Software Systems at the Edge: The SEI Portfolio

Kevin Pitstick

kapitstick@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

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.

DM22-1155

Why Focus on Edge Software Systems?

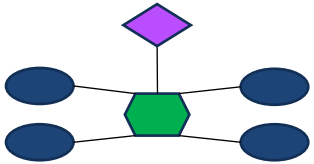
The number of devices that are connected to each other, cloud, and enterprise platforms, and the volume of data being produced by these devices are growing exponentially along with DoD future scenarios that rely on these capabilities.

Operating at the edge of the connected network, close to where data and computation are needed provides an approach to alleviate resource and computation challenges.

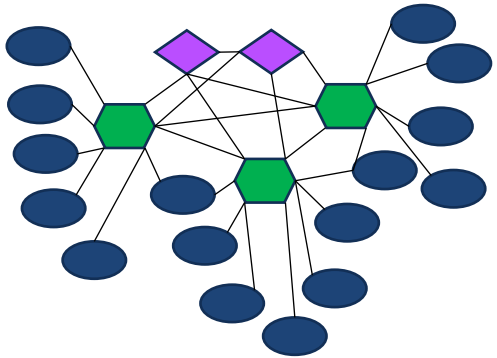
Edge software systems are often part of larger distributed systems where there is interaction between cloud nodes and edge nodes, with edge nodes serving as intermediaries between proximate users and the cloud

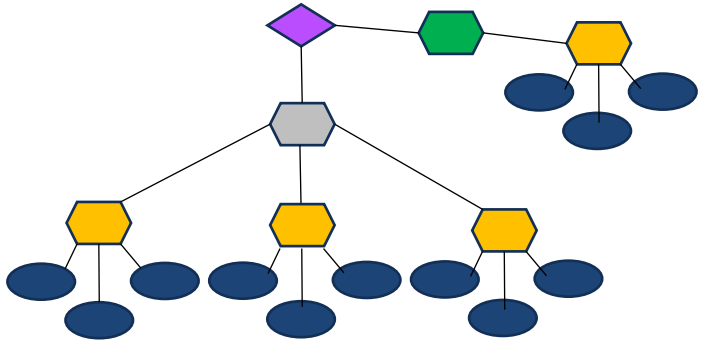
- *Cloud-to-edge continuum* is a term used to describe the interaction between cloud and edge nodes in which data and computation move between them as needed

Cloud to Edge Continuum - Examples



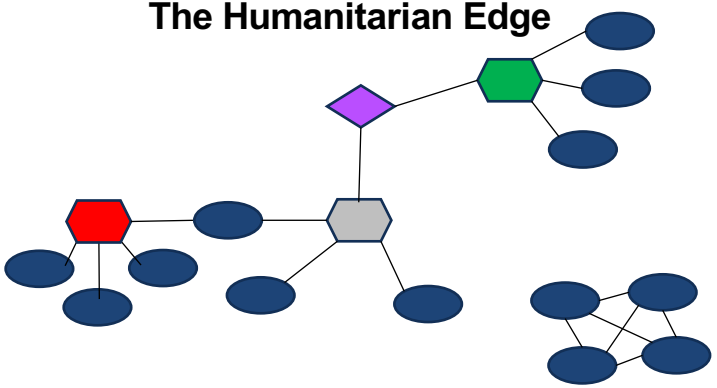
The Personal Edge



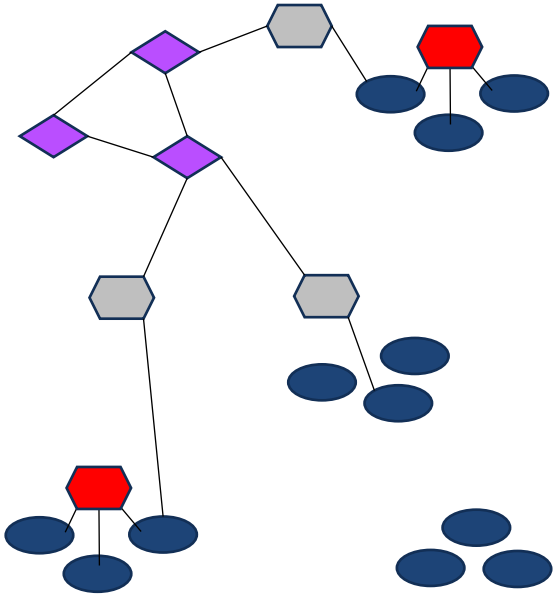


The Industrial Edge

The Humanitarian Edge



The Tactical Edge



Legend

- Commercial Infrastructure
- Industrial Infrastructure
- Mesh Network Radios
- Connection
- Edge Device
- Cloud Server
- Satellite Infrastructure

Challenges for DoD Edge Systems

Higher levels of uncertainty

- Systems operate in what is known as DIL (disconnected, intermittent, limited) environments
- Systems operate in environments with active attackers
- Systems need to adapt to changing missions and dynamic environments

Shorter timeframes for decision making

- Decisions need to be made with limited latency, regardless of state of connections to greater resources
- Data must be transformed into information and presented to decision makers in a timely manner (systems or operators)

Requirements for DoD Edge Systems

Systems at the tactical edge need to be

- network-aware
- location-aware
- resource-aware
- mission/environment-aware
- secure by construction (with active monitoring)
- highly modular to support computation distribution

Main driver for how to meet requirements is amount of uncertainty

Different levels of uncertainty require different tactics to overcome uncertainty, e.g., a system that operates in a 70% uncertain network environment will be architected different from one that operates in a 30% uncertain network environment

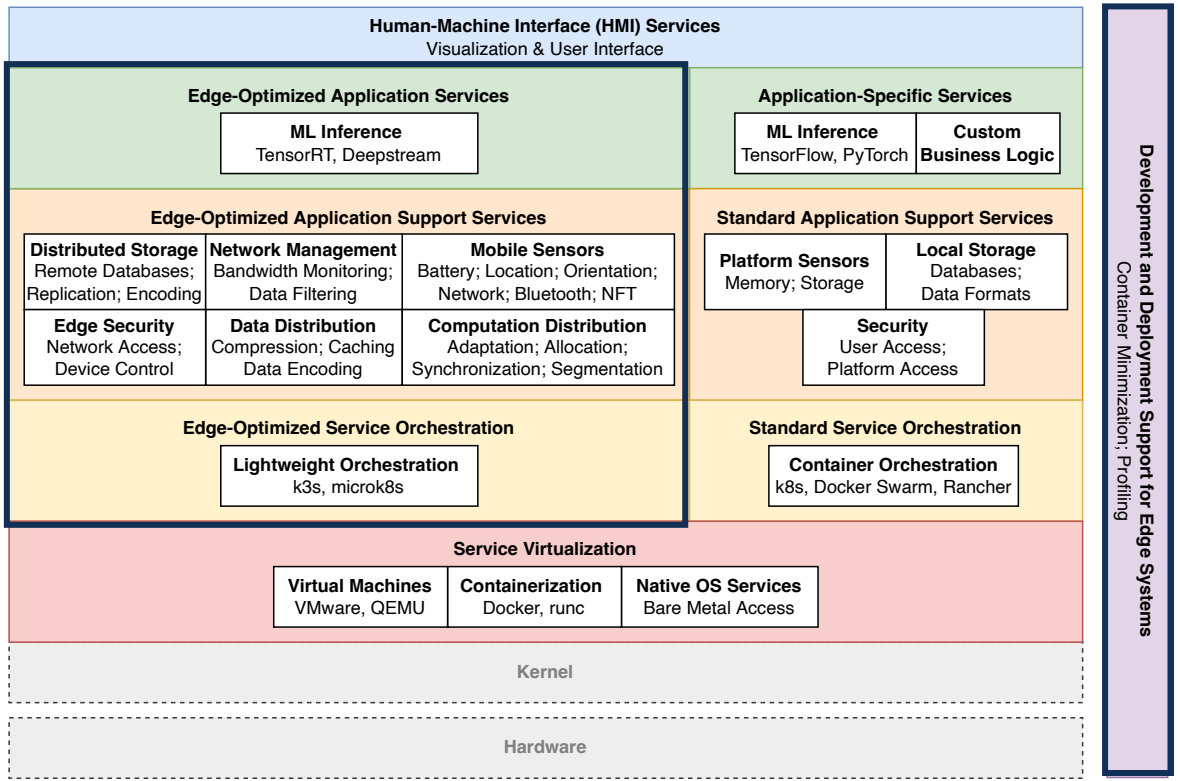
Data deluge continues to be a problem

As edge devices become more powerful, they can handle a larger number of connected sensors; however, the larger amounts of data collected cannot be properly managed at the edge

SEI Work in Software Systems at the Edge

We bring the much needed attention to software:

- define and improve the **software stack for systems** that operate at the edge
- define and improve **tools for development and deployment** of edge software systems



Example DoD Applications

Designing an open architecture for the tactical edge to support sensor collection/fusion and AI/ML analytics

- Supports both arm64/x86 Docker services connected with ROS2 messaging

Integrating AI/ML Capabilities into of Distributed Edge Processing Pipelines

- Audio: Transcription, Translation
- Video: Object Recognition (People/Vehicles); Facial Recognition, Car Type/Make/Model Classification, License Plate Detection/Recognition

Developing embedded software (ESP32 microcontroller) for surveillance / sensor collection (Bluetooth/WiFi sniffing, image capture, GPS)

- Data reported to AWS for aggregation/fusion with web UI

Developing an infrastructure for rapid mission adaptation at the edge

- Supports composition, deployment, monitoring, and adaptation of information services into data processing pipelines that can be distributed across cloud, traditional server and embedded hardware platform

SEI Edge Software System Research Portfolio

Network Awareness
ISE: Group Context-Aware Mobile Applications
High-Assurance Software-Defined IoT Security
Delay-Tolerant Data Sharing

Resource Awareness
Tactical Cloudlets
Enabling Object Detection at the Edge
Automating Container Minimization at the Edge

Location Awareness
ISE: Group Context-Aware Mobile Applications

Security by Construction
Tactical Cloudlets
Authentication and Authorization in Edge IoT Environments
High-Assurance Software-Defined IoT Security

Modularity
Tactical Cloudlets

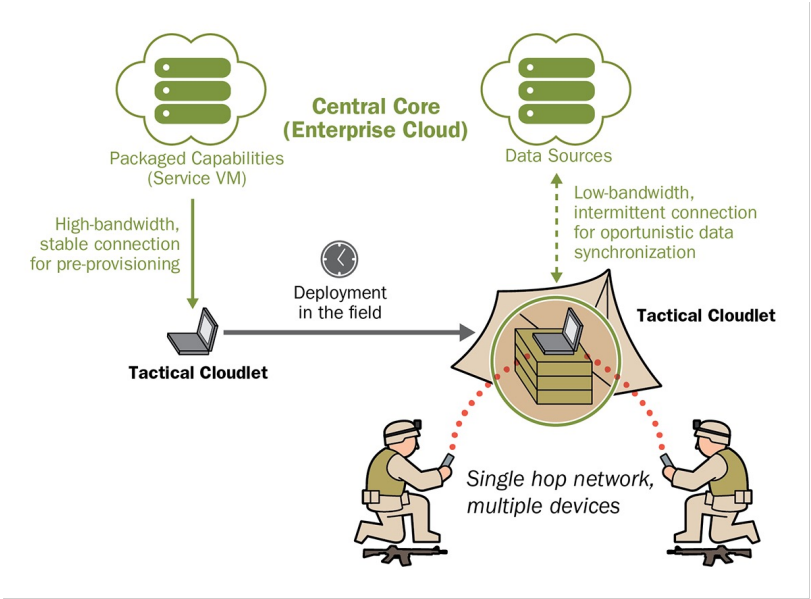
Tactical Cloudlets

Objective: To “carve out” a piece of the cloud and make it accessible to personnel operating in tactical environments

Challenge: Provide these capabilities in a secure, reliable, and timely manner, especially in fully disconnected environments

Solution: Forward-deployed, discoverable, computing nodes that can be hosted on vehicles or other platforms to provide

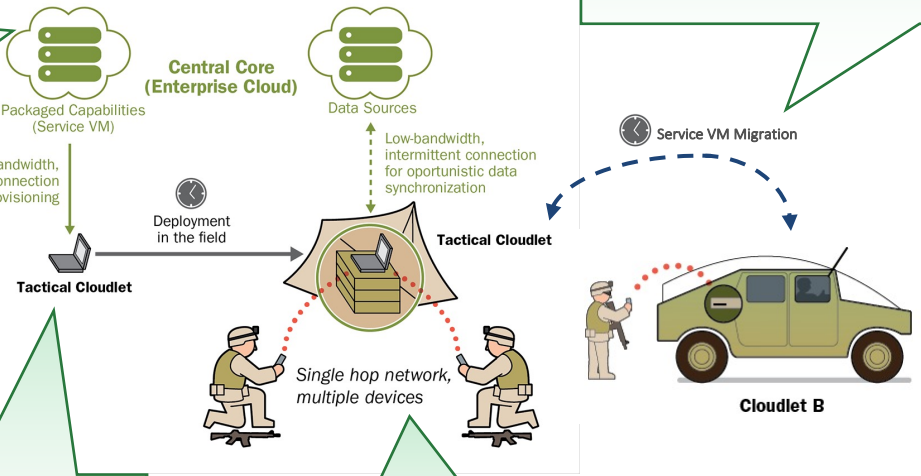
- infrastructure to offload computation
- forward-data-staging for a mission
- data filtering to remove unnecessary data from streams intended for mobile users
- collection points for data heading for enterprise repositories



Tactical Cloudlet Features

Standard Packaging of Services to support ease of provisioning, scalability, and reuse

Service Migration enables mobility and reintegration to a fixed infrastructure



Pre-Provisioned Cloudlets with App Store to support disconnected operations

Cloudlet Management Component provides console-like capabilities and credential management

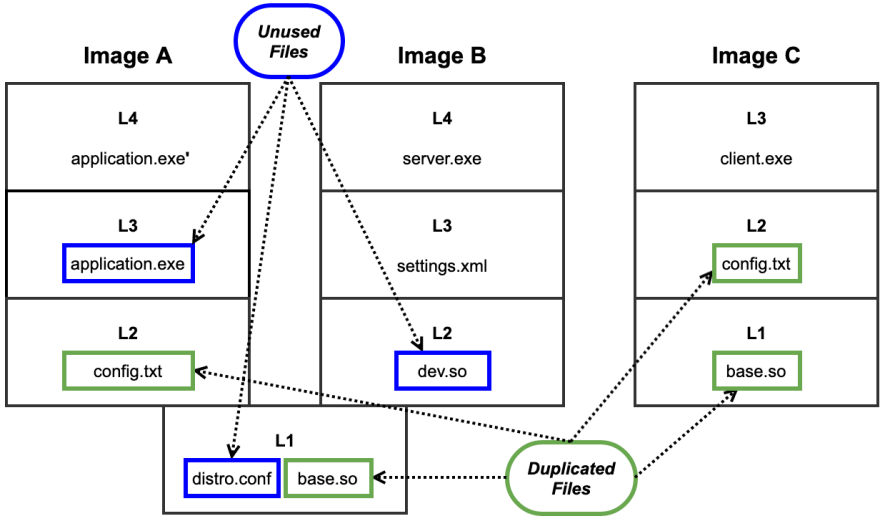
Secure Key Generation and Exchange for establishing trust between cloudlets and mobile clients

Automating Container Minimization for the Edge

Objective: To develop an automated technology that minimizes the storage size of a set of container images which can be used pre-deployment at the tactical edge.

Challenge: Container images are commonly larger (10X or more) than necessary. Resource limitations at the tactical edge conflict with storage waste in container images.

Solution: Create a tool using open source container tools (e.g., Kubernetes) to both prune unused files and deduplicate common files across multi-container systems.



Sources of Storage Waste

- *Unused Files*
  - Development files (dev.so)
  - Unused distro files (distro.conf)
  - Overwritten files (application.exe)
- *Duplicated Files*
  - Same files stored in multiple layers

Enabling Aerial Object Detection at the Edge

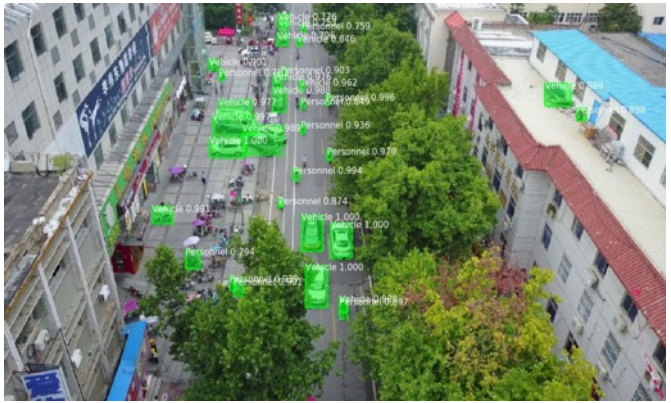
Objective: To demonstrate real-time object detection algorithm for unmanned aerial vehicle (UAV) video on constrained, low-power edge platforms.

Challenge: Edge devices have limited resources (e.g., CPU, RAM, GPU) that pose challenges for deploying resource-intensive neural network models.

Solution: Develop a compression / acceleration pipeline for object detection models and validate it with UAV imagery on Jetson edge devices.

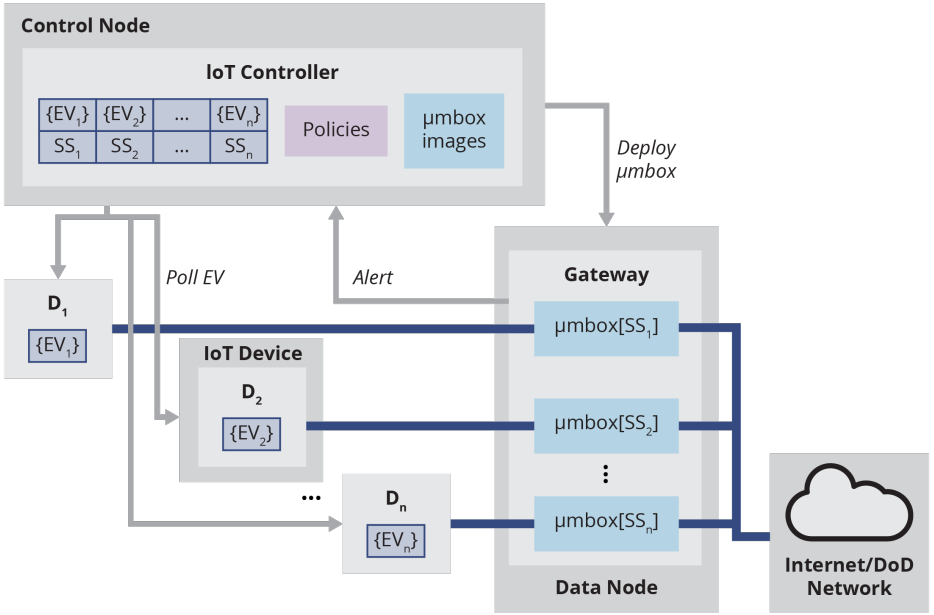
Compression/Acceleration Techniques

- Manual pruning of CNN layers
- Automated pruning of CNN filters
- Student-teacher training / knowledge distillation
- TensorRT inference engine w/ quantization



KalKi: Software-Defined IoT Security Platform

Objective: To enable secure integration of commodity (not fully-trusted) IoT devices into DoD networks



Challenge: Adapting security posture at runtime based on both network and physical threats

Solution: Develop a platform that uses a shared state space for IoT devices to create a richer set of security policies.

- Reacts using **network and environment** information
- Uses **different network defenses** for each device and state
- Adapts to **device-specific vulnerabilities** or limitations

The Future of Software Systems at the Edge

Edge systems are at the heart of the future of warfighting.

- “This is about dramatically increasing the speed of information sharing and decision making in a contested environment to ensure we can quickly bring to bear all our capabilities to address specific threats.” — Gen. Mark Milley, Chairman of the Joint Chiefs of Staff

Some key software challenges for the future will be

- Adapting computation across multiple nodes as resources change
- How to leverage AI/ML for fast, real-time decision making
- Managing and processing the immense amount of data available
- Dealing with compatibility and interoperability as hardware diversity increases
- Securing software systems (by construction and active monitoring) amidst contested environments