XIPHOS
AN EPIQ SOLUTIONS COMPANY

ON-ORBIT AI IS NOT READY - ACT NOW!

Why hardware decisions made today determine whether tomorrow's space missions can use AI at scale

Summary

On-orbit AI promises faster decisions, lower down-link burden, and scalable autonomy — but most missions are not designed with the compute, power, memory, and software flexibility needed to keep up with rapidly evolving AI models. The core mismatch is timeline: spacecraft hardware is chosen years before launch and must last 5+ years, while AI techniques and models can change in weeks or months.

The practical response is not to "pick the perfect accelerator," but to architect for adaptability: modular compute, open software environments, and clear trade decisions matched to platform constraints (CubeSat vs SmallSat vs station-class hosts).

Where On-Orbit AI Creates Immediate Mission Value

The highest-ROI near-term uses are those that reduce bandwidth and latency: pre-filtering imagery, event detection, feature recognition, and "tip-and-cue" behaviors where a detection triggers follow-on imaging or tasking. These behaviors turn a satellite from a passive sensor into an active participant in the mission loop. Example are shown in **Figure 1**. Wildfire detection is an obvious case where waiting for a human operator to start the next shift before detection occurs is far from ideal and better recognized real-time.

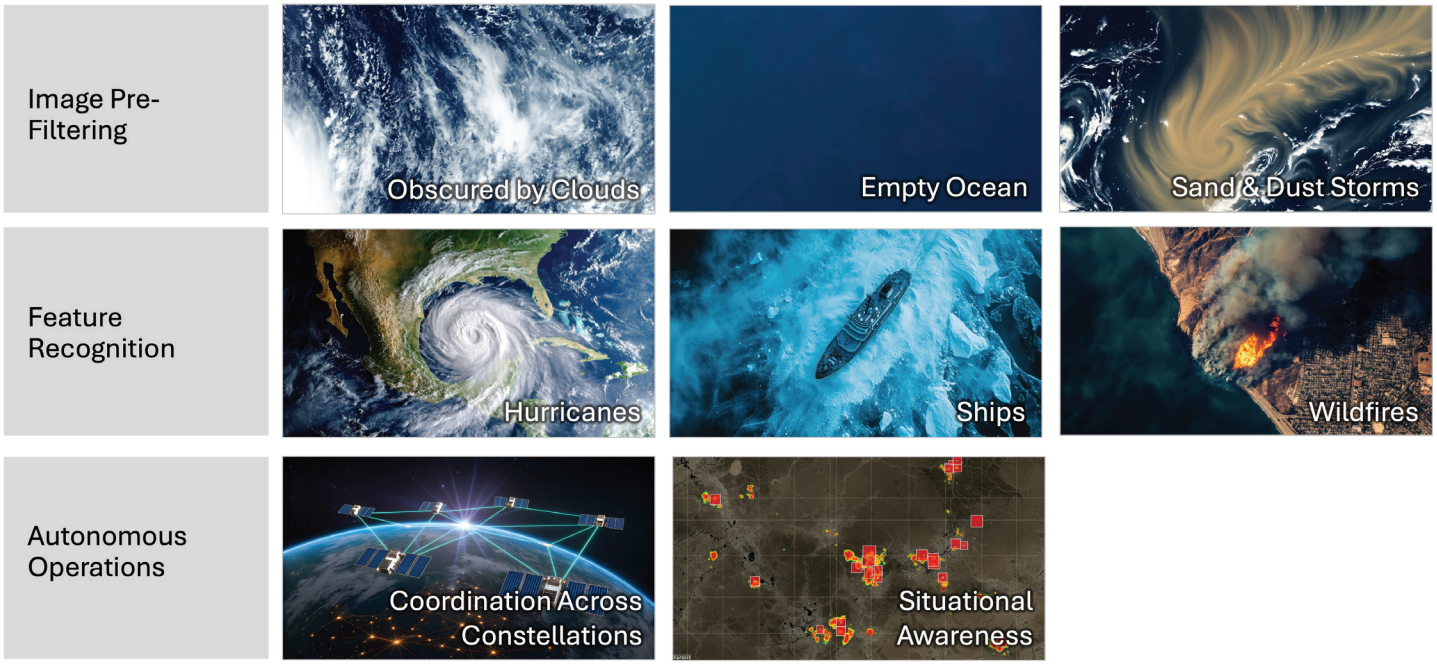


Figure 1: *Example use cases benefiting from AI detection, ranging from identification and pre-filtering of images containing no useful data, to subject recognition, to autonomous coordination between satellites in a constellation.*

A Concrete Example: Earth-Observation Latency and Downlink Waste

In a conventional workflow, a satellite collects data, down-links it, the ground segment processes it, and humans or downstream software decide what matters — often hours later. If the mission objective is to find rare objects (e.g., ships in a wide ocean scene), most of the data moved is irrelevant. This is illustrated in **Figure 2.**

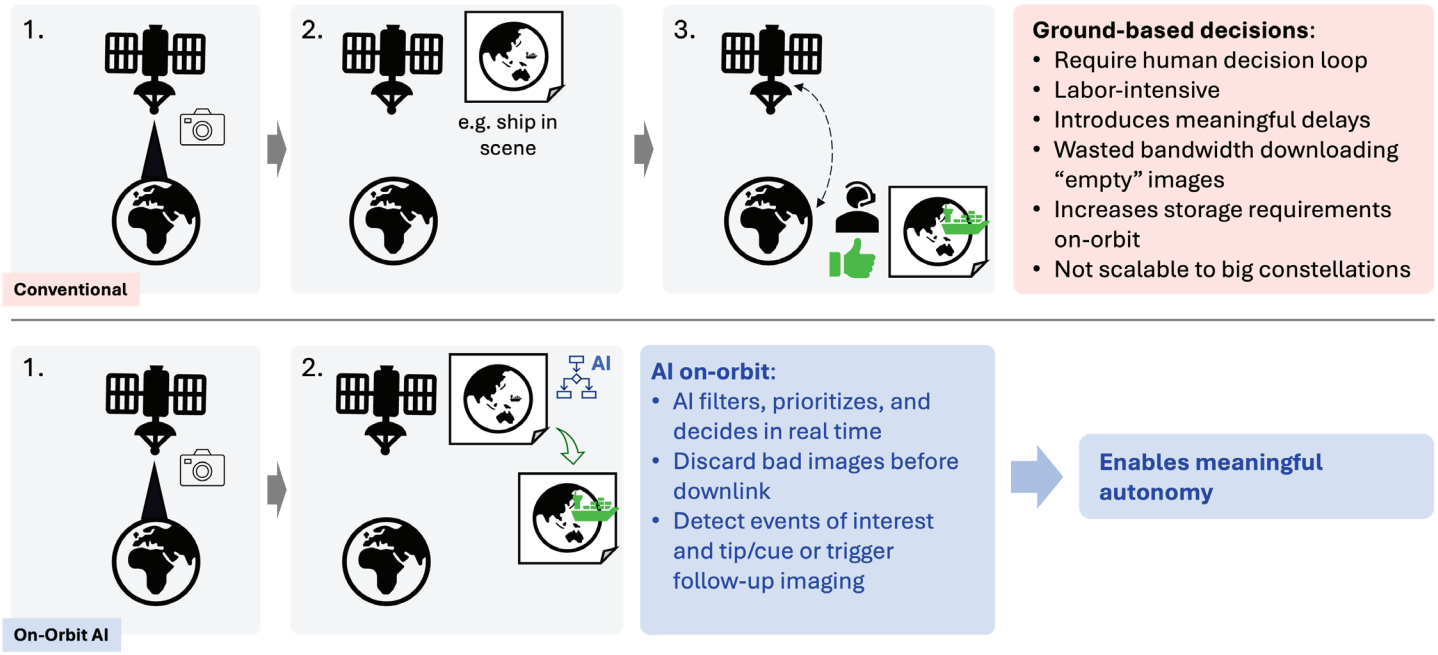


Figure 2: *Illustrated comparison of earth observation focused on identifying shipping, comparing conventional and on-orbit AI-enabled approaches*

Putting inference on-orbit enables rapid triage: detect the object, crop/flag the relevant region, and either down-link only what matters or immediately request additional observations. The result is less wasted bandwidth, less time-to-action, and a workflow that scales beyond "human-in-the-loop."

Scaling Beyond One Spacecraft: Autonomy Across Constellations

Constellations magnify both the opportunity and the problem. If one spacecraft can autonomously detect a target and cue another asset for confirmation or higher-resolution imaging, you gain coverage and persistence. But doing this at scale requires that decision-making happens in space; pushing every image and every decision to the ground becomes operationally impractical.

Why "Act Now" is a Hardware Problem as Much as an AI Problem

Mission teams often start with the algorithm and ask what hardware is needed. In space, the order must invert: you need a compute platform that can survive the mission lifetime and remain useful as AI evolves. Hardware selection and qualification are long-lead activities; waiting until AI is "ready" risks launching a system that can't run the models you'll want two years from now.

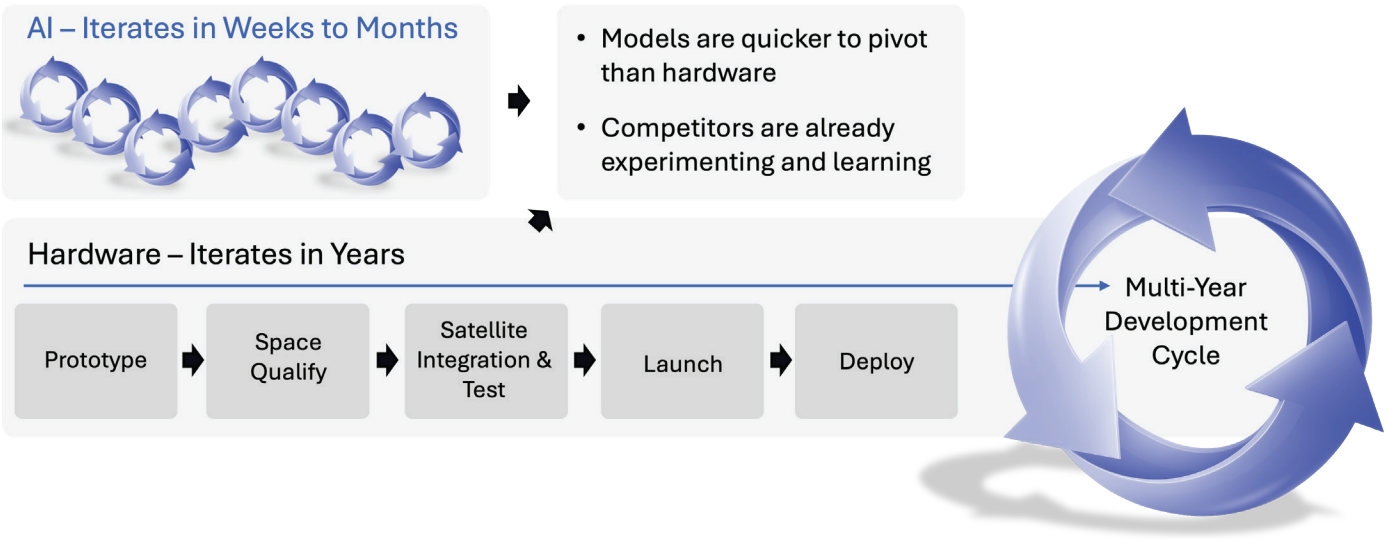


Figure 3: *AI models and on-orbit hardware have very different iteration timescales*

A pragmatic strategy is to design for change.

- Deploy hardware that can survive for 5+ years on orbit
- Use an open standards approach vs. a black box
- Lean forward on interfaces that are not yet common in space and leverage FPGA that can be re-configured to implement interfaces that don't even exist yet
- Use modular architecture where hardware can evolve with daughter board update without re-baselining a whole new compute platform

Platform-Driven Trade-Offs

Constraints vary dramatically by host platform, as illustrated in **Figure 4**.

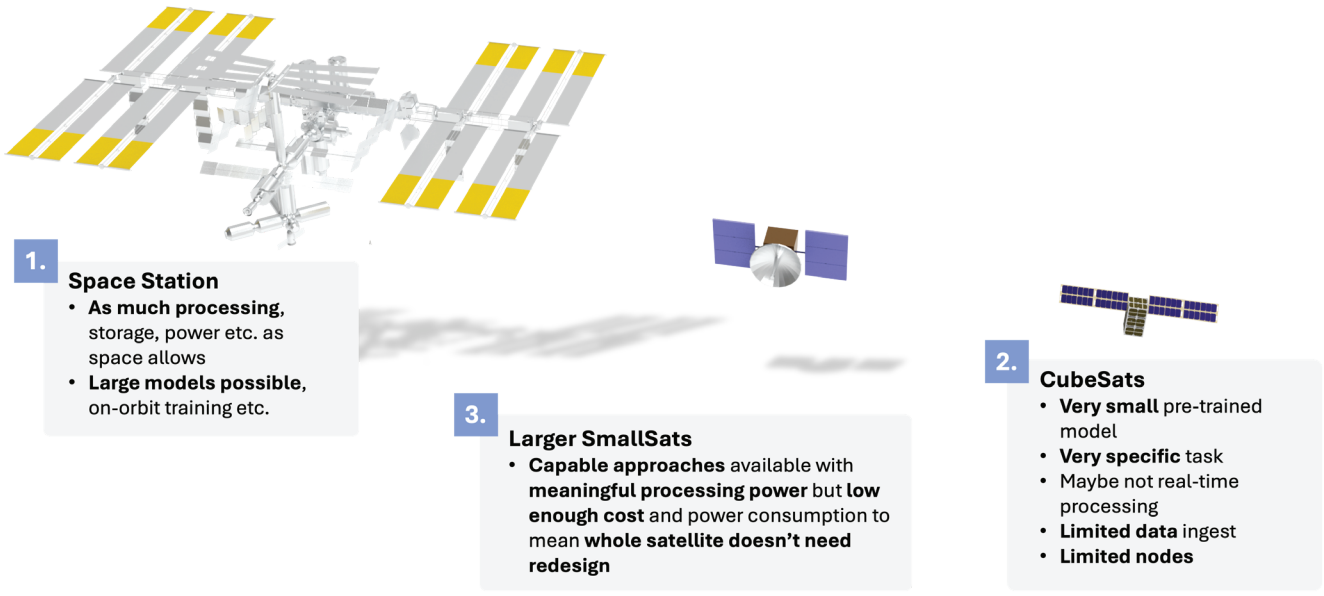


Figure 4: *The capacity for different platforms to host AI/ML models varies widely*

- Station-class payloads offer generous power/thermal/headroom, enabling richer AI compute.
- SmallSats often represent the “sweet spot” where meaningful AI compute is possible but must be carefully balanced against SWaP.
- CubeSats can run AI, but every watt and cubic centimeter forces a tougher trade: model size, memory bandwidth, and accelerator selection must be deliberate.

This is why there is no universally “best” accelerator — only the best fit for the platform and the mission objective.

Example Hosts for On-Orbit AI/ ML

It is a common misconception that AI/ ML can only be run on GPUs. Three example hosts-types are discussed below:

- **FPGA**: Good deterministic performance and efficiency but require upfront hardware optimization
- **CPU**: Flexibility but are performance-limited, mostly small models
- **GPU**: Deliver the best raw AI inference performance, but at higher power/cost

An example of efficient FPGA use for AI comes from Zaitra who have used the **Xiphos Q8** as the basis for their Skaidock platform which provides on-orbit cloud pre-filtering and object detection, and has been deployed since 2024 (**Figure 5**).

Figure 5: *Example application where an AI/ML model is on-orbit and hosted on an FPGA (Credit: Zaitra)*

Another example, this time using the relatively small CPU of a **Xiphos Q7**, is given by NASA’s Starling mission which uses four CubeSats in low earth orbit test technologies for synchronized operation with resources on the ground and was launched in 2023. The mission includes autonomous onboard decision-making, distributed reactive operations, and distributed automated planning among the four CubeSats.

Given trends in terrestrial AI, there is obviously intense interest in more specialized AI/ ML-engines, and the next section looks at examples.

Example Mission Trade

This will use a specific example as context for the discussion:

- 5-year mission in LEO
- ESPA class satellite
- Multiple payloads including high data rate multi-spectral camera
- Execute large custom AI model on orbit
- Stretch goal - migrate to training on orbit

With this in mind, three approaches are examined: An Nvidia GPU SOM, AMD x86 Ryzen, and an AMD Versal with AI/ML engines. A comparison is shown in **Table 1**.

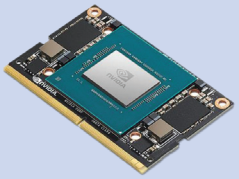
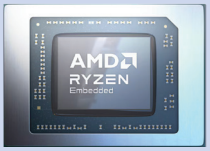
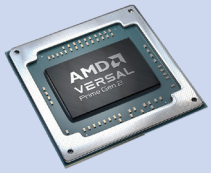
	Nvidia Orin NX GPU SOM 	**Ryzen 8000 Embedded SOM** 	**Versal 2602 with AI/ML Engines** 
CPU	ARM	x86	ARM
FPGA	No	No	Yes
AI Performance	78 TOPs	16 TOPs	101 TOPs
Radiation Performance	Only available as SOM with parts not designed for space	Limited public data, but sold as chip, so more flexibility in designing robust architecture around it	Fully tested, most radiation-robust Xilinx MPSoC to date
AI runtime	TensorRT	ONNX	Vitis AI or C++ kernels and ADF graphs on bare metal
AI Ease of use	Widespread community/framework support, considerable documentation and model library	Limited community/framework support, limited documentation and model library Due to lack of configurability, still easy	Somewhat limited community/framework support, less extensive documentation and model library Due to extensive configurability and access to bare metal, steeper learning curve

Table 1: *Comparison of device types for a SmallSat mission requiring significant AI/ML processing power*

A takeaway from **Table 1** is that each has its advantages, and no single solution is better under all circumstances.

- One differentiator is the presence of an on-chip FPGA, which might be relevant if the system needs a custom implementation of a particular interface – without it, the architecture will probably need an external hardware chip consuming space and power, which might be the right choice given our example is not radically power-limited.
- Another is in AI performance – there are many ways to measure this, but this comparison focuses on the throughput metric TOPS (Tera Operations Per Second). Both the Orin and Versal offer impressive processing power.
- Radiation is obviously important for a space mission, and there are clear differences here, too; the Nvidia is not available as a bare chip and is packaged with non-space qualified components whereas the Ryzen can be bought ready for third-party packaging for space; the Versal is ahead in this area.
- The final area of note is the ease of development and availability of tools for AI development – here the Nvidia is significantly ahead.

In each case, these platforms provide a springboard for experimentation and fast development while being upgradeable on-orbit as AI develops.

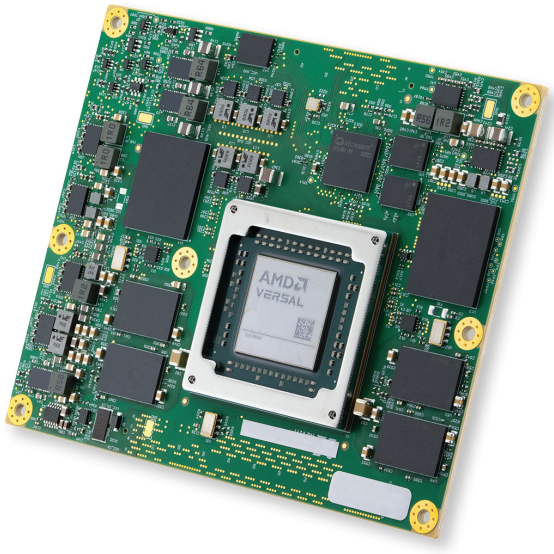
Call to Action: Treat AI as a Mission Capability, Not a Payload Add-On

If AI is expected to matter to a mission, it must be baked into design from day one. The most important decisions are:

1. Allocate realistic compute and power margins.
2. Choose architectures that support modular upgrades (hardware and software).
3. Prefer open, well-supported software stacks and avoid unnecessary lock-in.
4. Plan the verification/validation and update pipeline now — not after launch.

Waiting is riskier than acting: the cost of under-provisioned compute is paid for the entire mission life.

As discussed in this paper, Xiphos products are already on-orbit hosting AI/ ML applications. Our latest and most powerful platform is the **Xiphos Q9** *- more details here*



xiphos.com
sales@xiphos.com

3981 St-Laurent Boulevard, Suite 500,
Montreal, Quebec, H2W 1Y5, Canada

XIPHOS
AN EPIQ SOLUTIONS COMPANY