



# **Bilan DC-1**

*H. LOUVIN*

KP-0, Janvier 2020





## Messagerie NATS

- Mise en place d'un serveur NATS au LAL
- Mise à disposition de modules python pour faciliter la communication avec le serveur NATS, la base de donnée VHF et la SDB

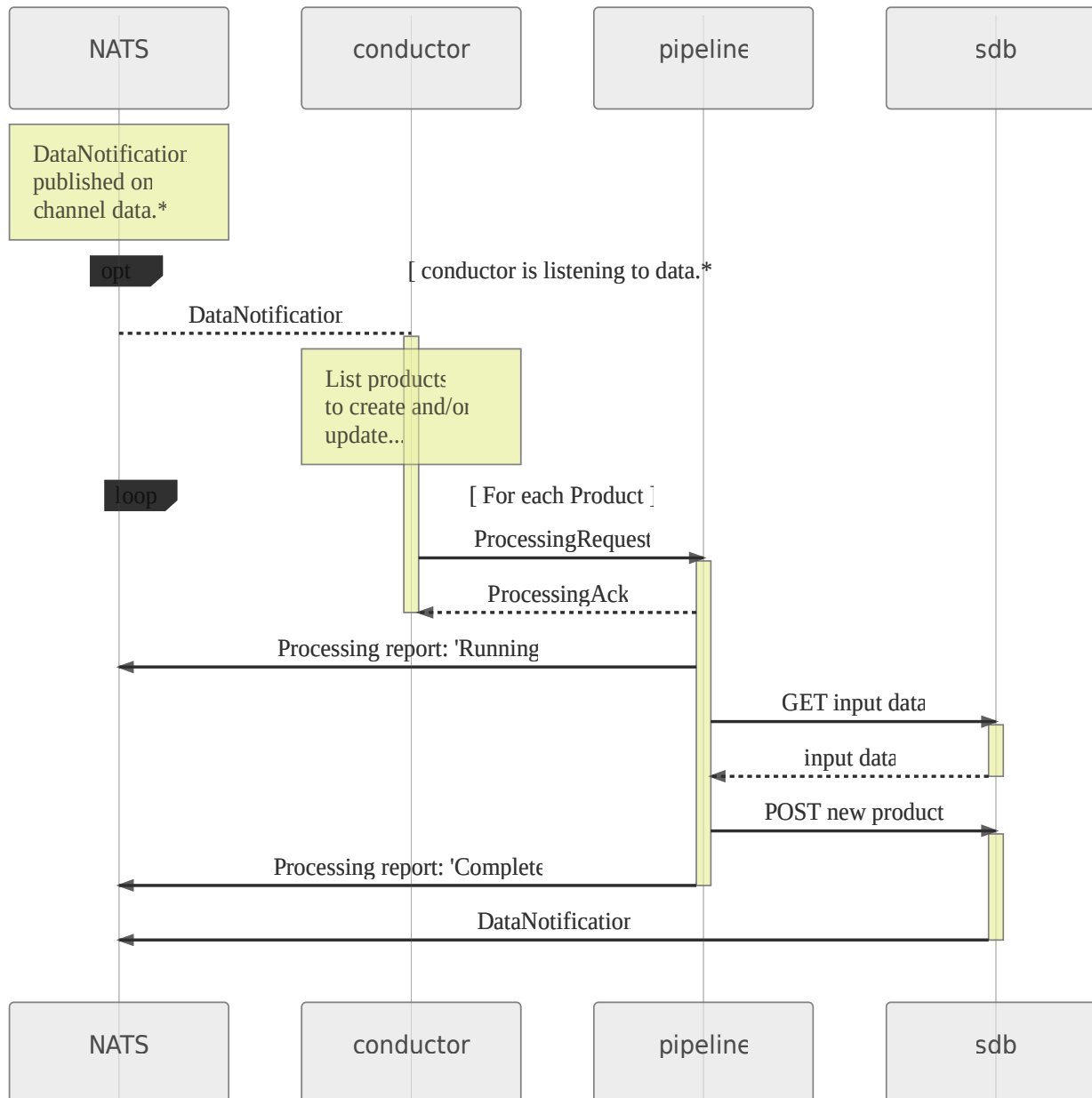
The screenshot shows the NATS Streaming Server Webpage interface. At the top, there is a header with the SVOM logo and the text "NATS STREAMING SERVER WEBPAGE". Below the header, there is a navigation bar with "Channel(s):" followed by "ALL", "data", "activity", "science", and "test". The main content area displays a JSON message received from the server. The message is a list of messages, with the first one being a "default" message and the second one being a "2020-01-20T10:37:43.079551 [activity.core\_program]" message. The second message contains a "service\_descriptor" and a "processing\_descriptor". The "service\_descriptor" includes "name", "uri", and "instrument" fields. The "processing\_descriptor" includes "process\_id", "process\_name", "target", "mode", "submission\_date", "completion\_date", "input\_data", "processing\_status", and "message" fields. A "Clear Panel" button is visible in the top right corner of the message display area.

```
    "links": [
      "default"
    ]
  },
  "activity": "default",
  "date": "2020-01-20T10:37:42.222885",
  "status": "wait",
  "info": "default"
}
}
2020-01-20T10:37:43.079551 [activity.core_program] {
  "service_descriptor": {
    "name": "OblcEclPipeline",
    "uri": "svom://vhfpreproc#oblc",
    "instrument": [
      "ECL"
    ],
    "mode": "CoreProgram",
    "creation_date": "",
    "version": "0.0",
    "links": null
  },
  "processing_descriptor": {
    "process_id": "svom_vhfpreproc_oblc_bcae45cc-3b70-11ea-8ff8-0242ac140a87",
    "process_name": "NoName",
    "target": {
      "apid": "None",
      "target": "OBLC_ECL",
      "orbit": "1",
      "mode": "CoreProgram"
    },
    "mode": "todo",
    "submission_date": "2020-01-20T10:36:37",
    "completion_date": null,
    "input_data": null,
    "processing_status": "Complete",
    "message": null
  },
  "messages": null
}
```

Messages received since Mon Jan 20 10:39:02 2020 (UTC)



# Orchestration






## Orchestrateur

- À l'écoute des canaux **NATS data**.\*
- A accès à la **liste des produits disponibles** pour un obsID donné (dictionnaire interne)
- **Combine les informations** de la notification NATS et de la liste des produits disponible pour déterminer si un nouveau traitement doit être déclenché
- **Envoie des requêtes** de *processing* au(x) service(s) pipeline(s) concernés
- **Trace les traitements en cours** et s'assure qu'ils sont finis correctement
- Communication du statut du service via NATS (canal *activity*)



## Orchestrateur

- Service up and running au LAL
- Deux pipelines associés au traitement par orchestration (Light curve ECLAIRs et notices VOEvent)

 ORCHESTRA WEBPAGE

OBSID	COMPLETE PROCESSES	RUNNING PROCESSES	QUEUED PROCESSES	FAILED PROCESSES
1	OBLC_ECL ( bcae45cc-3b70-11ea-8ff8-0242ac140a87 ) ecl_notice ( deadf0b6-393e-11ea-a3aa-0242ac140a8d )			
..	..	..	..	..

To send a fake VHF data notification to NATS on channel data.vhf, fill the form below and hit

Packet type:

ObsID:

To cancel pending processing requests click there

To clear the table click **this button**

*WARNING: This will delete all entries in the product manager table*



## VOEvents

- Pipeline+broker déployés au LAL
- Interface REST compatible avec l'orchestration (#pipeline-bricks)
- Déclenchement de traitement **par l'orchestrateur**
- **Récupération** des paquets VHF alertes ECLAIRs depuis la VHF-PKT-DB
- Construction de notices au format **VOEvent**
- **Diffusion** des notices par protocole VTP au broker Comet
- **Enregistrement** des notices reçues par le broker dans une voeventdb dédiée



## VOEvents

- Webpage associée à la voeventdb

SVOM | NOTICES WEBPAGE

Table
Creator

**Query tools**

Basic query ▾

Instrument  
All ▾

Date  
All ▾

Role  
Observation ▾

Reset Submit

Pattern search ▾

Cone search ▾

### Observation Notices

DATE ▾	IVORN	ROLE	WHEN	WHERE	ERROR
2020-01-15 13:35:22	<a href="#">ivo://org.svomeclairs#2067-01-15_1.1</a>	observation	2067-01-15 13:33:49	<a href="#">00h07m17.1701s -00d59m00.5166s</a>	4.32039607226106e-05
2020-01-15 10:06:27	<a href="#">ivo://org.svomeclairs#2020-01-09_1.1</a>	observation	2020-01-09 14:59:25	<a href="#">23h48m04.5345s -00d55m09.9731s</a>	0.0

Showing 1 to 2 of 2 rows

J2000 ▾
00 07 17.170 -00 59 05.2

FoV: 177.08°
ALADIN



## Pipeline OBLC\_ECL (w/ Tatyana)

- Pipeline déployé au LAL
- Interface REST compatible avec l'orchestration (#pipeline-bricks)
- Déclenchement de traitement **par l'orchestrateur**
- **Récupération** des paquets VHF light curve ECLAIRs depuis la VHF-PKT-DB
- Construction du produit **OBLC\_ECL** (format fits)
- **Enregistrement** du produit dans la SDB





## Travaux effectués :

- Générateur de code d'encodage/décodage de paquets VHF
- Maintenance de la base de données .cads pour le segment sol
- Aide à la définition des paquets VHF VT
- Simulateur de stations VHF
- Insertion de contenu scientifique pour certains paquets ECLAIRs de la séquence VHF DC-1



## VHF

### > Définition de la séquence DC-1

Step	Step Description	Expected Result
Retrieve the interface file from CNES	The interface file from CNES defines the sequence with one packet every 2 seconds.	All packets in the sequence are defined. CNES inserted the information relative to the time, the quaternions and the satellite position. In addition, CNES inserted the MXT scientific information in the packets: MXTPOSITO and MXTPHOTONL
Insert scientific content (1/3)	Insertion by CEA of the scientific content in the ECLAIRS packets	The ECLAIRs scientific information is inserted in the packets: ECLALERT1, ELLLCURHIP, ELLLCURLP1, ECLSUBIMAG
Insert scientific content (2/3)	Insertion by CEA of the scientific content in the GRM packets	The GRM scientific information is inserted in the packets: GRMLCURHIP, GRMLCURLOP
Insert scientific content (3/3)	Insertion by CEA of the scientific content in the VT packets	The VT scientific information is inserted in the packets: VTATTCHART, VTFVHARTR, VTFCHARTB, VTSUB1ALLR, VTSUB1ALLB

### > Simulation stations VHF

Step	Step Description	Expected Result
Start the sequence		The packets start to be processed by the VHF data manager
Stop the sequence		All VHF packets have been processed by the VHF data manager

## Pipelines orchestrables

### ➤ Préprocessing light curve (produit OBLC\_ECL)

Step		Step Description	Expected Result
The orchestrator launches the OBLC pipeline	✓	DataNotification on ECL LC is received on NATS queue data.vhf. A ProcessingRequest is sent to the OBLC pipeline.	The OBLC pipeline is launched. Can be verified NATs on activity.* queue as ProcessingReport running.
The OBLC pipeline generates an OBLC_ECL product	✓	The OBLC pipeline retrieves the data from VHFMR, completes the processing and generate the OBLC_ECL products with the available input packets.	A ProcessingReport complete can be checked in activity queue.
The OBLC_ECL product is stored	✓	The OBLC_ECL pipeline imports the new product in the SDB	A new LC product is available in SDB. Can be verified NATs on data.sdb

### ➤ Génération/diffusion de notices VOEvent

Step		Step Description	Expected Result
The orchestrator launches the notices pipeline	✓	DataNotification on ECL Alert is received on NATS queue data.vhf. A ProcessingRequest is sent to the notices pipeline.	The Notices pipeline is launched. Can be verified NATs on activity.* queue as ProcessingReport running.
The notices pipeline generates a VOEvent notice.	✓	The ECL Alert data is retrieved from VHFMR. The VOEvent notice is generated.	A ProcessingReport complete can be checked in activity queue.
The VOEvent notice is broadcasted	✓	The pipeline sends the notice for broadcasting to the Comet service.	The VOEvent is visible in Comet web page.

## Sonarqube DC-1 quality gates

- › *orchestra* → *PASSED*
- › *vhf-simulator* → *PASSED*
- › *vhf-products* → *PASSED* (OBLC\_ECL pipeline, w/ Tatyana)
- › *messaging* → *PASSED*
- › *packets-decoder* → *PASSED*
- › *notices* → **FAILED (taux de couverture 30%)**
  
- › **Sur l'ensemble de ces projets, la CI est mise en place et fonctionnelle, incluant pylint, sonar-scanner, construction et déploiement des images docker**