Conducttr™

Cognitive Warfare Exercises in a Multi-Domain Context

HLA INTERACTION
using Conducttr Federate Application

INTEGRATION WITH VR FORCES

COGNITIVE MODELLING
Actors + Auto-Response
Ai-Powered Role-Playing

CONDUCTTR API
3rd Party Application Control

www.conducttr.com

About Conducttr

Conducttr is a hybrid warfare and crisis exercise platform for realistic command post, field and classroom exercises. Its synthetic internet provides a virtual information environment for training and exercising in information operations, media operations, social media, OSINT, cyber, CIMIC, foreign affairs, homeland security, counter-terrorism, and humanitarian disaster relief.

Conducttr can be used on its own or alongside operational systems as a digital wrap for live exercises lasting from an hour to several weeks.

www.conducttr.com

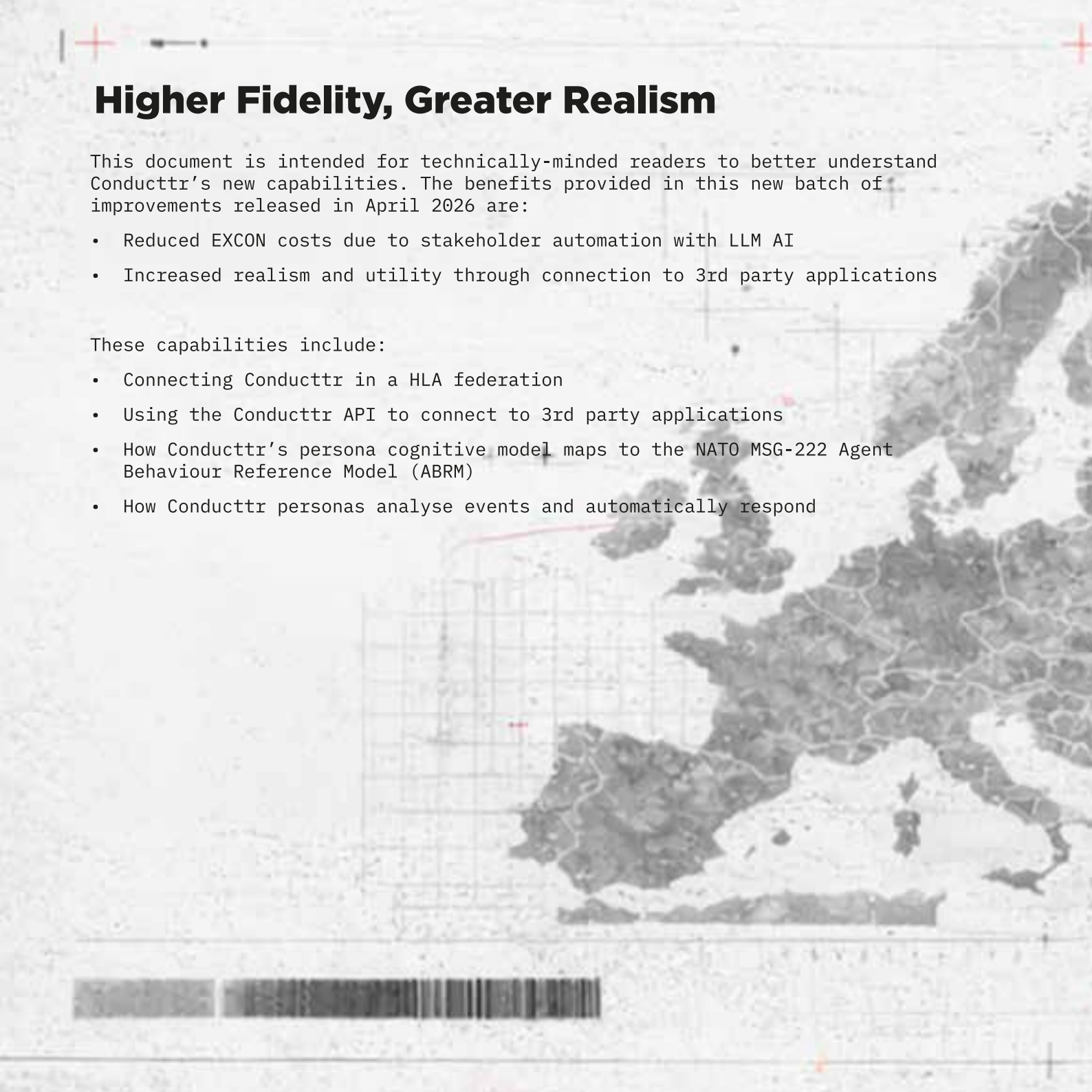
Higher Fidelity, Greater Realism

This document is intended for technically-minded readers to better understand Conducttr's new capabilities. The benefits provided in this new batch of improvements released in April 2026 are:

- Reduced EXCON costs due to stakeholder automation with LLM AI
- Increased realism and utility through connection to 3rd party applications

These capabilities include:

- Connecting Conducttr in a HLA federation
- Using the Conducttr API to connect to 3rd party applications
- How Conducttr's persona cognitive model maps to the NATO MSG-222 Agent Behaviour Reference Model (ABRM)
- How Conducttr personas analyse events and automatically respond



Connecting Conducttr to a HLA federation

Conducttr and VR-Forces can be integrated to create a single exercise environment in which physical activity and information effects shape each other. Using HLA and the VR-Forces remote control interface, Conducttr monitors events in the simulated kinetic environment, translates them into audience and information responses, and triggers new activity back into the simulation. The result is a closed-loop exercise in which the physical and cognitive domains interact as part of one orchestrated scenario.

Multi-domain exercises

Crises unfold across multiple domains, with physical activity, human perception, media activity, public behaviour, and institutional response all shaping the overall situation. Multi-domain exercising is important because it allows organisations to train in a way that reflects this wider operating environment.

By linking kinetic activity with information effects and audience response, a multi-domain exercise gives participants a fuller picture of consequence, escalation, and second-order effects. It helps teams understand how events are interpreted, how narratives spread, how populations react, and how decisions in one domain can influence conditions in another. This supports more realistic training and strengthens judgement, coordination, and preparedness.

How it works

The integration operates in two stages: the exercise design and build, in which the kinetic and information environments are prepared around a shared training objective, and the multi-domain execution, in which Conducttr and VR-Forces interact to run the exercise as a connected experience.

Exercise design and build

The design process begins with the training objective, exercise audience, and geopolitical context. These define the purpose of the exercise, the operating environment, and the kinds of interaction, escalation, and decision-making the exercise is intended to explore.

From this common design basis, two connected exercise paths are developed:

- in Conducttr, the information environment is prepared, including actors, audiences, beliefs, attitudes, narrative conditions, and behavioural expectations.

- in VR-Forces, the kinetic environment is prepared, including terrain, entities, movement, locations, and operational events.

The next step is to define how those two parts relate to each other. This includes identifying which physical events should generate information effects, which audiences are affected by activity in particular locations, and which information actions should trigger new activity in the simulated kinetic environment. This is where the cross-domain mechanics of the exercise are designed.

The integration is managed through the Conducttr HLA federate and its supporting files:

- a .env file provides the runtime and connection settings for the federate

- a IEData.xml file defines the starting information environment for the exercise, including actors, audiences, beliefs, attitudes, events, and their initial attributes

- a contract.xml file defines the event-response relationships that will govern how the federate interprets received HLA interactions and object state changes and what Conducttr actions follow from them

This design stage ensures that the exercise begins with a coherent operational scenario, a defined information environment, and a clear set of relationships between activity in the physical and cognitive domains.

Multi-domain execution

At run time, the Conducttr HLA federate loads its configuration and joins the RTI.

Now initialised, the Conducttr federate subscribes to the changes identified in the context.xml file and begins listening. As relevant activity is detected in the kinetic environment, Conducttr interprets that activity in terms of audience response, narrative development, and information effect.

If Conducttr needs to trigger activity in VR-Forces it does so via the remote control interface API.

ITEC 2026 Use Case - Conducttr & VR Forces

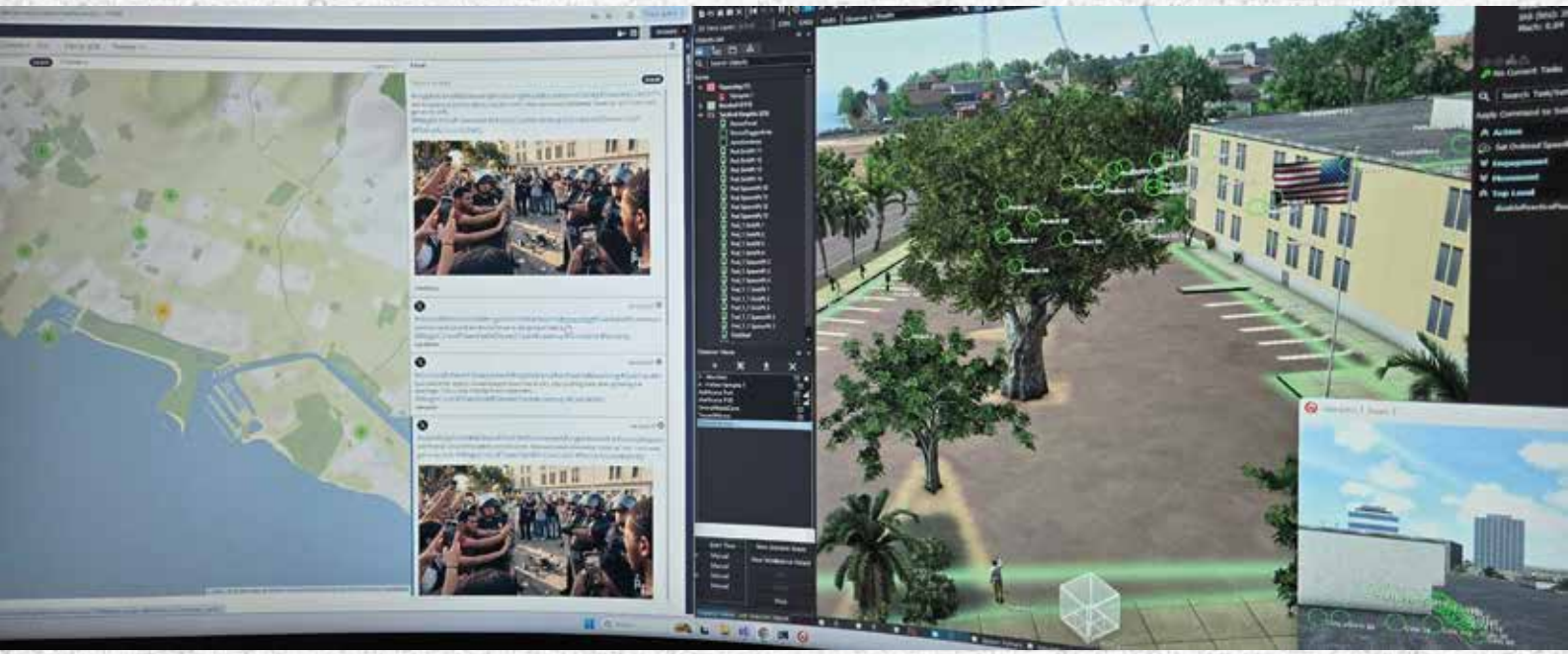
This section describes a use case demonstrated at ITEC 2026.

Scenario context

The ITEC 2026 demo is set in a fictional crisis between Goldland and Redland. Goldland is presented as an island nation whose central town square in the capital functions as both a busy civic space and a visible symbol of normal public life. Redland is applying pressure through a coordinated campaign of public messaging, diplomatic signalling, and online activity designed to unsettle Goldland politically and shape perceptions of weakness. The purpose is not immediate large-scale attack, but controlled escalation that creates strategic effect through a combination of visible kinetic action and information pressure.

This makes the town square an important setting for the demo: it is a physical location that can be represented clearly in VR-Forces, but it is also a symbolic and social space in which audience reaction matters. Activity there is visible, interpretable, and politically resonant. A drone entering that space is therefore more than a movement event. It is also a “message”. It attracts attention, prompts speculation, and creates conditions in which media content, social reaction, fear, agitation, and political messaging can all develop around the incident.

The exercise is built around that logic of controlled escalation. The square begins in a calm baseline state with civilians present and Redland’s messaging already circulating in the background. A Redland UAV then launches and travels across the marina towards the square. Once it enters the urban area, the square becomes the focal point of attention, the Goldland government website is hacked for psychological effect, and more people gather. The crowd grows while tension builds. Redland then releases a



small munition over the square. The blast is limited in physical damage but significant in interpretive effect. Competing explanations spread, pro-Goldland activists intensify calls to action, and the public space transitions into unrest.

Technical implementation details

The demonstration has the following components:

- Exercise Control (Conducttr): Acts as the master orchestration layer. It manages the Master Event List (MEL), publishes narrative injects, and issues API-based cues to advance the scenario.
- Integration Layer (Conducttr HLA Federate): A middleware component that bridges the exercise control and simulation domains. It interacts with Conducttr via API and communicates with the federation via the MAK RTI.
- Information Environment (Conducttr): Simulates audience and internet activity
- Kinetic Environment (VR-Forces): Executes the kinetic entities and environment behaviors through its remote control interface.
- Federation Object Model (FOM): The demo utilizes a specific stack for interoperability:
- INFO FOM
- RPR 2.0 Base FOM
- RPR 2.0 Warfare FOM

Exercise control

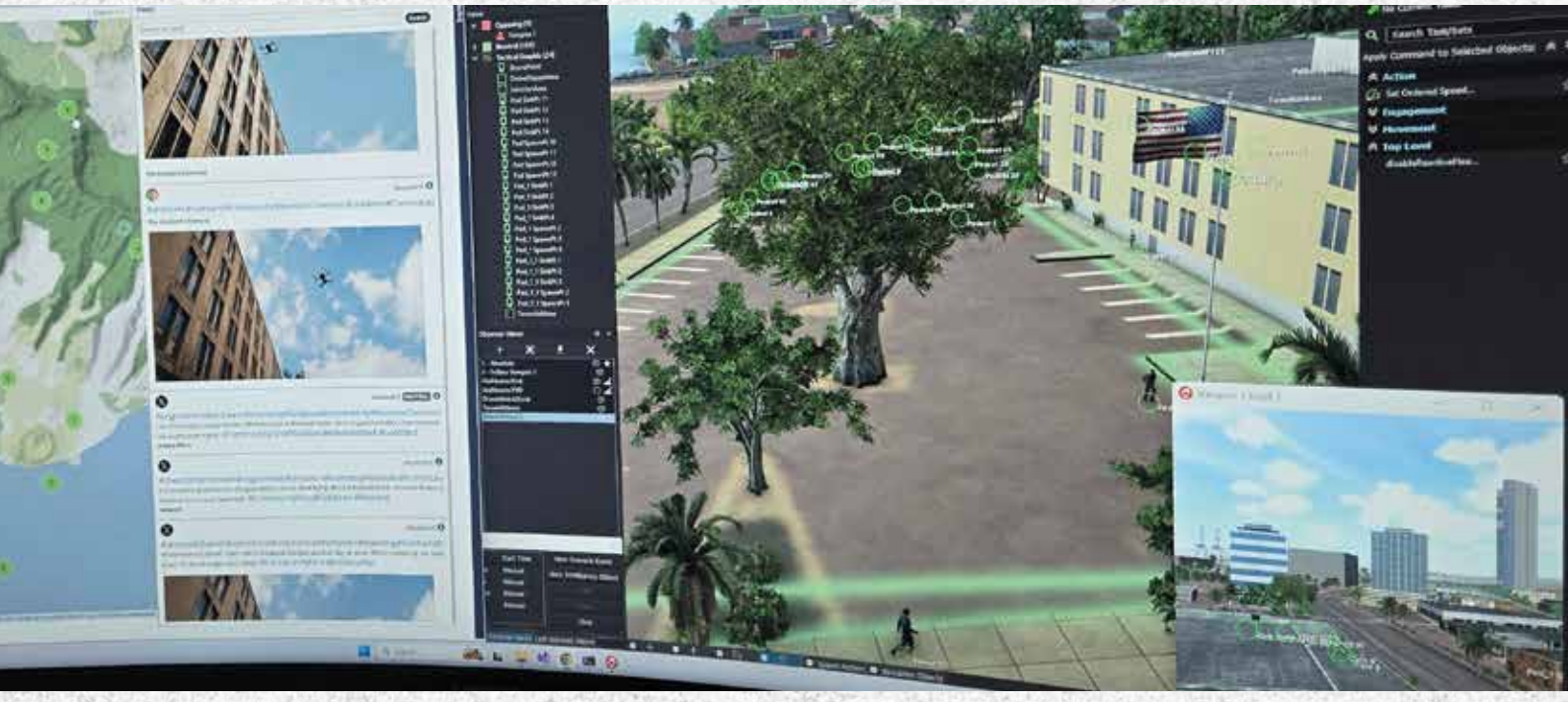
The scenario follows a command-execution-confirmation pattern to ensure synchronisation between the narrative and the kinetic simulation:

- Initiation: Conducttr publishes a STARTEX narrative event.
- Execution: The federate catches this event via Conducttr API and triggers the LaunchDrone Scenario Event in VR-Forces.
- Confirmation: The federate catches the kinetic activity in VR Forces and publishes IE reactions and alerts the exercise facilitator.

State Monitoring & Logic Triggers

The federate monitors the "kinetic picture" in real-time to drive information-domain consequences:

- Entity Tracking: The system subscribes to HLAobjectRoot.BaseEntity.PhysicalEntity (specifically WorldLocation and EntityIdentifier).
- Geofencing: The federate tracks the UAV's proximity to the Town Hall. Upon arrival, the location-based logic triggers the next phase of information effects (e.g., a website hack and public reaction).
- Event Confirmation: The system uses HLAinteractionRoot.Warfare.MunitionDetonation (via DetonationLocation) to verify kinetic outcomes.



Cognitive Modelling in Conducttr

Conducttr represents human actors and their behavioural responses through two tightly coupled subsystems: a Cognitive Model that defines the structural attributes of each actor, and a Cognitive Process Engine that processes operational stimuli and generates persona-authentic reactions.

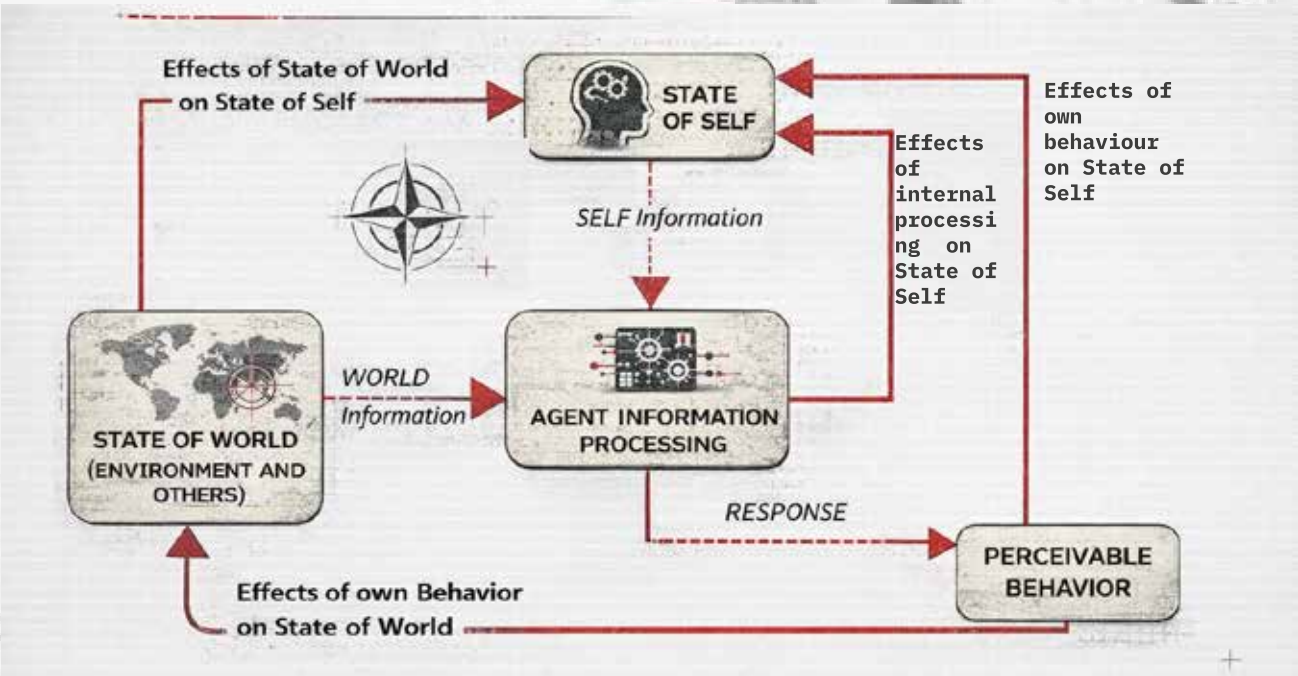
This approach maps closely to NATO's Agent Behaviour Reference Framework where State of Self is the persona cognitive model and the Agent Information Processing is performed by Conducttr LLM AI prompts. All use of AI is within UK MoD guidelines and the rationale for all persona responses is available for human review.

This section describes:

- The ABRF
- Conducttr's cognitive model
- Conducttr's agent information processing (cognitive processing engine)

The diagram below shows the ABRF.

Agent Behaviour Reference Framework(ABRF)



The ABRF represents human behaviour from the perspective of a single agent. It links the agent's internal state, the external world, information processing, and observable behaviour, showing how perception, judgement, and action might interact in a continuous feedback loop shaped by both the environment and the agent's own behaviour.

Conducttr and the NATO Agent Behaviour Reference Framework

The table below compares Conducttr to the Agent Behaviour Reference Framework (ABRF)

ABRM COMPONENT	CONDUCTTR EQUIVALENT	EXPLANATION
State of Self	 COGNITIVE MODEL	Found inside each Conducttr persona - see below
Information Processing	 COGNITIVE PROCESS ENGINE	How the persona thinks and feels.
Perceivable Behaviour	 GENERATED OUTPUT	What the persona says or does.
State of World	 THE FACILITATOR	At the time of writing Conducttr works with a human-in-the-loop approach expecting a human to inform the AI persona of details they think are important and might warrant a response.

Persona Cognitive Model

Personas are Conducttr's "non-player characters". The "State of Self" attributes fall into the following categories:

- **Identity** – stable characteristics defining who the persona is, including background, role, affiliation, and narrative identity.
- **Cognition** – the persona's beliefs, knowledge, values, and mental models used to interpret events and information.
- **Motivation** – goals, incentives, and priorities that drive decision-making and behavioural intent.
- **Affect** – emotional and psychological state variables such as morale, anger, fear, vigilance, and stress.

- **Social** – relationships, influence roles, group membership, and the persona's position within social or factional structures.
- **Memory** – stored experiences, learned information, and evolving narrative context that shape future interpretation and behaviour.
- **Physical** – operational and situational attributes of the persona such as location, activity, capability constraints, and physical condition.

Conducttr's Agent Information Processing (Cognitive Processing Engine)

In the ABRF, this is the "agent information processing" process. The table below show's Conducttr's current 4-step approach in which the persona moves from determining the revelance of an event to generating a response.

Relevance: The engine first asks, "Does this matter to me?" It uses the Elaboration Likelihood Model to decide if the persona should think deeply about the event or just react instinctively.

Resonance: If it's relevant, the engine checks how it feels. It uses Appraisal Theory (OCC Model) to map the event against the persona's goals and beliefs. Does this make them angry? Scared? Happy?

Reaction: This is the internal "shaking of the fist." Based on the Transactional Model of Stress and Coping, the persona evaluates if they can handle the situation. They choose a coping strategy (e.g., "I'll ignore this" vs. "I'll lash out").

Response: The final outward action. This is the Perceivable Behaviour of the ABRF which in Conducttr's case is likely in the information space - a social media post, an email, or to do nothing.

Connecting Conducttr to 3rd Party Applications

Conducttr's API allows third party applications to interact with Conducttr – sending messages, triggering audience reactions and getting simulation state information.

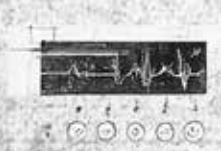
Conducttr includes a suite of endpoints that enable:

- Publication of messages
- Exchange of smartword array data for dynamic messaging with existing injects
- Control of pattern of life
- Interrogation of simulation space to reveal current status

A full explanation and details can be found at https://helpdocs.conducttr.com/feature-documentation/api/possible-use-cases

Publication of messages

Conducttr publishes messages to a team, on a channel, from a persona (actor or organisation). In order to publish anything applications need to GET teams and GET personas to ensure they have the right details. The API works with the exercise simulation space, not the exercise library. This means the exercise space must have a published exercise to work. This is published from the exercise Editor by the scenario designer.



Endpoint Reference

Personas

GET /personas

Use: Retrieves a list of all personas available within the specific simulation space.

Mechanism: The endpoint returns a presigned URL. The user must download the ZIP file from this URL and parse the contained JSON file to access persona details (names, IDs, and attributes).

Teams

GET /teams

Use: Returns all active teams defined in the simulation.

Data Structure: Teams are categorized into three types:

- S (Session): Includes all participants in the active session.
- M (Moderator): Facilitators and observers.
- T (Participant): Training audience and role-players.

Messaging

These endpoints facilitate the delivery of content into the simulation environment.

POST /messages Sends a general message.

POST /messages/position Sends a message to users based on their specific physical or logical position in the exercise.

POST /messages/role Targets participants assigned to a specific role.

POST /messages/team Sends a message to a specific team ID (retrieved via the /teams endpoint).

Smartwords

GET /smartwords Retrieves the current list of dynamic variables (Smartwords) and their values within the simulation.

POST /smartwords Updates the values of specific Smartwords to dynamically change simulation content or logic.

Exercise Control - Pattern of Life

These endpoints manage automated event stacks, often used for world-building or audience reactions to events

GET /pol Retrieves the current status and index of the Pattern of Life stack.

POST /pol Issues commands to control the stack.

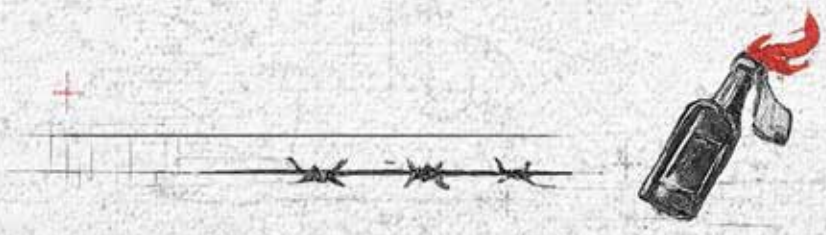
- start: Begins the stack from a specified 4-digit index
- stop: Ceases the currently running stack.
- continue: Resumes the stack from its current position.

Exercise Control - Current Inject

This endpoint informs an application what injects are currently active.

Note that Conducttr uses multiple parallel master events lists.

GET /injects/current Retrieves the active inject for all active MELs



Conducttr