AIAA Aviation Forum and ASCEND co-located Conference Proceedings
29 July - 2 August 2024, Las Vegas, Nevada
AIAA AVIATION FORUM AND ASCEND 2024

10.2514/6.2024-4853

Lockheed Martin's Ignite Ecosystem Facilitates Rapid Space Technology Development Using End of Life Satellite

Matthew Wescott[1], Brian Wenzel[2], Sara Ghika[3], William P. Honicky[4], Ryan J. Cutter[5]
Lockheed Martin Space, Littleton, Colorado, 80127, United States

Kyle Murbach[6], Oscar McCants[7], William Gwaltney[8], Tommy Morris[9]
The University of Alabama in Huntsville, Huntsville, Alabama, 35899, United States

John Turpish[10], Ryan Plauché[11]
Little Bear Labs, Boulder, Colorado, 80302, United States

Anshuman Prasad[12]
Protocol Labs, Wilmington, Delaware, 19801, United States

Dietrich Ayala[13]
Filecoin Foundation, Middletown, Delaware, 19709, United States

Establishment and healthy growth of the 21st century Space Economy will require not only key technological leaps but also radically different engineering and collaboration paradigms. Whereas the earlier era of space exploration was primarily shaped by governments and monumental engineering programs, we are now living in an unprecedented period of miniaturization, software-defined functionality, nontraditional investment, and new spacefaring entities both geopolitical and commercial. Lockheed Martin Space's Ignite organization is chartered to respond to these new realities, cultivate innovation, and accelerate technology maturation. Factors of risk intelligence (as opposed to risk aversion), multidisciplinary teams, organizational culture, collaborative partnerships, and early access

[1] Senior Systems Architect, Ignite, AIAA Member.
[2] Senior Staff Software Engineer, Ignite.
[3] Senior Mechanical Engineer, National Security Space – Space Protection Programs, AIAA Member.
[4] Systems Engineer, National Security Space – Space Protection Programs, AIAA Member.
[5] Senior Guidance, Navigation & Control Engineer, National Security Space – Space Protection Programs.
[6] Principal Research Engineer, Center for Cybersecurity Research and Education.
[7] Research Engineer, Center for Cybersecurity Research and Education.
[8] General Engineer - U.S. Army Space and Missile Defense Command, Center for Cybersecurity Research and Education.
[9] Director - Center for Cybersecurity Research and Education, Eminent Scholar and Professor - Department of Electrical and Computer Engineering.
[10] Senior Software Developer, Little Bear Labs – Engineering.
[11] Senior Software Developer, Little Bear Labs – Engineering.
[12] Product Lead, InterPlanetary File System.
[13] Technical Advisor, Filecoin Foundation.

Copyright © 2024 by Lockheed Martin Corporation, Filecoin Foundation, & The University of Alabama in Huntsville.
Distribution A. Approved for public release: distribution is unlimited, #4126, DATE (20JUNE2024).. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

to testing resources have been identified as consistent enablers to successful engineering programs within the Ignite portfolio. More specifically, early technology demonstrations right-sized to use flight-like testing platforms will generate valuable insights and feedback loops used to revise product features while also removing cost, schedule, and performance risk from final versions. The Ignite innovation ecosystem has shown the real-world application and benefits of such an engineering approach using a 12U Lockheed Martin In-space Upgrade Satellite System (LM LINUSS™) CubeSat prior to decommissioning. Having completed primary mission operations in early 2023, it was determined from vehicle state of health that LINUSS Space Vehicle 2 (SV-2) could continue to function on orbit for several additional months. Thus, the spacecraft was repurposed as an on-orbit testbed within the innovation ecosystem. This rare opportunity provided software teams the ability to rapidly test aspects of their applications in a true space environment under constraints of low data bandwidth, limited compute resources, and intermittent communications links. Through internal cross-collaboration and external partnerships, Lockheed Martin Ignite and LINUSS teams collected and assessed proposed on-orbit activities based on feasibility, cost, schedule, and impact to customer sets within the estimated remaining SV-2 life. The multiple successes of these activities signify a wider frontier of opportunities to advance the Space Economy through synergistic partnerships and the novel use of end-of-life small satellite platforms. This paper will explore the intersection of shifting customer risk appetite, company culture, and modern technologies that have fueled the characteristic positive disruption of the innovation ecosystem. An overview of LINUSS SV-2 will detail the primary mission, engineering and operations processes, key infrastructure, and mission extension phase as an on-orbit testbed. Three externally published collaborative test campaigns will be presented, including the Filecoin Foundation - InterPlanetary File System (IPFS), University of Alabama in Huntsville (UAH) - Center for Cybersecurity Research & Education (CCRE), and Operations Center of the Future (OCOTF). Limitations, lessons learned, and broader applications from this body of work will be discussed in the context of the expanding Space Economy and opportunities for future developers.

I. Nomenclature

12U	=	Standard CubeSat volume measuring 20x20x30 cm
ΔV	=	Delta velocity, change in velocity
GEP	=	Ground Entry Point
IPFS	=	InterPlanetary File System
LINUSS	=	Lockheed Martin In-space Upgrade Satellite System
SSC	=	Swedish Space Corporation
SSD	=	Small Satellite Defender
STARS	=	Space Testing and Resiliency Simulation
SWAP-C	=	Size, Weight, Power, and Compute

II. Introduction

A. Background on Lockheed Martin Ignite

Lockheed Martin Space established Ignite as the strategic and holistic response to the rapidly changing marketplace and underscored need for government and commercial customers to acquire space-based capabilities at new, accelerated speeds. As a dedicated innovation division, Ignite is chartered with strategic directives over select next-generation Internal Research & Development (IRAD), Collaborative Research and Development (CRAD), rapid prototyping and demonstrations, and fostering partnerships with key industry and university partners. Once sufficiently matured, select technologies from the portfolio are merged into core internal business areas and customer programs or may be developed into new products for external commercial sales. Ignite also consults with, supports, and aligns many positive disruption projects and groups across the larger Lockheed Martin Space business segment. Finally, the organization's cross-functional innovation ecosystem aggregates enterprise resources to enable rapid analysis, experimentation, and real-world deployment of new technologies. Resources are organized into four major categories:

1. Innovation Pipeline: An organizational framework for strategic road mapping, ideation, teaming, and testing to quickly produce results to inform follow-on work. Examples include internal and external partnerships, IRAD, and Development, Security, & Operations (DevSecOps) Software Factory[*].
2. Development & Test Environments: Lockheed Martin modeling tools, digital twin technologies, and visualization capabilities used to minimize costs of prototyping, time, and labor necessary to produce digital artifacts used in customer demonstrations.
3. Space & Terrestrial Test Assets: Utilizes Lockheed Martin platforms currently operating in ground, air, and space domains to provide testing services across Technology Readiness Levels (TRL) 1-9 [1]. Examples include ground terminals, unmanned aerial vehicles (UAVs), and LM LINUSS™ [2, 3].
4. Cloud Computing Platforms: Leverages enterprise cloud computing platforms, automated infrastructure, and other resources to reduce operating costs of software development and deployment.

Shifting customer risk appetite, company culture, and computing technologies have all fueled the formation of this enterprise division dedicated to positive disruption. As evidenced in published government space domain strategy and acquisitions, entities previously operating solely in NASA Class A and B mission profiles [4] now selectively define and request cost, schedule, and performance deliverables suitable to Class C or D. Authors within Lockheed Martin have experienced these requests and even more granular shifts in risk and opportunity management within active programs, e.g., the overall mission may be classified as C but individual subsystems may only be required to meet class D specifications. Paralleling this systematic change, Lockheed Martin leadership has encouraged programmatic and cultural shifts towards Agile development practices[†]. The Agile Manifesto[‡] values of 'working software over comprehensive documentation,' 'customer collaboration over contract negotiation,' and 'responding to change over following a plan' are implicit in leadership philosophy and staff initiatives to infuse systems thinking, risk intelligence, and rapid prototyping mindsets in their daily work while maintaining technical rigor at all levels. For instance, engineers are encouraged to iteratively and evolutionarily arrive at the 85% minimum viable prototype as proof of technical feasibility and merit of larger follow-on investment. Finally, Lockheed Martin adoption of cloud computing, DevSecOps, and other modern digital engineering technologies has lowered upfront development costs while expanding mission capabilities beyond hardware centric definitions. Specifically, aerospace flight hardware is increasingly being transformed from specialized components into interchangeable commodities which are defined using software configurations, hosted applications, and capability sets. Lockheed Martin's SmartSat™ product was developed for this express purpose, providing abstraction of the physical platform resources (e.g., compute, data storage, sensor feeds), definition, execution, and redefined mission functions using an application-based computing environment [5, 6].

In summary, the overarching directive behind Ignite, organizational structure, portfolio, and above catalysts all underlie the ethos and technical foundation which produced the rapid space technology developments and satellite-testbed-as-a-service model discussed in this body of work.

B. Background on Lockheed Martin In-space Upgrade Satellite System

1. Primary Mission

Lockheed Martin's LINUSS mission was a technical demonstration designed to show how small satellites can play a significant part in 'sustaining critical space architectures in any orbit' [7]. The major objective of the mission was Rendezvous and Proximity Operations (RPO) between two LM50™ 12U CubeSats [7]; LINUSS Space Vehicle 1 (SV-1) and SV-2. RPO missions define a *Chase* vehicle and *Resident Space Object* (RSO) target, with the goal of Chase maneuvering to close distance with the RSO in orbit.

The spacecrafts launched on USSF-44 in November of 2022 and were dispensed from their rideshare at geosynchronous orbit (GEO)+300km in early 2023 [7], at which point the primary mission began. This phase of the LINUSS program occurred over the course of approximately two months [7]. The SVs were dispensed about 870km apart and underwent system checkouts during their first days in orbit [7]. After establishing communications, the team conducted vehicle software updates, downlinked critical data, verified payload functionality, and performed a fly-by [7]. During the initial on-orbit checkout, the team was able to trend key information about thermal and electrical behavior. This trending was necessary in determining the planning constraints used later in the mission. In the following weeks, the Chase vehicle (SV-2) performed several inertial burns to reach close proximity with the RSO

[*] Lockheed Martin, "Lockheed Martin Software Factory," URL: https://www.lockheedmartin.com/en-us/capabilities/digital-transformation/software-factory.html
[†] Agile Alliance, "What Is Agile? | Agile 101 | Agile Alliance," URL: https://www.agilealliance.org/agile101/
[‡] "Manifesto for Agile Software Development," URL: https://agilemanifesto.org/

target (SV-1), approximately 25km, before initiating a terminal guidance burn to transition to relative navigation [8]. The Chase spacecraft then entered a spiral approach towards the RSO vehicle, followed by an ingress horizontal to the velocity vector [8]. This rendezvous culminated in a fly-by, accomplishing a range to target of less than 500m using autonomous three degree of freedom (3DOF) navigation [7, 8].

The mission was deemed a success as both vehicles were commissioned and not only achieved RPO within 500m but did so highly automized [7]. This was a great achievement considering the already challenging task of performing an RPO demo mission with CubeSats in GEO was executed in such a short timeframe. A thorough overview detailing the unique challenges and inventive solutions encountered throughout the primary mission lifecycle, from design to operations, is presented in Cutter et al [9].

2. Spacecraft & Payloads

The LINUSS mission utilized a cutting-edge small-sat architecture to achieve its RPO primary mission and extended operations objectives. Each LINUSS vehicle was packaged within a 12U volume, informally referred to as the size of a four-slice toaster, approximately 9x9x15 inches [7]. This LM50 platform included integrated avionics, power, star trackers, reaction wheels, and payload sub-systems; the majority of which were provided by Terran Orbital [7], who specialize in small to medium sized satellites[§]. The vehicles each had a fully fueled wet mass of 25kg, peak power of 110W, and capacity to execute ΔV maneuvers up to 30m/s [8]. The LM50 platform accommodated a 5kg, 4U dedicated payload volume with maximum power draw of 40W [8]. The SVs utilized this space for multiple visible-spectrum imagers which enabled them to complete the RPO concept of operations (CONOPs). Each imager featured a field of view chosen based on expected critical ranges between the SVs throughout the primary mission.

3. FlatSat & HWIL Testing Infrastructure

The LINUSS team leveraged multiple ground assets to ensure mission success - two of which were the flat satellite (FlatSat) and Hardware-In-the-Loop (HWIL) testbeds. The FlatSat is a ground-based Engineering Development Unit (EDU) flight hardware setup containing command and data handling (C&DH), payload processing unit (PPU), electrical power, and software systems identical to the SVs. This allowed for detailed checkout and testing of any on-orbit procedures and hardware commands. Conversely, the HWIL uses closed-loop simulations to emulate hardware sub-systems and interactions, specifically when given inputs such as ephemeris or attitude. The HWIL system was used to test a majority of the computer vision algorithms and other SV sub-systems required for the RPO mission [7].

The LINUSS ground architecture created seamless collaboration between stakeholders, as its cloud-based Horizon™ Telemetry, Tracking, and Commanding (TT&C) software suite[¶] [7] allowed engineers to perform test cases on hardware regardless of physical location. A specific Horizon instance was created for development and test to replicate all interfaces needed for primary and extended mission test cases. The FlatSat environments proved invaluable for running rigorous 'Test-Like-You-Fly' software checkouts in collaboration with Filecoin Foundation and University of Alabama in Huntsville teams.

4. Mission Operations

The primary mission was staffed around the clock with four teams of operators, technical subject matter experts, and chief engineers. LINUSS partnered with the Swedish Space Corporation (SSC) as the Ground Entry Point (GEP)[#], which provided commercial scheduling and remote connect services to a network of antennas [7] covering the entire LINUSS orbit. Due to the 4 deg/day longitudinal westward drift at GEO+300km, contact windows with the SVs were calculated and scheduled in advance with SSC to ensure maximum communication time.

Scheduled contacts were specific to each SV, during which a Lockheed Martin Ground Engineer stayed in continual contact with the GEP operator to coordinate any communication changes based on the mission itinerary. Routine operations included downlinking data from each vehicle, performing software updates, and scheduling maneuvers to occur both in and out of ground communication. The Vehicle Ground Controller (VGC) role was the sole authority for commanding the vehicles using the Horizon command and control (C2) terminal. This terminal also provided task scheduling and graphic user interfaces (GUIs) for displaying messages and data. Using the cloud-based Horizon C2 software for operations [7] enabled the team to remotely monitor or resume the mission regardless of unforeseen circumstances.

[§] Terran Orbital, "Terran Orbital Home Page," URL: https://terranorbital.com/

[¶] Lockheed Martin, "Horizon™ and Compass™ Satellite Software," URL: https://www.lockheedmartin.com/en-us/products/satellite-software.html

[#] Swedish Space Corporation, "Swedish Space Corporation Home Page," URL: https://sscspace.com/

III. Extended Operations Approach

Having completed all primary mission objectives in first quarter of 2023, the LINUSS engineering team continued contacting SV-2 on a bi-weekly cadence to perform health checks, memory cleanup, and maintenance activities. In parallel, program leadership began exploring transfer of the space vehicle to other interested Lockheed Martin internal organizations who would assume funding and define new strategy. The Ignite organization partnered with the LINUSS team and adopted SV-2 as an innovation ecosystem resource. The partnership provided access to SV-2 and staff of highly skilled vehicle engineers, ground operations support, mission planning, and payload software developers. Correspondingly, formal addition to the Ignite ecosystem, specifically the Innovation Pipeline and Test Assets families, opened the opportunity space to inputs from Ignite's network of initiatives, strategic internal collaborations, and external commercial partners. After assessing onboard payloads and vehicle capabilities, the payload side Innoflight Compact Flight Computer (CFC-400) and SmartSat computing environment were identified for opportunities to uplink and verify multiple software products in a series of testing campaigns.

Ignite and LINUSS program leadership staffed a small Agile Scrum[†,**] team with the directive to cross-collaborate with software product teams across the enterprise who would benefit from readily achievable access to the live satellite platform. Ideas for software-based test campaigns or other risk reductions were captured in a backlog for detailed planning. Backlog entries were populated with first order cost, labor, and schedule estimates as well as key dependencies. Stakeholders of each activity supplied business-centric value proposition statements, which were used to assess the impact of each activity if executed and successful. With the above information, the team prioritized the backlog, assigning top priority to those with highest impact to internal or customer programs, next ordered by shortest schedule duration, then by cost. Shortest schedule duration was a critical constraint, as SV-2 lifetime could be estimated on a basis of state of health trends, recent vehicle anomalies, and vendor rated lifetime of subsystems but there was the ever-present possibility of catastrophic single event upsets. The prioritized activities were assigned to the joint Scrum team to be executed in FlatSat ground testing, a critical integration phase and risk reduction prior to execution on SV-2. Other high business impact activities with long schedule estimates or critical dependencies blocking their immediate execution were assigned to team leaders to be separately deconflicted until ready.

In parallel to the programmatic coordination and ground testing of the multiple new software products, LINUSS SV-2 needed to be maintained in a healthy ready state. Successful execution of this *satellite testbed-as-a-service* mission extension phase required operating well past the originally envisioned design life and resolution to any disruptions. Of prime example is the challenge that operators faced in managing the spacecraft's reaction wheel momentum buildup. The best method of mitigating this was using the 6DOF-capable propulsion system to perform momentum management via expending propellant to de-spin the vehicle's reaction wheels, which are the primary attitude control actuators. This was performed as necessary to keep the total system momentum low and avoid saturating the reaction wheels. However, in the interest of propellant conservation, LINUSS vehicles can also depend on passive methods of momentum management. Considering the near-GEO orbit far from the influence of Earth's magnetic field, torque rods would have been ineffective for momentum management and were thus not included in the vehicle design. As the vehicles also do not experience observable drag from the Earth's atmosphere or gravity gradients, the only way to decrease momentum passively is to make use of the only other appreciable external force - solar radiation pressure (SRP). Thus, LINUSS SVs take advantage of SRP by maintaining a pointing vector defined by Guidance, Navigation & Control (GNC) engineers based on current sun-relative attitude and vehicle angular momentum.

IV. Extended Mission Examples

Over the course of five months the collaborating teams executed seven unique missions to advance operations and TRL of nine products. Of the nine, three have been publicly published and are available as illustrative examples of the satellite testbed-as-a-service model that the teams developed and replicated.

A. Filecoin Foundation

1. Background & Concept of Operations

The InterPlanetary Filesystem (IPFS)[††] is a content-addressing system and network designed to enable applications to operate on data of variable size and without predetermined server locations while providing transport security and content integrity guarantees. Created in 2015 by Juan Benet, founder of Protocol Labs, IPFS accommodates any

[**] Scrum.org, "What Is Scrum?," URL: https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team
[††] IPFS, "An Open System to Manage Data without a Central Server," URL: https://ipfs.tech/

underlying transfer protocol, security in transit, and is location-agnostic data storage. This set of capabilities enables ecosystem-level interoperability, application layer flexibility, and data integrity across space assets. The implementation of the protocol deployed in the test campaign on LINUSS SV-2 is named Myceli and is written in the Rust programming language. Figure 1 illustrates the Concept of Operations planned and executed by the Filecoin Foundation, Lockheed Martin Space, Protocol Labs, and Little Bear Labs in a series of bi-directional communication tests using IPFS protocol across SV-2 and SmartSat ground infrastructure.

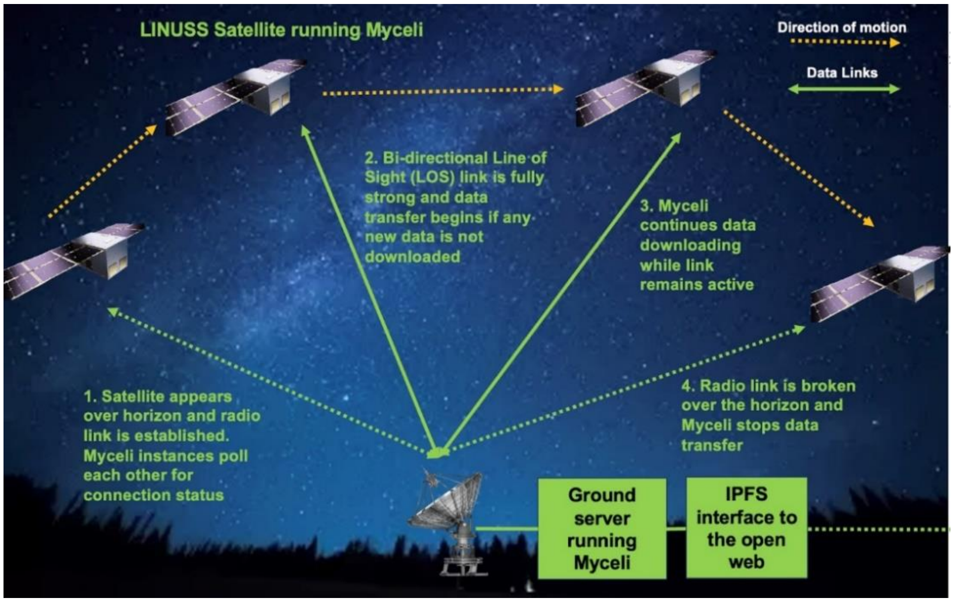

Fig. 1 Concept of Operations showing data transmissions between ground and space based IPFS Myceli instances.

2. Design & Setup

The goal of the on-orbit experiment was to transfer files of various sizes and formats bi-directionally between the two Myceli instances running on a cloud hosted SmartSat ground server and SV-2. The two selected test files were the IPFS whitepaper (166kB size) and a JPG format image (17kB size), which were loaded onto the hard disk of the ground server. The Myceli install consisted of a Linux binary and supporting application files compatible for deployment to the SmartSat virtualized environments running on Lockheed Martin ground station infrastructure and SV-2 respectively. Myceli was configured to use existing communication connections without any awareness of data link specific encoding or modulation schemes. Figure 2 illustrates the deployed Myceli architecture while Fig. A-1, appearing in Appendix A, outlines the sequence of IPFS commands used to execute a data transfer from satellite to the ground node.

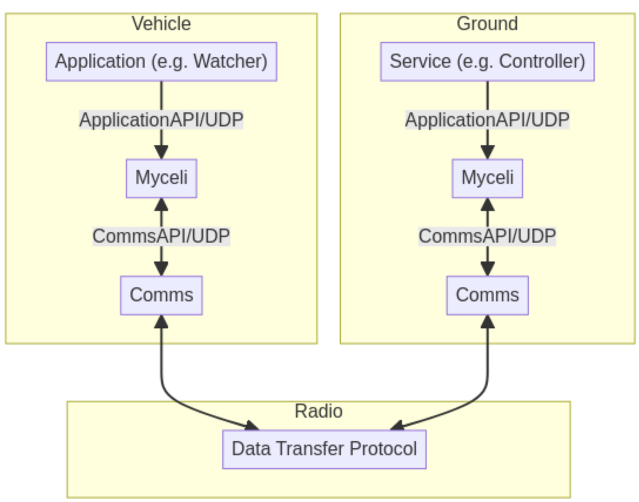

Fig. 2 High level Myceli software architecture.

The block diagram depicted in Fig. 3 shows the software interactions across installed IPFS onboard and ground instances, the TT&C communication link through Horizon to the FlatSat (or SV-2) bus avionics, routing through the avionics, and endpoint on the CFC-400.

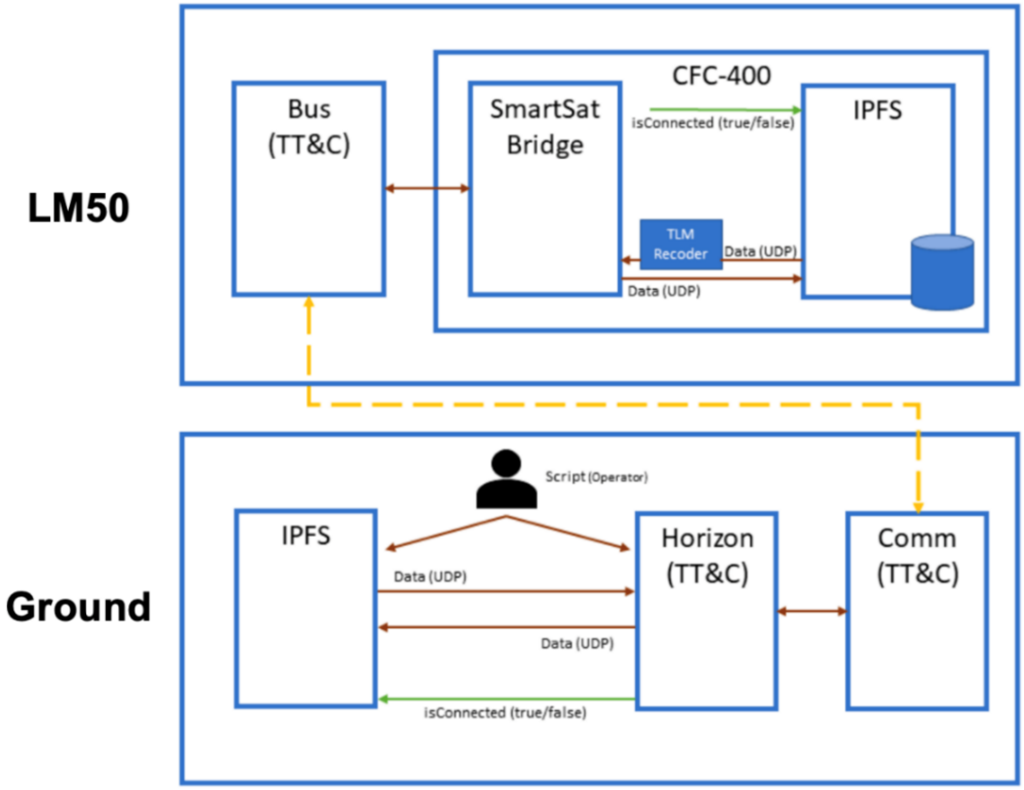

Fig. 3 IPFS testing on LINUSS FlatSat.

As discussed below in the Ground Testing portion of the Filecoin Foundation mission example, the planned FlatSat and SV-2 test campaign covered bi-directional transmissions of various sizes in order to validate performance and operational assumptions while exercising the full range of success and failure states within the system.

3. Ground Testing

The Lockheed Martin, Protocol Labs, and Little Bear Labs teams conducted FlatSat functional testing of the IPFS software in May 2023. This integration verified messaging bus services and interactions with the avionics and Innoflight CFC-400 payload processing unit which IPFS data passed through. A summary of the test campaign is shown in Table 1. Note that the tests were carried out sequentially with configuration settings, installed binaries and data files retained between steps. The success criteria for the IPFS functional testing included:

1. Installing IPFS to the payload processing unit
2. Commanding both ground and vehicle IPFS nodes
3. Transferring small data package (one contact) to and from onboard IPFS node
4. Transferring large data package (multiple contacts) to and from onboard IPFS node
5. Returning to original LINUSS flight configuration

Table 1 IPFS ground testing campaign.

Task	Goal	Results	Success
Validate FlatSat PPU Configuration	Validate the flight configuration of the PPU Software before test	Flight config validated	Criteria met
Send IPFS Install File	From Horizon, send a SmartSat file which includes the IPFS installer, and validate that it has been successfully received	IPFS Installer Received	Criteria partially met. Slow uplink speed meant the install file was sent over hardwire rather than simulated S-band to save test resources
Execute IPFS install/reboot	From Horizon, send a SmartSat command to execute IPFS install and reboot	IPFS Install Complete	Criteria met
Test connection between IPFS nodes	From Horizon, send an IPFS command to retrieve an IPFS file	IPFS on PPU responds with expected null data blocks	Criteria met
Send new IPFS File	From Horizon, send a new IPFS file via a SmartSat command	File received on PPU via IPFS and saved to disk	Criteria met over single and multiple contacts
Send IPFS file retrieve command	From Horizon, send an IPFS command to retrieve the IPFS file	File retrieved on Horizon node via IPFS and saved to disk	Criteria met over single and multiple contacts
Restore and validate PPU Configuration	Reformat PPU and validate the flight configuration after test	Flight config validated	Criteria met

All tests successfully passed with the 166kB IPFS whitepaper and 17kB JPG image files transferred to and from the payload processing unit. Engineers noted some configuration changes were required prior to beginning the SV-2 testing regime to mitigate the slow uplink rate. This included adjusting the Maximum Transfer Unit (MTU), window size, and retry duration of the link. Following these adjustments LINUSS operators uplinked, unpacked, and installed the Myceli binary on SV-2 in preparation for on-orbit testing phases.

4. On-Orbit Testing

The command and control interactions used in on-orbit testing differed from FlatSat with the inclusion of Amazon Web Services (AWS) GovCloud. As shown in the demonstration architecture of Fig. 4, an operator acting as the *IPFS Controller* started by issuing commands through a scripting tool within AWS GovCloud. Command messages were routed to the SmartSat Ground system which then identifies and directs them to the IPFS Ground Node pod within the AWS GovCloud. The IPFS Ground Node running the Myceli IPFS application accepts the commands and performs the specified tasks. When the IPFS Ground Node issues a file transfer command to the spacecraft the messages are consumed by the SmartSat Ground service and translated into payload commands that the on-board spacecraft payload processor can accept. Those payload commands are then sent to the dedicated LINUSS instance of Horizon C2, which wraps the payload commands into SV-2 bus commands. Onboard the spacecraft, the bus identifies the payload command and routes it to the payload processor containing the IPFS Satellite Node. The IPFS Satellite Node either accepts the new uplinked data, or a downlink command will initiate the process in reverse for data to flow down to the IPFS Ground Node.

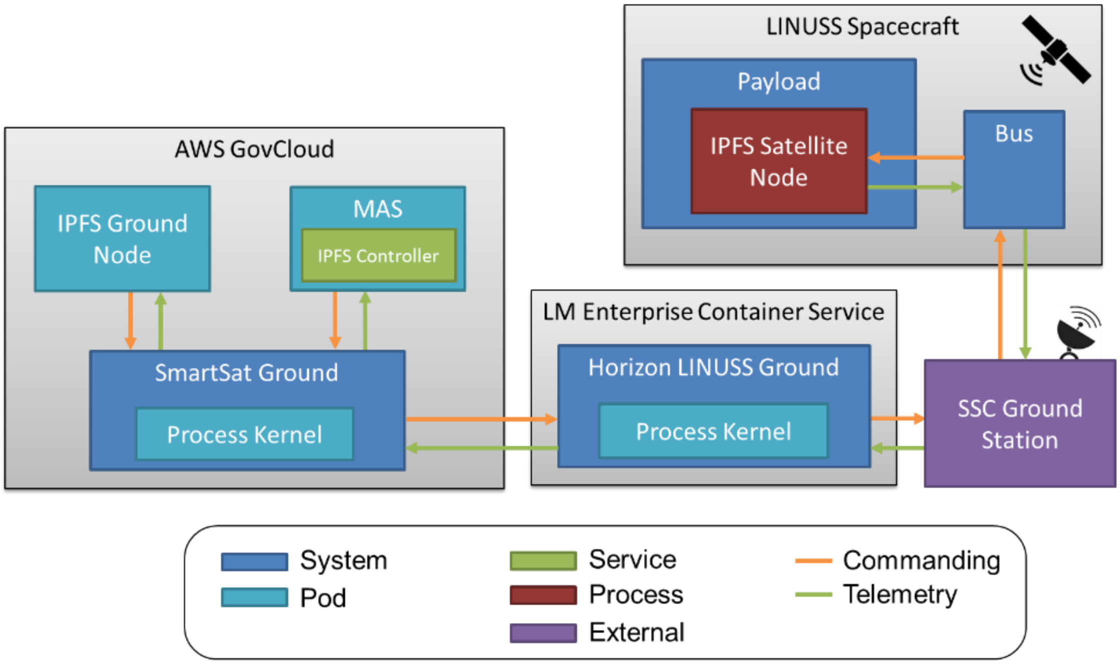

Fig. 4 On-orbit IPFS demonstration architecture.

The goal of the on-orbit demonstrations was to successfully address data by its contents within the IPFS Ground Node and then move it between the IPFS Ground and Satellite Nodes. IPFS generates a unique identifier based on the contents of files called a Content Identifier (CID). When data is transmitted, the CID can be recomputed by the receiver to check for any errors in transmission and to verify for data integrity. The controller used three main scripts to command the IPFS Ground Node for these demonstrations:

1. ipfs-import-ground-file is used to ingest data into the IPFS format and create a CID
2. ipfs-transmit-dag is used to direct a specified CID to be transmitted from the IPFS Ground Node to the IPFS Spacecraft Node, or vice versa
3. ipfs-set-connected is used to stop IPFS from continuing to transmit files and simulate no connection with the spacecraft

Three files of varying size and type were used during on-orbit testing campaign: a 3kB text file, a 33kB PNG image, and a 209kB PDF. Detailed steps and results from three separate SV-2 demonstrations using these test media are detailed in Table A-1 of Appendix A. Note that all these tests verify file transmission between the ground node and the satellite by computing the CID on the file before transmission and then comparing that to a recomputed CID of the file on the receiving node.

5. Results and Extended Impact

Working with the Ignite and LINUSS teams and deploying to SV-2 marks the first time IPFS has been operationally tested against challenges like link latency, short connection times, interference, and limited network topologies. By going through FlatSat and space-based testing cycles, the Filecoin Foundation was able to elevate Myceli from a TRL 2 to 7 [1] in a short duration. By working through initial CONOPs, prototyping, testing, operational integration and mission deployment, Myceli rapidly made progress towards being a re-usable space-based data link.

IPFS has proven to be functional in real space conditions, providing flexibility and portability at the application layer while ensuring data integrity. Myceli is the first implementation of IPFS purpose built for a non-terrestrial network and extends IPFS's reach from thousands of ground nodes into the space domain. This architecture supports a multi-domain, multi-provider, easily upgradable infrastructure, essential for future scenarios involving permanent off-planet assets and a heterogeneous space ecosystem and economy. Filecoin Foundation has licensed this software under an MIT license for re-use by other entities for future commercial research and deployments. By maturing Myceli to a TRL 7 and offering a stable open-source baseline, IPFS is also available for adoption by projects interfacing with other space platforms and sensors over Modular Open Systems Approach (MOSA) compliant applications.

B. The University of Alabama in Huntsville

1. *Background & Concept of Operations*

The University of Alabama in Huntsville's (UAH) Center for Cybersecurity Research and Education (CCRE) Space Testing and Resiliency Simulation (STARS) team was given the opportunity to run an in-house developed software designated Small Satellite Defender (SSD) on LINUSS SV-2. SSD is a low-resource-use intrusion prevention system (IPS), malware scanner, file integrity monitor, and system diagnostic tool designed to defend space systems against potential cyber threats. It was designed to replace free and open-source tools too resource intensive to run on low size, weight, power, and compute (SWAP-C) satellite hardware. Requirements specified by US Cyber Command as the primary stakeholder were to provide verbose system statistics and metrics, detect a custom remote access trojan the team created, and protect the system from standard enterprise threats such as malware, unauthorized applications, network intrusion, deleting or changing system files, and rogue users. The original use case was to defend a Raspberry Pi satellite simulator while allowing the resource intensive simulation processes to run without interruption or loss of service. SSD was presented at the CYBERRECON conference in April 2023. SSD was pure Python 3 at that time, and use of Bash shell scripts would be added later.

In collaboration with Lockheed Martin Space the STARS team defined the technology demonstration CONOPs to assess the real-world feasibility of using SSD system diagnostic and monitoring features on live SV-2 hardware. The experimental scenario consisted of a malicious action occurring on the satellite and then using SSD to capture relevant data that showed evidence of the event. SSD was used to capture an initial resource baseline, then the malicious action was executed while SSD continued overall measurements of onboard resource utilization, followed by a final dataset captured to observe resources returning to baseline or continued elevated usage. Several approaches were considered for the malicious test scenario, but many possibilities were precluded due to risk of unintentionally disrupting safe satellite operations. The STARS, Ignite, and LINUSS teams ultimately designed a malicious action using a disk fill payload for the test scenario. This involved writing 800 MB random numbers and 100 MB nulls to an onboard persistent storage device that could later be cleared by SV-2 operator commands.

2. *Design & Setup*

LINUSS SV-2 provided hardware resources (RAM, CPU, and storage) and performance comparable to the Raspberry Pi 3B and 3B+ single-board computers used by the STARS team. However, the SmartSat provided Xilinx Linux operating system (OS) lacked toolchains needed to run the original SSD program. The installation and managing of the requisite software packages were ruled out due to lack of uplink bandwidth to the spacecraft and need to maintain known, replicated configurations on the FlatSat ground testing environment and SV-2. Second, the Horizon C2 terminal used by LINUSS operators supported a limited database of vehicle and payload commands. Finally, file transfer-based access was only possible when in contact with the satellite as opposed to continuous connections originally designed into SSD. These constraints underscored the need to refactor the SSD code for execution on SV-2. The STARS team determined that the system shell programming language (Bash) was the best option as it is interpreted natively without calling an external interpreter. The ease of calling system utilities and commands from within Bash made it the ideal solution.

The STARS team utilized an Agile methodology to iteratively develop SSD into a product suitable for the computing environment on SV-2. This facilitated rapid engineering cycles to rewrite the SSD codebase within 2-3 months, with minimum bugs found in testing, and only one escaped defect for use on SV-2. The result of refactoring SSD was a single shell script with dependencies reduced from many to only four, all optional for the desired tests. The total size was reduced to 33.6 kB for version 1.0.2.4 which could be easily transferred to the satellite on the low-bandwidth uplink. As shown in Fig. 5, the test campaign planned for FlatSat and SV-2, the STARS team had to receive valid logs indicating that the different data collection tasks completed properly to deem tests successful. If logs were not received by the end of the contact window, then SSD and Defender Conductor processes would be terminated, and the mission would be paused until a later contact window when logs could be retrieved. FlatSat ground testing of SSD and Defender Conductor continued between contact windows if the logs were not retrieved.

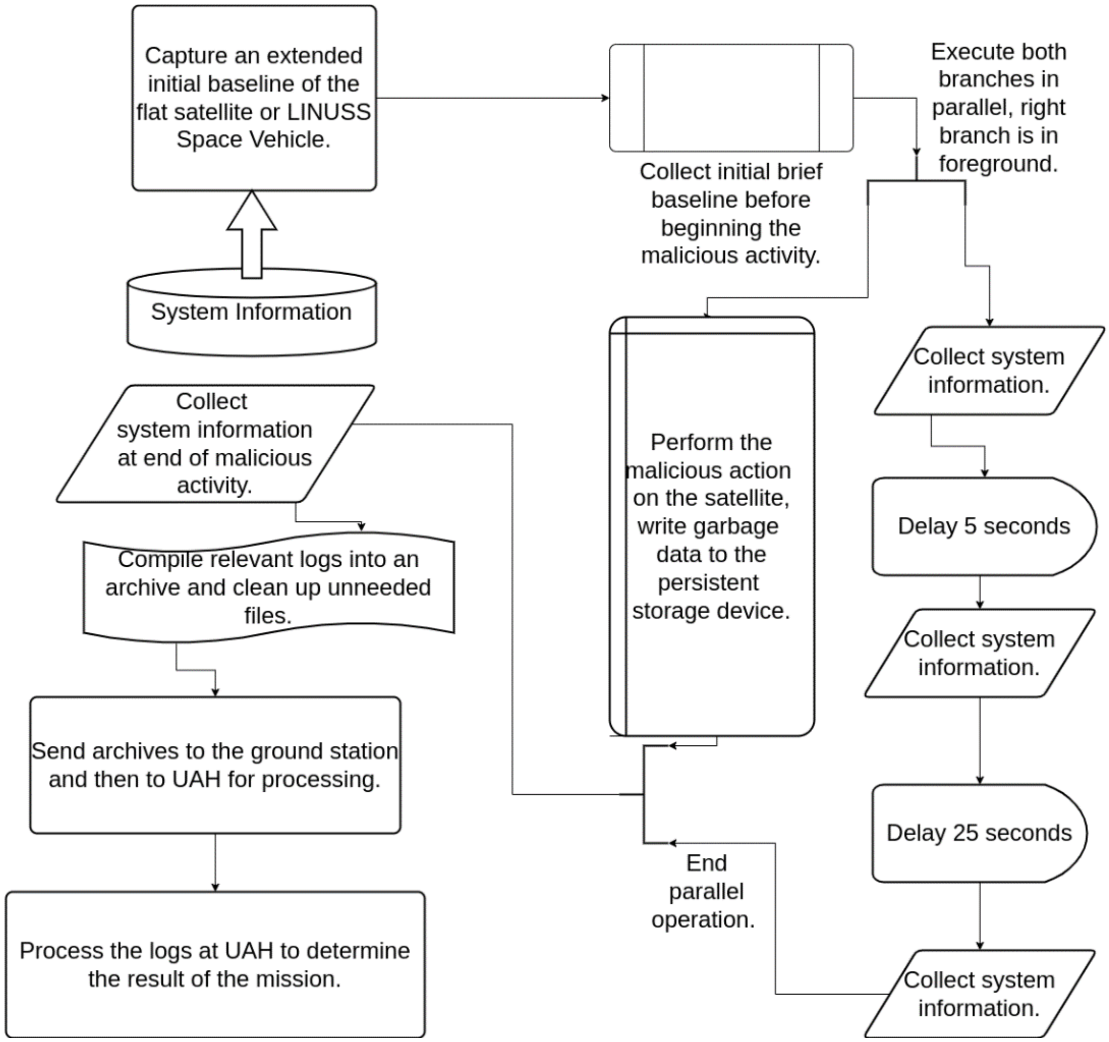

Fig. 5 Flow Chart of mission profile for SSD and Defender Conductor.

3. *Ground Testing*

Using the LINUSS operated flat satellite hardware SSD software releases could safely be tested in real time. This testbed included a live SSH terminal capability but otherwise mirrored the SV-2 environment as deployed. Known differences were that the FlatSat and space vehicle had different major versions of Python, the home directory of the root superuser was the entire filesystem on the space vehicle but was Root's proper home directory found near the base of the filesystem on the FlatSat. At one point in FlatSat testing an unknown difference allowed the malicious action to execute successfully on the testbed but later failed on the space vehicle.

Throughout the development process the STARS team adapted to changing requirements including the late addition of Defender Conductor, an orchestration script for use with SSD. This orchestration script supported the malicious scenario designed in partnership with Lockheed Martin Ignite and LINUSS teams, allowing for the execution of the tests with a single command per test. It required only SSD and basic system utilities, and the largest file size for Defender Conductor was 17.9 kB. Figure 6 shows the top-level block diagram for the SSD and Defender Conductor software programs from the point of view of the user running Defender Conductor on the LINUSS FlatSat and SV-2. Note that the process to take the initial baseline is much more involved compared to the malicious activity experiment, and that some basic system setup is possible. The programs were designed to run from the terminal as the root superuser, and there was no graphical output outside of text printed at the terminal when Defender Conductor and SSD were used.

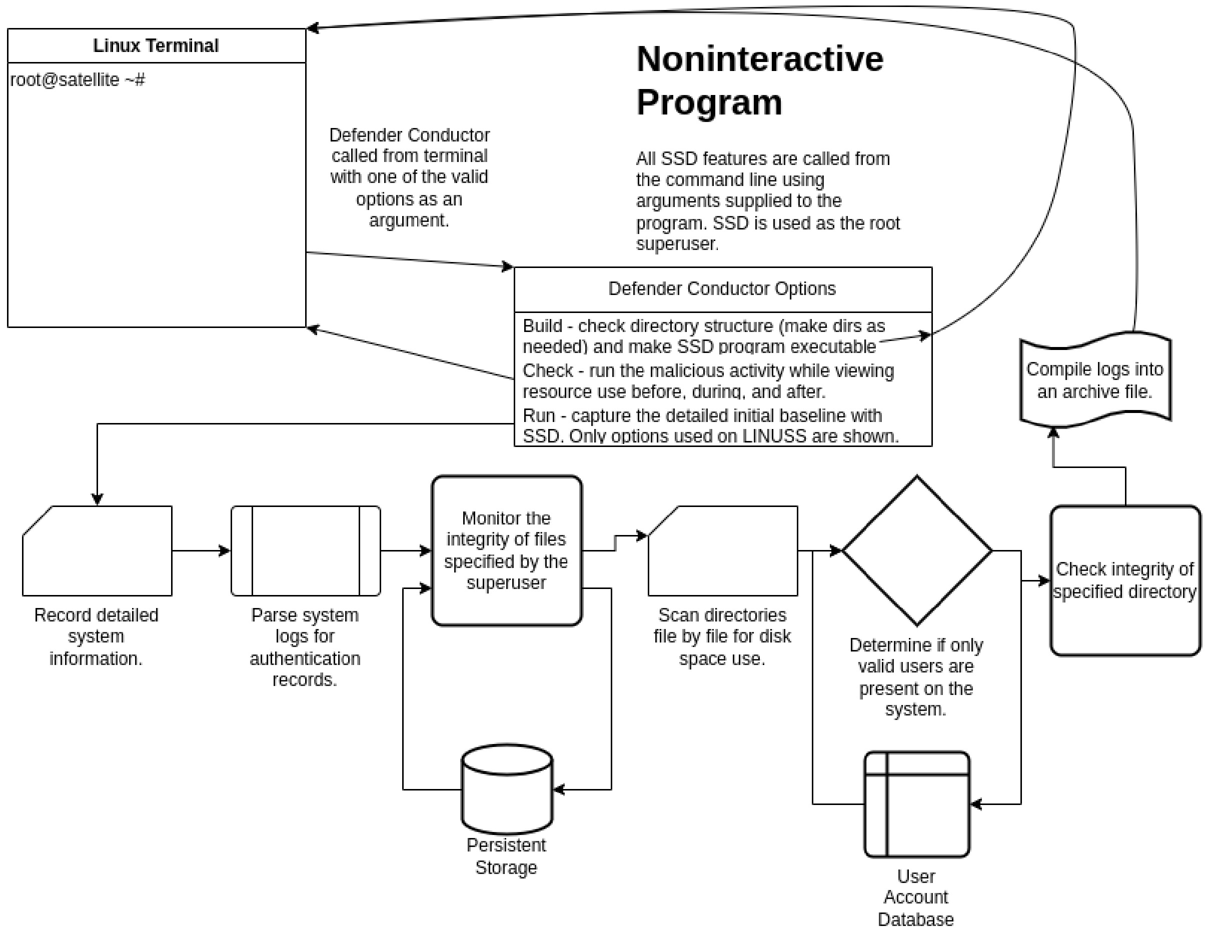

Fig. 6 Top level block diagram of STARS software deployed to LINUSS satellite.

Ground testing was performed multiple times on the FlatSat prior to use of SSD in space. Several versions of SSD were tested with the results of each iteration improving later releases. The availability and use of the existing LINUSS FlatSat obviated any new need for the STARS team to create their own Xilinx Linux test environment on virtual machines while also expediting the refactoring of SSD to function on the target OS.

Final results from the FlatSat testing shown in Table 2 summarize data captured prior, during, and following the malicious activity. Metrics collected at timesteps zero, one, and two represent the system state immediately, five seconds, and thirty seconds after starting the malicious disk fill payload activity. As expected, the system resource utilization climbed during all phases of the test scenario. On the FlatSat the space used in the persistent partition increased after the file creation payload executed, indicating that the generation of cryptographic data and writing it to a file to fill the disk can be detected by SSD. The used space increased by several orders of magnitude from 12 kB to 55 MB. Although this maximum was far less than the 900 MB originally planned for the malicious activity during FlatSat it was adjusted by engineer based on experimental feedback. The load average is a measure of the computational work (i.e., running or waiting processes using CPU, RAM, or disk storage) the system is performing averaged over several periods of time [10]. Load average metrics shown in Table 2 were collected using the 60s-time interval output from the standard Linux uptime command. Note the system load average decreased to 1.10 before increasing at a slower rate to 1.38, peaking at 30 seconds after the start of the malicious action. In operational use, this would indicate unusual activity on the system and warrant further investigation to determine cause. These results demonstrated the feasibility and viability of the SSD experiment to be performed on a real-world space asset.

Table 2   Results of SSD testing on LINUSS FlatSat.

Metric	Baseline	Timestep-0	Timestep-1 (5s)	Timestep-2 (30s)
Load Average	1.14	1.10	1.18	1.38
Persistent Temporary Directory Space Used	12 kB	12 kB	12 MB	55 MB
Persistent Partition Space Used	2.4 GB	2.4 GB	2.5 GB	2.5 GB
Persistent Log Directory Space Used	Empty	12 kB	24 kB	36 kB

Multiple refinements were made to the methods that SSD used when reading and saving log files, and the malicious action was reduced by over half its original size to 200 MB total disk use instead of 900 MB. Defender Conductor was adjusted for a shorter test duration as the timescales used during FlatSat were determined to be too long to support during live satellite contacts. The SSD version 1.0.2.3 incorporated these final adjustments and was accepted for upload to SV-2.

4. On-Orbit Testing

The first on-orbit testing of SSD was executed on August 31, 2023 by collaborating STARS, LINUSS, and Ignite teams. Initial testing uncovered an issue on SV-2 in which the entire filesystem was scanned. This resulted in an extremely large log file that had to be retrieved later. Only the initial verbose baseline was recovered, and the malicious action did not work as expected due to an unknown issue. As a result, the Defender Conductor script was further modified to address these early incongruities. The largest change informed by initial results was that the garbage data written to the persistent temporary directory was reduced a second time from 200 MB to 55MB to prevent Defender Conductor from stalling when writing to the disk. This was the dominant theory as to why the malicious action did not work on the space vehicle.

Further iterative versions were tested on SV-2 with the team working to resolve the unknown issue, which was preventing the malicious test action from executing successfully. Each test of SSD using the Defender Conductor would return the results from only the baseline collection, indicating a likely failure in the malicious action portion of the script. Defender Conductor version 8 and SSD version 1.0.2.4 testing lacked a conclusive identification and resolution to this issue; thus, no malicious activity capture exists from LINUSS SV-2. Only quantitative results from the FlatSat testbed are discussed in this paper due to this unresolved blocker encountered during on-orbit testing.

5. Results and Extend Impact

In orbit, the SSD baseline captures consistently worked as expected on SV-2 without error. However malicious activity logs were never captured during the on-orbit testing. After compensating for the root user home directory encompassing the entire file system on the satellite the logs produced were within a reasonable size with the largest one being only a few KB. The opportunity and test campaign has proven the value of running SSD on SV-2, giving operators the ability to perform file integrity and onboard resource monitoring on low SWAP-C satellite platforms and without interactive access.

As shown below in Table 3, the SSD and Defender Conductor test campaigns on SV-2 greatly advanced the maturity of both software utilities. Defender Conductor was assumed to be a part of SSD for the purpose of evaluating TRL, as Defender Conductor requires SSD to be present to produce meaningful results. A TRL of 9 was avoided due to the malicious payload not working in space which rendered the technology demonstration only a partial success.

Table 3 Achieved Technology Readiness Levels of SSD with Defender Conductor.

Version	1.0.2.1	1.0.2.3	1.0.2.4
TRL (Defender Conductor and SSD)	6	7	8
Rationale	Only demonstrated on the flatsat relevant environment [1]	Intermediate prototype demonstrated in relevant space environment [1]	Final version demonstrated and flight qualified in space environment but does not meet full mission success criteria [1]
Included Defender Conductor Version	Defender Conductor 2	Defender Conductor 5	Defender Conductor 7 (at release), later replaced with Defender Conductor 8

The findings from LINUSS testing continue to help improve the state of SSD. The overall collaboration, FlatSat integration, and on-orbit testing challenged many design assumptions previously inherent to the SSD software product. As previously stated, the Xilinx Linux OS specific to the SmartSat environment on SV-2 is highly optimized for space-based platforms and compute applications. Thus, many utilities and capabilities common to Linux desktop or server distributions were unavailable. Second, the common practice of pulling new software packages onto target devices was prohibited by the low bandwidth uplink and systems requirement to maintain parity between the flat satellite and SV-2 configurations. Third, the original SSD codebase relied on bi-directional real-time operator interaction through use of a terminal window. The LINUSS test campaign informed the redesign of SSD into a unidirectionally commanded shell scripted utility without reliance on external dependencies. These changes and iterative testing produced a version of SSD and newly developed Defender Conductor appropriate to the modern space platforms and operations they were created to protect, e.g., discovering indicators of compromise and generating detailed records before evidence of the malicious activity can be destroyed.

In future work this software baseline could incorporate the final results and lessons learned from version 1.0.2.4 testing into a further refined version of SSD. This new variation could achieve full space qualification at TRL of 9 on future satellite missions.

C. Operations Center of the Future

1. Background & Concept of Operations

The Lockheed Martin Operations Center of the Future (OCOTF) provides a test suite to validate new hardware and software technologies key to aiding tomorrow's satellite operators in the simultaneous management of multiple space missions using a web-based, secure cloud infrastructure [11]. The OCOTF system-of-systems currently combines Compass™ Mission Planning¶ and Horizon C2 software products [11] with mission push alerts, automated data pipelines, artificial intelligence, and machine learning capabilities. Finally, the OCOTF provides cloud-hosted remote accesses for distributed virtual teams as well as a physical operations center in Colorado used for in-person mission critical events, including launch and on-orbit commissioning.

Having already completed internal engineering reviews, the OCOTF needed to prove system wide operational readiness in a series of live satellite contacts before it could be employed in future Lockheed Martin and customer missions. Thus, Ignite and LINUSS teams utilized the OCOTF physical and virtual infrastructure in operations shown below in Fig. 7 supporting SV-2 state of health checks and customer software test campaigns.

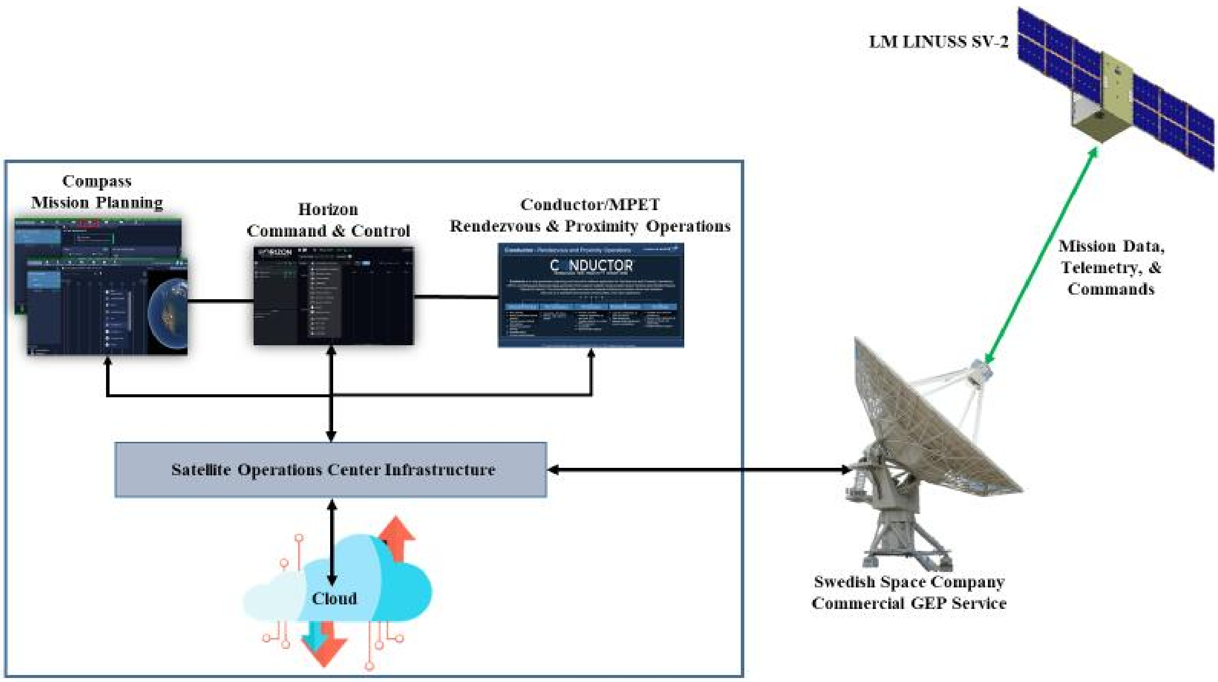

Fig. 7 Operations Center of the Future infrastructure and communications to SV-2 [7, 8, 11].

2. On-Orbit Testing

Over the course of several months, the Ignite and LINUSS teams contacted SV-2 and conducted nominal state of health checks, anomaly recovery, and several customer technology demonstrations using OCOTF. These activities proved all Lockheed Martin enterprise networks, security controls, commercial SSC GEP connections, and all deployed software product instances to be functioning as intended. Additionally, operators provided a repository of operational procedures and documentation from the primary LINUSS mission to OCOTF staff and leadership to be used in future missions.

3. Results & Extended Impact

Having achieved operational readiness, the OCOTF was positioned and has commenced command and control of the recently launched Pony Express 2 [12] mission and constellation. The LINUSS program's GEP connections to SSC through the OCOTF infrastructure effectively proved the connectivity needed to interface with Terran Orbital's ground system, the collaborating subcontractor to Pony Express 2. In the future, OCOTF will provide the common software baseline and operating model for upcoming Lockheed Martin technology demonstrations, including TacSat and LM400 missions [11].

V. Limitations and Lessons Learned

The extensibility of this body of work to the larger Space Economy is tempered with constraints. Most critically, such discretionary mission extension phases are constrained by the need to maintain sufficient engineering and safety margins such that proper disposal can still be completed as required. Second, extended mission phases require some form of continued support and stewardship, as demonstrated by the LINUSS and Ignite teams. This ensures vehicles are kept operational, infrastructure is maintained, testing assets are shared, and knowledge resources are made available to interested customers. Furthermore, this continuity of stewardship ensures information security controls are enforced by the accountable mission operators and parent company. For example, the Filecoin Foundation, UAH, and OCOTF experiments were protected under proprietary information controls, multi-party nondisclosure agreements, or university master research agreements before undergoing review and approvals for public release in this body of work.

Table 4 provides a synopsis of the many programmatic and technical lessons learned gathered by the Ignite, LINUSS, STARS, and Filecoin Foundation teams. While some of these are unique to the size and structure of Lockheed Martin Space organizations, the fundamental insights are applicable to future developers across the broader Space Economy.

Table 4 LINUSS Testbed-as-a-Service Lessons Learned.

Unified strategic direction and priority setting from leadership provided clarity in execution, thus saving engineering cycles.
Business centric and schedule driven prioritization of the backlog of ideas created urgency and clear return on investment.
The SmartSat computing environment reduced non-recuring engineering work and enabled flexible loading of new customer software applications.
The FlatSat testing environment and support by LINUSS staff was a critical development step for every experiment as well as time and cost saving enabler.
SV-2 service life was appropriately estimated from trending health data, known upsets, and engineering judgment but has proven very conservative. SV-2 has surpassed all expectations and continues to function as a satellite testbed-as-a-service for 12+ months.
The low bandwidth and intermittent satellite communications windows proved to be the most constraining reality experienced by software teams, which feeds back into product designs and deployment paradigms.
Multidisciplinary teams composed of systems engineering, software, mechanical, GNC, radiofrequency, and mission planning skill sets proved highly capable of problem solving within the group with little lost schedule.
Agile software development practices by the STARS team were critical to successfully refactoring SSD. The team adapted to platform and operational requirements as well as incorporating new findings throughout the test campaign. Rapid releases kept the bug count low and continuous testing uncovered most bugs before on-orbit testing.
The STARS team new experience with FlatSat ground testing illustrated the criticality of test-like-you-fly driven development. Critical flaws were uncovered and remedied prior to execution on SV-2 where lack of terminal-based debugging tools or verbose error codes would have prolonged iterative testing.
Configuration control between the FlatSat and SV-2 remained critical to safe operations as well as test-like-you-fly development by IPFS and UAH teams. Departure from configuration control would have led to divergent results and difficult to replicate runtime errors on the spacecraft.

VI. Conclusions and Opportunities for Future Developers

The present body of work has demonstrated innovative and accelerated technology maturation through Lockheed Martin Ignite's distinct iterative and collaborative engineering approach. An Agile Scrum model was applied to a cross-functional, multidisciplinary team composed of LINUSS and Ignite members. Adopting LINUSS SV-2 into the Ignite innovation ecosystem as a test asset opened new use cases to broader stakeholders and inputs from the Innovation Pipeline framework. The self-directed Agile Scrum team enacted leadership's initial strategic vision with rapid internal and external customer engagement, risk intelligence, subject matter expertise, and early access to testing resources. The on-orbit service life of SV-2 was extended well beyond the original mission phases though use of several momentum management techniques and vehicle health was maintained on a rigorous contact cadence. Seven test campaigns or other risk reductions used SV-2 hosted testing to advance nine total products. A detailed breakdown of publicly published collaborations by the Filecoin Foundation, University of Alabama in Huntsville, and Lockheed Martin Ops Center of the Future was used to illustrate the applied testbed-as-a-service model (i.e., engagement, CONOP development, design, setup, initial ground testing, initial on-orbit testing, and repeated ground/spacecraft test cycles) and benefits to several forms of customers. The outlined performance and stewardship limitations of this work do not prevent wider application in the Space Economy but are meant to inform future developers wishing to implement a similar model. Similarly, the included lessons learned cover programmatic and technical aspects applicable to any adopting teams and organizations regardless of scale.

The successes of these rapid space technology developments illustrate an underutilized, if not entirely new, frontier of opportunities to advance the Space Economy. On the traditional product lifecycle, space vehicles are often decommissioned and left to decay once funding is expended, or safe operations procedure prescribes disposal based on failing attitude and orbit control. However, at the programmatic and technical decision point between having achieved primary mission objectives and sunsetting, controlling entities can elect to advertise platform availability to internal collaborators as well as external synergistic partnerships. Both options offer the opportunity for teams and businesses to test their products in real on-orbit scenarios without heavily investing resources in dedicated satellite missions. The flexibility of this testbed-as-a-service phase is further augmented by the software defined nature of contemporary satellite platforms, thus allowing development and test approaches, which directly map to deployment on in-situ test targets. Overall, this body of work has shown that platforms deemed 'end of life' can instead be repurposed for innovative new use-cases outside of the original mission and the results encourage future outreach by spacefaring entities to partner or sell hosted services to the broader development community. If widely adopted as a business and lifecycle model, the resultant marketplace of abundant technology maturation opportunities will be an economic multiplier to build up the Space Economy in a boom of rapidly flight-qualified technologies, products, and services.

Appendix A – Supporting InterPlanetary File System Figures & Tables

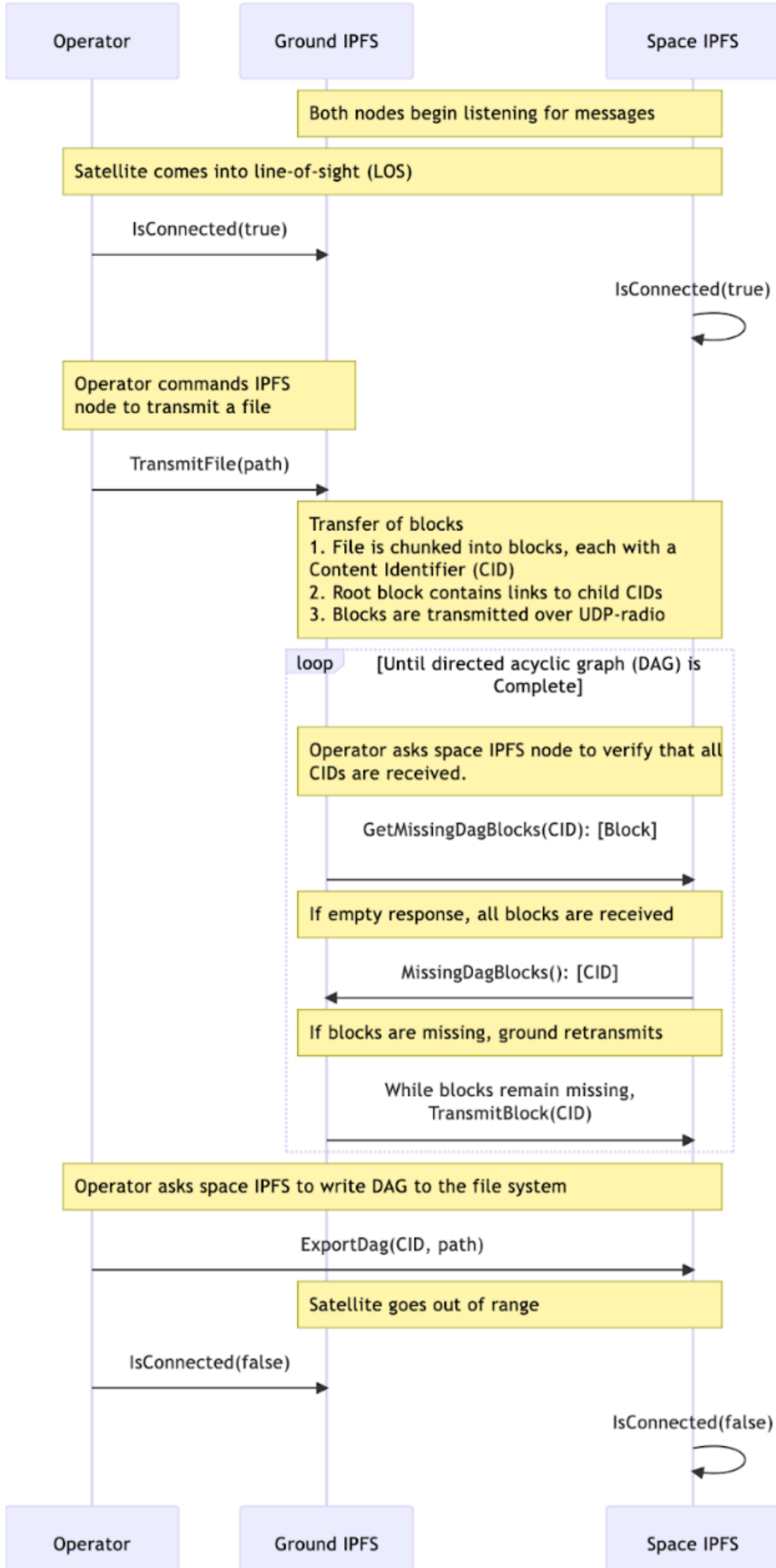

Figure A-1 A block diagram showing the command sequence of transferring data from a satellite to the ground using Myceli.

Table A-1: IPFS on-orbit testing campaign.

Task	Description	Steps	Observations
Small File Uplink	Transmit 3kB file on GovCloud via IPFS to the LINUSS satellite	1. Ingest file by executing the ipfs-import-ground-file script through IPFS Controller 2. Note the CID created during file import to the IPFS Ground Node 3. Initiate ground to spacecraft file transfer by executing the ipfs-transmit-dag using CID 4. Confirm IPFS Ground Node logs show file blocks being sent to SmartSat Ground routing 5. Confirm Horizon LINUSS Ground logs show file #1 payload commands transferring to spacecraft 6. Confirm file has successfully transferred to IPFS Satellite Node by confirming CID match	CIDs match with no observable anomalies
Medium File Uplink with Interruption	Transmit 33kB PNG file on GovCloud via IPFS to LINUSS satellite with simulated termination of connection	1. Ingest file by executing the ipfs-import-ground-file script through IPFS Controller 2. Note the CID created during file import to the IPFS Ground Node 3. Initiate ground to spacecraft file transfer by executing the ipfs-transmit-dag using file #2 CID 4. Confirm IPFS Ground Node logs show file blocks being sent to SmartSat Ground routing 5. Confirm Horizon LINUSS Ground logs show file payload commands transferring to spacecraft 6. Interrupt file transfer by executing ipfs-set-connected script to set connection to false 7. Confirm IPFS Ground Node logs show no file blocks being sent to SmartSat Ground routing 8. Confirm Horizon LINUSS Ground logs show no file payload commands transferring to spacecraft 9. Resume file transfer by executing ipfs-set-connected script to set connection to true 10. Confirm IPFS Ground Node logs show file blocks being sent to SmartSat Ground routing 11. Confirm Horizon LINUSS Ground logs show file payload commands transferring to spacecraft 12. Confirm file has successfully transferred to IPFS Satellite Node by confirming CID match	CIDs match with no observable anomalies
Large File Downlink	Transmit 209kB PDF file from LINUSS SV-2 to GovCloud	1. Note the CID created during file import to the IPFS Ground Node (prior to uplink) 2. Initiate spacecraft to ground file transfer by executing the ipfs-transmit-dag using file CID 3. Confirm Horizon LINUSS Ground logs show file payload telemetry transferring to ground 4. Confirm IPFS Ground Node logs show file blocks received from SmartSat Ground 5. Confirm file has successfully transferred to IPFS Ground Node by confirming CID match	CIDs match with no observable anomalies

Acknowledgments

This work was funded by a combination of Lockheed Martin Space, the Filecoin Foundation, The University of Alabama in Huntsville, and U.S. Army Space and Missile Defense Command.

Special thanks to the greater LINUSS team, whose critical support of ground and space operations made all these accomplishments possible, including David Barnhart, Karla Brown, Jesse Pedroni, Mitchell Moran, Adam Eveleth, Robert Templeton, Jared Garcia, and Alex Makowski.

The authors would also like to thank their colleagues Eric Bowler, Christian Morrow, and Shawn Nissley on the Lockheed Martin Horizon team for their contributions to the Filecoin Foundation – IPFS test campaign.

The authors would also like to thank their colleagues across Lockheed Martin National Security Space, Engineering & Technology, and Ignite for their support to these rapid technology development activities and reviewing the content of this paper, including Lance Williard, Jon Famous, David Shoemaker, Scott Messick, Joel Diebel, Kevin Clous, McKinley 'Mac' Weaver, and Aysha Rehman.

The authors would also like to thank their colleague Paul Choe for his contributions in developing Small Satellite Defender and support throughout ground and on-orbit testing.

Finally, the authors would like to thank our collaborating subcontractors: Swedish Space Corporation (SSC), Terran Orbital, and Innoflight.

References

[1] Manning, C. G., "Technology Readiness Levels - NASA," *NASA*, 27 September 2023, [online publication], URL: https://www.nasa.gov/directorates/somd/space-communications-navigation-program/technology-readiness-levels/ [retrieved 19 June 2024].

[2] "Lockheed Martin LINUSSTM Small Satellites Ready for 2021 Launch," *Lockheed Martin Newsroom*, 10 August 2021, [online publication], URL: https://news.lockheedmartin.com/linuss-small-sats-mission [retrieved 19 June 2024].

[3] "Lockheed Martin CubeSats Successfully Validate Essential Maneuvers for On-Orbit Servicing," Lockheed Martin Newsroom, 18 April 2023, [online publication], URL: https://news.lockheedmartin.com/2023-04-18-Lockheed-Martin-CubeSats-Successfully-Validate-Essential-Maneuvers-for-On-orbit-Servicing [retrieved 19 June 2024].

[4] "Risk Classification for NASA Payloads," NASA NPR 8705.4A, April 2021.

[5] "Lockheed Martin's First Smart Satellites Are Tiny with Big Missions," Lockheed Martin Newsroom, 20 March 2019, [online publication], URL: https://news.lockheedmartin.com/2019-03-20-Lockheed-Martins-First-Smart-Satellites-are-Tiny-with-Big-Missions [retrieved 19 June 2024].

[6] "Now Installing Update… SmartSat-Equipped Satellite Uploads New Mission On-Orbit," Lockheed Martin News Hub, 15 January 2024, [online publication], URL: https://www.lockheedmartin.com/en-us/news/features/2024/smartsat-equipped-satellite-uploads-new-mission-on-orbit.html [retrieved 19 June 2024].

[7] Barnhart, D. J., Shoemaker, D. M., King, E. T., Logue, T., and Lavis, M. J., "LM LINUSSTM - Lockheed Martin In-Space Upgrade Servicing System," Small Satellite Paper SSC23-XI-04, August 2023.

[8] Barnhart, D. J., Shoemaker, D. M., King, E. T., Logue, T., and Lavis, M. J., "Lockheed Martin's In-space Upgrade Satellite System (LM LINUSSTM)," Small Satellite Conference Presentation, SSC23-XI-04, Logan, UT, August 2023, URL: https://digitalcommons.usu.edu/smallsat/2023/all2023/122/ [retrieved 19 June 2024].

[9] Cutter, R. J., Maple, P., Janisch, B. S., Bishop, B. C., "Lockheed Martin LINUSS Mission Review," American Astronautical Society, Guidance, Navigation, & Control Paper AAS-24-141 (to be published).

[10] Hoffman, C., "Understanding the Load Average on Linux and Other Unix-like Systems," How-To Geek, 15 August 2014, [online article], URL: https://www.howtogeek.com/194642/understanding-the-load-average-on-linux-and-other-unix-like-systems/ [retrieved 19 June 2024].

[11] "Lockheed Martin Opens Futuristic Satellite Operations Center Test Bed," Lockheed Martin Newsroom, 11 September 2023, [online publication], URL: https://news.lockheedmartin.com/2023-09-11-Lockheed-Martin-Opens-Futuristic-Satellite-Operations-Center-Test-Bed [retrieved 19 June 2024].

[12] "Lockheed Martin's Pony Express 2 Tech Demo Satellites Declared Ready for Launch," Lockheed Martin Newsroom, 15 February 2024, [online publication], URL: https://news.lockheedmartin.com/news-releases?item=129461 [retrieved 19 June 2024].