Domain & Multi-Domain Integration with Vantiq

August 2023

Table of Contents

1 Introduction

The rise of IoT sensors and AI that interprets the data they produce, provides an opportunity to build systems that detect, analyze, and then orchestrate, a set of outcomes driven by situational awareness of the battlespace. To be valuable, such systems must operate at scale and with very low latency, because the threat is constantly changing and automated responses degrade in value as latency increases. Furthermore, such systems, if they are managing an activity of value, must be resilient to failure or attack of some component or resource.

A prime example of a high-value use case are Domain or Multi Domain Integration systems which are a concept of great interest within both the military and governmental security services. The concept is to integrate multiple Domains, e.g. multiple siloed systems, data sets and departments, to broaden the scope of threat detection, share information more widely across stakeholders and to react to threats in a timely manner. In practice, this means operating such systems in real time and integrating human decision-making (human-machine collaboration) as required into any automated workflow. **This is a requirement and primary challenge for the implementation of JADC2 (Joint All Domain Command and Control) in the US and MDI CP (Multi-domain Integration Change Program) in the UK.**

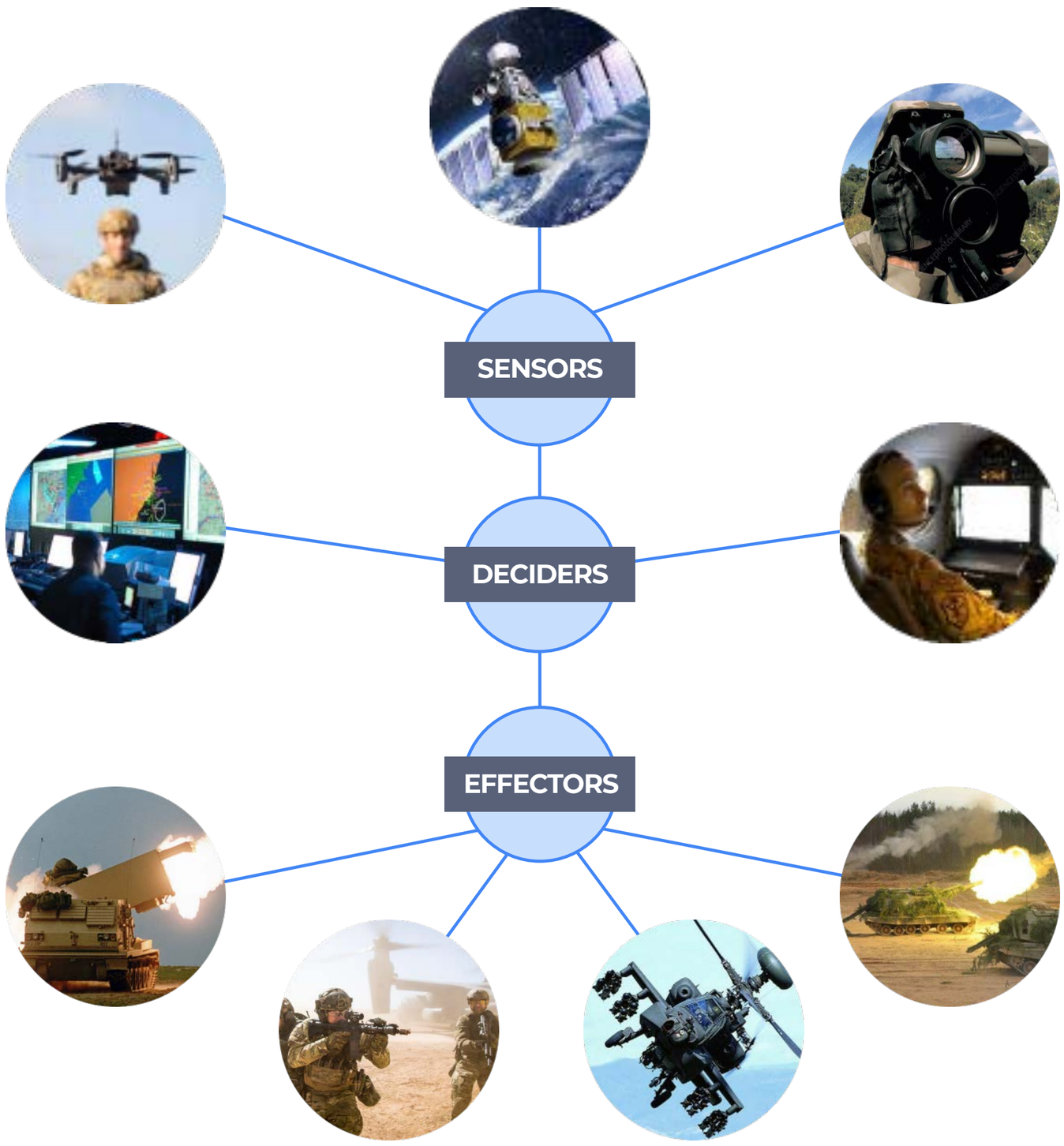
The nature of threats and the environment, especially from a military perspective, is getting more and more complex. No single service or agency or, in some cases, nation might be capable of resolving a threat on their own. This underlines the need for timely sharing of information across multiple stakeholders, hence multi-domain integration. The response to these threats may be different for each agency but they need to be coordinated.

Following this introductory Section 1, this paper is structured in two parts. Section 2 identifies the core challenges faced by software developers tasked with building systems that set out to deliver Domain, or Multi-Domain Integration. Section 3 describes how the Vantiq platform addresses these challenges in a way that makes it practical to build and evolve what, by necessity, are complex distributed and real time systems.
The emphasis of this white paper is on the challenges and available solutions to the software engineering effort of building multi-domain integration systems. Vantiq partners with defence customers and contractors, who add the domain expertise required to implement the specific use cases of such systems.

The emphasis of this white paper is on the challenges and available solutions to the software engineering effort of building multi-domain integration systems. Vantiq partners with defence customers and contractors, who add the domain expertise required to implement the specific use cases of such systems.

1.1 Domain Integration

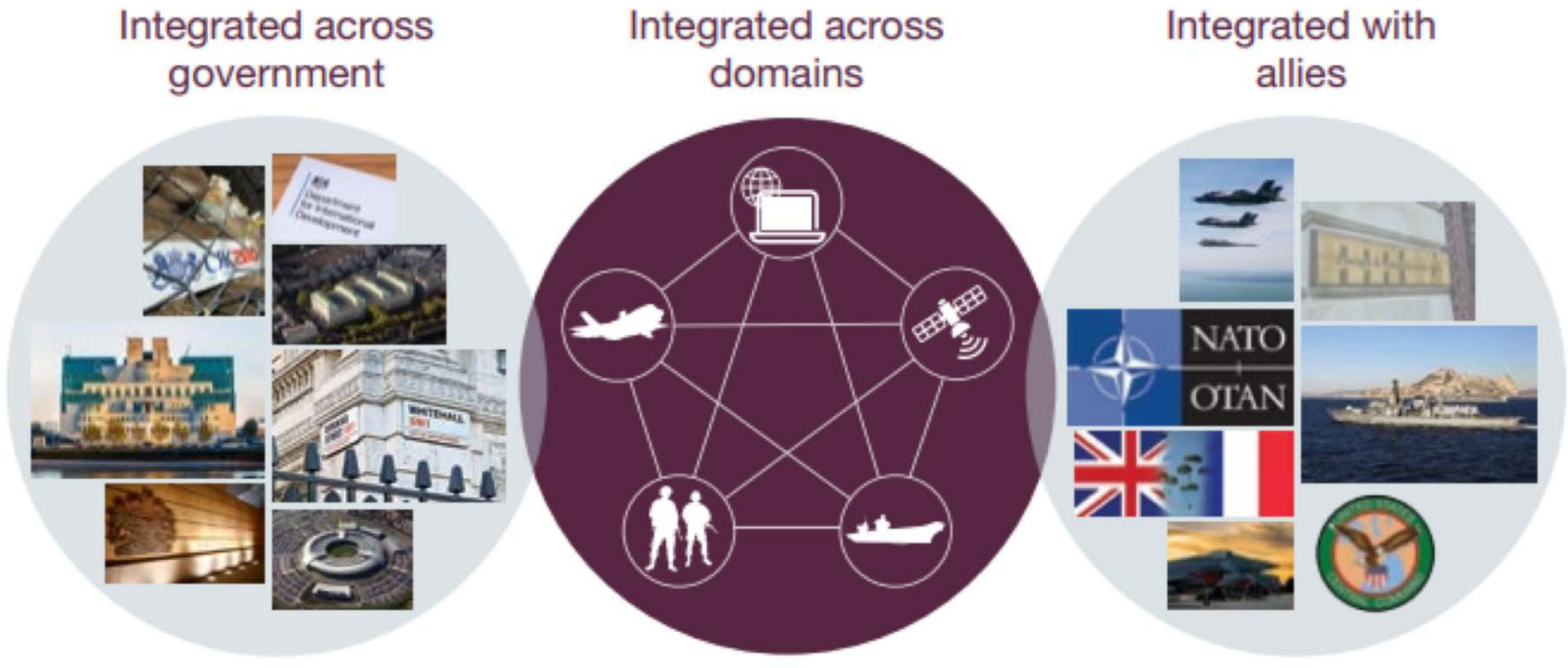
Domain Integration primarily deals with integration across multiple systems and sub-organizations within a specific organization. For instance, integration between land assets, such as deployed infantry, support and command and control vehicles, artillery, drones, forward operating bases, and so on. The same organizational boundaries might also be applied to aircraft, naval assets, space, etc.



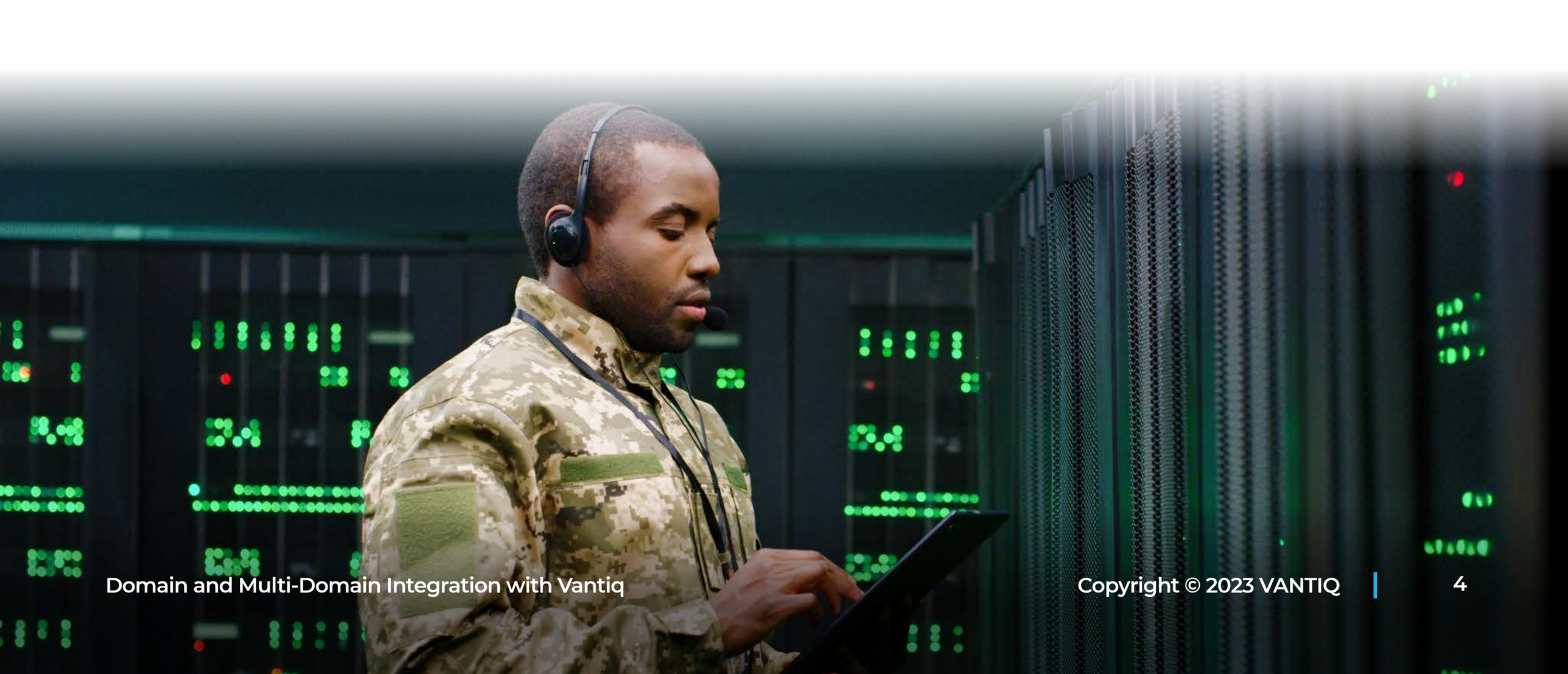
It is important to recognize that existing IT systems are in place that manage some aspects of military operations. More recently, the ever expanding deployment of sensors provides a compelling opportunity to integrate new situational awareness with data managed by existing systems. Existing data assets integrate with real time situational awareness inside new, event-driven, automated and collaborative response processes that are designed to reduce latency between threat detection and an appropriate response (e.g. destroy, avoid, deceive or defeat).

1.2 Multi Domain Integration

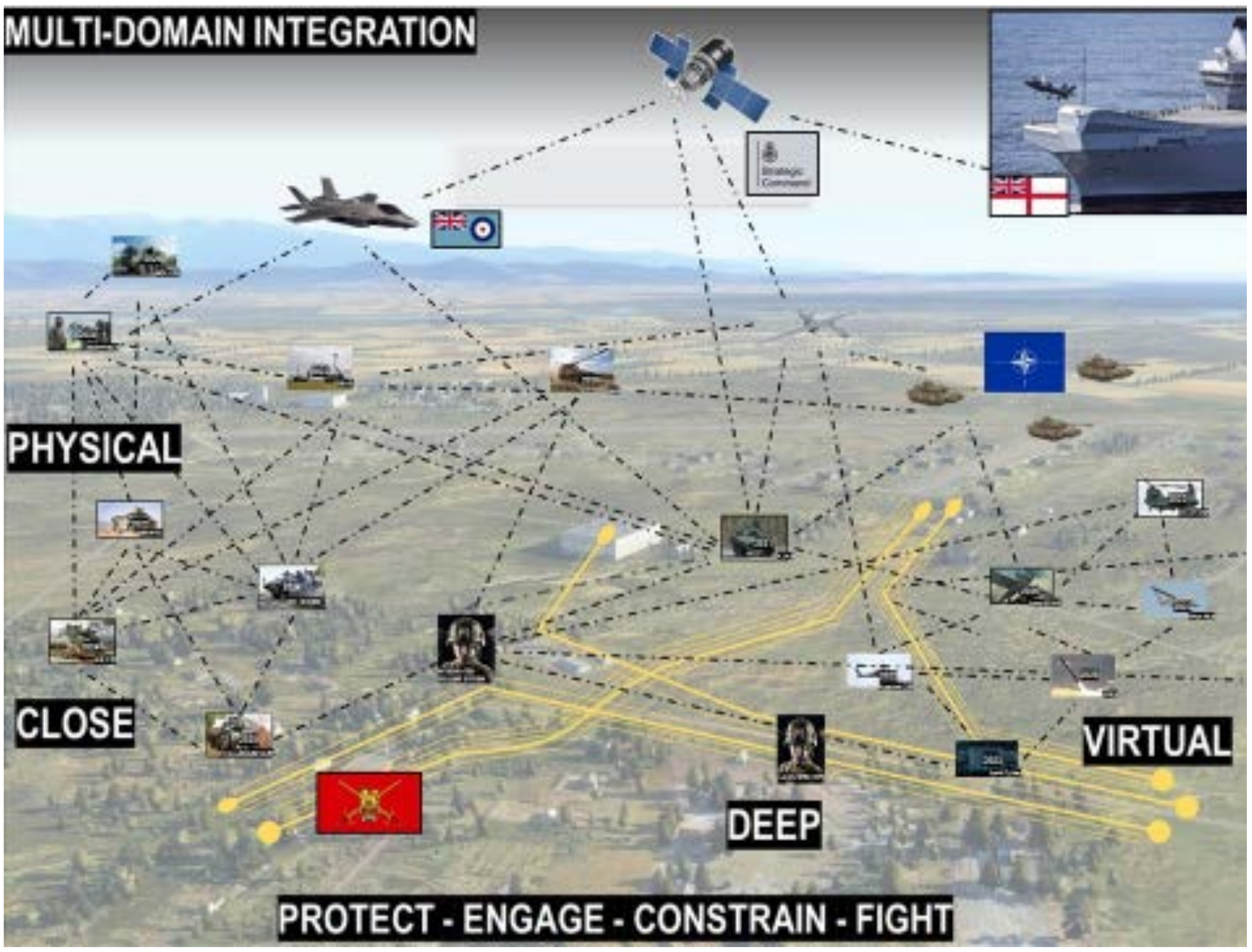
Multi Domain Integration involves integrating multiple domains across both military and civilian organizations, and potentially integrating with partner organizations or nations such as within NATO members, etc.



It is important to recognize that existing IT systems are in place that manage some aspects of military operations. More recently, the ever expanding deployment of sensors provides a compelling opportunity to integrate new situational awareness with data managed by existing systems. Existing data assets integrate with real time situational awareness inside new, event-driven, automated and collaborative response processes that are designed to reduce latency between threat detection and an appropriate response (e.g. destroy, avoid, deceive or defeat).



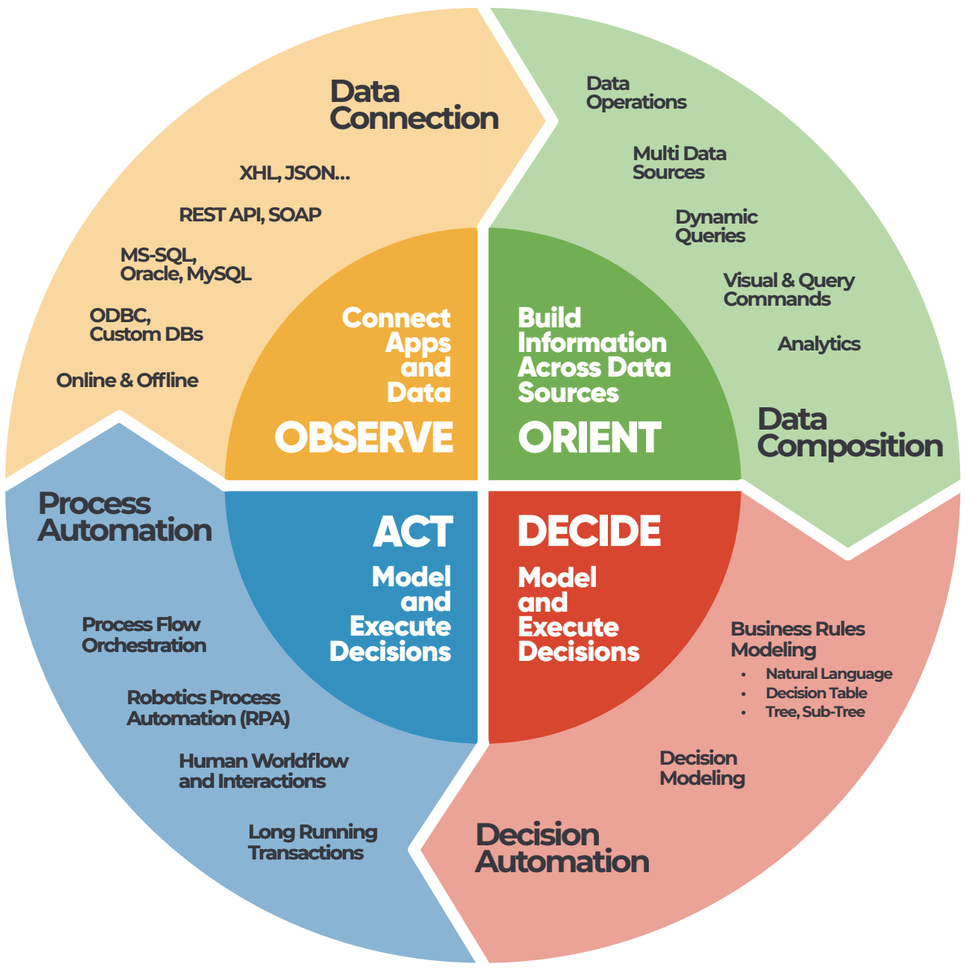
Specifically, an effective combined arms strategy may require collaboration between threat detection and response systems deployed across all branches of the armed forces (e.g., Maritime, Land, Air SF, Logistics, Space and Cyber, etc.). It is likely that this would extend to collaboration with other government departments, civilian organizations and allies within NATO. All these domains need to be integrated in a flexible and intelligent way to accommodate change. Data flowing between the domains needs to be tightly controlled but also real time in nature to allow fast decision making and to keep all parties aware of situations as they develop.



2. Domain and Multi Domain Integration Challenges

2.1 OODA (Orient, Observe, Decide, Act) Loop

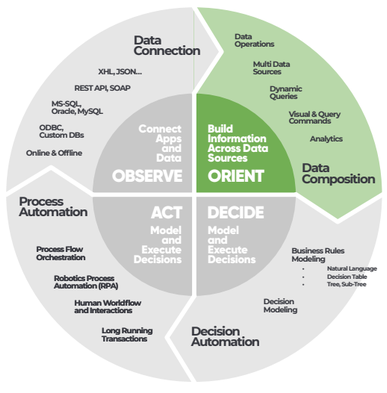
The behavior of systems that implement Domain and Multi Domain integration can be viewed as one or more implementations of an OODA loop. Building systems to automate aspects of the OODA loop presents several challenges in a miliary context:



2.1.1 Observe

The primary function of Observe is to connect with and ingest all data of potential relevance whether that is sourced from sensors or existing systems – both civil and military. Software integration should use encapsulation techniques so that sensors and systems can be replaced over time with minimal or no disruption. Integration must be location independent and could take place at the edge or within the cloud. **Examples include:**

- Sensor data in vehicles, people and devices and platforms
  See > Integration
- Real-time data from existing platforms and systems
- Filtering of irrelevant data
- The amount of potential event data could be very large; as such, processing data as close to the source as possible is critical
  See > Edge Processing
- Mobile or remote sources of data may not be connected 100% of the time so edge processing and autonomous operations need to be considered
  See > Reliability and Resilience Orient



2.1.2 Orient

Discrete events, as detected by individual sensors, may not convey a complete picture of a situation. Information enrichment occurs when logic contained within the system applies correlation and adds context across the entire information set captured within a given processing node, or as shared across multiple processing nodes.

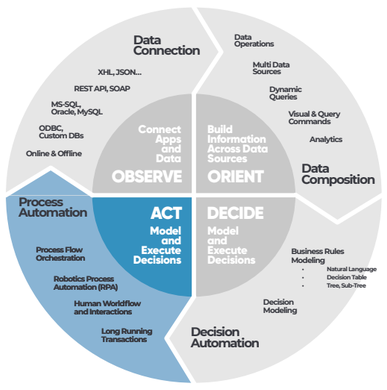
Examples of such operations are:

- Correlating the multiple sources of events and processing these to build context
- Augmenting event data with spatial information such as GPS location
- Correlating data based on time. For example, monitoring deltas in sensor data within a finite time period (application of temporal operators) to perform trend analysis and to build context
- Augmenting situational information with static data sources (e.g. existing systems or databases)
  See > Federated Data
- Events of interest can be subscribed to and received in real-time by other processing nodes who may further enrich the information. For example, mobile infantry may process and enrich sensor data at an edge node deployed, say, in a support vehicle. Events of interest may then be sent from the edge node to cloud servers for correlation with other mobile units

2.1.3 Decide

Given the explosion in data volumes now captured, it is highly desirable to integrate machine learning algorithms to automate the detection of patterns of interest. This improves the signal-to-noise ratio to reduce the workload on humans who may be called upon by the system to review any situations of interest. Or, should a compelling situation of interest be detected directly without AI, the system can be instructed to directly alert a human to make a decision. **Potential outcomes may include:**

- Automating a response up to the point where humans need to review the information and decide upon a course of action
  See > Real-Time Decision Making
- Other responses (whether automated or as an output from human intervention) may include deferring, sharing, or escalating decisions to other components within the system



2.1.4 Act

As this stage in the system, a number of automated responses can be triggered that anticipate the appropriate course of action in response to a decision. **This may include:**

- Integration with other command and control systems to act upon, or escalate, decisions to other actors or domains
- Coordinate actions between domains, which may involve higher-level workflows
- Under some circumstances (e.g. self-defence), an autonomous response may be authorised

2.2 Integration

Traditional integration approaches such as batch and ETL (Extract, Transform and Load) are useful for synchronizing siloed systems periodically. In a fast-moving and rapidly changing environment such as a battlespace, sharing data hourly/daily, etc. undermines the ability for people to make real-time decisions when the available information is out of date.

Furthermore, network bandwidth will be limited and unreliable so attempts to move large batches of information between nodes further complicates the problem. If a drone spots an interesting event, but the system requires MB or GB of data to be transferred between nodes for analysis before a decision can be made, this approach will significantly reduce the effectiveness of the information. And if data is lost or delayed due to network failures, this further impacts the issue.

Event-based integration makes use of streaming data to push relevant information to the right systems and people. Integration is taking place in real-time and flow rates and reliability is much easier to control and is more resilient to interruptions.

2.3 Edge Processing

Much of the data generated in a battlespace scenario is remote – sensors on people, vehicles, drones, aircraft, ships, etc. The amount of data these sensors generate is significant and only increasing. Network connectivity is often unreliable and may have low bandwidth, and the risk of the enemy blocking or attacking network infrastructure is material. Moving huge volumes of sensor data to a centralized location to be processed is simply not practical. This information needs to be analyzed close to the source of data, using edge nodes, to detect a situation of interest in real time and only situational data needs to be pushed in real time to other systems.

Edge processing also allows for a level of autonomy in the situation where network connectivity is limited or disabled. From time to time, remote units may need to be able to operate autonomously and make local decisions when disconnected from centralized systems. Any events of interest can be cached and forwarded when networks are restored (see Section 2.5.2) to other edge nodes or system components running in the cloud.

2.4 Federated Data

Integration of system and sensors typically means moving data between systems. Many systems and platforms will support specialized data formats such as video streams, audio streams, and satellite imagery.

Moving this data has several challenges. It is often very large, and not all systems will have the ability to store and ingest every data format, and updating systems to support all data formats is not practical. Also, making fast decisions may not always require all the actual data but perhaps requires only a small subset. Leaving specialist data in place and moving metadata across distributed nodes is often more efficient. Users can then access source data only when it is required.

This approach can be made more complex by the unreliability of networks, so a hybrid approach can be implemented. Rather than move a complete video stream, the associated metadata could include a small number of video frames that contains the most relevant information, thereby allowing a human to have at least some information if a drone for instance, is out of range or has been disabled.

2.5 Reliability and Resilience

There are several factors to consider when thinking about reliability and resilience:

2.5.1 Network Reliability

Networks within a battlespace cannot be relied on to be continuously available. Remote equipment may lose connectivity due to several factors, such as beyond range, poor reception, weather conditions, adversary action, etc. Even centralized and static forward operating bases may lose connectivity due to equipment failure and cyber-attacks. Therefore, having a level of autonomy within the distributed nodes such that individual nodes (edge) or clusters of nodes can still operate in an autonomous or degraded state is critical.

2.5.2 Store and Message Forwarding

When a node or cluster of nodes have lost connectivity, situational awareness data should not be lost. Therefore, messaging between nodes in the environment should make use of store and forward capabilities such that this information will reach it intended destination when connectivity is restored.

2.5.3 High Availability

Having the ability to cluster (create reliable environments), either at a centralized or forward operating base, or within any environment that has sufficient compute resources, is critical. This provides high levels of reliability and availability within designated locations. High availability within edge nodes is generally not practical therefore providing the ability to migrate state or synchronize state across edge nodes will provide a higher level of resilience to failure.

2.5.4 No Single Point of Failure

Where possible, single points of failure within the environment should be avoided. This covers messaging (event brokers) and security infrastructure. Any point within the environment that must be accessible for all nodes within a system creates a fragile environment. Security should use a Federated model rather than a centralized model – **See Security below**. Event Brokers and messaging should be decentralized so as to not introduce any single points of failure.

2.6 Security

Although centralized security models are easier to manage, they present a challenge when dealing with highly distributed and potentially remote and autonomous environments. They also offer a single point of entry for any potential cyber-attacks. Once a centralized security system is breached potentially the environment as a whole is vulnerable.

Federated security models offer several advantages:

No Single Point of Failure

- Autonomous and remote nodes within the environment can still operate without any central connectivity

No single attack surface

- If security is breached or if a single node within the environment is under a cyber-attack, this does not affect other nodes within the environment

2.7 Real-Time Decision Making

The battle space is a very dynamic and fast-changing environment. Real-time decision-making is critical to be able to shorten the time between information acquisition and decision-making. The amount of information gathered is also vast and highly distributed. The ability to ingest real-time information and detect and filter for situations of interest is critically important.

For example, sensor information may need to be correlated with other sensors or device data to further enhance and enable the detection of situations. Furthermore, the correlation of multiple sensors or devices needs to be temporal in nature as the relevance of these events can be very time sensitive. Two events that are an hour apart may not be relevant, but two events that happen within 5 seconds of each other may be highly relevant.

Once situations of interest are detected then, situational information needs to be automatically communicated to relevant decision-makers. The situations of interest need to be distributed in real-time and in a reliable fashion so that decisions can either be made, deferred or escalated in a timely fashion.

Decision makers (OODA Decide) may cross multi-tiered organisations, meaning certain decisions could be made locally within a small unit or they may need to be communicated to a more centralized command-and-control system (or both).

Situations of interest could also be geospatially important; information may need to be pushed to closely located assets of organizations based on the location of the situation or the location of units/resources on the ground.

Decision-making will often involve humans. Some decisions could be autonomous but, in the majority of the cases, humans need to be in the decision-making loop and, once again, this could be multi-tiered and involve escalating decisions to more senior operators or other organizations.

2.8 Agile Development and Deployment

The battle space is a very dynamic and fast-changing environment. Real-time decision-making is critical to be able to shorten the time between information acquisition and decision-making. The amount of information gathered is also vast and highly distributed. The ability to ingest real-time information and detect and filter for situations of interest is critically important.

For example, sensor information may need to be correlated with other sensors or device data to further enhance and enable the detection of situations. Furthermore, the correlation of multiple sensors or devices needs to be temporal in nature as the relevance of these events can be very time sensitive. Two events that are an hour apart may not be relevant, but two events that happen within 5 seconds of each other may be highly relevant.

Once situations of interest are detected then, situational information needs to be automatically communicated to relevant decision-makers. The situations of interest need to be distributed in real-time and in a reliable fashion so that decisions can either be made, deferred or escalated in a timely fashion.

Decision makers (OODA Decide) may cross multi-tiered organisations, meaning certain decisions could be made locally within a small unit or they may need to be communicated to a more centralized command-and-control system (or both).

Furthermore, systems will be subject to constant external change. New sensors and new AI algorithms may come online continuously. New threats are emerging all the time.

The key to success is two-fold:

- Use a systems architecture that is inherently flexible to accommodate change
  See > Section 3.2 below
- Use a combination of low code development tools (Section 3.1.1), automated dev/ops deployment tools (Section 3.1.2) and an integrated and hardened runtime system (Section 3.4, 3.5, and 3.6)

In Summary

> **The key to keeping functionality and deliverables in sync requires the use of agile methods, a supporting toolset and a flexible deployment architecture. This is particularly important when in technological competition with an adversary – whoever adapts the fastest wins.**

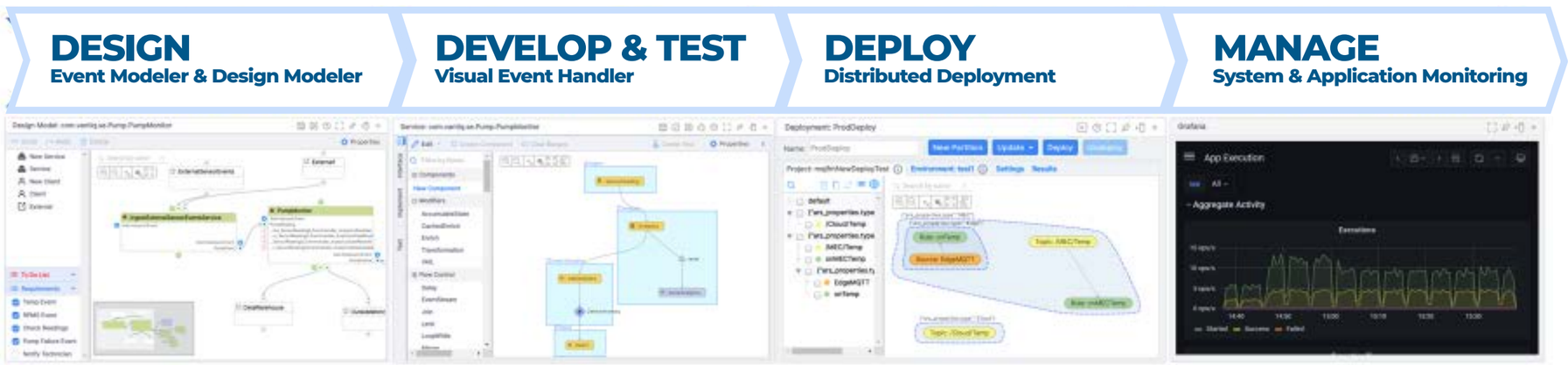
3. Vantiq and (Multi) Domain Integration

3.1 Introduction to Vantiq

Vantiq is a COTS (Complete, Off The Shelf) product that integrates: (a) low-code application development tools; (b) an automated dev/ops deployment system; and (c) a hardened run time system. Vantiq makes it possible to implement distributed, event-driven, real-time systems using agile methods that would otherwise require low level programming.

3.1.1 Low-Code Development Tools

Vantiq's development tools use a low-code approach with a drag and drop interface that allows developers to describe event flows and implement code that underpins the analytics and situational awareness required. Although the platform takes a low code approach, developers can also write more complex code using a high-level scripting language.



Vantiq's development model allows developers to very quickly create new or enhance existing applications. Vantiq's Event Catalog allows new sensor data and systems to be quickly integrated into existing applications, or to create new situational analysis using existing data when detection of new threats or scenarios is required.

3.1.2 Automated Dev/Ops Deployment Tools

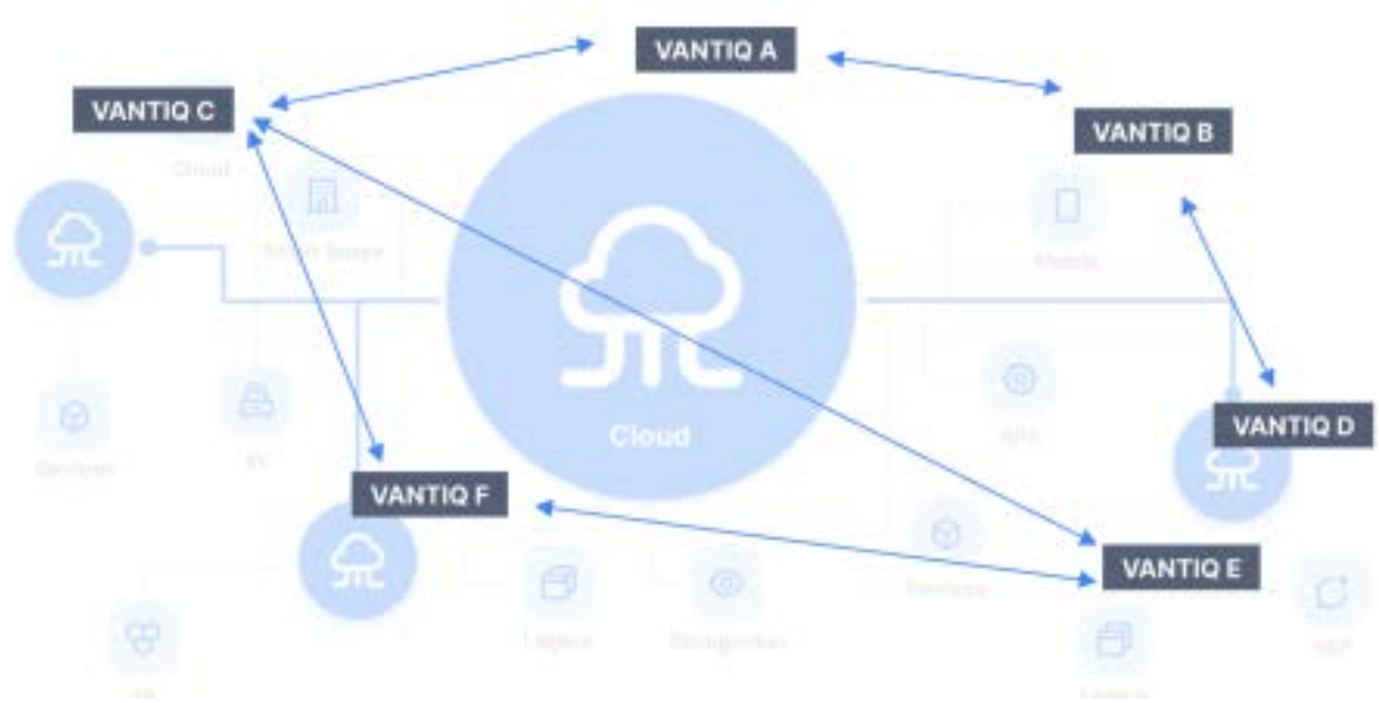
Once new or enhanced applications have been developed, it is important to be able to rapidly deploy, test and demonstrate these enhancements in a timely manner across a distributed system. Vantiq integrates visual distributed deployment tools that allow developers and operations to deploy applications into a distributed environment quickly and easily. Reliable deployments means that if Vantiq servers are offline or uncontactable at the point of deployment they will be updated when they are online again.

The tool enables deployment into multiple environments so that a single logical application can be configured to support different physical topologies as well as deployment to various test environments without making changes to the underlying application logic.

Vantiq applications are created as components

1. Vantiq intelligently installs components

2. Vantiq installs Mesh Network

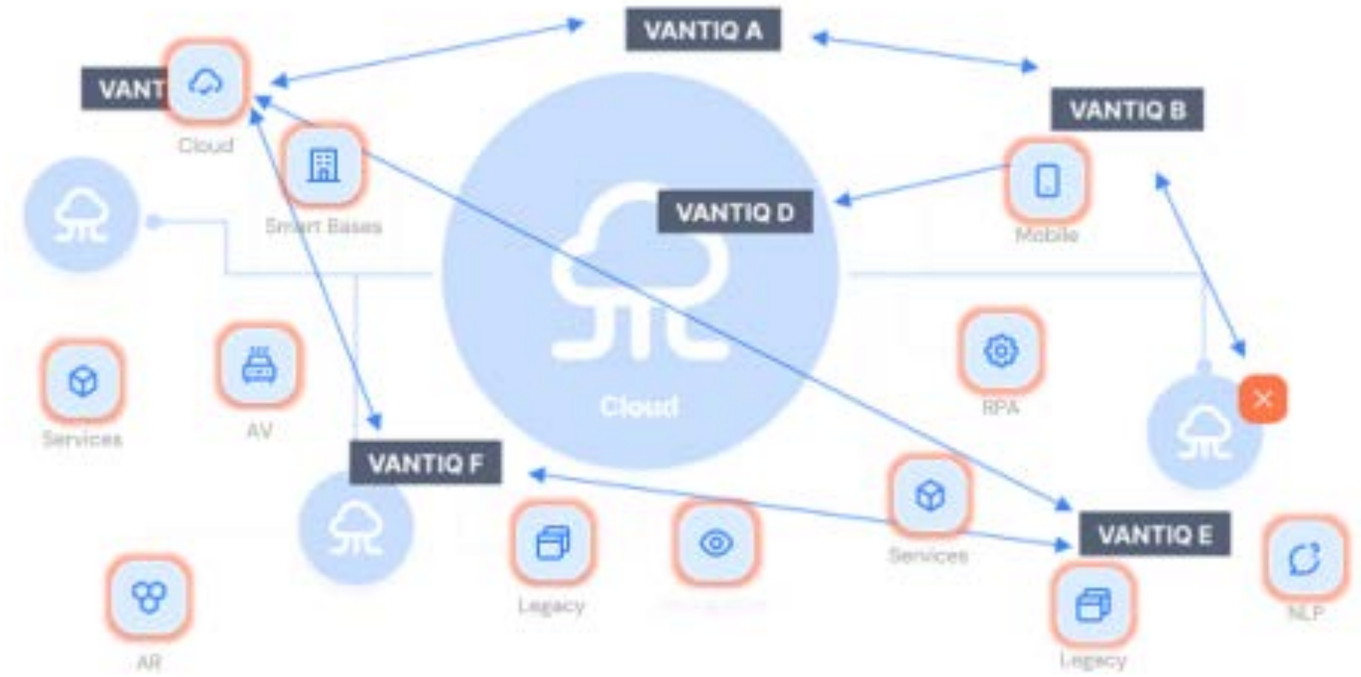


3.1.3 Distributed Runtime System

Vantiq development and deployment tools integrate with Vantiq's distributed runtime system that operates deployed Vantiq applications. The runtime system incorporates many unique capabilities that are specifically designed for operating mission-critical, distributed applications. The remainder of Section 3 examines the main capabilities of the Vantiq runtime system in the context of the requirements outlined in Section 2

Vantiq Unites everything legacy, present, future.

Vantiq applications connect with the things you already have, and integrates with the new things you'll add



3.2 System Flexibility via Event-Based Integration

Unlike traditional IT systems, which are driven using a request/response model and which tend to be database-centric, Vantiq applications are event-driven and, whilst events of interest may be persisted, there is no assumption of the use of a database for sharing information. Vantiq applications are event-driven, and they operate in memory in real-time. Sharing of information is implemented using a distributed event broker and a publish/subscribe model.

Vantiq applications are federated and inherently distributed. They ingest real-time events (state changes), will filter and enrich such data (often at edge nodes) and push events of interest to other systems in real-time, thereby eliminating the latency inherent in database-centric systems. Events may be routed to multiple systems and domains using a loosely coupled publisher/subscriber model that makes the system more flexible. New subscribers can be added without any requirement to make changes to the publisher of events.

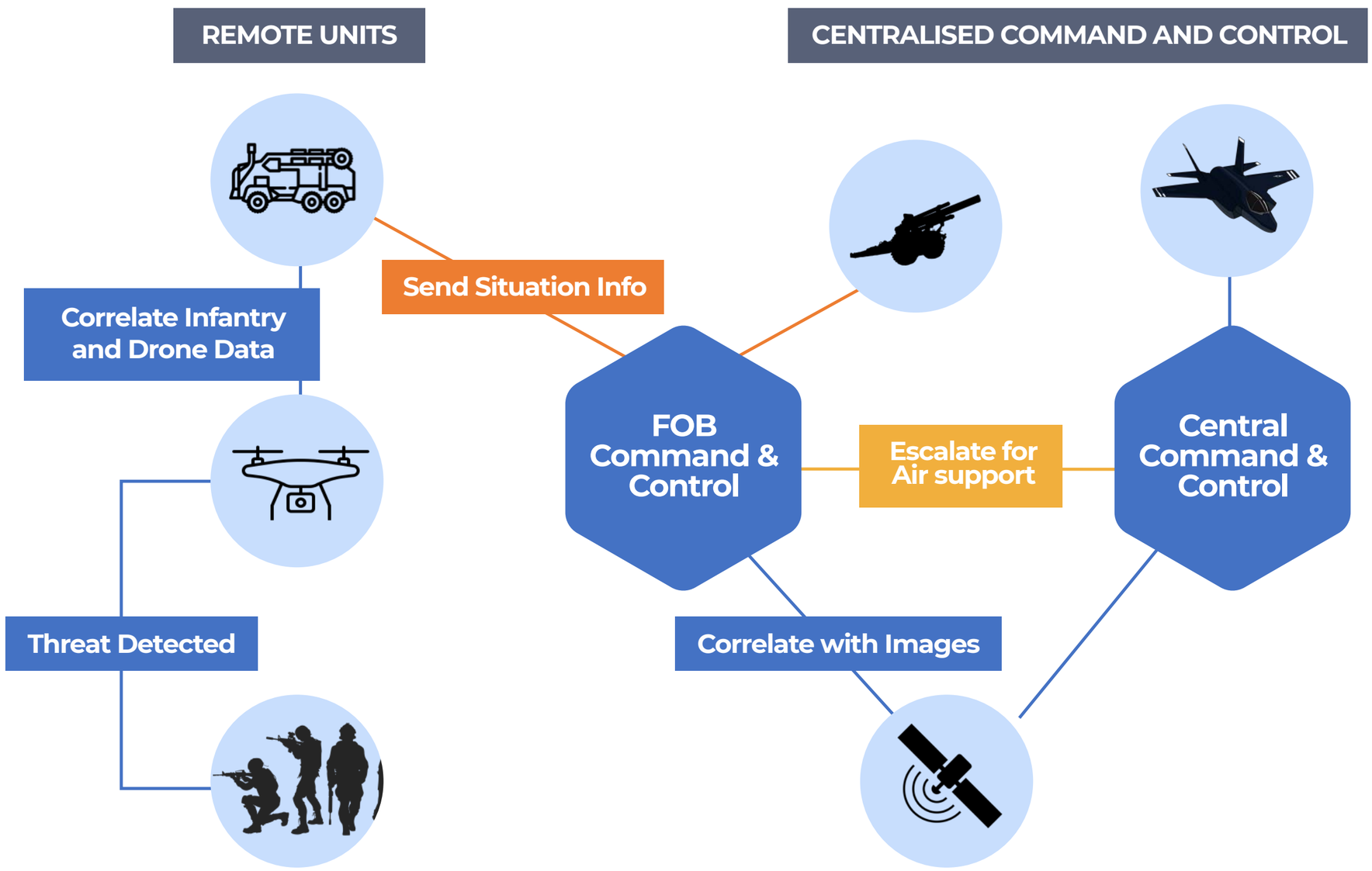
multiple Vantiq event handlers could subscribe to the same event and implement different processing requirements.

Loose coupling within event-based integration is also very important because tightly binding various services together results in a very inflexible and complex system to maintain and upgrade. Service-based integration tends to be tightly bound or it uses API gateways to reduce the tight binding. Events are by nature, loosely coupled, making changes and upgrades to a system much easier to achieve. With Vantiq, each event handler ingests and generates events, and the event handlers can be bound together at run time without changing the interface or implementation.

A further issue with API-based integration is that it inherently implies processing will take place on a message through the service's operations. This leads to either frequent changes to services or a proliferation of operations as the system evolves. With Vantiq, events do not imply any processing intent as they represent state changes. This means that multiple Vantiq event handlers could subscribe to the same event and implement different processing requirements. Also, events can easily evolve with additional properties without changes to the subscribing event handlers.

The systems that need to be integrated are not all centrally located so it is also important to have a decentralized integration model.

In the diagram above, events are flowing between the different systems (infantry, drones, etc.) and are being correlated locally. Newly enriched event data may then be published and forwarded to subscribers and, as the situation changes, events can flow from one command and control system to another. Events of interest may be persisted but there is no assumption or requirement to use a database as a point of synchronization, as these systems add unwanted latency. Rather, all these events are flowing in real time rather than in batch or bulk synchronization. Each discrete node will be correlating and enhancing the situational information with additional data, allowing local and remote operators to make informed decisions. Each system can subscribe to any events of interest and may implement different logic to perform further analysis.

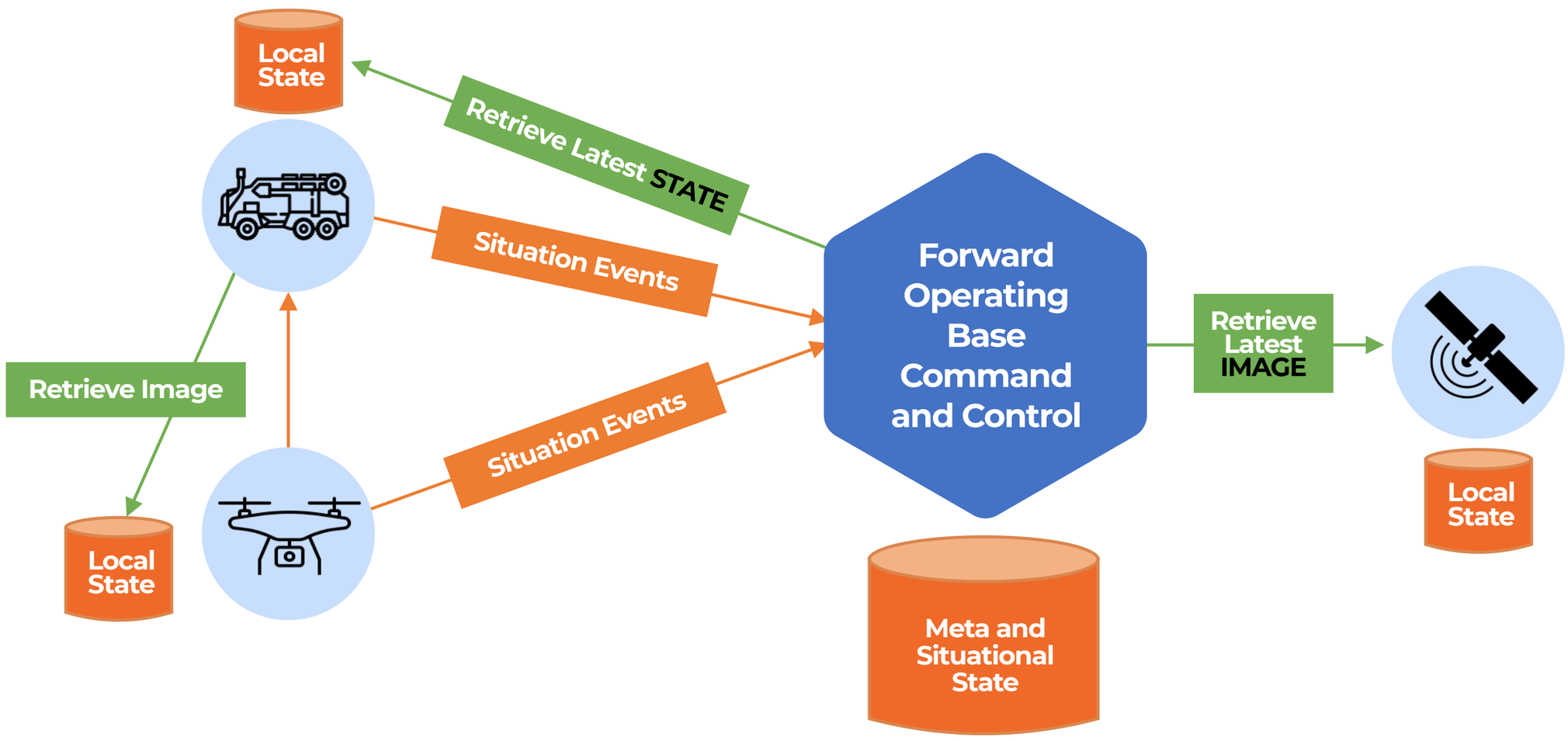


Also, using an event-based model allows far easier maintenance of the system over time. Subscribers may be easily added and removed, and new or enhanced applications may be dynamically added to the system. The publishers of events are not affected by any new subscribers being added. Each subscriber can make different usage of the events and perform different analysis without affecting the publishers. Subscriber applications could be subscribing to multiple events from different domains.

3.3 Federated Data Model

Much of the information that these systems generate could be very large and it may not be convenient or practical to migrate such data between different systems, as would be the case in a data lake or big data architecture. Also, different systems may not be able to support certain data formats. Analysis of data close to its source, detecting situations of interest and only moving the situational data around reduces latency and network load.
Centralizing data and centralizing the processing of situations of interest increases latency and reduces reliability when there are failures or attacks on the network. For example, drones could be generating video feeds but it is likely that not all systems interested in this content will be available to process or store these video feeds.

Using Vantiq's federated data model, just the metadata from video feeds or satellite images may be moved around and Vantiq keeps track of where the original source information is located. In this model, it is not necessary to actually move data until a threat is detected or relevant information needs to be shared, for example, with a human being, so that they can make decisions.



In the example above, the locally analyzed situational events are moving between the drone, vehicles and a forward operating base. These situational events contain information such as the analytics that have been performed against the video or specific images, rather than entire moves of bulky images or video streams. Where network connectivity permits, an operator in the forward operating base can retrieve selected frames or segments of the raw data to further enhance their decision-making process.

Vantiq's ability to perform distributed queries and execution of distributed operations makes this approach very easy to implement. The nature of Vantiq's distributed processing model also means that physical properties like IP addresses are not involved in the execution of distributed queries. This means that applications are very easy to configure, with distributed queries and operations implemented at a higher level of abstraction using meta data rather than physical networking information like IP addresses.

3.4 Distributed and Decentralized Event Broker

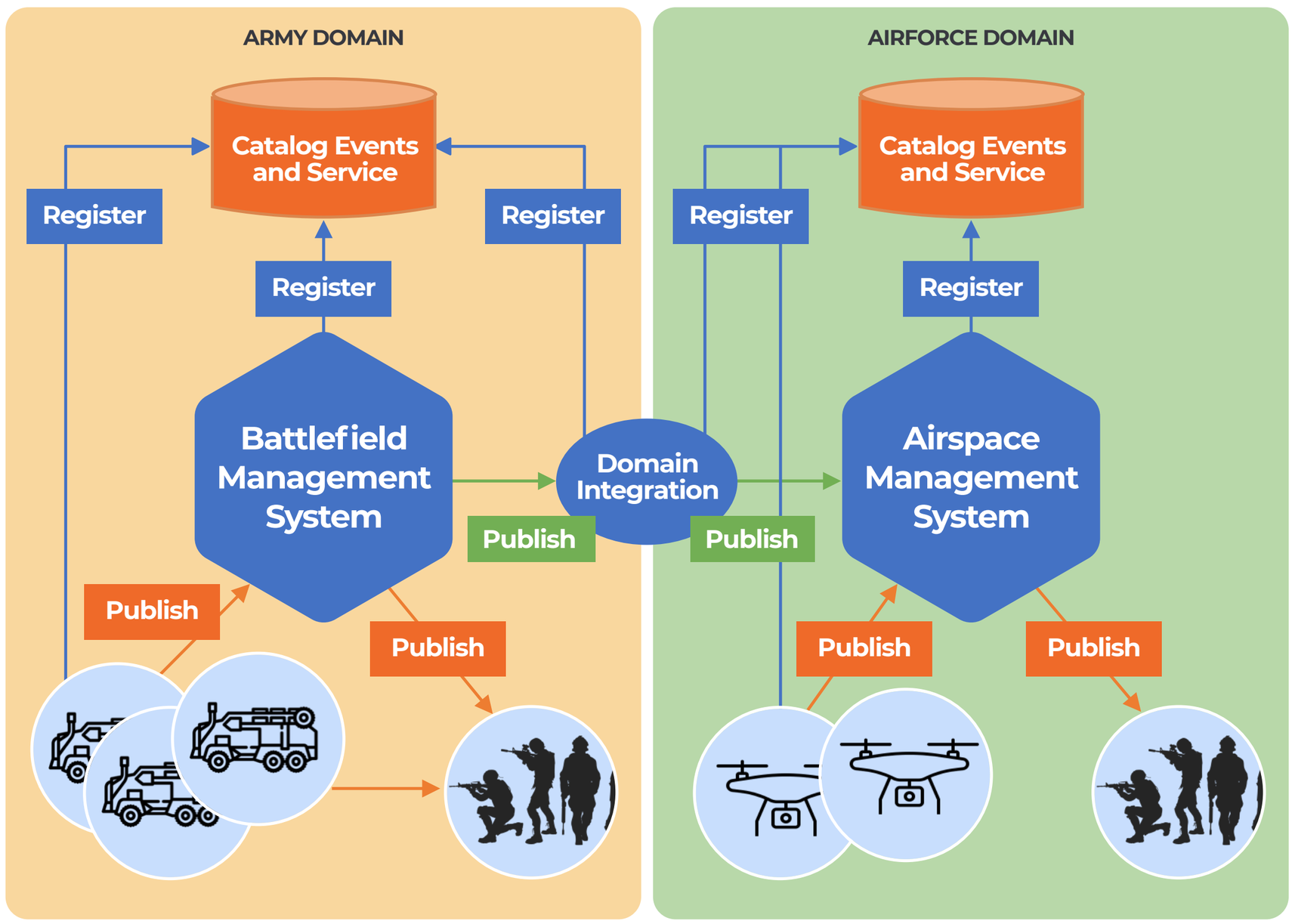
Most event brokers operate with a hub and spoke architecture, meaning that all events and messages flow through a centralized service. Whilst Vantiq implements a centralized catalog of domain events and services, it is only used at development time to improve development productivity.

In contrast to traditional event brokers, at run time, the Vantiq Event Broker uses a distributed and decentralized approach to communicate between publishers and subscribers who communicate directly with each other.

This architectural approach has several benefits:

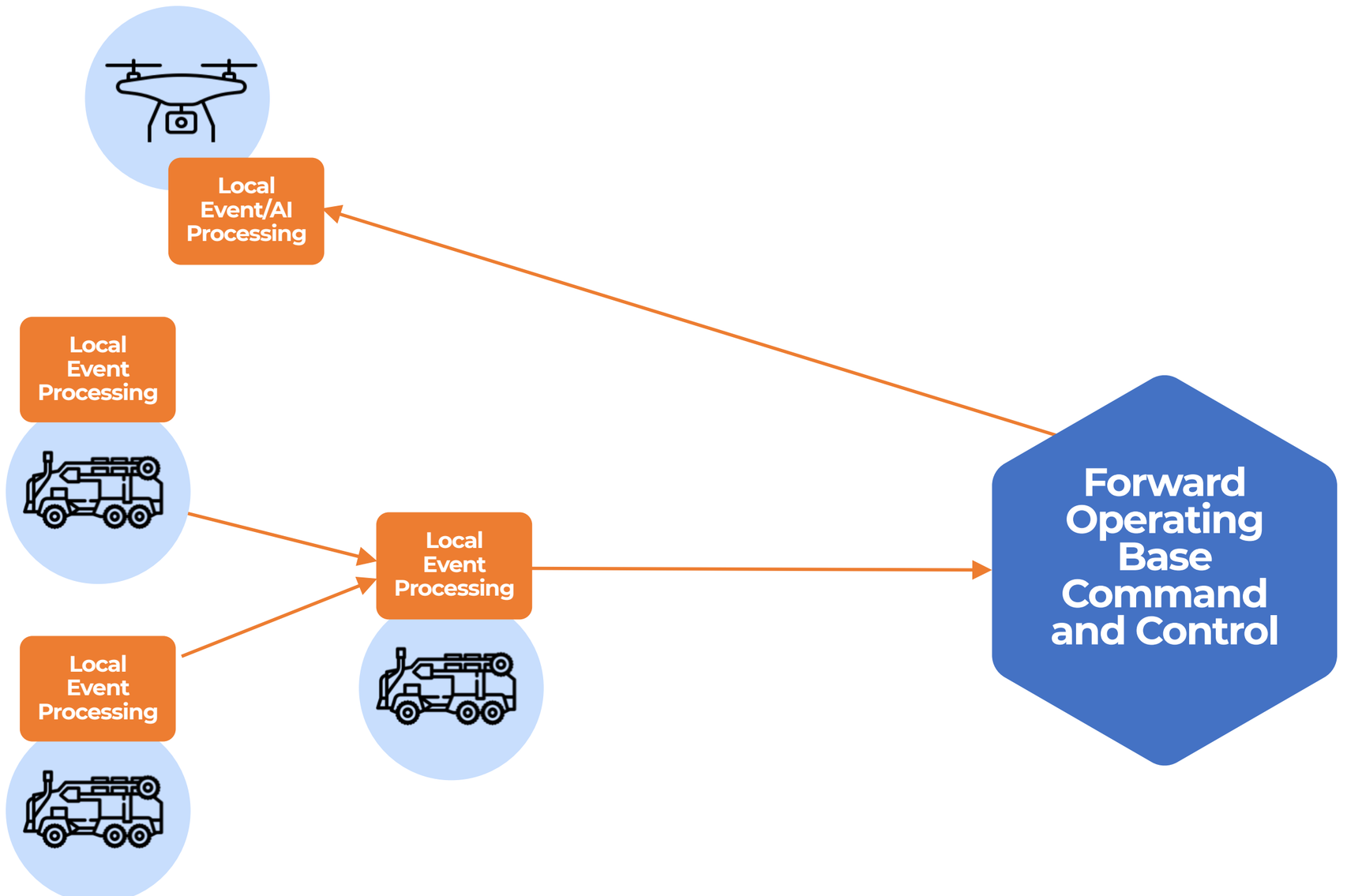
- No single point of failure
- Easier to scale
- Ability to route events through multiple intermediaries.

In a multi domain integration environment, it is very easy to combine multiple event brokers, one for each domain, and manage the security and filtering of the events that are passed between Event Brokers. Such combinations can embrace multiple instances of Vantiq's event broker, or Vantiq's event broker can integrate with existing 3rd party brokers.



3.5 Distributed Applications and Edge Processing

Domain and multi-domain integration is by nature a distributed environment and will require deployment of multiple edge processing nodes. Such systems will combine both centralized and regional systems and, in some situations, temporary command and control systems at the edge. Sensors themselves are highly distributed and demand edge processing to filter, aggregate and perform local situational awareness. Modern aircraft (e.g. F35) and vehicles will gather a great deal of onboard sensor data that cannot be streamed centrally due to network bandwidth and availability constraints, so will require local processing and a reliable edge-to-cloud delivery mechanism.



In the above example, each vehicle has some onboard local processing capability to aggregate, filter and analyze the local event data. Rather than streaming all this data to a cloud or other centralized system, a Vantiq application ensures that only events of interest are published to the central systems. Assets on the battlefield can also communicate information amongst themselves. For instance, each vehicle could be periodically communicating fuel and ammunition levels in real time to a command vehicle, giving a local commander sufficient information to make go-no-go decisions. Important situational awareness could also be published to a centralized system so that commanders at a base level are aware of the situation and could make decisions regarding logistical support such as organizing refueling, etc.

Vantiq applications are able to implement autonomous processing on edge nodes, such as filtering and aggregation of information. Autonomy of operation is very important when network availability is limited, and bandwidth constrained. Vantiq's event broker implements reliable messaging (buffering/queueing of critical events on edge nodes), which becomes critically important when networks are unreliable. Such information will be forwarded automatically to commanders once network issues are resolved.

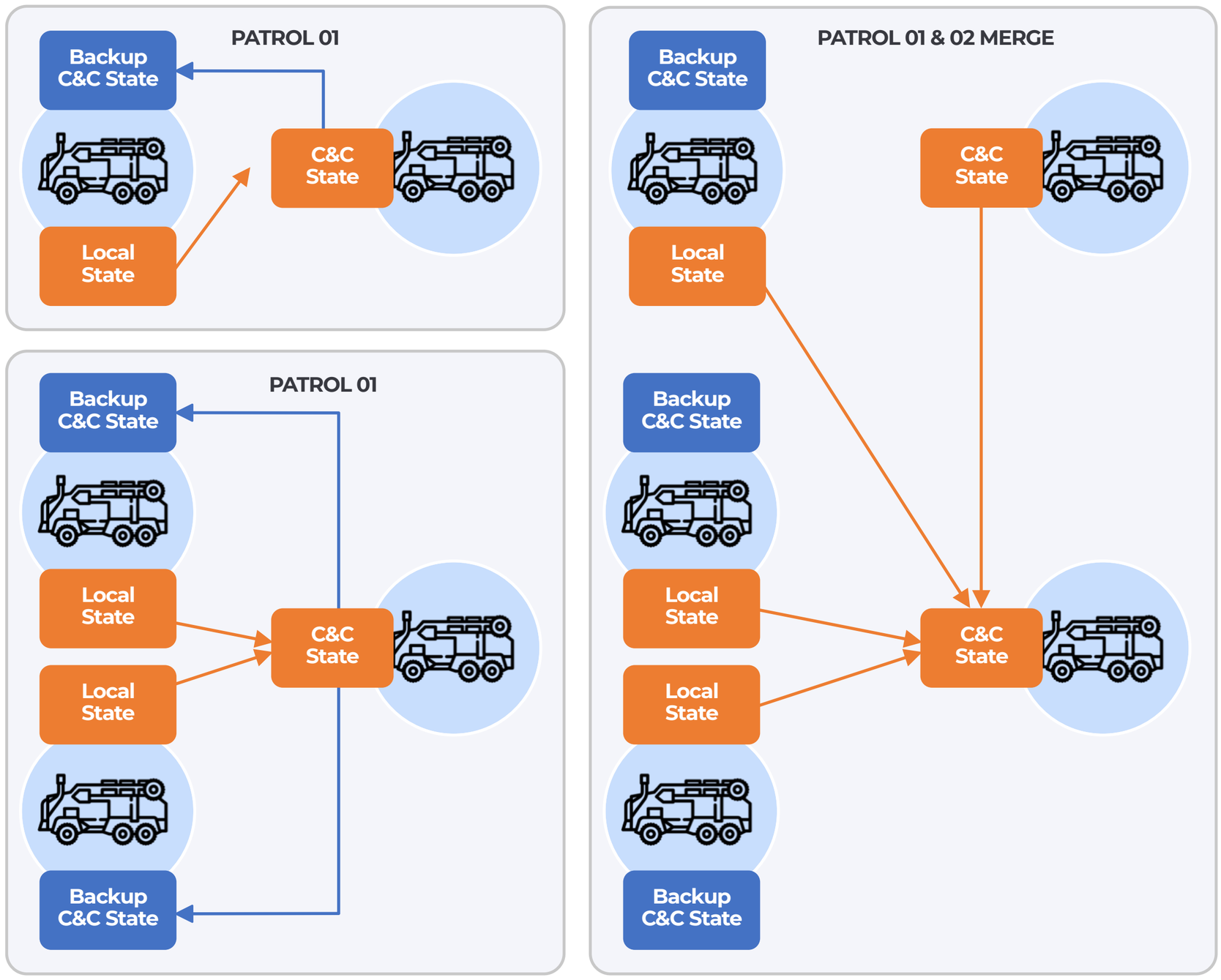
3.6 Hierarchical and Mesh Communications Model

Distributed and federated systems need the ability to both communicate in a fixed hierarchical model but to also be able to communicate across peers. It is precisely because the organization and movement of military assets is highly dynamic, that it is critical to enable relevant information to be communicated within dynamic small units rather than constrained only to move between static, hierarchical tiers.

In this context, Vantiq's ability to support both hierarchical and mesh-based topologies is important. The ability to adjust the mesh communications dynamically is also an important differentiator. Patrol vehicles leaving from a forward operating base are not static and will change their make up on a per mission basis so being able to adjust the mesh network at deployment time allows for easier and more dynamic configurations.

3.7 State Migration

Distributed, real time applications maintain state in memory and need to manage where state is maintained at run time. For example, state may need to be replicated for the purpose of handling damaged or out of action vehicles and/or multiple groups of vehicles coming together may wish to merge information in an ad-hoc manner. Vantiq's ability to distribute state changes across edge instances and have applications dynamically adjust their behavior based on changes to the environment is critically important in this environment.



In the above example two independent patrols have joined forces dynamically. Each patrol contains a command-and-control vehicle. When the two patrols join, one of the command-and-control vehicles is elected to be the merged patrol command-and-control vehicle and the state that was being maintained in each command-and-control vehicle is now merged.

3.8 Human Machine Collaboration

The vast volume of information becoming available in command and control nodes is causing information saturation, and risks command paralysis or miscalculation. Vantiq's ability to identify and link specific or multiple events together offers a solution to this challenge.

When threats are detected, it is critically important for humans to be in, or on, the decision-making loop. AI and ML may be very useful in terms of filtering and supporting this decision-making process, but in many situations, it is necessary that humans be the ultimate decider of any action. Therefore, they must be provided with up-to-date and relevant information to support a decision-making process. Collaboration needs to embrace both people directly

and support the use of existing systems. Different organizations/agencies will have their own systems to communicate with their staff and no single unified model is possible to achieve in practice. Therefore, the collaborative aspects of Domain integration must support multiple channels and not just be tied to a single mechanism.

humans must be provided with up-to-date and relevant information to support a decision-making process.

Vantiq's Human Machine Collaboration capabilities are developed using a low code drag-and-drop graphical editor that allows you to describe the collaborative aspects of an Ω very quickly.

4. Conclusion

Domain and Multi-Domain integration is a very complex and dynamic environment. It requires support from Edge, Distributed, and Federated capabilities. It also is a very dynamic and complex environment to develop and deploy. The capabilities of Vantiq and its low code and agile development and deployment capabilities perfectly match the general requirements of Domain and Multi-Domain integration. By offering agile development tools, and reducing the number of different technologies required, Vantiq dramatically reduces the time to market, cost, and associated risk of developing these complex systems.

The bottom line for the MoD/DoD and its partners is this: compared to traditional development approaches, Vantiq delivers unprecedented time-to-value and return on investment. Its support of an agile, dev/ops approach means new requirements can be incorporated into the application build without issue. And its robust run time system means that application components are immediately ready for deployment, without the usual "prototype then re-build and harden" wastage.

Vantiq is working with a number of UK MoD, NATO and DoD entities, and multi-national defense contractors, to demonstrate the benefits of the platform in the context of Multi-Domain Integration. We would love to meet with any readers of this paper to discuss our capability further.

About Vantiq

Vantiq is the leading low-code platform for building and deploying real-time distributed solutions. Built on a next-generation event-driven architecture, Vantiq enables highly scalable and low-latency analysis of real-time streaming data from IoT devices, cameras, and enterprise systems to drive situational awareness for safety, security, and operational efficiency. Vantiq was founded in 2015 by software veterans Marty Sprinzen and Paul Butterworth, co-founders of Forte Software.

For more information, please visit Vantiq at www.vantiq.com or email us at info@vantiq.com Twitter, and LinkedIn.

VANTIQ