



Irfu - FSC contribution to DC-2

M. BOCQUIER, A. CLARET, B. CORDIER, A. FORMICA, J-P. LEFEVRE,
H. LOUVIN, T. SADIBEKOVA, A. SAUVAGEON, K. TAZHENOVA, E. TRIGUI

KP-4, December 2020



Irfu-FSC projects

	A	B	C	D	E	F	G	H	I	J
1	Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
2	common / auth		😊	😊	😊	😊	😊	😊	😊	7
3	common / pipeline-bricks		😊	😊	😊	😊	😊	😊	😊	7
4	ecmgr / eclmgr-vhf		😊	😊	😊	😊	😊	😊	😊	7
5	messaging		😊	😊	😊	😊	😊	😊	😊	7
6	notices		😊	😊	😊	😊	😊	😊	😊	7
7	orchestra		😊	😊	😊	😊	😊	😊	😊	7
8	packets-decoder		😊	😊	😊	😊	😊	😊	😊	7
9	vhf / vhf-preprocessing		😊	😊	😊	😊	😊	😊	😊	7
10	vtac		😊	😊	😊	😊	😊	😊	😊	7
11	xband / xband-manager		😊	😊	😊	😊	😊	😊	😊	7
12	xband / xband-recovery		😊	😊	😊	😊	😊	😊	😊	7
13	mxt / mxt-ql		😊	😊	😊	😊	😊	😊	😊	7
14	too-mm		😊	😊	😊	😊	😊	😊	😊	7
15	mxt / mxt-xvhf-cp		😊	😊	😊	😊	😊	😊	😊	7
16	MIC / mic-server		😊	😊	😊	😊	😊	😊	😊	7
17	auth-ref-impl		😊	😊	😊	😊	😊	😊	😊	7
18	ifsc-tools		😊	😊	😊	😊	😊	😊	😊	7
19	mxt / mxt-xband-gp		😊	😊	😊	😊	😊	😊	😊	7
20	sdb / export-client		😊	😊	😊	😊	😊	😊	😊	7
21	sdb / api-import		😊	😊	😊	😊	😊	😊	😊	7
22	xband / xband-webui		😊	😊	😊	😊	😊	😊	😊	7
23	xband / l0d-recovery		😊	😊	😊	😊	😊	😊	😊	7
24	colibri-interface		😊	😊	😊	😊	😊	😊	😊	7
25	common / caldb		😊	😊	😊	😊	😊	😊	😊	7
26	vhf / vhfmgr		😊	😊	😊	😊	😊	😊	😊	6
27	common / crest		😊	😊	😊	😊	😊	😊	😊	5
28	common / pipeline-maker		😊	😊	😊	?	?	😊	😊	5
29	ecmgr / eclmgr-monitor		😊	😞	😊	😊	😊	😊	😊	5
30										
31	vhf / vhf-simulator		😊	😊	😊	😊	😊	?	?	5
32	common / vo-annotations		😊	😊	😊	😊	😊	?	?	5
33	eclairs-gp		😊	😊	😊	😊	😊	😊	😊	5
34			Not accepted							4
35	common / heasoft		😊	😞	😊	😊	?	😊	😊	3
36	common / monitoring-bricks		😊	😊	😊	😊	😊	😊	😊	3
37	common / packet-header-decoding		😊	😊	😊	😊	😊	?	?	3
38	common / pipeline-broker		😊	😊	😊	😊	😊	?	?	3

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

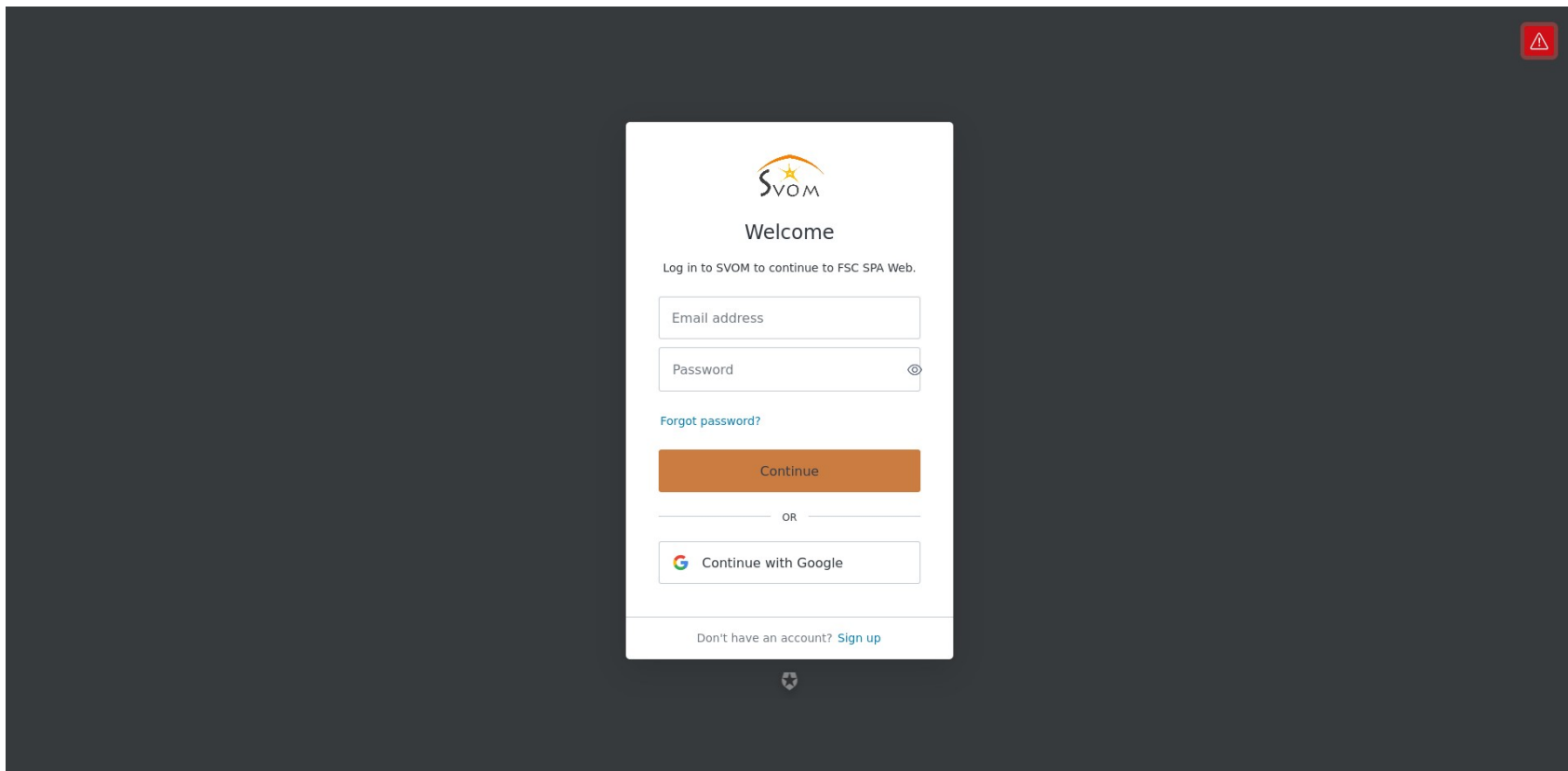


Infrastructure

M. BOCQUIER, A. FORMICA, J-P. LEFEVRE,
H. LOUVIN

FSC Authentication

- JWT for a human user connecting to FSC through a webUI accessible from internet (essentially <https://fsc.svom.org/>)
- Certificates for external services connecting to FSC through proxy.svom.org



The screenshot shows a login interface for SVOM. At the top center is the SVOM logo, which consists of a stylized orange and yellow 'S' with a star above it. Below the logo is the word 'Welcome' in bold. Underneath is the instruction 'Log in to SVOM to continue to FSC SPA Web.' There are two input fields: 'Email address' and 'Password'. The 'Password' field has a small eye icon to its right. Below the password field is a link that says 'Forgot password?'. A large orange button labeled 'Continue' is positioned below the links. Below this button is a horizontal line with the word 'OR' in the center. Underneath the line is a button with the Google logo and the text 'Continue with Google'. At the bottom of the form is a link that says 'Don't have an account? Sign up'. The entire form is centered on a dark gray background. In the top right corner of the background, there is a small red square icon with a white triangle inside.



FSC authentication status on 15-12-2020

- Client certificates delivered : **22**
 - 8** to entities/laboratories
 - 14** to users
- Auth0 user accounts for tenant `svom` : **25** (+2 dedicated to DC-2 tests)
- **7** API connected to the proxy `proxy.svom.org` :
 - SDB import, CalDB, CREST-DB, VHF-DB, SDB (web), eclgrm-monitor
- **4** UIs connected to the proxy `fsc.svom.org` without implemented auth :
 - VHF-DB, CREST-DB, eclgrm-monitor, EIC data, XBAND
- **6** UIs connected to the proxy `fsc.svom.org` with partially implemented auth :
 - NATS, orchestra, notices, inspector, MXT XBAND GP



KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
common / auth		😊	😊	😊	😊	😊	😊	😊	7
auth-ref-impl		😊	😊	😊	😊	😊	😊	😊	7

[auth-ref-impl](#)

Passed

Last analysis: December 15, 2020, 3:27 PM

0 A

🐛 Bugs

0 A

🔒 Vulnerabilities

0 A

☢️ Code Smells

90.5%

Coverage

0.0%

Duplications

1.1k S

TypeScript, Python

[auth](#)

Passed

Last analysis: December 1, 2020, 8:54 PM

0 A

🐛 Bugs

0 A

🔒 Vulnerabilities

0 A

☢️ Code Smells

94.8%

Coverage

0.0%

Duplications

719 XS

Python

Messaging status

- messaging stack at FSC + webUI with auth (<https://fsc.svom.org/nats/>)
- Python module `svom.messaging` for communication with NATS, VHFDB, SDB, CREST, taking into account latest FSC authentication implementation



The screenshot shows the NATS streaming server webUI. At the top, there's a header with the SVOM logo and the text "NATS streaming server". Below the header, there's a navigation bar with "Channel(s):" followed by tabs: "ALL", "data", "activity", "science", and "test". The "ALL" tab is selected. The main content area displays a list of messages received. The first message is a JSON object with fields: "creation_date", "version", "links", "activity", "date", "status", and "info". The second message is a JSON object with fields: "date", "message", and "message". The third message is a JSON object with fields: "date", "message", and "message". The fourth message is a JSON object with fields: "date", "message", and "message". The fifth message is a JSON object with fields: "date", "message", and "message". The sixth message is a JSON object with fields: "date", "message", and "message". The seventh message is a JSON object with fields: "date", "message", and "message". The eighth message is a JSON object with fields: "date", "message", and "message". The ninth message is a JSON object with fields: "date", "message", and "message". The tenth message is a JSON object with fields: "date", "message", and "message". The eleventh message is a JSON object with fields: "date", "message", and "message". The twelfth message is a JSON object with fields: "date", "message", and "message". The thirteenth message is a JSON object with fields: "date", "message", and "message". The fourteenth message is a JSON object with fields: "date", "message", and "message". The fifteenth message is a JSON object with fields: "date", "message", and "message". The sixteenth message is a JSON object with fields: "date", "message", and "message". The seventeenth message is a JSON object with fields: "date", "message", and "message". The eighteenth message is a JSON object with fields: "date", "message", and "message". The nineteenth message is a JSON object with fields: "date", "message", and "message". The twentieth message is a JSON object with fields: "date", "message", and "message".

```

{
  "creation_date": "default",
  "version": "default",
  "links": [
    "default"
  ]
},
{
  "activity": "default",
  "date": "default",
  "status": "wait",
  "info": "default"
}
}
2020-12-16T08:36:40.064087 [science.general_program] {
  "Response.get_serv_descriptor()": {
    "service_descriptor": {
      "name": "Server HTTP ECLAIRs General Program",
      "instrument": [
        "ECLAIRs"
      ],
      "creation_date": "Tue Dec 15 15:54:01 2020",
      "version": "DC-2"
    }
  },
  "date": "2020-12-16T08:36:40",
  "message": "Server status: OK"
}
2020-12-16T08:37:40.088961 [science.general_program] {
  "Response.get_serv_descriptor()": {
    "service_descriptor": {
      "name": "Server HTTP ECLAIRs General Program",
      "instrument": [
        "ECLAIRs"
      ],
      "creation_date": "Tue Dec 15 15:54:01 2020",
      "version": "DC-2"
    }
  },
  "date": "2020-12-16T08:37:40",
  "message": "Server status: OK"
}
}

```

Messages received since Wed Dec 16 08:38:38 2020 (UTC)



KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
messaging		😊	😊	😊	😊	😊	😊	😊	7

[messaging](#)

Passed

Last analysis: December 15, 2020, 3:55 PM

0 
🐛 Bugs

0 
🔒 Vulnerabilities

0 
⚙️ Code Smells

 81.4%
Coverage

 0.0%
Duplications

1.4k 
Python



Orchestrator

- **Automated pipelines triggering** upon reception of NATS messages notifying the availability of new data from VHF-DB, XDB or SDB
- **OBS_ID use** to gather data, and OBS_ID transmission in processing requests to allow pipeline to retrieve input data
- **Monitoring of ongoing processings** and multiple tries of failed ones
- **Storage of processings**, their status and their logs in database
- Offers **manual processing request** from the orchestrator webUI
- **BURST_ID** enabled (vhf-preprocessing pipeline only for now)



Orchestrator status on 17-12-2020

- orchestra stack with webUI
- 8 pipelines, 16 modes, 33 CP products, 3 GP products, 2 types of VOEvent notices

vhfpreproc:

```
vhfpreproc_oblc_ecl: OBLC_ECL,  
vhfpreproc_oblc_grm: OBLC_GRM,  
vhfpreproc_obtloc_ecl: TT_ECL, QPO_ECL, QCL_ECL, OBALERT_ECL,  
vhfpreproc_obtloc_grm: TT_GRM, QPO_GRM, QCL_GRM, OBALERT_GRM,  
vhfpreproc_obloc_mxt: OBPOS_MXT, OBPHOTL_MXT, OBPHOTMM_MXT,  
vhfpreproc_obloc_vt: OBATT_VT, OBFIND_VT, OBIM1B_VT
```

eclgrm:

```
qtemp_hr_ecl: QLC_ECL, QT90_ECL, QPF_ECL, QHR_ECL  
qtemp_hr_grm: QLC_GRM, QT90_GRM, QPF_GRM, QHR_GRM  
qhr_eclgrm: QHR_ECLGRM
```

mxtql:

```
ql_mxt: DT_MXT, QPO_MXT, QF_MXT, QLC_MXT, QTI_MXT
```

mxt_pseudox:

```
xvhfcp_mxt: QIMAGE_MXT, QSRCLIST_MXT
```

vtac:

```
qpo_vt: QPO_VT
```

notices:

```
n1_notice: N1 VOEvent notices (ECLAIRS localization)  
n1b_notice: N1b VOEvent notices (GRM localization)
```

ld_recovery:

```
xband_l1: SVO-???-EVT
```

mxt_pipeline:

```
xbandgp_mxt: MXT-EVT-CAL, MXT-SKY-IMA
```



WebUI

- Real-time content of the orchestrator DB is visible on a webUI connected to the FSC proxy, with access control and permission handling :
<https://fsc.svom.org/orchestra-web/> (avec authentification et permission)

The screenshot displays the SVOM orchestrator webUI. The top navigation bar includes the SVOM logo, the word 'orchestrator', and links for 'PROCESSINGS', 'GRAPH', and 'HL'. A left sidebar contains a 'Utilities' section with a 'Filter Data' dropdown and input fields for 'Date' (set to 'All time'), 'Target', 'BurstID', and 'ObsID'. Below these are radio buttons for 'Status' with options: running, queued, failed, complete, and vanished. The main content area features a search bar and a table of process data. The table has columns for PROCESS ID, DATE, TARGET, BURSTID, OBSID, STATUS, and URL. It lists five processes, with one marked as 'failed'. Each row includes eye and trash icons for actions. At the bottom right, there are pagination controls showing '12' items per page and '1' page.

PROCESS ID	DATE	TARGET	BURSTID	OBSID	STATUS	URL
svom_notices_pipeline_784816ee-3ef0-11eb-afff-02420a000206	2020-12-15T16:12:16	N1_NOTICE	sb20120700	2566920048	complete	http://notices_creator:5000/api
svom_notices_pipeline_7536e412-3ef0-11eb-be93-02420a000206	2020-12-15T16:12:12	N1_NOTICE	sb20120700	2566920048	complete	http://notices_creator:5000/api
svom_notices_pipeline_6b7a41b2-3ef0-11eb-a84a-02420a000206	2020-12-15T16:12:08	N1_NOTICE	sb20120700	2566920048	complete	http://notices_creator:5000/api
svom_vtac_pipeline_e4d20eec-3eef-11eb-9efd-02420a000213	2020-12-15T16:09:19	QPO_VT	sb20120700	2566920048	failed	http://vtac_pipeline:5000/api
svom_notices_pipeline_057ef0ba-3ef0-11eb-8202-02420a000206	2020-12-15T16:09:13	N1_NOTICE	sb20120700	2566920048	complete	http://notices_creator:5000/api



KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	
orchestra		😊	😊	😊	😊	😊	😊	😊	7
common / pipeline-maker		😊	😊	😊	?	?	😊	😊	5

- The `pipeline-maker` project contains only Dockerfiles : no gitlab-ci, nor Sonarqube

orchestra Passed

Last analysis: December 15, 2020, 4:44 PM

0 A
🐛 Bugs

0 A
🔒 Vulnerabilities

0 A
💩 Code Smells

85.6%
Coverage

0.0%
Duplications

1.7k S
Python, JavaScript



VHF

B. CORDIER, A. FORMICA, J-P. LEFEVRE,
H. LOUVIN, T. SADIBEKOVA, K. TAZHENOVA



DC-2 scenario (launched ONCE):

- Full orbit simulation (~1h30) with one VHF packet sent every 2 seconds
- Takes into account recurrent packets, packet priorities, holes and repetitions
- GRB detected by ECLAIRs 10 minutes after scenario start
- Slew requested in the 7th ECLAIRs alert packet
- Burst parameters:
 - `OBS_ID=2566914115` (meaning `OBS_TYPE=153` and `OBS_NUM=67`)
 - `TB0=2021-01-30T10:38:49.97`
 - Expected `BURST_ID=sb21013044`
- Scientific content for the following packets :
 - ECLAIRs alert, ECL and GRM light curves, MXT positions, MXT photon list
- **WARNING: NO VT PACKETS (removed due to heavy modifications in the packets definitions)**



ECLAIRs-first scenario (multiple launches):

- Simplified scenario : VHF packets with scientific content but outside of a realistic simulation (no priorities nor repetitions, but holes can be simulated):
 - ECLAIRs alerts (7 packets)
 - ECLAIRs light curve (SBP simulation, 64 packets)
 - GRM light curve (SBP simulation, 64 packets)
 - MXT position (48 packets)
 - MXT photons (10 packets)
 - High latitude VT alert sequence (529 packets)
 - VT attitude chart (24 packets)
 - VT finding chart (258 packets)
 - VT subimage (247 packets)
- Burst parameters :
 - `OBS_TYPE=153` (ECLAIRs) and `OBS_NUM` to be defined each launch
 - `TB0` and `BURST_ID` computed so that they coincide with the date and time of the first alert packet being sent



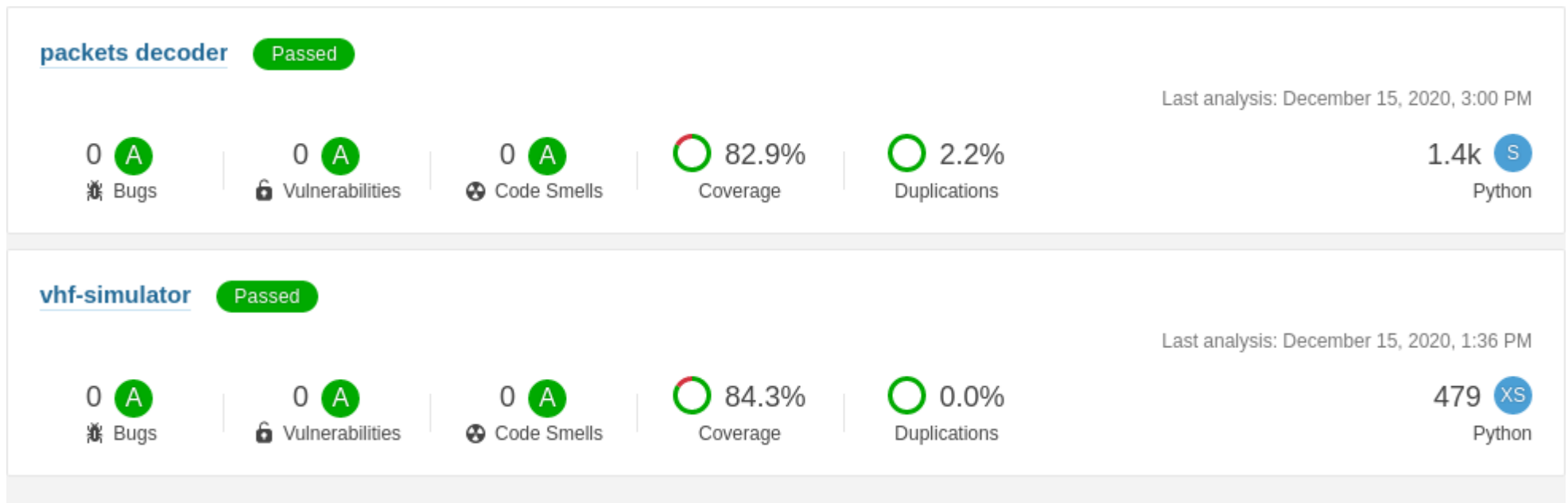
GRM-first scenario (multiple launches):

- Simplified scenario :
 - GRM trigger (1 packet)
 - GRM light curve (Goldstein simulation, 64 packets)
 - ECLAIRs alerts (7 packets)
 - ECLAIRs light curve (Goldstein simulation, 64 packets)
 - MXT position (48 packets)
 - MXT photons (10 packets)
 - Low latitude VT alert sequence (1014 packets)
 - VT attitude chart (24 packets)
 - VT finding chart (519 packets)
 - VT subimage (471 packets)
- Burst parameters :
 - For GRM packets `OBS_TYPE=231` (GRM) and variable `OBS_NUM`
 - For others `OBS_TYPE=153` (ECLAIRs) and `OBS_NUM` to be defined each launch
 - `TB0` and `BURST_ID` corresponding to the time of the GRM trigger simulation

KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
packets-decoder		😊	😊	😊	😊	😊	😊	😊	7
vhf / vhf-simulator		😊	😊	😊	😊	😊	?	?	5

- Simulations will be launched by hand from a machine outside of FSC
- The `vhf-simulator` is therefore not dockerized



VHF data flow

- **Encoded VHF packets reception** as hexadecimal strings through the REST API of the `vhfmgr` service
- Packet decoding using the `packets_decoder` service REST API to get packet content in **JSON format**
- Data insertion in **VHF-DB**
- Broadcasting of **NATS messages** notifying the availability of data (mostly for orchestrator use)
- NATS B.A. monitoring message for `ifsctools`

Vhfmgr status on 17-12-2020

- Duplication handling (repeated packets or same packet received from multiple VHF stations)
- BURST_ID creation using the burst time retrieved from the first received alert packet
- BURST_ID assignment to all alert sequence packets
- BURST_ID and OBS_ID broadcast in NATS notifications



KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
vhf / vhfmgrr		😊	😊	😊	😊	😊	😊	😊	6

[vhfmgr](#)

Passed

Last analysis: June 29, 2020, 12:17 PM

0 A
🐛 Bugs

0 A
🔒 Vulnerabilities

0 A
💩 Code Smells

80.7%
Coverage

0.3%
Duplications

7.9k S
Java



Data flow

- Processing triggered **by the orchestrator**
- **BURST_ID use** to distinguish VHF data
- **VHF packet retrieval** from the VHF-DB (JSON format)
- **Data model** retrieval from the SDB for associated products
- **FITS products creation** using the latest data model version
- **SDB import or update**
 - New import if the same product doesn't exist in the SDB for the same OBS_ID or BURST_ID
 - Product update otherwise



Pipeline status on 17-12-2020

- Dynamic data model using always the **latest product description** available from the SDB
- **16 products** handled currently for all mission instruments
 - « on-board » analysis : OBLC_ECL, OBLC_GRM, OBALERT_ECL, OBALERT_GRM, OBPOS_MXT, OBPHOTL_MXT, OBATT_VT, OBFIND_VT, OBIM1B_VT, OBPHOTMM_MXT (pas de données VHF)
 - Trigger parameters : TT_ECL, TT_GRM, QPO_ECL, QPO_GRM, QCL_ECL, QCL_GRM
 - Intended for 2021 : OBALERTD_ECL, OBSP_GRM, OBSUBIM_ECL, OBSHDWGRM_ECL



VHF-preprocessing pipeline

KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
vhf / vhf-preprocessing		😊	😊	😊	😊	😊	😊	😊	7

[vhf products](#)

Passed

Last analysis: December 16, 2020, 10:15 AM

0 **A**
🐛 Bugs

0 **A**
🔒 Vulnerabilities

2 **A**
💩 Code Smells

94.0%
Coverage

0.0%
Duplications

1k **S**
Python



« N1 » notices pipeline

- Pipeline+broker deployed at FSC (`notices` stack)
- Processing triggered **by orchestrator**
- Retrieval of QPO_ECL, QCL_ECL and TT_ECL products **from SDB**
- Generation of notice in **VOEvent format**
- **Broadcast** of notices using VTP protocol on a Comet broker located within FSC
- **Storing** of broadcasted notices in a dedicated `voeventdb`



« N1b » notices pipeline

- Pipeline+broker deployed at FSC (`notices` stack)
- Processing triggered **by orchestrator**
- Retrieval of QPO_GRM, QCL_GRM and TT_GRM products **from SDB**
- Generation of notice in **VOEvent format**
- **Broadcast** of notices using VTP protocol on a Comet broker located within FSC
- **Storing** of broadcasted notices in a dedicated `voeventdb`



WebUI

Real-time content of the voeventdb is visible on a webUI connected to the FSC proxy, with access control: <https://fsc.svom.org/notices/>

VOEvent notices table

UTILITIES

BASIC FILTERS

DATE

All time

CONE

RA

Dec

rad.

PATTERN

ROLE

☒ observation

☐ utility

☐ test

Submit

ALL FILTERS

TABLE

CREATOR

HL

Search

AUTHORED	IVORN	ROLE	RA	DEC	ERR	TIME
2020-12-15T16:10:34	ivo://org.svom/fsc#sb20120700_eclairs-7	observation	51.92	46.66	8.33	2020-12-07T00:09:16

6

1

32000

03 27 41.381 +46 30 35.15

IVORN

RA

DEC

ERR

TIME

ivo://org.svom/fsc#sb20120700_eclairs-7

51.9224212970037

46.6605941357851

8.3283149384234

2020-12-07T00:09:16.478000+00:00



KP-4 delivery

Project	Tag	Naming	Readme	Directories	Gittlab-ci	Sonarqube	Docker conf	Docker doc	
notices		😊	😊	😊	😊	😊	😊	😊	7
									7

[notices](#)

Passed

Last analysis: December 15, 2020, 12:18 PM

0 
🐛 Bugs

0 
🔒 Vulnerabilities

0 
💩 Code Smells

 84.2%
Coverage

 0.0%
Duplications

660 
Python



X-Band

A. CLARET, A. FORMICA, J-P. LEFEVRE,
A. SAUVAGEON, E. TRIGUI



Scientific scenario

- GP source observation during 1.6h
- Observation interruption following GRB detection
- Slew execution
- GRB observation sequence for 24h
- Resumption of the interrupted observation, and observation for another 24h
- Start of a new GP source observation during 1.6h



PP-1 data flow (L0C → L0D)

- Check every 15 minutes for the arrival of new **L0C data** on the FTP server <ftp://software.svom.fr> by the `xband-recovery` service
- **Triggering of the xband-manager service** if new data is available
- Ingestion of the data into the **XBand-DB** and production of the **L0D products** in FITS format
- Broadcasting of **NATS messages** notifying the availability of data (mostly for orchestrator use)



X-Band processing

PP-2 data flow (L0D→ L1)

- Processing triggered **by orchestrator**
- **Use of the** `tarfilename` to distinguish X-Band data
- **L0D retrieval** from the XDB (FITS format)
- Generation of the associated **L1 product(s)**
- **SDB import**



X-Band processing status on 17-12-2020

- XDB, FTP server and PP-1 operational and deployed at FSC
- Working PP-2 skeleton deployed at FSC (validating interfaces with orchestrator, XDB and SDB)
- PP-2 operational for orbito data (corresponds to L1 FITS products `SVO-ORB-CNV` and `SVO-ATT-CNV`) to be inserted into the PP-2 skeleton in FSC before DC-2
- Other L1 products will be copied inside the PP-2 service image and sent to the SDB during the DC-2 tests after creation and import of `SVO-???-CNV`
 - Science data received from the instruments teams on 16-12-2020
 - Will be formatted into L1 products before 18-12-2020



X-Band processing

KP-4 delivery - 1

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	
xband / xband-manager		😊	😊	😊	😊	😊	😊	😊	7
xband / xband-recovery		😊	😊	😊	😊	😊	😊	😊	7
xband / xband-webui		😊	😊	😊	😊	😊	😊	😊	7
xband / 10d-recovery		😊	😊	😊	😊	😊	😊	😊	7
Not considered									
xband / xband-11-generator		😊	😊	😊	😊	😞	😊	😞	3

xbandManager

Passed

Last analysis: December 14, 2020, 5:27 PM

0 Bugs

0 Vulnerabilities

1 Code Smells

80.9% Coverage

0.6% Duplications

3.9k Java

xband-recovery

Passed

Last analysis: October 12, 2020, 2:40 PM

0 Bugs

0 Vulnerabilities

0 Code Smells

80.3% Coverage

0.0% Duplications

447 Java



X-Band processing

KP-4 delivery - 2

Project	Tag	Naming	Readme	Directories	Gitlab-ci	Sonarqube	Docker conf	Docker doc	7
xband / xband-manager		😊	😊	😊	😊	😊	😊	😊	7
xband / xband-recovery		😊	😊	😊	😊	😊	😊	😊	7
xband / xband-webui		😊	😊	😊	😊	😊	😊	😊	7
xband / l0d-recovery		😊	😊	😊	😊	😊	😊	😊	7
Not considered									
xband / xband-l1-generator		😊	😊	😊	😊	😞	😊	😞	3

ldrecovery

Failed

Last analysis: August 31, 2020, 11:06 AM

0 A
Bugs

0 A
Vulnerabilities

0 A
Code Smells

0.0%
Coverage

0.0%
Duplications

53 XS
Python

l1-generator

Failed

Last analysis: June 23, 2020, 7:38 PM

0 A
Bugs

0 A
Vulnerabilities

2.2k B
Code Smells

0.0%
Coverage

27.6%
Duplications

14k M
C (Community)



Now what ?

M. BOCQUIER, A. FORMICA, J-P. LEFEVRE,
H. LOUVIN

New CC swarm (for DC-2?)

- After CC maintenance and `svomtest` restart issues : migrations to production platform
- Creation of a docker swarm in CC to deploy DC-2 services and test them :
 - How many machines ?
 - How much storage space ? (logs, docker images, DBs, etc.)
 - Which OS ? (no long-term support for CentOS-8)



Continuous production deployment (post-DC-2)

- Once the production swarm is up, what strategy to adopt for continuous deployment ?
- Probably a four-steps process :
 - Image build and IJCLab deployment using Gitlab CI and webhooks
 - Most services/pipelines already send messages to NATS when they are up and running
 - A dedicated service could listen to those messages and launch a test processing request to the newly deployed service
 - In case of test success, the service could be deployed on the production swarm using webhooks
- This requires a production swarm with all configs/secrets/docker-compose
- How to handle changes in configs/secrets/docker-compose ?