Fostering inclusive regional access to Sentinel-2 data

Gavin Brelstaff - CRS4

2021

Abstract

Here we detail a web based approach that gives an ergonomic view of the Multi-spectral image data available from ESA's Copernicus Sentinel-2 satellite over the island of Sardinia. Web calendars provide the entry point in the form of month-view visualisation. The approach is motivated by reference to a basic HTML JavaScript calendar and then elaborated via the use of web extension applied to Google Calendar pages. Mosaic like multi-tile view-panels allow for a synoptic view of the imagery available over Sardinia on any particular calendar date. Clicking on a given tile start a dialog the allow the tiled product data to be directly downloads. Our focus is on graphical illustration of the concepts involved and in documenting many links to the currently available internet resources: this is in lieu of a formal bibliography. Postscript: An improved calendar design has since been developed and is currently online here: https://crs4.github.io/IsolaS2/src

1 Introduction

Intended for inclusive land-use monitoring and analysis, particularly with EU regions – with its fine 10m^2 ground resolution and extensive range of bands – the multi-spectral image (MSI) data produced by the two Sentinel-2 satellites is available at no cost from ESA's Copernicus data-hub. Its public dissemination has, however, mostly seen uptake among well-resourced institutions such as universities, research entities and corporate enterprises – that have had the back-end facilities and the technical competence necessary to receive and manipulate it. Lower-resourced entities, members of the public, citizen scientists, and nascent micro-enterprises may have originally been deterred by the data's unwieldy volume, unfamiliar formats and large bandwidth requirements. Sentinel-2 MSI products are published in units of 100km^2 tiles, as mapped onto the UTM worldwide geographical grid and until recently only in large zipped archive containing all bands at once. One criticism that might be levelled at the ESA is that, in this age of heightened environmental concern, such voluminous units goes against localised inclusive community projects that might better cope with an option to download say 10 km^2 tiles instead. Nevertheless, inquisitive individuals have always been able to search the Copernicus *Sci-Hub* web-portal and see, in the list of resulting items, web-ready images coarsely sampled from the tiles available for their geographical locations – with the invitation to mark items for eventual download and visualisation at full spatial resolution.

That kind of high-resolution visualisation later came directly into the web-browser with the advent of the portal known as EO Browser (EOB: *Sinergise*, Ljubljana, Slovenia). Its *Google Maps* style web interface dynamically enables user panning and zooming as well as switching between selected composites of multi-spectral bands – i.e. other than the RGB "quicklooks" depicted in search result lists. Like *Google Maps*, the EOB also facilitates the sharing of geo-locations via deep-linked URLs. Though Sentinel-2's ground resolution is a noticeably coarser when compared to *Google Maps*' aerial photography, its acquisition cycle is far more frequent: typically, today, two or three times a week rather than once or twice a year – facilitating a more regular environmental surveillance. To that end, the EOB includes a date-picker/calendar widget to switch the display between the different archived acquisition dates, available at the displayed location – and the date selected is, furthermore, reflected in the URL, which when shared and accessed navigates directly to the intended place and date so to display the shared imagery.

As such, the EOB server acts as a de-facto multi-temporal, multi-scale visual cache for the contents of the ESA's *Sci-Hub* archive – mediated to all without registration – whose distribution online, on-demand is available for those regional-enablers who aim to foster interest in Sentinel-2 within their geographical region. As will be described below, this can be done by externalising the calendar of acquisitions, available for that region, in a more familiar and accessible form – such as a shared embedded Google Calendar. Equipped with such a calendar, individuals are then better placed to attempt their own summary visual assessments of the potential for Sentinel-2 data to fulfil their particular requirements – and may do so before investing resources towards a principled scientific analysis.

Besides provisioning a geographically-specific calendar, a regional research centre can further foster Sentinel-2 uptake by indicating streamlined effective low-cost ways for acquiring and processing the relevant archived data – which necessarily involves downloading spectral-band data expressed as JP2000 images that typically encode 12 bits per pixel and then extracting from them irregular polygonal areas as pixel values for numerical analysis. Below we look, in turn, at both the provision of a regional calendar and the pathways individuals may adopt towards affordable scientific analyses.

2 Regional Calendar

The calendar repeat pattern of Sentinel-2 acquisition for a given geo-location is determined by the initial dates on which each satellite (`S2A` and `S2B`) first made an acquisition there – and can be generated simply by adding 10-days iteratively to those start dates, on into the future. Sun-synchronous orbits mean that on those dates acquisition always occurs around noon local time. With both satellites operational, any given location is guaranteed to be visited twice during a 10 day period – since the two satellites follow the same orbital tracks, five days after each other. These tracks, known as orbital swaths, allow a 290km wide strip of images to be acquired as they are traversed. Locations that lie within the overlap of two adjacent swaths are fortunate since they accrue twice the number of calendar acquisition dates. Sardinia, for example, is intersected by two swaths that overlap over a wide area of the north and east of the island, as can be appreciated from figure 1 below:

The city of Oristano (standing in tile `32TMK`) lies under both swaths and so is visited on twice the number of calendar dates with respect to Cagliari (the island's capital sited in tile `32SNJ`) which lies under just one swath, `022`. To compile a list of dates for a shared web calendar it is necessary to determine which swaths intersect the user's location. That information is inherent in the internal calendar of the EOB and can be directly derived for our external calendar by making a location-based query to ESA's Sci-Hub as detailed later. There are three general classes of location: those (a) intersected by just 1 swath (b) intersected by 2 swaths and (c) at the boundary of a swath – where only orbital vagaries determine if a second swath intersects on a particular date.

Given the UTM tile of the chosen location it is possible to obtain its list of dates using ESA's Sci-Hub through a sequence of queries executed on their web-based API. This is a lengthy technical operation that incurs pre-registration, authentication and occasional failure-recovery procedures. As such it is a task best done by a regional research centre for it's citizens. The resultant list contains an estimate of the percentage cloud cover per tile per date. It can be published online represented as a Javascript object resource which may then be seamlessly imported as a date-indexed array within a calendar on the web, and then regularly updated to reflect new acquisitions.

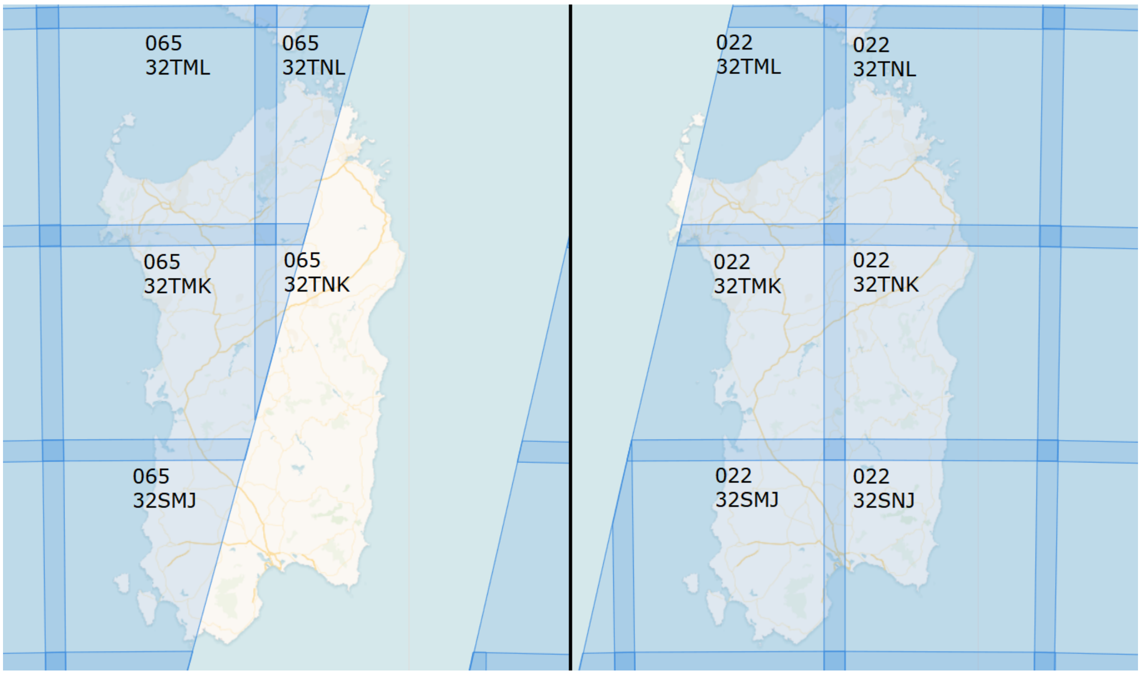


Figure 1: Two swaths intersect Sardinia. Shown in veiled blue, they are labelled using their relative orbit number: on the left `065` and on the right `022`. Sentinel-2's 100km^2 tile divisions are indicated using the UTM labels: `32TML` etc. The two swaths spatially overlap and neither covers the entire island.

2.1 Basic HTML Javascript month-view calendar

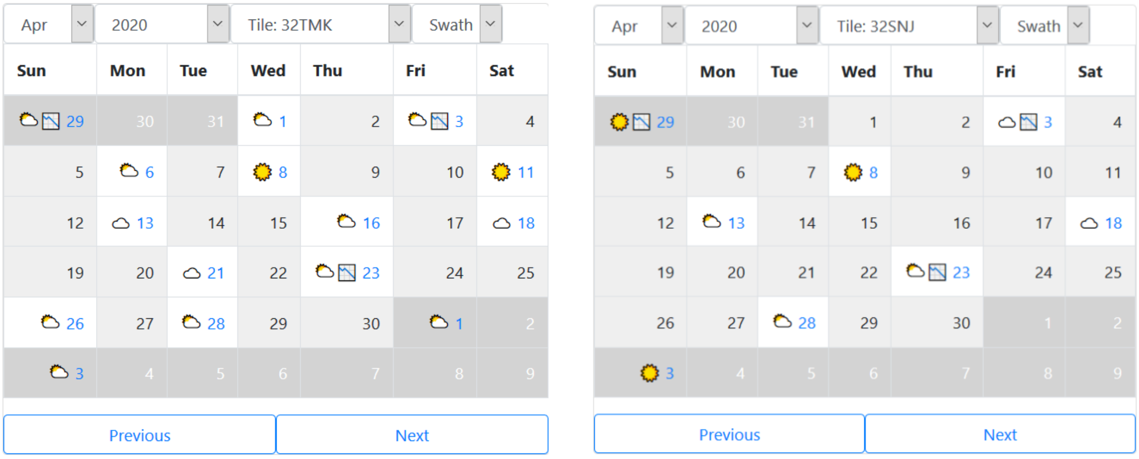
Below we implement a basic month-view type calendar using pure HTML and Javascript. Occupying more screen-space than a traditional date-picker, it permits users to rapidly navigate between current and past months to gain a quick overall impression of the temporal pattern and density of acquisition for their custom location. Users select the month the tile and swath (either one or both) and then any day-cell that matches an acquisitions gets highlighted in white rather than grey. Each such cell also gets scripted into a hypertext link – formulated as an *url* that encodes the calendar-date, a custom location's latitude, longitude as well as some other fixed parameters (e.g. zoom-factor) required to visualise the EOB. A typical *url* looks like this:

```
https://apps.sentinel-hub.com/eo-browser/?lat=40.66729&lng=8.35892&zoom=15&time
    ↪ =2019-10-09&preset=1_TRUE_COLOR&datasource=Sentinel-2%20L1C}
```

Once clicked it can take several seconds for the visualisation to appear – which is the same within the EOB itself: we thus contextualise the user's navigational choice by adding to the date-cell a symbol indicating high, medium or low cloud cover, based on the average cloud percentage reported for each acquisition: as illustrated in figure 2.1 below.

2.2 Web Calendars

Web-calendars – e.g. Google-Calendar and Microsoft-Outlook – are popular, ubiquitous, and have the benefit that they allow events to be easily shared and accessed from smartphone and PC's alike – at zero cost to users across the world in various internationalised forms. They allow instant switching between month-view, week-view, or an agenda list of events. Their more sophisticated purpose incurs a much more elaborate design compared to our HTML JavaScript month-view, and it is no longer feasible to import a custom Javascript Object directly into a Web Calendar. Instead our data must reformatted into the standard iCal or *de facto* CSV encoding and then imported using the methods sanctioned by the Calendar provider: e.g. manually via a web-form or programmatically through a scripting API served by the provider. A standard *one-day event* is



Apr | 2020 | Tile: 32TMK | Swath

Sun	Mon	Tue	Wed	Thu	Fri	Sat
⛅ ▨ 29	30	31	⛅ 1	2	⛅ ▨ 3	4
5	⛅ 6	7	☀ 8	9	10	☀ 11
12	☁ 13	14	15	⛅ 16	17	☁ 18
19	20	☁ 21	22	⛅ ▨ 23	24	25
⛅ 26	27	⛅ 28	29	30	⛅ 1	2
⛅ 3	4	5	6	7	8	9

Previous | Next

Apr | 2020 | Tile: 32SNJ | Swath

Sun	Mon	Tue	Wed	Thu	Fri	Sat
☀ ▨ 29	30	31	1	2	☁ ▨ 3	4
5	6	7	☀ 8	9	10	11
12	⛅ 13	14	15	16	17	☁ 18
19	20	21	22	⛅ ▨ 23	24	25
26	27	⛅ 28	29	30	1	2
☀ 3	4	5	6	7	8	9

Previous | Next

Figure 2: A basic month-view calendar indicating local Sentinel-2 acquisition dates (blue on white) for April 2020 for two different UTM tiles - on the **left** tile `32TMK` that covers Oristano, and on the **right** tile `32SNJ` that covers Cagliari. The sun/cloud symbols provide a rapid impression of clouds conditions. The other symbol indicates that an acquisition split into two products – as explained later. Demo here: `https://quicklook-606e5.firebaseapp.com/BasicCal/`

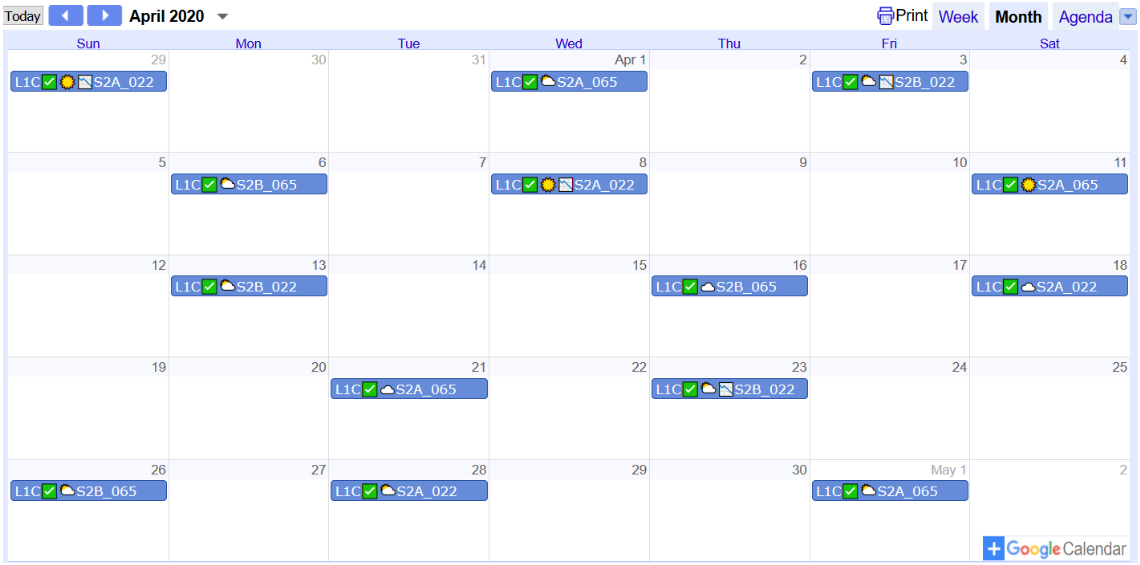
the natural way to represent a single Sentinel-2 acquisition on a web-calendar – acquisition being limited to once a day around the same time. Such an event can be minimally specified by its *date*, a plain-text *title* and *description* which may be in a rich-text format, or not. For example, the above *url* might be encoded as an HTML link in an event description, using iCal, thus:

```
DESCRIPTION:<a href="https://apps.sentinel-hub.com/eo-browser/?lat=40.66729&lng
    ↪ =8.35892&zoom=15&time=2019-10-16&preset=1_TRUE_COLOR&datasource=Sentinel
    ↪ -2%20L1C" target="_blank">EOB link</a>
```

Note however, event-descriptions remain invisible until the user clicks upon the event's title within the calendar grid: click-1 opens the event-panel – often as a pop-up; click-2 in that panel, on a link triggers a confirmation dialog – e.g. warning "you are leaving Google"; until finally click-3 navigates to the destination. Such disrupted user's experience can be cured by overriding the browsers normal behaviour for the specific calendar site using web-extension/add-ons. Click-2, for example, is obviated by the web extension *ClearURLs* – recommended by Firefox. The panel/pop-up can similarly be scripted-away by the web-extension described next. Later we show how the panel can, instead, be populated with the tiles' thumbnail images for the region on the given date, in addition to the chosen link to the EOB – which results in a synoptic regional calendar.

2.3 Web-calendar with titles as direct-links

Looking to gain the benefits that a web-calendar can provide, we first emulate the behaviour of our basic HTML/Javascript calendar using a web extension acting upon *Google Calendar* – see figure 3 above. This is achieved through a script that recognises the particular text forms of our event *title*'s in order wrap each of them in a direct-link to the EOB portal – for the particular date, at a custom location. Technically this should involve overriding the *onclick* handler for each matched event *title* so it no longer creates a panel but instead opens a new tab with the EOB *url*. However, this aim is disrupted by *Google*'s adoption of anonymous rather than named handler functions in their Javascript implementation. Which means we can no longer simply remove the existing handlers to replace them with our own. Instead, we must replace the entire HTML element to which each handler is attached and script our own links in order to get our desired *onclick* behaviour. This is not hard to achieve but incurs a cost: the web extension now has to request a special permission termed *innerHTML*. This, in turn, means it will not be accepted at the Chrome or



Sun	Mon	Tue	Wed	Thu	Fri	Sat
29 L1C✅☀📉S2A_022	30	31	Apr 1 L1C✅⛅S2A_065	2	3 L1C✅⛅📉S2B_022	4
5	6 L1C✅⛅S2B_065	7	8 L1C✅☀📉S2A_022	9	10	11 L1C✅☀S2A_065
12	13 L1C✅⛅S2B_022	14	15	16 L1C✅☁S2B_065	17	18 L1C✅☁S2A_022
19	20	21 L1C✅☁S2A_065	22	23 L1C✅⛅📉S2B_022	24	25
26 L1C✅⛅S2B_065	27	28 L1C✅⛅S2A_022	29	30	May 1 L1C✅⛅S2A_065	2

Figure 3: *Google Calendar* in embedded form with events imported from a CSV calendar file. Our web extension transforms each event-title into a direct-link to the EOB, which opens in a separate tab (not shown). Titles indicate the Sentinel-2 product type (L1C), the satellite id (S2A or S2B) and the swath (022 or 065).

Firefox hosted Add-on Stores – so its eventual (private) distribution and installation is no longer transparent nor comfortable for users. This distribution significantly counteracts those advantages that are being gained by harnessing *Google Calendar*. Indeed, our basic HTML/Javascript calendar remains an attractive alternative – while avoiding the onus of installing an add-on or the repetitive administrative task of importing files into *Google Calendar*. Those tasks become better motivated as we consider the synoptic regional, next.

2.4 Synoptic regional Calendar

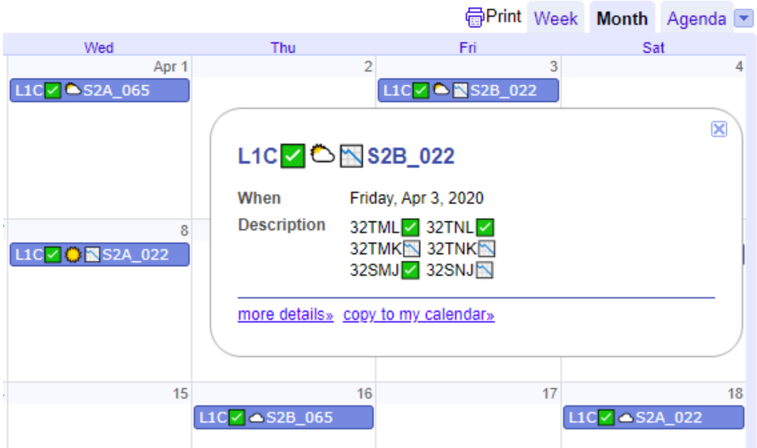
Here we examine the utility of *Google Calendar* for displaying event *descriptions* in panels in order to visually express the availability, layout and cloud condition of Sentinel-2 product across geographically compact region – in particular Sardinia and its six UTM tiles. If we limit the *description* to a purely textual format no web extension is needed – e.g: using a CSV format as below.

```
Subject,Start Date,All Day Event,Description
L1C✅⛅📉S2B_022,04/03/2020,TRUE,"32TML✅ 32TNL✅
32TMK📉 32TNK📉
32SMJ✅ 32SNJ📉"
```

From this single one-day event – using literal new-line characters inserted into the CSV *description* – we can obtain a basic 2x3 grid of tile names with Unicode symbols as rendered in the figure 4 below:

This constitutes a useful fallback service for lower-tech calendar clients or for low-bandwidth conditions. Technically, each tile name could also be wrapped within its own an EOB link to further emulate to the functionality of our basic calendar. This would entail adopting the more complex iCal format, where the *description* field can contain HTML hypertext links. On import, however, *Google* modifies all such links to redirect them (again) through the disruptive confirmatory dialogue – that only a web-extension might script away.

For our second, more graphical, web extension we stay with the less complex CSV format and



L1C✓ ▨S2B_022

When Friday, Apr 3, 2020

Description 32TML✓ 32TNL✓
32TMK▨ 32TNK▨
32SMJ✓ 32SNJ▨

Figure 4: *Google Calendar* pop-up panel with a purely textual event description.

choose, instead, to substitute each tile name with an HTML image element – scripted to have a *src* attribute corresponding to a preview thumbnail image of the relevant Sentinel-2 product. The result is a mosaic-like graphical presentation as illustrate in the figure 5 below:

Here a web extension is essential to insert the images (requiring *innerHTML* permission again) since web-calendars, in general, deliberately prevent the rendering of any image included in an event *description*. *Google Calendar* appends images as symbolic attachments that can be viewed individually once clicked upon. So instead of fruitlessly attempting to import *descriptions* containing HTML image elements we use our web extension to: (a) monitor whenever a new panel appears in *Google Calendar*, (b) check if the title fits our pattern, and if it does then (c) dynamically edit the existing text *description* so each tile-name becomes wrapped in the appropriate image element – with its *src* attribute address pointing to a preview image in one or others of the available online repositories.

2.5 Sentinel-2 preview-image online repositories

The online, publicly available sources of Sentinel-2 preview images differ significantly in their ease-of-access/web-readiness. In order of convenience they are:

2.5.1 roda.sentinel-hub.com

The providers of the EOB also serve a comprehensive set of Sentinel-2, 343x343 pixel, RGB, preview images as a public service. For example, the *url*
`https://roda.sentinel-hub.com/sentinel-s2-l1c/tiles/32/T/MK/2020/4/8/0/preview.jpg`
directly links to the preview of the tile `32TMK` on the 8th April 2020. Since such an *url* is composable without knowing any other product parameters and since it links to a JPEG image, that image is directly available on any external web page (including our panel) – simply by assigning its *url* to the *src* attribute of an HTML IMG element.

2.5.2 scihub.copernicus.eu

Although preview images can be viewed from within the Copernicus *Sci-Hub* portal in response to a product search, it is not possible to directly link to those images from other web pages. This is a shame, since these images are larger (512x512 pixels) and are of preferred provenance: ESA's official open-access repository. ESA undermines direct linking by:

- serving the JPEGs using a HTTP Content-Disposition response header – that enforces download rather than in-browser display;

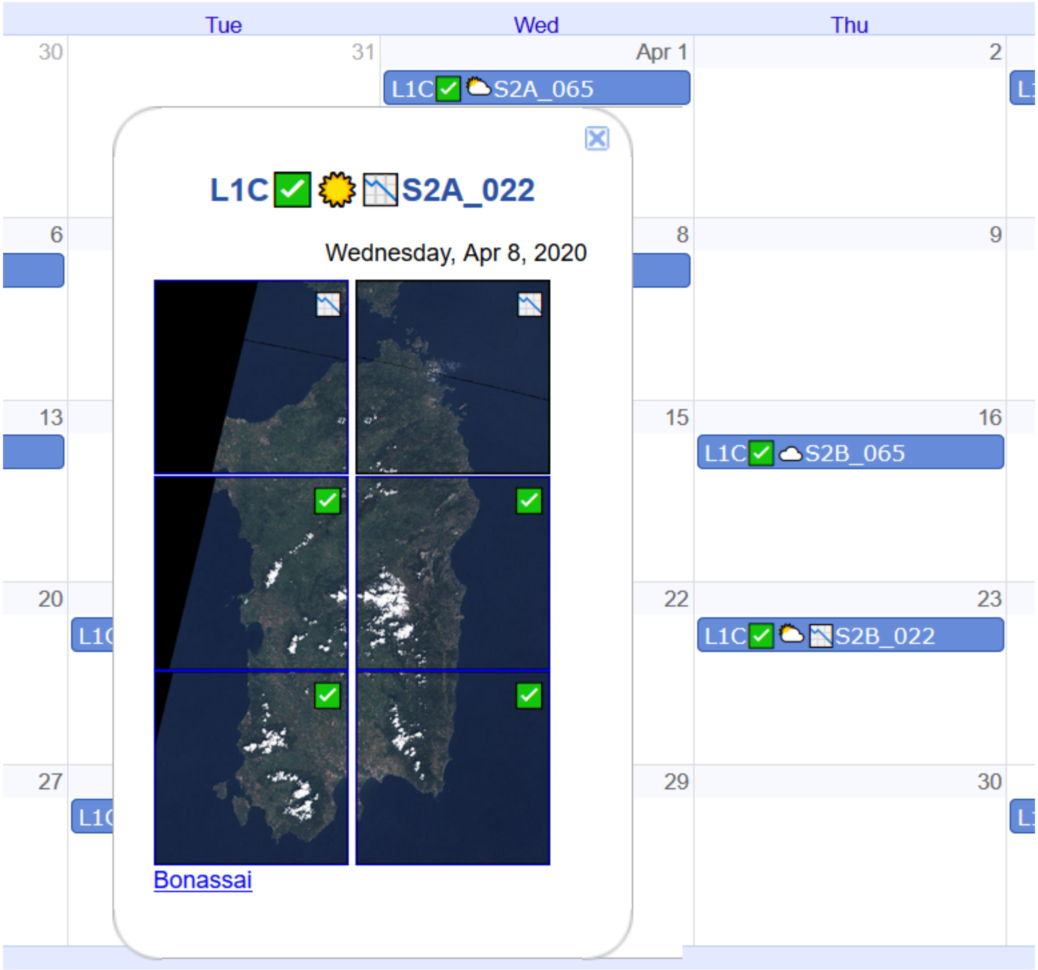


Figure 5: A mosaic-like representation in a *Google Calendar* pop-up panel achieved using our web extension to substitute names with preview images – on a day when two tile-splits occur. Clicking on a image brings up a larger thumbnail in separate tab. Clicking the lower link opens the EOB at that custom Sardinian location, in another tab. Demo: https://quicklook-606e5.firebaseapp.com/

- by incorporating a random-like *UUID* string in the path of each *url* – which cannot be known without prior querying of their catalogue;
- requiring access via a login session which regularly times-out.

As regional-enabler we might, however, download the set of previews for our region and re-present them in our own mirror repository that preserves the content whilst obviating the need for any *UUID* strings. We could do this by regularly running a batch-script that adopts ESA's *OData* API (see: https://scihub.copernicus.eu/userguide/ODataAPI).

For example, given product *UUID*, say `df7a633c-ff52-4414-86df-fd48eba85846`, found in the catalogue (as detailed later), we can use `wget` to download the preview associated that product as follows:

```
wget -q --content-disposition --continue --user=myLogin --password=myPassword -P
    ↪ myDirectory "https://scihub.copernicus.eu/dhus/odata/v1/Products('df7a633c
    ↪ -ff52-4414-86df-fd48eba85846')/Products('Quicklook')/\$value"
```

where, the content-disposition response header is used by the receiving software to assign the image received its official ESA name:

`S2A_MSIL1C_20200408T101021_N0209_R022_T32TMK_20200408T171107-ql.jpg` whose naming convention (see p.57 of https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook) usefully relays the following parameters:

- tile-name: 32TMK, acquisition date: 20200408, satellite id: S2A, swath id: 022;
- processing baseline N0209, processing timestamp: 20200408T171107.

The former set of which can help when populating our calendar – and the latter in disambiguating dates upon which split or "duplicate" tiles occur – as detailed later.

2.5.3 peps.cnes.fr

The French space agency *CNES* helpfully provides a mirror of Sentinel-2 products from the *SciHub* – whose coverage extends beyond France. Indeed, its *PEPS* Hub serves the same JPEG preview images of Sardinia as the *SciHub* but without the restrictions listed above. Nevertheless, in order to compose each of its *urls* – e.g.
https://peps.cnes.fr/quicklook/2020/04/08/S2A/S2A_MSIL1C_20200408T101021_N0209_R022_T32TMK_20200408T171107_quicklook.jpg
it is necessary to know ESA name for each preview image – and thus pre-compile the catalogue beforehand.

2.5.4 storage.googleapis.com/gcp-public-data-sentinel-2

Google Cloud provides its own comprehensive mirror of Sentinel-2 products and exposes their component parts along structured paths via externally accessible *urls*, all within the browser while logged into a Google Account – which at first glance seems to be useful for obtaining spectral bands on demand. Below is a typical *url* that serves a preview image:
https://storage.cloud.google.com/gcp-public-data-sentinel-2/tiles/32/T/MK/S2A_MSIL1C_20200408T101021_N0209_R022_T32TMK_20200408T171107.SAFE/GRANULE/L1C_T32TMK_A025044_20200408T101923/QI_DATA/T32TMK_20200408T101021_PVI.jp2
There are two problems here:

- the image is formatted as JPEG2000 rather than standard JPEG – which makes it manifestly difficult to render in any browser (except Safari) since no free, stable, cross-browser plugins have been developed, even 20 years after the format's debut;
- the *url* expresses the entire product path, rather than just the ESA filename: i.e. it contains hard-to-discover parameters such as A025044. Those parameters are best obtained from the product's metadata file MTD_MSI1C.xml – though that needs to be downloaded and parsed ahead of time.

Thus, to make use of the JPEG2000 image from Google they would first have to be downloaded and then locally converted into web-ready JPEG. The conversion could be achieved via the public domain gdal_translate utility – see https://gdal.org/gdal_translate.html. Once done, a mirror might then be hosted in which each image is renamed in a way that flattens the path and obviates the need to know of the hard-to-discover parameters.

2.5.5 Split Tiles

A discontinuity in a Sentinel-2 satellite's "data-take" can occur mid-tile during an acquisition – see https://forum.step.esa.int/t/sentinel-2-weeding-out-bad-products/15041/14. ESA then splits that tile into two distinct products – and gives each a filename that may differ only in its processing timestamp parameter. This immediately becomes apparent when we adopt the simple strategy of loading our calendar panel with tile-images from **roda.sentinel-hub.com** – since

only one of the two images gets rendered: leaving a blank corner as shown (twice) in the figure 6 (left) below. The missing fragment can sometimes be retrieved by substituting the final zero in the *url* with a one – e.g.:
https://roda.sentinel-hub.com/sentinel-s2-l1c/tiles/32/T/MK/2020/4/8/1/preview.jpg
Yet this is not particularly useful since:

- we cannot complete our rendering by superimposing the JPEG images on top of each other on the web page – since that requires an image format with an alpha/transparency layer (e.g. PNG);
- the roda.sentinel-hub.com nomenclature is not so straightforward. The *url* above might also refer to a "duplicate" product generated when ESA occasionally updates the `processing baseline` (i.e. system version).

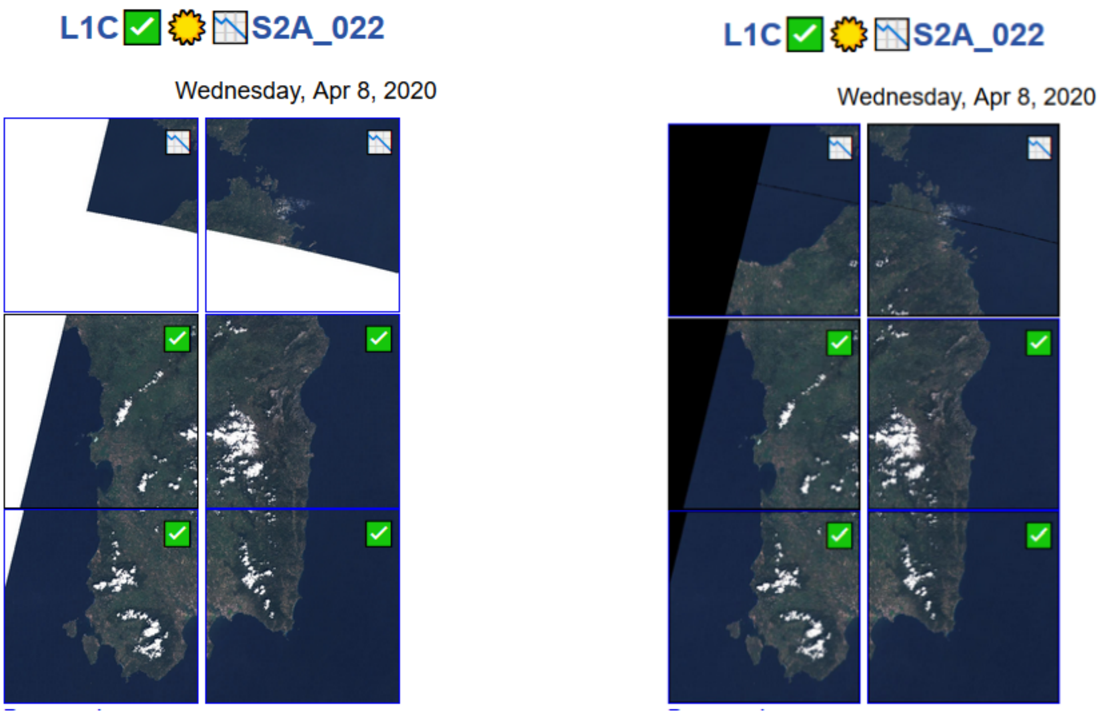


Figure 6: (left) Split tiles show up as preview images with corners missing unless provision is made – images are from roda.sentinel-hub.com, and so have white backgrounds. (right) Result of creating composite previews and serving them from our dedicated server – images are sourced from SciHub, and so have black backgrounds.

Ideally we should detect split-tiles without resorting to visual analysis the image. We achieve this by maintaining a catalogue of the tiles available for our region and analysing the constituent filenames – to systematically distinguish variations in `processing timestamp` and `processing baseline`, as described in a later section. This allows us to download the two JPGs for each split tile (from the SciHub) and then transform them into a composite image using the open-source *ImageMagick* utility `convert` as follows:

```
/usr/bin/convert preview1.jpg -fuzz 2% -transparent black preview1.png
/usr/bin/convert preview2.jpg -fuzz 2% -transparent black preview2.png
/usr/bin/convert preview1.png preview2.png -composite composite.jpg"
```

which leaves a small black line on the composite image as a deliberate reminder of the split nature of that tile – e.g. see figure 6 (right).
If the two intermediate PNG previews were added to the mirror, then it would be possible to achieve the same effect by mixing them live on the web page as follows:

```
<div style="position:relative;text-align:right;"><img src="preview1.png" width="
    ↪ 512" height="512" border="1"><img src="preview2.png" style="position:
    ↪ absolute;top:-512px;left:0px;"width="512" height="512" border="1"></div>
```

In practice, we create each composite offline using *ImageMagick* so not to add client-side complexity of our web extension code.

2.5.6 Timeliness and monitoring acquisitions

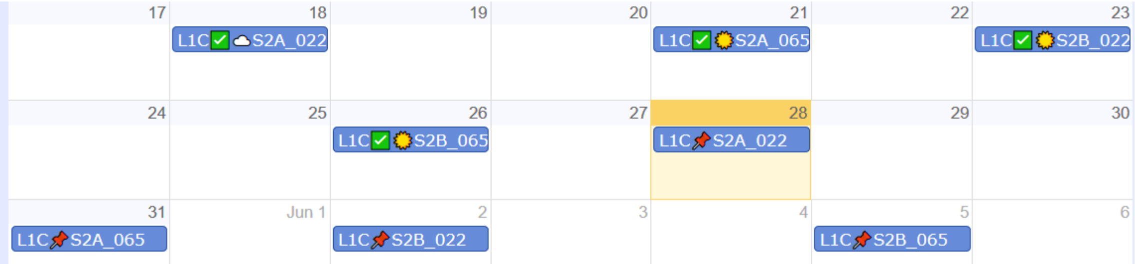
Timeliness is a defining feature of any calendar. In particular, promptly informing users of a cancelled event frees them up to carry out contingency plans. Indeed, as soon as an agricultural researcher discovers that yesterday's acquisition was cloudy or operationally cancelled they may re-plan today's crop-cutting and move it to a later date – so to optimise its transition contrast in the acquired MSI time-series.

Unrealised and upcoming acquisitions are expected to follow to the 10-day repeat patterns discussed earlier. Sardinia, for example, has four distinct interlacing patterns: that can be specified as recurring events beginning:

- 04-Jul-2015: satellite `S2A`, swath `022`
- 06-Aug-2015: satellite `S2A`, swath `065`
- 01-Jul-2017: satellite `S2B`, swath `065`
- 08-Jul-2017: satellite `S2B`, swath `022`

We can import such recurring events into a web-calendar by supplying, for each, the additional parameters of (a) start-date, and (b) repeat-interval (e.g. 10 days). Thus, four sequences of one-day-event running into the future can be automatically added. All events in a given sequence, though distinguished by *date*, necessarily have the same *title* and *description*:

- our *title* includes the satellite-id and swath-id plus a pin-push icon to show the event's unrealised status – as illustrated below;
- our *description* remains, as before, the tile-grid (see figure 4).



17	18 L1C ✓ ☁ S2A_022	19	20	21 L1C ✓ ☀ S2A_065	22	23 L1C ✓ ☀ S2B_022
24	25	26 L1C ✓ ☀ S2B_065	27	28 L1C 📌 S2A_022	29	30
31 L1C 📌 S2A_065	Jun 1	2 L1C 📌 S2B_022	3	4	5 L1C 📌 S2B_065	6

Figure 7: Calendar (extract) showing unrealised events with red pin-push icons. For the current date 28 May 2020 the acquisition may have been made but still await availability.

Thus, for each unrealised event our web extension is able to adopt the `roda.sentinel-hub.com` url scheme and so render the pop-up panel's preview images as soon as they become available – until then showing a blank. Users monitoring the transition from blank to filled images may immediately judge that an acquisition has been made and surmise if their location is cloud-free. This also constitutes a signal to update our catalogue so to reflect the now realised status of the acquisition event: which, in turn furnishes the parameters necessary to use the better quality previews from `peps.cnes.fr` – only falling back to the `roda.sentinel-hub.com` previews when necessary. In our web extension we do this using the HTML IMG `onerror` handler, e.g. as follows:

```
<img src="http://peps.cnes.fr/quicklook/2020/05/21/S2A/
    ↪ S2A_MSIL1C_20200521T102031_N0209_R065_T32TML_20200521T122533_quicklook.jpg
    ↪ " onerror="this.onerror=null;this.src='https://roda.sentinel-hub.com/
    ↪ sentinel-s2-l1c/tiles/32/T/ML/2020/5/21/0/preview.jpg'" width="96" height=
    ↪ "96" border="1">
```

except when a tile has been split and we must serve the composite image from our own server.

3 Maintaining a regional catalogue

The ESA *SciHub* releases Sentinel-2 data-product usually within a day of realising an acquisition – doing so in two stages corresponding to two product levels.

1. **Level L1C** - in which each image pixel is calibrated to be proportional to amount of radiation received by its spectral sensor This is termed *Top of Atmosphere* (TOA) radiance — the spectral "brightness" seen from the satellite. This L1C product arrives first so its preview image serves as our timely confirmation of successful acquisition – and thus we prefix our calendar event *title* with the label `L1C`.

2. **Level L2A** (arriving typically after six hours or so after L1C) - in which ESA has corrected pixels to approximate, as far as possible, *Spectral Reflectance* on the ground – i.e. for land not hidden by cloud. As such L2A spectral-band products are to be preferred over the L1C for any numerical land use analysis. Their availability can be conveniently monitored from the calendar by (a) clicking the EOB link and (b) when the EO Browser appears, observing a green rather than white `Show L2A` button. Note: roda.sentinel-hub.com does not separately serve previews of L2A products. At preview resolution, however, L2A have a very similar appearance to L1C so we stick with L1C in our calendar panel mosaics.

3.1 Level L1C catalogue

The L1C metadata required to compile the catalogue for our regional calendar can be obtained from the *SciHub* by running an automated batch script – following the SciHub *OData API* (see https://scihub.copernicus.eu/twiki/do/view/SciHubUserGuide/BatchScripting#Query_via_cURL). Our script submits a query for each realised acquisition date. The query requests the metadata concerning the six tiles of our tile-set on that date – as indicated in the example below:

```
DATE="2020-04-08"
TILES_Q="(32TML␣OR␣32TNL␣OR␣32TMK␣OR␣32TNK␣OR␣32SMJ␣OR␣32SNJ)"

query="q=${TILES_Q}␣AND␣(␣beginPosition:[${DATE}T00:00:00.000Z␣TO␣${DATE}T23
    ↪ :59:59.999Z]␣AND␣endPosition:[${DATE}T00:00:00.000Z␣TO␣${DATE}T23
    ↪ :59:59.999Z]␣)␣AND␣(␣(platformname:Sentinel-2␣AND␣producttype:S2MSI1C)␣)"

curl -s -u user:passwd --data-urlencode "rows=20" -o 2020-04-08.xml "$query"
    ↪ https://scihub.copernicus.eu/dhus/search
```

The query returns an XML file (here `2020-04-08.xml`) containing a large set of parameters for each tile found in the *SciHub* repository – including those essential for our calendar. In this way we store a local copy of the XML files for each realised acquisition date.

The script then creates a catalogue text file by extracting from the accumulated cache of files the parameters of interest for each combination of date and tile: i.e. (a) product-id, *UUID* and cloud percentage. For each date-tile combination we write out those parameters, space-separated, on a single line of text – as illustrated in the five line extract below:

```
S2A_MSIL1C_20200401T102021_N0209_R065_T32TNK_20200401T123732 a6867175-2434-4de9-
    ↪ a9cb-823c635764a2 1
S2A_MSIL1C_20200401T102021_N0209_R065_T32TNL_20200401T123732 d6d3f1a9-748e-47f5-9
    ↪ ebb-d483f394b643 0
S2B_MSIL1C_20200403T100549_N0209_R022_T32SMJ_20200403T135413 99524039-fb1c-46d3
    ↪ -8710-69d50dfe80ed 48
S2B_MSIL1C_20200403T100549_N0209_R022_T32SNJ_20200403T130157 c24e5abd-d0a4-408e-
    ↪ a8f0-5c1edbb02c51 97
S2B_MSIL1C_20200403T100549_N0209_R022_T32SNJ_20200403T135413 f701410a-b793-4dc2-
    ↪ aee4-ced255628625 70
```

We also filter out rare duplicate products – i.e. those that differ in processing *baseline* (e.g. `N0209`), but otherwise share attributes, by removing the line containing the higher *baseline*.

3.1.1 Basic Calendar

Since the product-id embeds both the date (e.g. `20200401`) and the tile-id (e.g. `32TNK`) it is a simple matter to create the Javascript file required by our Basic Calendar – for each line in the catalogue text file we create a *key/value* pair where the *key* is a concatenation of the date and tile-id and where the *value* is a concatenation of the level (`L1C`), the satellite-id (e.g. `S2A`), swath (e.g. `065`) and the cloud-percentage plus a split-flag – with concatenation done via an underscore separator. The above extract is thus transformed as follows.

```
{ ...
"20200401_32TNK" : "L1C_S2A_065_1_1",
"20200401_32TNL" : "L1C_S2A_065_0_1",
"20200403_32SMJ" : "L1C_S2B_022_48_1",
"20200403_32SNJ" : "L1C_S2B_022_83_2",
... }
```

Note, the last two lines have been unified into a single line – since they generate the same *key*, which identifies them as a split tile. Its *value* thus has the split-flag set to `2` rather than `1`, and the cloud percent is averaged. Now when the calendar requires the event *title* for a given date/tile-id combination we construct the *key*, look up the associated *value* and then compose the *title* – converting into icons the cloud-percent and split-flag.

3.1.2 Synoptic Regional Calendar

For our Synoptic Regional Calendar a more complex transformation is required: one that preserves product-id so it will be available when composing *url*s for the preview images. Here the product-id, excised of any tile-id is used as the *key* – e.g.

```
S2A_MSIL1C_20200401T102021_N0209_R065__20200401T123732
```

Here, we unify those lines, from the catalogue text file, that share this product *key* and so capture per line a list of tiles, in our region, for each acquisition date. The resulting lines are written to a new file that is (in the absence of splits) formatted, space-separated, as follows: date, cloud-percentage (averaged), product *key*, and the tile-list – e.g.

```
20200401 18 S2A_MSIL1C_20200401T102021_N0209_R065__20200401T123732 32SMJ 32TMK 32
    ↪ TML 32TNK 32TNL
```

which is then converted into Javascript key/value pairs – where now the key is the date, and value is the rest of the line, e.g.

```
{ ...
"20200401" : "S2A_MSIL1C_20200401T102021_N0209_R065_20200401T123732␣32SMJ␣32TMK␣
    ↪ 32TML␣32TNK␣32TNL",
... }
```

Once sent to the web extension each tile-id in the list can, in turn, be re-inserted into the *key* so to generate each tile's original product-id and the `peps.cnes.fr` *url* necessary to render its preview image.

In the case of split tiles we serve the composite previews from our own server – so the scheme above become slightly more complex. The presence of a split is indicated when two distinct product-ids share the same stem of leading characters (up to the tile-id) – e.g. products from the last two lines in the catalogue listing above:

```
S2B_MSIL1C_20200403T100549_N0209_R022_T32SNJ_20200403T130157
S2B_MSIL1C_20200403T100549_N0209_R022_T32SNJ_20200403T135413
```

which share the stem

```
S2B_MSIL1C_20200403T100549_N0209_R022
```

Here we unify lines in the output file that share the key-stems – e.g. as follows:

```
20200403 71 S2B_MSIL1C_20200403T100549_N0209_R022_20200403T130157 32SNJ 32TMK 32
    ↪ TML 32TNK 32TNL|S2B_MSIL1C_20200403T100549_N0209_R022_20200403T135413 32
    ↪ SMJ 32SNJ 32TMK 32TNK
```

where, following a '|' divider, the second product has been appended along with its own list of tiles. The new file may then be transformed into Javascript again using a date key but with the value now extended – e.g.

```
{ ...
"20200403" : "S2B_MSIL1C_20200403T100549_N0209_R022_20200403T130157␣32SNJ␣32TMK␣
    ↪ 32TML␣32TNK␣32TNL|S2B_MSIL1C_20200403T100549_N0209_R022_20200403T135413␣32
    ↪ SMJ␣32SNJ␣32TMK␣32TNK",
... }
```

Furthermore, the new file can also be transformed into the CSV calendar representation of all-day events – detailed earlier – and within each such event its tiles can be now annotated with the appropriate icons to show presence, absence or split.

3.2 Level L2A catalogue

The above procedure also serves to produce a catalogue of L2A prodcuts with two minor adjustments:

- specifying `producttype:S2MSI2A`;
- specify our region using a "polygon footprint" rather than a neat list of tiles – to mitigate a server bug[1].

Thus, for L2A the earlier query becomes:

[1] https://forum.step.esa.int/t/l2a-products-shown-in-eo-browser-but-not-found-on-official-hub/23362/3

```
DATE="2020-04-08"
FOOTPRINT="footprint:\"Intersects(POLYGON(8.02 39.00,10.12 39.00,10.12 41.00,8.02
    ↪  41.00,8.02 39.00))\""
query="q=${FOOTPRINT}␣AND␣(␣beginPosition:[${DATE}T00:00:00.000Z␣TO␣${DATE}T23
    ↪ :59:59.999Z]␣AND␣endPosition:[${DATE}T00:00:00.000Z␣TO␣${DATE}T23
    ↪ :59:59.999Z]␣)␣AND␣(␣(platformname:Sentinel-2␣AND␣producttype:S2MSI2A)␣)"
```

To reduce the risk of obtaining partial results, we run the L2A queries one day or more after acquisition. Note, the L2A back-catalogue starts in March 2018 – much later than L1C.

The L2A catalogue is to be used for a different purpose – the composition of *urls* for the download of the (much larger) L2A spectral reflectance band images. For ease of access, we adopt those band images available at the *Google Cloud* mirror, rather than the official *SciHub*. This minor compromise in terms of data provenance is necessitated by the already slow download rate obtained at the *SciHub* – that is now much worsened for many products, with the advent of ESA's *Long Term Archive* policy.

Each *url* will be specific to (a) acquisition date, (b) tile-id, and (c) spectral band-id. The former two are used to query our L2A catalogue file and so extract the corresponding L2A product-id. That product-id is then be examined to obtain the additional parameters required to formulate to structured path of the band-image on *Google Cloud*. This is achieved by first downloading the corresponding metadata file called `MTD_MSIL2A.xml` – e.g. for tile `32TML` on `20200111`
https://storage.cloud.google.com/gcp-public-data-sentinel-2/L2/tiles/32/T/ML/S2B_MSIL2A_20200114T101259_N0213_R022_T32TML_20200114T125034.SAFE/MTD_MSIL2A.xml
That `MTD_MSIL2A.xml` file contains lines wrapped in XML `IMAGE_FILE` tags which provide the additional parameters of the path required to compose the *url* for each spectral band image – e.g.

```
<IMAGE_FILE>GRANULE/L2A_T32TML_A014920_20200114T101254/IMG_DATA/R10m/
    ↪ T32TML_20200114T101259_B02_10m</IMAGE_FILE>
```

– which corresponds to the spectral band-id `B02` (blue) at 10m^2 spatial resolution: the *url* for which is thus composed as:
https://storage.cloud.google.com/gcp-public-data-sentinel-2/L2/tiles/32/T/ML/S2B_MSIL2A_20200114T101259_N0213_R022_T32TML_20200114T125034.SAFE/GRANULE/L2A_T32TML_A014920_20200114T101254/IMG_DATA/R10m/T32TML_20200114T101259_B02_10m.jp2
Not all combinations of band-id and resolution are valid. Table 1 lists those produced by ESA.

3.3 Hosting

Our design minimises the data that we need to host on our dedicated server. First, the calendar itself is hosted elsewhere by *Google Calendar*. Next, the preview images are hosted on either `roda.sentinel-hub.com` or `peps.cnes.fr` – the latter being preferred once an acquisition is in our catalogue.

We only host preview images in the rare case of split tiles where we have had to create a composite image – as described earlier. For example the the split tile `32TMK` acquired on date `20200423` is hosted at the following.
https://quicklook-606e5.firebaseapp.com/L1C/splits/S2B_MSIL1C_20200423T100549_N0209_R022_T32TMK-ql.jpg
where

- `quicklook-606e5.firebaseapp.com` is our dedicated (Firebase) server
- `/L1C/splits/` is the fixed internal server path

Band-id	Resolutions			Type	Peak	Bandwidth
B01			60m	Aerosol	443 nm	36 nm
B02	10m	20m	60m	Blue	490 nm	98 nm
B03	10m	20m	60m	Green	560 nm	45 nm
B04	10m	20m	60m	Red	665 nm	38 nm
B05		20m	60m	Visible IR	705 nm	20 nm
B06		20m	60m	Visible IR	740 nm	18 nm
B07		20m	60m	Visible IR	783 nm	28 nm
B08	10m			Panchromatic	842 nm	140 nm
B8A		20m	60m	Narrow NIR	865 nm	32 nm
B09			60m	Water vapour	945 nm	26 nm
B10			60m	SWIR Cirrus	1374 nm	30 nm
B11		20m	60m	SWIR	1610 nm	142 nm
B12		20m	60m	SWIR	2190 nm	240 nm
TCI	10m	20m	60m	True Color Image (RGB)		
AOT	10m	20m	60m	Aerosol Optical Thickness		
WVP	10m	20m	60m	Water Vapour Map		
SCL		20m	60m	Scene Classification		

Table 1: Permitted combination of L2A parameters for formulating download *urls* for *Google Cloud*: Given the Band-Id (column 1) any of the listed Resolutions (column 2) may be selected.

- `S2B_MSIL1C_20200423T100549_N0209_R022_T32TMK` is the product stem name common to the two split acquisitions.

A Firebase server is well suited, at time of writing, for our purpose – providing high availability at zero cost for a small scale static site that simply mirrors a local master directory. We use ours to host the Javascript and JSON catalogue files required by (1) our Basic HTML Calendar and by (2) our web extension enhanced version. Since new items are only added every two or three days no dynamic middleware or server-side database is necessitated. For each new satellite transit we run our shell-scripts to update our master directory containing the catalogue files before issuing *Firebase*'s shell command `firebase deploy` to automatically refresh them on the web-facing server.

Furthermore, for the rare occasions when both `roda.sentinel-hub.com` and `peps.cnes.fr` are simultaneous offline we maintain copies of the latter's preview images also on the *Firebase* server - to serve in their stead.

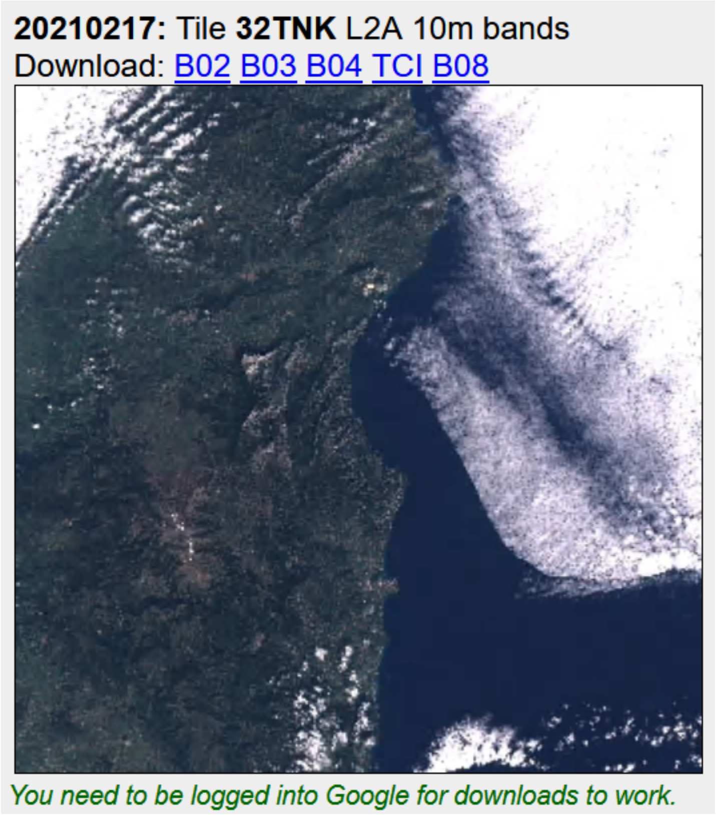
4 Assisting users to download L2A images

4.1 Calendar-assisted download

Having identified dates of interest from the calendar, users may then wish to download images for those dates and attempt their own analysis. Level-2A imagery is preferred for this purpose, since it already incorporates ESA's atmospheric correction. As it can be it can be tricky and time-consuming to track down and download such imagery we have added to our calendar, it provides the functionality to expedite the download of individual bands level-2A images directly from *Google Cloud* – without users having to separately navigate the cumbersome interface and *url*-structure involved. To this end, we provide a dedicated download page that users can access from the two different calendars:

- Basic calendar: by clicking on a download symbol next to any acquisition date;
- Web calendar: from a given date's event panel by clicking on the desired tile's preview image.

In either case, the page appears in a new browser tab. As shown in Figure 8 it is headed by the current date and tile-id and then lists, for an assortment of 10m^2 L2A bands, the direct download links to the images on *Google Cloud.* Users click on a link to initiate a download. Below those links we re-display the `TCI` preview image. The five most commonly used 10m^2 bands are listed – although, in principle, any of the band/resolution combinations permitted in Table 1 could be included. In the case of a split tile, two different sets of download links are needed – so they are placed side-by-side each above their preview image. Downloads fail if (1) the user is not already logged into any Google account, or (2) Google has yet to mirror the relevant product which may be a day or two after they appear on the data-hubs.



20210217: Tile **32TNK** L2A 10m bands
Download: B02 B03 B04 TCI B08

You need to be logged into Google for downloads to work.

Figure 8: Dedicated download page – for a given tile (`32TNK`) on a given acquisition date (`20210217`) – providing direct links (in blue) for downloading the tile's images from *Google Cloud* for a common assortment of 10m^2 L2A bands: `B02, B03, B04, TCI, B08`. Clicking on a labelled links initiates the respective image download. If there were a split-tile two such panels would be displayed side-by-side.

The download page generates, on being loaded, the relevant HTML including each download link using JavaScript that consults our L2A catalogue – with the latter compiled, as detailed earlier, and made accessible in the form of JavaScript object residing in an external file.

Note, the downloads are typically around 100 MB per band and are in JP2 image format. They can be viewed on a standard Windows PC using free standalone tools such as *Irfanview* equipped with its JPEG2000 plugin[2]. When viewing single band images the contrast may be initially low due to conversion from 12 to 16 bit pixels – but this can be easily fixed by pressing Shift U to *Auto-adjust colors.* It is also possible to pan and zoom within the images to better reveal their 10m^2 resolution features – though, compared to using the *EO Browser*, this can be slow and unwieldy. Here we are prioritising analysis over fluency of viewing.

[2] `https://www.irfanview.com/plugins.htm`

5 Postscript

Since this report was started advances in internet technology/resources have allowed the design of a much improved calendar system that avoids the need for web extension/deployment via Google Calendar. It is currently online here: https://crs4.github.io/IsolaS2/src