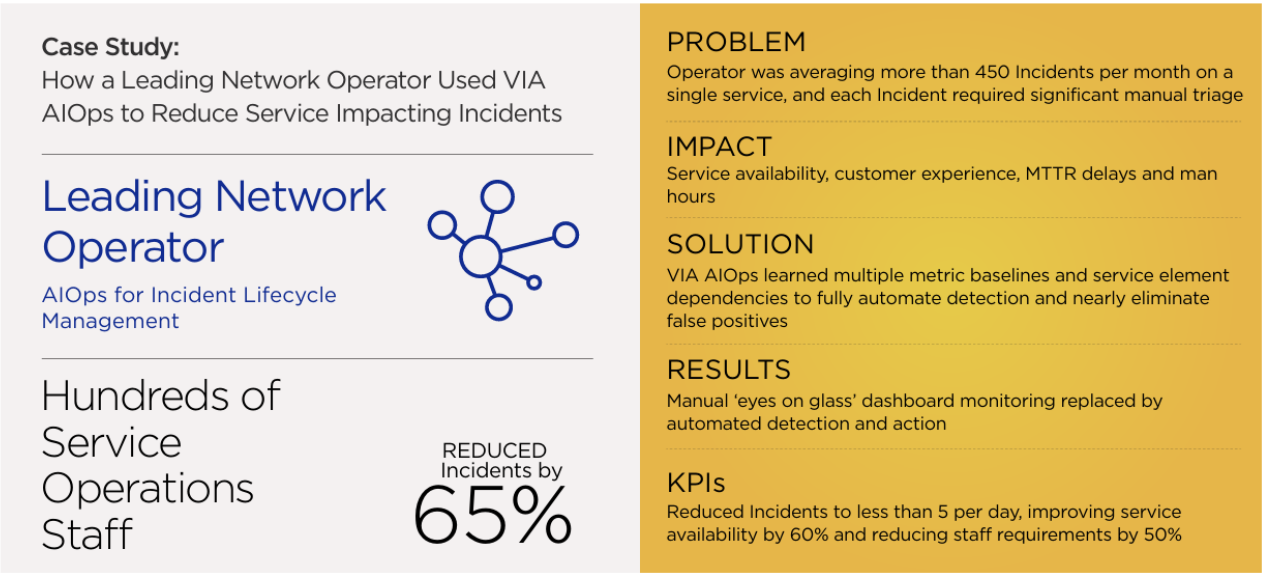
- **Across customers:** Over the last 10 years, increasing focus on service assurance has required data analysis from individual customers or services to identify customer-impacting events. The

continuing improvement in network resilience (from virtualisation and new automations) requires less focus on individual network issues from the network operations centre (NOC); allowing more focus on identifying and fixing problems that impact individual services and customers. Data from the network and a range of other sources (test data, weather patterns and customer sentiment) is needed – and new ML models have been developed to deal with the increasing volume of data and undertake anomaly detection, prediction and optimisation tasks. Our interviewees also reported that there was a sustained focus on assurance products for enterprises for VPN services, cloud gaming and other latency-sensitive applications.

New automations

Vendors interviewed noted that a good percentage of their engagements now included automation, as telcos look towards Levels 3 and 4 of the TM Forum Autonomous Network Framework. Automating root cause analysis using new ML techniques has been the most prominent activity in the last few years – and will remain the crucial first step in a self-healing network. Other automation areas include predicting future performance and the customer impact of actions taken, as well as dynamically setting thresholds and remediations such as the opening of trouble tickets.

Figure 4: Example of automated fault detection



Source: Vitria (VIA AIOps - Quantifiable Business Value - Vitria)

Part of service orchestration

End-to-end service orchestration is a single process that fulfils a customer's requirement for a new service by combining pre-existing solutions from the telco and its partners – from order handling through service design, service/resource provisioning and assurance. It is increasingly needed to support a much wider range and complexity of 5G services. Assurance data, along with other data sources such as inventories, will be required to support orchestrations and service design. For example, a newly designed service will notify assurance about how network functions are chained together and the KQIs/SLAs required to support the customer. The assurance platform will then track

compliance (i.e. checking that all metrics within the required range) and forecast potential threats to KQIs/SLAs.

Data to external parties

Meeting enterprise customer requirements around new services will require telcos to integrate their data and operations across diverse and complex ecosystems. Delivering the right data (expected to be a mix of assurance, inventory and external third-party data) to the right people and processes at the right time will provide new visibility for enterprise customers and other partners. In the future, it will likely also support a range of data feeds and automations that stretch from the telco into these customers and partners.

Self-healing networks

Telcos face challenges in supporting new, more dynamic networks that generate multiple concurrent issues – making the self-healing network concept very attractive.

The journey towards self-healing actually started many years ago. And, as discussed by one research participant, "the sad reality is that many problems can be fixed by just turning off the offending hardware/software and turning it on again", meaning that many of the first self-healing capabilities seen as far back as 2011 (see 3GPP's Self-Organizing Network (SON) Release 10) focused on the restarting of equipment after software glitches. The next iteration then used the term 'self-healing' to describe issues such as traffic rerouting in the event of a fibre cut.

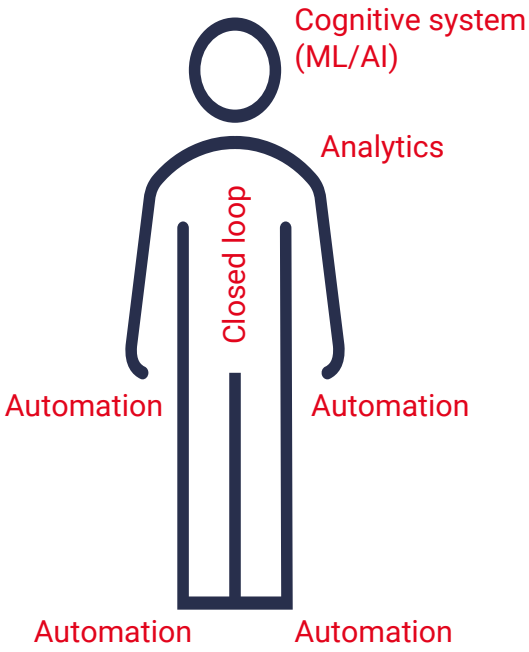
More recently, self-healing has focussed on architectural builds in the cloud where services failover automatically to standby hardware and backup links. Indeed, the advent of virtualised networks allows the term to expand from very simple activities (e.g., scheduling the nightly rebooting of a network function) to more complicated closed-loop activities such as adapting in real time to faults or new demands on the network. These require automations to understand the customer or service-level problem and then move to create resolutions in a particular network domain or multiple domains.

What is intelligence architecture?

The more complex types of self-healing such as those we have just described require the full range of intelligence and automation capabilities. To use an analogy with the human body and as illustrated in Figure 5:

- ML and AI provide the brain.
- Various analytics models act like the human nervous system and bring information to the brain.
- Multiple automations actuate instructions from the brain to implement self-healing and act as hands and feet. Closed-loop functionality is then required to provide information back to the brain to improve the model.

Figure 5: A self-healing analogy



Source: Charlotte Patrick Research

Creating this self-healing is one of the hardest parts of the autonomous network as it requires the creation of many nested closed loops which poll in near real time; fix the issues or request that orchestrators redesign the next best configuration; or otherwise, fix issues. This requires a range of ML models, knowledge planes, time-series databases and monitoring across the network to bring in data from multiple domains. Putting this together requires the deployment of an 'intelligence architecture'.

This architecture provides a blueprint for supporting the development and the deployment of the models and the federation of data that are needed for a telco to reach TM Forum's Levels 4 and 5 of its autonomous network model.

One of the discussion points around the development of a suitable intelligence architecture is the need for centralisation versus distribution of data, intelligence and control across the architecture. There are benefits and issues with both points on the spectrum, as shown in Figure 6.

Figure 6: Options for the intelligence architecture

	Centralised intelligence	Hierarchical intelligence	Distributed intelligence
It is…	• Centralised (or nearly centralised) • Focused in a single-point of control • Execution is also from a central point	• Concentrated at the top of the hierarchy with each layer of the hierarchy having particular tasks • There is a balance between centralised control and decentralised execution	• Pushed out across the network, edge and on-device • Leading towards autonomous agents that undertake both control and execution of specific tasks
Pros	• (Currently) gives access to more powerful foundational models via cloud	• Offers a trade-off with more intelligence in the centralised top levels of the hierarchy • Provides a workable day 1 solution (as described below)	• Agents can work together to solve complex problems • "Lives" in a local environment, understands local problems • Scalable • Can solve privacy issues of moving data
Cons	• Data has to be transported to the central location • Vulnerable to failure, attacks and overloads • Sophisticated models are difficult to create and manage	• Still some problems with bottlenecks and failure points	• Requires complex control techniques • More difficult to resolve issues • At the extremes of distribution, requires high-power compute at the edge. Leading to cost and sustainability issues
Best suited to…	• Complex problem solving	• General management tasks on the network	• Management tasks on the edge and very complex problem solving

Source: Charlotte Patrick Research

Centralised intelligence

One may conceptualise the idea of centralising intelligence as the creation of a giant telco brain that can perform multiple tasks, but there are many practical issues to this vision turning into reality: the know-how and tooling to build this brain are inaccessible to current foundation models, and a huge amount of computing resources would be required alongside novel software architecture. In addition, if/when such models do become available, AI models are good in narrow environments that can supply enough data; but training on multiple factors (e.g., technical and business intents) would be a challenge for an individual telco due to the limits on data availability. Additional barriers come from the tendency for models to drift, and the amount of time and compute power that it might take for a large model to learn something new (for example, when a new set of users are seen on the network or a new radio band are instantiated).

Furthermore, in complex systems such as a telco network, there is a general move towards decentralisation, as it becomes more affordable to place capabilities locally. Running counter to this is an argument that there may be a few places where centralising intelligence is beneficial – for example, where there are related tasks in the RAN that might benefit from a single model to avoid a network element receiving requests for data from multiple models simultaneously.

Distributed intelligence

At the other end of the scale is a distributed architecture. It provides a range of benefits:

- The complexity of a disaggregated 5G SA or 6G network, where every part is always in a state of change, may require a distributed intelligence that breaks down complex problems into discrete subtasks handled by specialised entities.
- Distributed systems offer a modular, flexible and resilient approach to automating tasks; being simpler to train and manage. Individual entities providing intelligence are easily swapped out, scaled or augmented. There may also be better explainability in the system as each entity undertakes a single task.
- They are also in line with architectural thinking such as open RAN.
- Intelligence is provided closer to the source of the problem – meaning data needs to travel less far and knowledge of the issue is stored locally.
- Privacy concerns over the movement of data across clouds, software and geographic regions are resolved.
- In addition, 6G brings new concepts around sensing and adapting to the changes in a network or its environment which may increase the need for distributed intelligence.

The starting point for this distribution of intelligence is the use of a MAS or an agentic system. This consists of multiple agents interacting to solve problems or achieve specific goals. In a hierarchical architecture, decision-making and control of the agents are most likely to be held at the top level. In

more distributed architectures, agents exhibit more autonomy and work together to leverage their collective intelligence, enabling them to address challenges that a single agent cannot manage alone. There are, therefore, many 'tools' which are used by the MAS; providing it with input information for decision-making, supporting the agents in their decision-making or acting as 'actuators' (carrying out instructions from the MAS).

However, there is a chasm to be crossed to achieve a truly distributed architecture:

- Neither MAS (using small or large GenAI models to power the agents) nor a simple agent system (using a single large model to understand requirements with a mix of models used downstream to make decisions and prescribe action) are yet capable of providing much distributed intelligence.
- As telcos use more GenAI, the benefits and challenges become better understood – this evolution is essential as in an autonomous system, the lack of reliability (and trust) is a major downside.
- Creating well-performing agents and good coordination between them requires significant work in areas such as conflict, control and error correction.
- Distributed systems can become very large and difficult to manage due to the number of nodes in the system and/or the amount of data needed to support them.
- There is a list of other potential issues which come from academic literature on agentic systems. One such example is the possibility of 'emergent behaviour' (complex and unexpected ways of acting emerging from interactions between agents).
- Distributed agents increase the attack surface which offers more opportunity for security failures or attacks within the distributed environment.

A description of the likely future moves by telcos, given all benefits and issues discussed, are set out in more detail in the final section of this report. However, we believe that a hierarchical architecture is the starting place for the telco move to MAS and may create a workable solution for many network processes.

Hierarchical intelligence

In the absence of the technology for a distributed architecture, telcos will need to start with the capabilities that they have in place and develop an intelligence architecture in hierarchical layers:

- **Top level**: Includes agents responsible for high-level decision-making, strategy and overall coordination. Typically, there is a single top-level agent, acting as a copilot and giving instructions to a range of agents or tools (defined as any system/model used by the agents) in the next layer of the hierarchy. This allows any intelligence using large models to be run in a centralised cloud.

- **Middle level**: These agents handle intermediate tasks, such as managing groups of lower-level tools and translating high-level directives into actionable tasks.
- **Lower level**: These agents/tools execute specific tasks, gather data and interact directly with the environment or the users. In the near term, putting intelligence at the network's edge is costly and less sustainable, so these lower layers of the hierarchy tend to include less intelligence.

The features of a hierarchical architecture are as follows:

- The network will have hierarchies created for different tasks – assurance, field services, operations, fraud, security, power management, RAN function management.
- A simple hierarchy may suit many basic tasks, such as diagnosing simple issues and providing suggested changes to an orchestrator. In this example, there would be a brain at the top, some data collectors providing data into a single pane of glass and some top-level control providing instructions to the orchestrators – with less intelligence at the lower levels.
- At the lower levels of a very large or complex network, the hierarchies will also be geographically aggregating – with lower levels being at the edge or at a radio node, aggregating to local area, then to province-level and then to country-level; or even a multicountry level for multinational telcos.

There is a closed-loop automation within the layers of the hierarchy (or up and down the layers) for improved cognition and training.

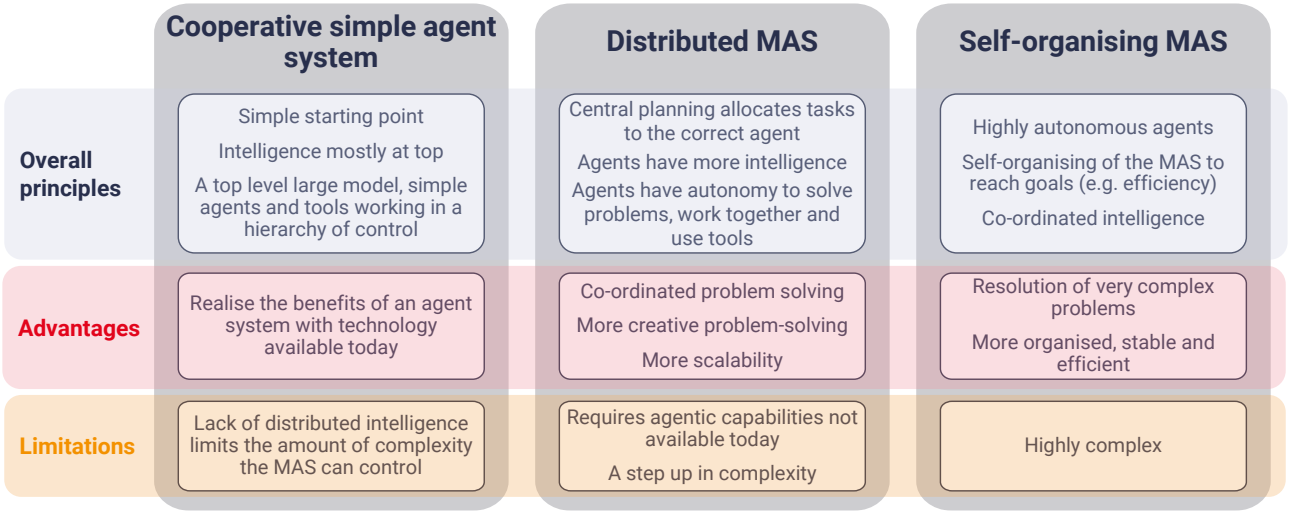
Creating an intelligence architecture

Developing hierarchical and distributed intelligence architectures

It looks likely that telcos will look towards implementing a MAS in the 5-10 year timeframe, at least for some if the more complex network processes.

Figure 7 shows three phases of increasing intelligence in the agents – each brings additional capabilities but with multiple trade-offs around cost and complexity.

Figure 7: Types of simple agent systems and MAS



Source: Charlotte Patrick Research

Cooperative simple agent system

As discussed in the previous section, telcos will start by creating a hierarchical intelligence architecture. This will require a cooperative simple agent system where relatively simple agents will work towards a common goal, mostly instructed by a copilot and, possibly, a reasoning engine, at the top of the hierarchy. However, some agents will start to build intelligence and support their own goals over time. In assurance, this is most likely to be the monitoring of a particular domain where the agent will undertake simple tasks such as proactive highlighting potential problem resolutions to orchestrators in that domain.

Distributed MAS

In this next part of the journey, agents sit within a network which may still have some hierarchical characteristics, but they will be more independent. This brings more autonomous decision-making and agents start contributing to collective goals, learning from past interactions and optimising their actions based on feedback from the environment or other agents. Tasks are allocated based on agent capabilities or availability by a centralised task management system. Communication is more

sophisticated and may include negotiation between agents and sharing learned knowledge, but multiple issues around control, conflict resolution and decision-making efficiency begin to appear.

Self-organising MAS

The agents in a self-organising system are significantly more sophisticated, and the MAS can autonomously adapt its structure or behaviour to changes in its environment or internal conditions without external intervention. It uses feedback loops to evolve towards higher organisational, stability and efficiency levels. Self-organising systems are dynamic and can reorganise their internal structure and processes in response to or anticipating environmental changes. They prioritise adaptability where the organisation of elements or agents within the system evolves towards optimisation or new functional states, often through mechanisms such as self-regulation, competition and cooperation.

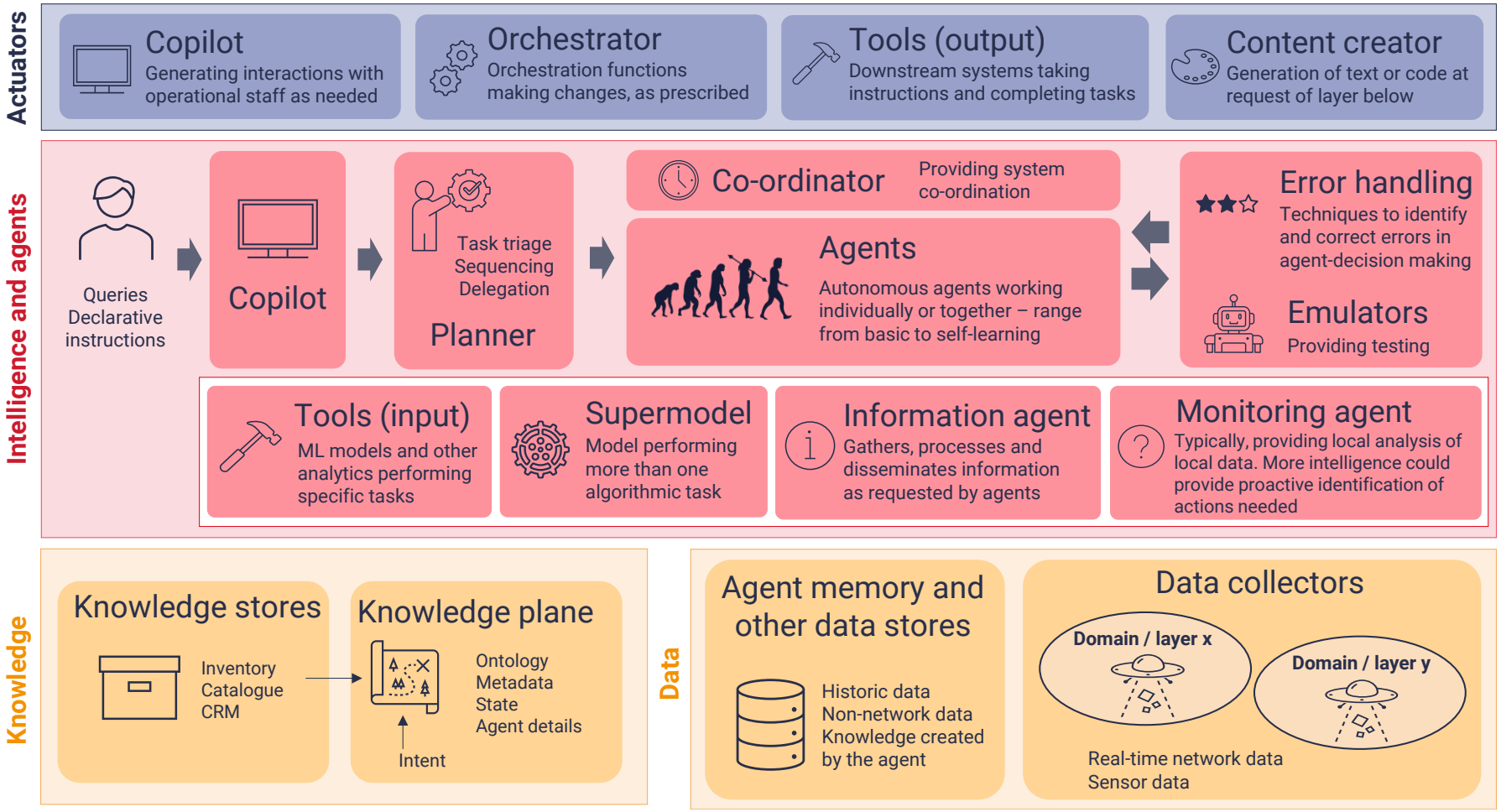
The main components of a MAS

A self-organising MAS is some distance in the future for a telco – and whether it will ever be needed at all remains to be seen. Figure 8, therefore, describes a range of requirements for a coordinated simple agent system or distributed architecture using a MAS.

Not all elements drawn on the diagram are required to provide a MAS, with some only needed in more complex distributed architectures – and a discussion on this topic will be found in the next section.

Please see the Appendix section for description of each of the elements on the diagram.

Figure 8: Elements of a simple agent system



Source: Charlotte Patrick Research

The development of assurance capabilities using a MAS architecture

Assurance is one of the first areas to see an intelligence architecture being considered and deployed.

MAS designed to support assurance requires:

- Measurement agents in individual domains/layers, bringing together data as needed to feed intelligence further up the hierarchy. These agents may require a degree of intelligence to understand the performance of their domains, identify potential deviations from KPIs which suggest service-impacting problems and proactively alert other agents or models further up the hierarchy.
- The use of a reasoning engine and/or individual assurance agents – typically performing observability and root cause analysis tasks. For example, a domain agent may see an issue and report back that it is not a problem in that domain, requiring the centralised intelligence to poll other domains for insight to understand the issue.
- Centralised intelligence responsible for providing prescriptive actions. This may include gathering information from systems unrelated to assurance, for example to prioritise actions based on their potential business impact.
- Actuators, including copilots alerting humans to events that need their action and updating humans generally on network and service performance.

Decision-making around development of an intelligence architecture

There is a lot of complexity in developing anything, except for a very simple hierarchical system with centralised intelligence. There are also trade-offs between the benefits of more complex MAS versus the increasing cost and problem solving needed.

Figure 9 below summarises some of the main decisions to be taken by a telco in the early days of MAS deployments and current industry thinking on these topics. A more detailed discussion of what each decision path entails follows after the diagram.

Figure 9: Thinking among interviewees on four areas of Intelligence architecture

Do we need a MAS?

- Enthusiasm differs – some industry participants are focused on moving towards sophisticated agentic systems.
- Others are unconvinced about the mid-term prospects for trustworthy large models and are betting on the use of reasoning engines supported by knowledge bases; adding a variety of AI/ML around them.

Note: A (machine) reasoning engine interacts with one or more knowledge bases to take in a view of the current expected state of the network; alongside a range of network entities (e.g., ML models) and copilots or other systems providing a view of intent. It continuously tries to find actions to close the gap between the current observed state and the desired state.

How much distributed intelligence is useful?

- The concept of agents with specific skills (whether using LLMs, SLMs or other AI/ML) offers a way for telcos to conceptualise their intelligence architecture.
- The timeline for deployment of more agentic systems is likely to be long – perhaps 5-10 years. And then, they may only be deployed to tackle the most difficult automations.

Is a knowledge graph necessary?

- The majority of participants considered a graph to be the way to store knowledge about the network.
- Uses would be as a single source of truth to underpin decision-making and retrieval-augmented generation (RAG).

Is a digital twin necessary?

- Most agreed that a digital twin was needed to support agents with testing of more complex decisions.
- However, a complete twin of the network was unlikely to be cost-effective and the instantiation of smaller twins as needed, more realistic.

Source, Charlotte Patrick Research

Do we need a MAS?

As GenAI heads over the top of the curve in the Gartner hype cycle, the current limitations of foundational models are increasingly well-understood by telcos. Research from six Apple employees

in October 2024[1] supports their anecdotal findings that current versions of these models can be untrustworthy and difficult to work with. During this examination, we noted three approaches to building agentic systems:

1. Vendors and telcos, particularly in Asia, have deployed simple agentic systems to support use cases such as major event assurance. They demonstrated the most enthusiasm for more complex MAS, with a longer-term strategy to create increasingly sophisticated systems as their learning increased and technology advanced.

2. Other interviewees had more limited expectations of MAS. Instead, they were trialling large models to ingest declarative instructions but focusing on building reasoning engines that are not based on LLMs and knowledge plane/graphs to support their moves to Level 4. They expected to build out MAS in the longer term, using agents to augment their mix of reasoning engines and knowledge planes, as necessary.

3. There was also a good number of respondents who were less far forward in their thinking and were looking towards a MAS without having either trialled or deployed any.

The significant barriers to creating anything more than a simple agentic system will stall progress towards more sophisticated systems into the next five years. However, one interview participant noted, "we can't just improve what we have today and expect to achieve autonomous networks at Levels 4 and 5". Deploying a MAS seems to offer the most likely solution – providing new abilities to break down complex problems into discrete subtasks handled by specialised agents – and if it can be deployed cost-effectively, this would offer a flexible and robust intelligence architecture for the network.

Where interviewees' current viewpoints differed was in the ability of GenAI to enable a complex MAS in a workable timeframe. For some, a reasoning engine offers an early-day solution to creating more adaptive intent-based automation – without getting caught up in the complexities of delivering a workable MAS.

A pragmatic starting point would seem to be:

- Think 'agent' from the beginning – much like open RAN, it provides a way of thinking about an intelligence architecture which is distributed, virtualised and disaggregated.

- Implement a copilot as a starting point for each new process that is to be moved to this agentic thinking; enabling knowledge of deploying foundational models in the area to be built up.

- Add the most simple and stable intelligence possible to move the process towards autonomy – this may well be a non-LLM-based reasoning engine.

[1] LLMs Can't Reason and Plan | AIGuys

- Use the basic modular approach of agentic thinking to allow this intelligence to be upgraded easily in future.
- Upgrades should be considered on a case-by-case basis:
  - More straightforward decisioning (for problems typically solved by Level 1 NOC engineers or straightforward automations) may continue to use more simple intelligence solutions.
- More complex decisioning for non-standard or multi-step problems (analogous to a Level 3 engineers' problem solving) is a good candidate for testing with agents using foundational and hybrid AI.

How much distribution of intelligence is needed?

As discussed previously, a distributed MAS offers the idea of agents which make independent decisions and contribute to collective goals. There is also a sense in which the distribution of intelligence is geographic, placing it as needed in the network to ensure that significant amounts of data do not have to be moved across the network.

The general idea of distributed intelligence and the use of the term 'agents' seem set to take off as they offer a practical way for those involved in creating an intelligence architecture to discuss what type of intelligence they require and where it needs to be placed. However, the time frame in which we will see agents becoming capable of more independent decision-making and other agentic features is unclear because of the current major shortcomings of LLMs and the cost of adding new capabilities to prevent LLMS from being a significant drag on deployment.

Geographic distribution is not an important consideration when network functions for multiple domains are in the same data centre. However, even moving large amounts of data around a single data centre may be burdensome. There are also specific use cases (such as the addition of agents at a base station) which might require geographic distribution, but this is not currently considered the most pressing agent use case. In ten years' time, there may be other geographic distributions with agents in devices or on customer sites – but currently, there are probably no immediate use cases for this.

Is a knowledge plane necessary?

A knowledge plane stores a variety of data including from across the network (such as codified domain expertise, product documentation and external data sources). There is general agreement that deploying a knowledge plane will be necessary for a MAS because:

- It will become the central place in which a telco would store its valuable and specific knowledge of its network;
- The stored knowledge is codified and federated, making it a source of truth for stable deductive reasoning;

- Identify the part it could play in the training and retrieval-augmented generation (RAG) processes for foundational models.

However, there are some significant difficulties with building an always up-to-date representation of such a large system (particularly with the complex multi-vendor environment and need to federate information from several data repositories) – and it is likely that building a perfect view of all permutations to all problems was unrealistic

A stable collection of knowledge will be crucial to future deployments of agents and reasoning engines and will require telcos to build some type of knowledge plane. An imperfect starting point will provide some benefits and, when used alongside different agents/models, there will be a synergistic improvement of both over time.

Is a digital twin necessary?

There was nearly an agreement at the interview that using digital twins to emulate the prescriptive outputs from either a reasoning engine or agents would be part of future processes within an autonomous network. It looks likely that more simple changes could be made without the expense of a digital twin; but that instantiating a twin for a specific emulation and running it close to the decision-making (rather than having one single mega model of the whole network) could be a cost-effective way to reduce risk and improve automations.

Conclusion

Developing an intelligence architecture for assurance is a major part of the move towards a self-healing network. It enables:

- A move from individual assurance solutions focused on a particular piece of network equipment or a specific domain towards a single end-to-end, cross-domain view of a service or the experience of an individual customer;
- The ability to collect all necessary insight for a variety of uses – from root cause analysis to more automated self-healing – and process it in an efficient manner;
- The provision of information to humans who are expected to remain in the loop for some time;
- The provision of an architectural blueprint that helps to reduce technical debt and costly integration by including different AI/ML models (many of which will already be deployed);
- An architecture which will spread across domains, horizontal stacks and out into customers, partners and ecosystems in the future; ffering solutions to future issues of automations that reach beyond the telco network.

Creating an intelligence architecture that supports assurance, as highlighted in this report, is not a straightforward exercise. At its core, it requires the federation of data that will support a range of agents and models, and then the development of intelligence that is able to extract the necessary insights from a very wide range of data types.

Over the last few years, the market has seen specialist vendors developing these capabilities, requiring them to bring multiple new capabilities around ML, data management, ontology and knowledge management, as well as use a range of telco-specific skill sets to enable swift and successful deployment. Telcos should consider their own needs in these areas as one of the first steps to developing intelligent architecture. Otherwise, they risk lacking a solid foundation on which to develop new architectural elements.

Appendix

The main components of a MAS

Error! Reference source not found. in the body of this report describes a range of requirements for a c oordinated or distributed architecture using a MAS. In this appendix, we provide more details about each of the elements drawn on the diagram.

Knowledge

The bottom layer of the diagram describes two sets of technologies needed to support a MAS. The first box discusses the provision of knowledge. This will tend towards being a centralised service in the intelligence architecture.

Knowledge plane/graph	Created from a variety of data including machine learning insight from across the network, codified domain expertise, product documentation and external data sources. Options include the use of a graph database, but other proprietary technologies are seen. The knowledge plane provides information about the production network; showing entities such as cell towers, network devices, customers, service providers, and their relationships. It is used for topology management (representing the current physical and logical structure of the network), service/subscription tracking (mapping services provided by the network and customer subscription), troubleshooting (provides view of affected elements and their dependencies) and predictive analytics (provides data for prediction of future network incidents). It also provides an ontology of intents; holding information on intents and their relationships.

Data

The second box on the bottom layer looks at the provision of data – this can be both centralised and decentralised, as required by the intelligence elements in the network.

Memory and other data stores	Includes any stores of network or non-network data used by the MAS, they may be linked into a data mesh or used by an individual agent/tool. For example, an individual agent will require access to local data such as historical trends which it reads/writes to its “memory”; it may also call on data from other stores elsewhere in the intelligence architecture (including digital twins).
Data collectors	Collecting observations from the environment and ensuring that the intelligence layer above has the correct insight to make decisions. Collectors can sit at any layer and in any domain of the network, and may extend to the edge on a user device. Examples include probes, sensors, extended Berkeley Packet Filter eBPF and on-device monitoring.

Intelligence and agents

The middle layer of the diagram describes a variety of intelligence sources. The amount, type and location of intelligence (centralised vs distributed) will depend on the sophistication of the intelligence architecture (more cooperative vs distributed) and the task being undertaken. There are four elements which provide input into the intelligence and agents above:

Tools (input)	The use of analytics or machine learning in the network by the MAS. These solutions may provide specific analysis such as diagnostics for a particular network domain or make simple predictions used in more sophisticated decision-making further up the hierarchy.
Supermodels	Discussion remains around the possibility of creating models which could undertake a range of algorithmic tasks. Developing really complex models is prone to be difficult and costly; but we will most likely see some specific models where tasks are related (e.g., control of the RAN or in planning).
Information agent	Gathers, processes and disseminates information within the system – includes database-query agents or agents that aggregate data for decision-making
Monitoring agent	This could take on a variety of tasks and include very little or quite a lot of intelligence. An agent with less intelligence will return data from a particular domain to other agents on request. More intelligence will enable it to handle simple tasks, such as turning real-time data into more structured information to handle the request. It may then be quite independent and goal-driven - able to respond to environmental changes that it observes

The next set of elements in the top of the layer use this input:

Human	Provider of declarative intents for the MAS and resolver of issues considered too sensitive to leave to machine intelligence.
Copilot	The copilot acts as a bridge between humans and machines – taking in declarative statements and providing information, as needed. Its use of GenAI also allows it to create documents, summarise and translate between languages.
Planner	In a more sophisticated distributed MAS, tasks arrive with the planning functionality – either from the copilot or from other agents in the system.

	The planner maps out the sequence of actions needed which could include data collection/analysis and the formulation/execution of strategies.
Agents	Agents use a range of intelligence capabilities (from basic RPA to large models) to comprehend and respond to inputs. They may take instructions from a range of sources, including other agents, perform decision-making, prescribe action and determine when to call on external tools to complete a task. More intelligent agents can learn from their environment, other agents and past actions. Each agent has a specific set of capabilities and, in a MAS, will coordinate with other agents to provide distributed intelligence (potentially using ensemble learning techniques as described above). Appendix 2 below describes a range of agents, becoming more intelligent as the list progresses.
Co-ordinator	The provision of coordination models and mechanisms to improve coordination between agents, avoiding conflicts and ensuring agents don't work at cross purposes or in efficient ways.
Error handling	Models used to correct errors within the agent system, they identify the issue and offer resolution. These are assumed to be used in more complex MAS where individual agents are not able to "see" errors occurring across the system.
Emulators	An "emulator" or digital twin is used by the agents to test the suggested solution on a simulation of the production network. For example, testing of a new routing protocol to evaluate its potential impact on latency and reliability. It also has potential future uses in the training and improvement of models – for example, it could estimate what information is missing about the state of the network for the reasoning engine/agents to run successfully.

Actuators

The top layer of the diagram describes several possible actuators for the MAS. These take instructions and data/information from the MAS and ensure that the task is completed.

Copilot	Used to provide information to humans about the decisions taken at the 'brain and agents' layer. For example, a NOC engineer receives a summarised email showing all problems occurring in the network and the decisions taken for resolution.

Orchestrator	Makes changes in the network via script-based automation.
Tools (output)	A variety of entities carry out tasks from the reasoning engine/agents. Requests arrive as information, data, and prescriptive commands and tools could be selected from the operations support system (OSS), business support system (BSS), network functions, customer-facing application programming interfaces (APIs) or other telco systems. In the future, they may also be systems/automations outside of the telco.
Content creator	An LLM-based tool creating content such as release notes for an upcoming network build or writing code for a new API required.

STL PARTNERS

Research

Consulting

Events

www.stlpartners.com