Automation for Operations

T-Mobile Journey

Conceptual patterns and lessons from large-scale operator environments

Feb 2026

T-Mobile

Agenda Overview

- Modern Operations: Challenges and Realities
- Operations Tasks Suitable for Automation
- Manual Operations to Automated Foundations
- Automation Use Cases: Reactive vs Proactive
- From Use Cases to Architecture Design
- Design, Lessons Learned, and What's Next
- Questions & Discussion

Presenters

Issa Abu Eid, CCIE #23629 (R/S, SP, DC)

Role:

- Senior Member of Technical Staff

Focus:

- Network Automation, Reliability, and Large-Scale Operations

Udaya Tadikonda, CCIE #39065 (R/S)

Role:

- Principal Engineer

Focus:

- Network Automation, Reliability, and Large-Scale Operations

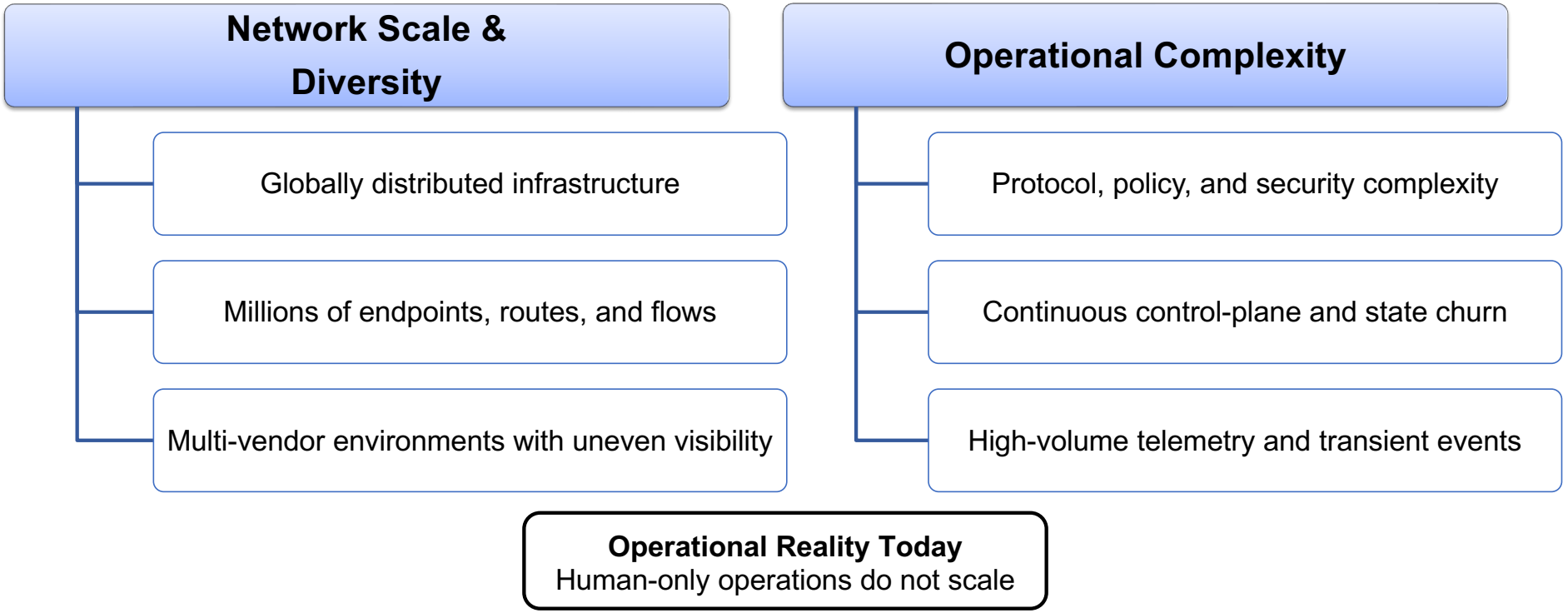
NANOG™

Modern Operations

Challenges and Realities

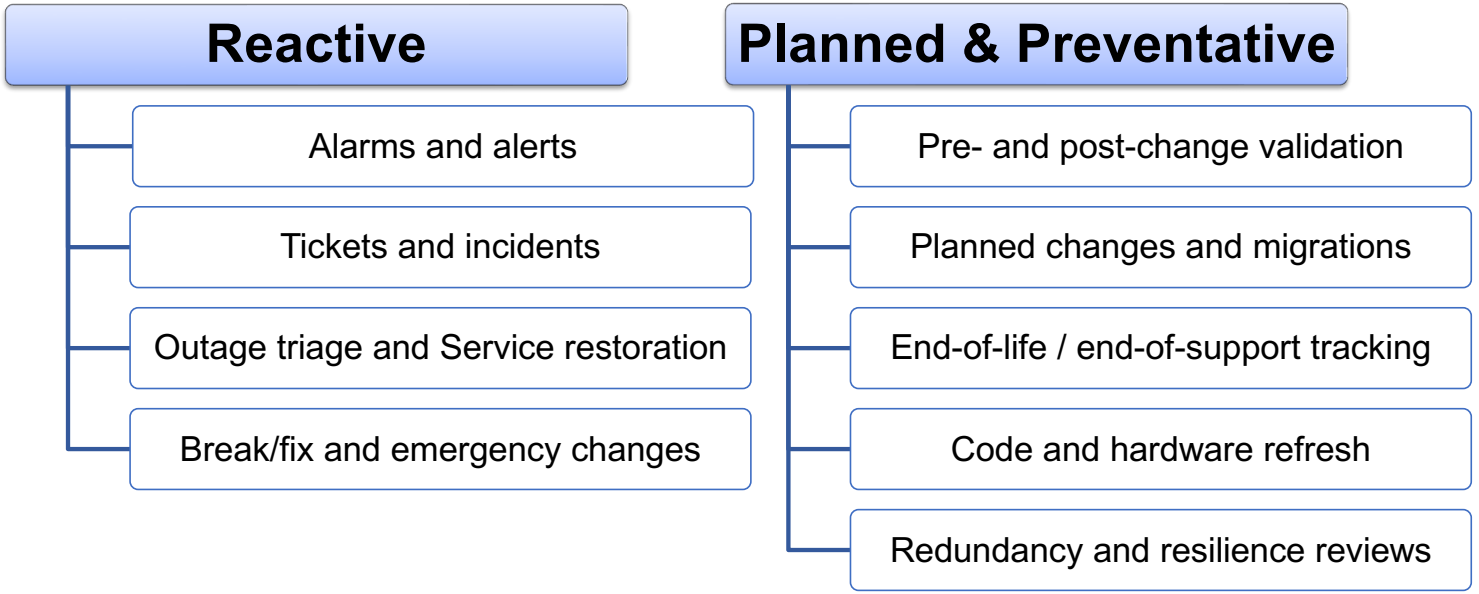
NANOG™

The State of Modern Operations



Core Operations Functions

What operations teams do to keep the network available and reliable



Operational Success Is Measured By
MTTR — Mean Time To Repair
MTBF — Mean Time Between Failures

What Is Automation in Operations?

Reducing manual effort through repeatable, predictable workflows

Automation in Operations

- Codified operational workflows
- Event-, schedule-, or request-driven
- Consistent and deterministic execution
- Reduces manual, error-prone effort

Automation vs. AI

Automation

- Deterministic and rule-driven
- Performs actions
- Predictable and auditable

AI

- Probabilistic and model-driven
- Assist Human decision making

Automation reduces operational load today
AI augments operators over time

Operations Tasks Suitable for Automation

Reactive & Proactive

Illustrative categories common across large-scale operator environments

NANOG™

Incident & Alarm Automation (Reactive)

Operational Reality

- High alert volume, low signal
- Multiple alarms for a single root cause
- Human operators forced into pattern matching

Automation Capabilities (Conceptual)

- Alert correlation and deduplication
- Incident auto-creation with enriched contextual information
- First-level triage signals (e.g., device, interface, recent change)

Illustrative example

- A single physical failure may generate many related alarms
- Correlation can consolidate signals into a single, appropriately scoped incident

Goal: Reduce alert noise without hiding real failures

Ticketing Automation: From Alerts to Action

Manual Model

- Tickets created late or inconsistently
- SLA clocks start too late
- Status updates depend on humans

Automation Capabilities (Conceptual)

- Automated ticket creation capabilities
- SLA tracking based on initial detection signals
- Automated updates driven by state changes
- Policy-based or conditional ticket closure

Illustrative example

- A transient network condition is detected
- A ticket can be created with relevant diagnostic context
- The ticket lifecycle can be updated or resolved when conditions normalize

Goal: Automation enforces consistency humans cannot

Outage Detection & Response

Outage Detection

- Multi-signal detection (not just ping)
- Service-aware vs device-only detection

Impact Identification

- Which services may be affected
- Which customers or regions may be impacted

Coordinated Response (Automation-Assisted)

- Pre-defined response playbooks with guardrails
- Automated notification workflows for stakeholders

Illustrative example

- A critical infrastructure component experiences a failure
- Potentially affected services are identified early
- Notifications can be sent proactively to relevant stakeholders

Goal: Identify impact and coordinate response before customers notice

Break/Fix Automation: Safe, Repeatable Actions

Automated Actions

- Candidate actions for automation (e.g., restarts, resets)
- Well-understood configuration corrections
- Known, repeatable recovery patterns

Explicitly Not Automated

- Ambiguous failures
- One-off fixes
- High-risk changes

Safety Mechanisms

- Redundancy-aware decision checks
- Capacity validation before action
- Auditable execution with rollback capability

Illustrative example

- A recoverable fault condition is detected
- Guardrail checks confirm redundancy and capacity
- A predefined corrective action may be considered

Goal: Enable predefined, guarded actions only when risk is understood and controlled

Change Management Automation (Proactive)

Operational Reality

- Manual change validation is inconsistent and slow
- Failures are often detected after customer impact
- Rollbacks depend on human reaction time

Automation Capabilities (Conceptual)

- Pre-change validation signals (e.g., baseline health, redundancy)
- Post-change health verification signals
- Policy-driven rollback considerations based on detected deviations

Illustrative example

- A planned change introduces unexpected behavior
- Health signals deviate from expected baselines
- Predefined rollback conditions may be evaluated

Goal: Reduce change-related incidents before customers notice

Compliance Automation (Proactive)

Operational Reality

- Device fleets drift over time
- EOL / EOS risk is often discovered late
- Compliance checks are periodic, not continuous

Automation Capabilities (Conceptual)

- EOL / EOS awareness and reporting capabilities
- Version and configuration compliance monitoring
- Drift detection across large device fleets
- Security visibility informed by compliance state

Illustrative example

- Configuration drift is detected on a subset of devices
- Non-compliant elements are identified ahead of planned changes

Goal: Maintain continuous compliance without manual audits

Proactive Monitoring & Early Warning Automation

Operational Reality

- Capacity monitoring with trend analysis techniques
- Resiliency and redundancy assessment signals
- Early warning thresholds for degradation detection

Automation Capabilities (Conceptual)

- Capacity monitoring with trend analysis
- Resiliency and redundancy checks
- Early warning thresholds

Illustrative example

- Resource utilization trends indicate approaching capacity limits
- Risk indicators are flagged early, before service degradation occurs

Goal: Detect degradation before it becomes an outage

Predictive Actions & Preventive Automation

Operational Reality

- Known failure patterns repeat over time
- Preventive action is rarely prioritized

Automation Capabilities (Conceptual)

- Pattern-based risk identification and prediction signals
- Safe, low-risk preventive workflow *candidates*

Illustrative example

- A recurring error pattern is identified over time
- Preventive actions are recommended or queued for review

Goal: Reduce failure probability, not just recovery time

Manual Operations to Automated Foundations

NANOG™

Foundation Evolution — Overview

Operational Reality

- Automation cannot succeed without reliable state
- Early efforts focus on visibility, not action
- Maturity requires incremental evolution

Automation Capabilities (Conceptual)

- Visibility → Intelligence → Enablement
- Each phase builds on the previous one
- No shortcuts without operational risk

Illustrative example

- Early automation attempts were limited by incomplete state
- Maturity improved as foundational capabilities strengthened

Goal: Show that automation maturity is an evolution, not a single step

Phase 1: Visibility — Making the Network Observable

Operational Reality

- Network state lived in devices, not systems
- Manual CLI access limited scale
- No shared view of current state

Automation Capabilities (Conceptual)

- Periodic collection of network state
- Device-level state capture for observability
- Configuration and forwarding state made accessible
- Raw data normalized for reuse across workflows

Illustrative example

- Periodic state snapshots enable baseline comparison
- Improved visibility exposes previously unknown inconsistencies

Goal: Make the network observable before attempting automation

Phase 2: Intelligence — From Snapshots to Change

Operational Reality

- Snapshots alone did not explain incidents
- Drift and unintended changes went unnoticed
- Humans identified issues after impact

Automation Capabilities (Conceptual)

- Increased state visibility over time
- State comparison and change detection
- Drift and unintended change identification
- Proactive issue identification signals
- Alerting and notification mechanisms
- Shared, reusable intelligence logic

Illustrative example

- Configuration drift is identified prior to broader impact
- Improved change visibility simplifies investigation workflows

Goal: Understand change before attempting automated decisions

Phase 3: Enablement — Automation at Scale

Operational Reality

- Intelligence alone did not drive action
- Manual workflows slowed response
- Automation needed to integrate with operations

Automation Capabilities (Conceptual)

- More frequent data collection where appropriate
- Metrics and KPI generation capabilities
- Health checks and validations
- Pre- and post-maintenance checks
- Deployment and workflow integration
- Scalable, shared data access

Illustrative example

- Maintenance windows can be validated using automated checks
- Health checks help reduce the risk of post-change incidents

Goal: Make automation a sustainable part of operational workflows

NOC Automations – Overview

There are two big roles automation plays in the NOC:	
Enable predefined actions when confidence thresholds are met	Give humans better context

Automation augments operators — it does not replace them

NOC Auto-Mitigation (Reactive)

Operational Reality

- Known failures recur frequently
- Human response time is slower than failure propagation
- Paging humans for repeatable issues adds no value

Automation Capabilities (Conceptual)

- Auto-mitigation patterns for known, repeatable failure scenarios
- Candidate recovery actions for interface and link failures
- Guarded responses to control-plane instability signals
- Resource exhaustion detection with predefined response options
- Coordinated response patterns for volumetric events

Illustrative example

- A known, repeatable failure condition is detected
- Guardrail checks confirm redundancy and capacity
- A predefined mitigation action may be initiated under controlled conditions

Goal: Restore service quickly while minimizing unnecessary human intervention

NOC Context & Data Collection Automation

Operational Reality

- Humans are overwhelmed by raw alarms
- Context is missing at incident time
- Investigation starts with data collection, not analysis

Automation Capabilities (Conceptual)

- Trigger-based collection of high-fidelity diagnostic context
- Event-driven data capture for common network failure signals
- Automated context gathering at incident time

Illustrative example

- A network control-plane event is detected
- Relevant diagnostic state is captured automatically
- The operator begins investigation with enriched context

Goal: Give humans context, not noise

NOC Scale & Noise Reduction Automation

Operational Reality

- High-volume incidents overwhelm NOC staff
- Multiple alarms represent a single root cause
- Mean time spent triaging dominates MTTR

Automation Capabilities (Conceptual)

- Alarm correlation and de-duplication
- Context-aware grouping
- Automated triage at scale

Illustrative example

- A single underlying failure generates many related alarms
- Correlation groups related signals into one incident
- Operators focus on one issue instead of dozens of alerts

Goal: Humans see one incident, not fifty alarms

Automations Beyond the NOC – Overview

When automation patterns mature in the NOC, they can be applied beyond it — into IT, partner, and engineering workflows.”

Automation for IT & Security Teams

Operational Reality

- Asset inventory is incomplete or stale
- Security context lacks network awareness
- Manual validation does not scale

Automation Capabilities (Conceptual)

- Inventory and configuration validation signals
- Threat-hunting data enrichment inputs
- Application flow visibility and mapping

Illustrative example

- Security alerts can be enriched with relevant network context
- Enriched context helps simplify investigation and reduce noise

Goal: Give IT and security timely, relevant network context

Automation with External Partners

Operational Reality

- SLA disputes lack shared data
- External monitoring is siloed
- Accountability is unclear

Automation Capabilities (Conceptual)

- Integration of external monitoring signals
- SLA and performance reporting *inputs*
- Controlled cross-domain data sharing

Illustrative example

- Shared metrics provide a common reference point during SLA discussions
- Improved visibility helps reduce ambiguity during incident resolution

Goal: Create shared visibility and clarity across organizational boundaries

Automation for Engineering & Vendor Accountability

Operational Reality

- Resource sprawl is invisible
- Redundancy assumptions are not validated
- Vendor issues surface late

Automation Capabilities (Conceptual)

- Orphaned or unused resource identification
- Redundancy validation signals
- Vendor circuit performance visibility and reporting

Illustrative example

- A redundancy gap is identified during pre-change analysis
- Engineering teams address the issue before planned work

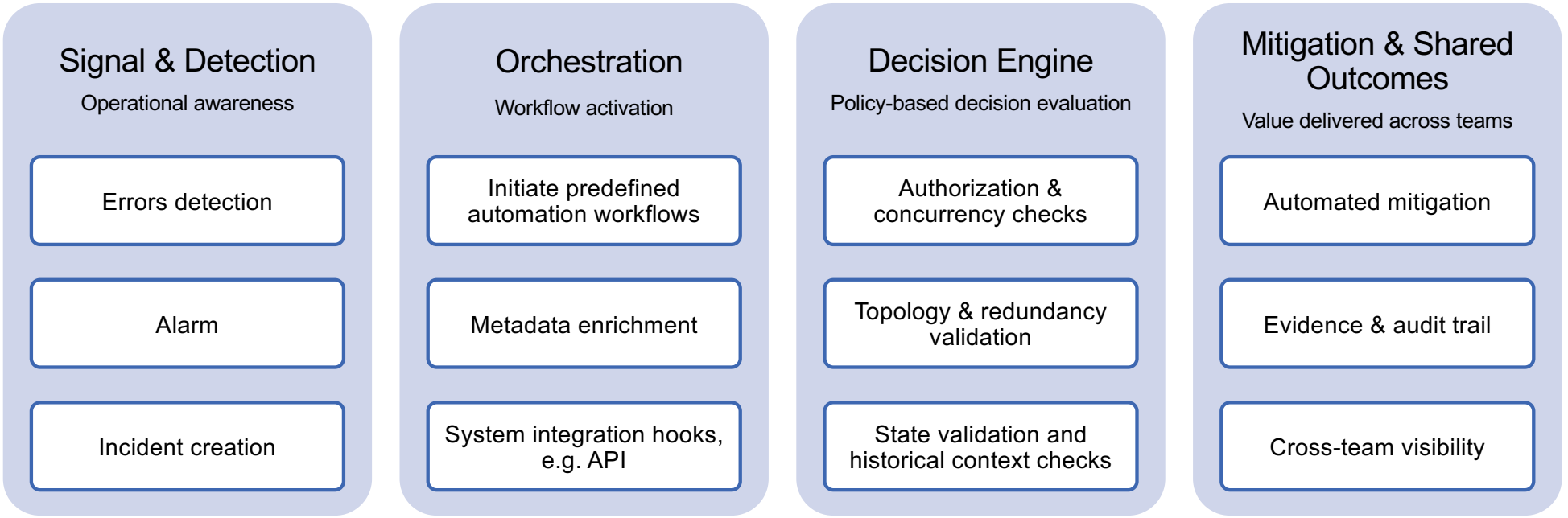
Goal: Improve transparency and accountability using shared data

Automation Use Cases

Reactive vs Proactive Cases

NANOG™

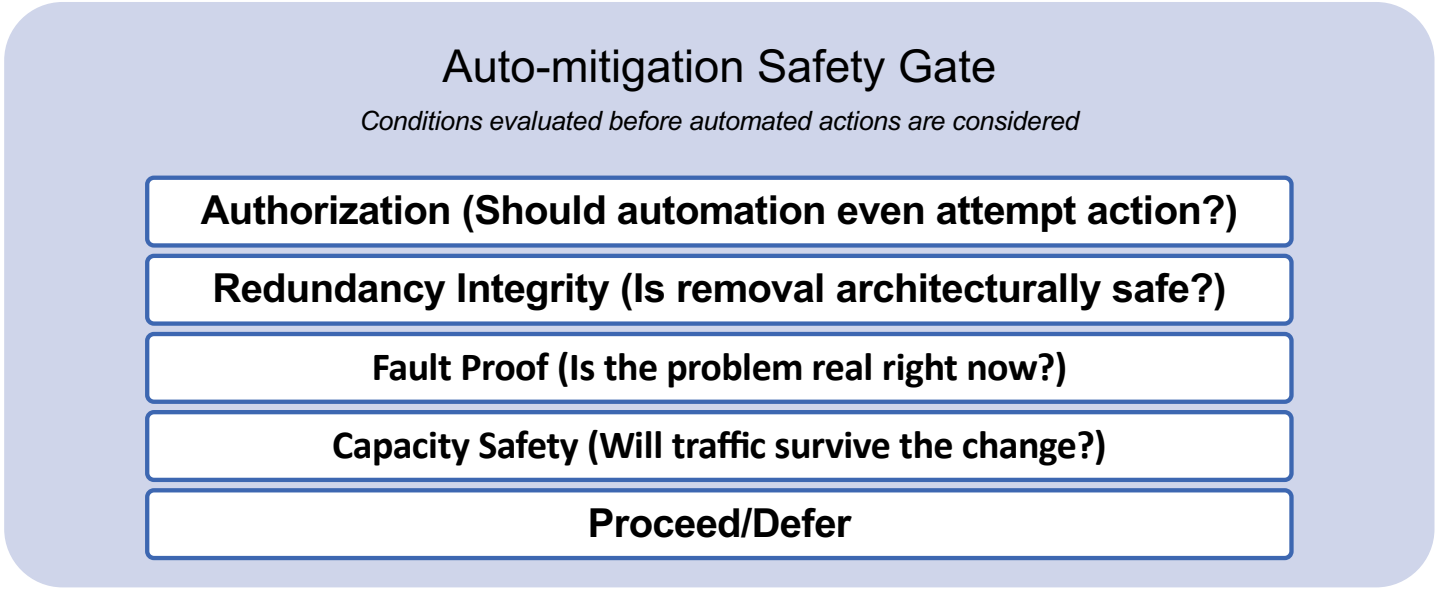
Auto-mitigation Pattern – Interface Error Scenarios



Reducing recovery time while managing risk through guardrails and auditability.
Auto-mitigation patterns emphasize guardrails such as redundancy awareness, capacity checks, and auditability.

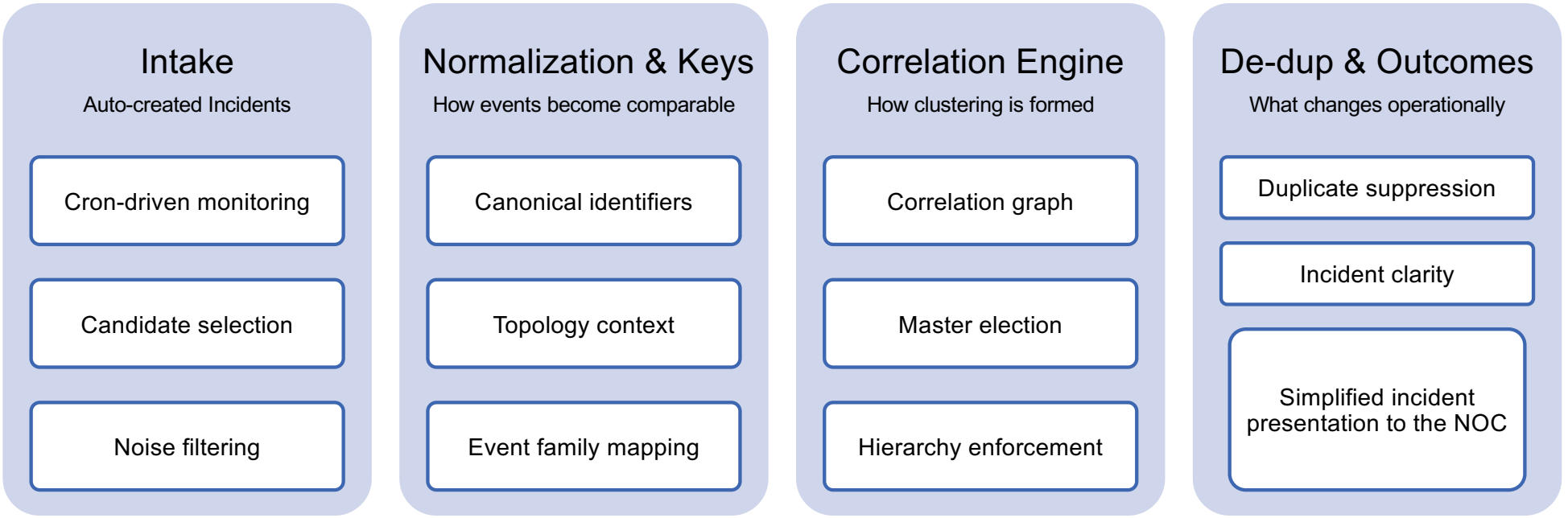
Illustrative reference architecture — not representative of any specific production implementation.

Auto-mitigation – Conditions



The safest automation is the one that knows when not to act.

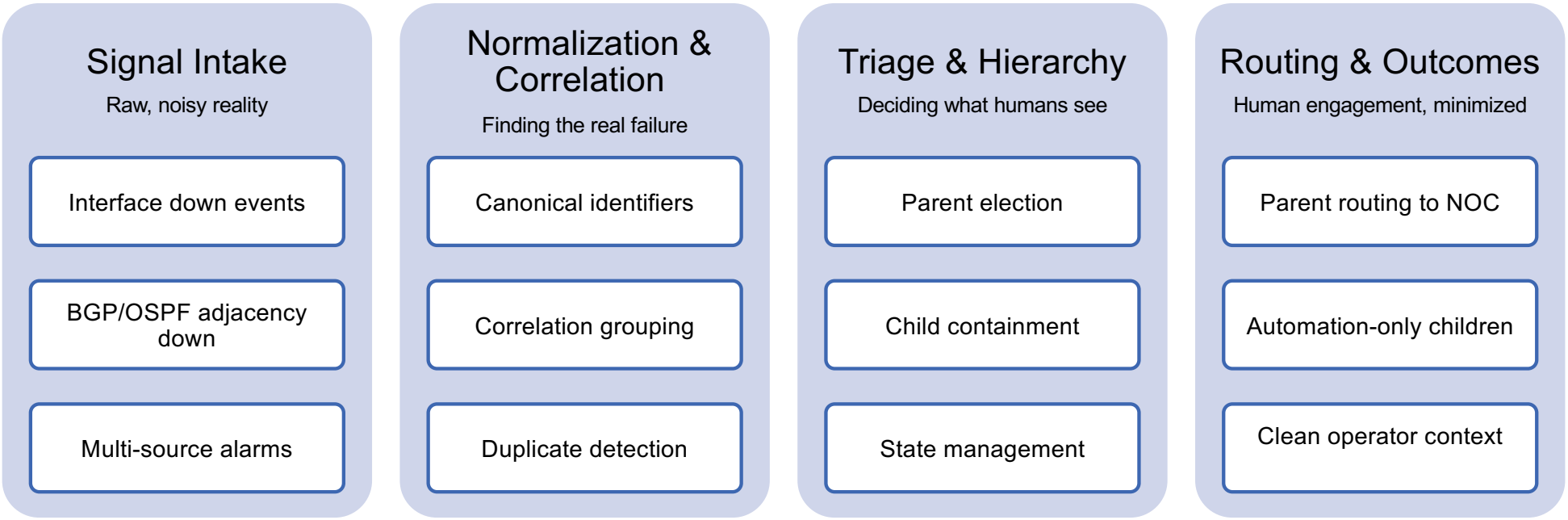
Correlation & De-duplication



Correlation can turn alert floods into a single, safer decision point.
De-duplication helps ensure the network speaks once — and clearly.

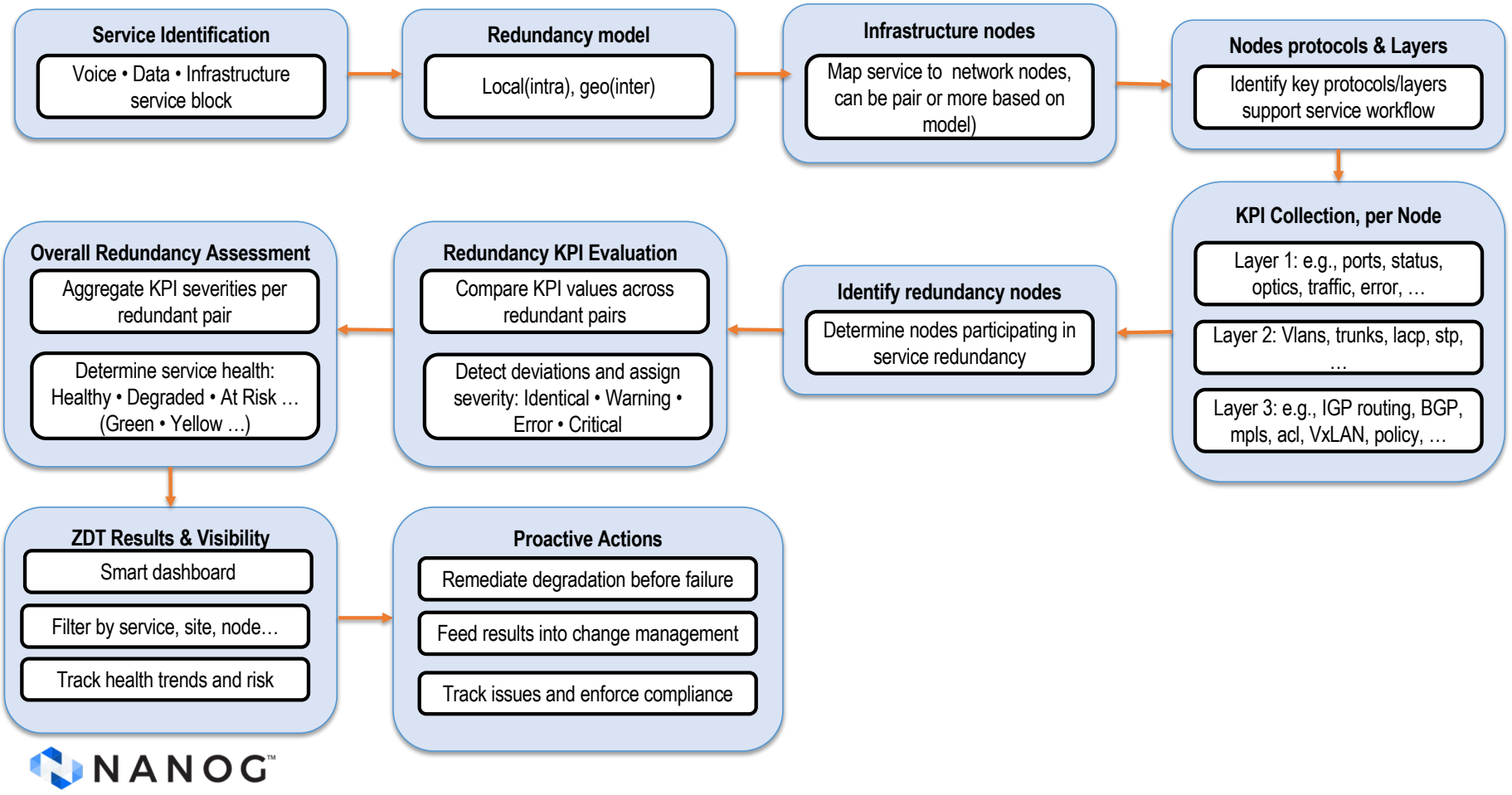
Illustrative correlation model — conceptual only.

Auto-triage – Interface Down/BGP Down



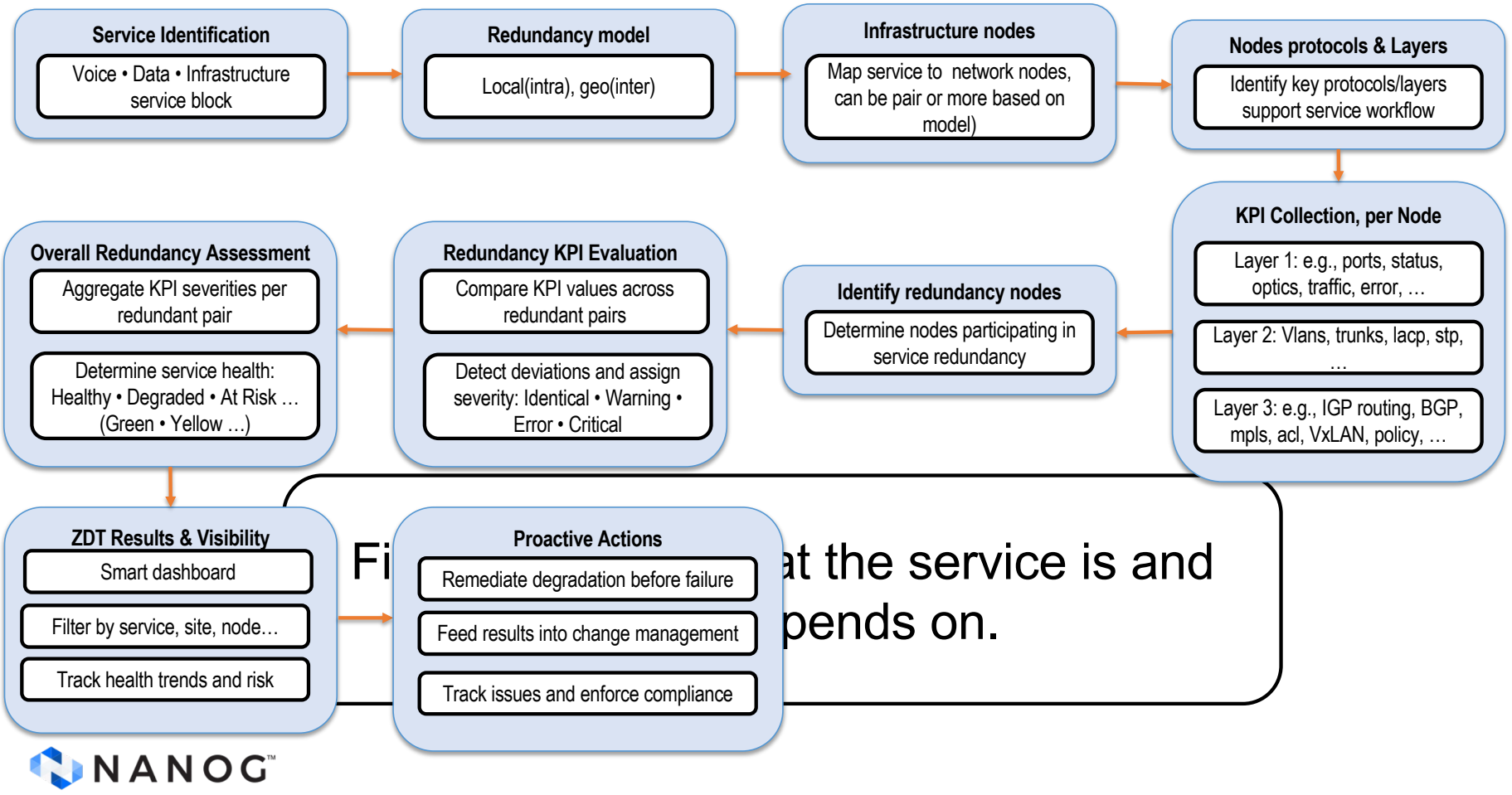
Auto-triage reduces cognitive load before humans ever touch the incident.

Service Redundancy Validation (Overview)

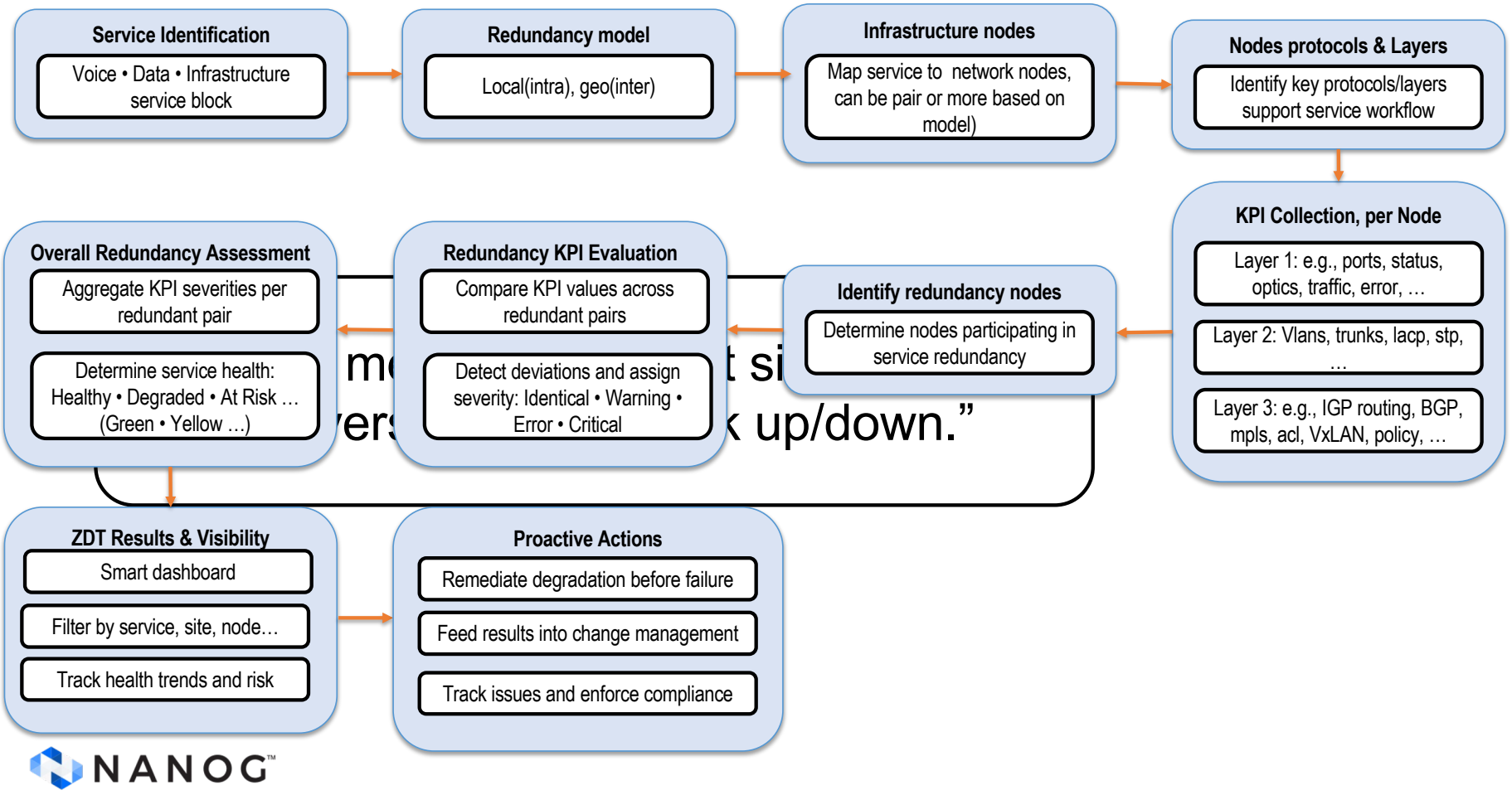


NANOG™

Service Modeling & Dependency Mapping

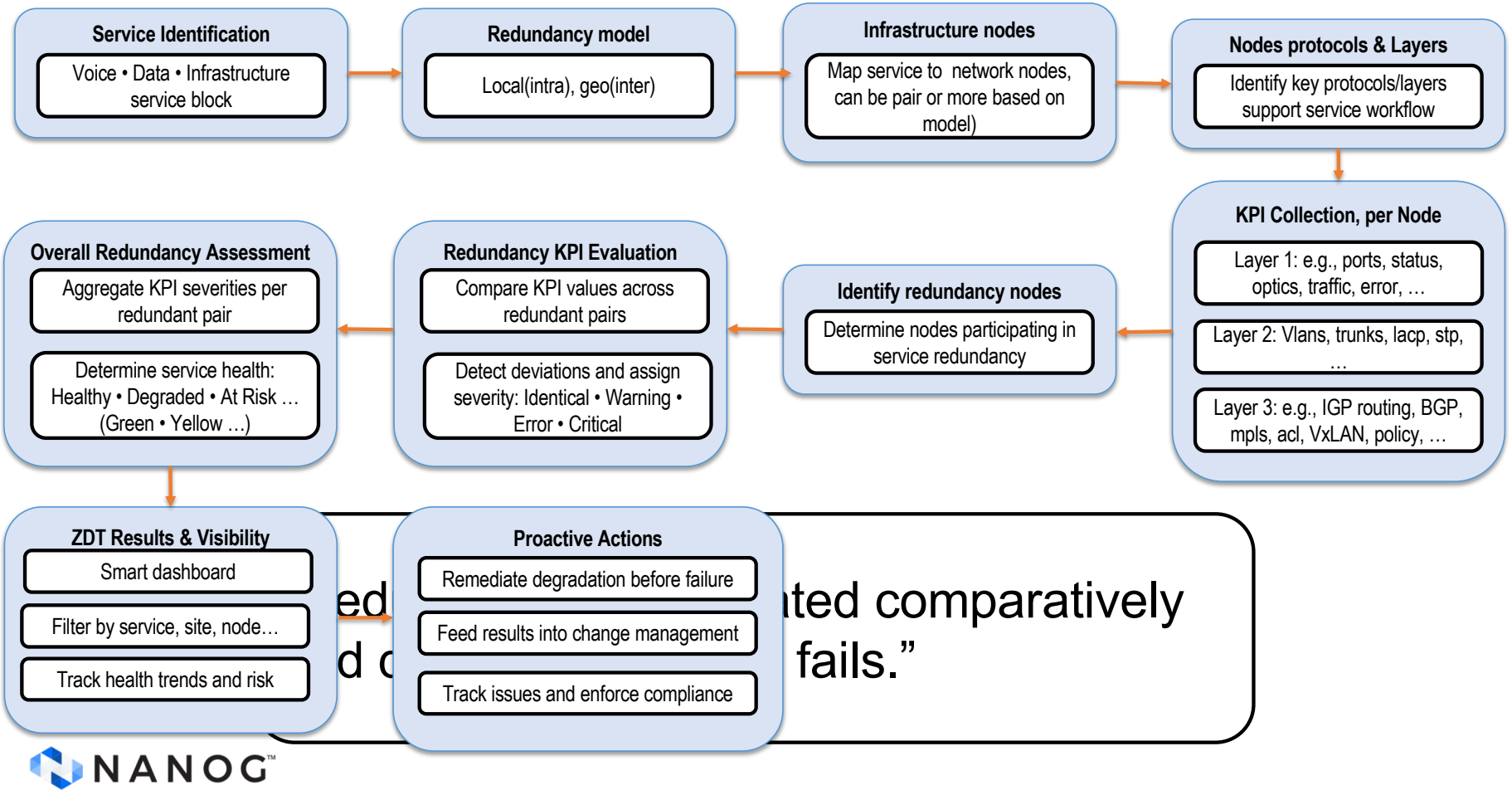


Protocols, Layers & KPI Collection



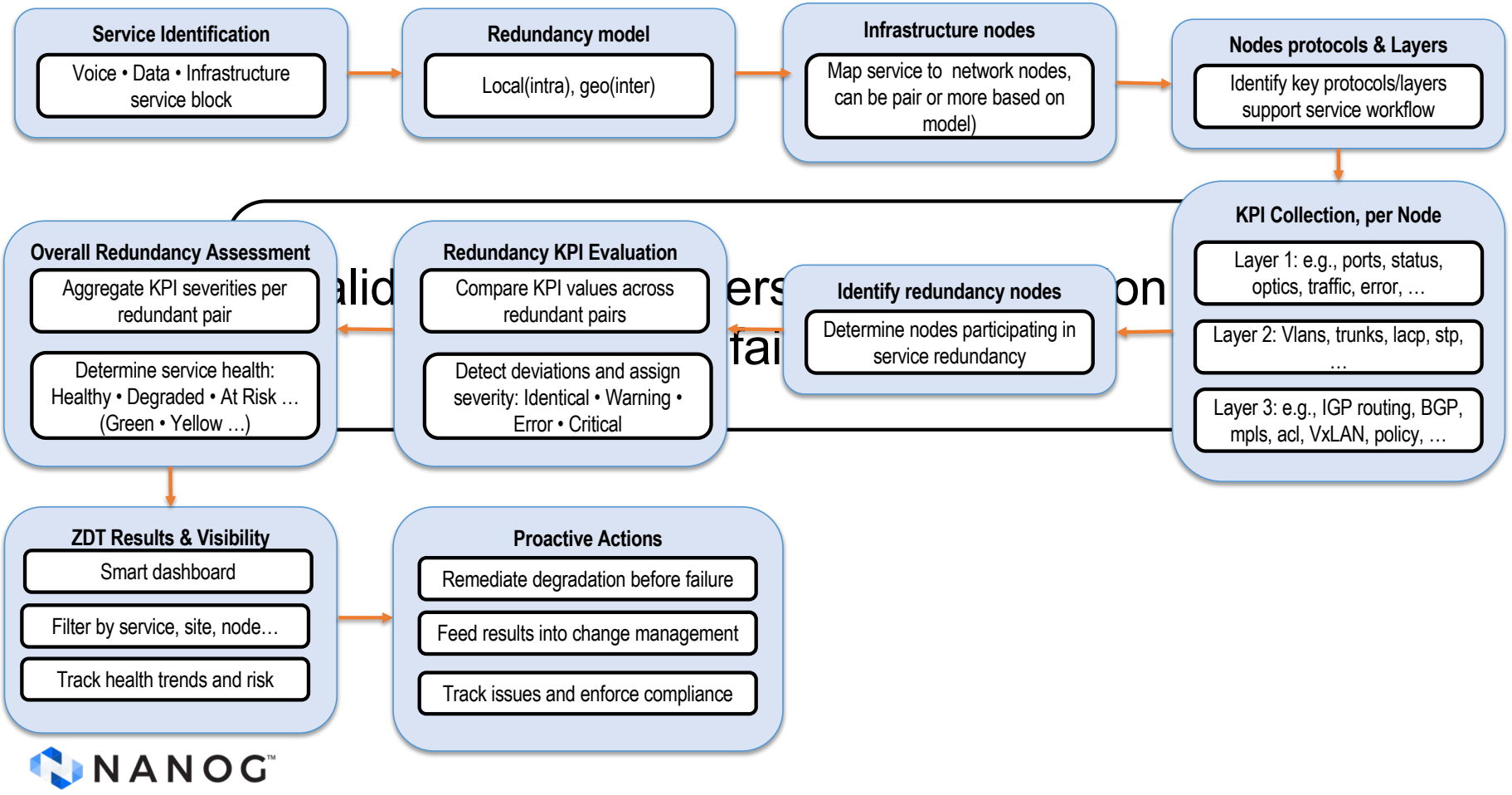
NANOG™

Redundancy Evaluation & Health Scoring



NANOG™

Visibility & Proactive Actions



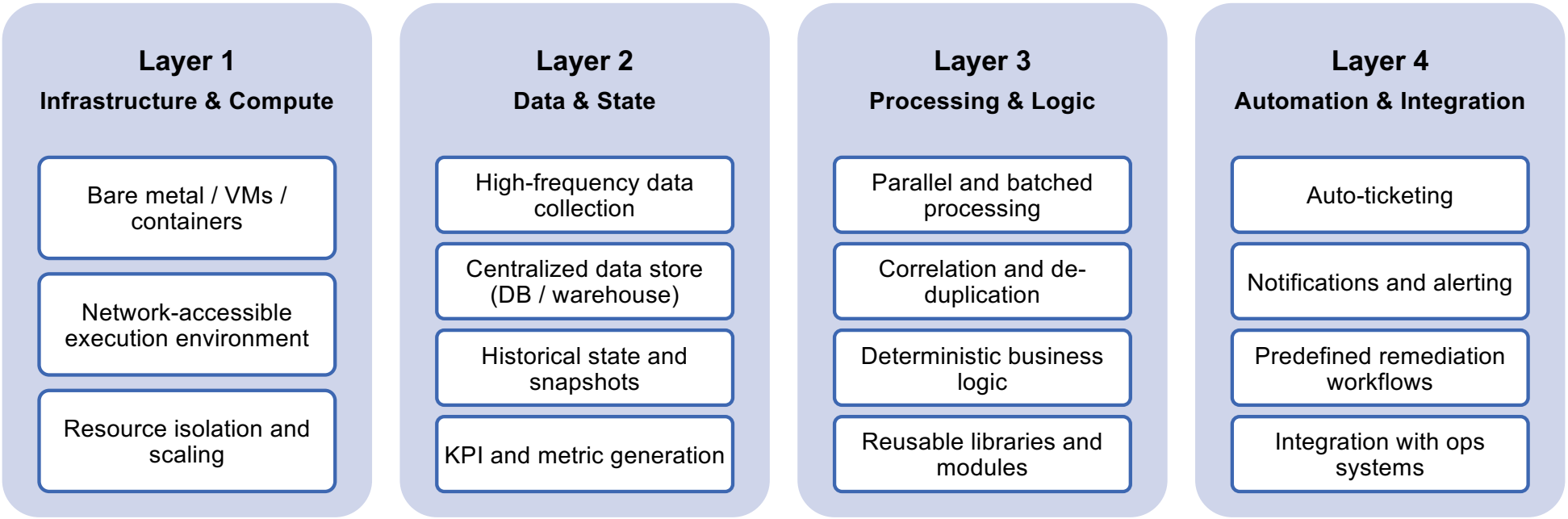
From Use Cases to Architecture

Design, lessons learned, and next steps

NANOG™

Automation Architecture

Building reliable automation from infrastructure to workflows



Automation is production software — it needs real infrastructure

Good automation depends on good state

This is software engineering, not scripting.

Automation executes known decisions safely.

NANOG™

Challenges & Constraints

Most automation failures are engineering problems, not tooling problems.

Technology & Platform Challenges

(*Constraints imposed by scale and infrastructure)*

- Resource contention at scale
- Stateful services and long-running daemons
- Authentication and secret management
- Data locality and replication
- Cross-region reliability and latency

Engineering & Process Challenges

(*How automation is built and maintained)*

- Ad-hoc scripts without ownership
- Non-parallel and inefficient execution paths
- Local state and duplicated logic
- Limited testing and validation
- Knowledge silos and tribal ownership

Automation reliability depends as much on engineering discipline as on infrastructure

The Future of Automation

From deterministic execution to AI-assisted operations

What Stays the Same

(*Automation remains the foundation)*

- Deterministic, repeatable workflows
- Human-defined guardrails and approvals
- Focus on reliability, safety, and auditability
- Automation executes known decisions

"Automation continues to do the work — safely and predictably."

What Evolves

(*AI augments operators, not replaces them)*

- Pattern detection across large data sets
- Faster root-cause hypothesis generation
- Intelligent prioritization of incidents
- Context enrichment for human decisions

AI helps operators decide **what** to do — not blindly do it.

The Operating Model Going Forward

(*Human-in-the-loop by design)*

- Operators remain accountable
- AI recommendations are explainable
- Automation actions remain controlled
- Trust is built incrementally

The future is assisted operations, not autonomous operations.

Strong automation foundations enable safe, effective use of AI

NANOG™

Thank you

Automation succeeds when treated as production software, grounded in strong engineering discipline, with AI used carefully to assist — not replace — human operators.