# BLENDED

# Using Blockchain and Deep Learning for Space Data Processing

ESA Living Planet Symposium 2022

27 May 2022 – Bonn

Bernard VALENTIN, Leslie GALE

BLENDED Consortium

# Outline of the Presentation

- The motivation behind the BLENDED project
- The technologies used in BLENDED
- The BLENDED Decentralised Training Platform
- The *Urban Expansion Monitoring* study case
- Achievements, benefits and potential continuation

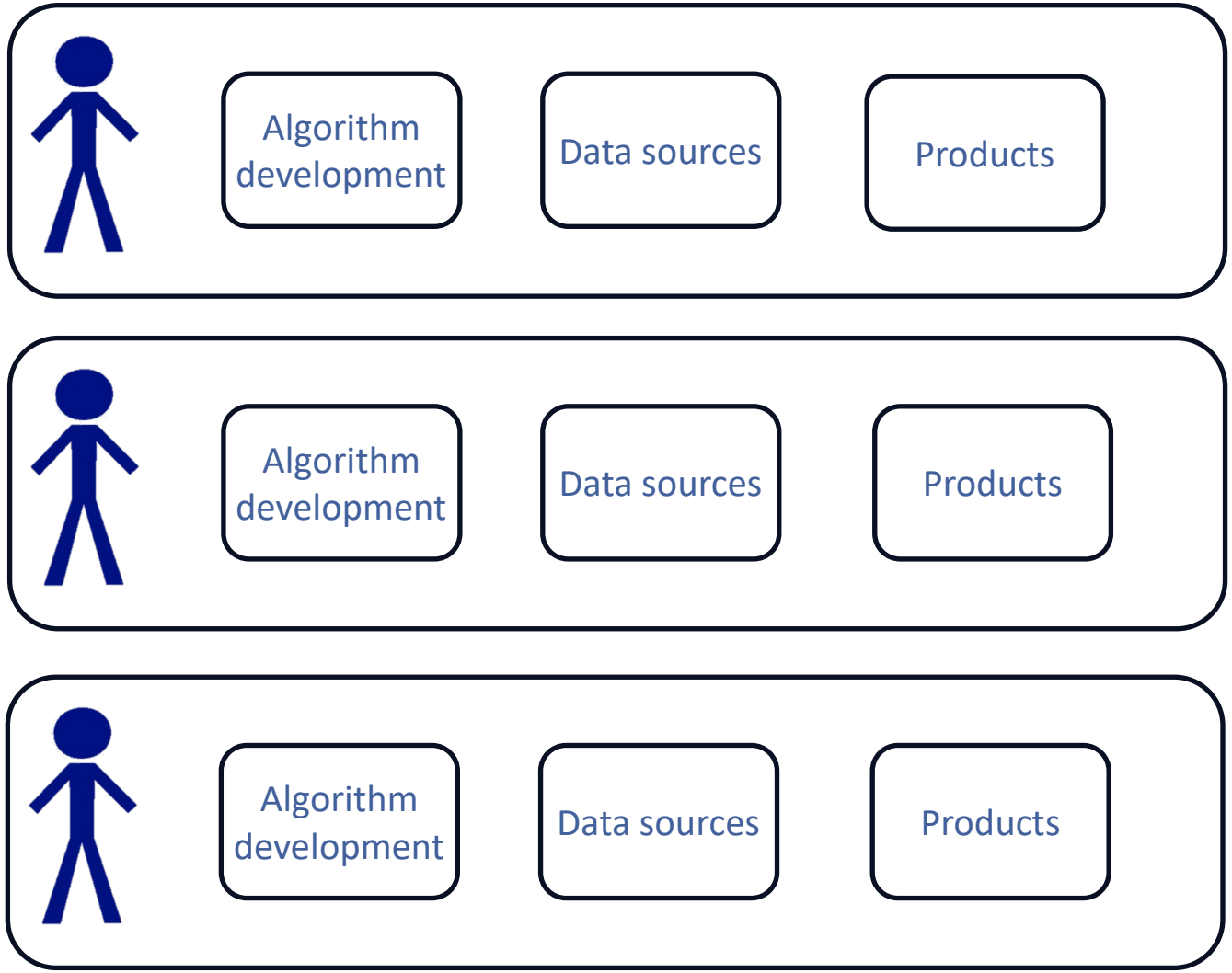
*BLENDED was a project performed under ESA Competence Domain 9: Digital Engineering for Space Missions looking into technical issues that are perceived to be blocking the massive exploitation of space data.*

*Technical Officer: Riccardo Duca, System Engineer, Directorate of Technology, Engineering and Quality.*

*The project started December 2019 and was completed November 2021.*

# The motivation behind the BLENDED project

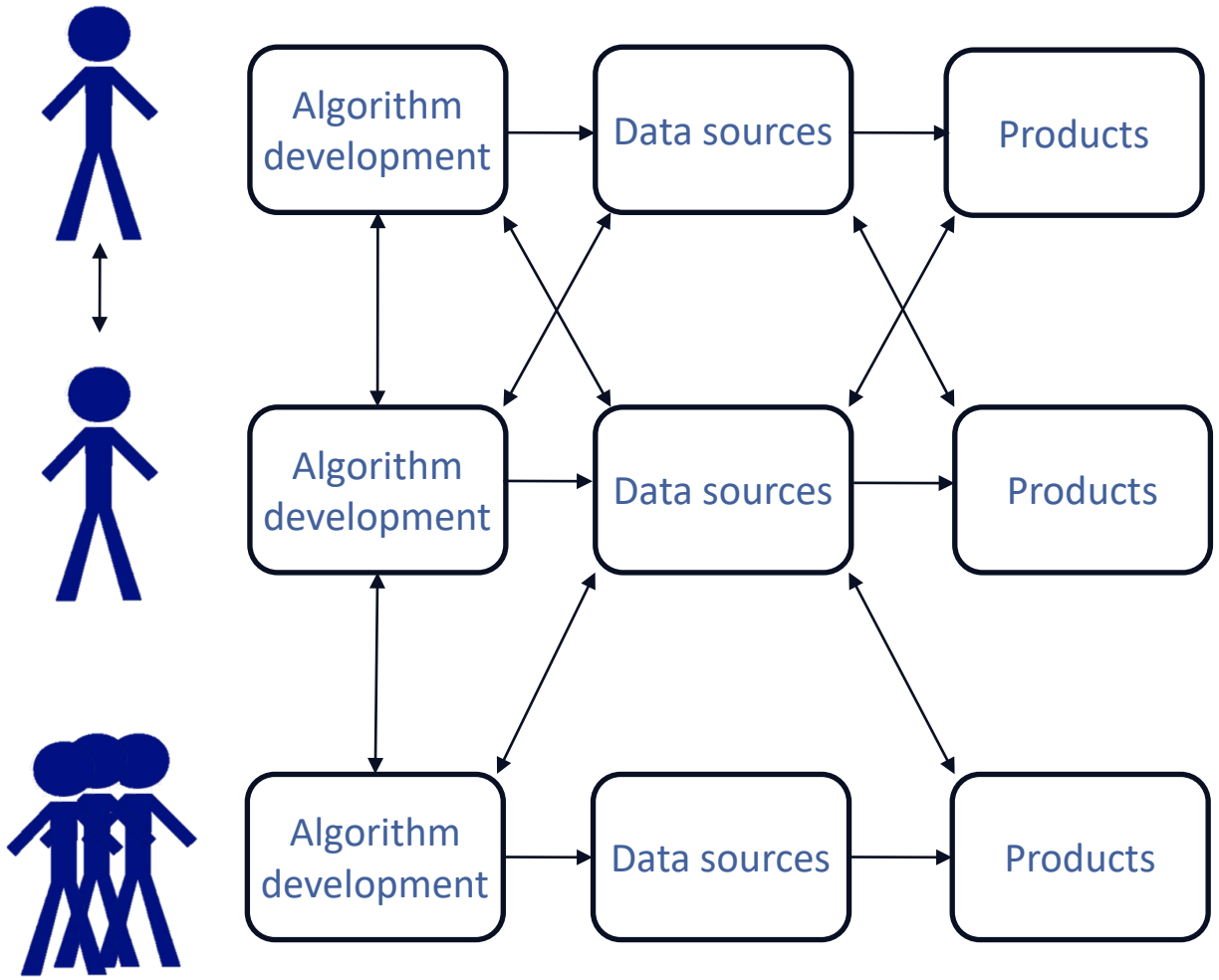
# Algorithm Development Today




**PRIVACY PROTECTION** results in:

- Training often limited to the datasets each individual project/person is able to collect
- One project cannot easily build on the experience and knowledge of a peer-project
- Sources of auxiliary data are not easy to share

# This is what we would like to have




## PRIVACY PROTECTION to PRIVACY PRESERVATION

- **NEEDED**: a secure and efficient way to share data and algorithms that can be trusted.
- **ADOPT PROMISING TECHNOLOGIES** : synergic use of technologies from non space domains to create a platform addressing IPR concerns:
  - Blockchain and Smart Contract
  - Decentralised Computing
  - Inter Planetary File System (IPFS)
- **A SUCCESSFUL PLATFORM SHOULD LEAD TO**:
  - More collaborations and faster development
  - Better algorithms and hence better science
  - Applications across multiple domains
  - Contribute to removing barriers to massive exploitation of space data

# BLENDED Activities

- **Develop a decentralised platform for data and algorithms to facilitate the training of Deep Learning models using data sets owned by different peers.**
- **Implement decentralised learning managed by smart contracts, with encryption to ensure secure use of resources and IPFS for linking and accessing these resources.**

**INVESTIGATION**

**Demonstrate the use of the platform with a *Urban Expansion Monitoring* study case:**

- **using Sentinel 1 & 2, Landsat 5TM, ERS1/2 and auxiliary data**
- **three dissimilar test locations**
- **applying Deep Learning model and supervised learning for automatically generating ground truth data.**

**PRACTICAL EXPERIENCE**

**Collect performance data and perform benchmarking.**

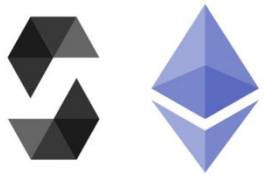
**Make recommendations for improving the architecture and for further using the platform with other space data.**

**DISCUSS and REPORT**

# The BLENDED Decentralised Training Platform

# The Main Technologies used in BLENDED

- Blockchain-based controller
  - Ethereum (private Blockchain)
  - Smart contracts implemented using Solidity



- Decentralised
- Trusted
- Controlled transactions

- IPFS Storage
  - Peer-to-peer decentralised file system
  - Immutability of data files / folders (content-based addressing)
  - Files metadata registered in a Resource Catalogue


- Decentralised
- Immutable
- Universally accessible

- Execution Platform
  - SpaceApps' Automated Service Builder (ASB)
  - Cloud-based flexible and scalable processing framework


- Workflow orchestration
- Distributed processing

- High-Performance Computing (HPC) Environment
  - IT4Innovations' HPC
  - HPC-as-a-Service concept implemented with HEAppE Middleware

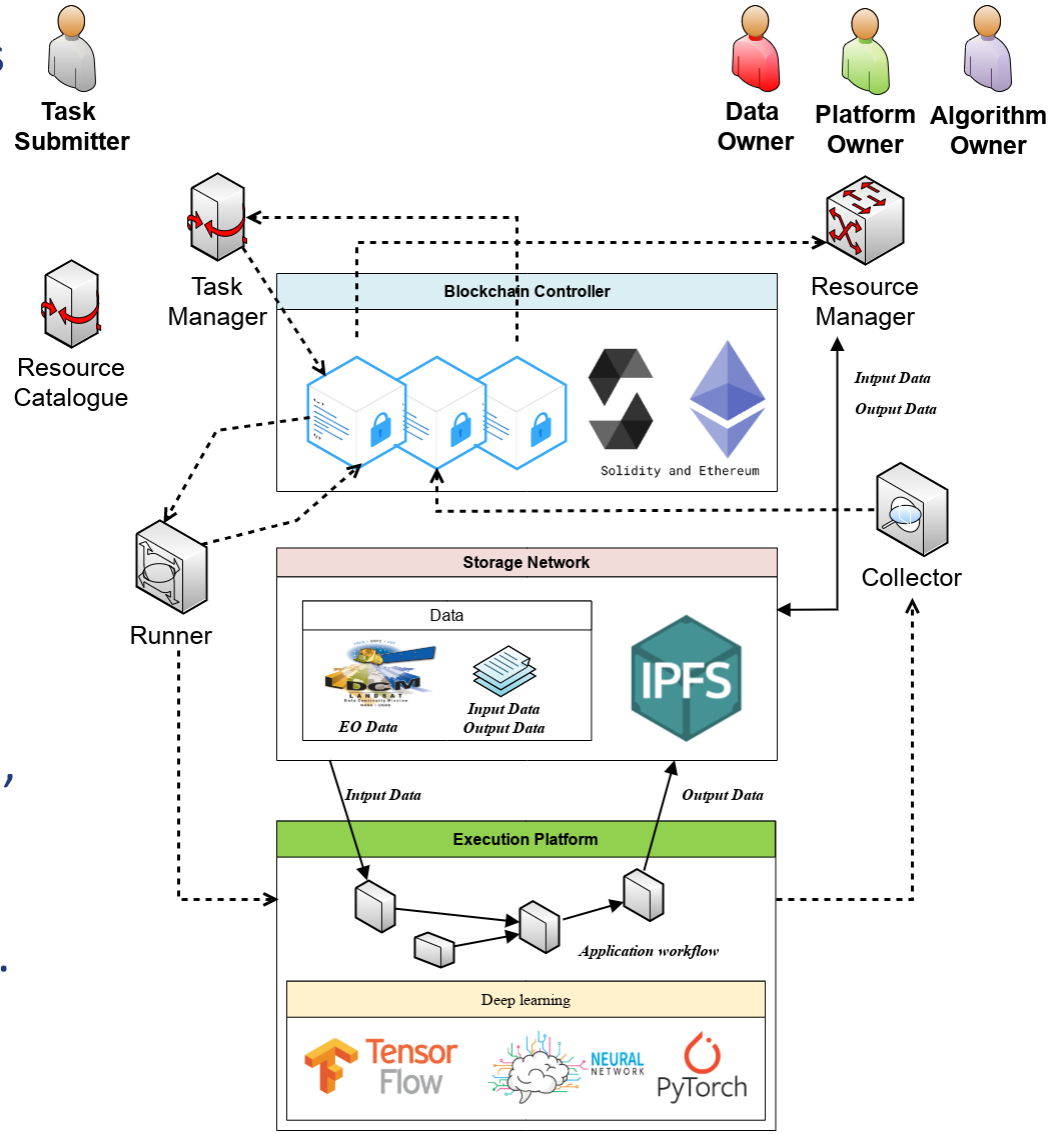

- Parallel processing
- Optimised for GPUs

- Deep Learning Tools and Algorithm
  - Tensorflow/Keras, PyTorch, Horovod
  - Multi-GPU, multi-node ready algorithm


- Comprehensive
- Flexible
- Community

# Processing Task Life-Cycle

1. The **Task submitter** selects an application, input datasets and a target execution platform in the Catalogue.
2. The **Task submitter** configures a new task (new contract) and may provide his public key for outputs encryption.
3. a. The **Algorithm owner** acknowledges the use of his application.
   b. The **Platform owner** provides a public key for inputs encryption and acknowledges the use of his platform.
   c. The **Data owner** uploads (optionally encrypts) input datasets and acknowledges the resource request.
4. The *Runner* node triggers the execution.
5. The application is orchestrated in the Execution Platform, training tasks are executed in the HPC, and the outputs are encrypted and stored in the IPFS network.
6. The *Collector* node updates and locks the smart contract.
7. The **Task submitter** accesses the execution results, downloads and decrypts the generated products.



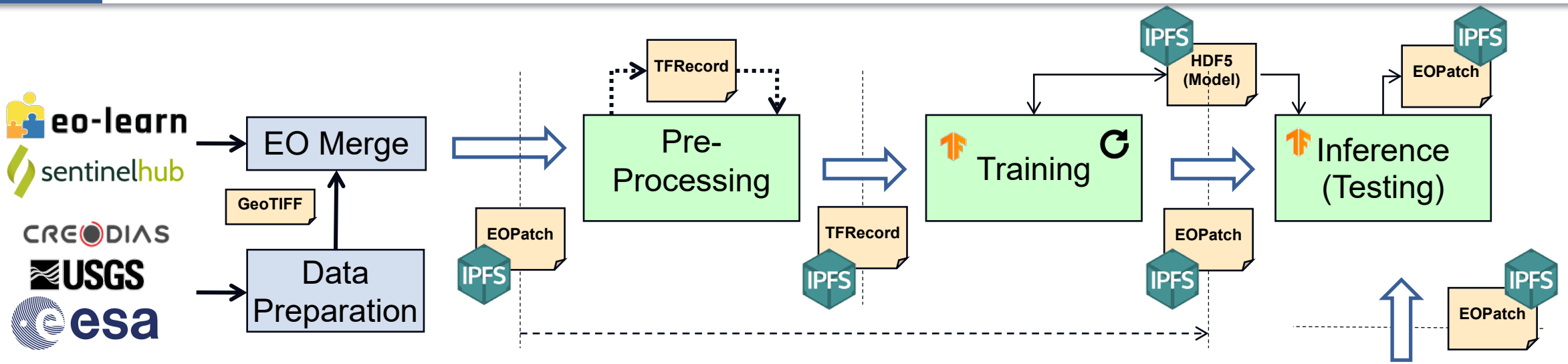
## Urban Expansion Monitoring

Detection and prediction of urban changes using Deep Learning

# The Urban Expansion Monitoring Study Case

- The Deep Learning Neural Network is trained with synthetic aperture radar (SAR) and optical (multi-spectral) remote sensing data.
- Two epochs were considered; for each we trained one neural model:
  - Epoch 1991 - 2011 with ERS-1/2 (SAR) and Landsat 5TM (optical)
  - Epoch 2017 - mid-2021 with Sentinel 1 (SAR) and Sentinel 2 (optical)
- Three dissimilar test locations:
  - Liège (Belgium), Rotterdam (The Netherlands) and Limassol (Cyprus)
- Remote sensing and ground truth data were retrieved from different sources, incl. ESA, USGS, SentinelHub, and official authorities (e.g. cadastral data).

# Processing Pipeline




1. Data Preparation: AOI-based selection of data from different sources. Re-projection, co-registration, terrain correction, cloud-masking, etc. Outputs *GeoTIFFs*.
2. EO Merge: Converts *GeoTIFFs* into *EOPatches* (eo-learn format, one file per observation, per source). *Alternatively*: downloads *EOPatches* directly from SentinelHub using eo-learn.
3. Pre-Processing: Performs windowing, tiling, synthetic labelling. Generates *TFRecords* that can be directly ingested in the NN during training.
4. Training: Ingests the *TFRecords* and trains the model, stores the trained model in *HDF5* format.
5. Create Ground Truth: Creates data for evaluating the network predictions in *EOPatch* format.
6. Inference: Uses trained model and prepared data to run prediction tasks.
   Testing: Uses ground truth data to validate the predictions.

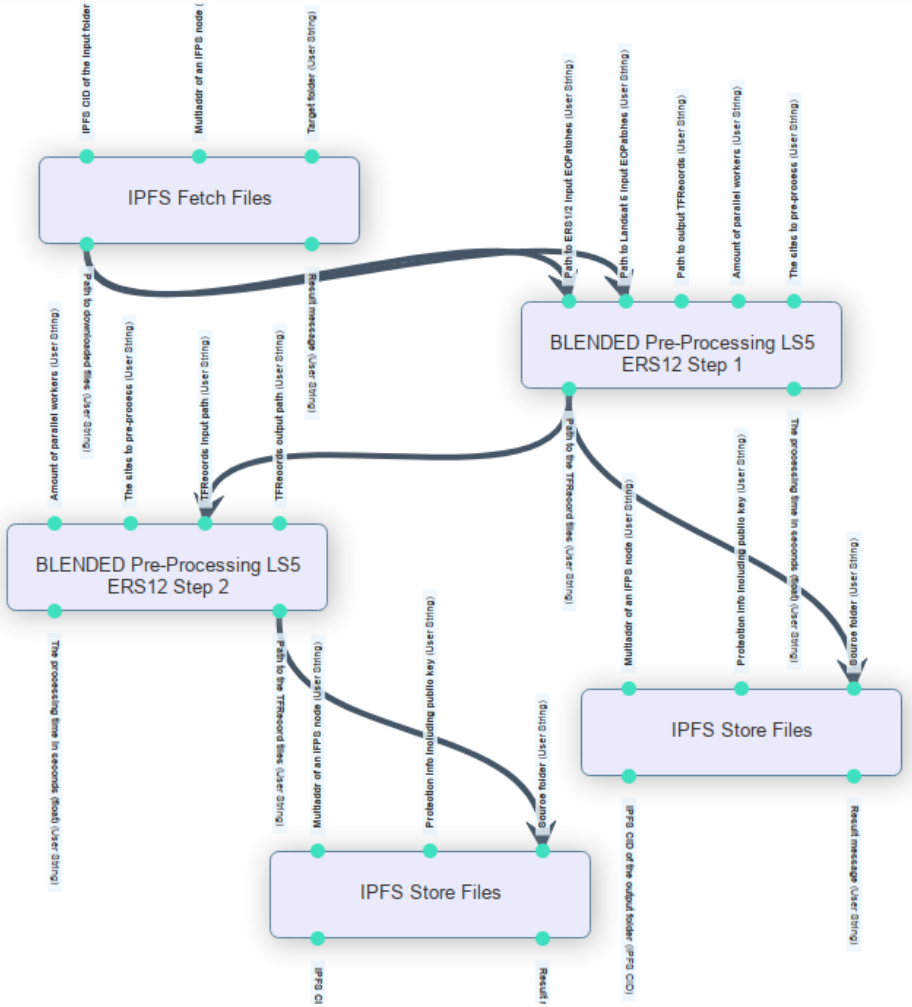
Create Ground Truth

Cadastre, etc.

# Example Workflow: EO Data Pre-Processing

## Workflow steps

1. Fetch input **EOPatch** files from the IPFS.
   One file per observation per source.
   Decrypt if necessary.
2. Pre-processing Step 1.
   Intermediate outputs: **TFRecords** containing individual observations (mixing optical and SAR layers)
3. Pre-processing Step 2.
   Outputs: **TFRecords** containing temporal stacks (windowed and labelled)
4. Store the **TFRecords** into the IPFS.
   Encrypt the files if requested by the Task Owner.




*Workflow created in the interactive workflow editor of ASB*

# Task Requests recorded in Smart Contracts

## Task Smart Contract – Controlled Transactions

```
$ task-manager status --address $CONTRACT
Task owner: 0x473622506C1Ed1461c146FF21634151B1E9f614b
Task state: Requests locked
Results security policy:
{}

Platform UUID: 627631768750495581l060bc8d891
Platform owner: 0x473622506C1Ed1461c146FF216
Platform status: Answer pending
Platform execution id:
Platform security policy:
none

Application UUID: 69359236b5c3473c965179bd71
Application name: admin/blended-testing-ls5_
Application owner: 0x473622506C1Ed1461c146FF
Application status: Answer pending

3 Resource requests:

Resource owner: 0x473622506C1Ed1461c146FF216
Catalog UUID: 67ba57f171e24f49897a6908cc9ff3
Application Name: ground-truth-cid
Status: Answer pending
IPFS address:

[…]
```

**Requests Locked state**

```
$ task-manager status --address $CONTRAC
Task owner: 0x473622506C1Ed1461c146FF216
Task state: Running
Results security policy:
{}

Platform UUID: 627631768750495581l060bc8
Platform owner: 0x473622506C1Ed1461c146F
Platform status: Acked
Platform execution id: 708
Platform security policy:
{}

Application UUID: 69359236b5c3473c965179
Application name: admin/blended-testing-
Application owner: 0x473622506C1Ed1461c1
Application status: Acked

3 Resource requests:

[…]
```

**Running state**

```
$ task-manager status --address $CONTRACT
Task owner: 0x473622506C1Ed1461c146FF21634151B1E9f614b
Task state: Finished
Results security policy:
{}

Platform UUID: 627631768750495581l060bc8d8916db
Platform owner: 0x473622506C1Ed1461c146FF21634151B1E9f614b
Platform status: Acked
Platform execution id: 708
Platform security policy:
{}

Application UUID: 69359236b5c3473c965179bd7164ea49
Application name: admin/blended-testing-ls5_ers12
Application owner: 0x473622506C1Ed1461c146FF21634151B1E9f614b
Application status: Acked

3 Resource requests:

[…]

1 Task result:

Result owner: 0x473622506C1Ed1461c146FF21634151B1E9f614b
IPFS address: QmSwfLpJngGX6YU9wvdCXWR786MxjXBiNjv2Hxxws7deVG
```

**Finished state**

# Achievements, Benefits and Potential Continuation

# Achievements – Blockchain and IPFS

*Assessment of IPFS technology, Blockchain smart contract and DL models to prototype a decentralised storage framework and a training platform for EO Data*

- The Blockchain technology has been assessed and a private Ethereum network has been deployed.
  - This private network may be extended or the BLENDED platform may be re-configured to use a public Ethereum network instead.
  - A smart contract factory has been created using the Solidity language. This allows registering task execution requests and managing their lifecycle.
- Benchmarking has shown that IPFS is a reliable and efficient technology for providing a scalable decentralised storage system.
- The Execution Platform built on Space Applications Services' ASB is integrated for running distributed applications, including in cloud platforms and HPCs.

# Achievements – Study Case

*Execute and evaluate the Urban Expansion Monitoring study case.*

- Time Series Analysis and Prediction algorithms implemented, integrated and demonstrated in the context of the study case.
- Model trained for urban and imperviousness changes with Rotterdam and Limassol data and used to predict changes in Liège.
- Complete end-to-end pipeline has been created which includes manual steps and processing steps executable in the BLENDED platform.
- Integrated algorithms cover 3 pipeline steps: EO data pre-processing, model training, and inference and testing.

# Benefits and Potential Continuation



## *Benefits*

- Blockchain smart contracts control the lifecycle of application execution requests.
- Ownership of the input and output data is preserved (IDs stored in the smart contracts).
- Data and models may be secured (encrypted) ensuring they are only readable by selected applications and platforms.
- Encrypted outputs are only readable by the users who requested them. This means provider Intellectual Property is protected and users are sure their data are kept private.

## *Potential continuation*

- The generic concepts developed in BLENDED can be applied
  - in any process where full traceability (governance and provenance) is required
  - where the wish of parties is not to have a centralised authority
  - with space or non-space applications.
- The BLENDED platform may be improved and demonstrated in a much more complex distributed setup where interfaces are more matured.

# Thank you

Bernard Valentin bernard.valentin@spaceapplications.com
Leslie Gale leslie.gale@spaceapplications.com

# Actors of the BLENDED Training Platform

**Task submitter –** the user who wants to obtain the results of a processing a task. A task is defined in terms of data, algorithm and computational resources


**Data owner** – provides data for the execution of the tasks, in defined formats.

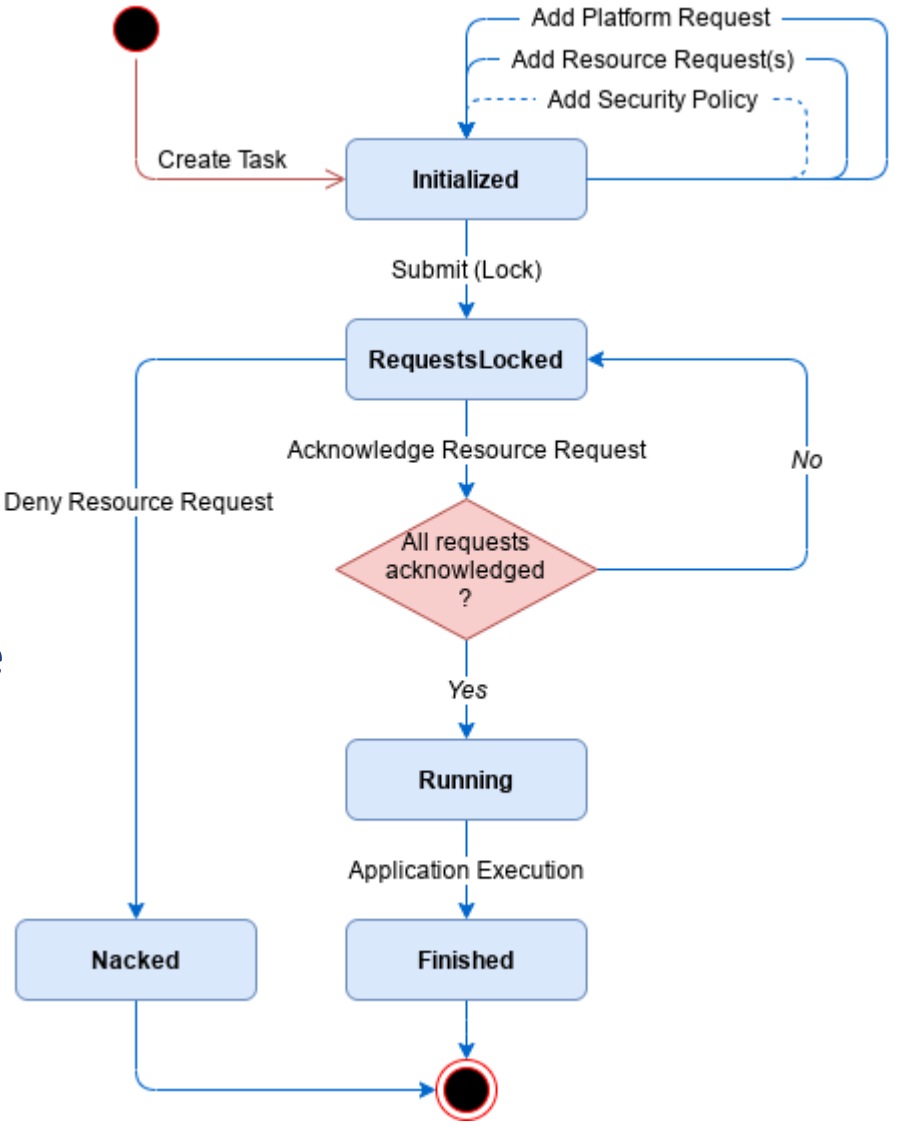
**Algorithm owner** – provides the algorithms or applications that use the input data and create useful outputs, e.g. a stakeholder providing trained ML/DL models.

**Platform owner** – provides computational resources that can be used to execute the requested tasks, with the provided data, models and algorithms.

# Smart Contract Life-Cycle

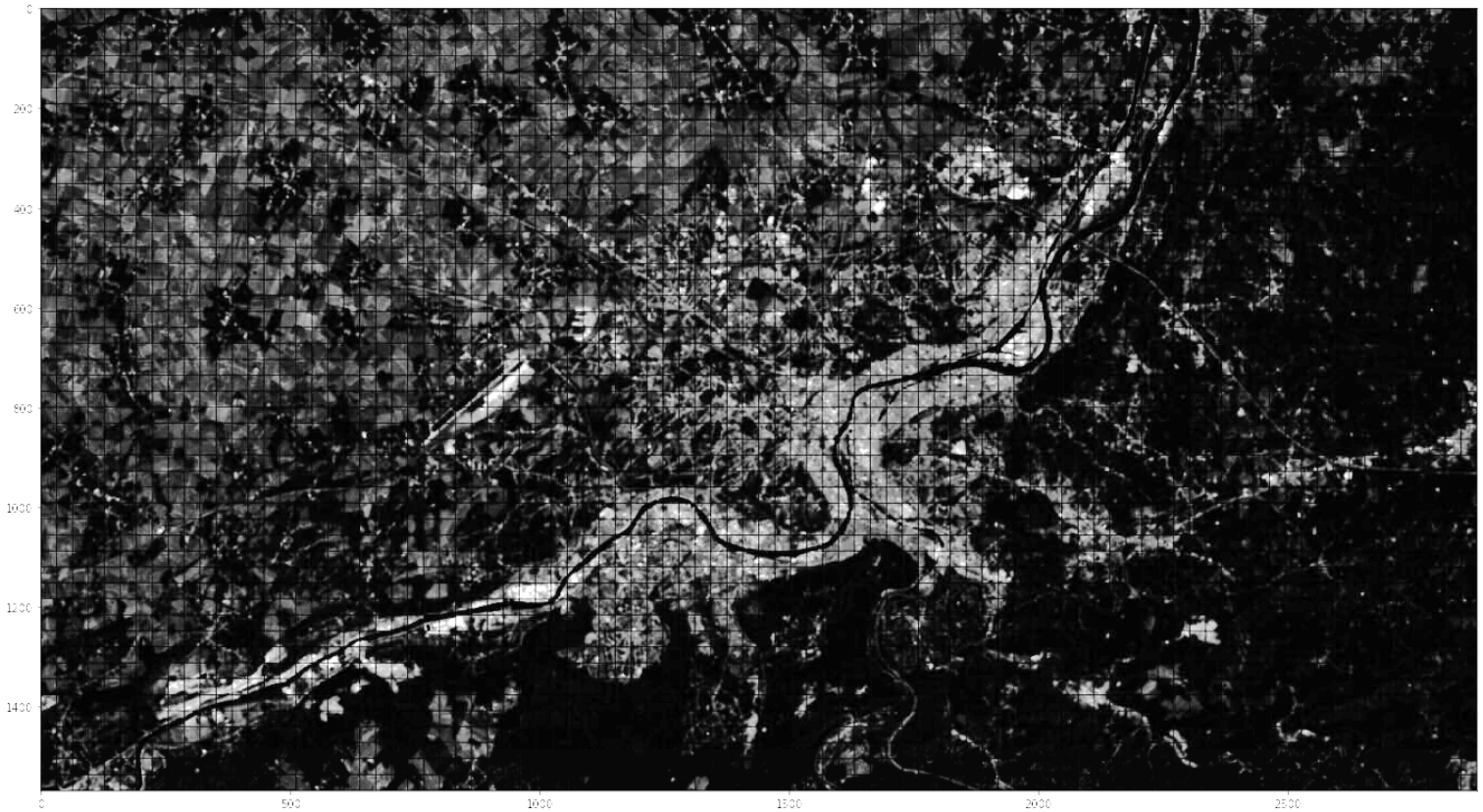
Solidity smart contracts track the full life-cycle of a *processing Task*, including inputs and outputs, and strict checks on state changes:

- **Initialized**: Task / contract created and being configured by the Task owner.
- **Requests Locked**: Contract locked and submitted. The contract remains in this state until all the resource requests have been acknowledged (or a request has been rejected).
- **Running**: All the resources have been acknowledged and the application execution has been initiated / is on-going.
- **Finished**: The application execution is complete, the smart contract has been concluded.
- **Nacked**: At least one resource request has been rejected.



Model trained with Rotterdam and Limassol data and used to predict urbanisation in Liège.

Urban and imperviousness changes detected in the Landsat 5 & ERS-1/2 time series (1991-2011).




*Heatmap (noise not filtered)*