Engineering of Edge Software Systems

A Report from the November 2022 SEI Workshop on Software Systems at the Edge

Carnegie Mellon University
Software Engineering Institute

Authors
Grace Lewis
Ipek Ozkaya
Kevin Pitstick

JUNE 2023

Introduction

INCREASING PACE OF INNOVATIONS IN HARDWARE AND ARTIFICIAL INTELLIGENCE (AI) SOLUTIONS is enabling a range of software systems to be deployed closer to their users, a paradigm referred to as edge computing. Software systems at the edge allow users to process data closer to where it is generated, resulting in timelier decision making. Increased timeliness in decision making has several advantages for defense and national security applications for edge software systems. However, innovative and successful software systems at the edge need to evolve rapidly to realize constrained scenarios requiring resource-awareness and dynamic mission adaptation. Today many of the edge systems supporting battle management command and control (BMC2) and forward-deployed military operations are not designed to be cyber resilient or flexible enough to adapt to limited computing resources, intermittent or denied network connectivity, security concerns, high cognitive load, and operational uncertainties.

While much of industry is focused on the hardware and networking aspects of edge computing, much less attention is being given to the software stack that enables mission capabilities, despite the operational uncertainty that is typical of tactical environments. To fill this gap, the Carnegie Mellon University's Software Engineering Institute (SEI), a federally funded research and development center, is working to define and improve the software stack for edge systems, as well as tools for their development and deployment.

Workshop Purpose

The future of defense and national security systems will increasingly be at the edge, as the need to enable computational decision making closer to users and where the data is generated will only increase.[1] Edge systems will need to not only incorporate AI-driven innovations swiftly, but also be able to operate in heterogeneous hardware and network environments. Software is the glue that brings together data, hardware, and networks for edge systems. The ability to respond to these emerging needs effectively requires a roadmap for edge software systems that identifies existing solutions, opportunities, and gaps. Towards this goal, the SEI convened an invitation-only workshop in November 2022.

The goal of the workshop was to bring together key national defense stakeholders of edge software systems to share challenges, solutions, and lessons learned and to identify top-priority areas that require greater research and development. Workshop participants included representatives from the Office of the Undersecretary of Defense for Research and Engineering (OUSD R&E), the U.S. Marine Corps, the U.S. Army, the U.S. Navy, the Naval Sea Systems Command, the Program Executive Office (PEO) Intelligence Electronic Warfare & Sensors (IEW&S), the Defense Advanced Research Projects Agency (DARPA), Qinetiq, MIT Lincoln Labs, and the RAND Corporation. Based on input from the workshop participants, we defined three main areas in which we need to collectively make progress to provide reliable capabilities at the edge and in turn enable deployment and use of AI and machine-learning (ML) capabilities at the edge. The focus areas are

- resource adaptation at the edge
- rapid deployment to the edge
- data and data architecting for the edge

This report describes the challenges and recommendations for these focus areas, which the workshop discussions elicited.

What is the Edge?

The edge is a term that refers to the edge of the connected network. **Edge systems** are systems that operate at the edge, close to where data and computation are needed. Edge systems are often part of larger distributed systems where cloud nodes and edge nodes interact, with edge nodes serving as intermediaries between proximate users and the cloud. The term **cloud-to-edge continuum** describes the interaction between cloud and edge nodes in which data and computation move between them as needed. **Edge computing** then refers to data processing and computation tasks near the edge. The terms edge, edge systems, and edge computing are often used interchangeably and tend to focus on *hardware, network,* or *data aspects.* We use the term **edge software system** to highlight the critical role that *software* plays, enabling advances in edge hardware and networks, as well as efficient data storage, access, processing, and analysis for decision making at the edge. Finally, the term **tactical edge** refers to the more constrained, diverse, uncertain edge environments in which military personnel execute missions in areas that range from air to sea to ground.

Challenges and Recommendations for Engineering of Edge Software Systems

The focus of this workshop was to envision the future of software systems at the edge so that we can ensure they can address these challenges, adapt as technologies rapidly evolve, and stay relevant to mission, threat, and operational realities. The three areas we identified as a result of the workshop represent areas that need higher priority research attention.

RESOURCE ADAPTATION AT THE EDGE

Edge software systems need to adapt to changing missions and dynamic operating environments. While connectivity to the cloud is not an issue in urban, industrial, or enterprise edge environments,

it cannot be guaranteed in tactical environments that experience higher levels of uncertainty. Software systems at the tactical edge operate in disconnected, intermittent, limited (DIL) environments, with active attackers, changing missions and environmental conditions, and shorter timeframes for decision making.[2, 3, 4, 5, 6, 7]

The characteristics of edge environments combined with resource limitations of edge devices pose challenges to both the design of adaptation mechanisms and their actual execution during operations.

CHALLENGE 1: TOOLS AND TECHNIQUES FOR ELICITING AND DESIGNING FOR UNCERTAINTY ARE LACKING.

Edge software systems operate in environments with high levels of uncertainty because they are deployed in DIL environments, often with attackers trying to sabotage the mission that the systems support. The amount of uncertainty in the operating environment and available resources drive architecture design decisions for coping with the uncertainty. For example, a system that is expected to operate in a 70% uncertain network environment will be architected differently from one that operates in a 30% uncertain network environment. While there is a large body of work studying adaptive and self-adaptive software systems, outcomes are often not mature enough to be deployed in industrial and defense edge environments.[8, 9] Moreover, these outcomes lack an explicit focus on enumerating sources of uncertainty, eliciting ranges of variability, and mapping design approaches accordingly. Consequently, users are left to manage operational risks while they have minimal trust in the systems. Developers today lack tools and techniques that can help with architecting systems that can quickly sense, adapt, and respond to environmental changes and security threats.

CHALLENGE 2: EFFICIENT COMPUTATION DISTRIBUTION REQUIRES TRUSTED EDGE NODES AND COMPUTATION TASKS.

Edge software capabilities may need to be split across multiple edge devices for several reasons, including: (a) resource constraints of edge devices limit the amount of computation that can be executed on a single device, (b) high risk of loss of edge devices requires replication of capabilities across multiple devices for higher resiliency, (c) dynamic availability of resources enables capabilities to be allocated on-the-fly to improve performance, and (d) limited battery and computation power requires edge devices to offload resource-intensive computation to more powerful resources (e.g., edge servers, cloud servers). However, distributed systems come with many complex challenges, especially considering the cloud-to-edge continuum. For example, when pooling heterogeneous resources to fulfill a mission, many questions arise that are not technical yet are critical: What is the policy for sharing resources? If I share a resource, can I trust authorized use? Can I trust all available shared resources? While these questions may be easy to answer if the same organization owns all nodes and computation tasks, they are not easy to answer in environments where many organizations come together to support a mission, especially in coalition environments and multi-domain missions. There is room for improvement both in technical and organizational solutions.

CHALLENGE 3: DATA DISTRIBUTION IS COMPLICATED BY UNRELIABLE AND HETEROGENEOUS EDGE NETWORKS.

Reliable network access and bandwidth cannot be guaranteed in edge environments. In multi-domain and coalition scenarios where different organizations and networks come together to support missions at the edge, edge systems will likely operate in environments with a combination of short-range and long-range networks of varying reliability. Some links might be reliable but low bandwidth, some might be reliable but only available at set times, and some may come in and out unexpectedly. These networks connect a variety of edge nodes, ranging from a simple mobile device to moderately sophisticated nodes (such as drones) to portable compute environments with server racks and network-attached storage. All these edge nodes have different network requirements. Dealing with this mix of network links and edge nodes and adapting to maximize the use of bandwidth when it becomes available are difficult design problems. One of the main challenges in these conditions becomes data distribution—ensuring that data arrives where and when it is needed despite the unreliability of the network. Different levels of reliability will drive different design decisions to manage fault tolerance and network failure and unavailability. For example, a system with planned periods of connectivity will be architected differently than a system with unplanned periods of connectivity. In addition to technical challenges, there are policy challenges related to deciding what data to send when networks and bandwidth become available and how to implement those policies in software.

Recommendations: Edge software systems must be robust and resilient against changes in the operating environment and against adversarial control and action. This means that they must be resource aware, network aware, and policy aware.

Closing Organizational and Policy Gaps

- Clear policies for resource prioritization and data prioritization in edge environments need to be defined and implemented in software to support automation. Understanding network requirements for edge capabilities and a prioritization of those capabilities based on policy can inform network allocation mechanisms so that bandwidth is assigned to the highest priority capabilities for the mission.
- Development processes must prescribe the use of threat-modeling techniques such as STRIDE[10] or Attack Trees[11] to help define security requirements for edge systems and understanding tradeoffs.

Closing Technical Gaps

- Greater research is needed on designing for uncertainty. Development processes for software systems at the edge must include activities that focus on understanding available resources at the edge and their uncertainty before developing a solution, as these factors will define many critical design decisions.
- Rapid progress needs to be made in mechanisms for decentralized authentication and authorization that will enable more dynamic and trusted computation allocation, especially for coalition and multi-domain mission support.

RAPID DEPLOYMENT TO THE EDGE

The Department of Defense's (DoD's) Software Modernization Strategy[12] lays out a vision where the DoD can rapidly deliver software capabilities to the battlefield, providing warfighters with the latest innovations they need for their mission. Increasing deployment speed at the edge will require more efficient and secure pipelines for software delivery, as well as new strategies to keep the software up to date in changing environments.

CHALLENGE 1: COMMERCIAL AND DOD DEVSECOPS APPROACHES DO NOT TRANSFER DIRECTLY TO PUSHING COMPUTATION TO THE EDGE RAPIDLY AND EFFICIENTLY.

The uncertain and dynamic conditions that warfighters operating at the edge experience require frequent capability updates. However, as edge environments are often DIL environments, this results in fewer opportunities to push software updates to edge devices. In addition, these updates must be small because edge devices are resource limited. As the DoD prepares its network infrastructure for conflicts with near peers, it will face new scenarios of active denial, such as adversarial ML attacks on 5G systems.[13] If edge systems are disconnected, it inhibits their agility and ability to be fully cooperative nodes in tactical systems. These constraints prevent new updates of capability from keeping up with the operational tempo of the warfighter. DevSecOps strategies embraced by the commercial sector and the DoD (such as the Air Force's Platform One) provide a good starting framework for utilizing automation to update cloud software in short timeframes. However, this automation does not extend to edge updates and deployment to meet mission needs in tactical timeframes. One major reason for this is that these edge systems often require safety certifications for any code modifications prior to deployment, and current DevSecOps practices lack rigorously defined processes with traceability of requirements to support this.[13]

CHALLENGE 2: EVEN IF NETWORK AND PLATFORM CHALLENGES ARE RESOLVED, PUSHING COMPUTATION TO THE EDGE SECURELY REMAINS A CHALLENGE.

The next major barrier to rapid edge deployment is security. As evidenced by the president's recent Executive Order (EO) 14028 on Improving the Nation's Cybersecurity, security is increasingly a major concern for all software across the government.[14] As the DoD looks to secure its software supply chain, edge software has the unique security challenge of operating in actively denied and adversarial environments, resulting in issues of trust and verification of identities of edge system users. As over-the-air (OTA) updates are deployed, guarantees need to be in place to ensure that software packages arrive unadulterated without breaking device operation. Another security concern that conflicts with rapid deployments is receiving authorization to operate (ATO), which can be a long and costly process. The last concern is how to incentivize organizations to prioritize security, given that security costs do not provide extra capability to the military operator.

CHALLENGE 3: DOD EDGE SYSTEMS ARE NOT BUILT AS CLOSED-LOOP SYSTEMS.

Current software supply chain pipelines focus mainly on pushing software out to the edge, but the feedback loop from the edge to the development and build environment is often slow and ill-defined—and sometimes even nonexistent. Much focus has been placed on collecting data from the edge, storing it, and analyzing it. However, the pathway and infrastructure need to include tasks and opportunities to revise software capabilities based on this data. Tactical edge environments tend to be filled with uncertainty that can cause deployed capability performance to degrade. Development teams can use the collected data to understand and address additional operator needs, such as vulnerabilities, new functionality, or broken functionality. In particular, the rise in deployment of AI and ML systems at the edge further amplifies the need for this feedback loop. As AI and ML models are very data driven and situation dependent, monitoring model performance is key to determining when the model is no longer relevant and needs to be retrained. In addition, data at the edge needs to be collected, labeled, and used for retraining. This is critically important for ensuring that warfighters receive the best situational awareness and that they can trust deployed capabilities. Closed-loops systems enable filling gaps in timeliness, accuracy, and trust.

Recommendations: To enable rapid deployment of capabilities to the edge, deployment pipelines for edge systems need to use feedback mechanisms to be more efficient and secure.

Closing Organizational and Policy Gaps

- Automated, immutable workloads (e.g., containers) must be used to deploy software to the edge instead of updating software through manual access, as directed by the Federal Zero Trust Architecture Strategy.[15] This increases security by limiting direct user access and reduces errors that manual changes introduce.
- Automated DevSecOps platforms need to be embraced and extended to support the safety-critical certification processes required for edge systems.
- Processes such as the Air Force's Fast-Track ATO or continuous ATO (cATO) need to be adopted to provide ATO pathways that are more agile.

Closing Technical Gaps

- Investment is required to bring connected network infrastructure to edge environments, specifically investment focused on techniques to combat adversarial and active denial of communications.
- Security for edge networking requires additional investment in areas that include authorization and authentication methods, secure-by-construction software architectures, and active security monitoring to detect new and evolving threats.
- Research is required for active monitoring of edge environments to detect and remedy situations where capabilities are degraded, such as ML models operating on drifted data, which is sometimes caused by adversarial action (e.g., data poisoning).
- To decrease the feedback-loop timeline, additional research is required to extend the development pipeline to the edge, such as continuous AI and ML model retraining at the edge.

DATA AND DATA ARCHITECTING FOR THE EDGE

The ability to collect and process data at the edge for timely decision making is of high priority for edge systems. Increasing availability of AI solutions provides opportunities for realizing advanced decision-making scenarios at the edge; however, the success of AI-enabled capabilities relies on access to data. Edge environments take each of the five Vs of big data challenges to the extreme: volume, velocity, variety, veracity, and value.[16] Mission uncertainty in edge environments introduces challenges for both creating data architectures that can serve multi-domain missions and architecting edge systems in a manner that incorporates data-related concerns, such as handling different levels of data classification at the edge.

CHALLENGE 1: DATA STANDARDS AND GOVERNANCE DO NOT ENABLE TIMELY LEVERAGE OF TECHNICAL INNOVATIONS.

Edge systems are characterized by their diversity in several dimensions, including but not limited to organizational and technical diversity. On the organizational side, edge systems often need to support multi-organization scenarios that may have different data and mission needs. On the technical side, these multi-organization scenarios are supported by multi-sensor, multi-platform, and heterogeneous hardware. Standards can assist in collecting clean and consistent data, which can also assist with governance. Without those standards, capabilities cannot be developed at speed. In addition, redundant, inconsistent, and incomplete data collection practices introduce an avoidable cost and time burden. Creating data standards is an important step in supporting use cases, such as the ability to quickly define data-access privileges in multi-domain and organizational settings operating at the edge and to support data fusion.[17] Especially today, military edge systems that support different services have different sensors that generate different images of different resolutions, file sizes, aspect ratio, metadata, etc. When data varies, the approaches to support decision making and algorithms to support data analysis vary.

CHALLENGE 2: TOOLS AND TECHNIQUES TO SUPPORT DATA TRANSFORMATION AT THE EDGE NEED BREAKTHROUGH IMPROVEMENTS.

Data transformation is the process of converting, cleansing, and structuring data into a usable format that can be analyzed to support decision-making processes. Each time data is processed through these transformation phases, some aspect of the data may get lost or misrepresented. While standards may help alleviate the data-loss challenge, especially by defining metadata, standards alone are not sufficient to address all data-transformation challenges, especially for edge capabilities where extreme heterogeneity and multi-modal data is the norm. Data transformation pipelines at the edge typically include the following stages: collection, cleansing, prioritization, sharing, and execution. Each of these stages may be executed on different platforms between cloud and edge systems. In addition to designing these pipelines with modularity in mind for scalability and ease of distribution across platforms, as we envision edge systems, we must accept that some data will be lost and therefore we must design capabilities with this assumption. Given storage-cost and scalability concerns, it is critical to develop techniques to: (a) improve decision making regarding what data to store and for what duration, (b) determine where data collection, processing, and analysis should take place, and (c) decide how data is consumed. As new techniques that support and inform data transformation are developed, the fidelity of decision making will increase, while the risk that originates from data loss throughout the data transformation process decreases.

CHALLENGE 3: EDGE SYSTEMS ARE NOT DESIGNED FOR DATA SCALABILITY, INTEGRATION, AND INTEROPERABILITY.

As edge systems become increasingly ubiquitous, the enormous scalability challenges around quantity, filtering, and storage of data are inevitable. To determine how the system can recognize new data, decide to collect data, and identify when to store or when to send data, edge systems must include monitoring capabilities that support data scalability and interoperability, as well as timely updates to models to accommodate critical changes. Edge systems face multidimensional scalability challenges related to data. For example, resources that support computation—including resource needs for AI model training and development—must be scalable. Plus, edge systems require storage solutions to support data at scale. Finally, edge systems must implement tactics to decide what responsibilities should rest at the edge nodes versus in cloud resources. Data integration and interoperability challenges often stem from the heterogeneity and uncertainty of edge environments. Systems need to be architected with an understanding of tradeoffs of strategies, such as common versus distributed storage, middleware-based versus application-based integration, and edge- versus cloud-hosted computation.

Recommendations: For enabling timely and accurate decisions at the edge, edge software systems need to be developed for heterogeneity and multi-modal data processing, considering that scalability, integration, and interoperability challenges will be the norm. Data standards and governance strategies need to be orchestrated along with technical strategies to address these challenges.

Closing Organizational and Policy Gaps

- Data needs to be treated as a priority and an essential artifact of all systems, but especially edge systems. To support such a shift in system ownership and design, initiatives developing data standards and policy need to be accelerated.
- Organizations need to focus their internal efforts on developing up-to-date data strategies that align, challenge, and integrate with other related, relevant data strategies.
- Data standards and policies that appropriately support heterogeneous edge environments will further enable policy-aware development of common capabilities, while still allowing multi-sensor, multi-platform, multi-hardware infrastructures.

Closing Technical Gaps

- Data pipelines that support the collect, cleanse, prioritize, share, and execute stages of data need to be architected both with both scale and uncertainty in mind.
- Research on data scalability, integration, and interoperability for edge systems needs funding focused on developing techniques that incorporate managing uncertainty and heterogeneity.
- Research initiatives that define and develop techniques for policy-aware edge systems need to be launched.

Enabling AI and ML at the Edge

AI and ML capabilities are increasingly being deployed at the edge to support a variety of missions, including improved situational awareness, support for intel analysts, and autonomous systems. While any type of edge software system faces the challenges outlined in the previous sections, solving these challenges is even more important in the context of edge software systems that include AI and ML capabilities. Solving resource adaptation, rapid deployment, and data and data architecting challenges as outlined in this report will contribute to realizing successful tactical AI and ML solutions at the edge.

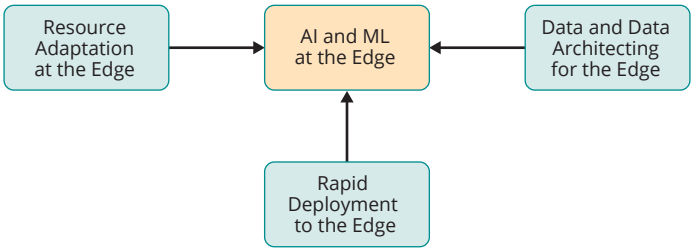


Figure 1: Realizing Successful AI and ML Solutions at the Edge

- **Resource adaptation:** AI and ML components are compute intensive, in many cases requiring specialized hardware such as GPUs to run. Tradeoffs between accuracy and resource consumption are well known in edge systems.[18] A solution for the resource-consumption problem is to distribute AI and ML tasks across edge nodes, which requires efficient computation distribution. In addition, the accuracy of AI and ML components will degrade over time due to data drift and concept drift. Data drift occurs due to differences between training data and production data, while concept drift occurs due to changes in the process that generates data (e.g., adversaries rapidly learn to change the appearance of things so that AI and ML models overlook them—same inputs, different labels). Regardless of the type, drift adds yet another source of uncertainty that needs to be analyzed prior to deployment and then monitored at run time to ensure proper responses to drift detection. Finally, data at the edge needs to be collected and sent to a centralized location for creating datasets for future use, which places yet another demand for already constrained bandwidth.
- **Rapid deployment:** Because AI and ML models tend to drift over time and because they are deployed in dynamic edge environments, these will likely have to be retrained with data collected at the edge and redeployed more often than traditional software components. Tools and infrastructure to support rapid model retraining, efficient and secure deployment to the edge, and closed-loop data collection will be necessary to ensure that models remain relevant to the mission.
- **Data and data architecting:** It is critical to have the right training data and to be able to augment that data with data collected at the edge to ensure continued operation of AI and ML capabilities. The potentially large amount of data collected at the edge along with storage limitations of edge devices will require efficient execution of data collection and prioritization policies and processes to decide: (a) which data is relevant for retraining and which can be discarded, (b) how to label collected data, and (c) which summarization and filtering processes to use to optimize bandwidth for data transmission. The most effective edge AI and ML systems will be those that expertly combine data labeling with data acquisition.

Call to Action

The workshop made clear that we need to make progress on technical, organizational, and policy issues in parallel to realize present and future edge scenarios. Top-priority actions that need immediate attention include: (a) developing techniques to support more adaptive architecting, development, and deployment approaches, (b) improving upon often incorrect communications and resource assumptions at the edge, and (c) addressing the uncertainty that is inherent to operating in tactical edge environments. Teams developing these systems as well as science and technology organizations developing next-generation techniques both need to address these actions.

As our recommendations reveal, the design, development, deployment, and evolution of edge software systems require both organizational and policy changes, as well as investments in research to close technical gaps related to resource adaptation, rapid deployment, and data management at the edge.

Finally, as stated in the DoD software modernization strategy, edge software systems require the development of software talent that understands the levels of uncertainty and adaptation required at the tactical edge.[12]

References

[1] Brose, Christian. *The Kill Chain: Defending America in the Future of High-Tech Warfare.* Hachette Books. 2020. ISBN 9780316533362. **https://www.hachettebookgroup.com/titles/christian-brose/the-kill-chain/9780316533362/**

[2] Lewis, Grace; Echeverría, Sebastián; Simanta, Soumya; Bradshaw, Ben; & Root, James. Tactical Cloudlets: Moving Cloud Computing to the Edge. Pages 1440–1446. In *Proceedings of the 2014 IEEE Military Communications Conference.* October 2014. **https://ieeexplore.ieee.org/abstract/document/6956959**

[3] Echeverría, Sebastián; Lewis, Grace A.; Root, James; & Bradshaw, Ben. Cyber-Foraging for Improving Survivability of Mobile Systems. Pages 1421–1426. In *Proceedings of MILCOM 2015–2015 IEEE Military Communications Conference (MILCOM).* October 2015. **https://ieeexplore.ieee.org/document/7357644**

[4] Echeverría, Sebastián; Klinedinst, Dan; Williams, Keegan; & Lewis, Grace A. Establishing Trusted Identities in Disconnected Edge Environments. Pages 51–63. *In Proceedings of the 2016 IEEE/ACM Symposium on Edge Computing (SEC).* October 2016. **https://ieeexplore.ieee.org/document/7774673**

[5] Echeverría, Sebastián; Lewis, Grace A.; Novakouski, Marc; & Boleng, Jeff. Delay-Tolerant Data Sharing in Tactical Environments. Pages 605–610. In *Proceedings of MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM).* October 2017. **https://ieeexplore.ieee.org/document/8170773**

[6] Lewis, Grace A.; Echeverría, Sebastián; Klinedinst, Dan; & Williams, Keegan. Secure VM Migration in Tactical Cloudlets. Pages 388–393. In *Proceedings of MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM).* October 2017. **https://ieeexplore.ieee.org/document/8170742**

[7] Echeverría, Sebastián; Lewis, Grace A.; Klinedinst, Dan; & Seitz, Ludwig. Authentication and Authorization for IoT Devices in Disadvantaged Environments. Pages 368–373. In *Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT).* April 2019. **https://ieeexplore.ieee.org/document/8767192**

[8] Salehie, Mazeiar & Tahvildari, Ladan. Self-Adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems.* Volume 4. Issue 2. May 21, 2009. Pages 1–42. **https://dl.acm.org/doi/10.1145/1516533.1516538**

[9] Szabo, Claudia; Sims, Brendan; Mcatee, Thomas; Lodge, Riley; & Hunjet, Robert. Self-Adaptive Software Systems in Contested and Resource-Constrained Environments: Overview and Challenges. *IEEE Access.* Volume 9. December 9, 2020. Pages 10711–10728. **https://ieeexplore.ieee.org/document/9288789**

[10] Microsoft Corporation. The STRIDE Threat Model. *Microsoft.com.* November 12, 2009. **https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)**

[11] Schneier, Bruce. Attack Trees. *Dr. Dobb's Journal.* Volume 24. Number 12. December 1999. Pages 21–29. **https://www.schneier.com/academic/archives/1999/12/attack_trees.html**

[12] Department of Defense. *Department of Defense Software Modernization Strategy.* Version 1.0. November 2021. **https://media.defense.gov/2022/Feb/03/2002932833/-1/-1/1/DEPARTMENT-OF-DEFENSE-SOFTWARE-MODERNIZATION-STRATEGY.PDF**

[13] Ferguson, Ian. The Role of DevSecOps in Modern Edge Systems. *Mobility Engineering.* December 1, 2022. **https://www.mobilityengineeringtech.com/component/content/article/adt/pub/features/articles/47194**

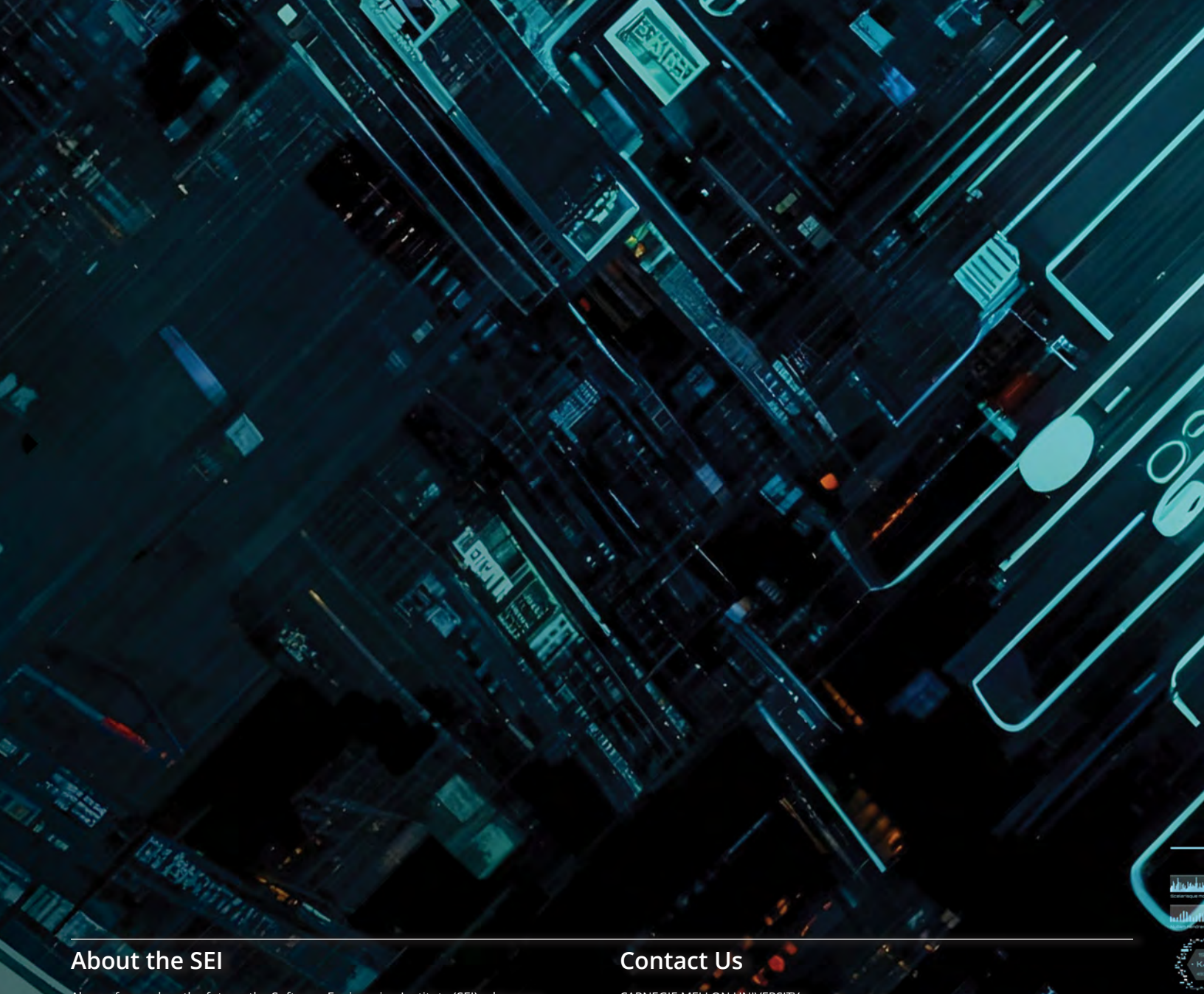
[14] The White House. *Executive Order on Improving the Nation's Cybersecurity.* EO 14028. May 12, 2021. **https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/**

[15] Office of Management and Budget. *Moving the U.S. Government Toward Zero Trust Cybersecurity Principles.* M-22-09. Executive Office of the President. January 26, 2022. **https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf**

[16] Yates, Christopher. How Military Teams Can Optimize Data, Decision-Making at Tactical Edge. *C4ISRNET.* October 19, 2022. **https://www.c4isrnet.com/it-networks/2022/10/19/how-military-teams-can-optimize-data-decision-making-at-tactical-edge/**

[17] Department of Defense. *DoD Data Strategy: Unleashing Data to Advance the National Defense Strategy.* September 30, 2020. **https://media.defense.gov/2020/Oct/08/2002514180/-1/-1/0/DOD-DATA-STRATEGY.PDF**

[18] Chen, Jiasi & Ran, Xukan. Deep Learning with Edge Computing: A Review. *Proceedings of the IEEE.* Volume 107. Issue 8. August 2019. Pages 1655–1674. **https://ieeexplore.ieee.org/document/8763885**



About the SEI

Always focused on the future, the Software Engineering Institute (SEI) advances software as a strategic advantage for national security. We lead research and direct transition of software engineering, cybersecurity, and artificial intelligence technologies at the intersection of academia, industry, and government. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and are based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering.

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100

DM23-0437