International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

Closed-Loop Network Automation: Architectural Principles from Catalyst Center-Scale Systems

Manevannan Ramasamy

Software Engineering Senior Technical Leader, Milpitas, California, United States of America

manevannan3@gmail.com

ABSTRACT: Enterprise controller platforms must perform operational loop bridging a large and heterogeneous environment of devices, clients, locations, applications and services and must maintain explainability, reliability and safety of changes. The paper is creating a vendor-neutral architecture layout, founded upon the publicly accessible scale attributes of Cisco Catalyst Center, and recognized best practices of intent-based networking, telemetry, assurance and autonomic regulation. The provided framework divides the closed-loop functioning into 6 synchronized planes that comprise: data gathering and contextualization, correlation and diagnosis, intent and policy management, workflow orchestration, outcome verification and external combination. It also introduces hierarchical range of control, evidence amalgamation, categorization of actions and control-budget system in place to keep the remediation activities in stand to reduce the risk of operation. Catalyst Center is considered to be an industry observation source besides source of inner proprietary application and as such the framework might be autonomous of proprietary internals. The framework is evaluated by such criteria as outcome coverage, telemetry freshness, and diagnostic accuracy, control-loop convergence, failover resilience, as well as operator understanding. The resulting architecture provides a procedural basis to the architecture of scaleable enterprise controllers that moderate automation and assurance, explain and guided remediation. The donation is an informed vendor-neutral and product-aware model of trustworthy closed-loop business operations.

KEYWORDS: Catalyst Center, Closed Loop Automation, Enterprise Controller, Assurance, Intent, Scale

I. INTRODUCTION

Enterprise networks have grown to form clusters of individually controlled devices into highly distributed operation spaces, including switches, routers, wireless, clients and applications, services and multiple management areas. With scale and heterogeneity of these environments comes growing challenges of achieving device-by-device configuration in view of demands of high speed service delivery, constantly assured and operationally reliable, and safe change management. Contemporary network management is thus becoming more oriented towards automation to be capable of monitoring network status, derive desired conclusions, coordinate activities and determine whether the activities have had the intended effect or not. This transformation characterizes the need of controller architectures, which may be used in closed-loop operation, not restricted to remote automation functions.

This automation can be traced back to the autonomic networking domain where self-configuration, self-optimization, self-healing and self-protection were considered as natural events in network systems [3]. Autonomic operation introduces the concept of the feedback between condition of the network and functions of management where the systems are able to evolve in the new changing environment as compared to the implementation of the preprogrammed procedures. The other commonality between the approach is its high level orientation through intent and network wide reports that form a dramatic distinction in conceptualization between the operational goals and low-level configuration processes [3]. This is needed in the case of large-scale controllers, where the result that its intended result or behavior is to achieve can be obtained with different actions depending on the topology, the capabilities of the devices, work load conditions, and the current state of the network.

This is further extended to include zero-touch network and service management through a service-oriented architecture to management through automated means. The SO definition of the ETSZSM reference architecture provides the tenets of the architecture, logical blocks of functionality and end-to-end management interfaces of the network and services [1]. It is created with modularity, scalability, interoperability and separation of management issues in mind and allows it to be automated across domains and technologies. These principles are applicable particularly to enterprise controllers which must organize, align, guarantee and streamline processes. However, functional decomposition is not

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

everything to realize large-scale closed-loop control. It too requires the mechanisms of determining the nature of what should be regarded as dependable evidence of the observations and how should the sequence of observations be matched and in what situations the automated action is permitted.

Telemetry hence becomes a major aspect in the control loop. Network telemetry provides operation information acquisition systems that can be used to determine an insight into the prevailing network activity, as well as to make sound management choices [4]. Telemetry is not to be thought of as a flow of measurements in a controller architecture. It is applicable according to the aspects of freshness, provenance, identity, topological relationships and contextual association. A controller having a vast unconnected telemetry can provide a huge amount of data, having less diagnostic ability. The scale of automation in the enterprise must hence be designed in a manner that converts raw observations into evidence context that is to be utilized in the diagnosis and decision making.

A second layer which is imperative in intent based networking is that one can describe desired network behavior in a more abstract way. The intent-based systems enable operators/external systems to express goals and desired outcomes without explaining all details related to the implementation on the device level [5]. This abstraction enables the controllers to match up operational goals to policies, workflows and configuration activities that fit the underlying environment. Intent also provides a standard over which the results of the operations may be assessed. Hence intent need not be limited to initial configuration, it could be the desired-state representation within all of a continuous assurance loop.

Autonomic principles, management by zero-touch, telemetry and intent combine and are the conceptual basis of the closed-loop enterprise control. A generalized loop can be viewed as a sequence of observe, contextualize, diagnose, decide, act and verify functions. The observation provides facts about existing states, contextualization provides facts to objects and relationships, diagnosis furthers excusable deviation of conditions, decision functions, which interpret motive, policy performance, execution, which is the execution of the authorized actions and verification whereby the resultant state is identical to the proposed condition. This kind of structure enables the controllers to transcend event based automation to the continuous and evidence based operational management.

However, there are certain problems with large-scale environments. There can be a multitude of loops running on device, site, service, application and enterprise levels. The observations on one loop, or actions having a result, may influence observations on another loop, leading to conflicting, or excessive remediation. Massive controllers must now consider the loop hierarchy, the scopes of actions, concurrency, the quality of evidence and risk of change rather than simply considering individual detected deviations as separate triggers. It needs to be explainable as well since operators must understand why an action was proposed or executed, and what evidence was taken to support this action and whether the resultant position is adequate to achieve the initial target.

The challenges identified in this article are dealt with in the context of a vendor-neutral architectural framework based on publicly visible capabilities of controller with regard to the enterprise and set networking principles. The framework outlines the data gathering and surroundings, correlation and diagnostics, willpower and policy, workflow implementation, outcome validation and integration outside. It also introduces hierarchical loop scopes, aggregation of evidence, action classes and a control budget model (capacitance) constraining parallel remediation. These systems will contribute to safety of change and responsiveness.

The resulting perspective views enterprise controller as an closed-loop control system, where evidence, intent, execution and verification is clearly related to one another. The framework offers a framework to assess enterprise controllers based on outcome cover, freshness of telemetry, diagnostic quality, convergence, failover behavior and understandability by operators. In so doing, it seeks to bridge the gap between individual automation abilities and centralized architecture of clarifiable, scalable, and manageable actions of an enterprise network.

II. RELATED WORK

The autonomous enterprise networking has gradually shifted network management operations out of non-moving, manually imposed processes that involve intent execution on telemetry and continually implemented control loops. Modern autonomic networking, intent-based networking (IBN), network telemetry, assurance, and service orchestration with zero-touch management are the building blocks of this transition. All these research and standardization efforts can be seen as the conceptual foundation of the enterprise controller architecture as discussed in this work.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

There is a good architecture foundation to autonomous network- and service-management: zero-touch. ETSI ZSM reference architecture splits the functionality of the management in both domains and supporting services and prioritizes automation, coordination and closed-loop management of the network as well as service environments [1]. Of particular concern here is its specialization in modular management services and cross domain coordination which can make many different technologies, locations and administrative domains at cross the line in terms of decision paths in a large enterprise controller. An architecture of ZSM in general is, however, not prescriptive in itself regarding how evidence is to be combined, how remediation is to be constrained or how operators are to regard the rationales of automated responses.

Controllers of modern enterprises are also exposed to automation via APIs. Cisco Catalyst Center provides programmatic interfaces to access controller capabilities and network-management capabilities [2]. These APIs illustrate how an external integration layer can be important where applications and more elaborate orchestration systems can utilize network-based information or initiate management processes. With a closed-loop architecture at an enterprise scale, however, one of the only parts of the control system is API exposure. Reliable automation must also require policy interpretation and explanation, contextual information, choice of action, execution protection and action follow-up.

Self-managing systems were conceptually based upon autonomic networking. The autonomic networking design principles put emphasize on the systems that can check on the states of their operation, analyze the data collected, generate responses to them and carry out those responses as much as possible without human intervention [3]. This state of control closely aligns with the structure of feedback-loop that is necessary to carry out enterprise network automation. The autonomic model is however generic in the sense that the contemporary heterogeneous environments with clear business purpose, a collection of telemetry, diverse range of management, and potentially a dispersion of activity that cannot be remediated nonetheless, necessitate alternative mechanisms.

Evidentiary base that these feedback loops require can be obtained through network telemetry. The network telemetry model outlines mechanisms, and architecture concerns that play a role in the collection, and movement of operational data of network infrastructure [4]. Early warning telemetry may help the controllers develop a dynamic image of the environment and to know whether the desired conditions are being met. Freshness of the telemetry and completeness in the context is hencecure useful to allow the closed exploitation of the enterprise as languid or unsynchronous appearance may lead to inappropriate diagnoses. The framework below is an expansion on this line of reasoning but telemetry is seen as a piece of evidence that must be connected to topology, identity, policy and operational context to consider automated action.

Intent-based networking Newest layer The semantic layer specifying the desired results and the network behavior. The IBN concepts and definitions have intent while based on desired operational results as the expression of intent and the distinction between intent and mechanisms that achieve desired operational results is brought forward [5]. This is what enables operators to instruct the network of what to do by stating what the result should be as opposed to listing out all the low-level configuration actions. The intent-policy disengagement, workflow execution and outcome verification proposed are therefore provided with a logical foundation by the purpose-based architectures. Nonetheless, the operationalization of intent into feasible aspects and taking evidence, dependency, and conflict and operational risk into account are vital in practical enterprise control.

Assurance research also supports the closed-loop model with respect to connecting intent fulfillment and continuous observation. Network assurance approaches that use machine learning explore how network behavior can be detected to deviate when it is observed in telemetry and history [6]. These approaches demonstrate how monitoring is not a solitary feature which ought to be joined with diagnosing. Their focus on forecasting or anticipatory certainty is yet another driving aspect in assessment such as diagnostic quality, telemetry freshness, and convergence. In order to extend this principle, the given architecture enacts evidence aggregation and explicit verification, after the working process was run.

The other important feature is intent-based resource orchestration. The possibility of high-level requirements influencing functional placement and resource allocation of virtual networks has been explored using multiple-tenant intent-based approaches [7]. These approaches demonstrate that intent processing is not restricted to configuration in order to decision making that is sensitive to resources. Similarly, interest-based compilers of scientific networks have tried to compile high-level specifications into automated network configuration with regard to performance

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

specifications and network conditions [8]. All these works confirm the fact that a translation of the declarative requirements to the specific operational actions, is possible, only that it is applicable to specific deployment or resource management scenarios, and not to a generalized enterprise controller cutting across a range of operational loops.
Allocating tasks on end-to end services brings intent-based control to heterogeneous platforms. The intentions based orchestration model of multi-platform environment may be applied to demonstrate how service requirements can be mapped to coherent activities in the different technological platforms [9]. The identification of intent based network slicing also justifies how imposition of declarative requirements can be implemented to the SDN based spaces without logical isolation of resources [10]. Such contributions support the workflow orchestration and scope of hierarchical control requirements in enterprise contexts with different domains possibly having different controllers, policies, and remediation domains.

Lastly, intent refinement research deals with essential refinement of the high-level goals to operationally relevant network requirements [11]. Refinement minimizes the semantic distance between human goals and policies that can be run on a machine, and is a significant precursor of safe automation. Intent definition, telemetry, assurance, orchestration and execution are frequently discussed as independent issues in the reviewed literature. The resulting gap is the absence of a single, vendor-free architecture with a clear connection between contextual evidence, diagnosis, intent, policy, controlled execution, verification and external integration in scalably hierarchical loops. The framework proposed in this article addresses this gap, by adding such functionality, and further extending that with the concept of evidence aggregation, action classes and the control budget, to avoid concurrent remediation at the expense of operational explainability and change safety.

III. ARCHITECTURE: MULTI-SCALE EVIDENCE-TO-ACTION CONTROL FABRIC

The architecture presented is to be used on enterprise controller platforms which should work on heterogeneous devices, clients, sites, applications, services, policies and management domains. With the representation of scale as a multidimensional property, rather than scaling depending on a parameter on the number of devices, the architecture expresses scale, but distributes both control duties across layers and hierarchies of loops. This technique connects the facts of action and intention, diagnosis, controlled practice and confirmation of the outcome and keeps well-definition of concurrency, authority and failure recovery.

3.1 Multidimensional Scale Model

The resolution of the scale in controllers is determined in accordance with different dimensions of workloads interacting. The number of devices impacts inventory size, configuration tapered fan-out, command and state reconciliation steadfastness. Client population increases cardinality of events, identity change, events of mobility, and authentication-observation. Site population brings about geographic partitioning, delegated administration, variance of latency, and bounds of failure domains. The rate of telemetry consists of the rate of ingestion, temporary buffering, storage, and analytic load. Applications, services and policies increase relationship density because there might be multiple dependencies and goals that an individual entity can be a part of. The pressure of execution is determined by in parallel workflows, diversity of hardware, software, protocol, and configuration place pressure on the size of planning branches.

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

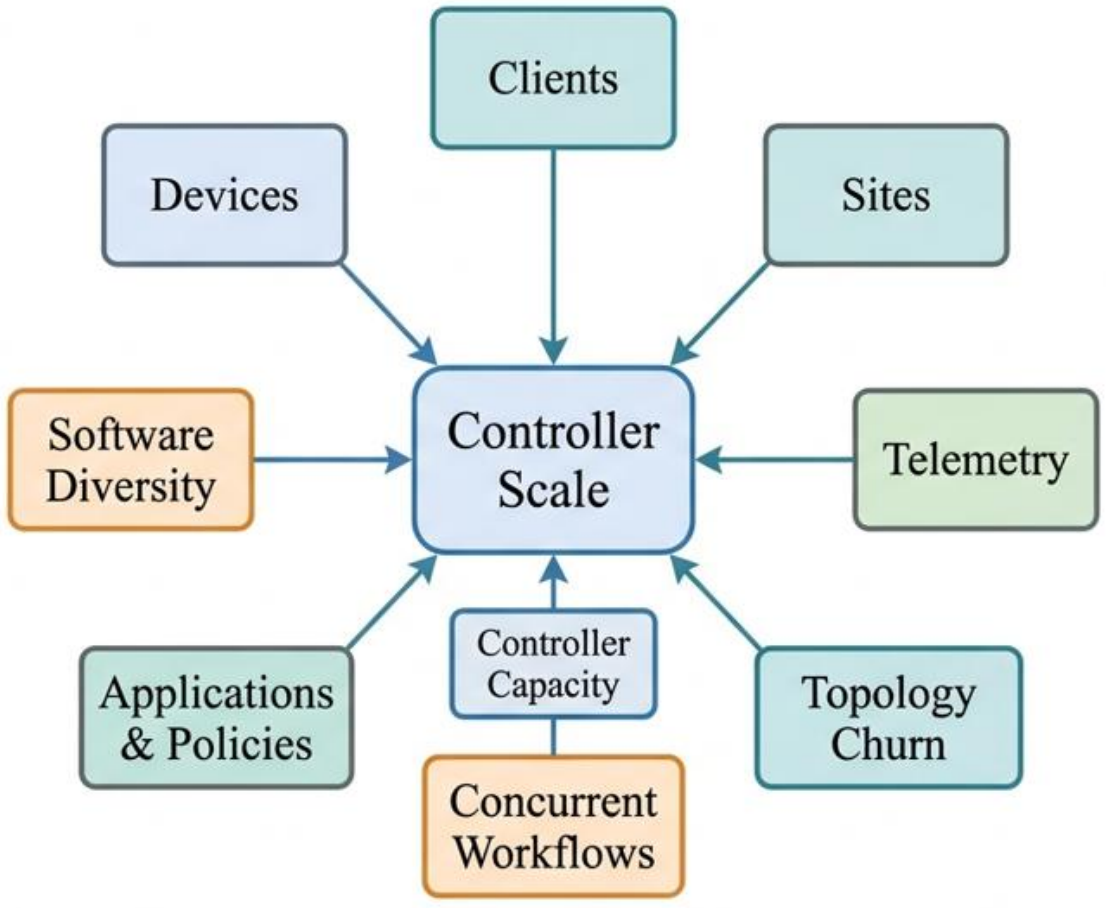


Figure 1- Multidimensional Enterprise Controller Scale Model

These dimensions cannot be expressed as a single metric in the number of the number of devices that are under control. A fairly small set of clients can overwhelm an operation load on a comparatively small set of devices even when a large inventory is contended by a controller: a burst of events, such as an event burst; a change of policy, such as a large configuration fan-out. The individual architectural services should in turn reveal capacity quantifications, comprising ingestion rate of telemetry, event backlog, graph-processing latency, and workflow concurrency, and verification delay. Another method of making sure that a single operational condition is not taking up too much platform capacity would be through resource budgets at the loop-level.

3.2 Layered Closed-Loop Architecture

The architecture is organized into six functional layers that collectively implement an evidence-to-action feedback cycle.

Collection and context layer: The layer accepts telemetry, logs, flow records, synthetic test results, inventory information, identity data, and addressing information, topology state, and controller task events. Incoming observations are normalized to such a state that they are identity-, time-, source-, and event-seminated and the source evidence can be easily analyzed. Collection resolution can be purpose of operation geared. Sampling that is economical may be used on monitoring a baseline, diagnostic cases may be used to improve the resolution of a collection and forensic cases may be used to temporarily preserve a more detailed evidence. This adaptation plan will help in getting sustained high resolution acquisition that will not burn resources where the need to explore relevant conditions arises.

Correlation and diagnosis layer: Decoding raw observations into operational findings through correlation of events that are influenced by user, application or service output. A temporal dependency graph is used to join the clients,

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

access infrastructure, controllers, policies, sites and applications. Deterministic rules are complementary to each of the other, as well as statistical baselines, anomaly models and topology reasoning, as opposed to applying a single mechanism of analysis. The results offer the confidence, evidence and contrasting results. This representation is important because the circumstances of enterprises can provide too many potential connections and an automated control will identify high-confidence outputs and unclear situations.

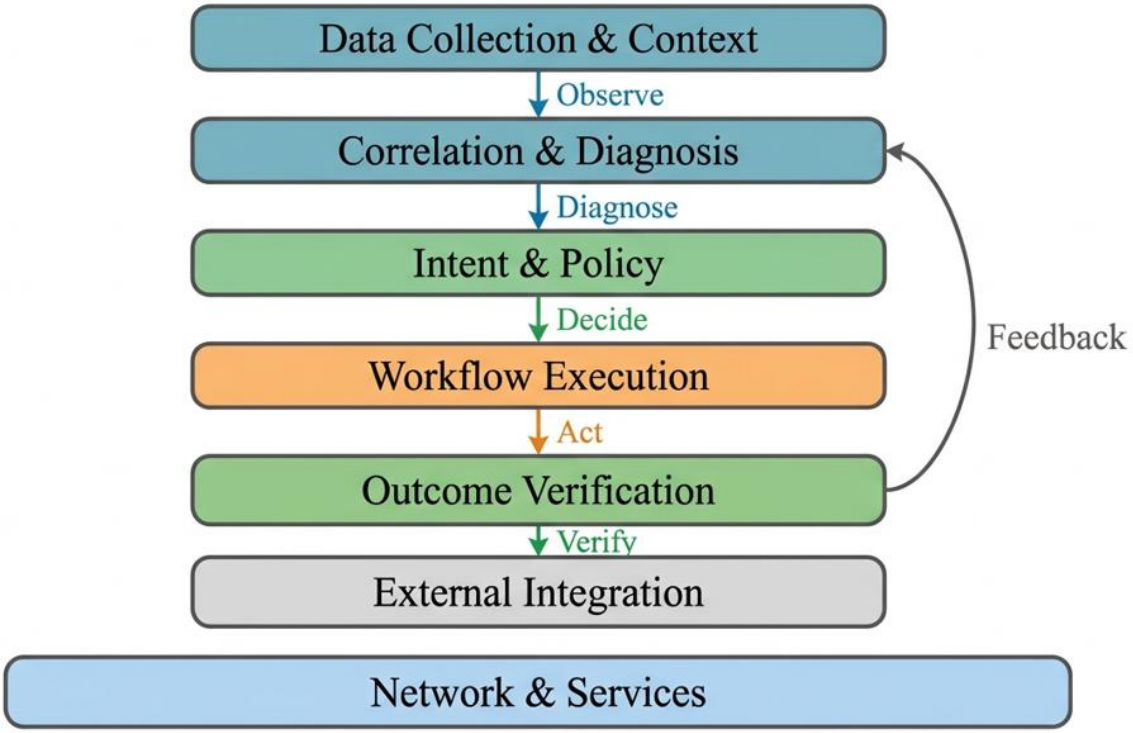


Figure 2- Layered Closed-Loop Enterprise Controller Architecture

Intent and policy layer: This layer is the desired outcomes, extent of operation, license and service objectives. Intents would provide desired-state reference on which operational state can be assessed. RFC 9315 dwells upon the importance of modeling both intent and operational states and comparing them in the environment of intent-based systems [5]. The architecture provided is expanded on this thought to add intent, explicit ownership and scope, authorization, and constraints of action. The layer determines what constitutes intent drift in a diagnosed situation and appropriate responding recommendable within the confines of pertinent policy and authority.

Workflow execution layer: Approved plans get translated to ordered, durable and preferably idempotent actions. Adapter devices and service adapters abstract differences between underlying technology and the execution checkpoints, dependencies and retries and partial-completion states are maintained by the workflow engine. Reversible efforts have compensation system and efforts are revocable. Workloads are separated depending on the site, ownership of resources or other regions of failures. Normal maintenance, repair, and more significant operational responses can have distinct and actual concurrency constraints to ensure that normal automation does not steal capacity.

Verification layer: Verification determines whether the original operational aim has been restored or not. The controller observes a settling period and examines the primary outcome and supporting objectives after the execution. The resultant is resolved, changed or inconclusive. This disparity cannot enable effective performance of command to be flipped with effective service recovery. An unsuccessful configuration command having terminated without giving the desired service state should therefore not be an actual successful configuration command but must be an open control loop.

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

Integration layer: Findings, objective indicators, workflow states and chosen operational information are provided to authorized external systems via versioned APIs and event interfaces. Results may be subscribed to by outside systems, creating events, request accepted processes, or offer more context. Intention of admission and policy is preceded by integration. Consequently, this means that an external API request cannot bypass authorization, classification of actions, or control-budget.

3.3 Hierarchical Control Scopes

Not every decision made in the enterprise takes only one centralized loop. The architecture is then a means of structuring control into scopes of control. Localized process, interface or connectivity conditions are solved on the device level. Site loops are loops that plan the access coordinate, wireless and local service behavior. Domain level loops are concerned with finer fabric, identity, software or assurance policies. A global loop verifies results which cross a site or domain boundary.

Each of the loops emits either a summarized state to its parent or constraints to higher-level goals. The ownership is clear a local loop can work under the assigned authority but the upper-level loops can guide the collective utilization of resources and inter-domain objectives. Time of control-loops also differs. Taxing loops have the capability to deal with protection, and collection remedy, while slower loops have the ability to deal with optimization, plan and planning and policy symbolizing. Versioned state guarantees a more recent decision will not overwrite a more recent local correction.

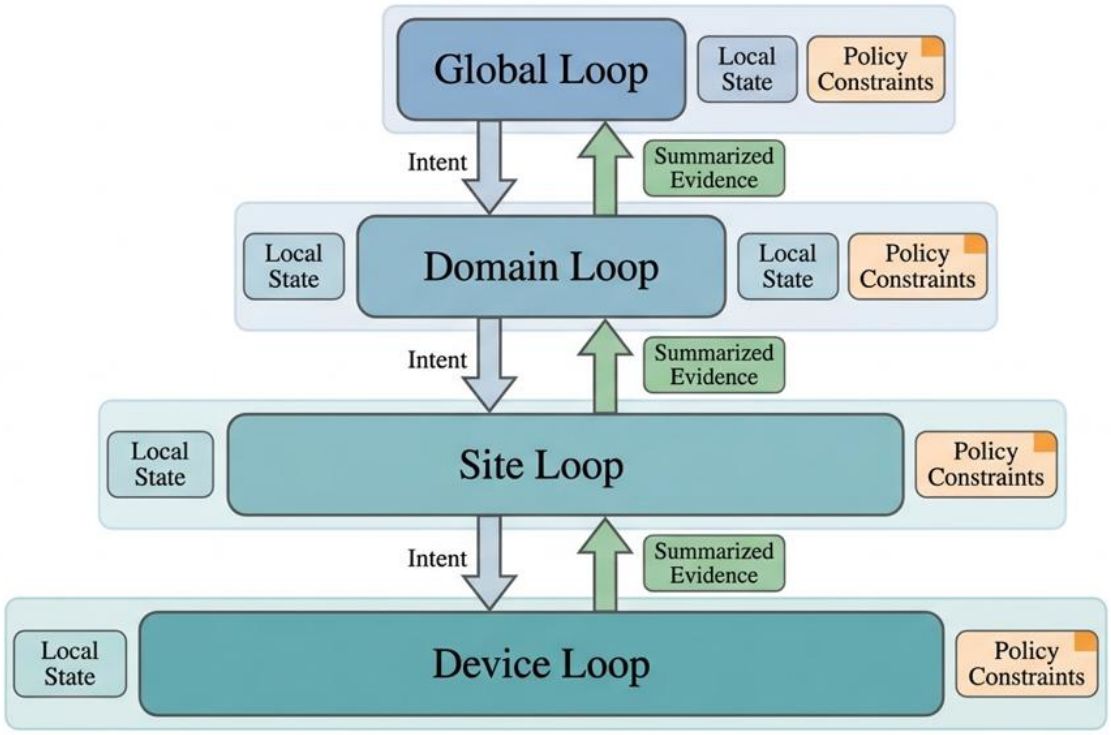


Figure 3- Hierarchical Multi-Loop Control Architecture

The evidence combination reduces the repetition of the central processing. The telemetry at high rates can be stored in partitions of the processing paths, and higher-level controllers fed on the findings, goal indicators, topology summaries, and exceptions. Total evidence is pulled where domain boundaries are spanned in instances where one has insufficient confidence in problems to address them locally.

3.4 Control Budgets and Action Classes

Enterprise-level automation needs to have clear boundaries or boundaries to the concurrency of changes. The architecture therefore inserts a control budget of concurrent remediation through site, device grouping, protocol,

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

resource holder and operational risk. This exerts correlated effect in the situation when corroborative analytical rule leads to many similar results and it retains execution capacity of higher priority cases.

The types of actions are divided into the reversible, scope of action, confidence requirements and strength of its verification. The work of evidence collection and diagnostic tests are in general relatively low intervention risk. Service restarts, configuration and software changes require progressively stricter authorization and validation. The successful and tested experience of operation in history may be able to give more freedom to narrow action classes and the unknown or high impact actions will be under more stringent restraints.

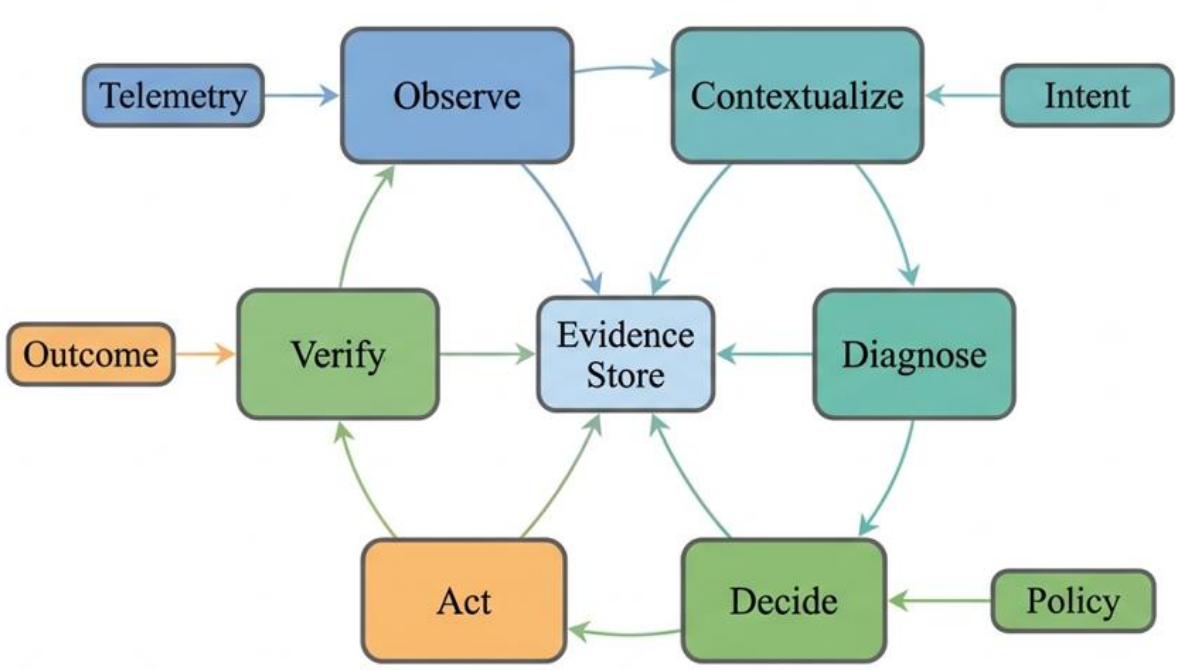


Figure 4- Evidence-to-Action Closed Loop

Circuit breakers are also another safety mechanism. In case of failure of repetition of remediation in enhancing the target outcome, the respective loop halts further endeavors, maintains diagnostic evidence and elevates the condition. This prevents duplication of activities which consume the control budget or of operation of the affected infrastructure which occurs most often.

3.5 High Availability and Loop Continuity

The principle of closed-loop availability does not require a variant of any operationally authority or state but, on the contrary, denotes that its functionality should be continuous. Interpreted intentions, workflow control points, ownership information and evidence associated with it including the execution state should be available even with the failure of components. Creating a clustered deployment, the quorum which can set up validate control must hold onto it, and the isolated minority must stop issuing independent measures.

After working on, reconcilers are also encouraged to test the state of recorded workflow with those of real external state even before the onset of suspicion operations. It eliminates the chances of duplicate executions whereby the action that the previous controller might have performed just before failure might have already been done. The degraded operation should also be observed to have freshness of assurance. Read operations or restricted configuration operations can be maintained and autonomous remediation temporarily turned off in the case of collection or analysis delays. To keep awake evidence as up-to-date otherwise, allow the controller to take a response based on an out of date view of the network state.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

In general, the architecture defines a multi-scale evidence-to-action control fabric where scale and hierarchy, evidence, intent, execution, verification and failure handling are considered as architecturally related. This provides a legal foundation under which closed-loop running of an enterprise might be carried out dependably without required to embroil the building of proprietary controller internals.

IV. FRAMEWORK ASSESSMENT: STRESS, RESILIENCE, AND OPERATOR-CENTRIC VALIDATION

4.1 Multidimensional Scale Assessment

It has to be tested on independent scales of controller scales of an enterprise and not just by the number of devices controlled to handle workload. The test should exercise the number of devices, clients, sites, the telemetry rate, topology churn, co-located findings, and the fan-out of workflow. It makes it possible to observe the potential bottlenecks that may be overlooked in case the entire dimensions of workloads are increased in equal measures.

Measures of assessment must include the entire closed loop. Ingestion loss and event freshness describe the ability of the collection layer to hold the evidence timely to operations. Creation delay and quality of root-cause ranking issue are correlation measures and issue diagnostics. The fraction of the operational goals that are describable and analyzeable by the architecture is referred to as intent coverage. An approximation of scheduling pressure is provided by age of queue in the execution layer and the throughput of workflows and convergence time measures how rapid a loop can increase, having identified a deviation, to the recovery. False remediation rate is a record of the times that automation causes a response that is incorrect or unnecessary. A control-budget saturation measures concurrent remediation, which is in the form of safety limits compliance. These measures combined will cover the evidence acquisition to decision execution to the accuracy of the decision.

4.2 Resilience and Failure Assessment

Failure assessment: Testing to check the possibility of the architecture being safely controlled in those instances when components or streams of information fail on an individual basis. Applicable scenarios: controller-node removal, service partition, slow topology update, duplicate telemetry or events and workflow termination on the receiving side of the external action. These are the situations, that is, under which traditional automation can periodically replicate the activity, be in stale state, or cannot even know who owns the workflow.

There are four main properties that should be tested in the assessment. Primarily, single ownership would avoid instances of two or more instances of controllers issuing conflicting remediation to the same workflow. Second, the known state preservation should ensure that committed intent, workflow checkpoints, ownership and any other relevant evidence can be visible following the case of a failure. Third, limited failover has to be a measurement of the duration and of the amount of time and effort required to bring the authoritative control back into service. Fourth, the controller should demonstrate pre-retry observation, particularly where the workflow termination may be performed by some external action that might have been delivered. The system should check actual external state before re-executing an uncertain operation. These tests show the availability and safety of change and the existence of mechanisms of resilience that keep change and its availability intact.

4.3 Closed-Loop Convergence Assessment

The measure of convergence can be based on the progression of the initial observation to diagnosis, approved action, execution and verified outcome. The comparison of the various loop scopes and classes of an action phenomenon should be considered in isolation because the local corrective activities can slow down faster than the cross-domain work processes. Sometimes even the evaluation has had to document those instances in which the verification is inconclusive in order that the framework could distinguish between the successful recovery and the successful implementation of the command.

4.4 Human-Centric Explainability Assessment

To consider the quality of automation must also be considered as far as the operator is concerned. This should be facilitated by finding records given to the operators and asking them to state the outcome affected, their suspected cause, evidence and steps taken or to be taken and uncertainty that remains. Identification accuracy, interpretation time and operator-to-operator consistency can be measured by evaluation. A find that produces automatic reaction, and does not provide proof of what it completed and has no clear explanation provides very little visibility to the manner in which affairs are operated. The human-factors test therefore complements the technical measures by providing a means

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

of determining the nature of the logic of the controller, that can be understood on normal functioning, degraded conditions and the more complex multi-domain events.

V. DISCUSSION: FROM PRODUCT OBSERVABILITY TO GENERALIZABLE CONTROL PRINCIPLES

Product architectures that are publicly documented can supply convenient evidence helpful in creating overall architectural ideas without turning an investigation article into product critique or marketing. In this case, the enterprise controller is a tangible point of focus where the issues of large-scale network management may be driven to the fore. This should not be a reproduction of the implementation specifics of the proprietary but to identify architectural challenges that can be similar to most controller platforms or deployment contexts.

There are several problems that begin to increase according to the size of the controllers. The heterogeneous context requires the association of the telemetry, identity, topology, inventory, applications and policy information that may be offered by different sources. Telemetry with high-cardinality creates significant ingestion, processing, storage and freshness requirements. Cross-domain findings create dependencies, which may not be solvable using any one local controller. Equally, long running workflows add the needs of persistent state, ownership, idempotency, partial completion support and afterwards verification. The other control boundary is external integration in which API and event interfaces must provide convenient extensibility without allowing external requests to circumvent authorization policy or execution policy.

It is also proven by continuity Quorum based continuity that high availability in closed loop systems does not simply imply having redundant instances of controllers. The accepted intent, state of workflow, ownerships and evidence must be coherent enough to prevent conflicting actions during letdowns or network partitions.

These visible challenges have been broken down their explicit and testable design rules in the suggested architecture. Multidimensional scale is viewed in isolation, scoping hierarchical loops limit the extent of decision, evidence aggregation controls the amount of information, action classes differentiate automation control and budgets constrain concurrent remediation. Checking guarantees that the success of execution is not mistaken to success recovery during recovery. Together, these principles provide enterprise controllers with vendor-neutral framework to compare on the foundations of scalability, resiliency, explainability and operational safety and not a characteristic set of proprietary features.

VI. CONCLUSION

Enterprise scale closed loop automation requires an architecture that results in persistent vision with minimal and verifiable action. A centralized model of control will not scale safely as the size of the device, client and site population, number of applications, number of policies, telemetry flows and the number of applications underway increases. The proposed architecture resolves this issue by using hierarchical scope of control, aggregation of evidence, explicit classes of actions, control budget, long lasting workflows and outcome-focused verification.

Hierarchical loops are applied to allocate operation decisions in terms of scope to allow local conditions to be addressed near to their source and enterprise-wide intent and policy limits to be upheld. Aggregation of evidence reduces the cost of mobility and processing of telemetry in large amounts to higher control spheres as well as retains access and details, where cross-domain diagnosis is needed. The difference is made between low risk diagnostic operations and the changes which require high confidence level of authorization and verification by explicit classes of action. Control budgets and concurrent remediation are both limited by control budgets, and the potential impact of erroneous decisions or correlated decisions can be limited as well.

The impact of sustained workflows is continuity with partial failure in execution and controllers and the use of post action verification to verify that the execution of configuration completion is not confused with service recovery. High-availability systems also retain ownership and accepted state and prevent uncertain actions and duplicate actions in the case of a failover.

The architecture in its entirety provides a vendor-neutral model of exposing automation, but allows scale to magnify both the uncertainty analysis and the uncertainty of the uncertain remediation and uncontrollable concurrency. It is grounded on the concept of restricted autonomy: the controllers are supposed to be autonomous subject to clearly

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 4, July- August 2022|

DOI:10.15680/IRJIE.2022.0604002

defined limits of authority, evidence, resource, and verification. This offers a viable base on algorithmic, elucidable, robust, and healthier closed-loop business network operations.

REFERENCES

[1] ETSI, Zero Touch Network and Service Management (ZSM); Reference Architecture, ETSI GS ZSM 002 V1.1.1, Aug. 2019. [Online]. Available: ETSI GS ZSM 002 Reference Architecture
[2] Cisco Systems, Introduction to Cisco Catalyst Center API. [Online]. Available: Cisco Catalyst Center API Documentation
[3] M. Behringer et al., "Autonomic networking: Definitions and design goals," RFC 7575, Internet Engineering Task Force, Jun. 2015. doi: 10.17487/RFC7575.
[4] H. Song et al., "Network telemetry framework," RFC 9232, Internet Engineering Task Force, May 2022. doi: 10.17487/RFC9232.
[5] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, "Intent-based networking—Concepts and definitions," RFC 9315, Internet Research Task Force, Sep. 2021.
[6] X. Zheng and A. Leivadeas, "Network assurance in intent-based networking data centers with machine learning techniques," in Proc. IEEE Int. Conf. Netw. Service Manag. (CNSM), 2021.
[7] A. Leivadeas and M. Falkner, "VNF placement problem: A multi-tenant intent-based networking approach," in Proc. Int. Conf. Internet Services and Networks (ICIN), 2021.
[8] M. Kiran, E. Pouyoul, A. Mercian, B. Tierney, C. Guok, and I. Monga, "Enabling intent to configure scientific networks for high performance demands," Future Generation Computer Systems, vol. 79, pp. 205–214, 2018.
[9] A. Rafiq et al., "Intent-based end-to-end network service orchestration system for multi-platforms," Sustainability, 2020.
[10] A. Rafiq et al., "Intent-based slicing between containers in SDN overlay network," Journal of Communications, 2020.
[11] A. S. Jacobs, R. J. Pfitscher, R. A. Ferreira, and L. Z. Granville, "Refining network intents for self-driving networks," in Proc., 2018.