The Monitoring and Control Subsystem User Interfaces (MUsIC)

Sonny LION

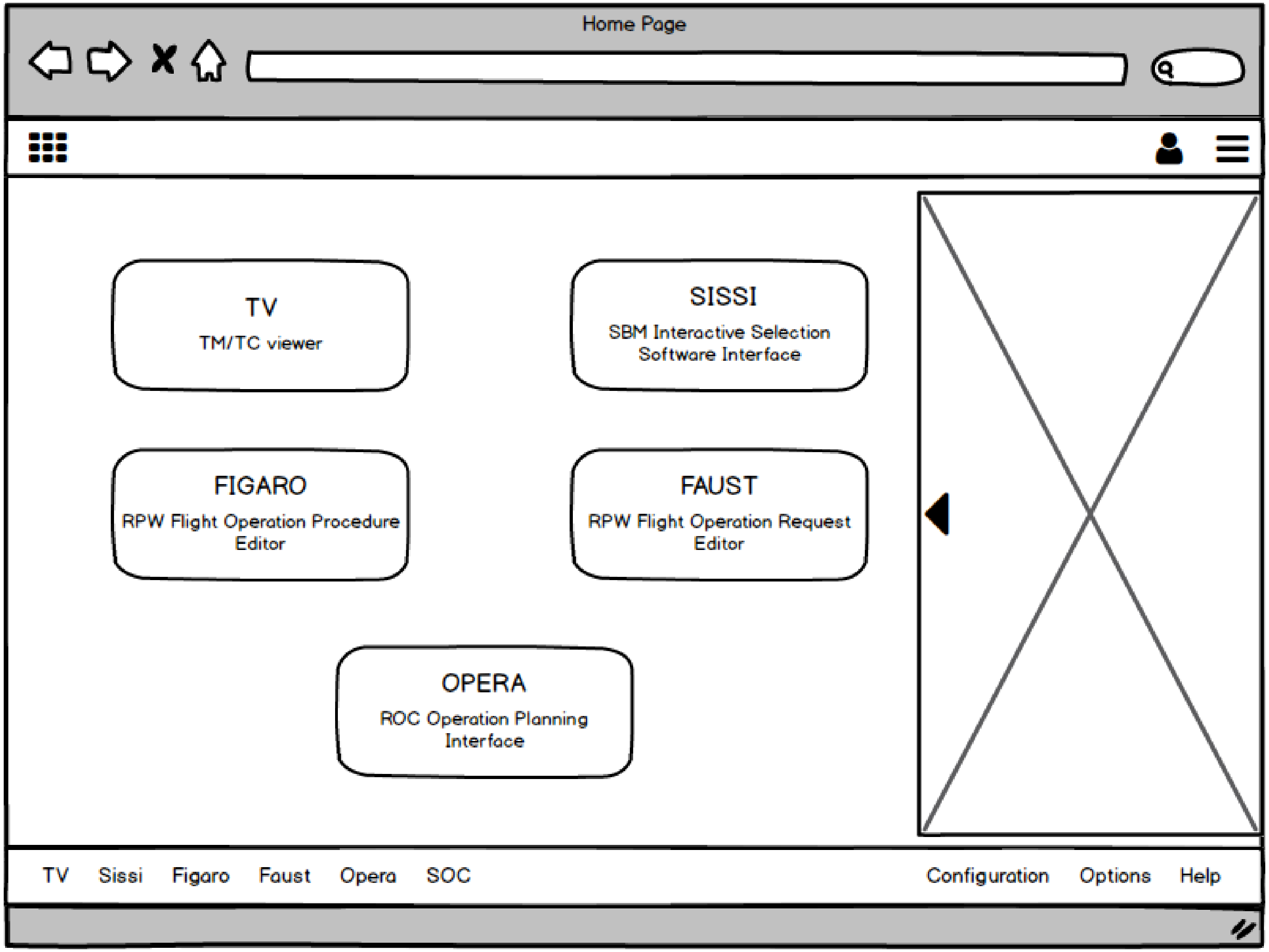
l'Observatoire de Paris LESIA

Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

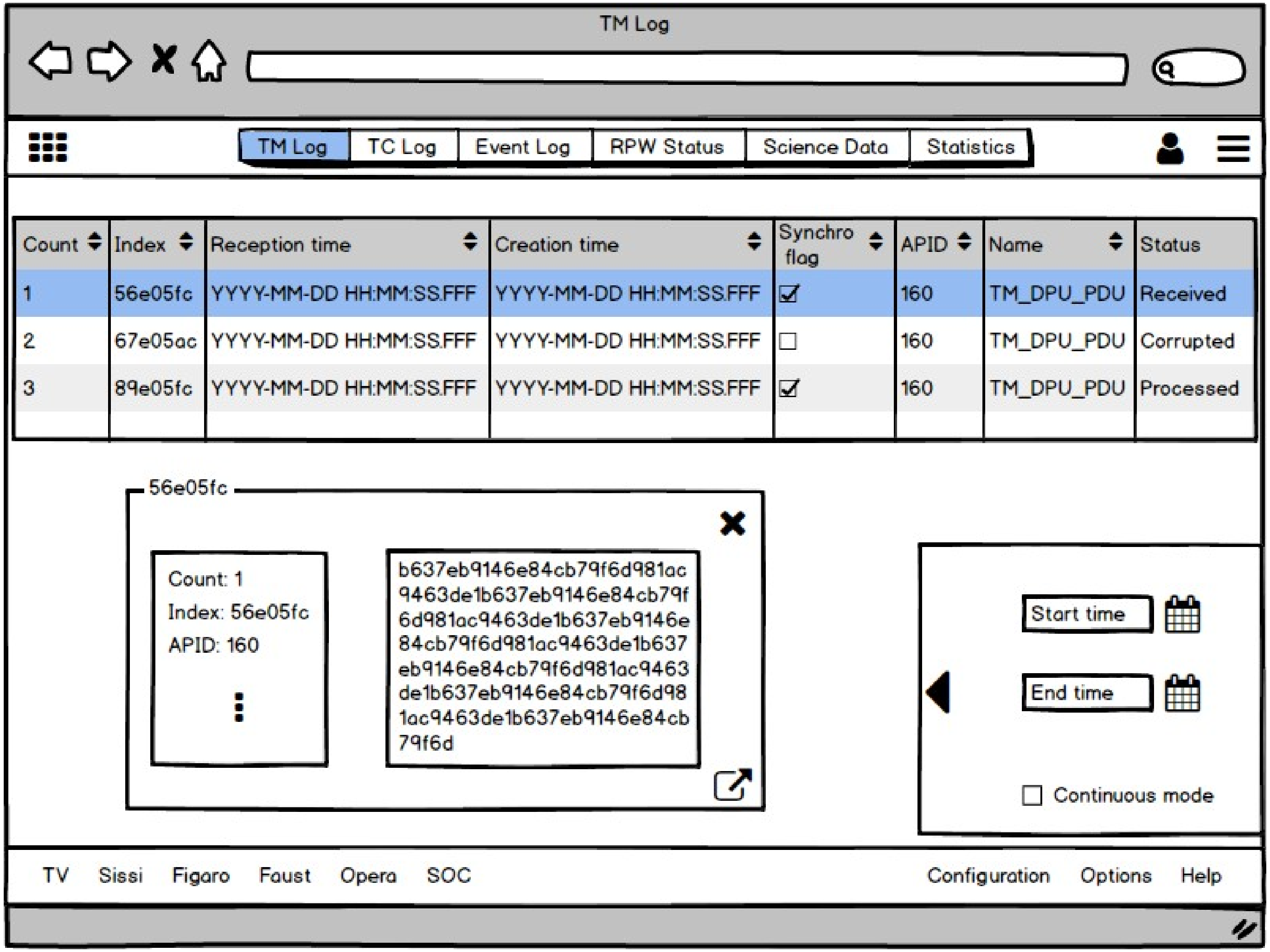
RPW ROC

Introduction

MUsIC is a collection of Graphical User Interfaces (GUIs) dedicated to the preparation of the instrument operations and to the analysis of the instrument data


Home Page
TV
TM/TC viewer
SISSI
SBM Interactive Selection Software Interface
FIGARO
RPW Flight Operation Procedure Editor
FAUST
RPW Flight Operation Request Editor
OPERA
ROC Operation Planning Interface
TV
Sissi
Figaro
Faust
Opera
SOC
Configuration
Options
Help

The RPW TM/TC Viewer , the main GUI to promptly monitor the instrument status, TM/TC history and statistics and HK/science data.



Count	Index	Reception time	Creation time	Synchro flag	APID	Name	Status
1	56e05fc	YYYY-MM-DD HH:MM:SS.FFF	YYYY-MM-DD HH:MM:SS.FFF	☑	160	TM_DPU_PDU	Received
2	67e05ac	YYYY-MM-DD HH:MM:SS.FFF	YYYY-MM-DD HH:MM:SS.FFF	□	160	TM_DPU_PDU	Corrupted
3	89e05fc	YYYY-MM-DD HH:MM:SS.FFF	YYYY-MM-DD HH:MM:SS.FFF	☑	160	TM_DPU_PDU	Processed

56e05fc

Count: 1
Index: 56e05fc
APID: 160
⋮

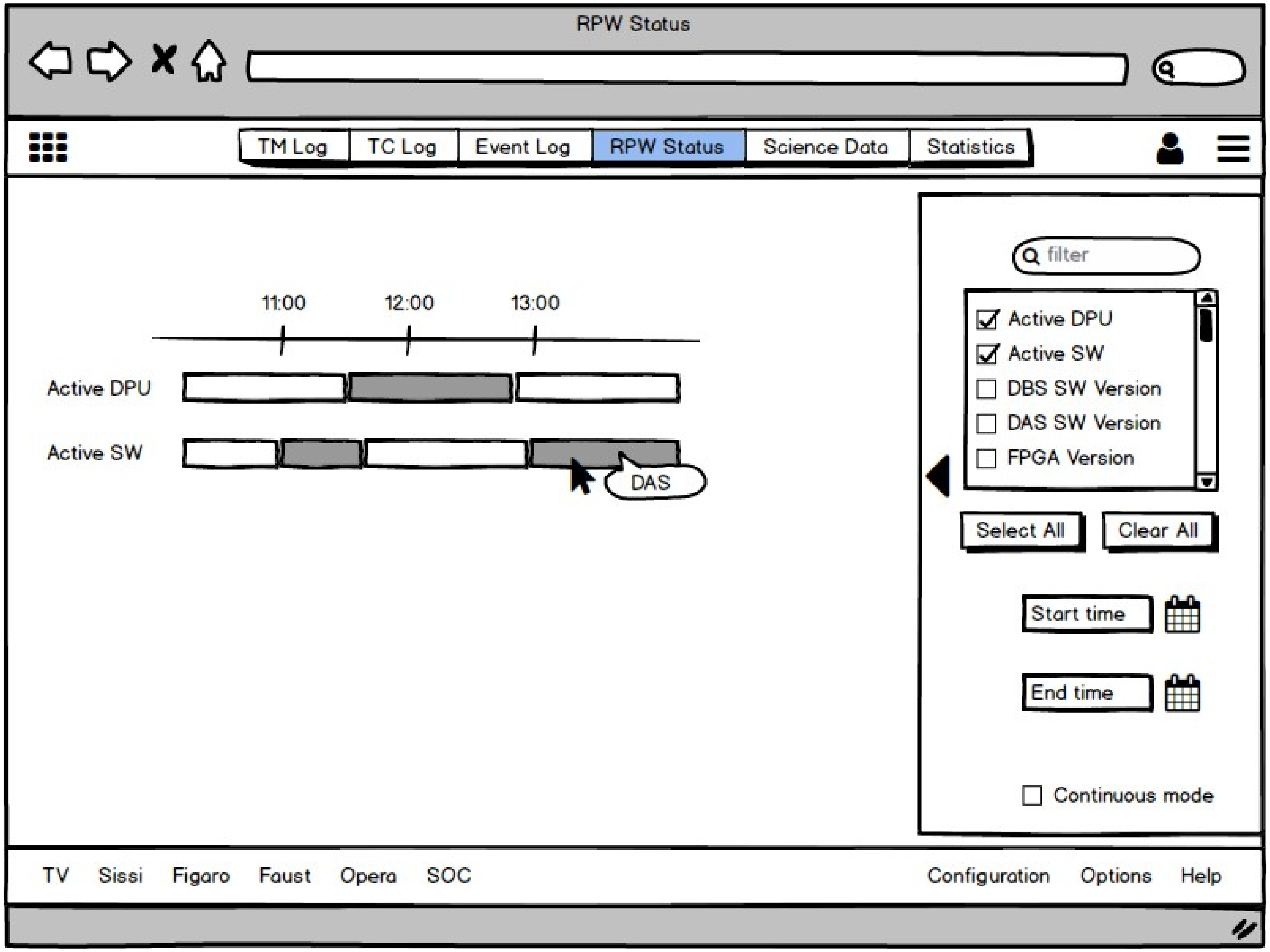
```
b637eb9146e84cb79f6d981ac
9463de1b637eb9146e84cb79f
6d981ac9463de1b637eb9146e
84cb79f6d981ac9463de1b637
eb9146e84cb79f6d981ac9463
de1b637eb9146e84cb79f6d98
1ac9463de1b637eb9146e84cb
79f6d
```

Start time

End time

□ Continuous mode

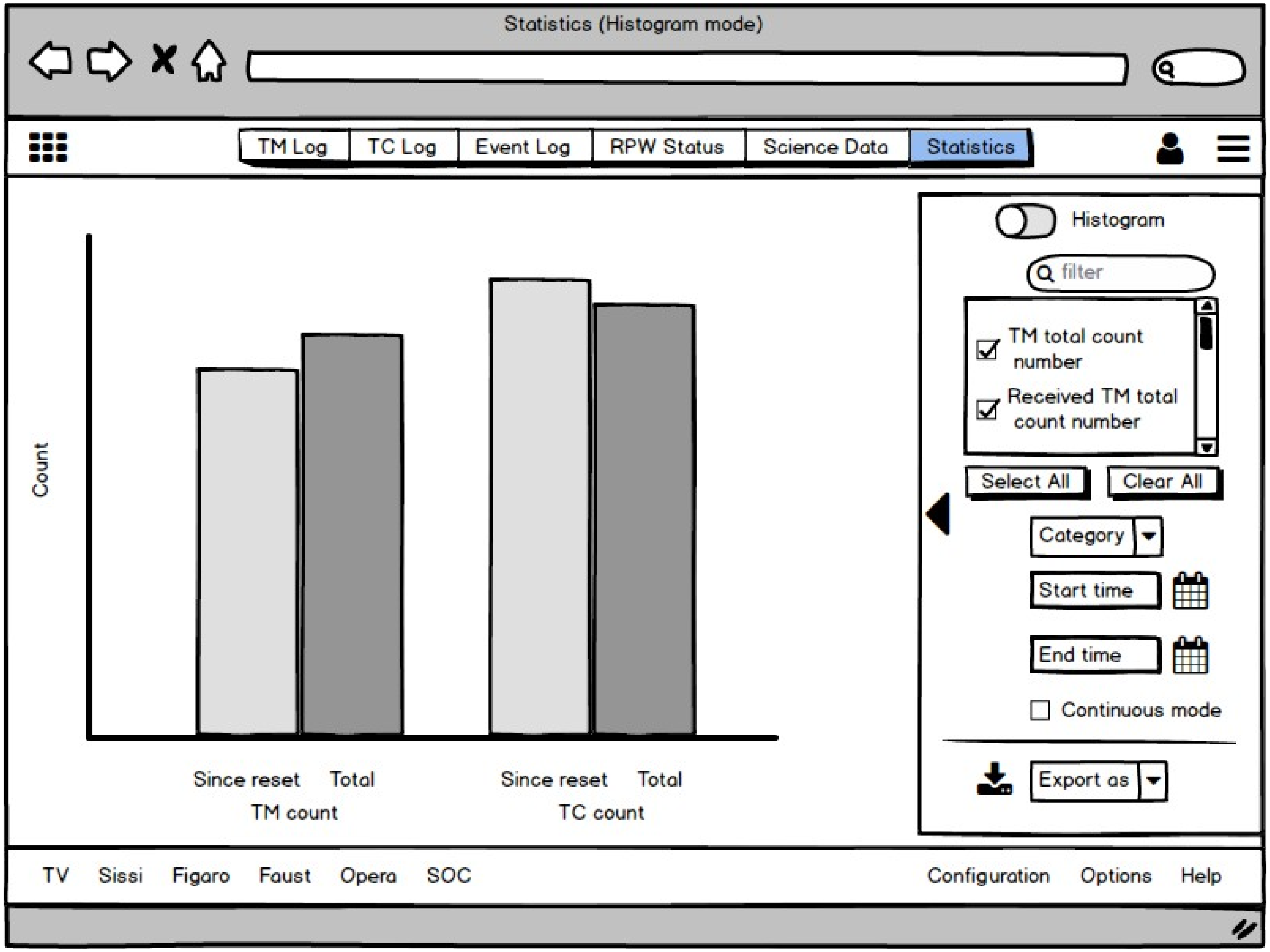
TV - instrument status



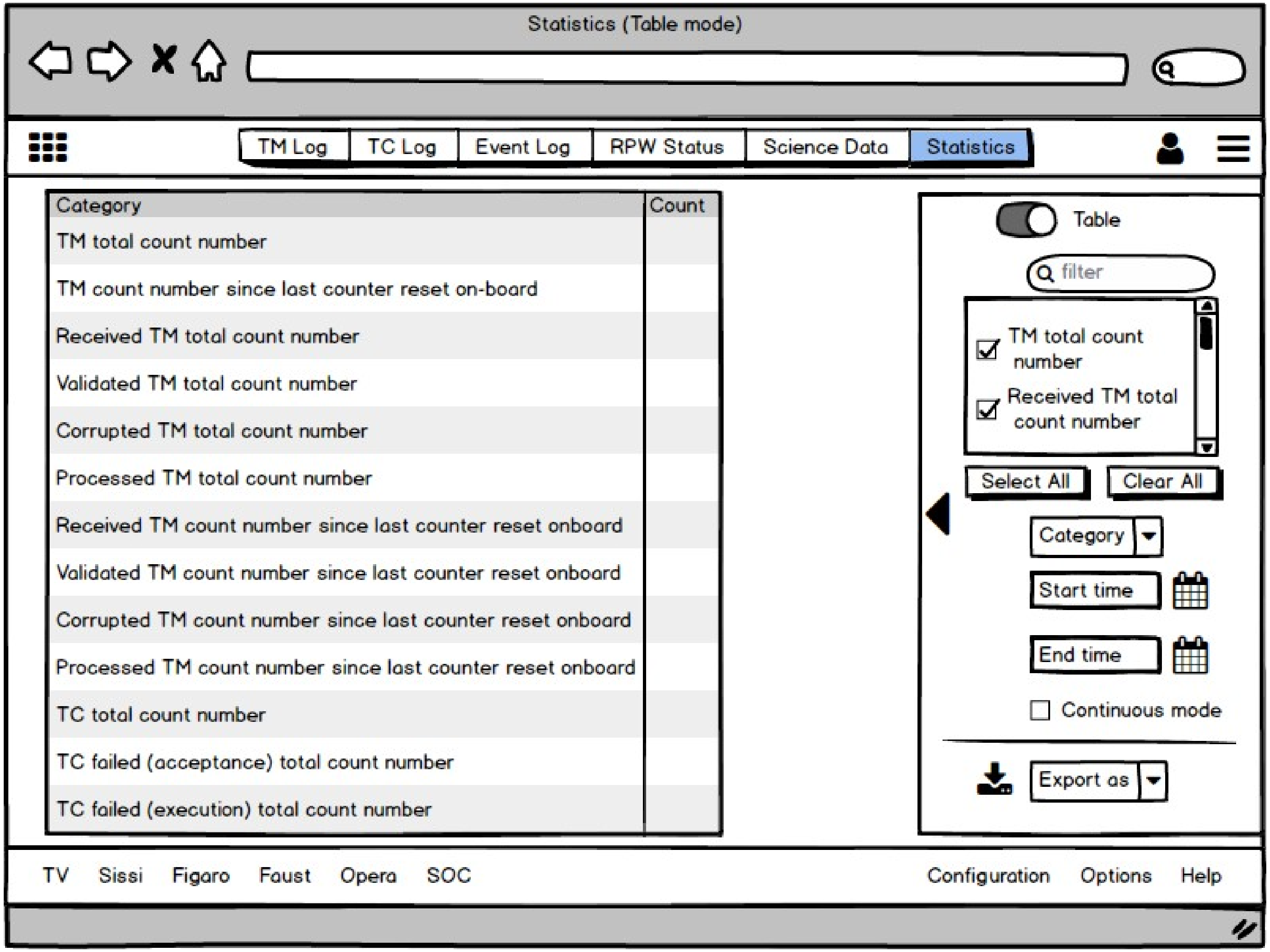
☑ Active DPU
☑ Active SW
□ DBS SW Version
□ DAS SW Version
□ FPGA Version

□ Continuous mode

TV - statistics



TV - statistics



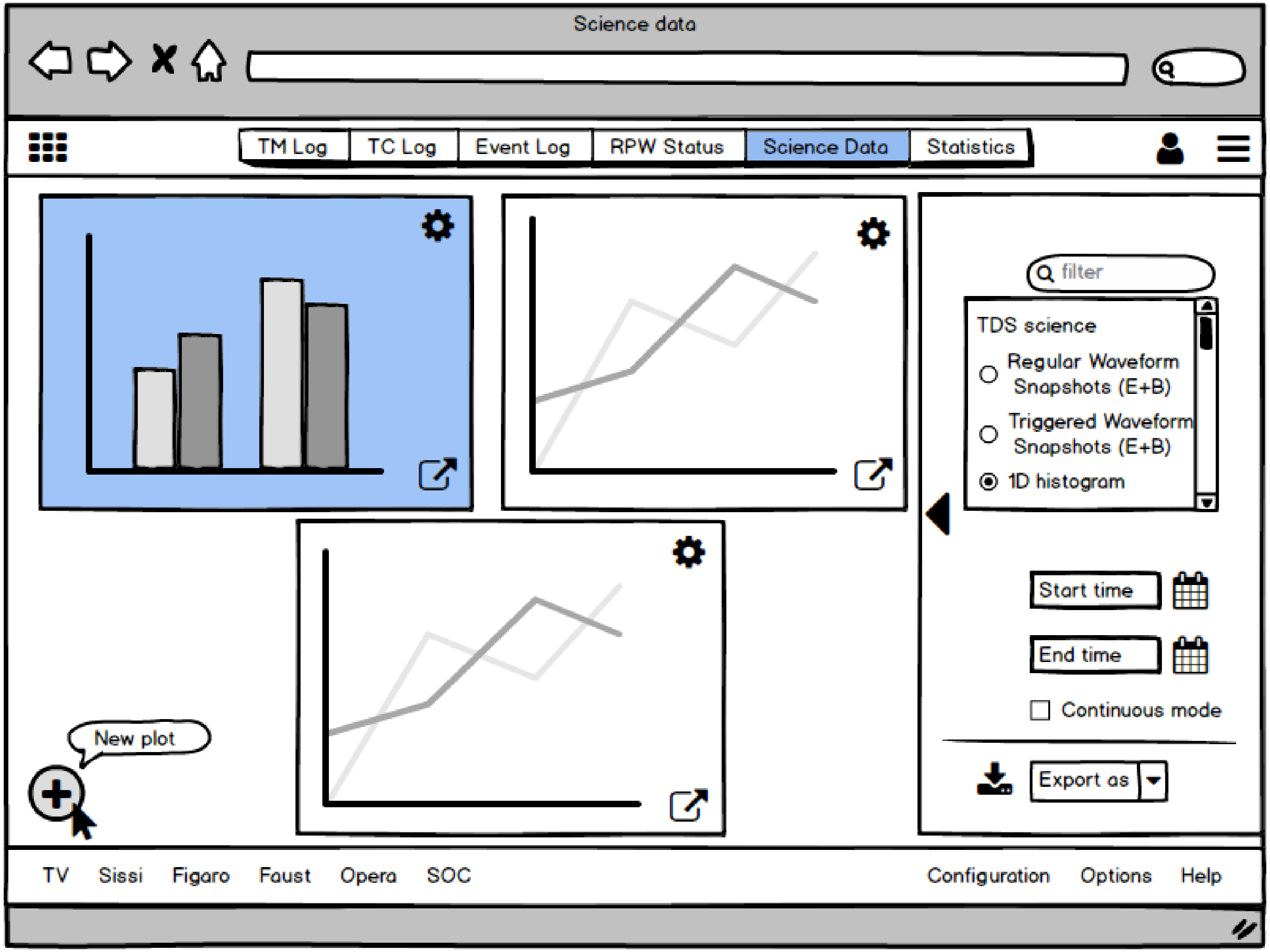
Category	Count
TM total count number	
TM count number since last counter reset on-board	
Received TM total count number	
Validated TM total count number	
Corrupted TM total count number	
Processed TM total count number	
Received TM count number since last counter reset onboard	
Validated TM count number since last counter reset onboard	
Corrupted TM count number since last counter reset onboard	
Processed TM count number since last counter reset onboard	
TC total count number	
TC failed (acceptance) total count number	
TC failed (execution) total count number	

☑ TM total count number

☑ Received TM total count number

□ Continuous mode

TV - HK/science data



Science data

TDS science

○ Regular Waveform Snapshots (E+B)

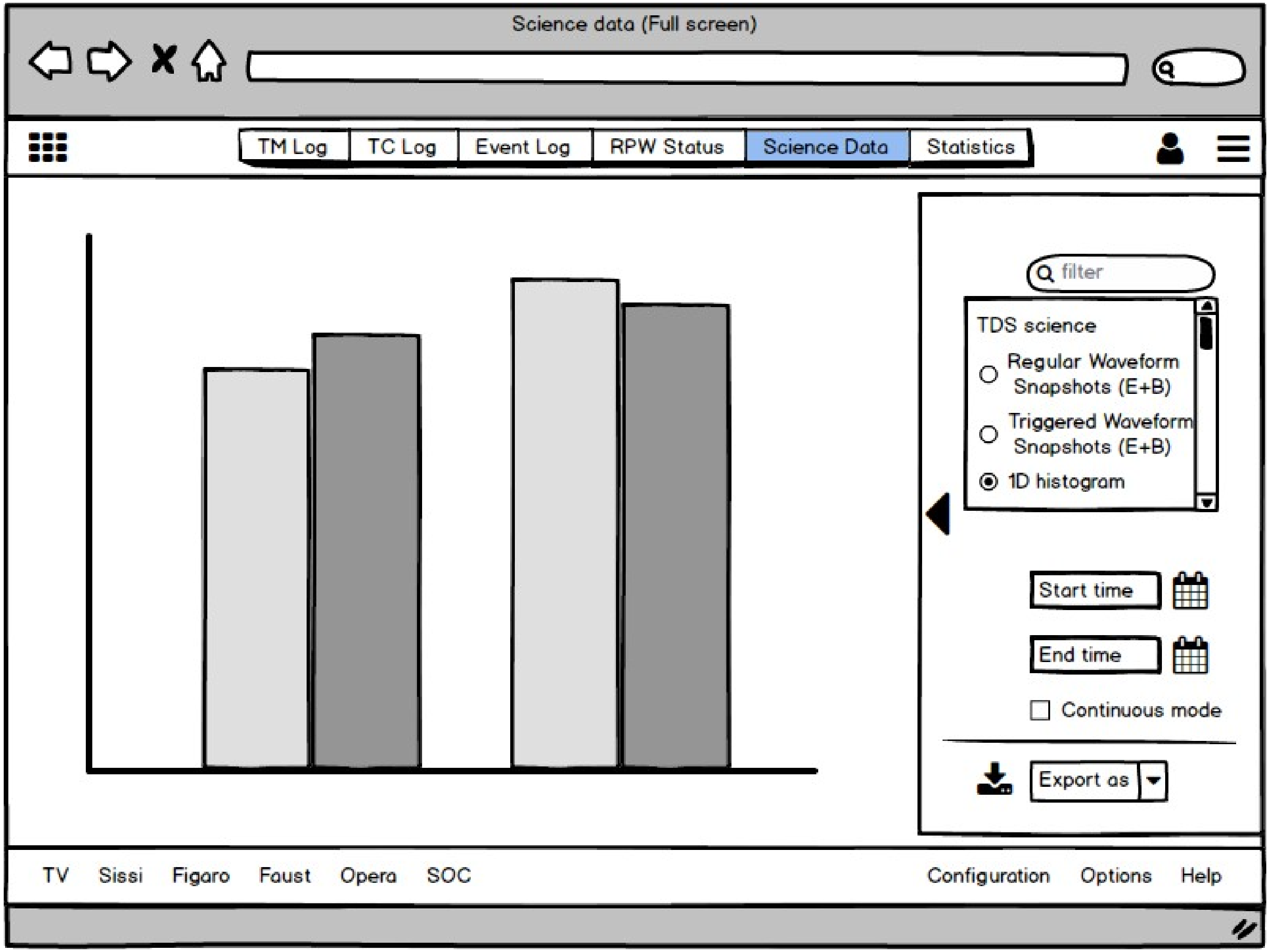
○ Triggered Waveform Snapshots (E+B)

● 1D histogram

□ Continuous mode

Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

TV - HK/science data



TDS science

○ Regular Waveform Snapshots (E+B)

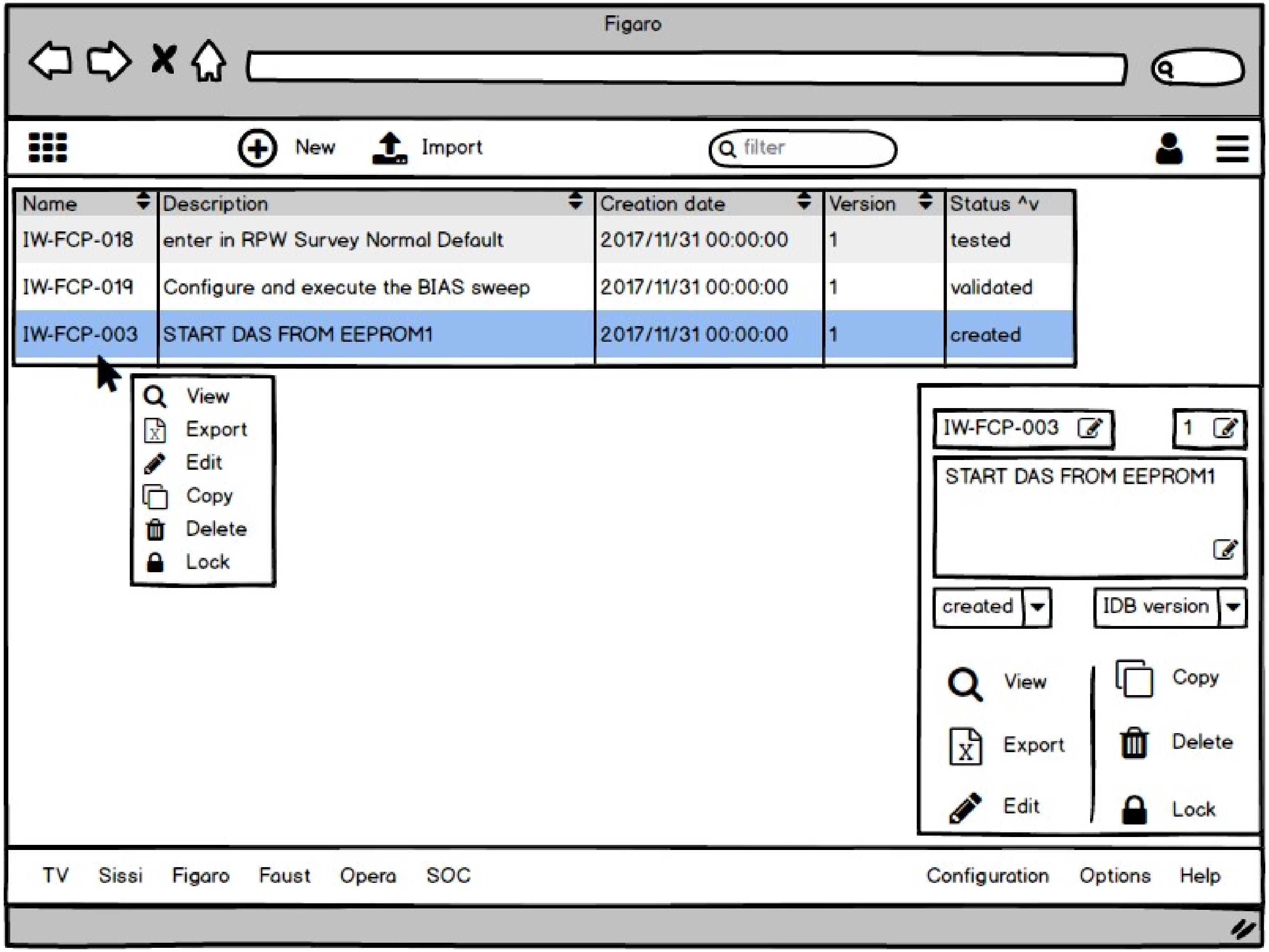
○ Triggered Waveform Snapshots (E+B)

● 1D histogram

□ Continuous mode

Figaro

The RPW Flight Operation Procedure Editor, a Web GUI to edit and save the RPW flight procedures (RFP). The RFP will serve as a primary library to generate the RPW operation requests (IOR, MDOR, PDOR).



Name	Description	Creation date	Version	Status ^v
IW-FCP-018	enter in RPW Survey Normal Default	2017/11/31 00:00:00	1	tested
IW-FCP-019	Configure and execute the BIAS sweep	2017/11/31 00:00:00	1	validated
IW-FCP-003	START DAS FROM EEPROM1	2017/11/31 00:00:00	1	created

IW-FCP-003

1

START DAS FROM EEPROM1

created

IDB version

Faust

The RPW Flight Operation Request Editor, the ROC GUI to prepare and submit the instrument commanding sequences, in accordance with the mission planning constraints and interface specification.

Main page

The ROR library (a table) contains the following ROR meta-data:

- name of the ROR;
- type of the ROR (MDOR, PDOR or IOR);
- (...)

Editing window (timeline or table mode)

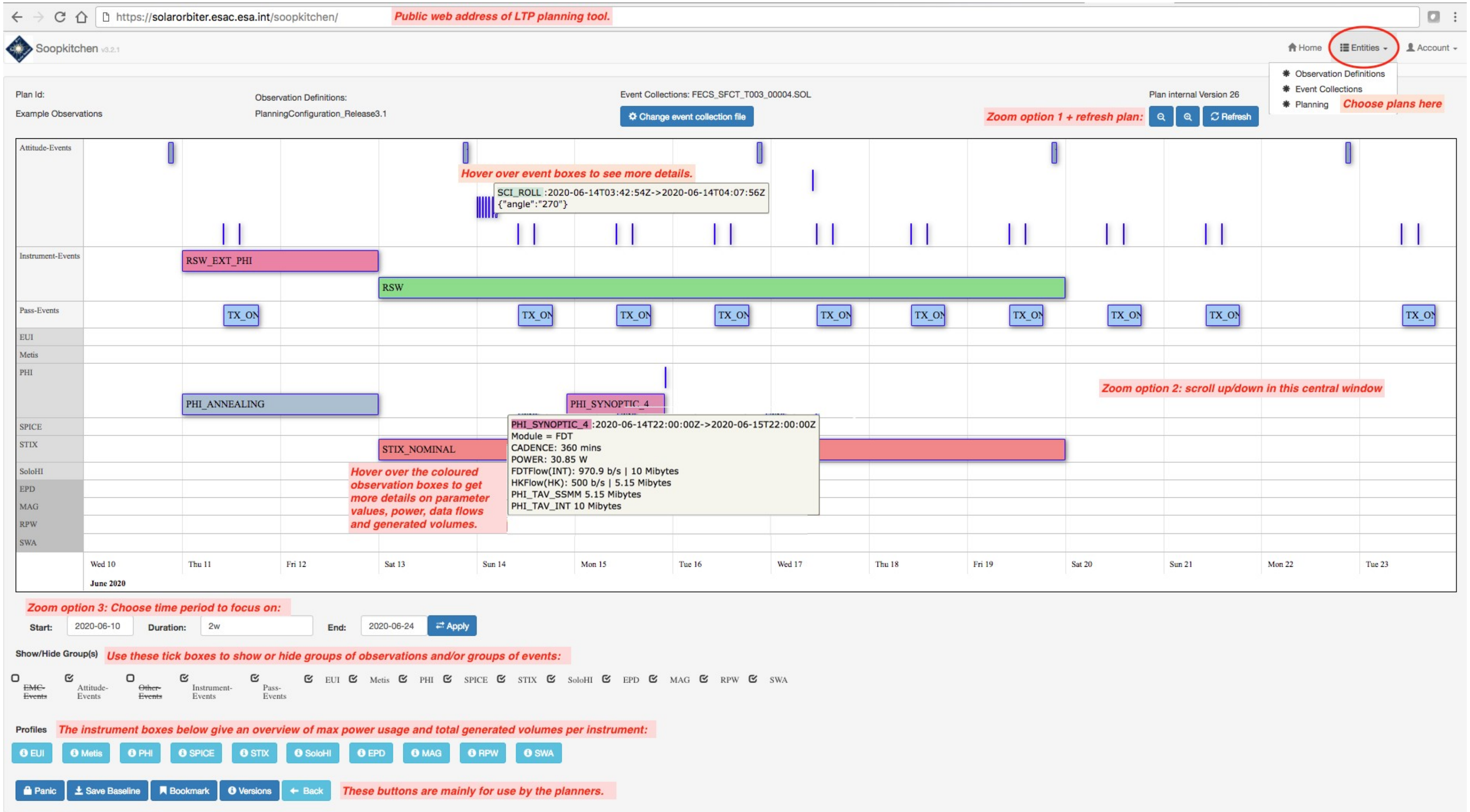
The user can drag and drop sequences and group of sequences from the sidebar to the table

Similarly, he can rearrange the sequences in the table

By clicking on a sequence, he can edit the formal parameters and adjust the starting time

Opera

The RPW Operation Planning Interface, a Web GUI to visualize and plan the instrument operations timeline.



https://solarorbiter.esac.esa.int/soopkitchen/ *Public web address of LTP planning tool.*

Plan Id: Example Observations

Observation Definitions: PlanningConfiguration_Release3.1

Event Collections: FECS_SFCT_T003_00004.SOL

Plan internal Version 26

Zoom option 1 + refresh plan:

Choose plans here

Hover over event boxes to see more details.

SCI_ROLL :2020-06-14T03:42:54Z->2020-06-14T04:07:56Z
{"angle":"270"}

Zoom option 2: scroll up/down in this central window

PHI_SYNOPTIC_4 :2020-06-14T22:00:00Z->2020-06-15T22:00:00Z
Module = FDT
CADENCE: 360 mins
POWER: 30.85 W
FDTFlow(INT): 970.9 b/s | 10 Mibytes
HKFlow(HK): 500 b/s | 5.15 Mibytes
PHI_TAV_SSMM 5.15 Mibytes
PHI_TAV_INT 10 Mibytes

Hover over the coloured observation boxes to get more details on parameter values, power, data flows and generated volumes.

Zoom option 3: Choose time period to focus on:

Start: 2020-06-10 Duration: 2w End: 2020-06-24

Show/Hide Group(s) *Use these tick boxes to show or hide groups of observations and/or groups of events:*

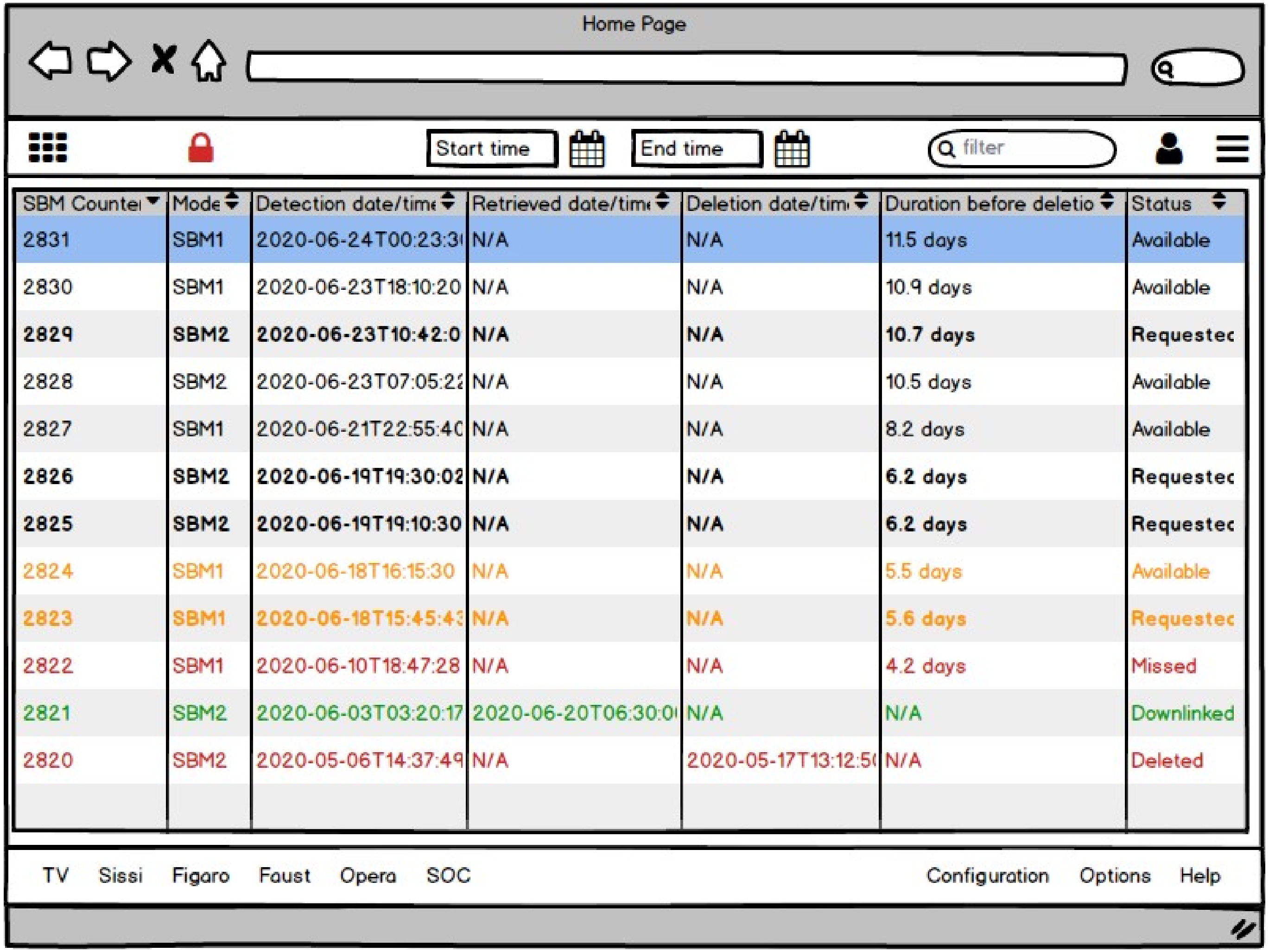
□ ~~EMC-Events~~ ☑ Attitude-Events □ ~~Other-Events~~ ☑ Instrument-Events ☑ Pass-Events ☑ EUI ☑ Metis ☑ PHI ☑ SPICE ☑ STIX ☑ SoloHI ☑ EPD ☑ MAG ☑ RPW ☑ SWA

Profiles *The instrument boxes below give an overview of max power usage and total generated volumes per instrument:*

These buttons are mainly for use by the planners.

SISSI

The SBM Interactive Selection System Interface, a Web GUI that allows ROC team to manage and select the SBM event data to downlink. The architecture of these GUIs includes interfaces with the RODP and MDB to retrieve/store related data and meta-data.



SBM Countei	Mode	Detection date/time	Retrieved date/time	Deletion date/tim	Duration before deletio	Status
2831	SBM1	2020-06-24T00:23:3	N/A	N/A	11.5 days	Available
2830	SBM1	2020-06-23T18:10:20	N/A	N/A	10.9 days	Available
2829	**SBM2**	**2020-06-23T10:42:0**	**N/A**	**N/A**	**10.7 days**	**Requestec**
2828	SBM2	2020-06-23T07:05:2	N/A	N/A	10.5 days	Available
2827	SBM1	2020-06-21T22:55:4	N/A	N/A	8.2 days	Available
2826	**SBM2**	**2020-06-19T19:30:02**	**N/A**	**N/A**	**6.2 days**	**Requestec**
2825	**SBM2**	**2020-06-19T19:10:30**	**N/A**	**N/A**	**6.2 days**	**Requestec**
2824	SBM1	2020-06-18T16:15:30	N/A	N/A	5.5 days	Available
2823	**SBM1**	**2020-06-18T15:45:4**	**N/A**	**N/A**	**5.6 days**	**Requestec**
2822	SBM1	2020-06-10T18:47:28	N/A	N/A	4.2 days	Missed
2821	SBM2	2020-06-03T03:20:17	2020-06-20T06:30:0	N/A	N/A	Downlinked
2820	SBM2	2020-05-06T14:37:49	N/A	2020-05-17T13:12:5	N/A	Deleted

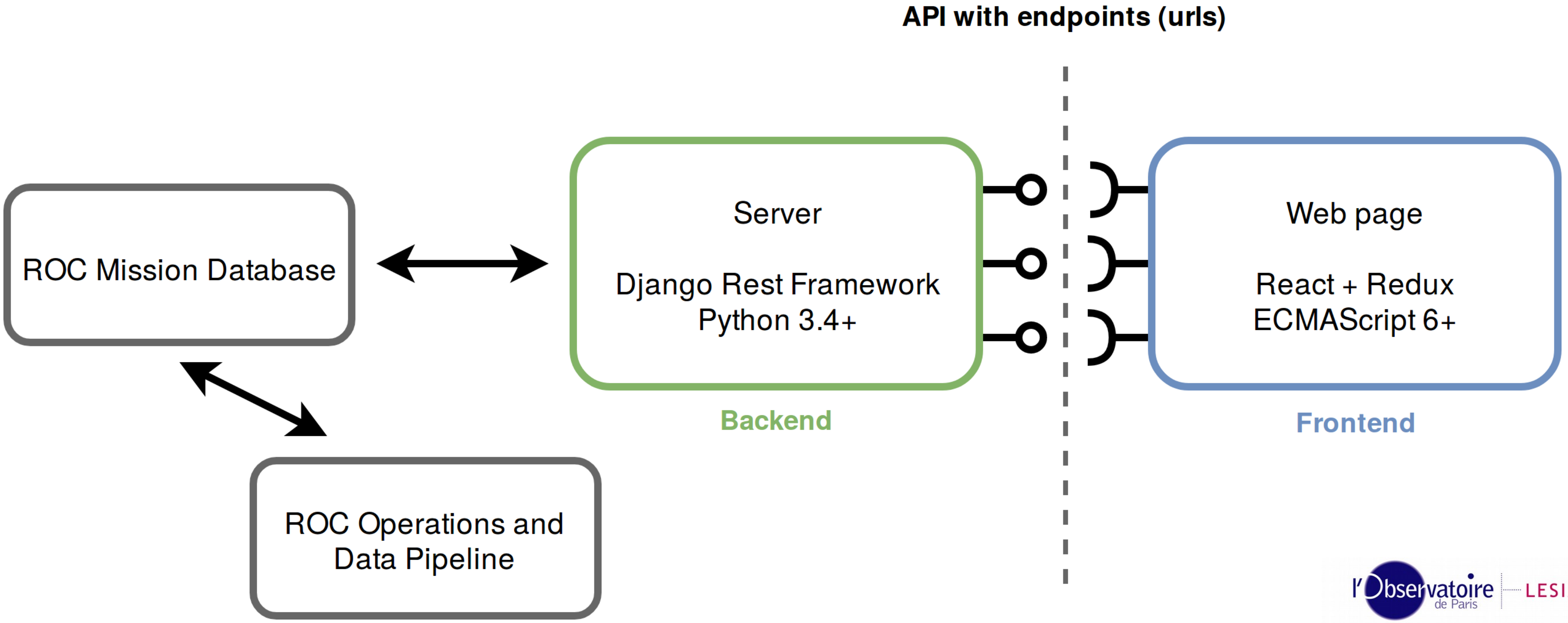
Software architecture overview

MUsIC is divided into two parts connected by an API:

- the frontend (a web page)
- the backend (a Django server)

API calls are only needed when user actions require or affect the database.

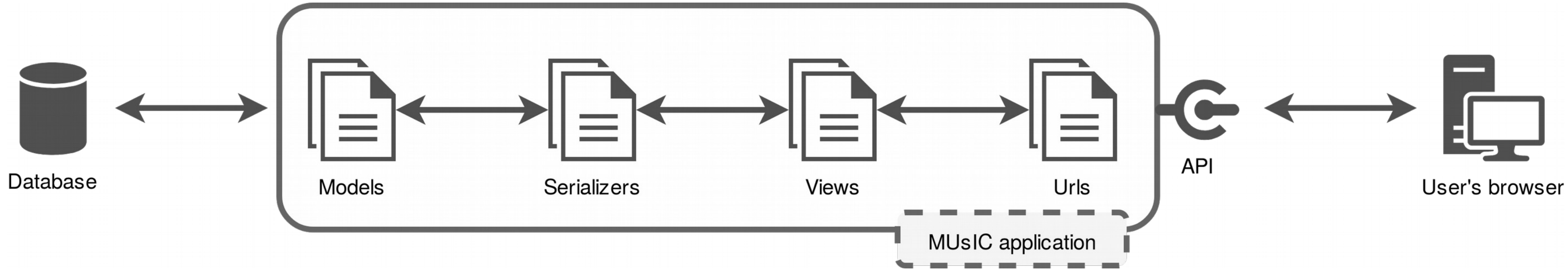
The backend is self consistent and can be used to interact with the database with any software able to send and receive HTTP requests/responses.



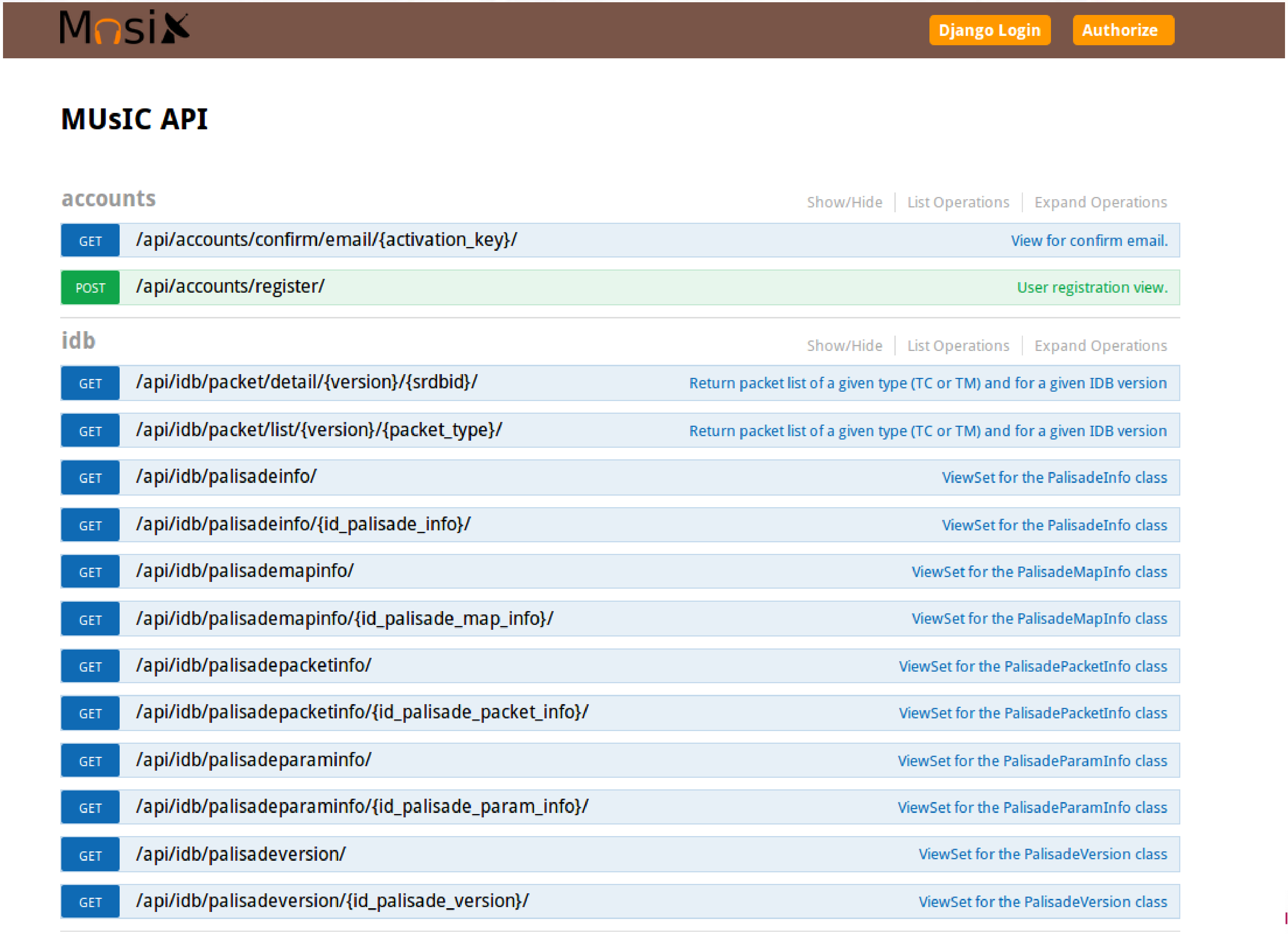
Backend - REST API

A **REST API** defines a set of functions which developers can perform **requests** and receive **responses** via **HTTP protocol** such as GET and POST.

- GET - /api/figaro/procedures - display all procedures
- POST - /api/figaro/procedures - create a new procedure
- GET - /api/figaro/procedures/{id} - display a procedure by ID
- PUT - /api/figaro/procedures/{id} - update a procedure by ID
- DELETE - /api/figaro/procedures/{id} – delete a procedure by ID



Backend - MUsIC API


Musix
Django Login
Authorize

MUsIC API

accounts

Show/Hide | List Operations | Expand Operations

GET	/api/accounts/confirm/email/{activation_key}/	View for confirm email.
POST	/api/accounts/register/	User registration view.

idb

Show/Hide | List Operations | Expand Operations

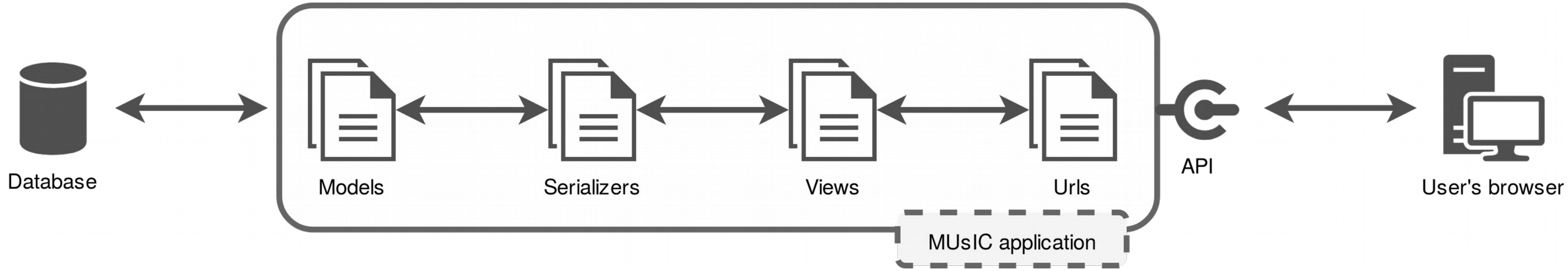
GET	/api/idb/packet/detail/{version}/{srdbid}/	Return packet list of a given type (TC or TM) and for a given IDB version
GET	/api/idb/packet/list/{version}/{packet_type}/	Return packet list of a given type (TC or TM) and for a given IDB version
GET	/api/idb/palisadeinfo/	ViewSet for the PalisadeInfo class
GET	/api/idb/palisadeinfo/{id_palisade_info}/	ViewSet for the PalisadeInfo class
GET	/api/idb/palisademapinfo/	ViewSet for the PalisadeMapInfo class
GET	/api/idb/palisademapinfo/{id_palisade_map_info}/	ViewSet for the PalisadeMapInfo class
GET	/api/idb/palisadepacketinfo/	ViewSet for the PalisadePacketInfo class
GET	/api/idb/palisadepacketinfo/{id_palisade_packet_info}/	ViewSet for the PalisadePacketInfo class
GET	/api/idb/palisadeparaminfo/	ViewSet for the PalisadeParamInfo class
GET	/api/idb/palisadeparaminfo/{id_palisade_param_info}/	ViewSet for the PalisadeParamInfo class
GET	/api/idb/palisadeversion/	ViewSet for the PalisadeVersion class
GET	/api/idb/palisadeversion/{id_palisade_version}/	ViewSet for the PalisadeVersion class

Backend - architecture

- **Models** map the database entries and facilitate entries manipulation and mutation (migrations)

- Database entries are formatted by the **serializer** to be used by the client

- The API act as a **security layer** limiting interactions between the users and the database



Backend - Authentication system

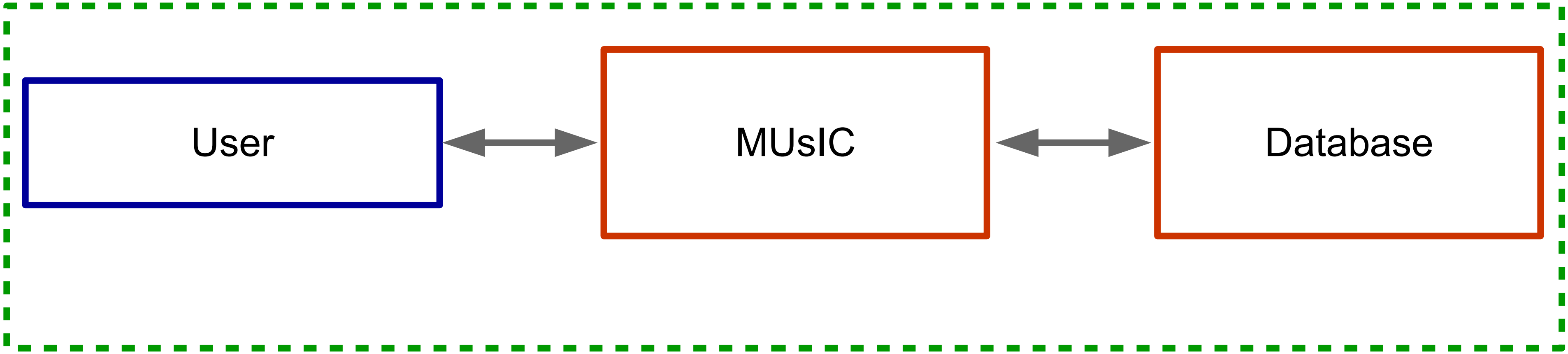
Built-in system

The authentication system is based on a login / password pair
➔ only the hash of the password is saved in the database

Once authenticated, the user receives a token that will be used in all his future requests and those until his disconnection.

Tokens are used to determine permission levels.

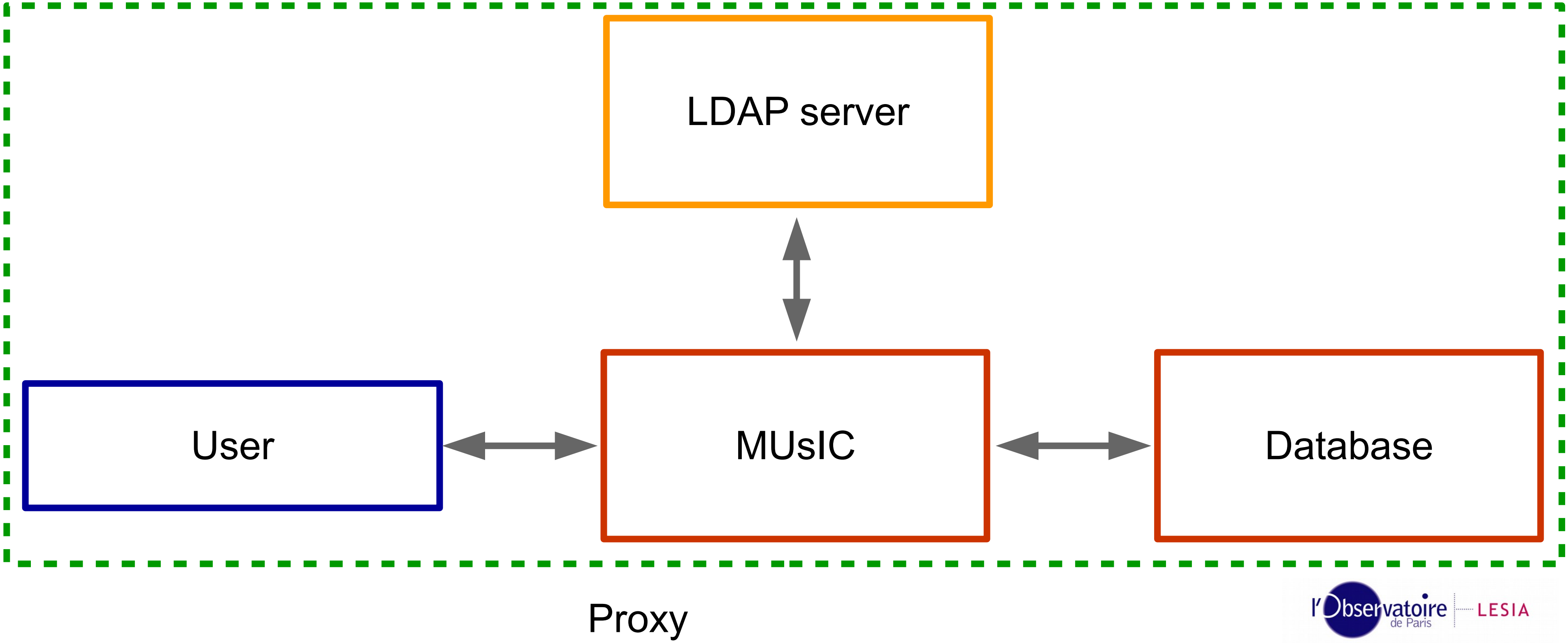
If a token reaches mid-life or if a new session is open, the token is automatically renewed



Proxy

LDAP accounts

A user can chose to log in with a LDAP account. The validation is then delegated to the LDAP server which returns a (LDAP) token to certify the user authentication. Then, as in the built-in system, the application generates a (MUsIC) token that will be used to to authenticate future requests.



Backend – Authentication system

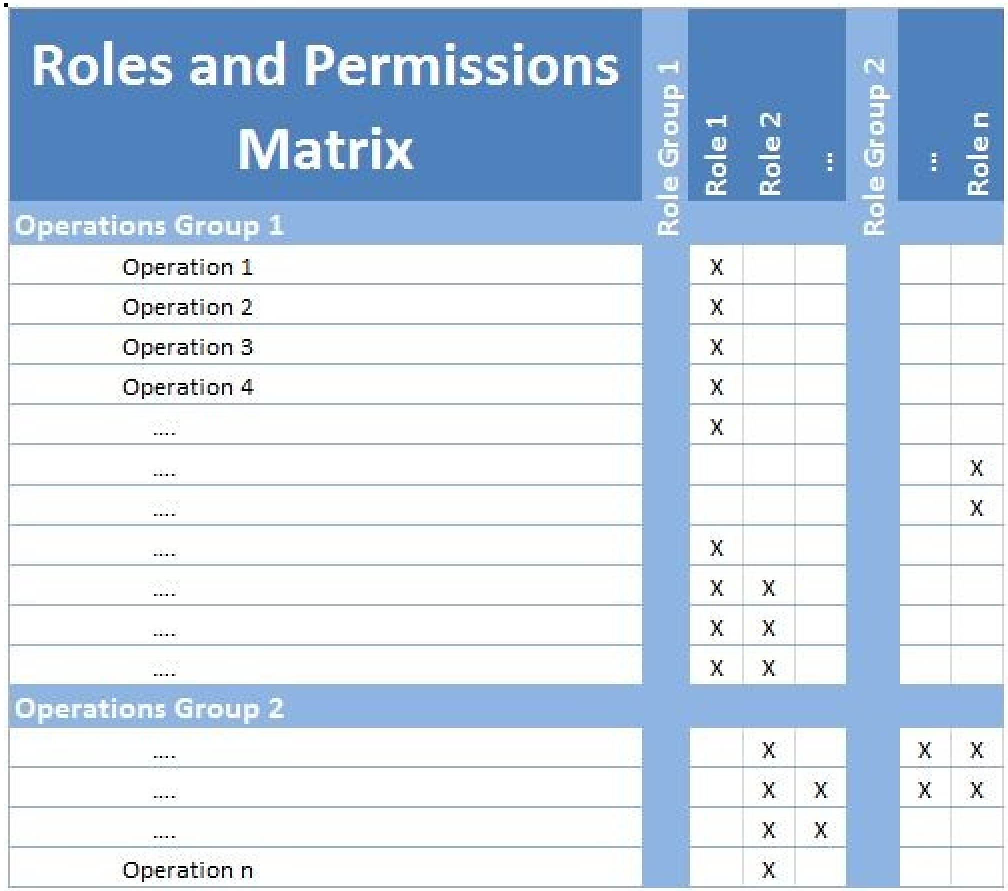
Permissions

The permissions system is divided between:

- The **user permissions system**, a suite of binary flags designating whether a user may perform a certain task.

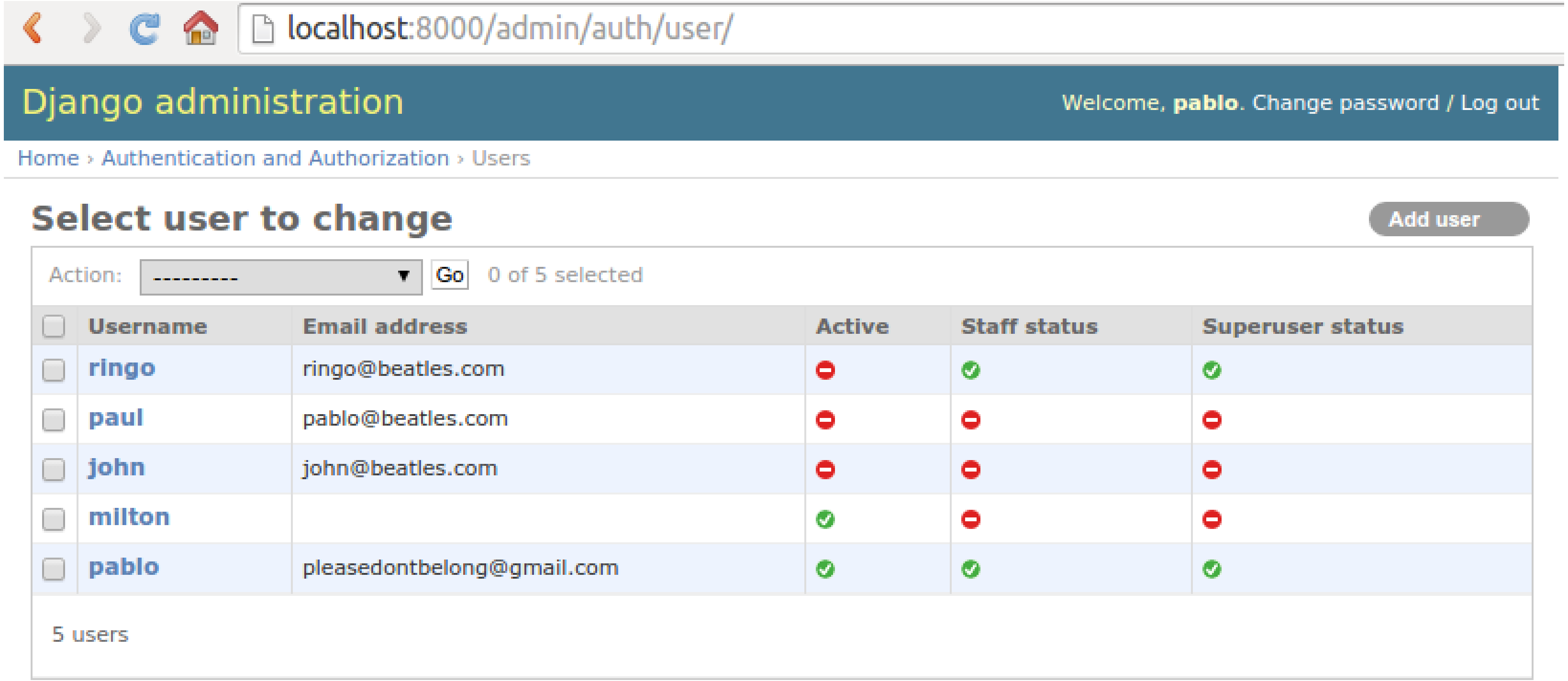
- The **group permissions system**, a generic way of applying labels and permissions to more than one user. The labels are defined in the RSSS

When a user is in multiple user groups, the **higher permission level** becomes the **effective level**



Roles and Permissions Matrix	Role Group 1	Role 1	Role 2	...	Role Group 2	...	Role n
Operations Group 1							
Operation 1		X					
Operation 2		X					
Operation 3		X					
Operation 4		X					
....		X					
....							X
....							X
....		X					
....		X	X				
....		X	X				
....		X	X				
Operations Group 2							
....			X			X	X
....			X	X		X	X
....			X	X			
Operation n			X				

Backend - Admin panel



Django administration

Welcome, **pablo**. Change password / Log out

Home › Authentication and Authorization › Users

Select user to change

Add user

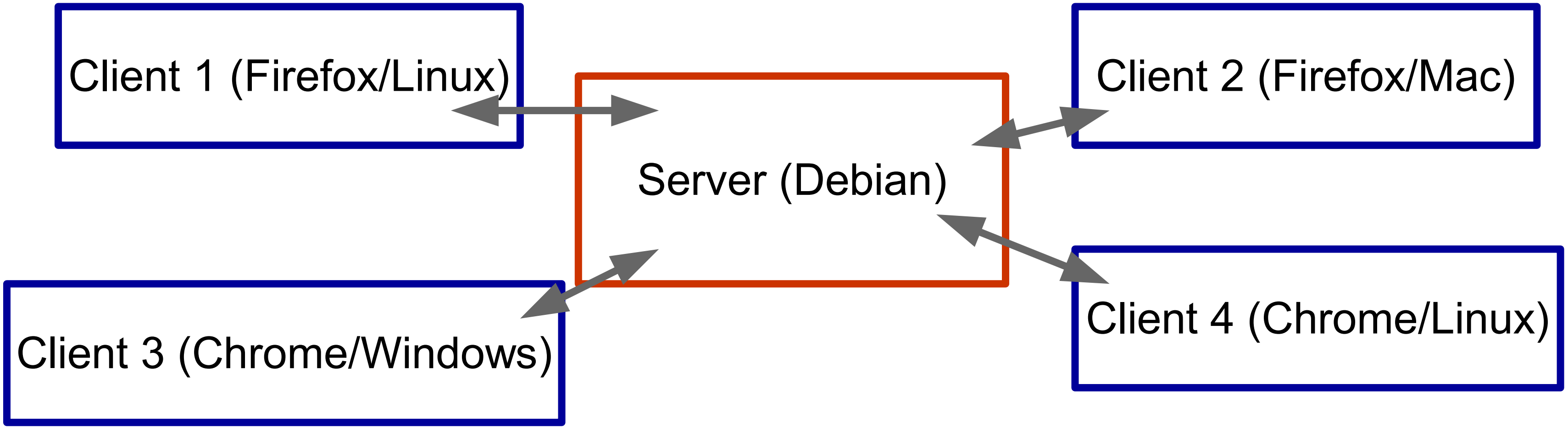
Action: ---------- ▼ Go 0 of 5 selected

□	Username	Email address	Active	Staff status	Superuser status
□	ringo	ringo@beatles.com	⊖	✓	✓
□	paul	pablo@beatles.com	⊖	⊖	⊖
□	john	john@beatles.com	⊖	⊖	⊖
□	milton		✓	⊖	⊖
□	pablo	pleasedontbelong@gmail.com	✓	✓	✓

5 users

Environment

The backend has been successfully tested on Linux/Mac/Windows but will ultimately run on a Debian server.



Web technologies allow to build cross-platform application with a single dependency: a web browser.

MusIC is developed to run on Firefox and Chrome but should be able to run on any recent browser.

Frontend - React

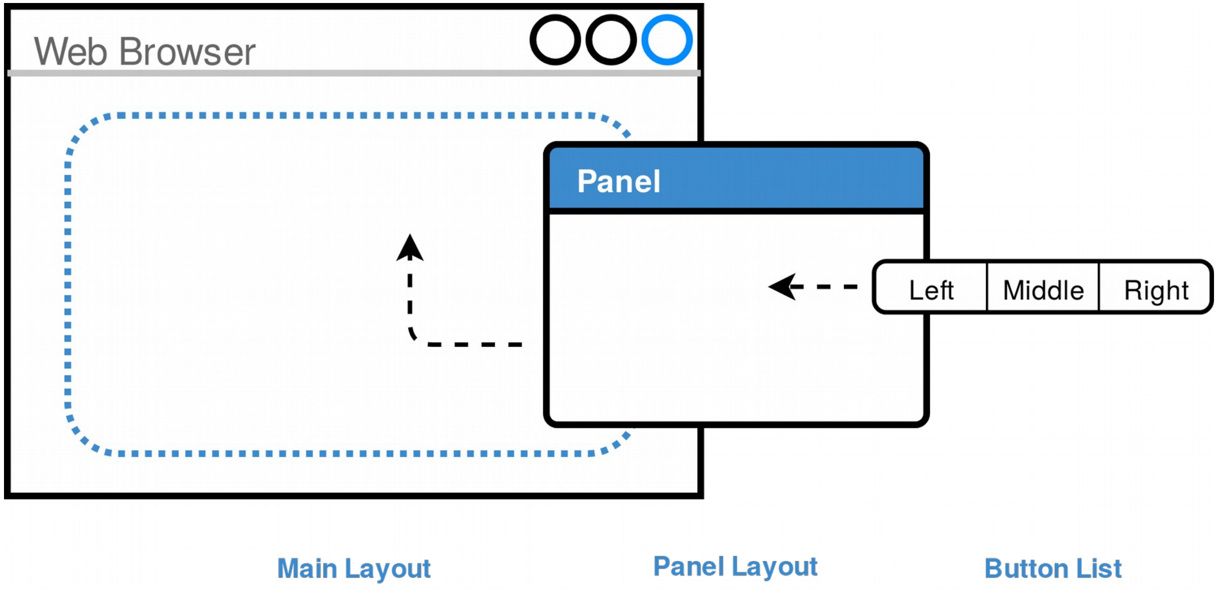
React is an open source, component based JavaScript library for building user interfaces.

A component defines a unit part of the application

Each component can be composed of other components

These components are:

- **Reusable**
- **Testable**



Frontend - Redux with synchronous data flow

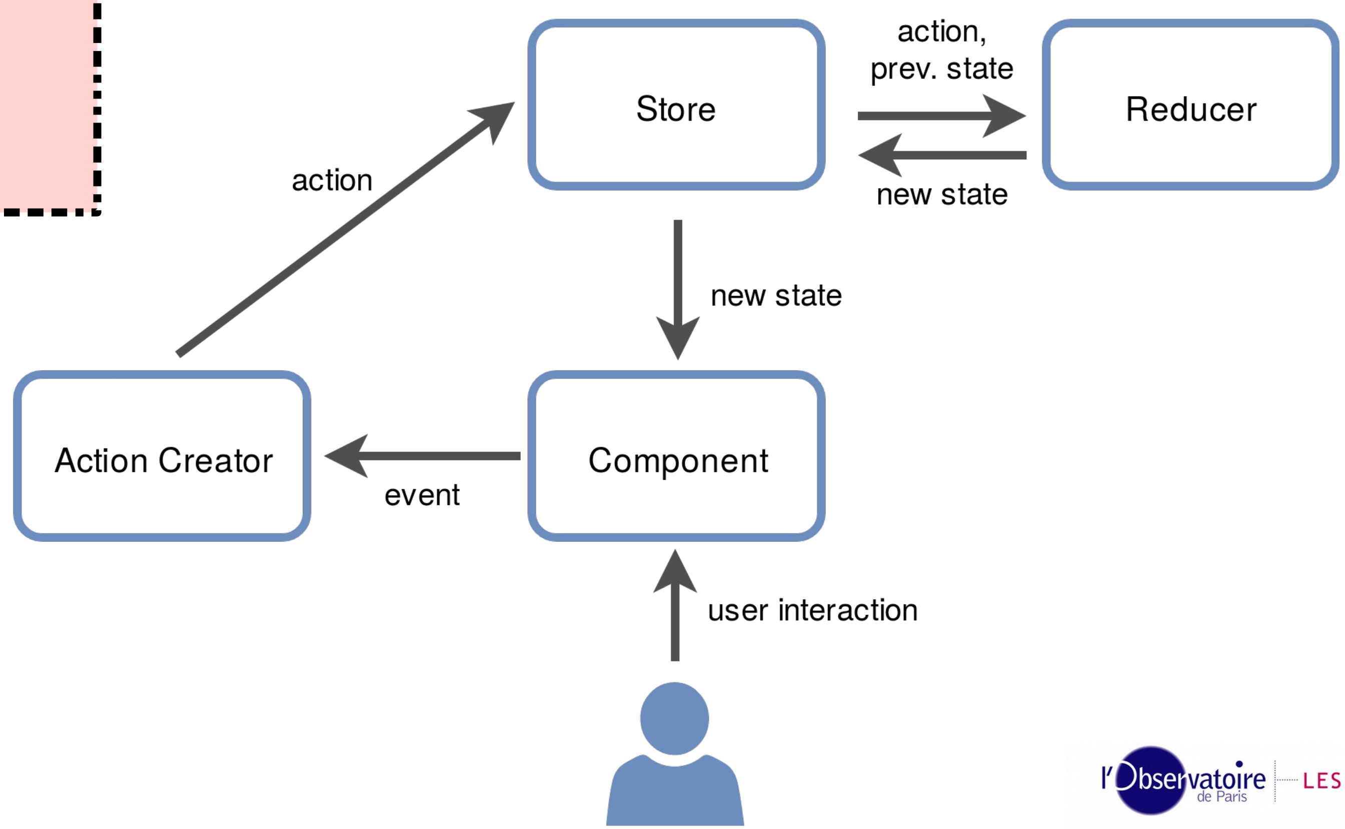
Redux is a way to organize data. Redux has strict guidelines of how data can move or flow through a project, which is known as **unidirectional data flow**

State

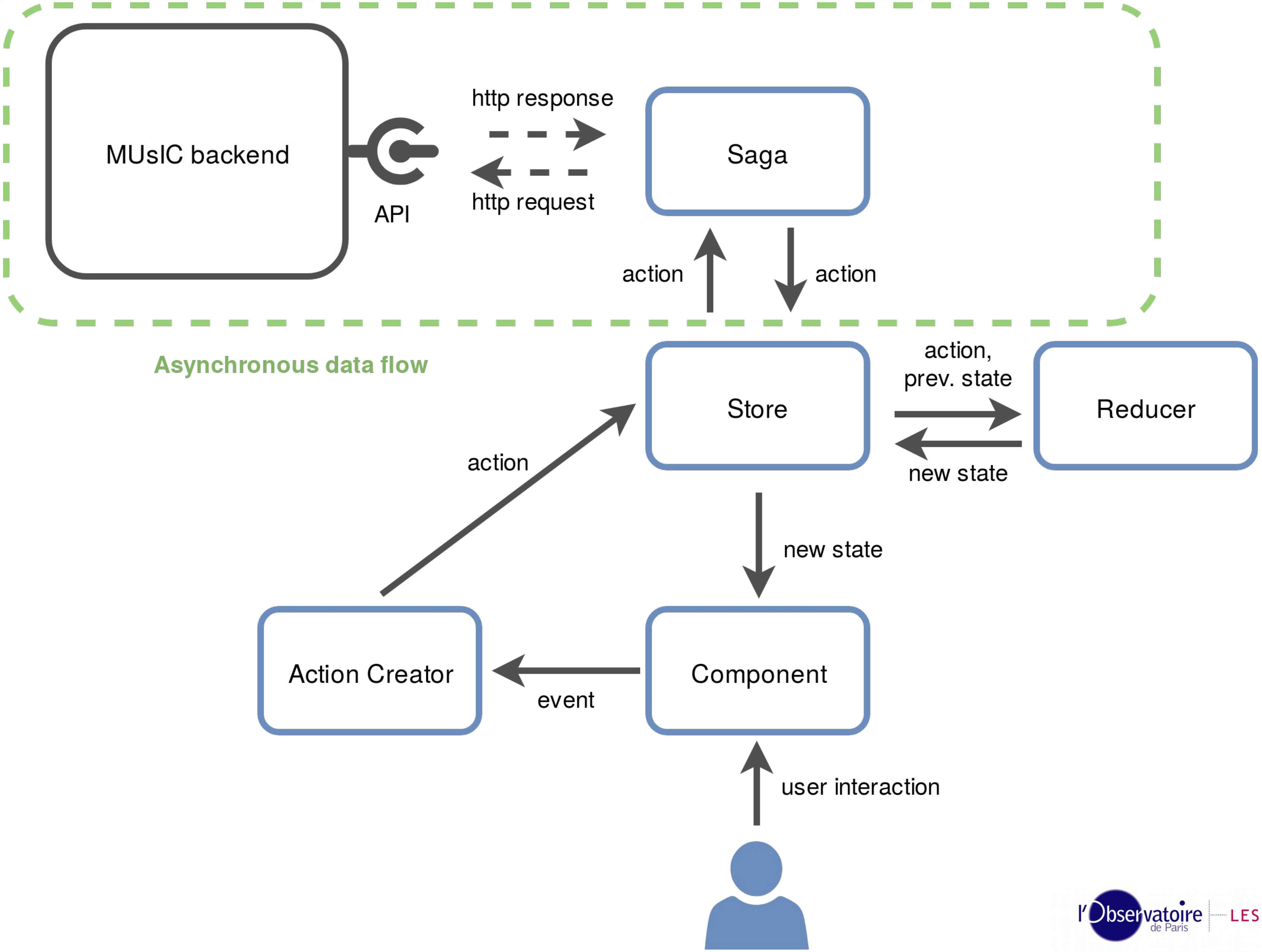
```
{
  var1 : 'var1' ,
  var2 : […],
  var3 : true,
  ...
}
```

Action

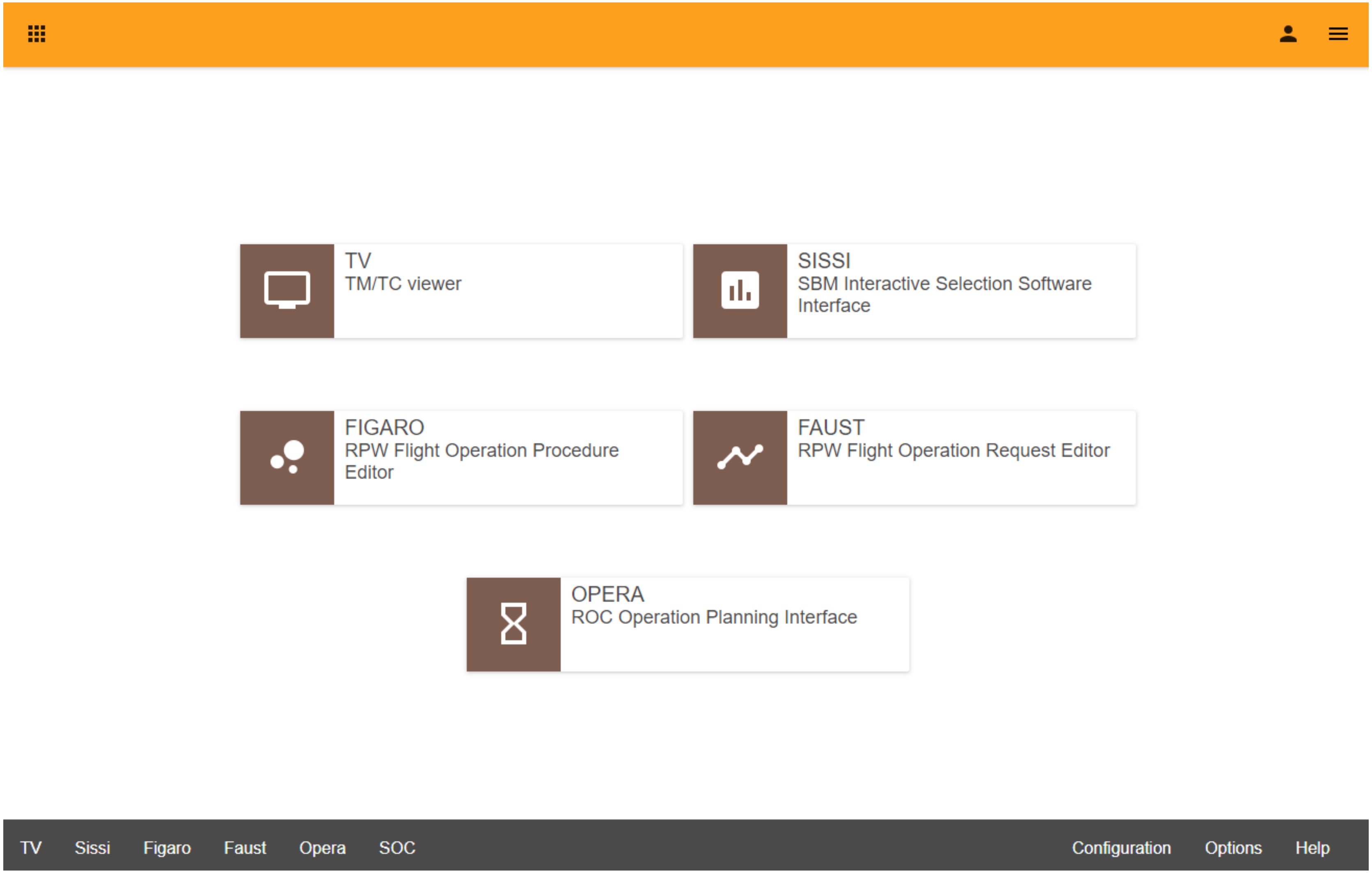
```
{
  type : 'LABEL' ,
  data : [...]
}
```



Frontend - Redux with asynchronous data flow



GUI design – Home page



TV
TM/TC viewer

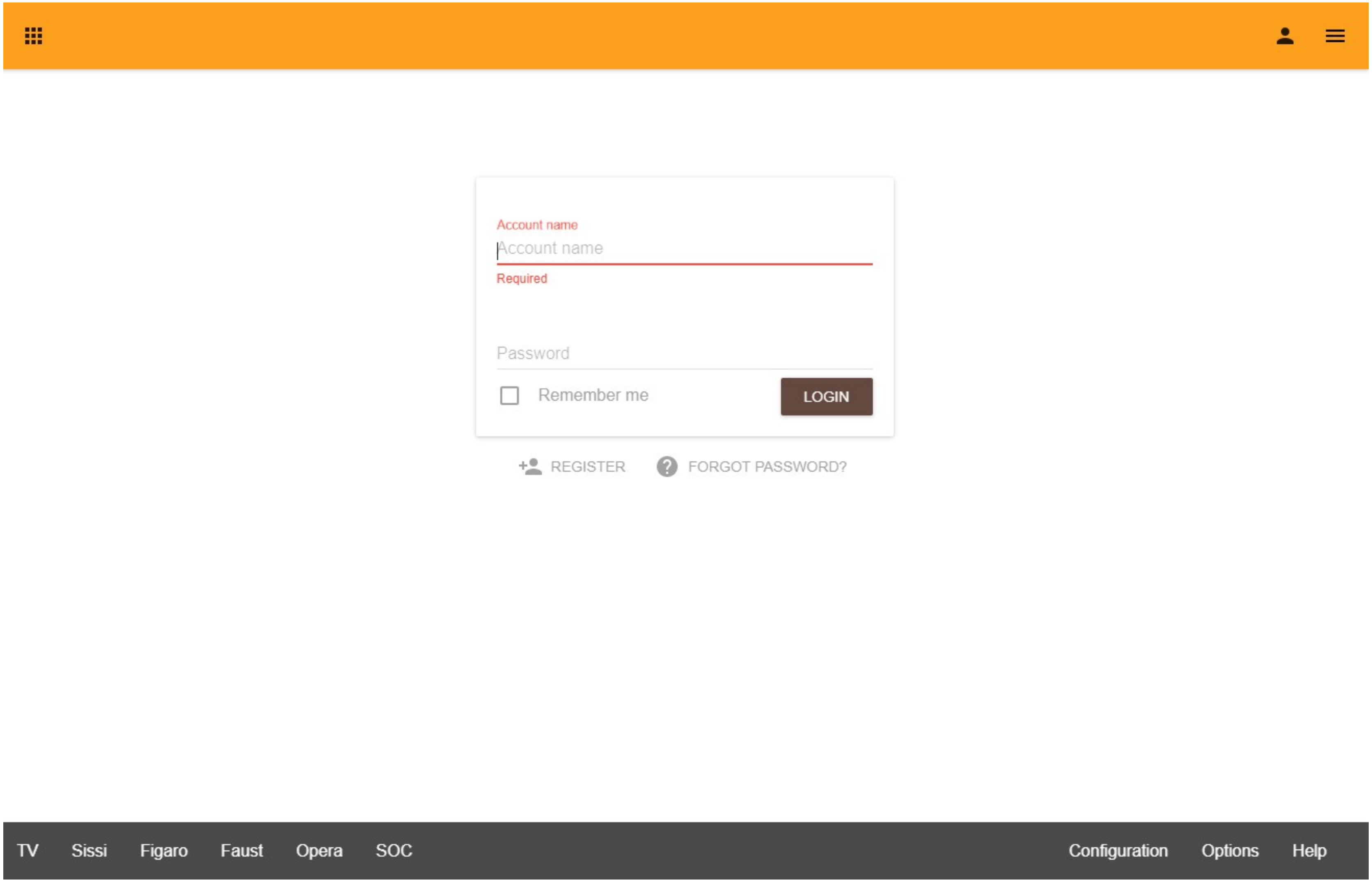
SISSI
SBM Interactive Selection Software Interface

FIGARO
RPW Flight Operation Procedure Editor

FAUST
RPW Flight Operation Request Editor

OPERA
ROC Operation Planning Interface

GUI design – Login page



Account name

Account name

Required

Password

□ Remember me

LOGIN

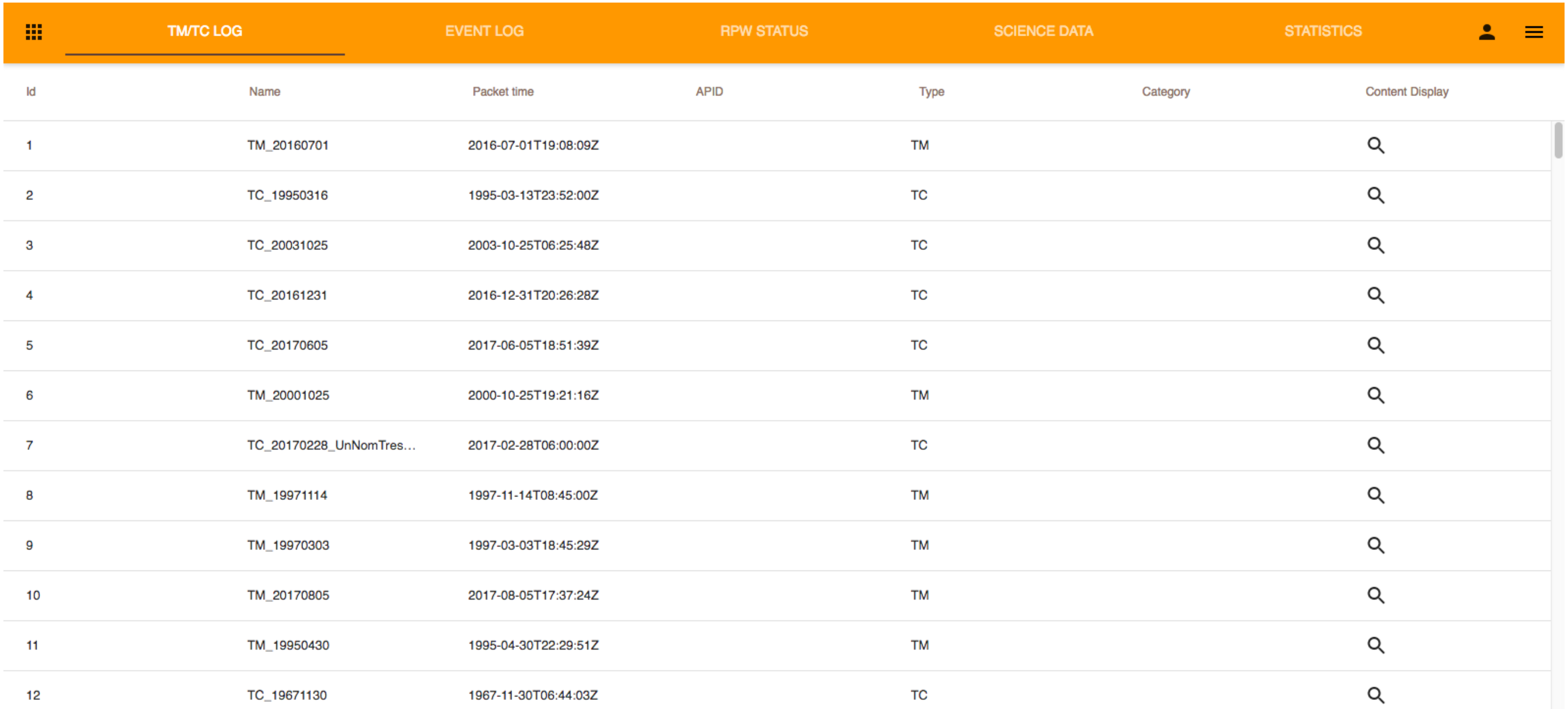
REGISTER

FORGOT PASSWORD?

TV Sissi Figaro Faust Opera SOC

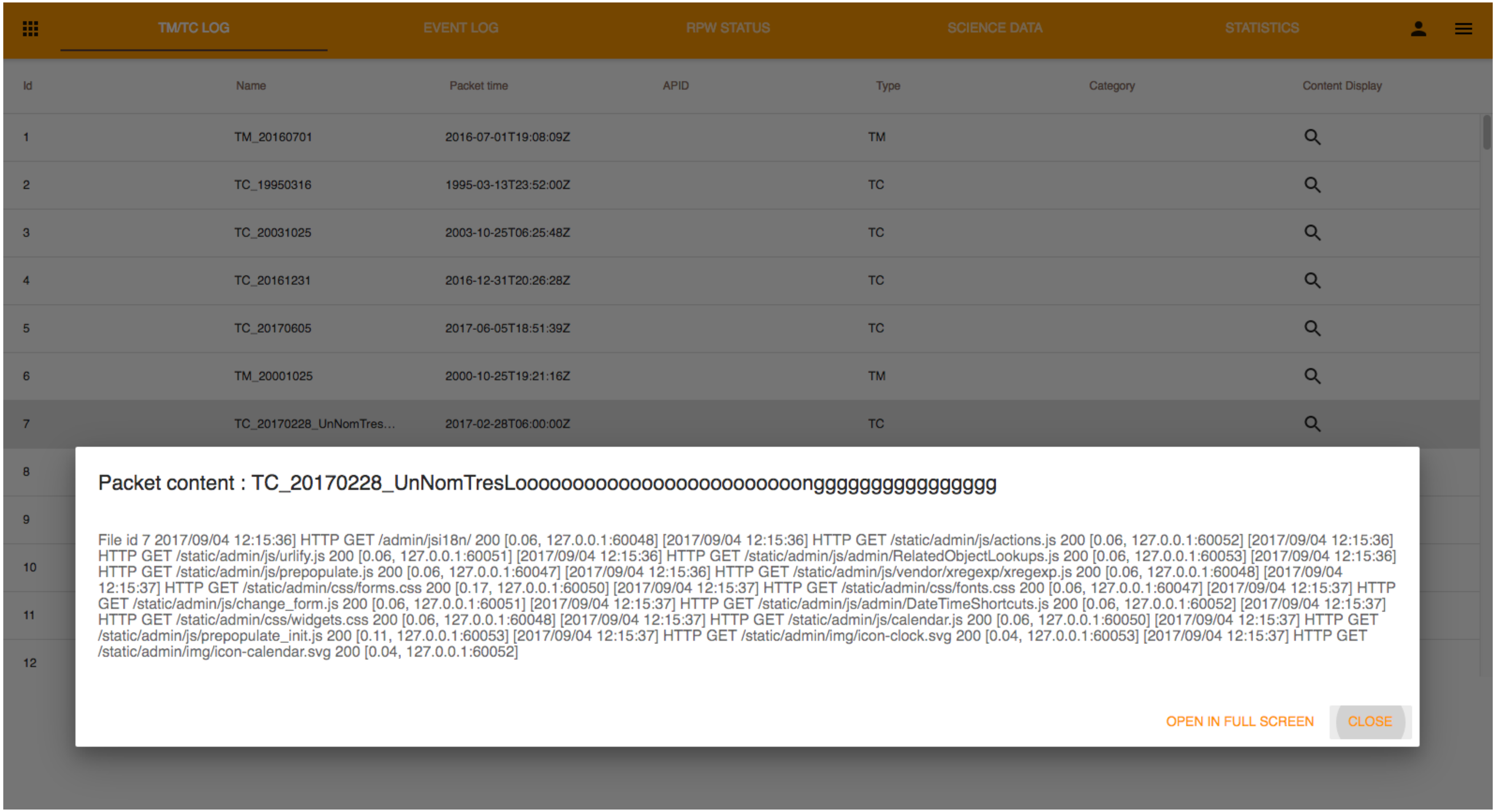
Configuration Options Help

GUI design – TM/TC log



Id	Name	Packet time	APID	Type	Category	Content Display
1	TM_20160701	2016-07-01T19:08:09Z		TM		🔍
2	TC_19950316	1995-03-13T23:52:00Z		TC		🔍
3	TC_20031025	2003-10-25T06:25:48Z		TC		🔍
4	TC_20161231	2016-12-31T20:26:28Z		TC		🔍
5	TC_20170605	2017-06-05T18:51:39Z		TC		🔍
6	TM_20001025	2000-10-25T19:21:16Z		TM		🔍
7	TC_20170228_UnNomTres...	2017-02-28T06:00:00Z		TC		🔍
8	TM_19971114	1997-11-14T08:45:00Z		TM		🔍
9	TM_19970303	1997-03-03T18:45:29Z		TM		🔍
10	TM_20170805	2017-08-05T17:37:24Z		TM		🔍
11	TM_19950430	1995-04-30T22:29:51Z		TM		🔍
12	TC_19671130	1967-11-30T06:44:03Z		TC		🔍

GUI design – Packet detail

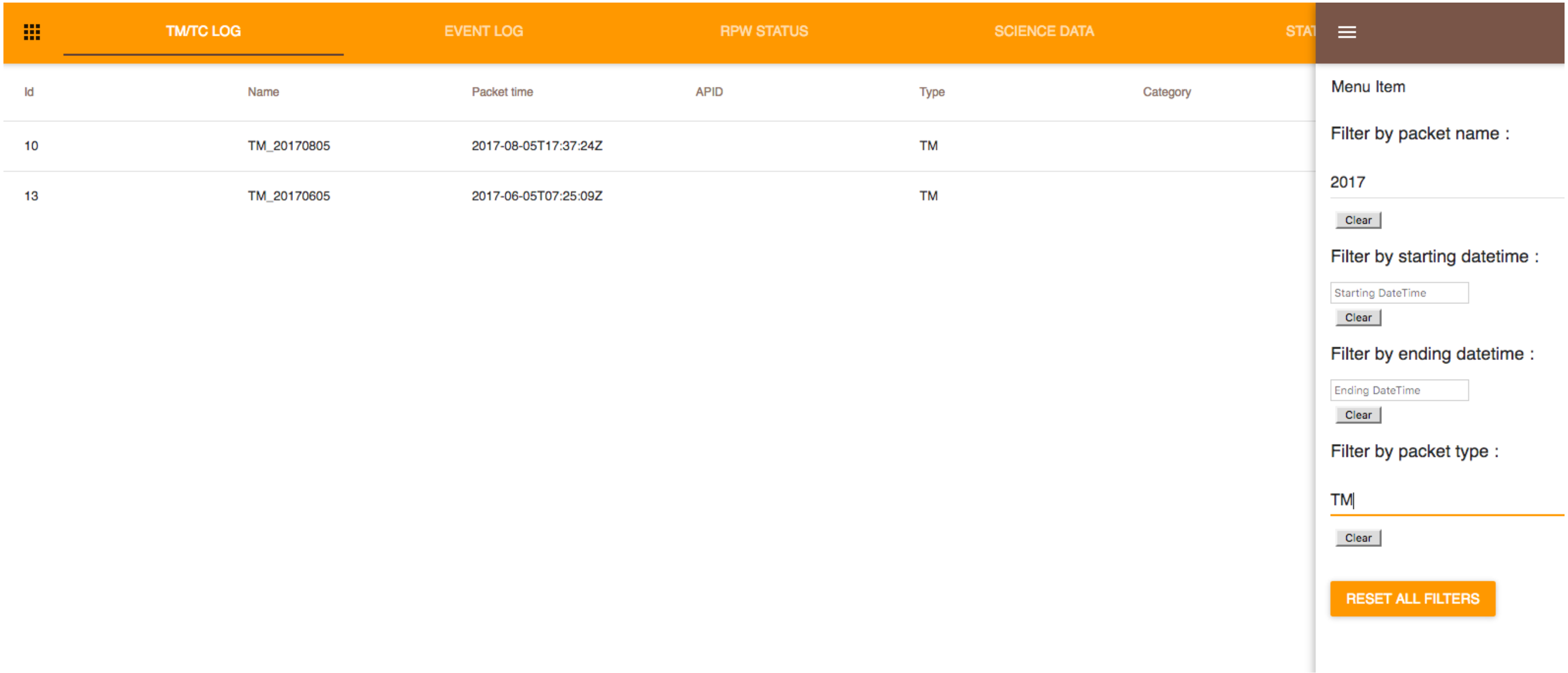


Id	Name	Packet time	APID	Type	Category	Content Display
1	TM_20160701	2016-07-01T19:08:09Z		TM		🔍
2	TC_19950316	1995-03-13T23:52:00Z		TC		🔍
3	TC_20031025	2003-10-25T06:25:48Z		TC		🔍
4	TC_20161231	2016-12-31T20:26:28Z		TC		🔍
5	TC_20170605	2017-06-05T18:51:39Z		TC		🔍
6	TM_20001025	2000-10-25T19:21:16Z		TM		🔍
7	TC_20170228_UnNomTres...	2017-02-28T06:00:00Z		TC		🔍
8						
9						
10						
11						
12						

Packet content : TC_20170228_UnNomTresLoooooooooooooooooooooooongggggggggggggg

```
File id 7 2017/09/04 12:15:36] HTTP GET /admin/jsi18n/ 200 [0.06, 127.0.0.1:60048] [2017/09/04 12:15:36] HTTP GET /static/admin/js/actions.js 200 [0.06, 127.0.0.1:60052] [2017/09/04 12:15:36]
HTTP GET /static/admin/js/urlify.js 200 [0.06, 127.0.0.1:60051] [2017/09/04 12:15:36] HTTP GET /static/admin/js/admin/RelatedObjectLookups.js 200 [0.06, 127.0.0.1:60053] [2017/09/04 12:15:36]
HTTP GET /static/admin/js/prepopulate.js 200 [0.06, 127.0.0.1:60047] [2017/09/04 12:15:36] HTTP GET /static/admin/js/vendor/xregexp/xregexp.js 200 [0.06, 127.0.0.1:60048] [2017/09/04
12:15:37] HTTP GET /static/admin/css/forms.css 200 [0.17, 127.0.0.1:60050] [2017/09/04 12:15:37] HTTP GET /static/admin/css/fonts.css 200 [0.06, 127.0.0.1:60047] [2017/09/04 12:15:37] HTTP
GET /static/admin/js/change_form.js 200 [0.06, 127.0.0.1:60051] [2017/09/04 12:15:37] HTTP GET /static/admin/js/admin/DateTimeShortcuts.js 200 [0.06, 127.0.0.1:60052] [2017/09/04 12:15:37]
HTTP GET /static/admin/css/widgets.css 200 [0.06, 127.0.0.1:60048] [2017/09/04 12:15:37] HTTP GET /static/admin/js/calendar.js 200 [0.06, 127.0.0.1:60050] [2017/09/04 12:15:37] HTTP GET
/static/admin/js/prepopulate_init.js 200 [0.11, 127.0.0.1:60053] [2017/09/04 12:15:37] HTTP GET /static/admin/img/icon-clock.svg 200 [0.04, 127.0.0.1:60053] [2017/09/04 12:15:37] HTTP GET
/static/admin/img/icon-calendar.svg 200 [0.04, 127.0.0.1:60052]
```

GUI design - Filtering



Id	Name	Packet time	APID	Type	Category
10	TM_20170805	2017-08-05T17:37:24Z		TM	
13	TM_20170605	2017-06-05T07:25:09Z		TM	

Filter by packet name :

2017

Filter by starting datetime :

Starting DateTime

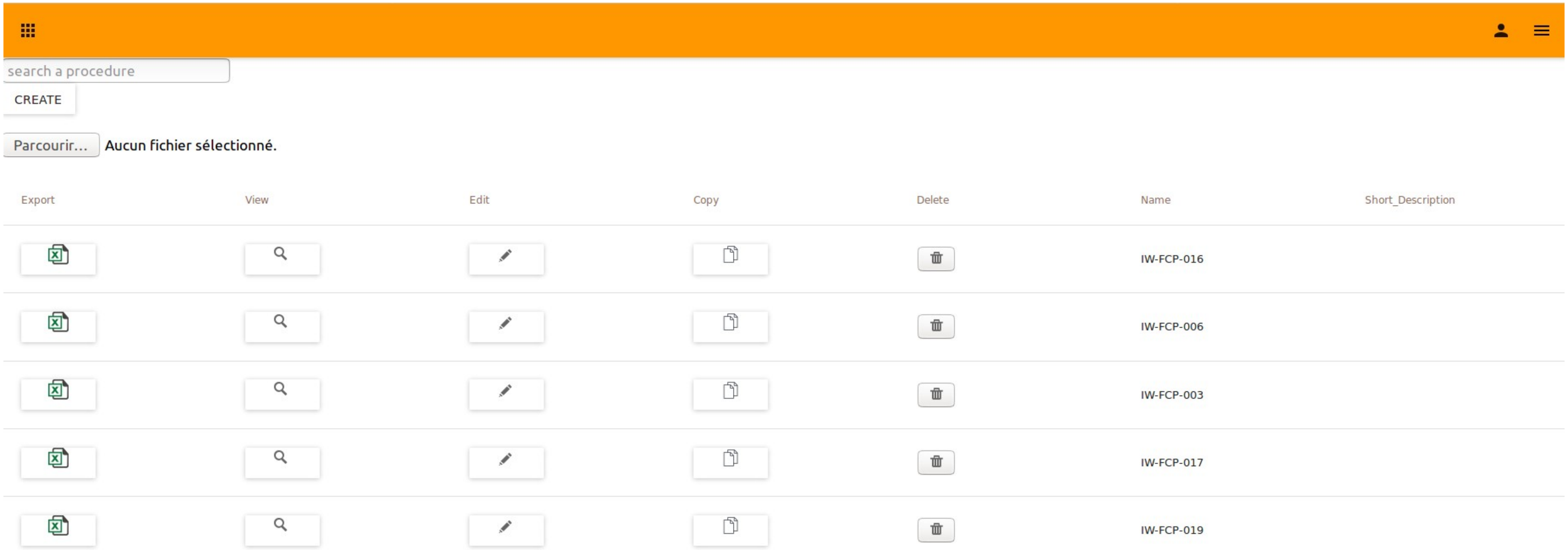
Filter by ending datetime :

Ending DateTime

Filter by packet type :

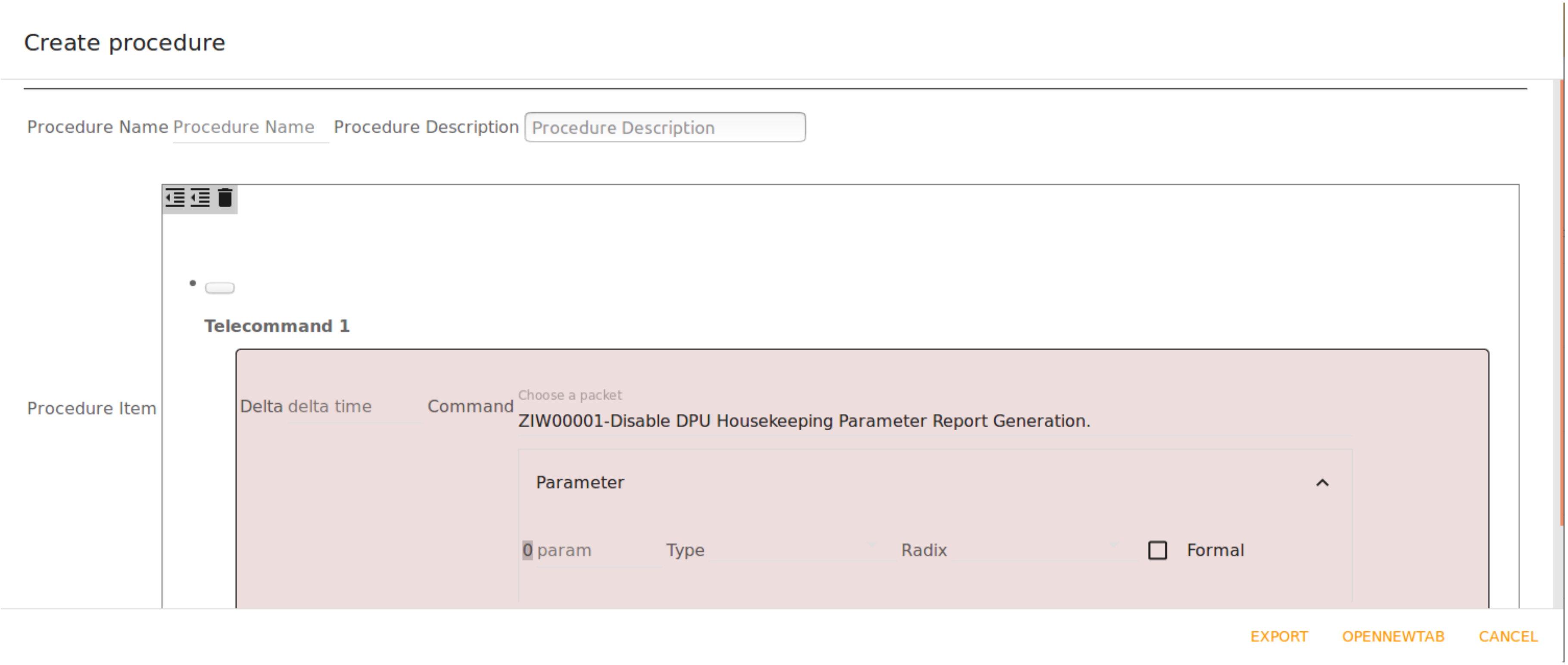
TM

GUI design – Figaro (main page)



Export	View	Edit	Copy	Delete	Name	Short_Description
☒	🔍	✎	🗐	🗑	IW-FCP-016	
☒	🔍	✎	🗐	🗑	IW-FCP-006	
☒	🔍	✎	🗐	🗑	IW-FCP-003	
☒	🔍	✎	🗐	🗑	IW-FCP-017	
☒	🔍	✎	🗐	🗑	IW-FCP-019	

GUI design - Figaro (editing window)



Create procedure

Procedure Name Procedure Name Procedure Description Procedure Description

Procedure Item

•

Telecommand 1

Delta delta time Command Choose a packet ZIW00001-Disable DPU Housekeeping Parameter Report Generation.

Parameter

0 param Type Radix □ Formal