

Note de synthèse sur la faisabilité des services et plateformes de type "Data Cube"

Projet EO4DM - Fonds Pacifique

1 Introduction

Dans le cadre du projet EO4DroughtMonitoring (EO4DM), une phase préalable a été menée afin d'évaluer la faisabilité d'utilisation et d'intégration des services de type Data Cube dans la chaîne de production opérationnelle des indicateurs satellites de sécheresse végétale. Nous rappelons dans la figure ci-dessous le principe général des différentes sous-chaînes telles qu'elles ont été développées initialement :

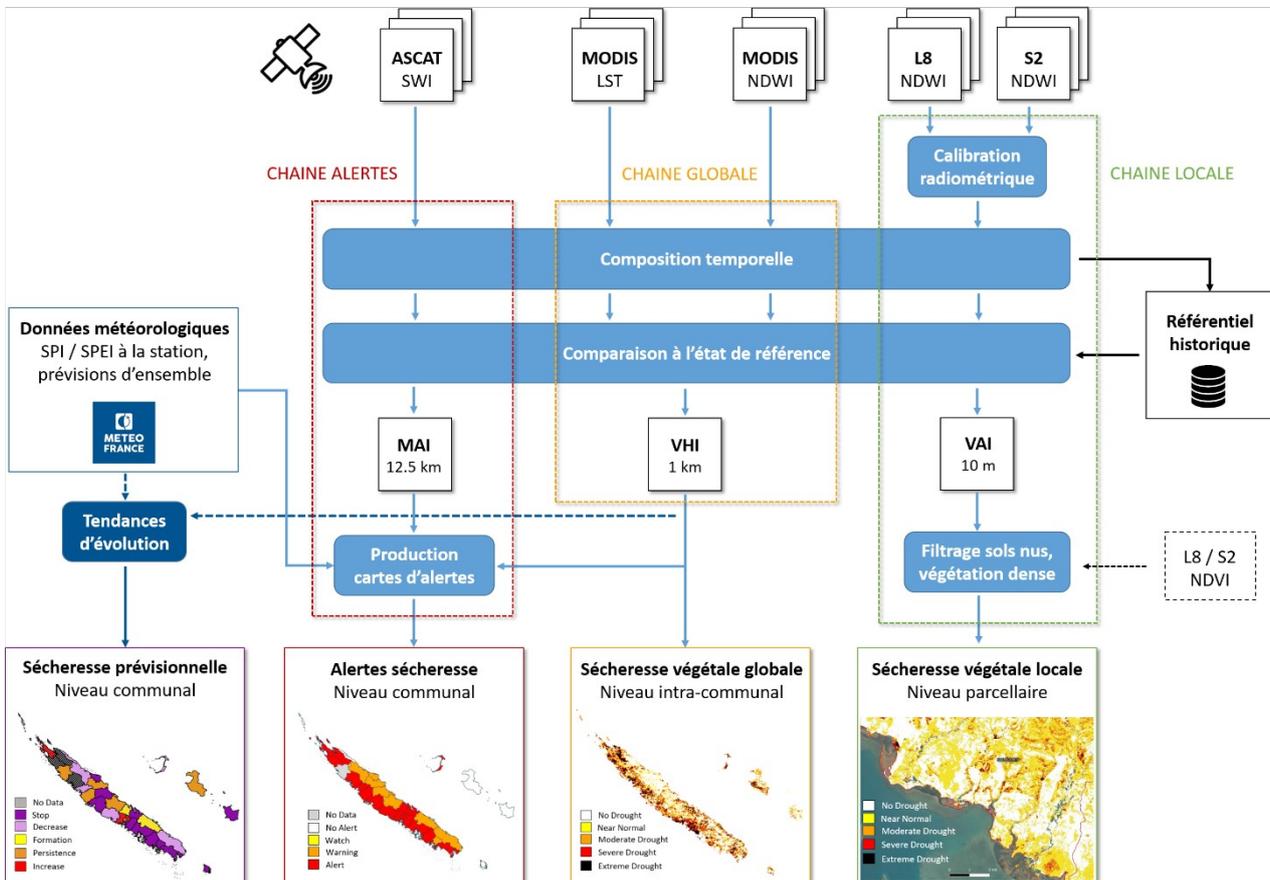


Figure Chaîne de production de l'indicateur sécheresse

Pour rappel, un data cube désigne un tableau de données à plusieurs dimensions. Cette représentation est notamment utilisée pour faciliter le traitement de grande quantité de données comme les séries d'images satellites. En particulier dans le domaine de l'Observation de la Terre, plusieurs services/plateformes ont été développés ou sont en cours de développements se basant sur des data cubes en ligne d'images satellites.

L'apport de ces services/plateformes dans le projet EO4DM serait d'une part de s'affranchir du stockage, téléchargement et prétraitement en local des produits satellites (indices précalculés, produits composites, nuages masqués, etc.), et d'autre part de permettre la réalisation de certains traitements directement en ligne sur le cube.

2 Services/plateformes étudiés

Nous nous sommes concentrés sur les services dédiés à l'Observation de la Terre et proposant un panel important de produits et fonctionnalités en imagerie spatiale. Ceux que nous avons étudiés sont parmi les plus connus, comprenant des plateformes reconnues au niveau mondial, ainsi qu'une plateforme spécifique à la région du Pacifique. Certains services sont en cours de développement, donc non intégrables en l'état à date dans le projet en cours EO4DM, mais nous les avons tout de même ajoutés à l'étude car présentant un intérêt potentiel pour le futur. Voici la liste :

- Google Earth Engine :
Plateforme développée par Google mettant à disposition un large catalogue de produits sur l'ensemble du globe, ainsi que plusieurs fonctionnalités de stockage (Google cloud, Google drive) et de traitements en ligne des produits (Earth Engine Code Editor, API Python).
- Euro Data Cube / Sentinel Hub :
Plateforme de l'Agence Spatiale Européenne (ESA) se basant sur le service Sentinel-Hub d'accès et de traitement en ligne des produits satellites open-data et commerciaux. Cette plateforme intègre, en plus, des services de stockage et de traitement haute performance sur le cloud adaptés aux données cubes (EDC EOxHub). Elle inclut également des services de création et mise en place de data cubes par les utilisateurs (EDC xcube Generator).
- Data-Terra :
Infrastructure de recherche mettant à disposition des catalogues de produits via leur service en ligne d'accès aux produits très haute résolution (Dimamis) ainsi que les services du pôle THEIA qui rassemble divers produits thématiques spécifiquement dédiés à l'étude des surfaces continentales. Un projet en cours (GAIA-DATA) vise à développer des services de stockage et de traitement en ligne des produits.
- Digital Earth Pacific :
Plateforme en cours de développement portée par la Communauté du Pacifique Sud (CPS) regroupant divers produits satellites opérationnels (prétraités) sur le Pacifique et visant un accès gratuit de ces données aux différents territoires de la région.

3 Critères d'évaluation

Voici la liste des critères qui ont été utilisés pour évaluer la faisabilité et l'intérêt des services dans le projet, regroupés en trois catégories :

- 1) Catalogue :

- Capteurs satellites disponibles : catalogue de capteurs en relation avec notre besoin
 - Niveaux de prétraitements : indices calculés, synthèse temporelle, filtres nuages, interpolation spatiale (gap-filling)
 - Délai de mise à disposition : temps entre la date d'acquisition des produits et leur disponibilité sur les plateformes
- 2) Fonctionnalités proposées :
- Possibilité d'opérer certains traitements en line (si pas déjà proposés dans le catalogue)
 - Interfaces de programmation (avantages/limites de l'API, langage, etc.)
 - Capacité de stockage sur cloud et Import/Export de données
 - Coûts éventuels
- 3) Autres critères :
- A cela s'ajoutent des critères annexes tels que la présence ou non d'une forte communauté autour du service (forums, codes disponibles, aide en ligne...) ou encore l'ergonomie des plateformes (facilité de prise en main des services, explorateur, prévisualisation des produits avant téléchargement...).

Un mois après le lancement du projet EO4DM, nous avons appris le décommissionnement des deux satellites embarquant le capteur MODIS : le satellite Terra décommissionné courant octobre 2022, et le satellite Aqua qui sera décommissionné courant 2023. Nous avons donc vérifié dans cette étude la disponibilité de produits pouvant remplacer dans la chaîne les produits optiques et thermiques moyenne résolution issus de MODIS, à savoir : les produits VIIRS pour la partie optique, et Sentinel-3 (S3) pour le thermique.

4 Synthèse générale et préconisations

Le tableau 1 ci-après fait la synthèse de l'évaluation pour les quatre services. Les résultats sont présentés pour chaque critère (colonnes du tableau) et chaque produit/capteur attendu en entrée de la chaîne de traitement EO4DM (lignes). *Ici nous n'avons pas intégré le délai de mise à disposition des produits (voir analyse spécifique en section suivante).*

Tout d'abord, concernant le catalogue, nous remarquons qu'aucune plateforme ne met à disposition l'ensemble des capteurs nécessaires au projet. Néanmoins, les quatre plateformes proposent les capteurs haute résolution (Sentinel, Landsat) présentant le plus d'intérêt ici. Seuls Google Earth Engine (GEE) et Euro Data Cube (EDC) intègrent les produits moyennes résolution MODIS et S3. Pour les produits S3, seuls les produits de niveau L1B sont fournis à chaque fois (pas de correction atmosphériques). Seul GEE propose les produits thermiques VIIRS, et aucune plateforme ne propose les produits ASCAT ou SMAP-S1.

Pour ce qui est des niveaux de prétraitements déjà réalisés, GEE propose seulement un indice NDWI MODIS (pas d'indices haute résolution). EDC et Digital Earth Pacific proposent quelques indices de végétation (agricoles) haute résolution précalculés. Data-Terra ne propose pas de produits déjà prétraités à ce stade. Aucune plateforme ne propose de produits Landsat-7 corrigés (gap-filling).

En termes de traitements réalisables en ligne, seules les plateformes GEE et EDC proposent des services opérationnels à ce jour. A priori, Data-Terra vise à

permettre ces services (*en cours de développement*). Digital Earth Pacific ne semble pas inclure ces fonctionnalités, proposant a priori uniquement des services d'accès aux produits.

Les interfaces de programmation proposées par GEE sont gratuites. Il s'agit d'un éditeur de texte en ligne codé en JavaScript (Earth Engine Code Editor) ainsi qu'une API Python pouvant être utilisée en ligne sous forme de notebooks (Google Colab) ou en local afin d'être intégrée aux applications des utilisateurs (librairie *earthengine-api*). EDC propose en revanche des services payants (cf. prix standard minimum affiché dans le tableau) : une API Python sentinel-hub intégrant plusieurs librairies pour le traitement d'images et de données cubes (analyse multi-dimensionnelle via la librairie *xcube*), ainsi qu'une API Euro Data Cube (EDC EOxHub) pour accéder à un environnement plus performant.

Pour ce qui est de l'espace de stockage disponible, GEE propose un espace gratuit de 250 Go (pouvant être augmenté par paiement mensuel). EDC propose un espace gratuit durant la première année d'utilisation, qu'il faut ensuite payer au mois et en fonction de la quantité de données stockées (0.03€/Go/mois). Pour les deux services (GEE et EDC), l'import de produits par l'utilisateur sur la plateforme est possible, tout comme l'export des produits prétraités issus de la plateforme. EDC demande néanmoins de disposer d'un entrepôt de type bucket Amazon S3 pour intégrer ses propres données.

Préconisations :

Au vu de ces premiers résultats, les services/plateformes qui présenteraient à ce stade le plus d'intérêt sont GEE et EDC. Ils présentent à la fois un catalogue de produits répondant en majeure partie aux besoins du projet tout en permettant des fonctionnalités de traitement et stockage en ligne. Pour ce qui est de Data Terra et Digital Earth Pacific, ces plateformes ne sont pas assez matures pour pouvoir les intégrer au projet en cours.

GEE présente une interface de visualisation et de programmation claire et facile d'utilisation, avec un panel de librairies prêtes à l'emploi. En revanche l'export en local se fait via un espace intermédiaire dédié de type Google Drive ou Google Cloud, qui s'avère être limité pour les versions gratuites (15 Go pour G-Drive) et avec un temps d'export relativement long (~10 min pour une tuile de 500 Mo). Cela peut donc être limitant si l'utilisateur souhaite exporter une masse importante de données de manière récurrente.

→ *Au vu de ces caractéristiques, le service GEE pourrait être utilisé dans ce projet afin de prétraiter les produits directement en ligne via l'interface de programmation (API Python notamment). Ces produits prétraités (indices calculés et filtrés avec masque de nuage) seraient ensuite exportés par l'utilisateur chaque mois en local (sur pc ou serveur utilisateur), où serait réalisé le calcul des indicateurs de sécheresse. Etant données les limitations de stockage, ce service serait surtout intéressant pour la mise en œuvre en mode routine de la chaîne.*

EDC se base sur une forte communauté Sentinel Hub qui met à disposition de nombreux codes en ligne via des forums et blogs ([forum sentinel-hub](#), [blog sentinel-hub](#)) et dépôts de scripts pour le traitement d'images satellites ([Github](#), [Customs script](#)). Néanmoins, EDC ne propose pas d'accès gratuit, impliquant un

coût minimal de 30€/mois via Sentinel Hub pour une utilisation maximale de 30 000 unités par mois et 300 unités/min (unité = image 512x512, 3 bandes). Un accès plus adapté aux besoins des entreprises (batch processing) est proposé pour 400€/mois. Il est également possible de faire une demande d'accès gratuit à ce service dans le cadre de [projets sponsorisés par l'ESA](#).

→ *Au vu de ces caractéristiques, le service EDC pourrait être utilisé de la même manière que GEE, c'est-à-dire comme service d'accès et de prétraitement en ligne des produits satellites. Le service minimal (Sentinel Hub à 30€/mois) pourrait suffire à la mise en œuvre du mode routine de la chaîne (~ 2500 unités/mois nécessaires à notre besoin).*

Tableau : Synthèse des critères d'évaluation pour les différents services (en rouge les éléments en cours de développement et/ou pour lesquels nous n'avons pas encore d'informations sûres)

| SERVICES/PLATEFORMES | CAPTEURS | DISPONIBILITE | NIVEAUX PRODUITS | TRAITEMENTS EN LIGNE | INTERFACE PROGRAMMATION | STOCKAGE CLOUD, IMPORT/EXPORT | POINTS ANNEXES POSITIFS | POINTS ANNEXES NEGATIFS |
|-------------------------------|--|--|---|---|---|---|--|---|
| GOOGLE EARTH ENGINE | Sentinel | S2 = OK S1 = OK | S2 = L2A ESA S1 = GRD - Calibré (pas filtre speckle) | S2 = Nuages, Indices, Composites, Gap-filling S1 = Composites, Speckle | Editeurs en ligne : Earth Engine Code Editor (JavaScript) Google colab (notebook Python) | Stockage Gratuit : G-cloud, 250 Go Stockage Payant : G-Workspace, +10 To=300\$/mois Export Gratuit : 5 Go (G-cloud), 15 Go (G-Drive) | Visualisation rapide via explorateur, Prise en main facile et efficace de l'éditeur, Communauté (codes), Traitements gratuits | Export drive = ~10 min/tuile (500 Mo) |
| | Landsat | L7, L8, L9 = OK | L2 - Collection 2 (L7 non gap-fillé) | Nuages, Indices, Composites, Gap-filling | API Python en local : librairie <i>earthengine-api</i> | Import données : OK | | |
| | MODIS | OK | Optique = Réflectances, 8j, 1j Thermiques = LST, 8j, 1j NWI = Journalier (pas composites) | Nuages, Indices, Composites | | | | |
| | VIIRS | OK | Optique = Réflectances, 1j Thermique = NON | Nuages, Indices, Composites | | | | |
| | S3 | NON (que radiances L1B) | - | - | | | | |
| | ASCAT | NON | - | - | | | | |
| | SMAP-S1 | NON (autres produits à 10km) | - | - | | | | |
| EURO DATA CUBE / SENTINEL HUB | Sentinel | S2 = OK S1 = OK | S2 = L2A ESA, indices, 10j S1 = GRD (pas calibré) | S2 = Nuages, Indices, Composites S1 = Composites, Speckle | SENTINEL HUB : API Python en ligne ou local (Payant) librairie <i>sentinelhub</i> -> Standard (min)=30€/mois -> Entreprise (bach process) = 400€/mois | SENTINEL HUB (Payant) : Besoin bucket Amazon S3, 1 Go = 0,03€/mois (1 an gratuit) | Communauté (codes) | Pas d'accès restreint gratuit |
| | Landsat | L7, L8, L9 = OK | L2 - Collection 2 (L7 non gap-fillé) | Nuages, Indices, Composites, Gap-filling | EURO DATA CUBE : espace haute performance en ligne (Payant) Jupyter notebooks (Python) -> Standard (min) 2CPU, 8 Go RAM = 100€/mois -> Perf GPU 16 Go, 4 CPUs 16Go RAM = 600€/mois | EURO DATA CUBE (Payant) : -> Standard (min) 30 Go = 100€/mois | | |
| | MODIS | OK | Optique = Réflectances, 16j Thermiques = NON | Nuages, Indices, Composites | | | Import données : OK | |
| | VIIRS | NON | - | - | | | | |
| | S3 | NON (que radiances L1B) | - | - | | | | |
| | ASCAT | NON | - | - | | | | |
| | SMAP-S1 | NON | - | - | | | | |
| DATA-TERRA | Dinamis : uniquement S2 ESA (tout territoire) THEIA : S2 THEIA et L8 THEIA (uniquement NC...) | | | Prévus dans le cadre du projet en cours GAIA-DATA... | Prévus dans le cadre du projet en cours GAIA-DATA... | Pas d'informations à ce stade... | Explorateurs informatifs, claires et synthétiques (Dinamis, Theia) | Services non opérationnels à ce stade |
| DIGITAL EARTH PACIFIC | Sentinel | S2 = OK S1 = pas encore disponibles | S2 = L2A ESA, indices agri, NDWI (lequel ?) S1 = GRD | A priori pas de traitements... | A priori pas d'interfaces de programmation... | A priori pas d'espace de stockage... | Earth explorer (en phase "pilote"), Indices précalculés | Services non opérationnels à ce stade (pas avant 2024 - 2025) |
| Landsat | Pas encore disponibles | L2 - Collection 2, indices agri, NDWI (?) | | | | | | |
| MODIS | Pas encore disponibles | Optique = Réflectances, 8j Thermiques = LST, 8j | | | | | | |
| VIIRS | NON | - | | | | | | |
| S3 | NON | - | | | | | | |
| ASCAT | NON | - | | | | | | |
| SMAP-S1 | NON | - | | | | | | |

5 Délai de mise à disposition et préconisations

Le tableau 2 présente pour chaque produit la date d'acquisition la plus récente disponible sur les plateformes par rapport à la date de consultation (J=0). Il s'agit ici de vérifier que les plateformes étudiées permettent une mise à disposition des produits qui soit suffisamment rapide pour répondre aux besoins du projet, à savoir une mise à jour à minima toutes les dizaines de jours des indicateurs sécheresse en sortie de la chaîne.

Ce travail a été fait sur les deux plateformes GEE et EDC sélectionnées en section précédente. Pour comparaison, nous avons également fait ce travail pour trois plateformes plus "classiques" ne présentant pas de services/fonctionnalités de type Data Cube : Sci Hub Copernicus, Earth Explorer USGS, AppEARS USGS.

Concernant les produits haute résolution (S1, S2, L8/L9), toutes les plateformes fournissent des produits dans un délai inférieur à la dizaine de jours. Nous remarquons que GEE présente un léger retard de mise à disposition par rapport aux autres plateformes, notamment pour les produits Landsat et Sentinel-1.

Pour ce qui est des produits moyenne résolution (MODIS, VIIRS, S3), le délai est plus long pour les produits MODIS, allant de 12 jours de délai sur GEE pour les acquisitions journalières jusqu'à 3 semaines dans le cas des images composites 8j obtenues sur GEE et EDC. La plateforme AppEARS permet en revanche de réduire ce délai en passant en dessous des 10 jours pour les produits journaliers. En revanche, les produits journaliers VIIRS de réflectances (optique) sont fournis dans les 5 jours sur GEE, ce qui reste comparable aux autres plateformes classiques. Concernant les produits journaliers S3 de températures de surface, ceux-ci n'étant pas fournis par les plateformes data cubes nous avons donc regardé leur mise à disposition sur la plateforme Sci Hub. Il s'avère que le délai ici est très court (2 jours).

Préconisations :

Au vu de ces résultats et de ceux précédents, nous préconisons d'utiliser la plateforme GEE pour les produits haute résolution (S1, S2, L8/L9) et les produits moyenne résolution optique (VIIRS). Celle-ci propose le plus de produits via un même service, avec des fonctionnalités et des interfaces intéressantes, et ce avec des délais de mise à disposition des produits qui restent en adéquation avec les besoins du projet. Les produits thermiques S3 seront quant à eux fournis via la plateforme Sci Hub. Celle-ci ne proposant pas de fonctionnalités de traitements, nous l'utiliserons uniquement comme moyen d'accès aux produits brutes.

6 Exemple d'utilisation : Google Earth Engine

Les principales étapes de pré-traitements des produits satellites ont été implémentées et testées en utilisant l'interface de Google Earth Engine. Le tableau 3 ci-dessous liste pour chaque phase de prétraitement les fonctions de la chaîne que nous avons recodées via l'API Python de GEE. Pour la plupart, celles-ci se retrouvent sur cette API.

L'export d'une image de ndvi calculée à partir de réflectances S2 fournies par GEE a été testé à partir de l'API Python en local (installation sous conda de la librairie *earthengine-api*). L'image finale a été exportée sur un compte google drive standard (gratuit) en une dizaine de minutes, puis téléchargée sur une machine en locale. Le script python permettant cela est présenté en Annexe. *Afin d'automatiser la phase d'authentification nécessaire au lancement de l'API, un compte de service google a été créé. Celui-ci permet la génération d'une clé utilisateur que le script peut ensuite lire à chaque fois de manière automatique,*

Tableau 3 : Tests d'implémentation des étapes de prétraitements sur Google Earth Engine

| PHASES | Fonction Chaîne | Fonctions GEE | Commentaires |
|-------------------------|--------------------------------------|--|--|
| <i>preprocSentinel2</i> | calcul indices NDVI, NDWI | <i>image.normalizedDifference</i> | |
| | masque nuages, ombres | - Filtrage strict : clouds, cirrus (QA60) - Filtrage proba : sen2cloudless (MSK_CLDPRB) | sen2cloudless = améliore filtrage strict en considérant également ombres |
| | masque saturation | bande présente (SCL) | Très faible impact (<1%) |
| | masque geo (pente, illum) | bande présente (SCL) | Très faible impact (<1%) sauf "Sun low" |
| | masque mer/continent | <i>image.clip</i> | |
| | masque outliers | ok | |
| | reprojection | ok | |
| <i>preprocLandsat</i> | calcul indices NDVI, NDWI | <i>image.normalizedDifference</i> | |
| | masque nuages, ombres | bande présente (qa_pixel) | |
| | masque saturation | | Très faible impact (<1%) |
| | masque terrain | | Très faible impact (<1%) |
| | masque aérosol | bande présente (qa_aerosol)) | |
| | masque mer/continent | <i>image.clip</i> | |
| | masque outliers | ok | |
| | ré-échantillonnage gap-filling L7 | <i>image.resample</i> <i>image.focal_mean</i> | |
| <i>preprocMODIS</i> | calcul indices NDVI, NDWI | <i>image.normalizedDifference</i> | |
| | conversion LST | ok | |
| | masque nuages NDVI, NDWI | QA et QA Bitmask | |
| | masque nuages LST | Déjà appliqué | |
| masque mer/continent | <i>image.clip</i> | | |
| <i>exportData</i> | <i>rasterio.write</i> | <i>batch.ExportImage.toDrive</i> | sauvegarde image sur G-Drive |

sans nécessiter l'intervention de l'utilisateur.

7 Adaptations nécessaires pour intégration dans la chaîne de traitement

L'intégration des produits et fonctionnalités proposées par la plateforme GEE dans la chaîne EO4DM implique certaines conditions (prérequis) et adaptations que nous précisons dans cette section.

Seules les chaînes *Globales* et *Locales* de la figure 1 seraient modifiées (les produits ASCAT de la chaîne *Alertes* n'étant pas disponibles via les services Data Cubes). Pour ces deux chaînes, la plateforme GEE opérerait les étapes de prétraitements des indices de surface. Voici une proposition de procédure intégrant la plateforme GEE :

- 1) Prétraitements en ligne (GEE) :
 - Calcul d'indices
 - Filtres qualité (nuages, pente, illumination)
 - Composition temporelle décade (10 j)
- 2) Téléchargement des indices composites
- 3) Traitements en local :
 - Calcul indicateurs sécheresse
 - Composition temporelle mois
- 4) Mise à jour du référentiel historique (stocké en local)

En raison des limitations d'espace de stockage en ligne de GEE, nous préconisons d'appliquer cette procédure uniquement en mode continu, c'est-à-dire pour une mise à jour par pas de 10 jours à 1 mois des indicateurs (et non pour une mise à jour en une seule étape de l'historique). Cela implique donc le calcul ainsi que le stockage préalable des indices historiques en local.

Une autre condition à l'intégration de cette plateforme dans la chaîne est la calibration radiométrique des indices issus de nouveaux produits. Ce travail préalable devra être fait pour les indices de végétation et température de surface issus de MODIS-VIIRS-S3 afin de permettre la comparaison temporelle d'indices provenant de différents capteurs. Ce même travail devra également être réalisé entre les indices S2-ESA et Landsat fournis par GEE (la calibration que nous avons réalisée jusque là se basant sur l'estimation de fonctions statistiques reliant indices S2-THEA et Landsat).

Annexe : Script Python GEE

```
import os
import ee
import time
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from oauth2client.service_account import ServiceAccountCredentials

# --- Authentication / Initialization ---
# Automatic authentication: needs creation of a google service account
path2wkdir = 'D:/MATHIS/0_Projet_Sécheresse/1_Scripts/vrac/GEE'
os.chdir(path2wkdir)

path2key = 'gee_accounts/account_param_mathis' # dossier où se trouve la clé
d'accès au compte de service
service_account = 'eo4dm-service@eo4dm-367504.iam.gserviceaccount.com'
credentials = ee.ServiceAccountCredentials(service_account,
os.path.join(path2key, 'eo4dm_service_key.json'))
ee.Initialize(credentials, project='eo4dm-367504')

# --- Load the Sentinel-2 ImageCollection and filter to specific period
and location ---
countries = ee.FeatureCollection("USDOS/LSIB/2017")
roi = countries.filter(ee.Filter.eq('COUNTRY_NA', 'New Caledonia (Fr)'))
dataset = (ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
.filterDate('2019-11-01', '2019-11-30')
.filterBounds(ee.Geometry.Point(165.4885, -21.5466)))

# --- Get the least cloudy image in the period and clip to NC land ---
image = ee.Image(dataset.sort('CLOUDY_PIXEL_PERCENTAGE').first())
image = image.clip(roi)

# --- Applies scaling factors and cloud mask (strict : clouds and cirrus)
---
def mask_scale_S2clouds(image):
    qa = image.select('QA60')

    # Bits 10 and 11 are clouds and cirrus, respectively
    cloudBitMask = 1 << 10
    cirrusBitMask = 1 << 11

    # Both flags should be set to zero, indicating clear conditions
    mask =
qa.bitwiseAnd(cloudBitMask).eq(0).And(qa.bitwiseAnd(cirrusBitMask).eq(0))

    return image.updateMask(mask).divide(10000)

image_cmasked = mask_scale_S2clouds(image)
```

--- Calculate vegetation INDICES ---

```
image_ndvi = image.normalizedDifference(['B8','B4']).rename('NDVI')
image_ndwi = image.normalizedDifference(['B8','B11']).rename('NDWI')
```

--- Export data to Drive ---

```
# Retrieve the projection information :
```

```
projection = image_ndvi.select('NDVI').projection().getInfo();
```

```
# Create a geometry representing an export region :
```

```
im_geom = ee.Geometry.Rectangle([165.35953389044525,-
21.415867726125654,165.41446553107025,-21.461885455171913])
```

```
# Export the image, specifying the CRS, transform, region, drive folder :
```

```
myfolder = 'INSIGHT_IRD'
```

```
task = ee.batch.Export.image.toDrive(**{
  'image': image_ndvi,
  'description': 'S2_ndvi_201911_NC_Clipped',
  'folder': myfolder,
  'crs': projection['crs'],
  'crsTransform': projection['transform'],
  'fileFormat': 'GeoTIFF',
  'region': im_geom.getInfo()['coordinates'],
  'maxPixels': 120582361
})
```

```
start_exportDrive = time.time()
```

```
task.start()
```

```
while task.active():
```

```
  task_run = 1
```

```
elapsed_exportDrive = round(time.time() - start_exportDrive)
```

--- Download on local machine ---

```
# 0) authenticate to Google Drive (of the Service account) :
```

```
gauth = GoogleAuth()
```

```
scopes = ['https://www.googleapis.com/auth/drive']
```

```
gauth.credentials =
```

```
ServiceAccountCredentials.from_json_keyfile_name(os.path.join(path2key,'eo4dm
_service_key.json'), scopes=scopes)
```

```
drive = GoogleDrive(gauth)
```

```
# 1) Choose starting point by inserting folder name :
```

```
folder_title = myfolder
```

```
folder_id = ""
```

```
# 2) Retrieve the folder id - start searching from root :
```

```
file_list = drive.ListFile({'q': "'root' in parents and trashed=false"}).GetList()
```

```
for file in file_list:
```

```
  if(file['title'] == folder_title):
```

```
    folder_id = file['id']
```

```
break
```

```
# 3) Build string dynamically (need to use escape characters to support single  
quote syntax) :
```

```
str_fold = "\"" + folder_id + "\"" + " in parents and trashed=false"
```

```
# 4) Iterating over files and downloading :
```

```
file_list = drive.ListFile({'q': str_fold}).GetList()
```

```
for file in file_list:
```

```
    filename = file['title']
```

```
    print(f'\nStart downloading {filename}')
```

```
    start_download = time.time()
```

```
    # download file into working directory (in this case a tiff-file)
```

```
    file.GetContentFile(filename, mimetype="image/tiff")
```

```
    # delete file afterwards to keep the Drive empty
```

```
    file.Delete()
```

```
    elapsed_download = round(time.time() - start_download)
```

```
    print(f'End downloading after {elapsed_download} sec')
```