A Distributed Data Center in Space (DCiS) Architecture for Small Satellites

Dan Peleg, Ran Ginosar
Ramon.Space, Ltd., Israel
[dan.peleg, ran.ginosar]@ramon.space

David Anderson, Eric Anden
Ramon.Space, Inc., USA
[david.anderson, eric.anden]@ramon.space

Abstract: Space-based data processing has emerged as a critical need for modern satellite missions in Earth observation, communications, and defense. We present DCiS (Data Center in Space), a distributed micro-data-center architecture optimized for SmallSat-class spacecraft. By deploying many small computing nodes across an orbital constellation and interconnecting them via high-speed links, DCiS enables cloud-like computing capabilities in orbit. The proposed architecture emphasizes modular design, fault tolerance, and autonomous operation, bringing terrestrial data center principles into space. We discuss the system's design – including standardized compute modules, virtualization across satellites, and thermal/power management – and demonstrate how DCiS can achieve scalable performance while overcoming the unique constraints of the space environment. Preliminary analysis and industry developments suggest that such distributed space computing is not only feasible but offers advantages in latency, resilience, and security over traditional ground processing.

1. Introduction

This paper makes the following contributions:

1. Distributed SmallSat-native data center architecture.
2. Disaggregated node architecture (compute, storage, networking, sensor ingress, management).
3. Dynamic workload orchestration model.
4. Thermal-driven argument for distributed scaling.

The growing volume of data generated by satellites and the demand for real-time analytics have exposed the limitations of the traditional approach where spacecraft transmit raw data to Earth for processing. Bandwidth bottlenecks and ground station access windows create delays and data dropouts, hindering time-sensitive applications [1]. For example, missions that perform Earth observation or missile tracking cannot afford the latency of waiting for ground processing; onboard or in-orbit computation can enable immediate insights and responsiveness [1]. Furthermore, sending all data to Earth raises security concerns, as transmissions can be intercepted or tampered with; processing data on satellites reduces exposure to such threats [1]. These factors motivate a paradigm shift: moving compute and storage resources into space, closer to the data sources [1].

Recent technological advances have made in-space computing more realistic. Radiation-hardened processors and AI accelerators are now available for small satellite use, and optical inter-satellite links (ISLs) can provide high-bandwidth, low-latency networking across satellites [2]. Industry and government interest is growing: **LEOcloud**, for instance, has planned to deploy a prototype "orbital cloud" data center on the International Space Station by 2025 [3]. Visionaries have even speculated about large-scale space data centers – Sam Altman (OpenAI) suggested building massive clusters off Earth – though practical plans focus on smaller, distributed systems [4]. Academic studies support the distributed approach: Bleier et al. (2025) showed that a constellation of micro-data-centers can be more cost-effective and resilient than a single monolithic satellite, thanks to easier scaling and graceful degradation on failures [1]. Likewise, researchers at Google proposed a constellation of computing satellites hosting AI accelerators, arguing that many small satellites are preferable to an enormous in-orbit facility that would require on-orbit assembly [2].

In this paper, we propose DCiS (Data Center in Space) – an architecture that applies cloud data center concepts to networks of small satellites. Each satellite in a DCiS constellation serves as a micro data center node, equipped with general-purpose and specialized processors, local solid-state storage, and high-speed communication links. These nodes collectively form a distributed computing fabric in orbit, managed with techniques analogous to terrestrial cloud orchestration (containerization, distributed file systems, etc.), but adapted for the intermittent connectivity and radiation environment of space. We describe the DCiS architecture in detail, including hardware modules, network topology, software stack, and fault tolerance strategies. We then examine key challenges for deploying such a system on SmallSat-class platforms, especially in terms of power supply, thermal control, and mass constraints. Finally, we outline potential applications of DCiS (from real-time remote sensing analytics to on-orbit AI inferencing for autonomous spacecraft) and discuss the roadmap for scaling this concept in future space infrastructures.

From Data Handling Systems to Spaceborne Data Centers: Traditional satellite architectures rely on tightly coupled data handling systems. DCiS shifts to infrastructure-centric computing, decoupling sensing from processing and enabling constellation-wide resource sharing.

2. DCiS Architecture Overview

Design Principles: The DCiS architecture is guided by modularity, distribution, and autonomy. Rather than one large satellite carrying a supercomputer, DCiS is composed of many small compute nodes distributed across one or multiple orbits. This inherently eliminates any single point of failure – the loss of one node only degrades capacity modestly, and the system gracefully reduces its service rather than failing entirely. The design is incrementally scalable: capacity can be increased by launching additional nodes over time, and older nodes can be upgraded or deorbited without a complete system redesign. Autonomous operation is crucial, since real-time human control is impractical; each node can manage itself and cooperate with neighbors to allocate tasks and recover from faults.

System Components: Figure 1 illustrates the conceptual architecture of DCiS, highlighting the network that links them.

Each DCiS node (satellite) comprises several subsystems analogous to a terrestrial data center:

- **Sensor Ingress Subsystem**: Interfaces for EO, SAR, hyperspectral and other sensors.
- **Compute Subsystem**: A set of processors (CPUs plus GPUs or FPGAs) capable of executing workloads such as AI inference, image processing, and data compression. These are radiation-hardened or radiation-tolerant chips that provide high performance per watt.

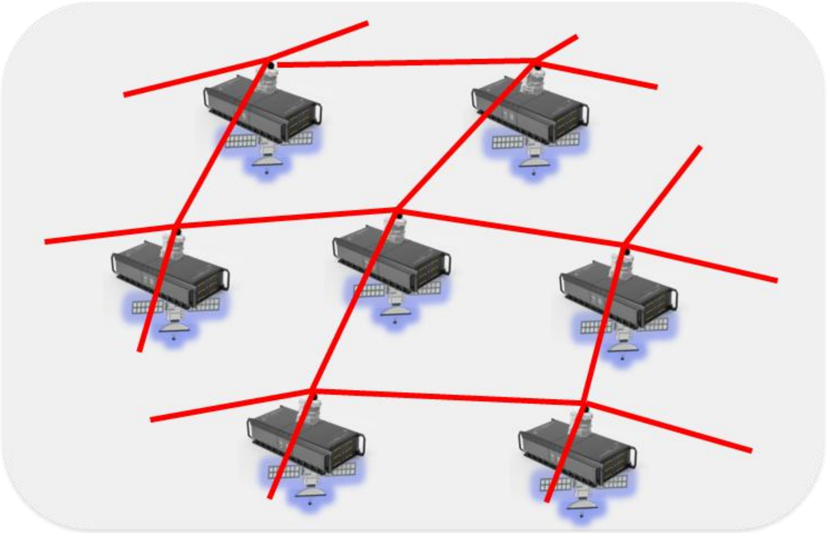

Figure 1: Conceptual DCiS architecture showing interconnected satellite nodes.

- **Storage Subsystem:** High-density non-volatile memory (e.g. radiation-tolerant SSDs) for storing large datasets and intermediate results. Data is maintained with redundancy and error correction to handle radiation-induced bit flips.
- **Networking/Communication:** Each node has an inter-satellite communication module (laser optical links or Ka/V-band RF) to exchange data with other nodes, forming a space-based network. This is supplemented by downlink radios to communicate with ground stations when needed to deliver final data products or accept new tasking.
- **Power and Thermal Management**: Solar panels and batteries provide power to the node. Waste heat from the compute electronics is removed via radiators. These subsystems are tightly integrated with the computing elements to monitor and adjust power usage and prevent overheating.
- **Attitude Control and Platform Bus:** As with any satellite, the node includes attitude determination and control (ADCS), propulsion (for orbit maintenance), and a structural bus. These ensure the satellite can point its antennas or laser links accurately and maintain its orbital position relative to other nodes.

Within each node, a high-speed **intra-node network** (similar to a backplane or local Ethernet/PCIe interconnect) links the compute, storage, and communication units, enabling them to function as a cohesive "server". The entire constellation of nodes then forms a network of these servers in space – an **orbital cluster** or cloud.

Modular Building Blocks: To accommodate different mission scales and satellite sizes, DCiS employs a modular hardware strategy. Computing elements can be combined to achieve the desired performance within a given satellite bus's constraints. For example, one commercially available solution for space computing is Ramon.Space's product line, which includes the **NuPod** and **NuBox** modules [5]. A **NuPod** is a compact high-performance computing module (with integrated multi-terabyte storage) designed to perform AI/ML processing on board small satellites [5]. A **NuBox**, on the other hand, is a larger multi-service computing platform that can host multiple processing units and also handle communications tasks [5]. In the DCiS architecture, we envision a similar hierarchy of modules:

- **DCiS-Pod**: a self-contained compute/storage module (analogous to NuPod) that delivers a certain compute capacity (for example, tens of TOPS of AI performance) and fits within a small satellite form factor (tens of kilograms, drawing on the order of a hundred watts).
- **DCiS-Box**: a larger assembly (analogous to NuBox) integrating multiple Pods and additional networking interfaces or specialized accelerators. A single satellite might carry one DCiS-Box or a few, depending on its size and power budget.
- **Constellation Fabric**: the network of DCiS-Boxes across many satellites forms the top level. The constellation can be homogeneous (all nodes identical) or heterogeneous (different performance tiers of nodes), but they all communicate with common protocols and collectively provide the DCiS cloud service.

This modular approach allows scaling: a CubeSat or small SmallSat might carry just one DCiS-Pod, whereas a larger satellite (or space station module) could host a full DCiS-Box with multiple pods. Figure 2 depicts this concept of scaling building blocks across different power envelopes and satellite sizes – from a tiny “compute pod” up to a full “DCiS Box” assembly. By standardizing interfaces (power, data, thermal coupling), these modules can be integrated into various platforms, making the architecture flexible and upgradable. Notably, the DCiS modules are designed with radiation shielding and fault containment in mind: if one module fails or suffers a radiation upset, it can be isolated without bringing down the entire node.

Figure 2: Modular scaling from DCiS-Pod to DCiS-Box to constellation.

Node Internal Architecture: The internal functional decomposition of a single DCiS node is illustrated in Figure 3.

<table>
<tr><td colspan="2">Management Plane</td></tr>
<tr><td colspan="2">Networking Plane</td></tr>
<tr><td colspan="2">Compute Plane</td></tr>
<tr><td>Storage Plane</td><td>Sensor Ingress</td></tr>
</table>

Figure 3: Internal node architecture with functional planes.

We organize the node’s functions into separate planes:

- The **sensor ingress plane** handles acquisition and preprocessing of data.
- The **compute plane** consists of the processors/accelerators executing application workloads.
- The **storage plane** manages local data persistence, including buffering incoming sensor data and caching data from other nodes.
- The **networking plane** handles intra- and inter-satellite communications (routing data between satellites, managing the high-speed ISL links, etc.).
- The **management/control plane** includes the on-board controller that orchestrates the node’s operation, health monitoring, fault management, and interfaces with the constellation-wide orchestration system.

Within a node, these planes interact via well-defined APIs. For instance, an application running in the compute plane might request a data set, which the storage plane fetches either from local storage or by streaming it from a peer node via the networking plane. The management plane monitors telemetry (power levels, thermal sensors, link status) and can instruct the compute plane to throttle or migrate tasks if, say, the node is overheating or running low on power. This design mirrors terrestrial data center architectures (compute, storage, network subsystems with a management overlay) but is tailored for autonomous operation in space and reliability under radiation. All critical components are either inherently radiation-hard or are backed by redundancy and error correction (e.g., ECC memory, watchdog processors, and in some cases triple-modular redundancy for critical computing elements).

Distributed Network Topology: Each DCiS node is interconnected in an orbital mesh network. In a simple DCiS deployment, all satellites might be in the same orbital plane (e.g., a “string of pearls” in LEO) with neighbors communicating via direct ISLs. More complex deployments could span multiple orbital planes with inter-plane links. Figure 4 shows, conceptually, a constellation-level virtual network where multiple satellites form a single logical compute fabric. The topology may be configured as a full mesh or a clustered mesh. Because satellites move relative to each other (especially in LEO), the network is dynamic – links appear and disappear as nodes come in and out of line-of-sight. Therefore, the DCiS network employs a form of **delay-tolerant networking** for reliability (store-and-forward techniques for data), while maintaining as many persistent high-bandwidth links as possible for low-latency peer-to-peer communication.

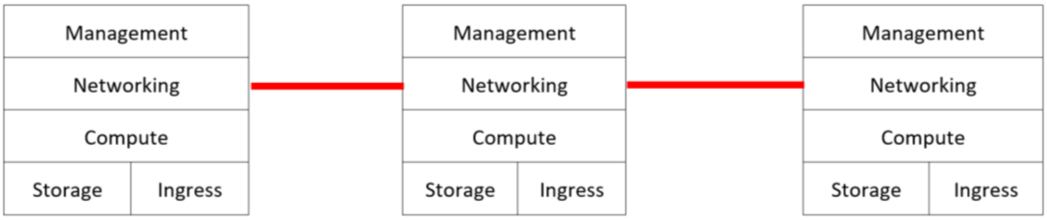

Figure 4: Constellation-level virtualization.

Key network parameters include the link bandwidth and latency. Today’s optical ISLs offer on the order of 10–100 Gbps throughput [2], and future systems aim for even higher data rates (potentially via WDM optical links or inter-satellite laser networks with aggregate tens of Tbps capacity [6]). Latency is governed by the distance between satellites; within a cluster of nearby satellites (tens of kilometers apart or less), one-way light propagation is only microseconds to milliseconds. DCiS leverages this by scheduling tightly-coupled computing tasks on satellites that are close together (for minimal communication delay) [2]. Meanwhile, more independent tasks can be assigned across farther nodes. The network layer handles **adaptive routing** – if one link goes down (due to satellite eclipse or pointing issues), traffic is rerouted through alternate paths.

All inter-node communications are encrypted end-to-end for security. Techniques like frequency hopping or laser beam agility help mitigate jamming and interference. The network layer also performs **resource discovery**: nodes advertise their available compute and storage resources to the rest of the constellation. This way, the orchestration system knows where there is spare capacity or where specific data resides.

Software Stack and Virtualization: On top of the physical infrastructure, DCiS runs a distributed software framework providing a *cloud-like* environment. Each node hosts a lightweight virtualization layer (e.g., container runtimes or a partitioned hypervisor) to isolate and manage workloads. A distributed orchestration service (similar to Kubernetes or a federated scheduler) runs across the constellation to coordinate these workloads. The result is that the collection of satellites behaves as a single **cloud platform** from the end-user's perspective.

This concept of **constellation-level virtualization** means that a user or mission application does not need to interact with individual satellites; instead, they submit jobs or data to the DCiS as a whole, and the system allocates resources internally. Figure 4 illustrates how multiple satellites appear as one logical computing pool. Enabling this requires several software components: a **distributed file system** (so that any node can access data regardless of which node originally collected or stored it), a **distributed database/state store** for keeping track of task assignments and metadata, and consensus algorithms to keep the system coordinated despite communication delays. We implement these with space-specific optimizations – for example, using gossip protocols for state sharing which can tolerate high latencies, and partition-tolerant consensus that allows the network to temporarily operate in independent segments if connectivity is lost.

In summary, the DCiS architecture brings together modular **hardware building blocks**, a resilient **network fabric**, and a cloud-inspired **software layer** to create a scalable in-orbit computing platform. We next discuss how such a system operates and handles the challenges of the space environment.

3. Operational Concepts and Fault Tolerance

One of the key advantages of DCiS is enabling new Concept of Operations (CONOPS) for space missions. In traditional missions, each satellite operates largely independently with its own dedicated payload and limited processing. In a DCiS-enabled mission, satellites collaborate—sharing both data and computing tasks. Several operational scenarios highlight the benefits:

- **Dynamic Task Allocation**: Instead of pre-defining which satellite will process which data, the DCiS network dynamically allocates tasks based on current resource availability and mission priorities. For instance, if a natural disaster is detected in imagery by one satellite, a heavier image processing task (e.g., detailed change detection using AI) can be offloaded to a neighboring satellite that has spare compute capacity, while the first satellite continues collecting new data. This ensures critical workloads get the necessary resources without manual intervention or waiting for ground downlink.

- **Load Balancing and Migration**: Workloads (such as AI inference pipelines) can be moved from one node to another in response to changing conditions. If a satellite enters an eclipse (losing solar input and running on battery), it might shed non-critical workloads to other nodes that are in sunlight. Similarly, if a node's thermal sensors indicate approaching limits, the system can migrate some tasks away to reduce the heat generation on that satellite. This is analogous to load balancing in a terrestrial data center – DCiS implements it in orbit, maintaining optimal performance without human command.
- **Data Caching and Redundancy**: The DCiS can cache data across multiple satellites to reduce latency and increase resilience. For example, recent high-value imagery or sensor data might be stored on several nodes such that at least one satellite with that data is always in view of a ground station. If users request an older archive, the system retrieves it from whichever node currently has it (or re-computes it from source data). This distributed object storage model means no single satellite failure results in data loss; critical data is redundantly stored (via replication or erasure coding) across the constellation.
- **In-Situ Data Fusion**: Satellites with different sensors (imagers, synthetic aperture radar, signals intelligence, etc.) can send their raw data into the DCiS fabric for on-orbit fusion. Rather than downlinking all individual sensor streams to Earth for post-fusion, the satellites can combine them in space to produce enriched information (for example, correlating electro-optical and radar observations of the same target). The fused result, which is smaller and more meaningful, is then downlinked. This in-situ processing reduces bandwidth needs and can enable real-time multi-intelligence insights that were not possible before.

To support these operations, the system's software provides robust **fault tolerance** mechanisms. Space is a harsh environment – nodes can fail unpredictably due to radiation-induced errors, hardware malfunctions, or even micrometeoroid impacts. The DCiS approach assumes that **failures will happen** and is designed for resilience:

- **Node Failover**: When a node fails or becomes unresponsive, its neighbors and the network detect the loss (e.g., via missed heartbeat signals). The tasks that were running on the failed node are either restarted on other nodes or were already running redundantly. Critical workloads (especially persistent services or important AI analytics) can be replicated across multiple satellites so that if one goes down, another has a recent copy of the task's state and can take over. This can be done in an active-active fashion (multiple nodes process in parallel and one result is used) or active-passive (a standby node only activates if the primary fails), depending on the importance of the task and available bandwidth. In practice, less critical batch jobs might simply be restarted elsewhere after a failure, whereas time-critical services might be actively mirrored on two satellites.
- **Partition Tolerance**: Temporary network disruptions might split the constellation into sub-networks (partitions). For instance, if inter-plane communication is lost, each orbital

plane of satellites may form its own cluster until links are restored. DCiS's control plane uses decentralized coordination (borrowing concepts from RAFT/Paxos but adapted to high-latency links) to allow each partition to operate independently. Each partition continues handling local tasks, and when connectivity returns, the partitions synchronize their state. This ensures that a communications outage doesn't halt all computing – it merely localizes it until the network heals.

- **Graceful Degradation**: The system is designed to degrade gracefully under stress or component loss. If one or more nodes fail, overall capacity diminishes but the system continues to function. For example, if 10% of the satellites in a constellation unexpectedly go offline, the remaining 90% autonomously redistribute the orphaned tasks among themselves. Users might experience slightly longer processing times or reduced redundancy, but core services remain available. This is akin to how cloud services on Earth remain online despite server outages. In an extreme scenario, even if only a single satellite survives, it can still perform basic operations (using its own local compute) to support mission-critical functions, rather than a total mission loss.

- **Autonomous Fault Management**: Each DCiS node has onboard fault management that works in concert with the network. If a node experiences an anomaly (power surge, radiation storm causing multiple errors, etc.), it will attempt to enter a safe state and offload work before things get worse. For example, in a high-radiation event, a satellite might proactively checkpoint its running tasks to a neighbor and then temporarily shut down sensitive components to avoid damage. Once conditions normalize, it can recover and reintegrate into the network. The other nodes adjust seamlessly, as they are designed to handle dynamic changes in the membership of the constellation.

- **Cybersecurity**: The distributed nature and physical separation of nodes provide an inherent security advantage: there is no single "central" target to attack. Additionally, each node and inter-node link is secured via strong encryption (e.g., AES-256 or post-quantum algorithms for key exchange). If a potential cyber incident is detected (say a node starts behaving erratically or sending invalid data), the network can quarantine that node by revoking its credentials and routing around it until ground controllers can inspect the issue. This way, a compromised node (or one malfunctioning due to a fault) can be isolated without compromising the entire system.

Overall, the operational philosophy of DCiS is autonomy and resilience. The system should require minimal ground intervention, handling most anomalies on its own, and continue delivering computational services even in degraded states. This is crucial for applications like defense or deep-space operations where ground contact may be limited or jammed – the computing infrastructure in space must be self-reliant.

4. Power, Thermal, and Mass Considerations

Deploying data-center-class computing on small satellites poses significant engineering challenges, chiefly in power supply and thermal management. In terrestrial data centers, abundant power and cooling (HVAC, fans, water chillers) are available; in space, every watt is precious, and every joule of heat must be radiated away in vacuum.

Power Supply: Each DCiS node relies on solar power (photovoltaic arrays) with battery storage for eclipse periods. Small satellites have limited surface area for solar panels, which constrains available power. A 12U CubeSat might generate on the order of 50–100 W, while a larger 100-200 kg microsatellite with deployable panels could support 500 W to 1 kW of continuous power. By comparison, a single high-end server on Earth can easily consume 300–500 W or more. Thus, DCiS must be extremely power-efficient. This drives the use of specialized low-power chips (mobile-class SoCs, FPGAs, or ASIC accelerators optimized for specific workloads) and aggressive power management. Nodes schedule jobs not only based on compute availability but also power availability – if a satellite's batteries are running low or it's entering night, it will defer or offload energy-intensive tasks. Conversely, in orbits with constant sunlight (e.g., a dawn-dusk sun-synchronous orbit), a satellite can run near peak capacity continuously.

An oft-cited advantage of space-based computing is access to constant solar energy. Indeed, some proposals suggest orbital data centers could leverage 24/7 sunlight to eliminate dependence on terrestrial power grids [4]. With launch costs dropping to roughly $1,500 per kilogram [4], it becomes more feasible to loft satellites that are essentially "solar-powered servers" into orbit. In DCiS, we harness this by scaling the number of satellites to meet power needs: instead of one satellite trying to supply 10 kW (which is impractical for a smallsat), we use 10 satellites at 1 kW each, for example. Additionally, future advancements like space-based solar power beaming or ultra-lightweight solar arrays could further boost power availability. In the nearer term, high-efficiency multi-junction solar cells (≈30–35% efficient) and improved batteries (solid-state, radiation-tolerant) will incrementally improve the power budget for each node.

Thermal Management: Cooling is perhaps the most critical limiting factor for DCiS. In space, convection is not an option – all waste heat must be radiated as infrared energy. The capacity of a radiator is proportional to its area and the fourth power of its temperature (Stefan–Boltzmann law). For a rough sense of scale: a 1 m² radiator at ~45°C (with emissivity ~0.86) can dissipate on the order of 1 kW of heat in free space [1]. If a small satellite were to use, say, 500 W of power for computing, it might need a couple square meters of radiator surface area to maintain a stable temperature. This is a substantial fraction of a small satellite's size. Bleier et al. (2025) analyzed a 4 kW space micro-data-center and estimated that roughly a 4 m² radiator (total, both sides) was required to reject that heat load [1].

DCiS addresses thermal constraints through distribution and smart thermal control. By spreading computing across many satellites, we avoid creating a single hotspot that's unmanageable. Each node only needs to handle cooling for its share of the load. If a given node starts to approach its thermal limit, the orchestration software can throttle its processors or migrate tasks elsewhere before temperatures become dangerous.

We design DCiS nodes with high-temperature-tolerant components and, where possible, use thermal storage (heat sinks) to buffer transient peaks. In some cases, if a satellite is in eclipse and cooling is less efficient (since radiators might face Earth or other heat sources), the node can intentionally idle more until it's back in better conditions. Active thermal control (e.g., electric heat pumps to raise radiator temperature) can enhance radiator performance [1], though at a power cost. Radiators with adjustable emissivity are being researched [1] and could allow a satellite to radiate more when needed and less when it's in a cold condition (to avoid overcooling). These advanced techniques are on the horizon, but DCiS primarily relies on architectural choices (many nodes, moderate per-node power) to keep thermal demands within reasonable bounds. Thermal management ends up dictating a lot of the system design: it's a primary reason our architecture favors many small nodes over a single powerful node [2] – the latter would need an impractically large cooling system or high-temperature operation.

Mass and Mechanical Constraints: Every added component (processors, batteries, radiators) increases the mass of a satellite. DCiS nodes must fit within launch vehicle mass and volume constraints, and often we want to launch many at once (rideshare). There is a synergy (and trade-off) between compute capability and the support infrastructure mass. For instance, adding more processors means we also need to add more solar panel area (mass), more battery capacity (mass), and larger radiators (mass). These in turn might require a sturdier structure and more attitude control capability to handle the increased inertia and perturbation forces (e.g., solar radiation pressure on large panels). Prior analyses have noted that increasing radiator size not only adds mass but also impacts propulsion needs (for orbit maintenance or maneuvering) due to higher drag in low orbits and greater moment of inertia [1]. DCiS must account for this: there is a practical upper limit to how "big" (in terms of mass and size) a single smallsat can grow before it's no longer in the realm of small satellites. We mitigate this by capping the per-satellite power and using more satellites for greater capacity, as discussed.

Radiation protection also adds to mass. Compute electronics may be housed in radiation-shielded enclosures (e.g., aluminum or polyethylene boxes) to reduce the total ionizing dose they receive over the mission life. Typically, a few millimeters of aluminum can attenuate a lot of charged particle radiation, but at the cost of several kilograms if the box is large. Engineers must balance shielding mass against the inherent radiation tolerance of the components and the acceptable error rates. DCiS leverages the fact that the system as a whole can tolerate individual upsets (through error-correcting codes and redundancy) – so we can often use COTS or slightly hardened components with moderate shielding rather than extremely heavy shielding of a fully radiation-impervious system.

In summary, **power and thermal budgets** are the key drivers of the DCiS design. Our architecture tackles these by distributing workload (to stay within what each satellite can handle) and by actively managing resources (power throttling, task migration for thermal control). The viability of DCiS on small satellites is supported by trends in technology – solar panels and batteries improving, electronics getting more efficient – and by a paradigm shift to using *many satellites working together* rather than one extremely complex satellite. The next section looks at what such a distributed space computing network can do in practice.

5. Applications

A space-based data center network like DCiS unlocks a range of new capabilities for both commercial and government space missions:

- **Earth Observation & Disaster Response**: Satellites imaging the Earth can process imagery immediately to detect changes (floods, wildfires, deforestation) and generate alerts within minutes rather than waiting for ground processing. For instance, a DCiS constellation could run large-scale AI models on imagery to identify structures damaged after an earthquake and downlink just the critical alerts to first responders. This reduces data volume (only "interesting" results are sent to Earth) and dramatically cuts latency for emergency response. Such real-time onboard analytics have been demonstrated in limited form on specialized missions; DCiS would make them routine across many satellites.

- **Intelligence, Surveillance, and Reconnaissance (ISR):** In defense and security applications, a network of satellites can collaboratively track moving targets (vehicles, missiles) by sharing observations and running sensor fusion algorithms on orbit. Multiple satellites could pool their sensor data (electro-optical, infrared, radar) within the DCiS network to continuously track an object, handing off the task as the target moves through different satellites' fields of view. The low latency of on-orbit processing is a game-changer for time-critical threat detection – decisions (like cueing another sensor or initiating intercepts) can be made within the theater of operations, without waiting for data to travel to Earth and back.
- **Secure Communications & Routing**: Satellites can serve not just as relay bent-pipes but as smart routers and processors for communications. For example, a tactical user could uplink raw encrypted data to the DCiS cloud in space; there, the data is decrypted and processed (perhaps translating formats or filtering content), then re-encrypted and downlinked to another theater or user. Because the processing (and temporary data storage) occurs entirely in orbit, it can be more secure and resilient. Even if ground infrastructure is compromised or unavailable, the space-based cloud keeps critical communications and computing going above it all. This concept also applies to internet services delivered from space – DCiS nodes could host edge computing for satellite internet constellations, caching popular content or doing local compute for connected devices to reduce latency.
- **Autonomous Constellation Operations**: As satellite networks grow in size (dozens to thousands of satellites), managing them from the ground in real-time becomes impractical. DCiS can enable the constellation to manage itself. The satellites can collectively run algorithms to optimize network configuration, perform collision avoidance calculations, and schedule their activities without constant ground commands. For example, if one satellite in a train detects a possible conjunction (collision risk) with a piece of debris, it can use DCiS resources to quickly simulate avoidance maneuvers and coordinate with neighboring satellites to ensure that any maneuver won't cause chain reactions. The plan can be executed autonomously, with ground stations simply informed after the fact. This level of autonomy will be essential for "self-driving" satellite constellations in the future.
- **Deep Space and Lunar Support:** While our focus is on Earth orbits, the same distributed computing concept can extend to cislunar space or beyond. A cluster of DCiS-enabled smallsats around the Moon could provide computing services for lunar surface missions (e.g., processing data for a rover or acting as an edge cloud for a lunar habitat where local computing is limited). In fact, companies are already exploring off-Earth data centers: for example, Lonestar Data Holdings has experimented with the idea of lunar data storage and computing, even attempting to send a prototype data server to the Moon (carrying some data files like an archival music recording) — though the lander carrying it crashed during descent [4]. DCiS constellations around the Moon or Mars could one day handle data routing and processing for an entire planet's exploration infrastructure, reducing the reliance on Earth-based computing.

- **Scientific Data Processing**: Space science missions often generate more data than they can downlink, or require real-time data analysis to make decisions. A DCiS network in orbit could serve as a *science processing hub*. For instance, a network of gamma-ray detectors across multiple small satellites could stream their readings into the DCiS cloud to correlate signals and detect a GRB (gamma-ray burst) in real time, then trigger other observatories. Similarly, a space telescope capturing large images could offload the heavy image processing (like stacking or filtering) to nearby DCiS nodes, receiving back a cleaned or annotated image, thus closing the loop faster. By doing this in orbit, scientists get results sooner and can adapt observations on the fly (which is especially valuable for transient or evolving phenomena).

In all these cases, the common thread is that **moving compute closer to the source of data** opens up possibilities that were previously impossible or impractical. It reduces reliance on bandwidth and ground infrastructure, and it enables faster, smarter decisions in space. Some of these capabilities (like real-time disaster monitoring) have clear societal benefit, while others (like autonomous military satellite operations) address emerging strategic needs. DCiS essentially transforms a constellation of smallsats into a **spaceborne supercomputer** with wide-reaching impact across domains.

6. Conclusion

We have presented DCiS, a distributed micro data center architecture for small satellites, and argued that such an approach is both feasible and advantageous for the next generation of space missions. By shifting from the traditional payload-centric model (where computing is a minor onboard function) to an infrastructure-centric model (where each satellite devotes significant resources to computing and networking), DCiS enables a space-based cloud computing paradigm. This paradigm promises significantly reduced latency, improved resilience to failures, and enhanced security for handling sensitive data entirely above the atmosphere.

Through a modular design (illustrated by concepts like NuPod and NuBox building blocks) and a robust distributed software stack, DCiS can scale from a handful of satellites to hundreds, providing incremental growth in capability. Importantly, our architecture acknowledges and addresses the physical constraints of the space environment – particularly power and thermal limits – by spreading out the load and incorporating intelligent resource management. The result is a system that can grow in aggregate performance without encountering a single “thermal wall” or a single-point catastrophic failure.

In conclusion, DCiS offers a blueprint for **bringing the cloud to orbit.** The concept is no longer science fiction: it builds on proven technologies (distributed computing, small satellite constellations, space-qualified processors) and addresses urgent needs for on-site processing in space. By networking many small satellites into a cohesive computing cluster, we gain a platform that is greater than the sum of its parts – a flexible, upgradable, and resilient **spaceborne computing infrastructure** that can empower a new era of smart satellite services. The implications span from faster response to Earthly events, to more autonomous space operations, to entirely new commercial services delivered from orbit. As we look ahead, the Data Center in Space may become as indispensable to space missions as ground data centers are on Earth, heralding a transformation in how we utilize the final frontier.

References

[1] R. Bleier et al., “Architecting Space Microdatacenters: A System-level Approach,” Proc. HPCA, 2025.

[2] B. Agüera y Arcas et al., “Towards a future space-based, highly scalable AI infrastructure system design,” arXiv:2511.19468, 2025.

[3] “LEOcloud to demonstrate space-based data center onboard ISS,” 2024.

[4] S. Hurwitz, “Big Tech dreams of putting data centers in space,” WIRED, 2025.

[5] Online: https://ramon.space/product/