Zero-Trust Architecture for Autonomous Edge Computing

Abraham K. Ishihara, Moustafa Abdelbaky, Sandeep Shetye

I. Abstract

We are at the apex of an aviation revolution where autonomy will play a central role in enabling complex, multi-agent systems to communicate, interact, and collaborate on a myriad of applications spanning autonomous swarms to wild-fire management. Autonomy is not an absolute but rather a spectrum ranging from a system requiring significant human intervention to one requiring little to none [1]. For example, the extreme, in the case of an autonomous aircraft, is one that operates independently in the airspace interacting with all other elements (air traffic controllers, other pilots) as if it were a human pilot. Critical to this vision is an architecture that enables autonomous agents to interact with minimal latency.

Edge computing is an emerging architecture where compute and storage is pushed to the 'edge' of the network in order to minimize the round-trip time from agent to resource thereby mitigating the latency associated with cloud-only based approaches. Additionally, services can generate massive amounts of data (e.g., video feeds), which may require analysis in near real-time. Moving this data to the cloud for further processing may not be feasible due to latency, bandwidth, and cost. Privacy, security, and reliability can also be improved by edge computing architectures.

However, this geo-distributed and dynamic* architecture complicates the establishment of unambiguous network security boundaries and can lead to vulnerabilities including man in the middle attacks, replay attacks, physical security breaches of edge nodes, signal interception, etc. This motivates the need for zero-trust architectures [2–4] which de-emphasize the notion of static network perimeters and, as the name implies, do not instill any innate trust in any particular agent. It is required that all agents must be authorized and approved in every transaction. In this paper, we present a zero-trust architecture suitable for edge-computing applications that demand significant low-latency, security, privacy, and reliability.

This work builds on the Data and Reasoning Fabric (DRF) project which is a framework for seamlessly connecting data and reasoning service providers to vehicles and other service consumers [5]. Motivated by the fact that current data platforms are simply not designed for the projected scale of geographically dispersed, heterogeneous entities with low-latency requirements nor the required support for ubiquitous closed feedback loops, DRF aims to build a distributed, decentralized marketplace enabling the execution of safety-critical operations in real-time. DRF aims to retain current levels of safety even with increased air travel density, complexity, and user communities while ensuring interoperability and security across the cloud-to-edge continuum. In particular, DRF is an open and scalable framework delivering the interfaces, protocols, tools, and unifying software architecture to connect nodes across vehicles, edge and cloud infrastructure to seamlessly work together, exchanging data as well as reasoning services for real-time and non-real-time decision-making by all users of the airspace system.

Fig. 1 highlights the architecture for the edge computing zero-trust framework that will be described in detail in the final version of the paper. Leveraging the core capability of the current DRF platform [5] the zero trust architecture will enable secure transactions between consumers and providers on the edge.

The remainder of this abstract describes the application that will be explored under this architecture.

Application: We will investigate the application of an autonomous swarm where, due to severe computational and energy constraints, there exists the need to migrate complex operations to the edge during the operation. Since there are numerous tradeoffs in term of computational performance and latency, various objective functions are used and fed into an optimization algorithm. The scenario begins with a swarm of UAVs deployed in an urban environment in response to a disaster management objective or search and rescue operation. The task requires *collaboration and communication* in addition to the complex dynamics of the swarm trajectories as a whole. Each agent must interact with other agents to

*Edge nodes are often dynamic and allocated based on the needs of the agents and associated applications. In the case of urban air mobility (UAM) applications, computation may need to migrate dynamically from edge node to edge node as the vehicle may move with significant velocity and/or its mission may change abruptly in flight.

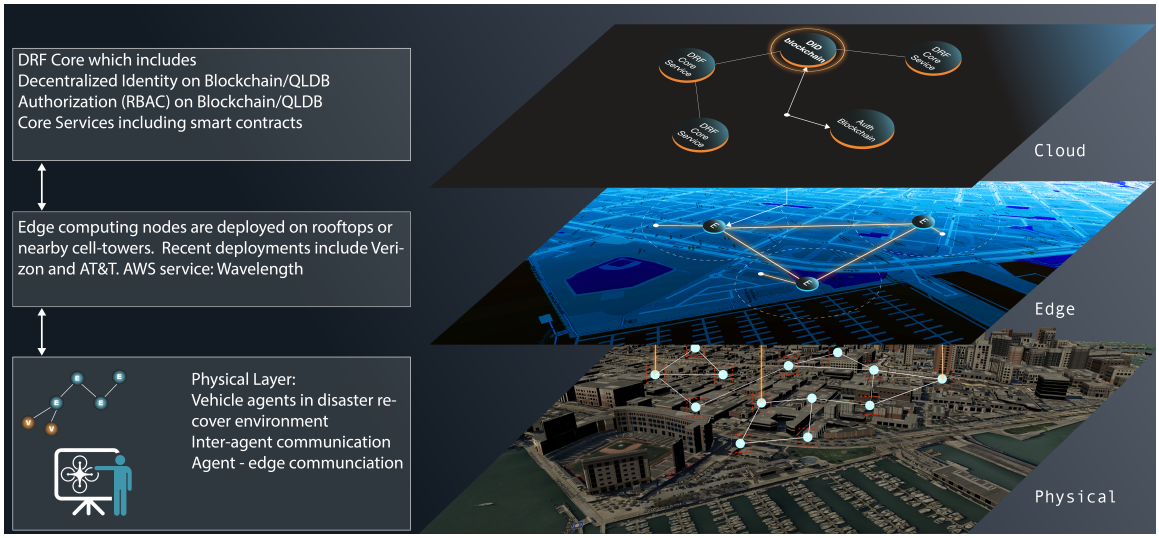

Fig. 1 Zero-Trust Edge Computing Architecture in DRF

maintain the swarm and in addition must offload computation to the edge. The dynamics of the swarm are given below

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = \beta \sum_{j=1}^{N} \beta_{ij} \left(v_j - v_i \right) + l_i \\ x_i(t_0) = x_{i0}; v_i(t_0) = v_{i0} \end{cases} \tag{1}$$

where $\beta > 0$; $\beta_{ij} : \bar{\mathbb{R}}^+ \to \bar{\mathbb{R}}^+$ represents the influence agent j has on agent i and is given by

$$\beta_{ij} = \zeta_{ij} \phi(\|x_i - x_j\|) \tag{2}$$

where

$$\zeta_{ij} = \begin{cases} 1 & \text{if } j \text{ connects to } i \\ 0 & \text{o/w} \end{cases} \tag{3}$$

and $\phi : \bar{\mathbb{R}}^+ \to [0, 1]$ is a non-increasing, non-negative function with compact support, that is, there exists a $\sigma > 0$ such that $\phi(r) = 0$ for all $r \geq \sigma$ with $\phi(0) = 1$; $l_i : \bar{\mathbb{R}}^+ \to \mathbb{R}$ represents attraction, repulsion, and leadership terms; $(x_{i0}, v_{i0}) \in E \times E$ denote the initial conditions of agent i where $E = \mathbb{R}^d$ and $d \in \{1, 3\}$, and $i \in \{1, 2, \ldots, N\}$.

Previous Work: In Vehicular Ad hoc Networks (VANETs), vehicles can communicate and transmit data using cellular networks or Road Side Units (RSUs). However, vehicle obstructions can hinder this ground-ground link thus lowering service quality. In addition, each vehicle has a finite communication range and locally has limited computational ability which is not ideal for certain latency-sensitive tasks. [6] proposes a mobile edge computing (MEC) architecture using UAV-assisted Vehicular Ad hoc Networks. By using UAVs as an air base stations, data transmission can be improved by air-to-ground wireless channels. In this scenario, UAVs are equipped with MEC servers where a ground vehicle can offload its tasks. By selecting the optimal MEC server for task offloading, the processing delay in this air-to-ground communication can be minimized. Similar work in optimal task offloading can be found in [7, 8]

Approach: The approach presented here is similar to [6] but is applied to a swarm of UAVs instead of ground vehicles, with fixed edge nodes in known locations. We assume there are M UAVs which act semi-autonomously and are assigned real-time computational tasks such as terrain imaging, object tracking, or trajectory-related calculations for the given mission. At any given transmission, a UAV may offload parallelized components of computational task packets to K number of edge nodes, labeled j, where K is a (small) integer. Each parallel task packet is endowed with an attribute array ζ_i where i is the UAV index (same index in equation (1). The details of the attribute array ζ_i are as follow:

$$(d_i,\ c_i\, T_i^{th},\ \{\lambda_{i,j}\}_{j=1}^{N}) \tag{4}$$

where d_i represents the size of computing task, c_i represents the computing resources required for task execution, T_i^{th} represents the task minimal allowed latency, and $\lambda_{i,j}$ represents ratio of offloaded task size to total task size for a UAV-node pair. The array λ is one of the variables which we will be optimized while other attributes are treated as parameters i.e. inputs fed into the algorithm.

To summarize, at any given WiFi transmission interval, UAV vehicle i can offload a task of size $\lambda_{i,j}d_i$ to a ground edge-node j, and compute the rest of the task, sized $(1-\lambda_{i,j})d_i$, locally; then the various offloaded, parallelizable sub-tasks must be executed on the respective edge nodes, and the results transmitted back via WiFi to the originating UAV vehicle i.

Other optimization variables include array $x_{i,j}$ which defines the node j selection of UAV vehicle i for task offloading, and array $f_{i,j}$ which represents partial computation frequency of offloaded UAV subtasks on edge nodes. In general, the optimization problem we are dealing with involves trade offs between the computing capacities/speed and data transmission latencies. The total processing delay T_{ζ_i} of a task is a function of local computation time $T_{i,j}^{loc}$ and edge node processing delay $T_{i,j}^{node}$. The total is a sum of three different time components as follow: (1) the task transmission time $T_{i,j}^{trans}$, (2) the task returning delay $T_{i,j}^{rtn}$, and (3) the task computing time $T_{i,j}^{Edge}$.

The general details of our optimization problem is as follows. At a given clock time, we will minimize the sum of computation times (locally and on edge nodes) of <u>all</u> UAV's upcoming tasks. The variables which be optimized are task-offloading connectivity array x, sub-task fractional size array λ, and partial computation frequency array of offloaded task f. The optimization is subjected to a list of constraints outlined in the equation below.

$$
\begin{aligned}
&\min_{x,\lambda,f} T(x,\lambda,f) = \min_{x,\lambda,f} \sum_{i=1}^{M} T_{\zeta_i} \\
&s.t.\ \text{C1.1} : T_{\zeta_i} \le T_i^{th},\ \forall i \in \{1,...,M\} \\
&\quad \text{C1.2} : \sum_{j=1}^{N} x_{i,j} \le K,\ \forall i \in \{1,...,M\} \\
&\quad \text{C1.3} : x_{i,j} \in \{0,1\},\ \forall i \in \{1,...,M\},\ \forall j \in \{1,...,N\} \\
&\quad \text{C1.4} : 0 \le \sum_{i=1}^{M} x_{i,j} f_{i,j} \le F_j,\ \forall j \in \{1,...,N\} \\
&\quad \text{C1.5} : 0 \le f_{i,j} \le F_j,\ \forall i \in \{1,...,M\},\ \forall j \in \{1,...,N\} \\
&\quad \text{C1.6} : 0 \le \sum_{j=1}^{N} x_{i,j}\lambda_{i,j} \le 1,\ \forall i \in \{1,...,M\} \\
&\quad \text{C1.7} : 0 \le \lambda_{i,j} \le 1, \forall i \in \{1,...,M\},\ \forall j \in \{1,...,N\}
\end{aligned}
\tag{5}
$$

The constraint C1.1 ensures that neither the local computation time $T_{i_j}^{loc}$ nor the processing delay of the subtask assigned to edge node j, $T_{i,j}^{node}$, can exceed maximum allowed latency T_i^{th} of the task ζ_i. Constraints C1.2 and C1.3 indicate that each UAV selects at most K edge nodes for task offloading at any given time. Constraints C1.4 and C1.5 guarantee that the total amount of computing resources allocated by the nodes to all tasks cannot exceed the total computation capacities of the respective nodes. C1.6 and C.1.7 represent that total offloaded task fractions cannot exceed unity.

Lastly, we will augment (5) with latencies and constraints associated with the zero-trust architecture in which the UAVs are consumers of edge node providers. All entities are DRF participants and therefore have DRF decentralized identifiers (DIDs). Their interaction under the zero-trust paradigm results in additional constraints that impact the solution of the optimization problem. We will examine corresponding tradeoffs and present simulation results in the final version of this paper.

References

[1] Council, N. R., et al., *Autonomy research for civil aviation: toward a new era of flight*, National Academies Press, 2014.

[2] Wylde, A., "Zero trust: Never trust, always verify," *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2021, pp. 1–4. https://doi.org/10.1109/CyberSA52016.2021.9478244.

[3] Zhang, P., Tian, C., Shang, T., Liu, L., Li, L., Wang, W., and Zhao, Y., "Dynamic access control technology based on zero-trust light verification network model," *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, IEEE, 2021, pp. 712–715.

[4] Rose, S., Borchert, O., Mitchell, S., and Connelly, S., "Zero trust architecture," Tech. rep., National Institute of Standards and Technology, 2020.

[5] Abdelbaky, M., Chen, J., Fedin, A., Freeman, K., Gurram, M., Ishihara, A. K., Joe-Wong, C., Knight, C., Krishnakumar, K., Reyes, I., et al., "DRF: A Software Architecture for a Data Marketplace to Support Advanced Air Mobility," *AIAA AVIATION 2021 FORUM*, 2021, p. 2387.

[6] He, Y., Zhai, D., Huang, F., Wang, D., Tang, X., and Zhang, R., "Joint task offloading, resource allocation, and security assurance for mobile edge computing-enabled UAV-assisted VANETs," *Remote Sensing*, Vol. 13, No. 8, 2021, p. 1547.

[7] Mukherjee, A., Misra, S., Chandra, V. S. P., and Obaidat, M. S., "Resource-optimized multiarmed bandit-based offload path selection in edge UAV swarms," *IEEE Internet of Things Journal*, Vol. 6, No. 3, 2018, pp. 4889–4896.

[8] Mukherjee, A., Misra, S., Sukrutha, A., and Raghuwanshi, N. S., "Distributed aerial processing for IoT-based edge UAV swarms in smart farming," *Computer Networks*, Vol. 167, 2020, p. 107038.