18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 83

A CLOUD BASED, SECURE AND COST-EFFICIENT PLATFORM FOR SPACE DERIVED VALUE PRODUCTION

Christoph Acham[a], Arne Nylund[b]*, Romain Poly[c]

[a] *Kongsberg Satellite Services AS, Prestvannveien 38, 9011 Tromsø, Norway*, mailto:christoph@ksat.no
[b] *Kongsberg Satellite Services AS, Prestvannveien 38, 9011 Tromsø, Norway*, mailto:arne@ksat.no
[c] *Kongsberg Satellite Services AS, Prestvannveien 38, 9011 Tromsø, Norway*, mailto:romain@ksat.no
* Corresponding Author

Abstract

Satellites generate an exponentially growing volume of downlinked data, and being able to produce and extract value from these data in a timely manner is increasingly important to unleash the full potential of the data. This necessitates an innovative, efficient, and simple management solution to handle data reception, processing and delivery. In this paper, we address this challenge by leveraging the maturity of public cloud technology.

Public cloud is a mature and versatile technology. It offers global accessibility, scalability, reliability, and redundancy, it addresses a substantial portion of the security challenge, with a minimum level of capital expenditure. Adoption of this technology keeps increasing, allowing organizations to focus more on their core business.

As an increasing volume of downlinked data is being moved to the public cloud, it is appealing to further carry out processing and delivery in that environment. A public cloud presence still necessitates substantial effort for associated maintenance and governance to fully benefit from the infrastructure. In addition, the growing focus on the security chain and cybersecurity, the perpetual development, and enhancements of public cloud functionality, call for continuous and demanding technical development of the staff overseeing the Cloud infrastructure.

This paper addresses how to develop a cost-efficient, secure, and governed cloud-based Platform as a Service (PaaS) to help users focus on their value generation activities and not having public cloud technical expertise. The paper will present a solution designed to simplify the ground segment operation by establishing a preconfigured and secure presence in the public cloud, that facilitates regular, automatic, and optimized processing of satellite data to enhanced value.

Keywords: PaaS, STAC, Analytics, Processing, Near-Realtime, Cloud-Native

Acronyms/Abbreviations

Application Programming Interface (API)
Cloud-Optimized GeoTIFF (COG)
Do It Yourself (DIY)
Earth Observation (EO)
Human User Interface (HMI)
Identity and Access Management (IAM)
Infrastructure As Code (IAC)
Kongsberg Satellite Services (KSAT)
Machine to Machine (M2M)
Open Geospatial Consortium (OGC)
Platform as a Service (PaaS)
Role Based ACcess (RBAC)
Spatio Temporal Asset Catalogue (STAC)
Syntetic Aperture Radar (SAR)
Virtual Private Cloud (VPC)

1 Introduction

The global space sector has undergone rapid transformations over the last years, with lower launch costs, introduction of small satellites, an increasingly number of EO (Earth Observation) satellites, and more commercial actors [1,2]. This growth may again fuel a surge in the volume, diversity, and velocity of data generated, paving the road to shifting expectations around how data is processed, accessed and monetized. Thus, we expect that organizations working with EO data will see a need to move more quickly from data reception to actionable value.

At the same time, public cloud technology has matured into a viable foundation for mission-critical workloads. It offers elastic scalability, robust global infrastructure, native security tooling, and a thriving ecosystem of services. A public cloud presence still needs to be treated as a Datacenter with the associated maintenance and governance activities. With an increased focus on cyber security, and with the continuous development and enhancements of public cloud functionality, continuous technical development of the staff responsible for the Datacenter is often needed.

1.1 Background and motivation

Moving our own processing to public cloud is enabling us to extend service offerings beyond what was previously possible. After data being received and processed at the ground station or on prem, the service often ended by making data available for customers, either by pulling or pushing. By sending data automatically to the "cloud" whenever received at our ground stations, we both "put" data closer to customers and provide a more complete service monitoring.

Based on the above, we started talking to different user groups, ranging from mission operators and data providers to government users and commercial analytics companies. We experienced that although demands where quite common, most solutions were DIY (Do It Yourself), and that there was a need for solutions that "just work", easy to use, cost-efficient, provide value fast, and scale seamlessly.

1.2 Purpose

One way of addressing this is a Platform-as-a-Service (PaaS) solution designed for mission operators within Earth Observation (EO), combining state-of-the-art cloud technologies with deep industry expertise to deliver scalable, secure, and flexible ground segment capabilities for EO data management and processing. Such solution is built to streamline the ingestion, processing, and delivery of EO data while ensuring robust security, cost-effective scalability, and integration options tailored to customer-specific needs.

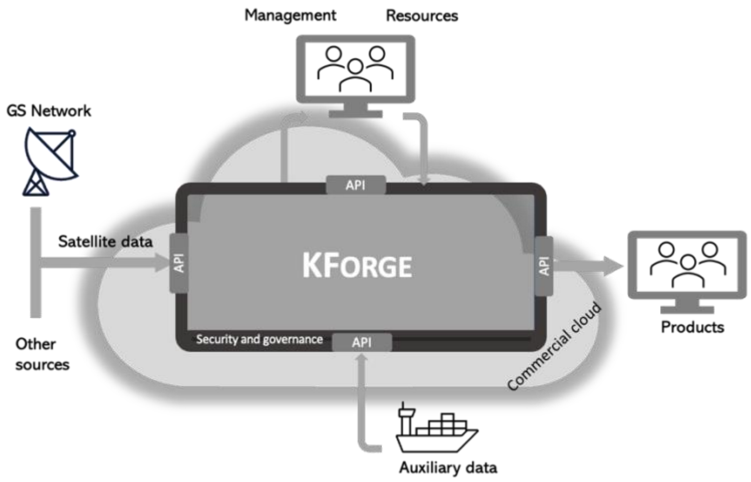


Figure 1 Top-level view of the proposed solution

This paper addresses our approach to simplifying EO value generation in the public cloud – KForge.

KForge will be a secure high velocity and high-volume satellite data system, focused on user-experience, and integrated with a satellite ground station network. It will include interfaces to allow for low latency transfer, storage, management, and processing of satellite data. It will run on a commercial cloud platform, compliant to industry security standards and able to inject large volumes of data with timely processing to end user products. It will be flexible with regards to processing functions (algorithms – embedded or provided by users) and the order in which these are invoked. It will leverage cloud-native patterns such as event-driven architecture, microservices, API-first design, and Zero Trust [11,12] security models, and enable users to focus on their unique value-generation logic, such as analytics and algorithm development.

This paper sets out to demonstrate how KForge meets its purpose through a set of architectural principles and business-aligned capabilities. We argue that the future of EO value extraction lies in platforms that are designed from the ground up to operate in cloud environments—not simply migrated there. KForge embodies this approach by combining KSAT's operational excellence in ground segment services with the transformative capabilities of public cloud infrastructure. By shifting the focus from infrastructure operations to value creation, KForge not only reduces time-to-insight but also lowers the barrier of entry for new users - enabling innovation, accelerating adoption, and ultimately amplifying the value of EO data across industries.

2 Architecture and Design

2.1 Design principles

KForge is grounded in pragmatic, cloud-era design principles that directly address the challenge described in our abstract: enabling secure, cost-efficient, and governed satellite data processing in the cloud – even for users without deep cloud expertise. By leaning into cloud-native patterns, modularity, and automation, KForge transforms the traditionally complex satellite ground segment into a pre-configured Platform-as-a-Service (PaaS) that accelerates value creation while reducing operational burden.

2.2 Cloud-native by design

KForge is architected for cloud-native execution from the outset – not retrofitted for it. This means that the platform builds directly on service offered by public cloud providers such as scalable storage, managed compute, native security, and telemetry tools. It does not rely on virtualized legacy models for custom infrastructure layers.

2.2.1 Cloud-native by default

KForge is built cloud-native from the ground up – not retrofitted. It avoids legacy virtualization models or lift-and-shift deployments. Instead, it uses services offered natively by public cloud providers such as scalable object storage, managed compute, identify and access managed (IAM), and telemetry.

As the Cloud Native Computing Foundation defines it, "Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds" [3]. This allows KForge to inherit key cloud benefits: elasticity, availability, cost-efficiency, and automation [4].

KForge also adopts cloud-native data-formats such as:

- Cloud-Optimized GeoTIFF (COG), which allows clients to request only the specific part of a raster file they need [5].
- SpatioTemporal Asset Catalog (STAC), which standardizes metadata and enables discovery of EO data across providers and cloud environments [6].
- GeoParquet, a high-performance tabular geospatial format optimized for cloud-scale analytics and I/O [7].

By using these formats, KForge avoids costly pre-processing or indexing layers, and can serve data efficiently to analytics pipelines without requiring gull downloads – critical when handling large volumes of EO data.

KForge also leverages cloud-native patterns:

- Event-driven workflows, which automatically trigger processing when new data arrives, enabling near-real-time reactions [8].
- Microservices, which isolate functions for independent scaling, resiliency, and faster iteration
- Infrastructure as Code (IaC), which provides consistent, governed deployments, reducing human error and configuration drift [9].

The combined result is a platform that operates with minimal human intervention, delivers high performance and resilience by design, and supports governance and scale from day one. As the Open Geospatial Consortium notes on cloud-native standards, "Cloud-native geospatial data, when utilizing formats and standards designed for the cloud, facilitates efficient access, processing, and management, particularly when leveraging cloud storage and potentially managed services for on-the-fly operations" [10]. This design philosophy enables users to extract value from their data pipelines without needing to build or manage the underlying infrastructure themselves.

EO data isn't constant – it arrives in bursts (e.g. satellite passes) or in peaks driven by events (e.g. natural disasters). KForge is designed to elastically scale compute and storage to match this variability. Processing is automatically distributed across available resources and scaled up or down based on real-time workload. This dynamic scaling is critical in EO platforms, where bursty and unpredictable data volumes require responsible systems that avoid idle infrastructure [11].

2.2.2 Abstracts cloud complexity for end users

KForge is explicitly built for users whose primary role is data processing, analytics, or mission execution – not infrastructure management. It is our experience that satellite operators, analysts, and even startups often lack the internal resources to manage IAM roles, VPCs, scaling policies, or compliance monitoring.

We abstracts these concerns through:

- Secure, high-level APIs and STAC-compliant metadata catalogs
- Declarative workflow definitions, letting users define "what" to run, not "how"
- UI or automation interfaces for status tracking, cost awareness, and control

Infrastructure is handled transparently – including storage lifecycle management, autoscaling, secret management, retry mechanism, and auditing. This ensures that organizations can remain focused on their domain expertise and delegate operational responsibilities to the platform.

As noted in the OGC's best practice for cloud-native systems: "Cloud-native applications promote user-centric workflows by abstracting operational complexity into infrastructure services and APIs" [10].

2.2.3 Modular by design to support phased adoption

Cloud readiness varies significantly between users. Some organizations may want immediate end-to-end processing, others only need storage. To address this, KForge is designed as a modular and composable platform – enabling customers to start small and scale functionally over time.

For instance:

- Begin with KForge Storage: a governed, cloud-native ingestion layer
- Later integrate KForge Processing: event-triggered workflows for user-defined algorithms
- Finally, adopt KForge Analytics: APIs and tools to deliver products and insights

Each module operates independently but shares a consistent API surface, metadata model, and governance structure, ensuring forward compatibility and composability.

This modularity lowers the barrier to adoption and creates business value by:

- Allowing gradual onboarding, aligned with budget and staffing realities.
- Enabling service expansion without architectural rework.
- Avoiding vendor lock-in, since each layer users open formats and APIs like STAC, COG and OGC standards.

This approach supports real-world conditions where users may want to prototype a capability, evaluate it in isolation, and only then grow it into a broader operational workflow – all without redesign.

3 Core capabilities

The operational value of KForge rests on a set of foundational capabilities that address the critical concerns of organizations working with Earth Observation (EO) data in the cloud. These capabilities are implemented natively in the platform to reduce friction, improve compliance, and enable mission success – even for users without deep technical cloud knowledge. While further technical details is provided in later sections, the following introduces each capability with a focus on its practical role and business relevance.

3.1 Security

Security in KForge is not a configurable add-on – it's an intrinsic part of every data transaction, workflow, and access pattern. Rather than relying on manual configuration or bolt-on controls, the platform applies security through native cloud mechanisms such as identity-aware access policies, encrypted storage, and managed key services.

This ensures:

- Consistent protection of data at rest and in transit.
- Isolated execution environment for processing tasks.
- No operational overhead for users to configure or maintain their won security stack.

This approach reduces risk exposure and aligns with widely adopted cloud practices like Infrastructure as Code, which supports automated enforcement and auditable controls. KForge's Zero Trust architecture systematically verifies and secures every transaction, ensuring that access is strictly need-to-know, which significantly mitigates potential security breaches.

3.2 Data Sovereignty and Residency

Organizations operating under jurisdictional or regulatory constraints – such as national governments, defense contractors, or space agencies – must retain control over where their data lives and who can access it. KForge accommodates these needs by enforcing geographic data residency, policy-based access, and customer-controlled key management.

- Storage can be region locked.
- Processing occurs within defined jurisdictions.
- Access logs and metadata are fully visible to the data owner.

This gives users confidence that their data stays where it's legally required and is never moved or shared without visibility – a feature specifically emphasized in best practices for EO data platforms. This robust framework ensures compliance with global data protection regulations, reinforcing trust and legal conformity across all operations.

3.3 Cost Efficiency

In the cloud, uncontrolled growth of compute and storage costs is a well-known risk. KForge mitigates this by making cost-awareness a core part of its execution model. The platform only activates resources in response to events, and it uses lifecycle-aware storage policies to shift data between access tiers based on usage patterns.

Key benefits include:

- No idle workloads, thanks to event-based scheduling
- Data tiering to reduce storage bills over time
- Tagging and visibility into per-object usage

This ensures that organizations don't need to spent time tuning infrastructure to be cost-efficient - it's embedded by design. This dynamic resource allocation not only optimizes cost but also enhances the operational agility of the platform.

3.4 Governance and Compliance

Whether serving a regulated agency or commercial customer, organizations need to demonstrate who did what, when, and why. KForge offers that visibility and control through standardized metadata models, role-based access, and immutable audit trails.

This supports:

- Internal data governance, with minimal configuration
- External compliance audits, backed by traceable records
- Federated data environments, where access can be scoped across teams and partners

Rather than reinventing governance, KForge leverages proven cloud-native mechanisms and enforces them automatically. This is aligned with emerging best practices for cloud-native EO platforms. KForge seamlessly embeds robust governance tools directly into the platform, enabling powerful control and comprehensive oversight with minimal effort.

3.5 Timeliness and Near-Real-Time Processing

Timeliness is essential for mission outcomes. KForge accelerates time-to-insight through event-driven orchestration, ground-to-cloud integration, and automated AI pipelines. Key components include:

- Event-triggered workflows begin execution immediately upon satellite downlink arrival.
- Elastic scaling and task prioritization ensure time-critical data is processed immediately, while lower-priority jobs are deferred.
- Integrated support for data quality checks and validation ensures reliable outputs, with build-in observability for detecting anomalies in ground segment operations.
- Provides built-in advisory features for satellite operators, delivering tailored insights to enhance mission performance within KForge.
- Where applicable, mission-specific AI-powered feature extraction and anomaly detection accelerate value extraction in emergency situations (e.g., disaster response, tactical surveillance).
- Retransmission support allows scheduling a new satellite pass if data loss is detected mid-pipeline.

For users, this means that satellite imagery can move from acquisition to actionable insight in minutes – not hours– supporting high-value missions from maritime tracking to crisis response. KForge's swift processing ensures that essential data is ready when needed, making real-time decisions possible and effective.

4 KForge Platform: Simplify Satellite Data Operations

KForge integrates three primary layers – Storage, Processing, and Analytics -- into a seamless platform designed for comprehensive satellite data management. Each layer is engineered to support the lifecycle of satellite data management. Each layer is engineered to support the lifecycle of satellite data, form initial acquisition to advanced analytical insights, enabling efficient value extraction and streamlined workflows.

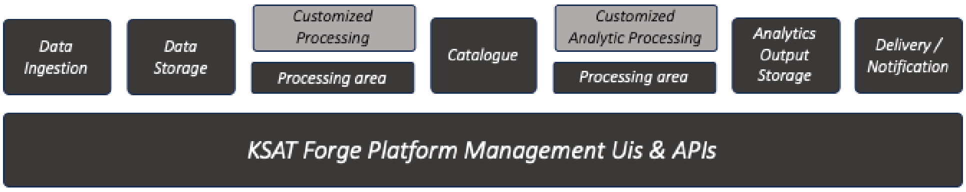


Figure 2: KForge level architecture with platform-managed services and customer-specific compute.

- Storage Layer: Serves as the entry point for incoming data, ensuring secure and rapid ingestion and storage. It supports all stages of satellite data processing and employs cloud-native formats like COG and STAC to facilitates immediate querying and efficient data retrieval without full data transfer.
- Processing Layer: Transforms raw data into actionable information through scalable, data and event-driven pipelines. This layer allows customers to apply custom processing logic, significantly reducing the latency typically associated with satellite data handling by enabling real-time processing capabilities.
- Analytics Layer: Focuses on extracting actionable insights directly at the source, minimizing data movement and latency. This integration facilitates immediate analytical processing, supporting dynamic decision-making and operational responsiveness.

KForge introduces a streamlined approach to satellite data management, balancing core platform responsibilities with user-customizable features to enhance security, efficiency, and user autonomy. The architecture ensures that essential functions like data ingestion and storage are robustly managed by KForge, while granting users the flexibility to adapt access controls and analytics to their specific needs. This shared responsibility model simplifies operations, reduces complexity, and enables cost-effective scalability. By automating resource adjustments and integration advanced data handling tools, KForge supports dynamic and economical satellite data operations, facilitating enhanced discovery and seamless integration with diverse data sources.

5 User Experience and Interaction

KForge will provide full transparency, with control and visibility into data, cost, workflows, processing resources, governance, and so on. The primary interface to interact with KForge will be through is management and resource APIs, to streamline M2M interactions. However, to support a wider variety of users, KForge will also provide a full-fledge Human User Interface (HMI), as shown in Figure 3.

The HMI will be built on top of the API, ensuring a good and effective user journey both when setting up and configuring the system, and when running the system for production purposes.

5.1 *User-Centric Design*

KForge is built from the user outwards – not from architecture diagrams inwards. Every feature begins with user journeys, prioritizing real workflows, needs and outcomes over technical assumptions. Development runs iterative cycles, where we prototype rapidly, validate early, and incorporate direct user feedback. This ensures the platform evolves in lockstep with the people who rely on it.

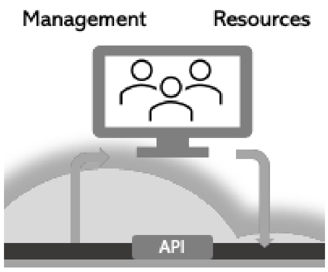


Figure 3. Access through APIs or HMI

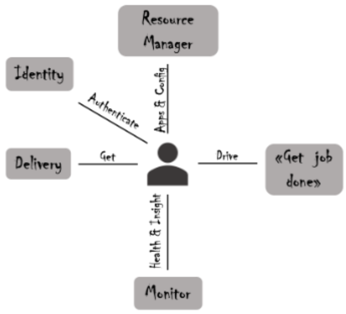


Figure 4. Illustration of simple interaction patterns with KForge

5.2 *Typical patterns*

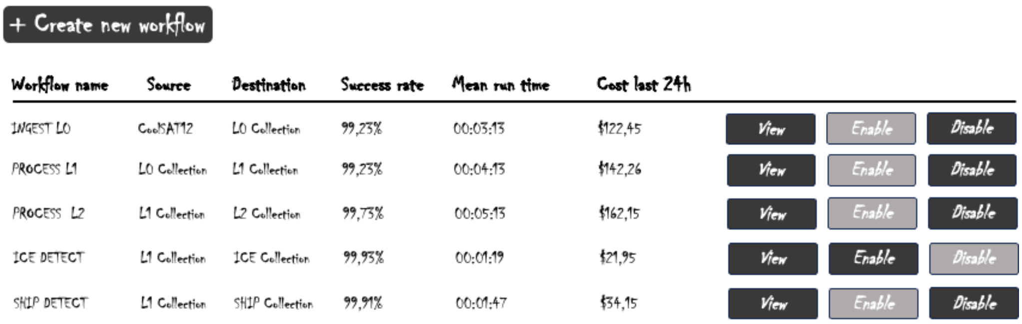
The typical interactions patterns are as follows (and visualized in Figure 4):

- Authenticate: The user first engages with the platform ecosystem by *registering* and *authenticating* a profile. This opens the entryway into the full capabilities of the Data Processing Platform, based on correct privileges for the requested action(s)
- Apps and Config: The user can *inject* processing applications and *configure* directed graphs of work to be performed on the incoming data. The workflows are driven by the incoming data.
- Drive: These graphs of work, or workflows, are then used to *drive* value production, manually or automatically based on available data
- Monitor: The user can at any time and easily gain *insights* into processing progress and component *health* statuses, as well as get a firm grip on *cost* of getting the work done
- Get: As the data processing is done, there exist several API-based options on getting the value *output* from the system; all the way from delivered as a stream during processing, getting notified of done status to something downstream clients can pull

Note that these patterns are independent of used by the HMI or over the APIs.

Examples of interactions are discussed below.

5.2.1 *Workflow management*



+ Create new workflow

Workflow name	Source	Destination	Success rate	Mean run time	Cost last 24h			
INGEST L0	CoolSAT12	L0 Collection	99,23%	00:03:13	$122,45	View	Enable	Disable
PROCESS L1	L0 Collection	L1 Collection	99,23%	00:04:13	$142,26	View	Enable	Disable
PROCESS L2	L1 Collection	L2 Collection	99,73%	00:05:13	$162,15	View	Enable	Disable
ICE DETECT	L1 Collection	ICE Collection	99,93%	00:01:19	$21,95	View	Enable	Disable
SHIP DETECT	L1 Collection	SHIP Collection	99,91%	00:01:47	$34,15	View	Enable	Disable

Figure 5 Example of workflow overview

Figure 5 shows an example of workflow overview, including workflow identifier, the data source and destination, success rate, as well as mean running time and associated cost. In addition, each workflow can be viewed in more detail and enabled/disabled according to current production needs.

Figure 6 shows an example of workflow manipulation, i.e. creation, editing and deletion. Workflows can be seen as a type of directed graph, involving one or more steps, sequential or in parallel. The individual steps can either be based on existing functions or new functions being uploaded to KForge.

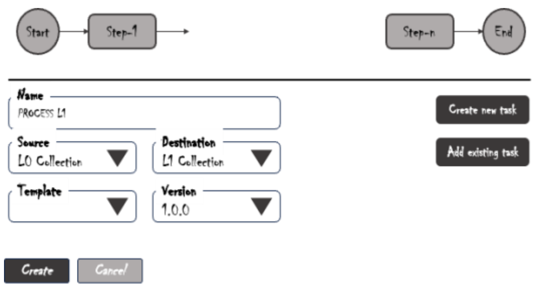


Figure 6. Example of workflow creation and editing

Once a workflow is created, it will be visible as shown in Figure 5.

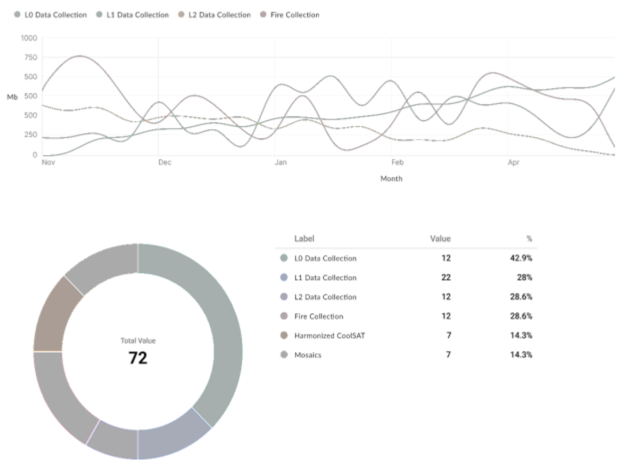


Figure 7. Example of storage cost overview

5.2.2 Cost management

Cost management includes both mechanisms to monitor cost and mechanisms to limit cost. Examples of the first are shown in Figure 5 and Figure 7, and available over the corresponding API for M2M. The latter is foreseen to include the following mechanisms:

- Configurable overall thresholds, based on total consumption for a given time period
- Configurable workflow thresholds, based on a given workflow's consumption for a given time period
- Workflow priority, assigning a predefined priority to a given workflow. Edge examples may be:
    - Archive only – time not important so keep cost predictable
    - High importance – run as fast as possible regardless of cost

With more "normal" production priorities in between

It should be noted that thresholds can be cascaded, meaning that some can merely acts as warnings while others can impose actions.

6 Discussion

Embarking on the KForge journey, our ambition is to provide a secure, easy to use and efficient platform for processing and analytics of satellite data. Although KForge is not finished yet, engaging with potential users has demonstrated that such a platform may provide value to the user community. Our intention is to continue these discussions throughout development to ensure that the final product meets the demands of actual users.

Public cloud is a powerful enabler to unlock the full value of satellite data, though it requires deep investment in cloud-native design, architecture, governance and security. We argue that this should reside outside of the focus of the mission operators and value adders, as we believe the customization benefits of DIY is largely offset by duplication of effort, fragmentated organization of workflows, higher security risks and eventually increased operational cost over time. KForge eliminates that burden, allowing customers to immediately focus on their mission outcomes.

The purpose of KForge is to simplify operations for satellite operators, and offload infrastructure responsibilities and automates ground segment functions. It will enable fast, secure transition from data acquisition to value delivery. It will further enable rapid testing and scaling without large DevOps team and allows operators to focus on their core competences, mission designs and customer engagement. For value providers, KForge will standardize data indexing, enable timely access to processed data, and seamless collaboration with and between data providers. The pre-designed environment will allow the analytics companies to experiment, scale and deliver products in cost-aware and compliant architecture, without months of setup.

To ensure operational resilience and adaptability, KForge has a cloud exist strategy that emphasizes the importance of evaluating and mitigation vendor lock-in risks, including cost implications. These measures facilitate cost-effective transition between different technologies or vendors and are in line with our commitment to flexibility and sustainability in cloud operations.

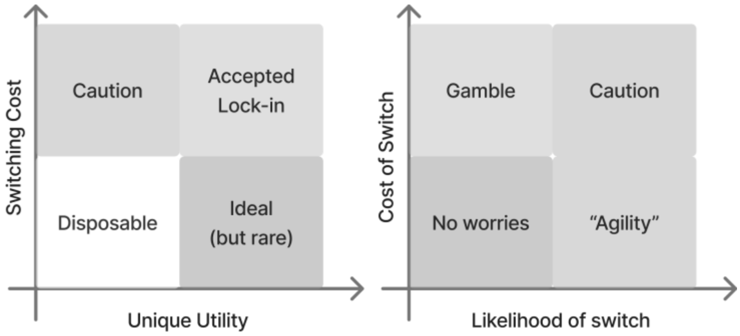


Figure 8: Matrix evaluating utility and switch likelihood.

Drawing on Gregor Hophe's framework [13] for evaluating technological choices, we assessed the unique utility and switching cost of managed services, alongside the likelihood of a switch, to develop a tailored strategies that minimize cost and maximize operational efficiency within KForge.

7 Conclusions

This paper presents the architectural vision and operational model for KForge as it is being realized. While much of what is described is implemented and operational, some features and capabilities are in active development and may evolve beyond what is captured here. Our intent has been to communicate the direction, structure, and value of the platform – even where implementation details may shift over time.

KForge is built in close collaboration with its users, and it is our belief that it exemplifies a shift in how satellite data is transformed into value – away from infrastructure-heavy, bespoke environment towards governed, cloud-native platforms that scale by design. It is built with a focus on operational realities, enabling mission operators, analytics providers, and commercial users to move rapidly from raw data to insight without sacrificing governance, performance, or security.

The platform is meant to abstract complexity and standardize the difficult parts – storage, ingestion, orchestration, security, and compliance – while empowering users to focus on what differentiates them: their algorithms, their mission, and their outcomes. For an industry where timing, agility, and data trust are critical, KForge offers a foundation that is as secure as it is scalable, and as pragmatic as it is powerful.

References

[1] John Coykendall, Kate Hardin, Alan Brady and Aijaz Hussain, Deloitte Insights | Riding the exponential growth in space. March 22, 2023, https://www2.deloitte.com/us/en/insights/industry/aerospace-defense/future-of-space-economy.html, (accessed 31.3.2025)

[2] Alan Burkitt-Gray 23, 'Fourfold increase' in satellites over the next 10 years to 17,000. December 10, 2021, https://www.capacitymedia.com/article/29tguvmaqy9h43clx4mio/fourfold-increase-in-satellites-over-the-next-10-years-to-17-000, (accessed 31.03.2025).

[3] Cloud Native Computing Foundation, Cloud Native Definition. https://github.com/cncf/toc/blob/main/DEFINITION.md (accessed 31.03.2025).

[4] Amazon Web Services, Cloud-Native Modernization Whitepaper. https://pages.awscloud.com/rs/112-TZM-766/images/cloud-modernization-aws.pdf (accessed 1.4.2025).

[5] *Cloud Optimized GeoTIFF*, https://www.cogeo.org/ (accessed 31.03.2025).

[6] *STAC Specification*, https://stacspec.org/ (accessed 31.03.2025).

[7] *GeoParquet - Cloud-Native Geospatial Formats Guide*, https://guide.cloudnativegeo.org/geoparquet/ (accessed 31.03.2025).

[8] ISPRS International Journal of Geo-Information, *Serverless Geospatial Data Processing Workflow System Design*, https://www.mdpi.com/2220-9964/11/1/20 (accessed 31.03.2025).

[9] HashiCorp, *Infrastructure as Code*. https://developer.hashicorp.com/terraform/ (accessed 31.03.2025).

[10] Open Geospatial Consortium, *Towards a Cloud-Native Geospatial Standards Baseline*. https://www.ogc.org/blog-article/towards-a-cloud-native-geospatial-standards-baseline/ (accessed 31.03.2025).

[11] National Institute of Standards and Technology. NIST Special Publication 800-270: Zero Trust Architecture. https://csrc.nist.gov/pubs/sp/800/207/final (accessed 1.4.2025)

[12] National Institute of Standards and Technology. NIST Special Publication 800-270A: A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments. https://csrc.nist.gov/pubs/sp/800/207/a/final (accessed 1.4.2025)

[13] Don’t get locked up into avoiding lock-in – Gregor Hohpe, https://martinfowler.com/articles/oss-lockin.html (accessed 4.4.2025)