AI-eXpress: A new satellite-as-a-service concept reshaping the Earth Observation value chain and beyond

Vito Fortunato (1), Cristoforo Abbattista (1), Leonardo Amoruso (1), Gianluca Furano (2), Stefano Antonetti (3), Lorenzo Feruglio (4)

(1) Planetek Italia S.r.l., Via Massaua 12, I-70132 Bari, Italy, +39 0809644200, fortunato@planetek.it, *abbattista@planetek.it*, *amoruso@planetek.it*
(2) ESA, ESTEC Keplerlaan 1, 2201 AZ Noordwijk, Holland, +31 71 565 6631, Gianluca.Furano@esa.int
(3) D-Orbit spa, Viale Risorgimento, 57, 22073 Fino Mornasco CO, Italy, +39 02 3792 0900, *stefano.antonetti@dorbit.space*
(4) AIKO srl, Corso Castelfidardo, 30/A, 10129 Torino TO, Italy, lorenzo.feruglio@aikospace.com

Abstract

AI-eXpress (AIX) is an AI-enabled flight acquisition and processing framework, that embarks on satellites and then brings into the market a new concept of satellite a service. AIX makes available in-orbit resources/services on-demand: services shall include EO data acquisition, processing (i.e., actionable information extraction), downlink, and distribution. Ready-made applications (e.g., fire detection and warning service) will be available on a dedicated app store, and services can be also combined to build custom applications. Valuable information can be then transferred back to Earth as notifications and alerts, or directly exploited on-board in autonomous decisions workflows. The data that is demonstrated to be not relevant to user applications can be processed and then deleted, thus freeing space in memory and using less bandwidth. Furthermore, emerging NewSpace companies may now test their innovative AI algorithms and their proof-of-concept directly in space and prove their value to the market. Traditional space institutions and research entities may test a new approach changing from "makers" to "enablers".
AIX is developed, by Planetek Italia, D-Orbit and AIKO within the ESA Incubed program and the services will be available from Q4\2024.

1 THE CONTEXT AND THE NEED FOR A PARADIGM SHIFT IN EO

The Space missions paradigm is constantly evolving. Nevertheless, especially in the EO market, the transition from a traditional service model into a commercial one is showing a few bottlenecks and barriers in specific applications, preventing new market opportunities from flourishing and limiting the effectiveness of the services delivered to the ground ([1]-[2]-[3]).
Some of the barriers are currently being addressed (e.g., launch availability & satellite deployment flexibility, versatile ground infrastructures, and software, suitable regulations, etc.). However, existing or incoming satellite operators and end-users of EO data are still experiencing severe inefficiencies, affecting traditional mission operational models:

1. missed information due to limited downloadable data from satellites (small satellites usually offer limited power, bandwidth, and controllability);
2. poor quality or irrelevant downlinked data,

3. missed observation opportunities due to limited onboard autonomy (data analysis is currently primarily executed on the ground);
4. delays in decision-making due to ground operator intervention;
5. late systems failures detection;
6. high complexity and flexibility of algorithms and science SW development and deployment.

Almost all of these problems may be tackled today thanks to the increasing availability of onboard computing power and advanced techniques and algorithmic solutions, such as artificial intelligence, enabling autonomous operations. In particular, the advanced functionalities based on detection, extraction, and exploitation of data information content will be the core value points in this paradigm shift, moving the focus beyond “raw” data. These new functionalities will enable the management of huge raw data volumes thanks to structured and automated information refinement processes. They enable systems to react in near real-time to single pieces of information and to touch off new tasking for improved EO acquisitions specifically targeting the detected needs. We used to define “SpaceStream” such a workflow, processing data where it is more convenient, and we identified AIX as a new way to interface EO users with both Space and Ground segments, calling these Spacedge services.

2 A USER-CENTRIC APPROACH

An AIX/Spacedge distinctive feature, as depicted in Figure 1, is the set of deeply configurable services it provides, spanning from pay-per-use to full missions-as-a-service. These services allow the evaluation of new approaches to space missions and the validation of novel concepts in fully representative or real operational environments.

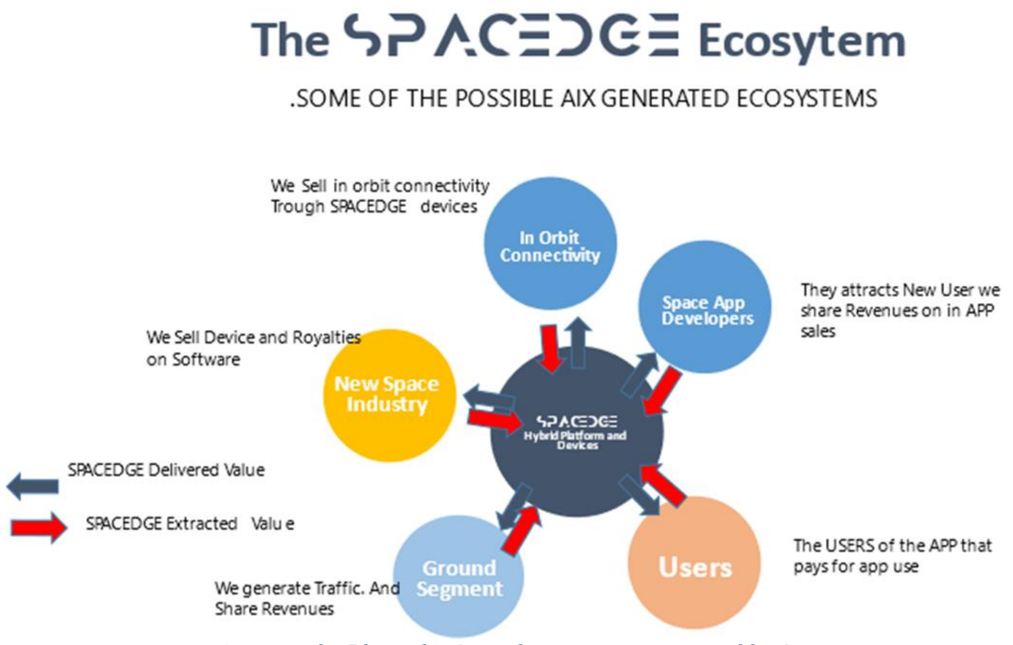

The SPACEDGE Ecosytem
.SOME OF THE POSSIBLE AIX GENERATED ECOSYSTEMS
We Sell in orbit connectivity Trough SPACEDGE devices
In Orbit Connectivity
Space App Developers
They attracts New User we share Revenues on in APP sales
We Sell Device and Royalties on Software
New Space Industry
SPACEDGE Hybrid Platform and Devices
SPACEDGE Delivered Value
SPACEDGE Extracted Value
Ground Segment
Users
The USERS of the APP that pays for app use
We generate Traffic. And Share Revenues

Figure 1. The Planetek’s Spacedge ecosystem, powered by AIX
Credits: https://www.aiexpress.eu [4]

The offered flexibility and configurability enable the early execution of in-orbit tests and the applications fine-tuning through continuous, iterative testing, grafting innovation at different levels:

a. By implementing and validating a novel, ever-evolving technological framework and corresponding testbed that provides advanced tools and basic building blocks for applications (AI\inference-based too) to be built upon;

    i. thanks to this framework and the standard interfaces it provides, it is going to ease developers in onboard SW implementation and re-use on different S/C (OBDP) platforms;

  b. By supporting AI-experienced users in testing their algorithms and techniques in real EO cases, with resources and payloads available in orbit;
  c. By setting up an infrastructure of in-orbit and ground facilities, available on-demand through transactions that exploit the same core technologies blockchain is based on (DLT);

Such an ecosystem is taking the current IOD/ IOV services one step forward, anticipating and actively contributing to the definition of the forthcoming evolution of the "NewSpace Economy". It makes available the on-the-fly configurability of services and allows the monitoring of areas and topics of interest, detection of changes, and alarm raising, in response to specific data processing events, supported by algorithms and tools included in the framework (including AI\inference based ones).

3 METHODOLOGY

Powered by AIX ([4]), Spacedge answers this user's needs by making satellite assets available as a service through DLT and allowing the execution of EO applications augmented with Trustworthy AI. To this end, an experimental testbed is defined, targeting the evaluation of AI and blockchain-related state-of-art technologies and algorithms in the space environment, explicitly targeting items like reactivity, responsiveness, and latency with dedicated processing units (from RISC-V processors to VPU, TPUs, …).
The testbed's main requirements are:

1. an orbiting platform, such as D-Orbit's ION Carrier [5], providing hosted payloads and ready-to-deploy CubeSats;
2. an SW framework infrastructure (implementing services and an abstraction layer towards sensors and onboard resources);
3. a catalog of onboard resources (HW/SW components and also complete CubeSats in standard configurations);
4. a catalog of processing functions and applications algorithms based on AI;
5. onboard service configuration;
6. CubeSats deployment on-demand;
7. support service to custom design needs;
8. Ground Segment support services for operations.

Once these system functional requirements have been derived from the high-level user requirements described in Section 2 above, the overall system composition is defined according to components described in the next Section 3.1.

3.1 Overall system components

To provide services to end-users, the whole SpacEdge-enabled system has been modeled as a composition of three (3) segments including:

1. a ***Space segment***, constituted by two main functional/physical (sub-)components on the ION carrier, the Platform(==ION), and the Hosted payloads (==the SW framework);
    a. The ***hosted payloads*** [5] are connected to the Carrier with standard interfaces already provided by ION's On-Board Data Handling sub-system (e.g., CAN, I2C, USART, USB), so there is no need to adopt ad-hoc solutions for the framework: full control from a custom Mission Control Center is already guaranteed;
    b. The ***ION Platform*** is designed [5] with a modular architecture, suitable for an easy expansion of its functionalities. This characteristic is exploited to integrate the HPC into the Carrier as a "hosted payload" and to serve the capabilities offered by the SW framework to all the other 3rd party "hosted payload" (clients') embarked (and

to the Carrier itself whenever needed). The main functional modules exposed by ION are:

i. OBDH
    1. Routing and storage of data
    2. Platform/Payloads interface
    3. TM/TC

ii. AOCS
    1. Attitude control
    2. Environment sensing (Sun, Earth, Position, Time)
    3. maneuvers allowing the pointing of the payloads

c. The ***SW framework*** per sè, with its Services module, manages the exchange of data between the HPC and the AIx-enabled client's Payloads and between Space and Ground Station segments. A basic list of functionalities belonging to this Carrier's subcomponent are:
    i. Basic Payload data management
    ii. access to platform auxiliary data
    iii. time management
    iv. execute ML/DL algorithms (inference)
    v. library of deterministic functions
    vi. spatial filters, interpolators, spatial registration, features, and novelties detection
    vii. data-adaptive compression techniques
    viii. the smart selection of data sub-sets
    ix. TM packet formation functions
    x. access to low-level processing resources through APIs provided by OS and/or BSP

2. a ***Ground segment as a service*** portal, devoted to operations;
3. A u***ser segment***, constituted by a ***user frontend*** and a ***service configuration backend***, shall be in charge of exposing/enabling customers to exploit a "configurable mission on-demand" in terms of:
    a. starting from the user's application;
    b. selecting/ configuring sensors;
    c. tasking acquisitions;
    d. applying pre-processing;
    e. selecting algorithms (eventually, configure AI\inference from pre-trained networks);
    f. defining actionable information
    (to use or downlink).

The basic usage scenario is functionally sketched in the following Figure 2 and explained in subsection 3.1.1

3.1.1 Operational services

The functional workflow diagram of these three main segments highlights the core flow baselining three different concepts of operations (ConOps) to which the services (and therefore the underlying products) must provide enabling modules and interfaces characterizing AIX.

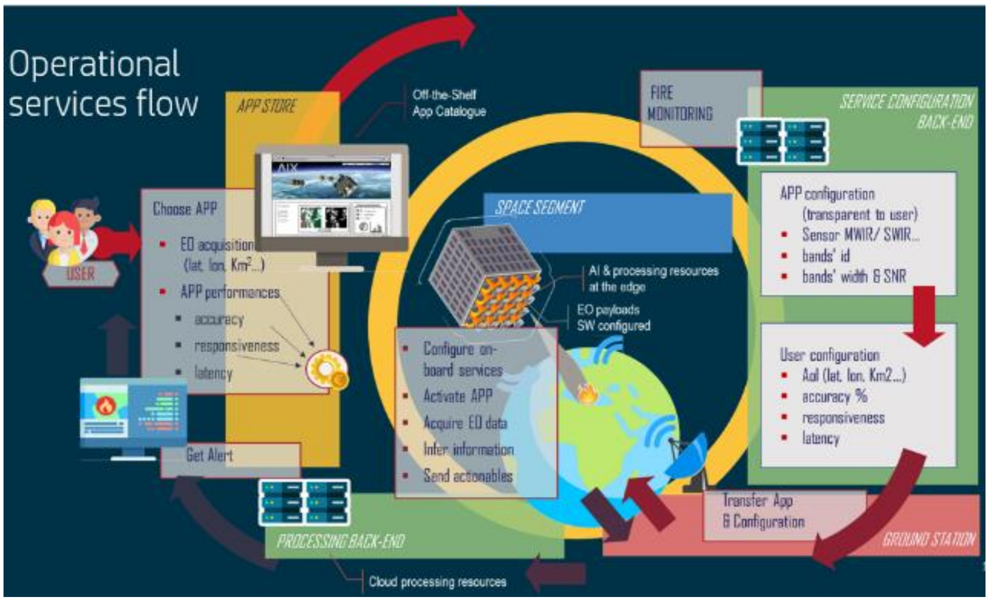


Figure 2. AIX operational services flow

As a baseline, the web portal (i.e., the app store) represents the ***access point for users*** to interact with the system. They can configure/activate available services, including access to the managed EO payloads and processing resources. Furthermore, customers can select, configure, customize, and activate services in timeslots, through the ***Service Configuration Backend*** by uploading their application workflow(s), transparently via the Ground Station system, to the orbiting platform (the ION Carrier), which executes it onboard, in the scheduled time interval, and then sends back the results to the user as planned (again via Ground Segment and web portal). After execution, the application onboard is disabled, the involved resources are freed, made available to the next scheduled workflow and the system continues its cycle. The software application provided by the customer may also contain code that uses ML/DL techniques. The system is capable of operating inference by exploiting a network pre-trained on the Ground and modeled as a configuration to be loaded onboard. The inference will leverage the available high-performance computing cluster directly in Space, thus transforming locally acquired data into actionable information.

3.2 SW framework design

The proposed SW-framework organization, depicted in Figure 3, is composed of different layers. Each layer can use functionalities of the lower ones through a level of abstraction to improve reuse and safety.

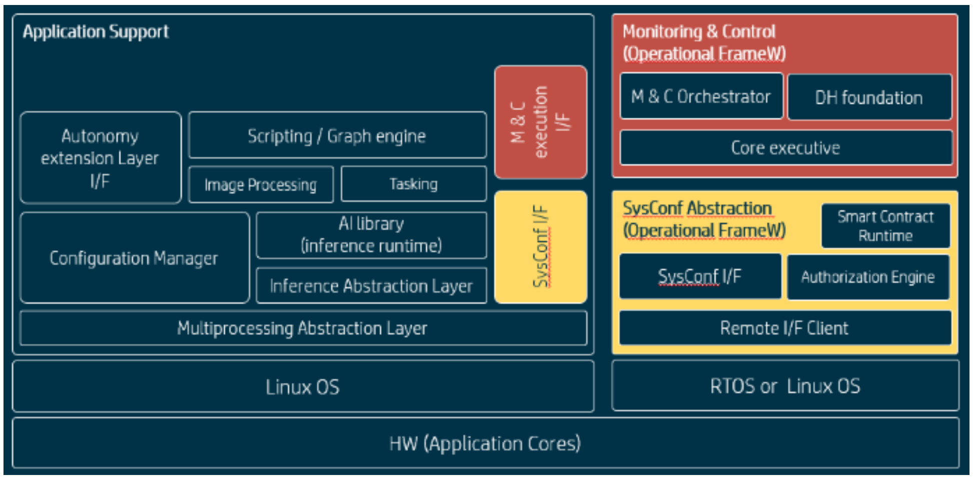


Figure 3. AIX framework: the reference high-level architectural schema

It is composed of three (plus one) different layers (four if the P1 is also considered as a HW layer):

1. System Controller abstraction layer
    a. In addition to the HW layer and its abstractions.
2. Monitoring & Control execution layer
3. Application Support layer

Each layer implements its functionalities including Image Processing functions, AI-Supporting functions, and a set of ready-made AI-based applications.
Hereafter we describe core characteristics, features, and a few subcomponents, for those layers.

3.2.1 System controller abstraction layer

A layer of abstraction is necessary to decouple the System Controller from the applications/services running on the application cores. This layer allows the management of communication between the two sides, in terms of RPC interfacing, authorization for the use of available RPCs, and smart contracting that allows the negotiation and control of these authorizations. Applications and services executing on the platform application cores can use a smart contract runtime to autonomously negotiate the use of application cores resources and be granted the appropriate authorizations for this by a security-ring-based authorization engine.
Given the necessary authorizations, applications can use traded and agreed resources to perform tasks on sensed data coming from other connected payloads, receive commands, and downlink actionable information to the Ground. The communication links through the S/C platform to other payload sensors and the ground must be established using a remote interface client to the services through the System Controller abstraction interface.

3.2.2 Monitoring & Control layer

The Monitoring and Control services layer provides commanding and observability capabilities for the different application support contexts of the AIX-hosted payloads. A core executive subset provides an infrastructure of essential time, event reporting, publish and subscribe messaging, and data tables updating and verification services. These can be used directly by applications, but are also used by the Data Handling (DH) foundation functions that provide Housekeeping, File, Memory, Network, FDIR, and TC/TM management services. Finally, an M & C orchestrator provides abstraction layers for the System Controller and for user applications/ services that need to interface with this execution layer.

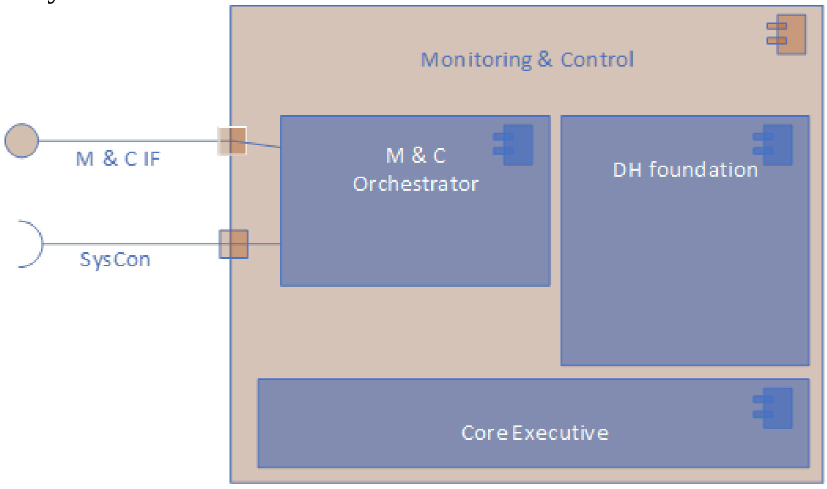


Figure 4. Monitoring & Control layer

A companion pilot/building-block application for demonstrating this specific capability is the orbital_OLIVER product from AIKO ([6]), whose goal is to detect and prevent potential contingencies before they affect satellite operability. To do that, it continuously monitors telemetry and housekeeping data streams from the platform.

3.2.3 App support services

The Applications' support services provide a set of functions to support applications at runtime. Applications that aim to provide deterministic and/or AI-based image processing or generic data processing services, can use services offered by this layer.
It is composed of a set of APIs, aimed at contextually 1) de-coupling the application code from the 3rd-party, and HW optimized, inference runtime, 2) executing processing pipelines represented by user-defined graphs. by which the user can (transparently) create a graph of operations and export it as a data file represented in a specific format and then deploy it onboard so that it can be executed by the Graph Engine, 3) enabling the application to access acceleration devices available onboard (e.g., GPUs, OpenCL-enabled devices, etc.) for off-loading parts of the algorithms to exploit parallelism.
All these features rely on the design and have been deployed via Planetek's SpaceKit (or PkSpaceKit) framework, through which, two aspects of the AIX\Spacedge (SW) ecosystem have been enabled: a) the generic data processing services and the b) AI-augmentation services.

3.2.4 Data processing services

PkSpacekit is a ***multi-platform***, ***C++1x*** (C++17) development environment (an ***SDK***) for creating ***software applications*** spanning from ***console-based*** data processors to ***interactive graphics tools***, mainly devoted to ***image processing and visualization.*** Its main goal is to provide *a cross-platform development environment and runtime for creating general-purpose applications* that can run on desktops, servers, and embedded systems with a single codebase. It achieves this goal by putting together a *rich set of dedicated classes and the functionalities provided by the C++17 runtime library*.
As a framework, it is composed of a ***collection of dynamic libraries*** (DLL on Microsoft Windows, SO on Linux, and DYLIB on Apple macOS), and their respective include files. The ***foundation libraries are*** the following ones:

1. *core*, which is responsible for abstracting the underlying operating system services for the upper layers of any application, so that access to the filesystem, some inter-process communication facilities, and other low-level services become cross-platform for the user of the framework;
2. *math*, which provides basic mathematics functions and structures, such as vectors, matrices, and simple algorithms;
3. *acc*, which provides image manipulation and processing capabilities, abstraction of acceleration facilities (such as those based on OpenCL), and implementations of thread schedulers for multi-processing architectures.

On top of these three libraries, we created other libraries with further specializations and a rich ecosystem of plugins (i.e., executable modules loaded at runtime and embedded into the address space of the process) that augment the capabilities of the overall framework in terms of:

1. ***Support for image formats*** (such as jpeg, tiff, png, hdf5, etc.) both in reading and in writing (not all formats support writing at this time: only jpeg, hdf5, and tiff);
2. ***Support for more algorithms*** both of general usage and very specialized, encapsulated into entities named "***engines***", which are special-purpose plugins that mostly create instances of the "fast" (see a dedicated document on this topic) concept, provided by *acc* ;
3. ***Support for more acceleration devices and APIs*** (for example, the OpenCL engine provides an implementation of the Accelerator interface found in *acc*, thus allowing access to a GPU or a CPU-based OpenCL platform).

On top of all the ***libraries and plugins***, there's also a set of higher-layer libraries and applications that provide a basic foundation for creating graphics features based on OpenGL and Qt. The following Figure 5 represents the relations among these elements, highlighting the composability and flexibility of the framework.

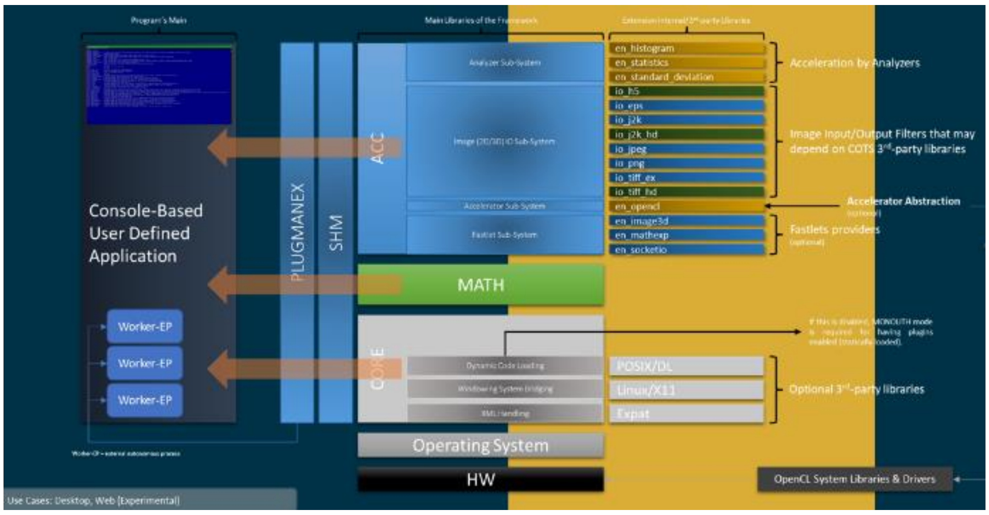


Figure 5. App support service, powered by PkSpacekit

3.2.5 AI-augmentation services

Another distinctive feature of such plugin-based architecture, is the support, via a dedicated inference engine, of user-defined NN (and CNN) models via some of the most widely used and generic frameworks like Apache TVM [9], VectorBlox (OpenVino) [10], TensorFlow and the Open Neural Network eXchange (ONNX) format [8]. To this end, based on the actual HW availability and developing support it is guaranteed:

1. Framework interoperability -> Models can be trained using the preferred training framework and easily exported to ONNX format.
2. Inference acceleration -> Easy access to hardware accelerators to improve inference speed
3. Quantization support -> Different quantization techniques can be applied to get lightweight models

Next Figure 6 depicts such a mechanism, sketching the basic behavioral model supported, and extensible, via the framework.

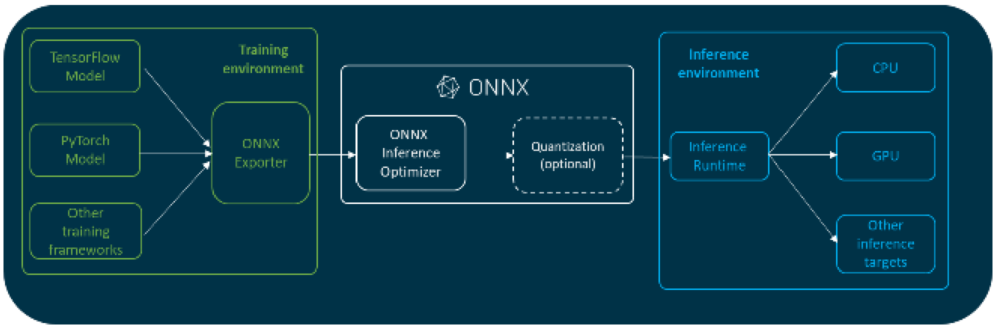


Figure 6. AIX support for neural networks: the inference engine perspective

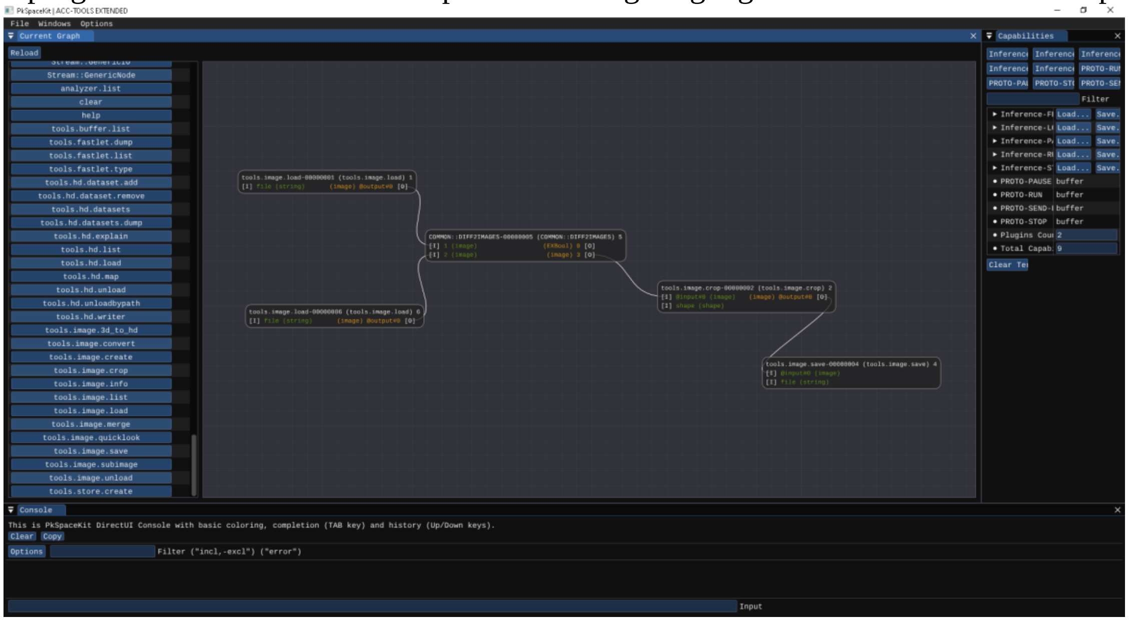
This is meant to provide higher flexibility for the AI community, leveraging on the power of the cited frameworks to expose a minimum (and developing) set of standard ML operators covering tasks like:

- *Classification Tasks*
  Analysis of the image (or tile) and discrimination of its appropriate type among the defined ones. Forest or agricultural field recognition tasks are part of this category for example.
- *Segmentation Tasks*
  Pixel-by-pixel image analysis that categorizes each of them according to given classes. Cloud detection task is part of this category
- *Object Detection Tasks*
  Image analysis to find instances of different object types, locate their position in the image, and classify them accordingly to given classes.

Such an element only depends on the development of the community around those frameworks as well as the adaptation of specific HW vendors for optimizing inference performances.
A pilot/building-block application for demonstrating this specific capability is the cloudy_CHARLES product from AIKO ([6]), whose goal is to filter out cloudy frames, deliver only profitable imagery data products to ground, avoid waste of bandwidth resources, and reduce downlink costs.

3.2.6 Blueprint editor: a development kit for AIX

As part of the User Segment, AIX also provides a feature based on the proprietary concept of the SpaceKit pluggable functions pattern. It allows composing workflows, made of C++11 source code function blocks, in a GUI-based application using graphs. This graphical design may replace the specific programmer skills and the boilerplate code for gluing together all the invokable components.



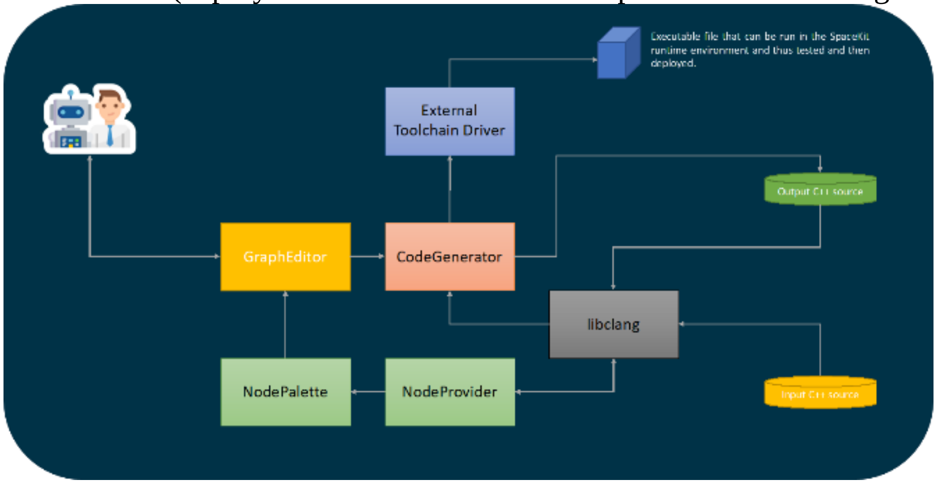
```
tools.image.load-00000001 (tools.image.load) 1
[I] file (string)        (image) @output#0 [O]

COMMON::DIFF2IMAGES-00000005 (COMMON::DIFF2IMAGES) 5
[I] 1 (image)                 (EXBool) 0 [O]
[I] 2 (image)                  (image) 3 [O]

tools.image.load-00000006 (tools.image.load) 6
[I] file (string)        (image) @output#0 [O]

tools.image.crop-00000002 (tools.image.crop) 2
[I] @input#0 (image)    (image) @output#0 [O]
[I] shape (shape)

tools.image.save-00000004 (tools.image.save) 4
[I] @input#0 (image)
[I] file (string)
```

```
▶ Inference-F  Load...  Save.
▶ Inference-L  Load...  Save.
▶ Inference-P  Load...  Save.
▶ Inference-R  Load...  Save.
▶ Inference-S  Load...  Save.
● PROTO-PAUSE  buffer
● PROTO-RUN    buffer
● PROTO-SEND-I buffer
● PROTO-STOP   buffer
● Plugins Cou  2
● Total Capab  9
```

```
This is PkSpaceKit DirectUI Console with basic coloring, completion (TAB key) and history (Up/Down keys).
Filter ("incl,-excl") ("error")
```

Figure 7. Prototypal demo implemented with the imgui-node-editor open-source project [11]

In full compliance with the company's mission to simplify the complexity, such a graphical editor appeals not only to that class of users in the EO (and AI) landscape but also to all those who have never interfaced with an onboard application SW for managing and processing payload data (in general).
The baselined architecture (deployment view on this S/S is depicted in the next Figure 8.



Executable file that can be run in the SpaceKit runtime environment and thus tested and then deployed.
External Toolchain Driver
Output C++ source
GraphEditor
CodeGenerator
libclang
NodePalette
NodeProvider
Input C++ source

Figure 8. Code Generator for SpaceKit

To test the expandability and thus reusability of such tools in a microservices-oriented backend ground, the following Figure 9 shows an example of deployment and enrichment of the Blueprint editor in a (C++) development environment based on Jupyter notebook for the creation of executable elements to be deployed in an embedded environment, e.g., RISC-V.

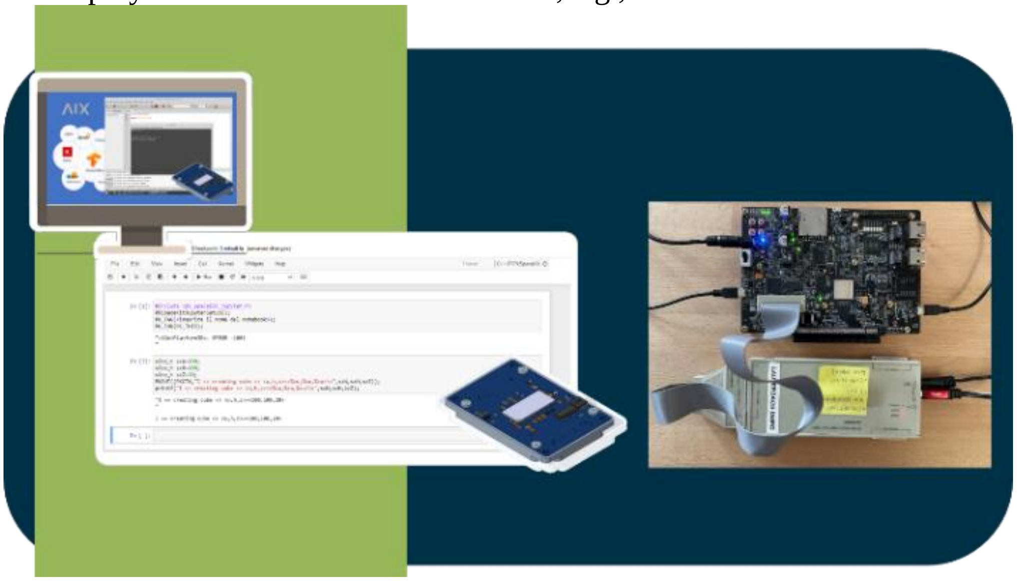


Figure 9. Editor augmentation and deployment: an experiment with Jupyter and a RISC-V platform

4 CONCLUSIONS

Smallsats are expected to be the future of the space industry, enabling novel applications, making space accessible, and delivering the NewSpace paradigm. Perception within the industry has been more mixed, but commercial and government players alike broadly anticipated a net-positive impact from smallsats. The pillars of the new paradigm are:

1. the increased commoditization of data acquired from satellites,
2. the availability of mini sat constellations,
3. the lowering cost of high-performance onboard computing resources,
4. the increased onboard processing capabilities,
5. the increased autonomy (shift from human to machine user) in operations,
6. the introduction of machine-to-machine autonomously communicating orbital network nodes

They bring the potential to solve the inefficiencies affecting traditional missions' operational model: delays in decision-making due to ground operator intervention, missed observation opportunities due to limited onboard autonomy, poor quality or irrelevant downlinked data, and late systems failure detection. These inefficiencies reduce mission effectiveness and increase operational costs (ranging from 4% to 35% of the total mission cost).

Furthermore, autonomous operations and Artificial Intelligence will impact the whole space Value Chain, from the user-service interaction to satellite platforms, from service performances to ground segments, resources management, and mission planning. AI (Trustworthy AI), together with DLT/Blockchain and GPU & hardware-based acceleration device technologies, represent the key points of this new approach and candidates to be real game changers ([4]). As novel technologies, their application in space scenarios still needs evaluation, testing, tuning, and demonstration. The approach currently adopted in the mainstream design process implies either long time frames (not fitting NewSpace) or high-risk rates (and high failure rates).

The services AI-express provides bridge these gaps by also providing access to actionable information, the NewSpace true value, directly in-orbit and as-a-service, decoupling high-level concepts and application design from practical implementation times and risks.

5 REFERENCES

[1] Prospects for the Small Satellite Market, Euroconsult 2018
[2] Nano/Microsatellite Market Forecast, SpaceWorks 2018
[3] Map of main space private actors organized per technology, Seraphim Capital 2019
[4] AI-express, https://www.aiexpress.eu
[5] D-Orbit ION service, https://www.dorbit.space/inorbitnow
[6] AIKO orbital_OLIVER, https://aikospace.com/products/products/orbital-oliver
[7] AIKO cloudy_CHARLES, https://aikospace.com/products/products/cloudy-charles
[8] ONNX, Open Neural Network Exchange, https://onnx.ai/
[9] Apache TVM, https://tvm.apache.org/
[10] VectorBlox™, https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/vectorblox
[11] ImGui node editor, https://github.com/thedmd/imgui-node-editor