esac

European Space Astronomy Centre
P.O. Box 78
28691 Villanueva de la Cañada
Madrid
Spain
T +34 91 8131 100
F +34 91 8131 139
www.esa.int

DOCUMENT

Solar Orbiter SOC Engineering Guidelines for External Users

Prepared by	**Carr, Richard**
Reference	**SOL-SGS-TN-0006**
Issue	**1**
Revision	**0**
Date of Issue	**2015-03-04**
Status	**Approved/Applicable**
Document Type	**Technical Note**
Distribution	

European Space Agency
Agence spatiale européenne

APPROVAL

Title Solar Orbiter SOC Engineering Guidelines for External Users	
Issue 1	**Revision** 0
Author Carr, Richard	**Date** 2015-03-04
Approved by Jose Marcos, QA	**Date** Digitally signed by Jose Marcos DN: cn=Jose Marcos, o=ESAC, ou=TEC-QQS, email=jose.marcos@sciops.esa.int, c=ES Date: 2015.03.05 08:53:09 +01'00'
Authorised by Luis Sanchez, Development Manager	**Date** Luis Sanchez Digitally signed by Luis Sanchez DN: cn=Luis Sanchez, o=European Space Agency - ESAC, ou=SRE-ODS, email=Luis.Sanchez@esa.int, c=<n Date: 2015.03.05 15:20:55 +01'00'

CHANGE RECORD

Issue 0	**Revision** 5		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Since 0.1 – Many changes following the discussion with instruments at SOWG #5; some as agreed then and others as a more concrete SOC proposal for further discussion.	2014-09-11	Throughout.	

CHANGE RECORD

Issue 0	**Revision** 6		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Include *restricted* and *private* product option.	2014-09-25	15-16.	
Corrected *timelines* to *pipelines*.	2014-09-25	6.	

CHANGE RECORD

Issue 0	**Revision** 7		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Appliance file to be delivered '.ova' (archive) not '.ovf''	2014-12-17	11.	6.1
Specify SoloHI reference image as provided as auxiliary data to the relevant pipeline.	2014-12-17	20	6.3

SOLOHI not SoloHI for directory name.	2014-12-17	15	6.2.6
Add and revise material on logging	2014-12-18	19,21	6.2.10, 6.4.2
Repaired some cross references, minor rewording.	2014-12-18		
Clarify that specific file content not described here.	2015-01-02	6	2
Description of deployment configuration.	2015-01-02	12	6.2.1
Drop mention of iSCSI mounting.	2015-01-02	15	6.2.2
Post-processing description made more concrete.	2015-01-13	9	5.2.4
Specify post-processing output frame by reference to metadata description document.	2015-01-13	9	5.2.4
Add logs directories to figure 1.	2015-01-13	16	
Co-ordinate frame specified also for In Situ.	2015-01-13	9	5.2.4

CHANGE RECORD

Issue 1	**Revision** 0		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Revert SOLOHI to SoloHI, as per action from SOWG #6 Low Latency Splinter.	2015-02-24	13.	6.2.6
Add output directory *failed* where processing request responses are created when the appliance knows that processing has failed. Figure 1, data directory tree, updated to add *failed*.	2015-02-25	15, 16.	6.2.8. Figure 1.
Add the Dataset Description Documents as one of the specifications of the products to be created by the appliances.	2015-02-25	5,15	2, 6.2.7
Removed comment relating to DDS file naming.	2015-02-25	17	6.2.9
Removed 'private' subdirectory for products, and described SWT 16 decision that LL products shared between instruments.	2015-02-25	17	6.2.10
Clarify that Main telemetry dataset for processing is in one file and, if provided, that from a preceding period is in a separate file.	2015-03-04	14, 15	6.2.6
Clarify that only appliances which have passed acceptance tesdts will be deployed, and only at the discretion of the SOC manager.	2015-03-04	20	7.5
Telemetry retrieval by list of APIDS, not list of PIDS	2015-03-04	14	6.2.6

Table of contents:

1 INTRODUCTION

The Solar Orbiter SOC Document Tree description [SOL-SGS-TN-0001] specifies the role of the current document as describing 'the engineering procedures to be followed by external users (e.g. Instrument Teams) to develop, validate and integrate data and SW in the SOC...'.

2 SCOPE

There are no (but see below) external users other than the instrument teams who will contribute significant newly developed software elements for integration in the SOC. The software elements to be contributed by the instrument teams are the Low-Latency data pipelines.

Note that there is at least one significant bespoke software element that is likely to be used by the SOC, namely the Experiment Planning System (EPS). The associated MAPPS software may also be used. However this is not *external* in that it is developed inside ESA (TEC), nor are the developers users of the SOC. In addition, although modifications may well be made for Solar Orbiter purposes, this is not a new software product. This is excluded from the scope of this document.

As such, the central role of this document is to describe conceptually (initially) and concretely (as the document is refined) the interface between the Low-Latency data pipelines and SOC infrastructure systems, including the mechanism by which these pipelines will be delivered to the SOC and accepted. It will also provide guidance where possible on mechanisms that could aid instrument teams in meeting this interface.

The specific low latency product files to be produced by the pipelines are not described here. Specifications for the formats are given in the ICD documents, one each for Remote Sensing (FITS) and In Situ (CDF), listed as applicable in 3. The particular products to be produced by each instrument will be defined in the Dataset Description Document for Low Latency Products – one document per instrument, TBW.

3 APPLICABLE DOCUMENTS

The following documents form part of this document to the extent specified herein. They are referenced in this document in the form [AD XX]:

[AD01] SOL-SGS-ICD-0005, Solar Orbiter Interface Control Document for Low Latency Data FITS Files, TBW
[AD02] SOL-SGS-ICD-0004, Solar Orbiter Interface Control Document for Low Latency Data CDF Files, Draft

4 REFERENCE DOCUMENTS

The following documents, although not part of this document, extend and/or clarify its contents. They are referenced in this document in the form [RD xx]:

[RD01] Solar Orbiter Low-Latency Data: Concept and Implementation, SOL-SGS-TN-0003
[RD02] EDDS External User Interface Control Document, EGOS-GEN-EDDS-ICD-1001
[RD03] Metadata Definition for Solar Orbiter Science Data, SOL-SGS-TN-0009

5 CONSIDERATIONS & OBJECTIVES

Here we describe the factors driving the choice of architecture for hosting and delivering the low latency pipelines. **The concrete agreed specifications for these pipelines and for the delivery mechanisms follow in** 6 and later sections.

5.1 Deliverables

Each of the ten Solar Orbiter instrument providers will supply a low latency data processing pipeline software system. This will be a single integrated software product per instrument, thus the SOC will host ten such pipelines.

5.2 Interfaces

In order for instrument teams to be able to produce and configure their Low-Latency Data processing pipelines to be run at the SOC it is necessary to define in detail a number of interfaces. Specifically we must define:

- How the software will retrieve the necessary input data.
- How, where and in what format the software will store the generated products.
- How, and in what form, each instrument team will deliver the pipeline software to the SOC.

5.2.1 Uniformity

It will be to the great advantage of the SOC, but indeed of all parties, to keep these interfaces as uniform as possible across all instruments. The advantages of this simplicity to the SOC should not need explaining, but the instrument teams will also benefit in being able to learn from each other and perhaps share procedures or code. With this uniformity the SOC will also be in a better position to assist, and should be better able to provide feedback.

5.2.2 Data Retrieval

The input data for the LL pipelines in use at the SOC will be fetched (from the MOC) by SOC processes separate from the pipelines and presented to the pipelines as per the scheme outlined below [6.2.1, 6.2.6].

That is, the pipeline appliances will not be responsible for the fetching of telemetry nor auxiliary data. Rather, they will be presented with it locally. Such an architecture, with

centralised fetching of the input data, allows for the SOC systems to coordinate the different pipelines and use them in an integrated fashion as will be described.

Of course when running at their own sites the instrument teams will have to fetch the data to feed into the pipelines. They will have to implement their own fetch-and-present-data functionality in a manner similarly decoupled from the core LL processing pipeline. This should allow identical (or close to identical) pipelines to run at instrument team sites and at the SOC.

The baseline for data retrieval by instrument teams at their own sites is that they fetch data directly from the MOC, and only (auxiliary) products not available from the MOC would be fetched from the SOC. Instruments do however have the option to fetch all data from the SOC.

The nature of the interface via which the instruments will retrieve data from the SOC has not yet been decided. This might be significantly different from the DDS interface to the MOC.

5.2.3 Products

The products of the low-latency pipelines discussed here must be appropriate to fulfil the purposes defined in [RD01]. The roles for LL data described there can be paraphrased as follows:

- Allow crude checks of instrument performance and science data quality.
- Allow instruments implementing selective data downlink to choose the telemetry to retrieve.
- Allow the improvement of pointing profile and/or retargeting when tracking solar features.

The SOC is in on-going discussion with each instrument team to agree the details of the pipeline products and ensure that they both meet the needs of the SOC (and the mission as a whole) and can be produced without unwarranted difficulty by the instruments. The outcome of these discussion have been captured in and will be further detailed in later versions of [RD01], as well as in ICDs [AD01] and [AD02].

The products output by the LL appliances will not in all cases be those which are then circulated and used – see below [5.2.4]

Since the purpose of the LL data is strictly limited as described above, and since the timeliness aspect supports the idea of pipelines that are simple and robust, some guidelines for the processing to be performed by the instrument-supplied LL pipelines can be enumerated

- LL pipelines do not produce science. The output is not suitable for publication.
- The need for updating of calibration parameters within the LL pipelines shall be avoided if possible, and otherwise minimised.

- LL pipelines nominally run only once against the data set as it is available following the end of the pass. Later reprocessing of LL datasets due to pipeline improvement is excluded. Reprocessing due to gap-filing is considered exceptional.

The SOC anticipates that, for many instruments at least, the LL pipeline delivered to and run at the SOC would be very similar (or perhaps even close to identical) to a core element of the pipeline which the instrument teams will wish to run for their own purposes at their own sites.

5.2.4 Post-Processing by SOC

The SOC considers it advantageous to perform centralised post-processing of products output from the instrument-team supplied LL pipeline appliances. Any such post-processing would be under SOC- responsibility. The perceived advantages are in cases:

- where there are common manipulations to be performed across multiple instruments.
- where there is the need to access auxiliary data that could be inconvenient for the pipelines to access.

The planned post-processing of LL pipeline output is as follows:

1. For the RS instruments the instrument-supplied pipeline outputs image products (these will all be FITS files) with header data that provides pixel directions simply relative to the instrument bore-sight. SOC post-processing will add all the necessary additional header information for interpretation in a basic WCS reference frame, as described in [RD03].
2. For in situ instrument pipelines output should be in instrument coordinates for vector and tensor quantities. SOC post processing will rotate these to spacecraft centric RTN coordinates (Radial, Tangential, Normal) where R points radially antisunward, T is the cross product of the Sun's spin axis and R, and N completes a right handed set (so ~ North for a spacecraft in the equatorial plane of the Sun).
3. All time-stamping in pipeline output files will be done in terms of On Board Time. SOC post-processing will perform time correlation and output UTC time-stamped version of these files.

5.3 Testing and Testability

5.3.1 Test Environment

In order to be able to develop and provide software which will function correctly on the SOC infrastructure for Low-Latency data processing, each instrument will require an environment which sufficiently matches that to be used at the SOC.

The proposed concrete specification for LL pipelines below [6] describes how these must fit into their environment. Specifically, the planned SOC architecture for such an environment is described in 6.3.

5.3.2 Configurability for Testing

Even a near perfect test environment available to an instrument team will not be identical to the environment in which it has to be run. As such it will be necessary to make configurable certain behaviours of the software, for example, the location to which products are stored.

5.3.3 Provision of Test Data & SOC Pipeline Acceptance

The SOC will require a mechanism to verify that delivered LL pipeline appliances deploy correctly to the SOC environment and function there as anticipated. That is, there will be a SOC acceptance process.

So as to allow such a process, the SOC will request, from each instrument team, input data for a series of test cases, corresponding appropriate output data, and a mechanism for determining whether the actual output when a test is run is correct.

It is anticipated that such test cases would be based on those used internally by the instrument team. These tests would be reviewed by the SOC so as to understand the extent of the testing performed .

The SOC wishes to automate testing where at all possible. As such, it is highly desirable that the mechanism referred to above for determining whether a test passed, be a script.

SOC acceptance testing would not be limited to just those tests provided by the instrument teams.

5.3.4 Connectivity versus Isolation and Diagnostics

Computer security considerations, architectural simplicity, and the SOCs ability to fully understand the configuration of the software it is running are all served by isolating the running LL pipelines as much as possible. On the other hand it is understood that instrument teams may be better able to diagnose problems occurring at the SOC if they can remotely connect to the environment in which those problems occurred. The expectation that each team has a test environment which duplicates rather faithfully the SOC deployment environment (see 6.4) should allow replication of errors in most cases, but there are likely to be other cases where this is not possible.

6 SPECIFCATION FOR LL PIPELINES TO BE DELIVERED

Based on the considerations listed above, the specification for the Low-Latency pipelines to be delivered to the SOC is as follows.

6.1 Form of LL Pipeline Software

The pipeline software will be delivered as virtual appliances. Further explanation of the virtual appliance concept can be found in Appendix A, and a justification of the approach is in Appendix B.

The virtual appliances to be used are ***for the Intel x86 architecture***. It is assumed these will be 64-bit VMs, but 32-bit will also be acceptable.

The OVF archive files to be delivered will be named according to the following scheme:

LLDP-<Instrument Name>-XX.YY.ZZ.ova

Where <Instrument Name> is replaced by precisely the same instrument name string (including use of uppercase) specified in 6.2.6 for use in the data directory tree.
XX, YY, and ZZ will be the instrument team's major, minor, and fix version numbers respectively.

6.1.1 Standards

There are various providers of virtualisation systems, and each of these has developed its own means of capturing the state of a *guest* virtual machine and saving it to allow it to be launched again (in the state in which it was captured), from the same, or from a different *host*.
There is also now an open standard format for the storage of such virtual machine ***snapshots***. This is the Open Virtualisation Format. It has now been supported by the major virtualisation providers for some time.
As stated above OVF will be used for the delivery of all virtual appliances to the SOC.

6.1.2 Operating Systems

The SOC would very much like to encourage instrument teams to use Linux of one flavour or another if at all possible. Failing that, another Unix derivative such as FreeBSD is thought likely to be easier to work with than (for example) Microsoft Windows.
In whatever case the Operating System should be one which is widely supported by virtualisation vendors. Note that this is not expected to be a difficulty as these systems typically support the great majority of recent and reasonably-commonly used OS variants.

During development instruments should ideally regularly verify that their systems are deployable on some target virtualisation infrastructure other than the one which they use day-to-day.

6.1.3 Licencing

It shall be the responsibility of the individual instrument teams, for the duration of the mission, to provide any licences for commercial (or other licenced) software (including OS's) which their pipeline appliances require.

However for the purposes of technical and administrative simplicity – specifically to avoid the need for access to external licence servers - appliances making use of certain licenced software may do so using floating licences from ESAC pools. Specifically this will be

allowed for IDL. If an instrument team is using an MS Windows OS a similar mechanism will be allowed if delivery of an already activated VM is very problematic.
In the case of IDL, instruments should configure themselves for use of ESAC licences by fetching an entire licence file from a location given via the start-up configuration mechanism described in 6.2.1. This should be placed in the normal IDL licence file location.
If and when it becomes known that an instrument intends to use Microsoft Windows and needs an ESAC licence then a configuration mechanism shall be described. This is not expected to be complex.

No LL pipeline will make use of a product which requires node-locked licences nor access to a remote licence server other than as described above (ESAC licence pools). The SOC urges teams to check this carefully before choosing licenced software.

In addition, if any software falling outside the above categories requires licence configuration which cannot be done in the VM before it is delivered to the SOC this should be brought to the SOC's attention as early as it is known.

6.2 Required Behaviour of LL Pipeline Software

6.2.1 Appliance configuration at deployment

So as to be deployable in multiple environments –both multiple at the instrument site where they are created and multiple at the SOC – it must be possible to configure the appliance VMs at the point of deployment.

Rather than expecting those deploying the VMs to understand their internals and modify the images before booting them, the appliances will be required to configure themselves on start-up according to a series of parameters/properties which will be passed to them via the virtualisation environment.

These properties will take the form of *token=value* pairs where the value is a string. These properties will be passed via a file made available to the VM at boot time. The precise location of the file is TBD.

A provisional list of the properties to be read and honoured by the appliances is as follows.

Purpose	**Token**	**Example**
IP Address	LLVM_IP	192.168.1.100
Net Mask	LLVM_NETMASK	255.255.255.0
Gateway	LLVM_GATEWAY	192.168.1.1
Host Name (FQDN)	LLVM_FQDN	instrument-ll.subnetn.lan
Primary DNS Server	LLVM_DNS1	192.171.2.18
Secondary DNS	LLVM_DNS2	192.171.2.19
Data Input Dir	LLVM_INPUTDIR	esac-nfs.netx.lan:/data/input
Data Output Dir	LLVM_OUTPUTDIR	esac-nfs.netx.lan:/data/output/instrument

NTP Server (time)	LLVM_NTP	192.171.2.20
IDL licence file	LLVM_IDL_LIC	ftp://192.168.1.150/IDL/licence.dat

6.2.2 Access to Data

Access by individual appliances to input data and similarly storage by them of the generated data products will be conducted exclusively to and from mounted disks. This should allow rather simple adaptation of the appliances to the development (instrument site) and deployment (SOC and other) environments, as well as to test environments at any location.

The mechanism for the mounting of these disks will be NFS. Thus the appliance to be delivered must support this.
Exceptions are the following:

- If an instrument strongly wishes to deliver a Microsoft Windows based appliance (discouraged and understood to be unlikely) it may be possible to support mounting via CIFS.

Each appliance will mount input and output directories separately, the input directory mounted read only (this will be an advantage should it be necessary to clone the VM). See 6.2.9 below for data tree description, and 6.2.1 for the means of configuration of the locations to be mounted

6.2.3 Appliance Configuration Accounts

All appliances will be delivered configured such that access to the root/administrative account is possible with a specific password agreed in advance between the instrument team and the SOC. The primary purpose envisaged for this is to enable SOC diagnostic activities.

6.2.4 Processing Batch Size

Pipeline appliances will be able to process, and produce useful output from, the telemetry acquired from a single ground pass and its associated auxiliary data, in semi-isolation from the preceding history of the mission and the instrument. The exceptions to this 'isolation' are limited to:

1. The telemetry and auxiliary data from the immediately preceding downlink will normally also be made available when the request is made to process the current downlink. Note that the pipeline must be able to function when no such preceding dataset is included in the request – but there is no requirement that it create products from any data at the beginning of the latest download which constitutes the incomplete tail-end of a data subset (e.g. of an image).
2. Any improvements needed in the processing mechanism, for example in (presumably limited – low latency data is not intended for science) calibration, will be implemented via the delivery of a new appliance to the SOC. Exceptionally calibration improvement essential to the LL purposes may be made via instrument team delivered auxiliary files to be presented in processing

requests, where it can be clearly shown that this can be done simply and is the only practical mechanism.

6.2.5 Pipeline Statelessness

Pipeline appliances will be ***stateless.*** That is to say that, presented with the same input data they produce identical output (excepting of course processing timestamps, version documentation etc.), unaffected by the history of their previous activity.
This statelessness will be very important for the SOC's ability to request the reprocessing of previously handled data should there be some processing failure or some data loss.

In addition to the statelessness of appliances with respect to the processing of separate batches of telemetry, where the telemetry in a batch can be broken up into separate elements (perhaps images) for processing, the appliance will do this in such a way that the failure of the processing for one element does not prevent the correct processing of subsequent elements.

6.2.6 Processing Request Mechanism

Pipeline appliances will process data, in discrete batches, only when explicitly requested to do so by an external system (in operational deployment, a SOC system). The detection of such a request will be the prompt which spurs the appliance, which will at other times be idle, into life.
A request for the processing of new data shall be presented to an appliance by the creation of a new directory corresponding to the request in a 'Requests' directory (specific to that instrument) monitored by the appliance.

The directory where requests are placed shall be:

<data mount point>/input/<instrument name>/requests

The name of each specific processing request directory shall take the form:

request_YYYYMMDD_HHmmSS _<any name>

The instrument name strings used will be the short form instrument names as specified in numerous documents, for example the Science Management Plan, SOL-EST-PL-00880 issue 2.2 section 2.6. These directory names shall be precisely as the instrument names in that document. All are entirely capitalised with the exception of SoloHI. The directory names will thus be:

- EPD
- EUI
- MAG
- METIS
- PHI
- RPW
- SoloHI
- SPICE
- STIX

- SWA

YYYYMMDD_HHmmSS of course represents the date and time at which the request is created.

<any name> represents a string which can be chosen freely by the requesting client so long as it **contains only alphanumeric characters and (further) underscores**. The low latency pipeline appliance need not interpret this string in any way.

As is illustrated below in Figure 2, in deployment at the SOC these processing requests will be created by SOC systems after data has been acquired.

This new directory constituting the request will include all the LL telemetry available from the MOC DDS for this instrument for a particular time period, and any corresponding auxiliary data.

Note that the architecture here described is designed to handle the passing of any request-specific auxiliary data alongside the telemetry, and also the provision to the pipelines of any more globally (across-instruments and time) shared auxiliary data. However the current SOC proposal (due to the nature of the SOC post-processing proposed) is that no such auxiliary data will be needed by most of the LL pipelines themselves and the corresponding auxiliary data directories will be empty. One exception to this is currently seen as likely and is under discussion. The auxiliary data directories may be empty, but they will not be absent.

The telemetry will be in a sub-directory *telemetry* of the request. Any auxiliary data (or none) will be in the subdirectory *auxiliary*.

The telemetry described above will be in a single file precisely as returned by the EDDS – see [RD02]. By default this file will be the EDDS XML version. However an instrument may request that it be delivered instead the EDDS binary format.

This file will contain all the telemetry packets available for the time period concerned which have APIDs from a list (very possibly of 1 entry) provided by the instrument team to the SOC beforehand. It is anticipated that this list of APIDs will change very rarely if at all during the mission. In any case a new list will be communicated to the SOC at least 10 days prior to its intended use. It is expected that a new APID list would also involve the delivery of a new pipeline appliance version.

An instrument team may include APIDs identifying house-keeping data in this list if they consider it necessary.

The request may include a similar dataset for an immediately preceding time period. This earlier dataset, if it is present, will be used only as a source for the initial parts of data items (e.g. images) only the end of which appear in the dataset which is the principal target of

processing. Normally these two datasets will correspond to all the LL data for the instrument from the latest downlink, and from the previous.

The dataset for the preceding time period will, if present, be in a separate single file in a subdirectory *preceding* of the request. If such a dataset is not available the subdirectory *preceding* will be absent.

If the preceding dataset is present the pipeline will reconstitute the split data subset and produce all the corresponding products as part of the output for the current request.

In addition to the case where the data for the preceding period is not made available, on occasions there will also be packets missing from the current downlink. In such cases the LL pipelines should attempt to produce products wherever possible unless it is clear that such products would be deceptive or inappropriate for the Low Latency purposes specified in 5.2.3.

So as to guarantee integrity when detected by the relevant appliance, these request directories will be created by the SOC in a staging area before being atomically moved to the relevant 'Requests' directory. It is currently assumed that no explicit locking mechanism of which the appliances will have to take account will be needed.
Notably in the case of reprocessing of data the SOC may create multiple requests for a single instrument at the same time or in rapid succession. It is for the instrument team to decide whether their appliance can safely handle these requests in parallel or they should be done in series.

6.2.7 Processing Output

In response to a request for processing, an appliance will create a directory with precisely the same name as the original request directory, and it will populate this directory with all the corresponding products as specified in [AD01] and [AD02] (format) and in the Dataset Description Document for Low Latency for the instrument in question (enumeration). This 'requested products' directory will be created in *temporary* a staging area directory *on the output mounted disk*. It will be moved to its final location by the appliance (only) when complete, and this move will be performed atomically.

6.2.8 Failed Processing

If a virtual machine becomes aware that processing has failed for a specific request – that is, it has not been possible to produce the expected useful products – then instead of creating a new directory with the results inside *products*, a new directory with the same name (identical to that of the request) will be created inside directory *failed* in the instrument output directory.

This may mean making the final atomic move of the request response directory from *temporary* to *failed* instead of to *products*. In such a case it may make sense for the appliance to leave partial products inside the response directory. As much information as possible should be included in the log.

6.2.9 Data Tree

A partial tree of directories to be shared by the SOC systems with the appliances is shown in Figure 1.

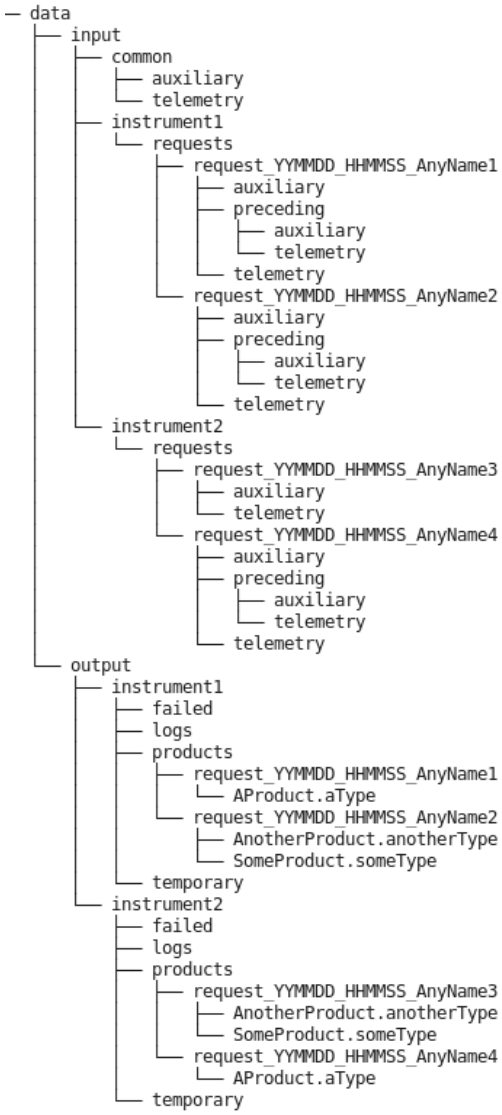

Figure 1 Partial Tree SOC Hosting System /PipelineAappliance Shared Disk.

In interpreting Figure 1 note the following:

- The name *data* is shown as the root of the tree – this is just illustrative. Appliances will be deployed to mount separately the input and output directories shown here at */data/input* and */data/output/<instrument>*. */data/input* will be mounted read-only. The appliances will be designed for and tested with this configuration.
- Input and output directories are shown only for 2 instruments. There will be 10.
- The instrument names will not be instrument1, instrument2 etc. Instead they will be the names specified in 6.2.6.

- In other cases all directory names shown are precisely as they should be (including variable suffixes) and fit the descriptions (also) given in 6.2.6
- The file names for products are examples. The actual names shall be defined in [AD01] and [AD02].
- The files containing the telemetry inside the requests are not shown. There shall be a single file per telemetry directory.

6.2.10 Restricted and Private Products

Inside the directory containing the products for a particular request, an instrument LL pipeline may create a directory *restricted*. Any products created inside this directory will, be made available to all instrument teams (and the SOC), but not publically. SWT 16 decided that all Low Latency products will be shared between all instruments for operations purposes, so no greater restriction than that described above will be necessary.

6.2.11 Logging

Each instrument will, in directory output/instrument/logs maintain a log of its actions. This will contain only one file with extension '.log'. This will be the log file currently being written to. If the instrument wishes to implement log rotation or archiving then non-current files should have a different extension.

The logs should contain, a clear record of when the processing of each request is started and when it is completed. These tokens in the log should be easily found by searching for the request directory name.

6.3 SOC Deployment and Context

This section describes the way in which the SOC expects to deploy the delivered appliances. This description should allow the instrument teams to understand the context in which the software they are delivering will be used. Amongst other uses, such understanding should allow appropriate test design.

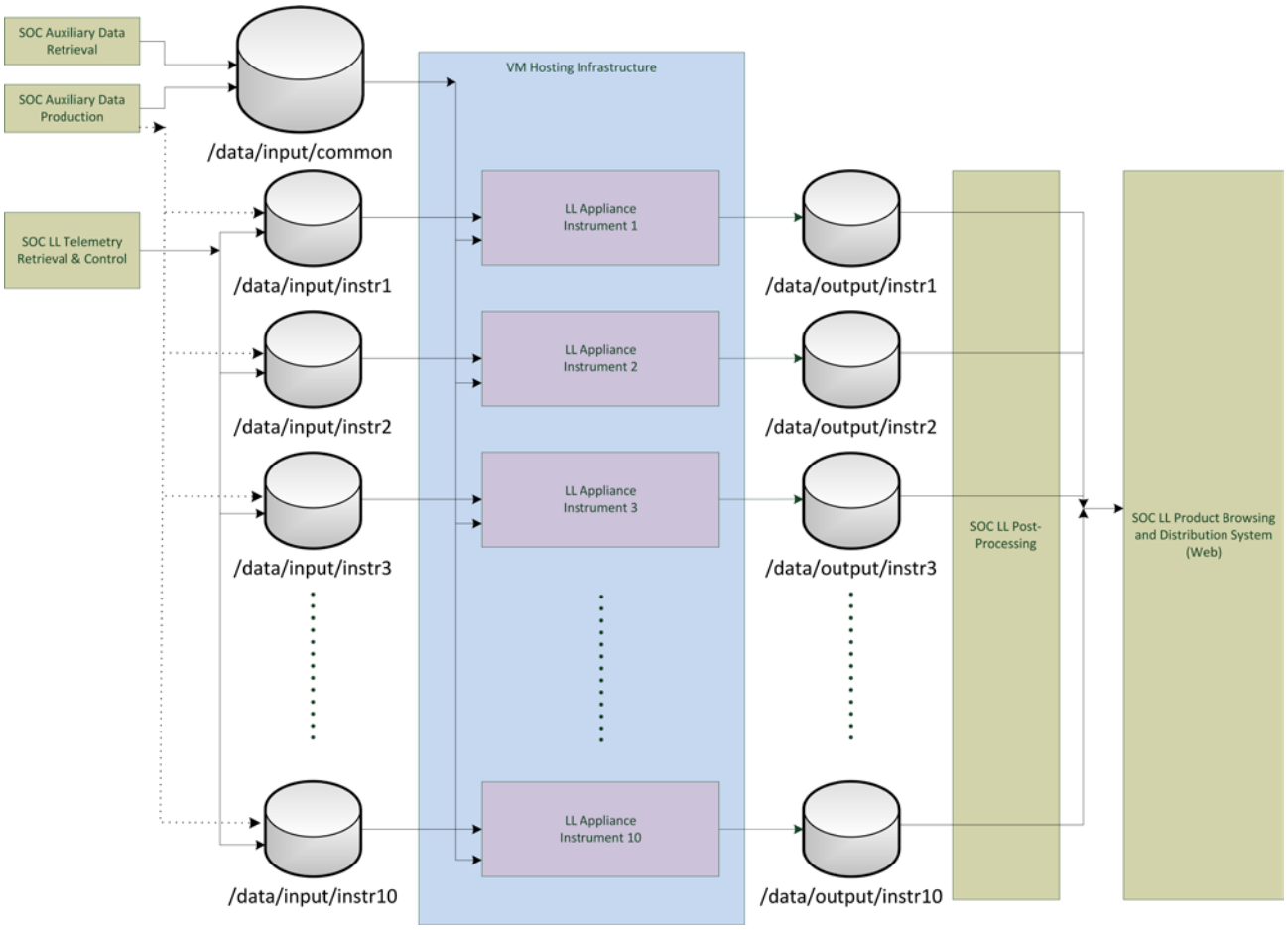

Figure 2 SOC Low-Latency Appliance Deployment Context

When reading Figure 2 please note the following:

- As mentioned previously, it is now envisioned that in only a small number of cases will per-instrument auxiliary data be provisioned by the SOC. Specifically SoloHI reference images will be made available to the VM in this way. For which other instruments there will be such data is not finalised – thus the dotted lines.
- There is currently no envisioned content for /data/input/common. However, since such common (across instruments) input elements are not yet completely excluded, it is left in place as an architectural element for now.
- The *instrX* directory names will of course correspond to the names specified in 6.2.6 above.
- The input and output disks will be exported separately for sharing by the SOC systems. The input disk will be mounted read-only.
- Each appliance will mount only its own area of the output disk.

6.4 Non-Functional Specifications

6.4.1 Size and Resources

Each appliance image will not occupy more than 30 GB.

An appliance shall not 'grow' during operations such that more than 30GB is needed for a new image of that VM. It is expected that this will follow almost necessarily from the statelessness of the appliances (see 6.2.5 above).

Each appliance will be capable of running (and fully performing its function) when provisioned with:

- 4GB of memory (this on request – 2G B will be provided by default)
- Two virtual cores operating at 2.9GHz from (as a reference) Intel(R) Xeon(R) E5-2690 CPUs.

6.4.2 Timing

Each appliance will detect and begin processing a request within 1 minute of the request being created.

The above constraint on the delay before processing begins will not apply while an earlier request is underway, although the appliance may choose to process requests in parallel.
Each appliance will, given the computer resources described above, process the data corresponding to any set of LL telemetry envisaged to be downlinked in a single pass, in a period of less than 30 minutes.

7 TESTING, DELIVERY AND ACCEPTANCE

7.1 Instrument Team Development Context

Instrument teams will require an environment (or multiple such) representative of the SOC deployment architecture (excluding the elements specific to other instruments) shown in 6.3 above. This will be required to develop and then to run tests of the appliance before attempting to deliver it to the SOC.

7.2 Provision of Test Data

Each instrument team shall provide to the SOC:

1. input data for a series of test cases
2. corresponding appropriate output data
3. algorithms for determining whether each test output is correct i.e. test passed

It is anticipated that such test cases would be based on those used internally by the instrument team. These test would be reviewed by the SOC so as to understand the extent of the testing performed.

The SOC intends to automate testing where at all possible, and requests that all tests are such that the results do not require manual checking.

Because of its intended use (see 7.3 below) the test set to be run at the SOC should not take an excessive length of time to run. In cases where the entire proposed set of tests does not complete in 2 hours the SOC and instrument shall discuss restricting the scope.

These tests need not be as extensive as those run by the instrument teams.

7.3 Delivery of Appliances

The SOC will provide a web interface via which instrument teams can upload appliances as OVF archive files.

The intention of the SOC is that this process also automatically trigger the testing of the new appliance against the agreed tests currently held for the instrument in question. The baseline is that these tests will simply be those delivered by and agreed with the instrument team previously. It is not excluded that some others might be added by the SOC e.g. to verify interoperability with other SOC subsystems. If the tests are passed then the new appliance will be assigned a (SOC) version number and archived. Otherwise the instrument team will be informed automatically of the error.

The mechanisms for the automatic deployment (for testing) of a new appliance are still in preparation. A fall-back mechanism is that after upload the appliance must wait for some manual intervention for deployment to a test environment and (very likely still automated) testing.

Before attempting to deliver any new appliance which (intentionally) requires updated tests, the instrument team must have delivered and agreed the new tests with the SOC and arranged for their deployment.

The SOC anticipates that instrument teams would also be able to upload their test sets via a web interface.

7.4 Acceptance

Passing the tests described above will constitute acceptance of the appliance by the SOC.

7.5 Deployment to Operations

The deployment of a newly delivered appliance (which has passed acceptance testing as described) to the SOC operational infrastructure (replacing any previously existing one) will take place at the discretion of the SOC Development/Operations manager. The instrument team and will be informed when the appliance version running at the SOC changes.

APPENDIX A VIRTUAL APPLIANCE BACKGROUND AND TERMINOLOGY

Following a suggestion from EUI and internal analysis, the SOC proposes taking delivery of the Low-Latency pipelines as virtual-machine images loadable into system virtualisation software; that is, such software as VM Ware, VirtualBox etc. This mechanism, where an operating system (or *just enough* operating system) is bundled alongside an application for distribution is known as a Software Appliance, and in the virtualisation context, as a ***Virtual Appliance***.

The virtualisation being discussed here is System Virtualisation where each virtual machine can host a separate operating system. This is to be contrasted with virtual machines which host single (application) processes – this latter is most familiar to many in the form of the Java Virtual Machine (JVM).

A system Virtual Machine such as we propose involves a ***guest operating system*** which runs in a virtual machine created and managed by a ***hypervisor*** – the core virtualisation software – which itself runs (often) under the control of a ***host operating system***. Such a VM is an *appliance* (and so a *virtual appliance*) when it encompasses and is dedicated to a single software application.

For example, pipeline S/W for an instrument is written in Python running on an instance of Ubuntu which itself was started in VirtualBox running on a Red Hat Enterprise Linux server. Here, this version of Ubuntu (encompassing the Python pipeline) is the guest operating system, VirtualBox is (or encompasses) the hypervisor, and RHEL is the host operating system.

Note that the *host operating system* is not always present. There are versions of *hypervisors* which run on 'bare metal'.

APPENDIX B JUSTIFICATION OF USE OF VM'S

The use of Virtual appliances is being proposed by the SOC after having looked into the suggestion. The main points taken into account in coming to the conclusion that this is the most appropriate mechanism for pipeline delivery are as follows.

Advantages

The virtual appliance mechanism should almost entirely eliminate the software integration task on the operational systems, making deployment semi-trivial. It is no longer necessary to concern oneself with the software environment in which the application runs – the accounts and directories in which the software is installed, the libraries available in which locations and their exact versions, and any conflicting configuration requirements from other software needed on the same system. The great majority of this environment is delivered along with the application S/W.

There is still some need for the configuration of the environment in which the guest OS – the appliance – is run. However this is expected to be limited to such high-level items as network addresses and disk mount points.

One motivating consideration is the long term over which this software must be in use. It can only be kept in use for as long as the platform on which it runs remains available, or, alternatively, as long as the effort of porting to new platforms can be maintained. This second depends both on the availability of manpower and of know-how; it can be costly and uncertain.

Thus any mechanism which means that the first option can be relied upon, that is, that we can have confidence in long-term continued availability of the platform is a great advantage. Until recently continued platform availability hinged on (as well as the reliability of the existing hardware) the continued availability of hardware which supports the operating system (version) in use. Virtualisation changes this.

A number of hardware architectures have come and gone from the market in recent decades – but these were always lesser or niche players; the Intel x86-based PC architecture is so widespread and the existing body of SW so great that it is hard to believe that any end to production of backward compatible hardware is imminent. However, current backward compatibility is not complete (e.g. various tricks are needed to get modern PC hardware to boot DOS), and we must assume that at a hard-to-guess point appropriate machines will no longer be available.

Virtual environments mean that the platform on which a system (including OS) runs is theoretically no longer tied to a particular hardware architecture. This platform is now the *virtual* machine. The VM creates (or is) an environment for the guest OS which meets all the guest's interface expectations – most often (today) it appears just like a known physical hardware platform – and that is most often the x86 architecture. However so long as an

appropriate VM is implemented, the underlying hardware need not be that which the VM simulates. That is the theory. However current virtualisation environments, while practically very effective, are predominantly geared towards providing many virtual x86 environments on x86 hardware, and to do this they depend – for some features at least – on the underlying physical architecture and on specific hardware support for virtualisation.

Nonetheless it seems evident that virtualisation considerably reduces the barriers to maintaining the availability of a platform. Whichever way hardware architecture evolves in the coming years the size of the installed base means that there is a great incentive for virtualisation providers to support guests based on the x86 architecture. It may even be that the popularity of virtualisation reduces in the eyes of hardware providers the need for direct backward compatibility, and so reduces the expected lifetimes of any systems not taking advantage of virtualisation.

Possible Drawbacks for SOC

For ESA this approach to the delivery of pipelines to the SOC is novel. Common practice for Science Operating Centres which host data-processing pipelines has been that the SOC provide some level of infrastructure software and the instruments centres provide source code for the implementation of the particular pipelines. These have been integrated, and in part debugged at the SOC.
As such, many SOCs are accustomed to having sufficient understanding of and access to the pipeline source code so as to be able to at least provisionally investigate anomalies pending fuller instrument team investigation (where appropriate).
This drawback, if it is such, must be understood in the light of the fact that the SOC experience described comes from Astronomy missions with a small number of instruments – to date no Solar System missions have run pipelines – also no quick-look pipelines - at ESAC.
This issue can in any case be considered academic in that it is unclear if even a very large scale SOC could sufficiently understand the pipelines of 10 instruments and support them in the way described above. Solar Orbiter clearly will not have the SOC manpower necessary to follow this approach.
It remains worth noting that the ‘black-box’ pipeline concept inherent in the proposal made here is new to ESAC.

Possible Drawbacks for Instrument Teams

It will of course be necessary that teams understand something of virtualisation. This technology is however already rather mass-market and in widespread use by non-specialists. Although virtual appliances are not the most prominent use of the technology, it does not yet seem that there are obscure or particularly difficult areas of virtualisation involved. It seems likely to the SOC that many instrument teams will be using virtualisation already, and perhaps even for similar purposes, and so the additional knowhow needed will not be very substantial.

This knowhow will most likely also be useful to the instrument team internally, as the deployment advantages described above can apply also to the internal deployments the instrument teams make – for testing purposes and others.

Maturity & Experience

The use of virtualisation as a means of adding flexibility to computer system maintenance is no longer novel. Its use is now very widespread and its reliability and ease of use generally those of a mature product. There are now various suppliers of virtualisation software, including open-source systems.
In the ESAC Computer Support Group there is more than 10 years of experience of more than one virtualisation system. This has not involved virtual appliances to a great extent, but the XMM mission has delivered virtual appliances ***from*** the SOC for a number of years, and recently, for Euclid, ESAC has been receiving virtual appliances from third parties. No particular difficulties arising from this have been brought to the attention of the SOC.

APPENDIX C ALTERNATIVES CONSIDERED

Alternatives to virtual appliance software delivery

These are listed starting with those most similar to the main proposal.

Live CD

This is a mechanism to provide a Software Appliance as an ISO CD image. This allows for (very simple) deployment to a standard hardware platform. As such, this format does not require any use of virtualisation - but then would require dedicated hardware – typically a PC with the resources necessary to meet the performance requirements.
Use of a Live CD does not however exclude virtualisation. The Live CD can be used as the boot medium to create a new guest operating system for the chosen virtualisation system.

It is assumed that what would be provided is more likely to be an ISO image sent electronically rather than a physical CD.

Creation by SOC of an appliance for deployment

This is not a complete alternative, but rather refers to the possibility that the SOC, having received a QL pipeline delivery in some other format, create from this a virtual appliance which is then deployed.

This would most likely be a mechanism to meet SOC internal needs in the case where for some reason universal delivery as appliances, virtual or otherwise, was not viable. Most likely it would be of interest in the case that almost all instruments delivered appliances, but one or two for some reason could not. This mechanism could allow the SOC to preserve a uniform deployment model across instruments.

It might also be resorted to as an interim measure while an instrument acquired the skills needed to create an appliance – but this assumes a greater degree of difficulty in virtual appliance creation than anticipated. It also assumes that the difficulties would be ones more easily addressed by the SOC than the instrument team – probably also unlikely.