Centralized vs. Decentralized Security for Space AI Systems? A New Look

Noam Schmitt
Orange Research
noam.schmitt@orange.com

Marc Lacoste
Orange Research
marc.lacoste@orange.com

***Abstract*—This paper investigates the trade-off between centralized and decentralized security management in constellations of satellites to balance security and performance. We highlight three key AI architectures for automated security management: (a) centralized, (b) distributed and (c) federated. The centralized architecture is the best option short term, providing fast training, despite the hard challenge of the communication latency overhead across space. Decentralized architectures are better alternatives in the longer term, providing enhanced scalability and security.**

I. Security of Satellite Systems

New Space is a new frontier where no AI nor Cloud had much gone before... At least until very recently. Among many: a frenzy of initiatives on future large-scale infrastructures and non-terrestrial networks (e.g., launching multiple satellite constellations, space cloud computing) from multiple competing stakeholders; stringent constraints on communications; multiple new threats and significant momentum to develop automated security solutions using AI. All those elements crystallize into an emerging new field: *Space AI Security*, to detect, protect and respond to threats [1].

Yet, a fundamental question remains: **which is the "right" AI architecture to manage security in such space systems: centralized or partly/fully decentralized?** With which trade-offs for security, performance (latency, scale) and utility?

The simplest design is based on a *large central model*. This choice was deemed not practical due to the limited contact time between satellite and ground station, low bandwidth and high latency without inter-satellite links (ISL). Decentralized design alternatives around distributed variants of *Federated Learning* (FL) were then investigated as more viable to perform training or inference in space and viewed as more scalable and secure [2]. Recently, advances in satellite technologies such as ISL communications – and increasingly bandwidth – have opened new possibilities. Centralized architectures seem to be making a comeback, and are now adopted by many major industry constellation systems providers. What then of the centralized vs. decentralized alternative?

In this paper, we take a closer look and distinguish 3 key AI architectures for automated security management: (a) centralized, (b) distributed, and (c) federated, highlighting benefits and limitations. We then explore on a case study the centralized vs. decentralized security architecture trade-off regarding accuracy and latency.

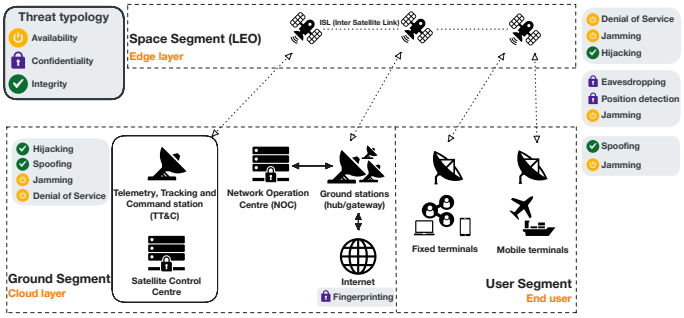

Fig. 1. Satellite constellation: architecture and main threats

Architecture. Figure 1 gives an overview of the main entities in a satellite constellation. The **User** segment includes the end users of the service or system *tenants*, deployed on fixed and mobile terminals with different risks. The **Space** segment includes all satellites in orbit, connected via links to the User segment, the Ground segment and neighboring satellites. They are typically deployed in a Low Earth Orbit (LEO). Some operators (e.g., OneWeb) use higher altitude orbits to make the constellation more dynamic depending on connection and load. The **Ground** segment includes *Ground stations (GS)* distributed across the planet and connected to the Internet to manage the satellite payload. The constellation is supervised by the Satellite Control Center, which notably manages orbits.

Threats. *Confidentiality* threats include *position detection* or *user tracing* specific to the satellite network architecture. *Eavesdropping* deals with risks of interception and spying on communications. *Integrity* threats include *spoofing*, e.g., to usurp a signal, a location or an end-point and *hijacking*. The main *availability* threat remains *Denial-of-service*, in forms common to virtualized infrastructures or specific to LEO constellations. Mitigations include anomaly detection, packet filtering and firewalling. *Jamming*, in base stations or satellites, is another potent threat to disrupt communications. To prevent, detect, and mitigate those threats, a wide range of counter-measures are available [1].

II. Space AI Security Architectures

Satellites face stringent embedded constraints (e.g., response time, hardware architecture, energy efficiency). For risk mitigation, deployment of counter-measures must be automated and orchestrated dynamically. Three options are available for a distributed AI security architecture (see Figure 2).

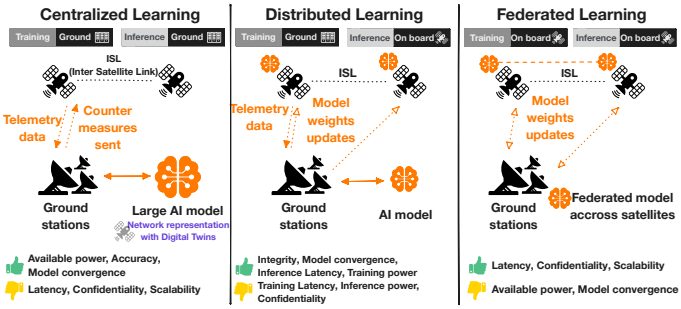

Fig. 2. Space AI architectures: (a) centralized, (b) distributed, (c) federated

(a) Centralized Architecture. A central AI model is deployed on the ground for training and inference. The model learns from satellite telemetric data sent from space. The ground system behaves as a digital twin of the constellation [3]. It triggers the required counter-measure updates, then sent back.

Deploying large models results in fast training and inference: unbounded computing and data processing are available. Additionally, there are no embedded barriers, e.g., for energy efficiency nor need for additional AI hardware on-board. Threat response remains slow due to large latencies across space between telemetric data sending and security update. The attack surface is wide and scalability not guaranteed.

(b) Distributed Architecture. Satellites send telemetric data to the GS for model training as in (a), but receive weights for on-board model update. Inference is performed on-board.

On-board inference lowers latency of the security response. The system is more protected than in design (a) against integrity threats, although confidentiality threats remain. Embedded constraints may nonetheless lower model accuracy. Scalability may also remain difficult at training time.

(c) Federated Architecture. Training and inference are performed on board, without sending telemetric data [2]. Gradients are shared with other satellites and GS for aggregation.

This class of architecture has the lowest latency. It is more secure than centralized designs (a) and (b). It should also be highly scalable up and out, notably, being embedded by design. AI accuracy might be a challenge due to smaller models operating under embedded hardware and energy constraints. Model convergence and satellite synchronization might also be issues for large constellations. Federated learning specific threats should also be addressed.

III. Experimental Results

We compare centralized vs. decentralized designs, in terms of *accuracy* and *performance* during *training* and *inference*. All experiments are run in Python on the CIFAR-10 dataset and ResNet20 model. FL simulations with IID partitioning use the Flower library. Latencies are computed using the StarPerf constellation simulator on Kuiper satellites with ISL.

Training: Accuracy. Figure 3 shows accuracy results for centralized and different FL configurations. Centralized learning converges faster than FL. Time-to-accuracy for FL scales well with the number N of satellites – only X13 overhead for a X50 satellite increase for a centralized baseline. Aggregation optimizations have minimal impact on accuracy.

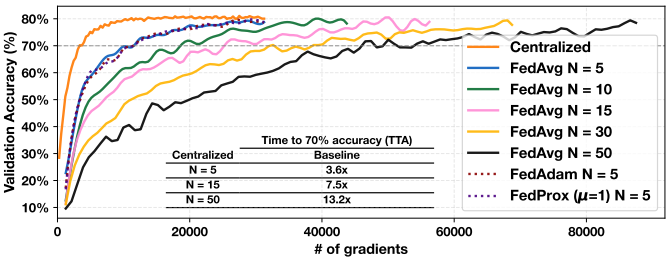

Fig. 3. Training accuracy: centralized vs. federated architectures

Inference: Latency. Table I shows inference results. Federated latency is highly scalable, independent from N. Centralized latency includes a hard physical lower-bound (RTT) and the GS model latency. How fast this last term increases with N depends on the parallelism level (α) in the central server implementation. Scalability is thus more challenging. Overall, RTT in space induces a heavy penalty on the response time.

TABLE I
INFERENCE LATENCY RESULTS (↓ IS BETTER)

# Sat.	Avg Lat. – Centralized (ms)	Avg Lat. – Federated (ms)
1	125.64 [124.20 + 1.44]	23.75
10	125.64 – 134.28	
100	138.60 – 225	

Cent. Avg. Lat. = RTT (124.2ms) + $\alpha N \times$GS Inf. Lat. (1.44ms) with $\alpha = 0.1$ to 0.7
Testbed configuration: Desktop PC, CPU: AMD Ryzen 7 7800X3D, 32GB RAM, GPU: Nvidia RTX 4090, 24GB VRAM, WSL. 128 batch requests per satellite.
Centralized & fed. inference lat.: PyTorch CUDA (GPU) & CPU devices respectively.

Discussion and Next Steps. Preliminary results show centralized designs tend to be the best option in the short term, security set aside, with faster training, but a hard physical lower bound on latency which requires distributed designs to go lower. In the longer term, decentralized designs might be preferable: training, while slower, yields much lower latencies for inference. However, such designs have clear benefits in terms of scalability at training and inference.

Next, we intend to assess more in-depth security considering other threats such as tenant isolation [2]. Digital twin [3] designs and split learning [4] are promising to enhance security, respectively to place additional relevant counter-measures in the introduced mediation layer, and to avoid sending raw telemetric data by sharing inference between satellite and GS. Considering the imbalance between public vs. siloed data in terms of data volume, another promising avenue could be to explore decentralized learning approaches that share only a fraction of the model and yet enhance security [5]. Further experiments should also include more representative testbeds and datasets, with bigger models and HPC resources, closer to realistic large-scale space AI tasks.

References

[1] European Union Agency for Cybersecurity, *LEO satcom cyber security assessment – February 2024*, 2024. [Online]. Available: https://data.europa.eu/doi/10.2824/317096

[2] Z. Lin et al., "FedSN: A Federated Learning Framework Over Heterogeneous LEO Satellite Networks," *IEEE Transactions on Mobile Computing*, vol. 24, no. 3, pp. 1293–1307, 2025.

[3] P. Ferreira et al., "Reinforcement Learning for Satellite Communications: From LEO to Deep Space Operations," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 70–75, 2019.

[4] J. Sun et al., "An Efficient Privacy-Aware Split Learning Framework for Satellite Communications," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 12, pp. 3355–3365, 2024.

[5] S. Biswas et al., "Noiseless privacy-preserving decentralized learning," *Privacy Enhancing Technologies*, 2025.