Poster: Challenges in Applying COTS Secure, Resilient Boot and Update Capabilities for Space Systems

Gabriel Torres*, Raymond Govotski *, Samuel Jero *, Gruia-Catalin Roman†, Joseph "Dan" Trujillo‡, Richard Skowyra*, Samuel Mergendahl*

* *MIT Lincoln Laboratory*
Secure Resilient Systems & Technology
Lexington, MA
{gabriel.torres, raymond.govotski, samuel.jero, richard.skowyra, samuel.mergendahl}@ll.mit.edu

† *University of New Mexico*
Department of Computer Science
Albuquerque, NM
gcroman@unm.edu

‡ *Air Force Research Laboratory*
Space Vehicles Directorate
Albuquerque, NM
joseph.trujillo.4@spaceforce.mil

***Abstract*—As space systems increasingly leverage commercial-off-the-shelf (COTS) technology to reduce mission cost and facilitate faster deployment timelines, satellite systems must also consider the cybersecurity achievable from the commercially available technology prior to adoption. In this work, we explore the challenges and trade-offs in applying COTS technology for satellite systems. In particular, we introduce a generic, high-level architecture for secure, resilient boot and update typically required to achieve an appropriate cybersecurity posture onboard a satellite. Moreover, we explore the challenges we encountered when instantiating this architecture on three generations of COTS technology. Namely, we find that COTS systems often provide secure, resilient boot and update capabilities, but the cost benefits of COTS technology often come with inflexibility which leads space system architects to choose between either limited suitability of COTS deployment for their specific space mission needs or expensive extensions to the COTS platform.**

I. Introduction

Space systems offer many critical capabilities for society. For example, weather forecasting [1], guidance and navigation [2], communications [3], broadcasting [4], and financial transactions dependant on accurate timestamps [5] all often rely on space systems. In order to facilitate faster timelines and reduce mission cost for these critical infrastructures, a growing trend in the deployment of space systems is to leverage commercial-off-the-shelf (COTS) technology. For example, many space systems now rely on commercial satellite launches [6] with hardware components outsourced as COTS [7].

However, given their designation of critical infrastructure, the cybersecurity of satellite systems must be considered. While satellite security has traditionally focused on the protection of communication links [8], more expansive threat models have also been proposed [9]. For example, the London School of Economics investigated the fallout of losing these constellations, and found that in under two weeks, we would experience a slow economic collapse, grounding of flights, power grid disruptions, fresh water shortage, and more [10]. They concluded that "in contrast to military satellites which are commonly designed so that the security aspect is taken into account, commercial satellites are more vulnerable to attacks because of a lack of awareness and implementation of security. Often, manufacturers of satellites use off-the-shelf technology to make the costs more reasonable. Some of these components can be screened by hackers for vulnerabilities in open-source technology and software."

Moreover, as satellites are increasingly comprised of COTS components, space system architects must consider that these components were often originally designed for terrestrial use cases. For example, satellite systems are much more remote with bandwidth (and delay) constrained communication links [11] and will be subject to environmental effects such as radiation-induced faults [12]. In this study, we enumerate a number of challenges in applying the capabilities offered by COTS platforms to the cybersecurity of a satellite system. In particular, we aim to instantiate secure, resilient boot as well as secure, resilient update functionality for a satellite using the capabilities found on three hardware platforms from one commercial vendor. We find that we can indeed instantiate our desired security architecture using these COTS components, but encounter a number of challenges in their deployment for space systems.

This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-B002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Air Force.

Workshop on Security of Space and Satellite Systems (SpaceSec) 2026
23 February 2026, San Diego, CA, USA
ISBN 978-1-970672-02-2
https://dx.doi.org/10.14722/spacesec.2026.23091
www.ndss-symposium.org

II. Desired Security Architecture

In this section, we describe our high-level, target architecture for secure, resilient boot and update. This functionality is one (but not the only) part of a proper cybersecurity posture for satellite software. While our proposed architecture follows standard design for secure and resilient boot and update, we hope this high-level architecture can help guide future space missions with a roadmap on how instantiate specific capabilities offered by different COTS vendors.

A. Secure and Resilient Boot

Our system's Root of Trust (RoT) stems from an immutable zero-stage bootloader (ZSBL) found in the BootROM that securely loads the next valid boot image. The ZSBL starts the image found in the primary boot storage device. This image contains multiple partitions: a first-stage bootloader (FSBL) and the primary system image. The ZSBL authenticates and decrypts the FSBL partition of the boot image, whereas the FSBL authenticates and decrypts the primary system image partition. Optionally, the FSBL may extend the boot chain-of-trust by instead loading a second-stage bootloader (SSBL). In turn, the SSBL starts the primary system image and other application code found in a third storage device. Each bootloader uses keys stored on the platform. Asymmetric keys for authentication are found in each partition's boot image header with their hash stored in immutable eFuses to validate the key's integrity. Symmetric keys for decryption are stored in battery-backed random access memory (BBRAM) unencrypted and inaccessible after boot. A boot image partition is considered invalid when a bootloader fails decryption or authentication, either due to malicious corruption or environmental effects. Should boot checks fail, a system register that identifies the next valid boot image is set, and the system is reset to alert the ZSBL to load the next valid image. Multiple images can be stored in the primary and alternate storage devices for increased resiliency to errors, and the system reset and registers to control the next valid boot image location are exposed to the application code.

B. Secure and Resilient Update

The process begins with the ground station transmitting an encrypted and signed image (with the keys provisioned on the platform for secure boot) to the flight software that stores the update in a temporary staging area. The image and metadata are uploaded in chunks where each chunk consists of 4092 bytes and a 4-byte cyclic redundancy check (CRC). This allows the system software to store each chunk and CRC on a page in the staging area. Should a chunk of data become corrupted in transit, on storage, or fail to arrive, the system can efficiently request only a specific chunk be resent. Some entries in the metadata include the update size and the CRC of the entire update. Once the update has been fully processed, the ground station transmits the command to finalize the update, then the system software modifies the exposed registers to alert the ZSBL to boot into the update firmware located in either the primary or alternate boot device.

Once booted, the update firmware checks for a pending update. If there is a pending update, it validates the integrity of the update by computing its CRC. Next, it checks the boot device for an empty update slot. Each boot device supports a modified A/B update pattern, where there are many "A" slots that represent valid images and at least one "B" slot for a pending update. The updater places the image in the open slot, taking care to verify the integrity of the data as it is moved. After storing the update in the open slot, it verifies the integrity of the entire file once more. To finalize the update, it sets an `UPDATE_PENDING` flag in the temporary staging area's metadata page and boots into the new image.

If the update is corrupted and fails to boot, the platform's multi-boot feature will load the next configured, valid image. Similarly, if the new image successfully boots but hangs from an error, a hardware watchdog may trigger a reset to load a different image. After an image successfully boots, the ground station can monitor the satellite's behavior and determine whether the update was successful or if an old image was loaded. If the update is successful, the update flag, staging area, and old boot image are cleared.

III. Challenges Deploying on COTS Platforms

In this section, we describe challenges we encountered applying our architecture from Section II on multiple COTS platforms. We instantiate three platforms from AMD/Xilinx into our architecture: Zynq 7000, Zynq Ultrascale+ MPSoC, and AMD Versal, and enumerate challenges a mission may encounter in their adoption. In the future, we plan to extend to other vendors and platforms.

1) **Implicit Vendor Trust:** A system must implicilty trust that the vendor's implementation of the hardware RoT is free of bugs. In particular the ZSBL in the BootROM is often propriety and unobservable.
2) **Rigid Security Features:** Platforms often consist of fixed algorithms and security features potentially missing needed mission requirements (e.g., out-of-date cryptographic algorithms).
3) **Dependence on Vendors:** Users must wait until vendors release new devices to fulfill mission requirements or spend costly effort to extend these platforms themselves.
4) **Inheriting Shortcomings:** Devices such as flash storage have proprietary implementations of safety critical behavior that can leave engineers with no choice but to implement their own in-order to guarantee known, safe behavior.

IV. Conclusion

In this work, we investigated the challenges in applying COTS technology for satellite systems. Specifically, we explored secure, resilient boot and update mechanisms required for space missions and proposed a high-level architecture to meet the needed cybersecurity posture for satellite systems. We found that the available capabilities on COTS platforms can provide secure, resilient boot and update, but often come with inflexibility that may dilute their cost effectiveness.

References

[1] N. US Department of Commerce. (2024, Apr) NOAA-National Weather Service Satellites. [Online]. Available: https://www.weather.gov/marine/wxsat

[2] "Satellite Navigation – GPS – How It Works | Federal Aviation Administration," Nov. 2025, [Online; accessed 13. Nov. 2025]. [Online]. Available: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks

[3] "Satellite Communications | NOAA / NWS Space Weather Prediction Center," Nov. 2025, [Online; accessed 13. Nov. 2025]. [Online]. Available: https://www.swpc.noaa.gov/impacts/satellite-communications

[4] G. M. Drury, "Broadcasting by satellites," *International Journal of Satellite Communications*, vol. 12, no. 4, pp. 395–417, 1994. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.4600120407

[5] V. Yodaiken, "Clock Synchronization In Finance And Beyond," 2017.

[6] A. V. Dolgopolov, P. M. Smith, T. Stroup, C. B. Christensen, J. Starzyk, and T. Jones, *Analysis of the Commercial Satellite Industry, Key Indicators and Global Trends*, 2020. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2020-4244

[7] S. Spremo, M. C. Lindsay, P. D. Klupar, and A. J. Swank, "Cost Optimization and Technology Enablement COTSAT-1," in *Small Satellites Systems and Services 4S Symposium*, no. ARC-E-DAA-TN1656, 2010.

[8] L. Yu, J. Hao, J. Ma, Y. Sun, Y. Zhao, and B. Luo, "A Comprehensive Analysis of Security Vulnerabilities and Attacks in Satellite Modems," in *ACM Conferences*. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 3287–3301.

[9] J. Willbold, M. Schloegel, M. Vögele, M. Gerhardt, T. Holz, and A. Abbasi, "Space Odyssey: An Experimental Software Security Analysis of Satellites," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 1–19.

[10] W. Peeters, "Cyberattacks on Satellites." [Online]. Available: https://www.lse.ac.uk/ideas/projects/space-policy/publications/Cyberattacks-on-Satellites

[11] B. Elbert, *The satellite communication ground segment and earth station handbook*. Artech House, 2014.

[12] R. Ecoffet, "Overview of in-orbit radiation induced spacecraft anomalies," *IEEE Transactions on Nuclear Science*, vol. 60, no. 3, pp. 1791–1815, 2013.

Challenges in Applying COTS Secure, Resilient Boot and Update Capabilities for Space Systems

Gabriel Torres[1], Raymond Govotski[1], Samuel Jero[1], Gruia-Catalin Roman[2], Joseph "Dan" Trujillo[3], Richard Skowyra[1], Samuel Mergendahl[1]
[1]MIT Lincoln Laboratory, [2]University of New Mexico, [3]Air Force Research Laboratory Space Vehicles Directorate

Motivation

- Space systems are critical infrastructure that must protect against cyber attacks
- Custom space solutions are expensive and face long launch timelines compared to ubiquitous COTS alternatives
- The security of COTS components may be less extensive than custom solutions
- Most COTS components are designed for terrestrial systems and don't account for the nuances of space

Desired Architecture Goals

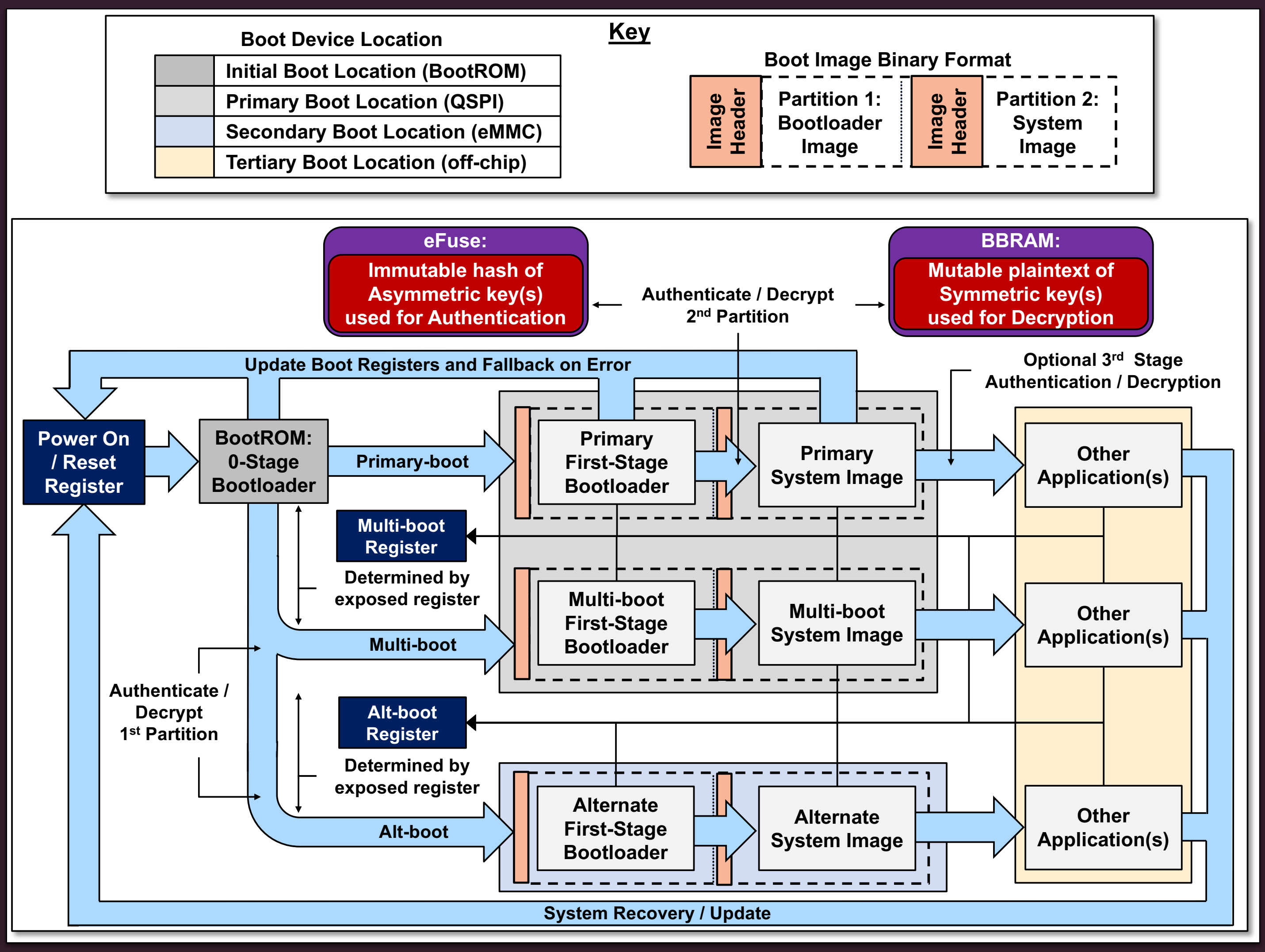
Security: Boot and update procedures must maintain confidentiality and integrity as well as only accept authorized images

Resiliency: Both procedures must properly handle environmental faults and operate under constrained satellite links

Performance: The architecture must not limit nominal performance as to not interfere with the safety of the mission

Cross-Platform Applicability: Architecture must be able to be instantiated by multiple platforms

Secure, Resilient Boot Design


Key
Boot Device Location
Initial Boot Location (BootROM)
Primary Boot Location (QSPI)
Secondary Boot Location (eMMC)
Tertiary Boot Location (off-chip)
Boot Image Binary Format
Image Header
Partition 1: Bootloader Image
Image Header
Partition 2: System Image
eFuse:
Immutable hash of Asymmetric key(s) used for Authentication
Authenticate / Decrypt 2nd Partition
BBRAM:
Mutable plaintext of Symmetric key(s) used for Decryption
Update Boot Registers and Fallback on Error
Optional 3rd Stage Authentication / Decryption
Power On / Reset Register
BootROM: 0-Stage Bootloader
Primary-boot
Primary First-Stage Bootloader
Primary System Image
Other Application(s)
Multi-boot Register
Determined by exposed register
Multi-boot
Multi-boot First-Stage Bootloader
Multi-boot System Image
Other Application(s)
Authenticate / Decrypt 1st Partition
Alt-boot Register
Determined by exposed register
Alt-boot
Alternate First-Stage Bootloader
Alternate System Image
Other Application(s)
System Recovery / Update

Secure, Resilient Boot

- Root of Trust (RoT) starts with an immutable Zero-Stage Bootloader (ZSBL)
- A secure chain-of-trust is created between the ZSBL, First-Stage Bootloader (FSBL), and eventually the primary system image

Secure Update and Recovery

- System can fallback into multiple alternative images
- Multiple boot images facilitates system updates that follow an A/B update pattern

Impact

Understand the limitations of deploying secure, resilient boot and update on space systems using COTS platforms

Future Work

- Extend the chain-of-trust from system image to each individual application
- Add updatable trust anchors to support evolving public keys and new algorithms

Challenges Facing the Adoption of COTS

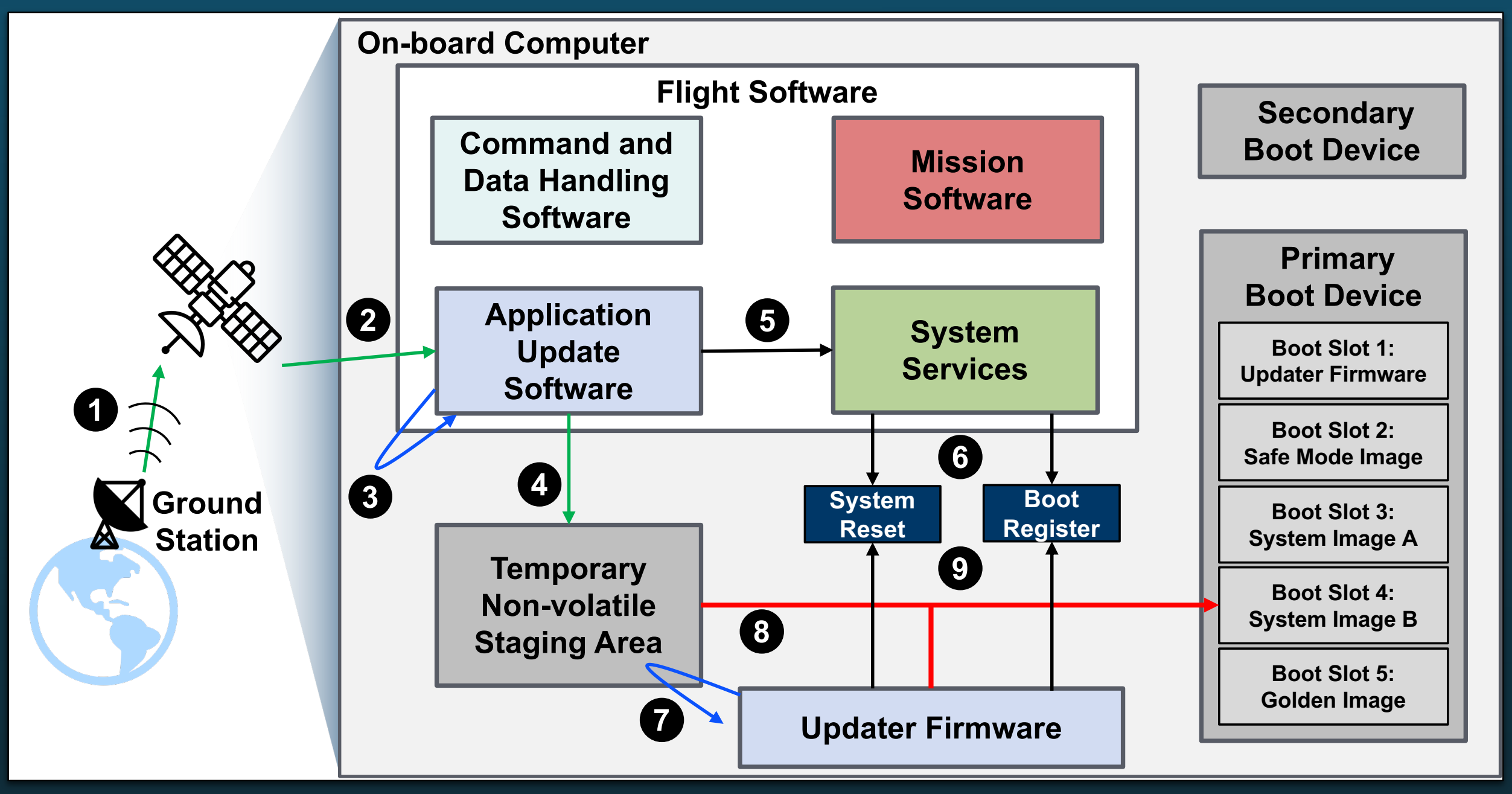
Implicit Vendor Trust:
A satellite system that adopts a COTS platform must implicitly trust the security critical, but often propriety and unobservable hardware Root-of-Trust (RoT) implementation.

Rigid Security Features:
Platforms often consist of fixed algorithms and security features potentially missing needed mission requirements.

Dependence on Vendors:
Users can spend costly effort to extend a COTS system or wait until vendors include wanted features into newer devices. While hardware-based security is resilient, efficient, and fast, it cannot be updated in the field.

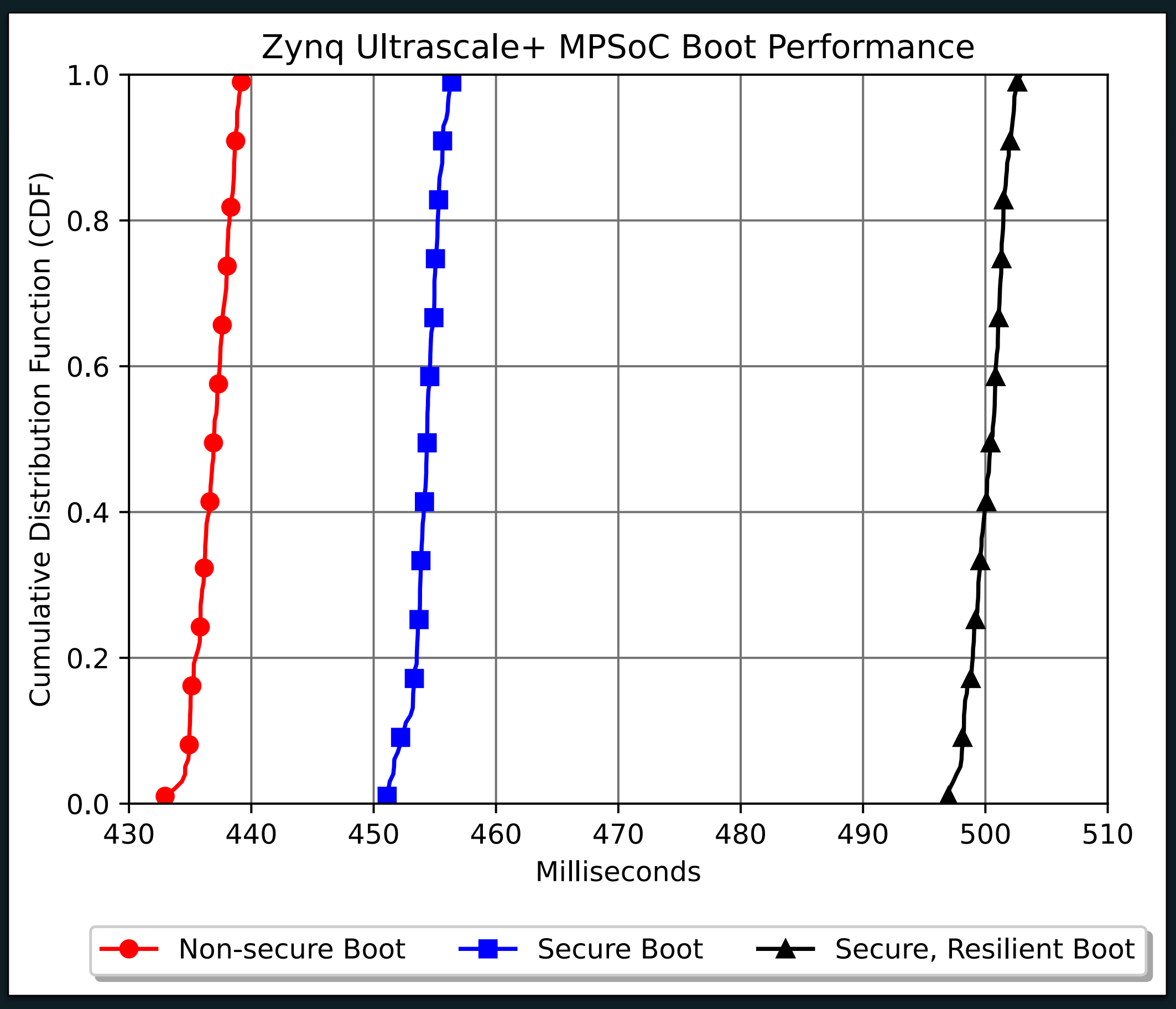
Inheriting Shortcomings:
Devices such as flash storage have proprietary implementations of safety critical behavior that can leave engineers with no choice but to implement their own in order to guarantee known, safe behavior.

Secure, Resilient Update Design


On-board Computer
Flight Software
Command and Data Handling Software
Mission Software
Secondary Boot Device
Primary Boot Device
Boot Slot 1: Updater Firmware
Boot Slot 2: Safe Mode Image
Boot Slot 3: System Image A
Boot Slot 4: System Image B
Boot Slot 5: Golden Image
Application Update Software
System Services
Ground Station
Temporary Non-volatile Staging Area
System Reset
Boot Register
Updater Firmware
1
2
3
4
5
6
7
8
9

- Updates are transmitted in chunks with integrity checks
- The A/B update pattern ensures the system can successfully recover from failed updates
- An independent firmware image finalizes updates and limits access to boot devices

Evaluation



- Measurements were taken of 100 boot sequences
- On average, using secure boot features only increases the boot time by 17.41 ms
- Including both security checks and multi-boot fallback, boot time only increases, on average, 63.32 ms