White Paper

Internet of Things Technology
Intelligence from Edge to Cloud

Situational Awareness at the Dynamic Edge

Abstract

Situational Awareness at the Dynamic Edge (SADE) is a reference software infrastructure that enables the deployment and management of modular, scalable, and resilient edge solutions for defense, aerospace, and emergency personnel. The infrastructure is powered by Intel® technologies and maximizes the use of commercial off-the-shelf (COTS) technologies and open source software. SADE provides a means for legacy and state-of-the-art applications to coexist and interoperate under demanding conditions.

This white paper highlights the available technologies, future trends, and strategies for incorporating COTS platforms and open source software in modern situational awareness frameworks.

Situational awareness is the comprehension and contextual analysis of a current environment. It typically uses detection and deep learning techniques to isolate, identify, and prioritize objects relevant to the user and critical for tactical decision-making. Situational awareness has increased importance at the dynamic edge. Here the ability to capture, correlate, and analyze available sensor data efficiently allows personnel to assess potential dangers, communicate with stakeholders, and make quick, data-driven decisions.

The kinetic environment of the dynamic edge

The dynamic edge is a chaotic, kinetic, and fragmented environment of people, devices, vehicles, and autonomous systems that must be able to operate without assistance from data center or cloud resources. Despite highly reliable communications infrastructures and redundant paths through cellular and Wi-Fi, continuous communication with command-and-control resources isn't guaranteed at the edge. Cloud-to-edge topologies for direct management and orchestration of devices remain a challenging proposition for highly kinetic use cases. Numerous mission-critical decisions must be made using local sensor data, processed, and presented in real time. The edge frequently has a different operational tempo than the cloud, with varying node and communications conditions demanding immediate adaptation. Because of this tempo, the edge must first be defined and optimized for autonomous operation before establishing the means for cloud interoperability.

The dynamic edge has become the focus for a growing number of use-case scenarios in commercial as well as defense and government sectors. The pressure to reduce costs and centralize assets has driven economies toward the cloud. But critical-edge event-driven activities cannot withstand the potential latency and bandwidth issues inherent in scale edge-to-cloud topologies. And many scenarios must confront the risk of passive or active denial of communications between the edge and cloud that can affect operational safety, node survivability, and mission assurance at the edge.

Situational awareness and critical decision-making

Situational awareness is a continuous process that spans the computing continuum from the edge to the cloud, delivering insights on the state of events within a geographic location as they unfold. At the dynamic edge where tactical emergency and defense operations often occur, situational awareness is critical to decision-making that impacts immediate mission objectives and operator survivability. Here, the data sets may be small, but their value is significant in understanding immediate conditions. And the relevance of these data sets may expire rapidly under changing conditions.

At the cloud level, situational awareness supports data analytics at scale for strategic planning and logistical analysis. Data sets acquired from multiple ground units are synthesized into an overall situational awareness seeking to understand the larger trends and areas of growing interest and concern. Consequently, the decisions made may impact many edge units. In a broad way, the dynamic edge must operate appropriately to local conditions and asynchronously to the tempo of the cloud.

Figure 1 highlights the continuous data interaction and analysis required to deliver situational awareness. This edge-to-cloud loop of operations can be viewed as a series of concentric rings cutting across each segment of the computing environment. The smallest ring in the center represents the edge, and a series of larger rings represents intermediate and cloud infrastructure. Abstractions and data set sizes vary for each ring, and each has its own cadence based on how quickly local conditions change.

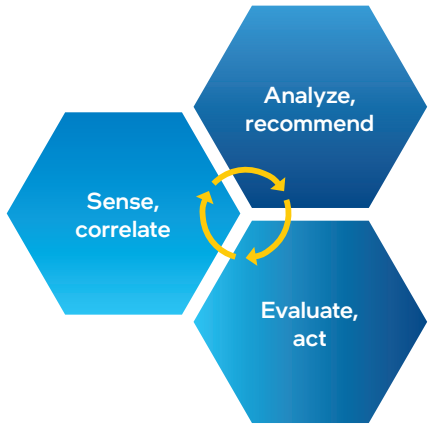

Figure 1: Situational awareness is derived from a continuous loop of operations.

Deploying Situational Awareness at the Dynamic Edge

Situational awareness value and responsiveness can be represented as a scale of relevant time-critical recommendations by theater of action vs. logical distance (latency) from the edge (Figure 2). Under nominal conditions with excellent communications and traffic environment, an edge-to-cloud-to-edge operation round trip will take perhaps hundreds of milliseconds. The actual time depends on the complexity of data and computational operations. It could be even longer if a human operator is in the loop.

So long as a response isn't tied to an immediate mission-critical decision, the impact of higher-latency operations may be trivial. An example is cloud operations, which may include interpreting field sensor data, working in a decision loop, and finally communicating a recommendation in a low kinetic environment.

At the other extreme in high operations at the dynamic edge, local situational awareness impacts immediate mission-critical operations decisions that influence mission success. These decisions can be sporadic in nature, using highly localized sensor data that can expire quickly. But to provide a reasonable window for reaction time to events, a machine-to-machine class of operational speeds may be necessary. In any case, the potential delays in a round trip to the cloud and back can deplete so much of the window for decision-making as to render the edge operators vulnerable to rapidly changing local conditions.

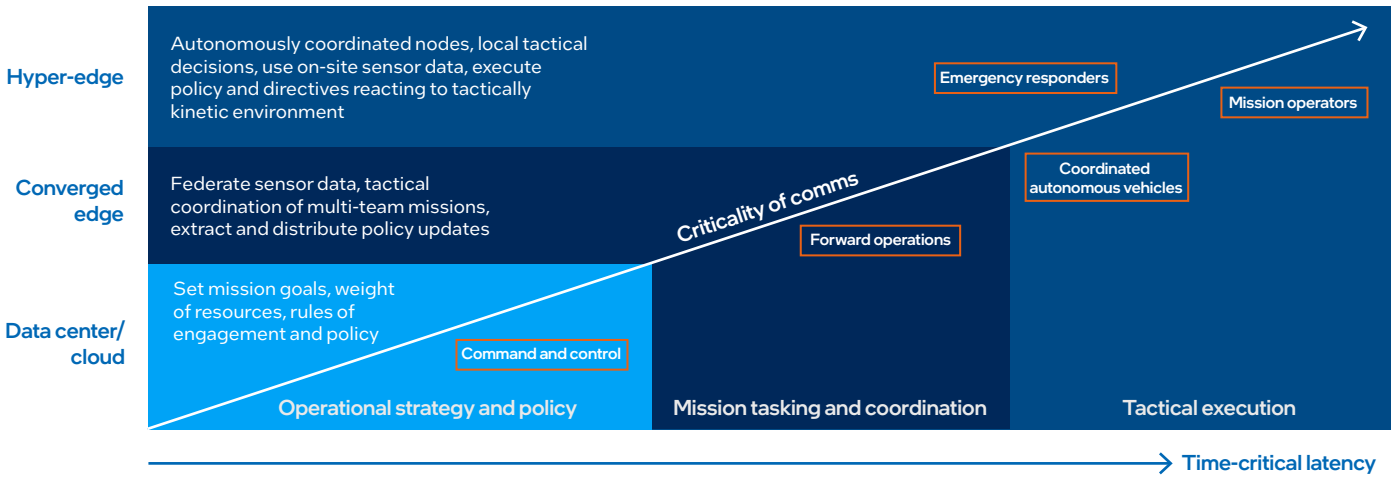

Figure 2: End-to-end characteristics of cloud-to-dynamic-edge topology.

Requirements for Situational Awareness at the Dynamic Edge

Large emergency and defense operations must have a situational awareness solution that can scale across expansive geographic areas. The solution also must correlate inputs from hundreds or even thousands of operators, sensors, and endpoints. At the dynamic edge, beyond normal line of sight, modular, extensible, and resilient connected solutions are essential assets to orient personnel quickly and guide their decision-making as they adapt to local environments.

Solutions deployed at the edge must be highly adaptive and self-healing to withstand a range of environmental and artificial conditions. They must also offer the design flexibility to perform optimally in both small isolated units and large heterogeneous networks with whatever local assets are available. When deployed at the edge, solutions must be designed to optimize autonomously with minimal operator intervention.

Dynamic edge nodes could find themselves separated from the cloud due, for example, to passive or active communications denial by aggressors. In this situation it's vitally important that they're able to locate teammates, establish a local, trusted mesh network, and share computational and sensor resources as a micro–data center. Continued operation of critical applications and sharing of valuable sensor data and analytics are crucial for decision-making. The goal is to provide all legitimate devices with maximum possible situational awareness to allow continued mission operations and survivability of assets. Below is a summary of some key features for a dynamic edge software fabric.

Edge software architecture requirements for situational awareness

A software architecture for situational awareness at the dynamic edge:

- Dynamically identifies, connects, and configures multiple sensor types
- Fuses multiple sensor data types, synchronizes and stores data at the correct local edge node to optimize data and analytics operation
- Intelligently synchronizes data across the edge and cloud to ensure ground truth (GT) data coherency
- Provides infrastructure to:
  - Train neural networks for object identification, radio frequency patterns, extraction/classification, and natural language comprehension
  - Learn or reinforce neural networks using small data sets, preferably at the edge
- Isolates key functional capabilities in containers or virtual machines with the ability to manage and orchestrate these workloads—and the capability to do the same with native applications running at the edge
- Dynamically reconfigures and creates stand-alone micro–data center teams from disconnected edge nodes. Manages multiple workloads and requirements to continue team operations
- Provides web standards for connecting and rendering data in web applications or browsers to support multiple target platforms and the availability of services to field operators
- Collaborates, transiently, with the cloud providing a means for zero-trust connectivity and asynchronous operations
- Designs for multiple heterogeneous and lightweight edge nodes

Optimizing data with Intel's modular framework

Intel has analyzed the technical requirements for achieving effective situational awareness from the air to the ground. The framework, known as Situational Awareness at the Dynamic Edge (SADE), can coordinate and manage multiple heterogeneous platforms and sensor packages with varying levels of autonomy. The Intel Internet of Things Group uses commercial, open source, and modular open systems approach (MOSA) expertise to bring together architectural innovation and industry-tested customer solutions.

SADE is a portable, scalable software framework that enables efficient software deployment and scale management of multiple mission-critical applications and services across multiple, connected heterogeneous nodes. At its core, SADE is a managed, high-speed software messaging bus, network, and application orchestration infrastructure. It is also a containerized services and applications layer supporting a variety of applications and services (see Figure 3). The SADE messaging bus enables services and applications to coexist, share information, and collaborate efficiently on single or multiple nodes.

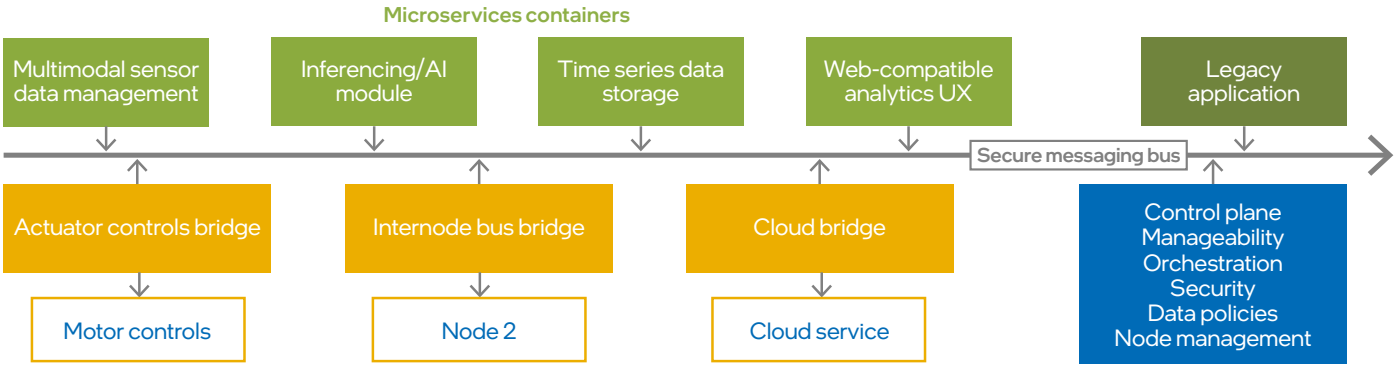

Figure 3: Situational Awareness at the Dynamic Edge (SADE) framework.

The SADE architecture isolates functions in microcontainers (light green boxes). It also may include legacy application containers (dark green box) for compatible interoperation and ease of deployment. With their connections to the SADE infrastructure, functions may be applied to other applications or services (Function as a Service/FaaS) on the same platform or distributed across multiple physical platforms. By reducing the amount of redundant code and intelligently distributing the processing, sensor, and accelerator requirements across multiple platforms, SADE enables sophisticated applications to be developed on lightweight devices.

The framework has a control plane responsible for management engines that control:

- Physical nodes attached to the network, their configurations, and network security
- Container placement and load optimization
- Policy engines that coordinate data management and prioritization of application execution
- Bridge connection to cloud-based systems and management/orchestration infrastructure

The SADE control plane can operate connected to the cloud/command and control data center. It can also use inherited policy settings in a hierarchical fashion that optimizes isolated stand-alone operation. This management and orchestration engine can support assets at both the mission operator and autonomous system levels when disconnected from the cloud or command-and-control systems.

Using the SADE approach, monolithic applications would, optimally, be refactored into lightweight, distributed functional building blocks housed in Docker-compatible containers exposing a standard set of data and APIs. These containers coexist with similarly packaged, commercially tested technologies that communicate with one another using the SADE messaging protocols and data/APIs.

SADE's modular, open systems approach greatly simplifies the deployment process, and the component-level architecture speeds updating of in-field components. Through the messaging interface, applications can be assembled in a filter graph–like fashion whereby input and output functions, algorithms, and data formats are shared among applications. This enhances the system's adaptability to address new use cases while reducing overall code complexity and redundancy

Leveraging COTS and open source technology

Traditionally, systems tasked for situational awareness have been built on dedicated, proprietary hardware with custom software, data formats, and connectivity. This approach limits the flexibility of a system. It is difficult to adapt or upgrade the system to handle new problems, interoperate or share workloads with other systems, and evolve to new standards and technologies. In the future, compute compatibility and networking across devices such as wearables, vehicles, autonomous systems, radar, and communications platforms will allow operators greater flexibility. They will be able to transfer critical workloads across any available asset or dynamically assemble micro–data center clusters from heterogeneous nodes to handle more-intensive tasks.

The Intel design philosophy focuses on ensuring the compatibility and scalability of Intel® architecture-based platforms and their alignment with industry-standard approaches to connectivity, memory, and storage. An enormous ecosystem of open source and proprietary software offerings is built upon Intel® architecture, supporting customer needs and providing multiple avenues for incorporating the latest technological advances.

The SADE framework is designed to take full advantage of how COTS technology is developed and deployed. Ultimately, SADE serves as the foundation for modern software development, enabling new applications to be built and features quickly integrated or updated to address changes in the operating environment. The framework takes maximum advantage of open source technologies, leveraging the availability of commercially developed code, as well as the speed at which code is evolving.

This modular SADE architecture aligns with the latest techniques in application containerization successfully deployed by cloud service providers and corporate data centers, but reimagined and architected with the edge as the primary target. Developers can easily add functionality and deploy solutions as "plugin" modules with high confidence in their solutions' operational stability. When building on the Intel® architecture and ecosystem, software solutions scale across a wide range of well-tested, industry-proven power/performance hardware, network, and storage configurations.

Intel: Driving the future of situational awareness

Future-proofing is of paramount importance in the design philosophy driving Intel® platform and software development. Accordingly, the SADE framework is easily adapted to evolving technology and industry standards, so operators can take full advantage of the latest capabilities in the future. As system designs and improvements evolve in AI accelerator technology, CPU and GPU processing power, and FPGAs and platform functionality, Intel's SADE framework enables customers quickly to evaluate, select, and optimize their platforms.

More importantly, SADE's foundational design efficiently supports new functionality to meet the future needs of mission operators while maximizing compatibility with the latest advances in COTS hardware performance and innovation.

Learn more

For more information about the Internet of Space Things discussed in this white paper, please contact your Intel account executive, or email us at **IOTG-PublicSector@intel.com**.

Background information

Messaging bus architecture: **zeromq.org**

Docker container background: **docker.com**

Kubernetes orchestration layer (SADE supports Kubernetes and Docker Swarm constructs): **kubernetes.io**

Application management infrastructure: **openness.org**

AI inferencing using the Intel® Distribution of OpenVino™ toolkit: **openvino.ai**

1221/LM/CMD/PDF