

AI-Driven Attack Mitigation in Resilmesh

Ekam Puri Nieto, Jorge Bernal Bernabe, Antonio Skarmeta Gomez
University of Murcia (UMU)

September 2026

Executive Summary

Resilmesh[1] is an EU-funded research initiative developing a next-generation Security Operations and Analytics Platform Architecture (SOAPA) for heterogeneous, distributed digital infrastructures. Within that architecture, the **Mitigation Manager** is the component that turns security alerts into enforced countermeasures. It sits in the Security Operations Plane, receives alerts from the platform's SIEM, tracks multi-step attacks as they unfold, and selects the least disruptive set of Course of Action (CoA) playbooks to apply, which it then hands to the Playbooks Tool and Workflow Orchestrator for execution.

This whitepaper describes the general idea behind the Mitigation Manager and, in particular, how it employs Artificial Intelligence to decide on mitigations. The core of that decision process is an **AI planning engine** built on constraint-based optimisation: the mitigation problem is expressed as a set of facts, decision variables and constraints, and a solver searches for the response plan that satisfies every hard requirement while minimising cost and disruption. This approach was introduced in the second version of the component and has since been refined through two project pilots and an integration with the CACAO-native X-MESH orchestration environment.

The sections that follow begin by framing the attack-mitigation challenge and locating the Mitigation Manager within the Resilmesh architecture. They then trace the mitigation process from alert intake and attack tracking through response selection and enforcement, before examining the AI planning model in detail. The final sections chart the component's evolution across the project, discuss its use of open standards, and outline its current limitations and roadmap.

Table 1 summarises the Mitigation Manager's core capabilities and their place in the Resilmesh architecture.

[1] https://resilmesh.eu

Capability	Role	Where it sits
Alert intake and normalisation	Turns heterogeneous SIEM alerts into a consistent internal representation	Interface with the SIEM over the platform's NATS message broker
Attack tracking	Correlates alerts into multi-step attack patterns modelled on MITRE ATT&CK and Attack Flow	Internal state of the Mitigation Manager
AI planning	Selects the optimal set of mitigations through constraint-based optimisation	Planning engine embedded in the Mitigation Manager
Situational enrichment	Draws on the cyber situational awareness model to validate conditions and fill playbook parameters	Interface with the Infrastructure and Services Information Model (ISIM)
Enforcement	Triggers CoA playbooks that act on the infrastructure	Playbooks Tool (PT), Workflow Orchestrator (WO), and X-MESH SOARCA

Table 1. Core capabilities of the Mitigation Manager.

1. The Attack Mitigation Challenge

Modern security operations centres face a structural imbalance. Detection has become abundant: a SIEM ruleset applied to a mid-sized infrastructure produces thousands of alert types, and machine-learning detectors add anomaly alerts on top. Response, by contrast, remains largely manual. An analyst has to read the alert, decide whether it is part of a larger attack, choose an appropriate countermeasure, check that the countermeasure will not harm legitimate operations, and finally execute it. Each of these steps takes time that an attacker does not need to grant.

Automating this chain is not simply a matter of attaching a script to each alert. Three properties distinguish a useful automated mitigation capability from a brittle one:

- **Attack awareness.** Alerts rarely arrive in isolation. A port scan, a brute-force attempt, a successful login and a privilege escalation are four alerts but one attack. A mitigation component that treats them independently will either over-react to reconnaissance or under-react to the step that matters.
- **Situational awareness.** The right countermeasure depends on where the affected asset sits, what services depend on it, and how critical it is to the organisation's mission. Isolating a workstation is cheap; isolating the server that hosts a critical service is not.
- **Principled choice under constraints.** Given several applicable countermeasures, some cheaper and some more thorough, the component must be able to justify its choice, respect hard rules set by operators, and do so within a bounded reaction time.

The Resilmesh Mitigation Manager was designed around these three properties. The rest of this document explains how.

2. Where the Mitigation Manager Sits in Resilmesh

Resilmesh organises its components into planes. The Mitigation Manager belongs to the **Security Operations Plane**, together with the Playbooks Tool (PT) and the Workflow Orchestrator (WO). The plane's purpose is to evaluate the information gathered, classified, analysed and enriched by the planes beneath it and to translate that information into enforced courses of action.

Figure 1 situates the Mitigation Manager within the relevant Resilmesh planes and shows its principal interactions with the components involved in mitigation.

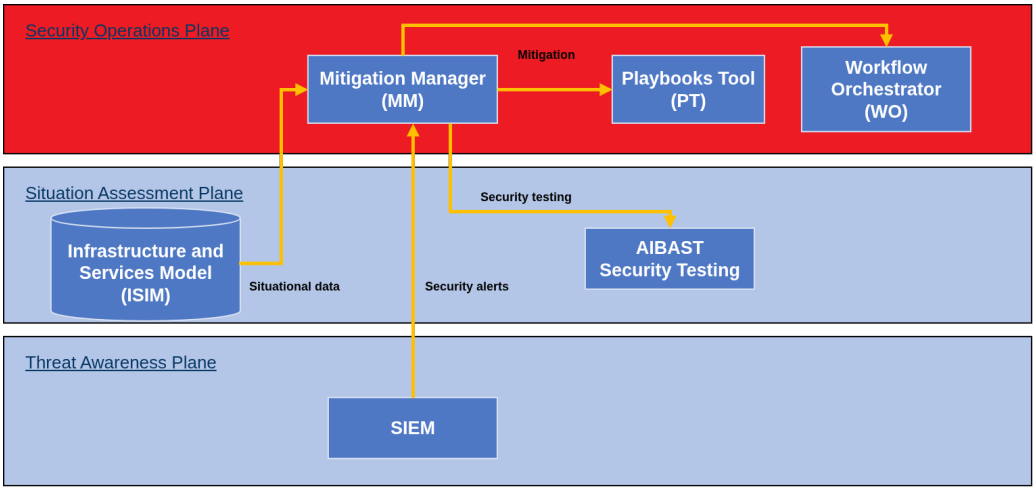


Figure 1. Mitigation flow within the Resilmesh architecture.

Figure 2 shows the Mitigation Manager's internal structure and its main interfaces with other Resilmesh components.

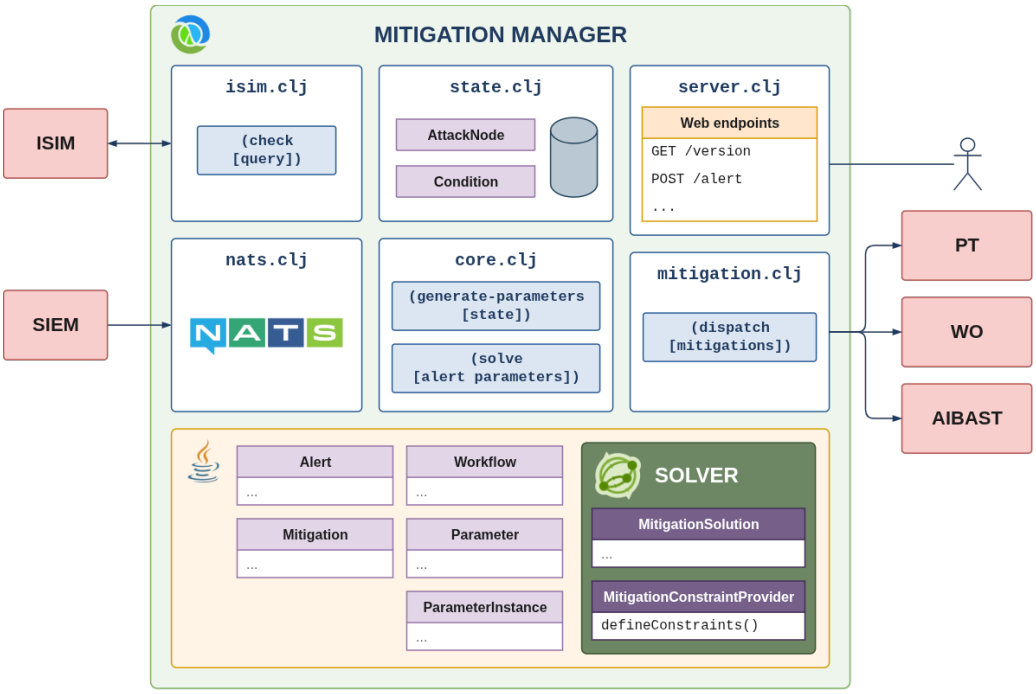


Figure 2. Mitigation Manager architecture.

The Mitigation Manager's interfaces reflect this position:

- **Inbound: the SIEM.** Security alerts reach the Mitigation Manager through the platform's NATS[2] message broker. Since the second version, this is the single entry point for the mitigation flow, which simplified the component's coupling with the rest of the architecture.
- **Lateral: ISIM.** The Infrastructure and Services Information Model provides the cyber situational awareness picture. The Mitigation Manager queries it on demand to validate the conditions of attack steps and to compute playbook parameters, such as the address of the server that must be protected.
- **Outbound: enforcement.** Selected playbooks are dispatched to the Playbooks Tool for straightforward, well-scoped actions, or to the Workflow Orchestrator for longer-running procedures. The SOARCA[3] engine contributed by X-MESH is available as an additional backend for CACAO playbooks. Enforcement typically ends in OpenC2 commands delivered to actuators on the affected hosts or network devices.
- **Outbound: security testing.** The AI-based Automated Security Testing (AIBAST) component can be invoked as a consequence of a mitigation, allowing the platform to verify that an exploited weakness has in fact been closed.

> *The Mitigation Manager decides; it does not act on its own. Every countermeasure is executed by a playbook that operators have authored, reviewed and registered with the platform. The component's contribution is to choose, parametrise and time the execution of those playbooks.*

3. How the Mitigation Manager Works

The mitigation process follows the same sequence for every alert: understand the alert, update the picture of ongoing attacks, plan a response, and enforce it.

3.1 Understanding alerts

SIEM alerts are large, heterogeneous JSON documents whose fields depend on the log source and the decoder that parsed it. The Mitigation Manager reduces each alert to a small, consistent set of named fields through a declarative **field mapping**. The mapping mirrors the shape of the alert and states which fields are of interest and under which name they should be remembered. Because different decoders reuse the same field names for different meanings, the mapping can be made conditional on the decoder that produced the alert.

Two families of information are always extracted: the **MITRE ATT&CK technique identifiers** attached to the alert by the SIEM ruleset, which drive attack recognition, and the **operational data** (addresses, ports, users, file paths, process identifiers) that later become playbook parameters. Where the alert carries richer structured content, such as a STIX 2.1 bundle produced by an upstream detector, the same mechanism exposes that content for later processing.

[2] https://docs.nats.io/
[3] https://cossas.github.io/SOARCA/docs/

3.2 Tracking attacks with attack graphs

The Mitigation Manager's model of an attack is a graph. Each node represents an adversary step, described in prose and tagged with one or more ATT&CK techniques; edges describe the order in which steps can follow one another. This structure is deliberately close to the MITRE Attack Flow data model, so that attack patterns documented by threat-intelligence teams in Attack Flow can be turned into graphs the component can track.

A node carries two further elements:

- **Conditions** that must hold for the step to be considered confirmed. A technique identifier alone is weak evidence: an alert tagged as exploitation of a public-facing application should only advance an attack graph if, for instance, the targeted address really is a public-facing service according to ISIM. Conditions can inspect the alert, the situational model, or both.
- **Extractions** that record facts about the confirmed step for later use. When a reconnaissance node is confirmed, for example, the attacker's address is extracted into the attack's **context**, so that a firewall playbook triggered several steps later still knows whom to block.

When an alert arrives, the component first checks every attack already being tracked to see whether the alert confirms one of its expected next steps. It then checks whether the alert matches the starting step of any known attack graph, in which case a new attack instance is created. Attacks that have reached the end of their graph are retired. The result is a live picture of every attack in progress, how far each has advanced, and what each is likely to do next.

Not every alert needs a multi-step graph. Attack patterns that warrant an immediate, self-contained reaction are represented as single-node graphs, so that the same machinery covers both one-shot mitigation and long-running attack tracking.

3.3 Choosing a response

The candidate countermeasures are **workflows**: descriptions of CoA playbooks that state which ATT&CK techniques they counter, what parameters they require, what conditions must hold for them to be applicable, and what their execution costs. Cost is an abstract measure supplied by the playbook author; it captures how disruptive the action is to normal operations rather than any monetary figure.

For every attack in progress, the Mitigation Manager instantiates the workflows that could apply, filling in their parameters from the alert and from the attack's context. Instances whose parameters cannot be resolved, or whose conditions fail, are set aside. The remaining instances form the search space handed to the AI planning engine, which is described in Section 4.

3.4 Enforcing the response

The planning engine returns a set of mitigations, each pairing an alert with a fully parametrised workflow instance. The Mitigation Manager dispatches each one to the appropriate backend: a webhook on the Playbooks Tool, a request to the Workflow Orchestrator, or a CACAO playbook trigger on SOARCA. The backend executes the

playbook's steps, which typically include gathering additional evidence, consulting external services such as reputation databases, and finally issuing OpenC2 commands to actuators. The component records the outcome of each execution.

4. AI for Mitigation: The Planning Engine

4.1 Why planning rather than rules

The first version of the Mitigation Manager selected countermeasures with a rule-based procedure: filter workflows by matching technique, discard those whose conditions fail, and pick the cheapest that remains. This works well when one alert maps to one mitigation, but it does not generalise. It cannot weigh two mitigations against each other when applying both would be wasteful, it cannot decline to act when no acceptable option exists, and every new operational requirement has to be coded into the selection procedure itself.

The second version replaced this procedure with an **AI planning engine** based on constraint-based optimisation, implemented with the OptaPlanner[4] library. The engine treats mitigation selection as a search over possible response plans, guided by metaheuristic algorithms such as Tabu Search, Simulated Annealing and Late Acceptance. These techniques are well established in domains where resources are finite, requirements conflict and the search space is large, which is exactly the situation of a security operations centre deciding how to respond.

4.2 The mitigation problem as a planning model

A planning problem in this paradigm is expressed in four kinds of element, and each has a direct counterpart in the Mitigation Manager:

- **Planning facts** are the information that does not change during a planning run. Here they are the incoming alerts and the pool of instantiated workflows described in Section 3.3.
- **Planning entities** are the elements the solver is allowed to arrange. Here they are **mitigations**: slots that the solver may fill or leave empty. A solution therefore contains zero or more mitigations, up to a configurable maximum.
- **Planning variables** are the attributes of an entity that the solver changes. Each mitigation has two: the alert it addresses and the workflow instance it applies.
- **Constraints** score a candidate solution. **Hard constraints** must never be violated; a plan that breaks one is infeasible. **Soft constraints** express preferences; among feasible plans, the solver prefers the one with the best soft score.

In the current model, the hard constraints require that every alert in the problem is addressed by at least one mitigation, and that no mitigation pairs an alert with a workflow that is inapplicable to it or that has unresolved parameters. The soft constraint minimises the total execution cost of the selected workflows. The combined effect is that the engine finds the cheapest plan that fully handles the alerts, and reports when no such plan exists.

[4] https://docs.optaplanner.org/9.43.0.Final/optaplanner-javadoc/index.html

4.3 What the planning engine makes possible

Several behaviours follow from this formulation that the earlier rule-based procedure could not offer:

- **Declarative operational policy.** New requirements are added as constraints rather than as code paths. A rule such as “never isolate a host tagged as mission-critical in ISIM” or “prefer network-level over host-level mitigations for this asset class” becomes one more constraint, and the solver takes it into account alongside every other.
- **Explicit search space.** Instead of pre-filtering candidates statically, the Mitigation Manager hands the solver the full set of instantiated workflows. This allows combinations to be found that a fixed filtering order would have discarded early.
- **Bounded reaction time.** The solver runs under a configurable time budget. For time-sensitive attacks, the budget can be tightened and the best plan found within it is used; for low-severity alerts it can be relaxed in exchange for better optimisation.
- **Multiple alerts per plan.** The model accepts more than one alert in a single run, enabling batches of related alerts to be resolved together rather than one at a time.
- **Permission to do nothing.** A feasible plan may contain no mitigations if none is required, and the solver may report that no feasible plan exists at all. In that case the component logs an explanation of which constraints could not be satisfied, which is the starting point for an operator to relax a constraint or add a playbook.

> *The Artificial Intelligence in the Mitigation Manager is a constraint solver, not a learned model. Its inputs, constraints and scoring are explicit, its decisions are reproducible, and the reasons for accepting or rejecting a plan can be inspected. This matters for a component that is authorised to act on production infrastructure: operators must be able to audit why a countermeasure was chosen, and to change the rules that led to it.*

4.4 The role of situational awareness

The planning engine is only as good as the search space it receives, and that search space is shaped by the cyber situational awareness picture. ISIM contributes at two points. During attack tracking, conditions can query ISIM to confirm that an alert is consistent with the infrastructure, for example that a targeted address belongs to a network the platform’s firewall can actually protect. During workflow instantiation, parameters can be computed from ISIM, for example locating the directory server that must be told to disable a compromised account. Risk scores maintained by ISIM are intended to feed the same channel, allowing the planner to weigh the criticality of affected assets when scoring plans.

5. Evolution Across the Project

Table 2 summarises the component’s evolution across the project.

Stage	Period	Key developments
Version 1 (D5.7)	Up to December 2024	Attack graph model with attack and condition nodes; attack-front tracking across alerts; rule-based workflow selection by technique and cost; Shuffle[5] as playbook engine; validation in Pilot 1.

Stage	Period	Key developments
Version 2 (D5.8)	Up to October 2025	Re-implementation on the JVM; introduction of the AI planning engine; streamlined interfaces (SIEM as sole inbound source, ISIM as unified situational and risk source); AIBAST link.
Pilot 2	Spring to summer 2026	STIX-based alerts from correlated detectors; ISIM-driven playbook parameters; multi-technique playbooks.
Current state	September 2026	Unified query model for conditions and parameters; attack instances with persistent context; planning driven by tracked attacks; decoder-aware alert parsing; X-MESH SOARCA integration with CACAO playbook import; Attack Flow import under development.

Table 2. Evolution of the Mitigation Manager.

5.1 Version 1: establishing the model

The first deliverable set out the conceptual foundations that still hold: the Mitigation Manager as the decision-making component of the Security Operations Plane, attack graphs modelled on MITRE Attack Flow as the means of tracking multi-step attacks, and workflows as the bridge between the decision and the CoA playbooks that enforce it. The implementation was a Python service, and workflow selection followed the rule-based procedure described in Section 4.1. The document already identified planning-based selection, MITRE D3FEND [3] countermeasure matching, and time-aware attack modelling as promising future directions.

Version 1 was validated in **Pilot 1**, a civil regional infrastructure scenario in which the platform faced a denial-of-service attack and a credential brute-force attack against public services. The Mitigation Manager tracked both against attack graphs derived from Attack Flow descriptions provided by the pilot's security team and reacted by triggering a playbook that reconfigured a virtual firewall to block the offending traffic. The pilot confirmed that the end-to-end reactive flow, from SIEM alert to enforced countermeasure, operated automatically and within a time frame suitable for the scenario.

The resulting end-to-end mitigation flow is illustrated in Figure 3.

[5] https://shuffler.io/docs

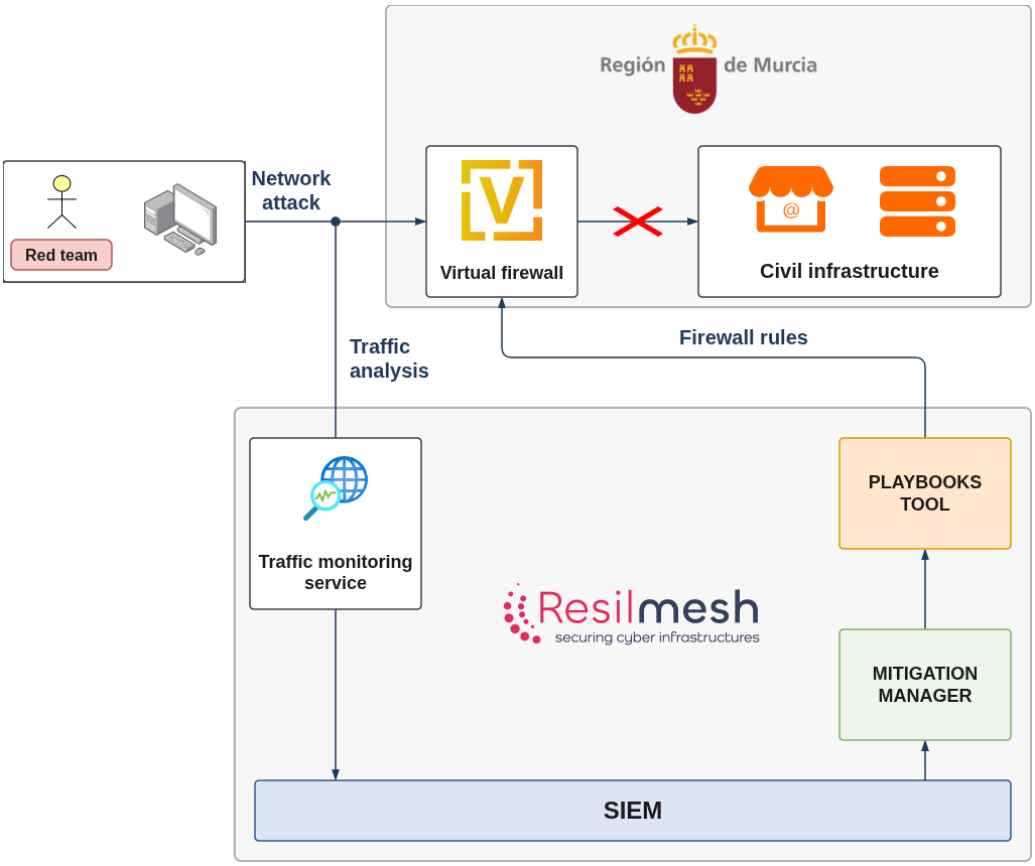


Figure 3. Pilot 1: Automated mitigation protecting civil infrastructure.

5.2 Version 2: introducing AI planning

The second deliverable delivered the most significant architectural change: the component was rewritten as a joint Clojure and Java service in order to embed the OptaPlanner engine, and mitigation selection became the planning problem described in Section 4. The external interfaces were simplified at the same time. The SIEM became the sole source of mitigation requests, ISIM absorbed the risk-score function previously attributed to a separate component, and a direct link to AIBAST was added.

5.3 Pilot 2

Where Pilot 1 validated the architecture, **Pilot 2** exercised its versatility against a wider range of attack patterns: exploitation of a public-facing web application, active reconnaissance of known vulnerabilities, phishing, and credential attacks against a directory service. Two developments characterise this stage.

First, alerts became richer. The pilot's detectors emitted **STIX 2.1 bundles** describing the observed attack, its indicators and the assets involved. The Mitigation Manager's field mapping and extraction mechanisms were used to draw attacker and victim identities, targeted URLs and account names out of these bundles into the attack context.

Second, playbooks became more dependent on situational awareness. Disabling a compromised directory account, for instance, requires knowing which server holds the directory; that parameter is resolved at planning time by querying ISIM for the relevant service and selecting the instance within the affected network. Several playbooks were also

registered against more than one ATT&CK technique, reflecting that a single countermeasure, such as blocking an address at the firewall, is an appropriate response to several distinct adversary behaviours.

5.4 The current state

Development since the second deliverable has concentrated on the data model that feeds the planner. Conditions and parameters, previously two different mechanisms, are now expressed in a single **query** language that can read from the alert, from the attack's context, or from an arbitrary expression combining both. Attacks in progress are first-class, persistent objects that carry their own context, and workflow instances are generated per tracked attack rather than per raw alert, so that the planner only ever sees countermeasures grounded in an attack the component actually recognises. Alert parsing became decoder-aware, allowing one mapping to serve several log sources whose fields collide. Work is under way on a parser that turns MITRE Attack Flow documents, including their branching and merging operators, directly into trackable attack graphs.

The **X-MESH** Open Call project contributed a CACAO-native SOAR environment built around the SOARCA execution engine and CACAO Roaster[6]. The Mitigation Manager was extended so that CACAO playbooks stored in SOARCA are imported as workflows automatically: the ATT&CK techniques a playbook declares become the techniques it counters, its externally settable variables become parameters resolved from the attack context, and its declared impact becomes its cost. Playbooks selected by the planner are then triggered through SOARCA's API with correctly typed CACAO variables. This brought the Mitigation Manager into the Open XDR[7] standards stack (CACAO, STIX 2.1, OpenC2) that the Interworking Mesh is built on.

6. Open Standards and Interoperability

The Mitigation Manager's design leans on open standards at every interface, which is what allows it to be extended by third parties such as X-MESH:

- **MITRE ATT&CK** [1] is the vocabulary that links detection to response. Alerts are tagged with techniques by the SIEM ruleset, attack graph nodes are defined in terms of techniques, and playbooks declare the techniques they counter. This common vocabulary is what makes it possible to add a new playbook without touching the attack graphs, or a new attack graph without touching the playbooks.
- **MITRE Attack Flow** [2] is the model for attack graphs, so that threat-intelligence products authored in Attack Flow can be consumed with minimal translation.
- **STIX 2.1** [4] is the format in which richer detections arrive, and the format of the indicators the platform's CTI infrastructure shares.
- **OASIS CACAO** [5] is the playbook format of the X-MESH backend; the Mitigation Manager reads CACAO metadata to derive workflow descriptions automatically.
- **OpenC2** [6] is the command language in which playbooks ultimately act on actuators.

[6] https://github.com/opencybersecurityalliance/cacao-roaster
[7] https://github.com/opencybersecurityalliance/oxa

> *Together these standards place the Mitigation Manager within the Open XDR Architecture that underpins the Resilmesh Interworking Mesh. For organisations subject to the NIS2 Directive [7], this supports the incident-handling obligation of Article 21(2)(b) through automated response that is documented, reproducible and expressed in machine-readable, standards-based form.*

7. Limitations and Roadmap

The Mitigation Manager is a working system, and the directions in which it is growing are as much a part of its design as the capabilities already described. The following areas are under active development or envisaged for the next stages of the component.

Batched and configurable planning. The planning model already accepts several alerts per run. The next step is to group related alerts into a single planning run, and to expose the maximum number of mitigations and the solver's time budget as deployment configuration, so that operators can tune reaction time and thoroughness to each environment.

A reusable situational query framework. ISIM is consulted through queries embedded in attack graph conditions and workflow parameters. We envision a library of reusable, workflow-driven ISIM queries, as outlined in D5.8, so that a new playbook can draw on situational data by naming the question it needs answered rather than writing the query itself.

Hierarchy-aware technique matching and D3FEND mapping. Matching now exploits the ATT&CK hierarchy: an attack graph node or playbook that names a technique is also satisfied by alerts tagged with any of its sub-techniques, which makes both easier to author and more broadly applicable. Building on this, we envision mapping countermeasures through MITRE D3FEND, as proposed in D5.7, so that playbooks can be matched to attacks by the defensive technique they implement rather than by hand-listed offensive techniques.

A growing attack graph library. The component reacts to alerts that match a known attack graph. This is by design, so that it never acts on alerts the platform does not understand, and it means that operational value grows with the library. The Attack Flow importer under development will let attack patterns documented by threat-intelligence teams be turned into trackable graphs directly, and single-node graphs offer a lightweight way to register alert families that warrant an immediate response.

Richer parameter binding for imported playbooks. Playbooks imported from SOARCA resolve their variables from the attack context, which keeps them grounded in an attack the component has recognised. We plan to document this contract and to let imported playbooks declare alert-sourced fallbacks for their variables, so that they become applicable to a wider range of attack graphs without changes to the graphs themselves.

Closing the loop on effectiveness. The component records whether a playbook executed successfully. The link to AIBAST offers a way to go further and verify that an exploited weakness has in fact been closed. We envision feeding that verification back into the planner's view of the attack, so that a mitigation that did not achieve its purpose leads to a further planning round rather than a closed case.

Large language models in the decision process. Work is under way to explore whether large language models could support the decision-making stage. A model might improve workflow selection by adjusting the expected cost of each playbook to the alert at hand, pre-ranking the candidate workflow instances, or proposing weights for the soft constraints, so that the planning engine starts from better-informed numerical values. It might also help explain a chosen plan by presenting the reasoning behind it as a narrative that an operator could read and question, or as structured data that other tools could process.

8. Conclusions

The Mitigation Manager demonstrates a practical way of putting Artificial Intelligence at the centre of automated incident response without giving up control or auditability. Attack graphs modelled on MITRE Attack Flow give the component an understanding of attacks as sequences rather than isolated events. Cyber situational awareness from ISIM grounds its decisions in the real infrastructure. A constraint-based planning engine turns the choice of countermeasure into an explicit optimisation whose rules operators can read, change and extend. Open standards at every interface, from ATT&CK through STIX and CACAO to OpenC2, let the component consume third-party intelligence and drive third-party enforcement, as the X-MESH integration showed.

Across two deliverables and two pilots, the component has moved from a rule-based prototype to a planning-driven decision engine validated against realistic attack scenarios. The developments set out in the previous section build on that working system, and each of them fits within the same architecture, extending what the planner can see and act on without changing how it decides.

Glossary

Term	Meaning in this document
AIBAST	AI-based Automated Security Testing, a Resilmesh component that can be invoked after a mitigation to verify its effect.
AI planning engine	A constraint solver that treats mitigation selection as an optimisation problem. It was introduced in v2 (D5.8) on top of the OptaPlanner library.
Attack context	The facts extracted from confirmed attack steps and remembered for the lifetime of a tracked attack.
Attack graph	A directed graph of adversary steps, each tagged with MITRE ATT&CK techniques and guarded by conditions, used to recognise and follow an attack across several alerts. Modelled after the MITRE Attack Flow data model, it has been present since v1 and progressively enriched.
CACAO	OASIS Collaborative Automated Course of Action Operations, a standard format for security playbooks.
CoA playbook	A Course of Action playbook: a sequence of security actions executed by the Playbooks Tool, the Workflow Orchestrator or SOARCA.
Hard / soft constraint	Rules a mitigation plan must satisfy versus preferences the planner optimises among feasible plans.
ISIM	The Infrastructure and Services Information Model: a graph database describing devices, services, users and their risk scores. This Situation Assessment Plane component is consulted by the Mitigation Manager during planning.

Term	Meaning in this document
Mitigation Manager (MM)	The core Resilmesh component that acts on incoming security alerts and provides mitigation actions as output. Led by the University of Murcia, it has two released versions (D5.7 and D5.8) and continued development through Pilot 2 and beyond.
OpenC2	Open Command and Control, the language in which playbooks issue commands to actuators.
OptaPlanner	The open-source constraint-solving library underlying the AI planning engine.
Planning fact / entity / variable	The immutable inputs, the arrangeable elements, and the attributes the solver assigns, respectively.
PT / WO	Playbooks Tool and Workflow Orchestrator, the Security Operations Plane components that execute playbooks.
SOARCA	The open-source CACAO-native orchestration engine contributed through X-MESH.
STIX 2.1	Structured Threat Information Expression, the format of rich detections and shared indicators.
Workflow	The Mitigation Manager's parametrised description of a CoA playbook: the ATT&CK techniques it counters, its parameters, conditions and cost. Workflows are executed by the Playbooks Tool (Shuffle) or, through the X-MESH integration, by the SOARCA engine using OASIS CACAO playbooks.
X-MESH	The Open Call 2 extension of the Resilmesh Interworking Mesh, providing a CACAO-native SOAR environment comprising SOARCA and CACAO Roaster. It is integrated with the Mitigation Manager as a second playbook backend.

Table 3. Glossary of terms.

References

1. B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington and C. B. Thomas. *MITRE ATT&CK: Design and Philosophy*. MITRE, March 2020.
2. MITRE Center for Threat-Informed Defense. *Attack Flow, version 4.0.*
3. P. E. Kaloroumakis and M. J. Smith. *Toward a Knowledge Graph of Cybersecurity Countermeasures*. MITRE, 2021.
4. OASIS Open. *STIX Version 2.1*. OASIS Standard, 10 June 2021.
5. OASIS Open. *CACAO Security Playbooks Version 2.0*. Committee Specification 01, 27 November 2023.
6. OASIS Open. *Open Command and Control (OpenC2) Language Specification Version 1.0*. Committee Specification 02, 24 November 2019.
7. European Parliament and Council of the European Union. *Directive (EU) 2022/2555 of 14 December 2022 on measures for a high common level of cybersecurity across the Union.* Official Journal of the European Union L 333, 27 December 2022.

Resilmesh is funded by the European Union. Views and opinions expressed are those of the project consortium and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.