

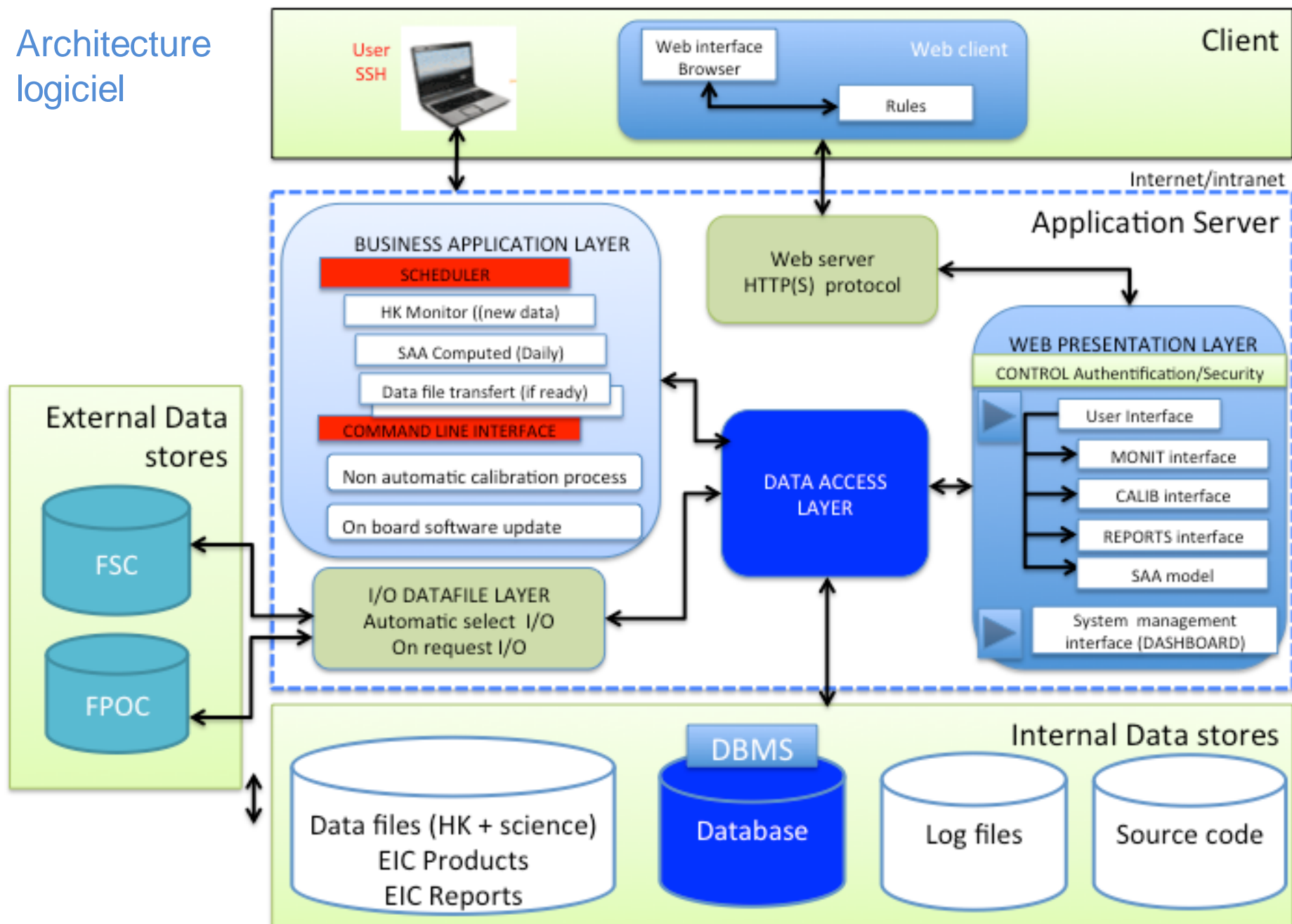
# Eclairs Instrument Center

L'EIC sera opérationnel 24h/24

- Réaliser la surveillance de l'instrument sur le moyen et long terme et du logiciel de bord
- Quantifier l'évolution des performances
  - Suivre l'évolution des performances du trigger bord
- Mettre à jour les fichiers de configuration bord et auxiliaires
  - Effectuer la maintenance du logiciel bord
- Définir les contours de la SAA vue par Eclairs
- support scientifique aux BAs

**M. Boiziot, L. Bouchet, P. Maeght de la part de l'équipe EIC**

# Architecture logiciel



# + PLAN DE DEVELOPPEMENT/ CALENDRIER

Bafghera: EIC-PL-A32-006-IRAP 1/0 ECLAIRs Instrument Center Development Plan

Redmine: <https://forge.in2p3.fr/dmsf/files/7053/view>

Start of the interfaces/system tests

Launch at the end of 2021

WP	Reference	Name	PHASE C												PHASE D				PHASE E1
			2017		2018				2019				2020				2021	2022	
			T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1		
			♦ PDR				♦ V pré-alpha				♦ V beta								
1	EIC-MGT-IRAP-001	Scientific coordination																	
	EIC-MGT-IRAP-002	Project management & Documentation & Quality																	
3	EIC-SW-IRAP-003	Medium and Long term monitoring of the status of the instrument																	
4	EIC-SW-IRAP-004	Calibration of the instrument in operation																	
5	EIC-SW-IRAP-005	Calibration of the instrument on ground																	
6	EIC-SW-IRAP-006	Update of the SAA contour (seen by ECLAIRs )																	
7	EIC-SW-IRAP-007	Management of On-Board Software																	
8	EIC-SW-IRAP-008	Monitoring of the on-board scientific Trigger																	
9	EIC-SW-IRAP-009	Management of instrumental configurations (TCs) according to the WorkPlan (ToO)																	
10	EIC-SW-IRAP-010	Support to scientists and Burst Advocates (BA)																	
11	EIC-SW-IRAP-011	Tests and simulations on the spare model																	
12	EIC-SW-IRAP-012	Input/output, Data Base, Web site management																	
13	EIC-SW-IRAP-013	Information system management (Monitoring of the application)																	
14	EIC-SW-IRAP-014	Support and maintenance of hardware, software (monitoring, upgrade, update, modifications)																	
15	EIC-SW-IRAP-015	Data simulators																	
16	EIC-VAL-IRAP-016	Functional acceptance tests, integration, internal validation tests																	
17	EIC-VAL-IRAP-017	Interfaces tests																	
18	EIC-VAL-IRAP-018	Delivery & System tests																	

Transverse activity which must be continuously maintained		
Development and tests of specific WP		
Correction and completion to obtain the final version		
Planned margin		

# + PLAN DE DEVELOPPEMENT ET TESTS

- Le développement de l'EIC se fait parallèlement au développement de l'instrument.
  - La méthode adoptée est le développement « incrémental ».
- Une première version (MONIT, CALIB) a été implémentée, en partie, pour les périodes de tests avec le prototype du DPIX (1/8 du plan). Une autre version sera mise à jour pour les tests proto-EQM DPIX, puis pour les tests sur les modèles FM (DPIX & ECLAIRS).
- Rédaction des IRDs et des ICDs (EIC-FSC, EIC-FPOC)
- Les tests d'interface seront gérés avec les centres FSC et FPOC.
  - Spécifications des boucles système testées, hypothèses de travail, degré de représentativité des fichiers échangés et un planning (au niveau du FGS)
- Des simulateurs pour valider les processus (spécifications en cours)

# + PRIORITES 2019

## ■ Finaliser les interfaces FSC-EIC / FPOC-EIC

- Utilisation du format fits défini pour stocker les données de calibration proto :
  - Création d'une première version de certains fichiers de calibration
  - Jeux de données/paramètres "réalistes" disponibles pour tester les pipelines

## ■ Monitoring [P Maeght]

## ■ Site Web, base de données interne EIC [M. Boiziot]

## ■ Logiciels bord (interface avec le CEA)

- Comment, quelles interfaces?
- Maintenance du logiciel de bord (WP 7)
- Surveillance du trigger scientifique bord (WP 8)
- Discussion avec l'équipe système ECLAIRs à prévoir

## ■ Logiciels scientifiques

## ■ Participation aux tests d'intégration et d'interfaces avec les autres centres

## ■ Mise en pratique au cours des phases d'étalonnage sol (tests & simulations)

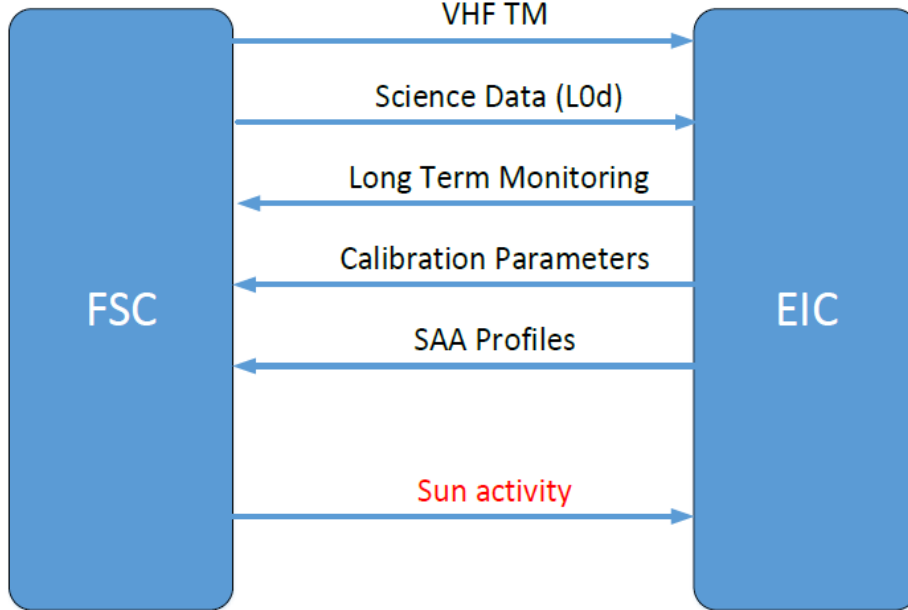
- Adaptation des interfaces et tests de certains outils/procédures prévus pendant le vol
- Ce travail sera réalisé en collaboration avec l'équipe de développement de l'EGSE ECLAIRs

## ■ Serveur EIC, hardware, logiciels

- Serveur en commande

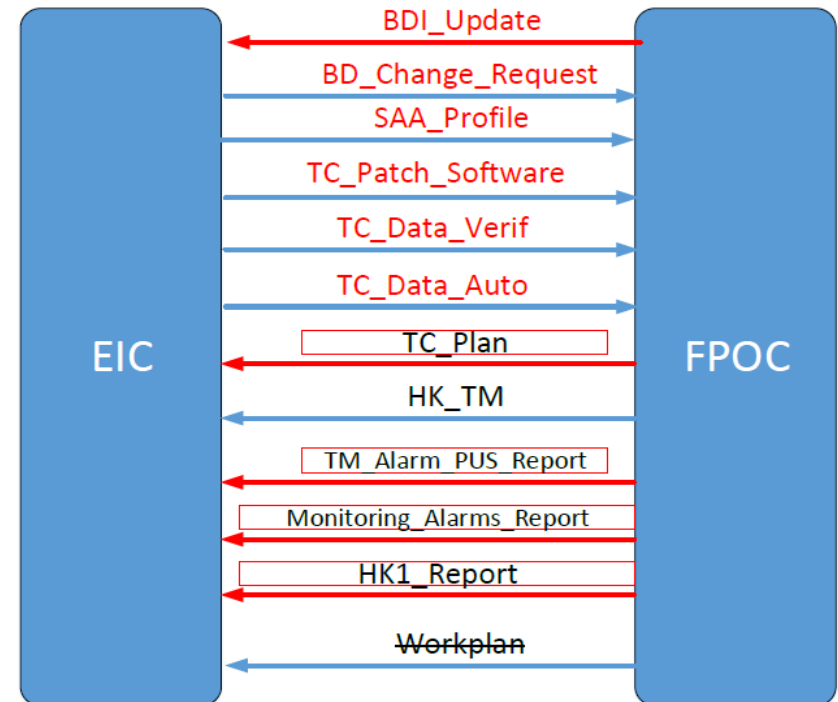
# + INTERFACES

Baghera: EIC-SP-A32-002-IRAP: 4/0 -Spécifications Techniques de Besoins de l'EIC



## ■ FPOC-EIC

### ■ Fichier XML pour SAA



## ■ FSC-EIC

### ➤ « Détails » format des fichiers échangés

- [Baghera: EIC-SN-A32-022-IRAP (1/0) - ECLAIRs Instrument Center - FITS Keywords (V1.1) ]

# + Produits EIC

7

## ● Traitement des données ECLAIRS / configuration DPIX

### Calibration (CALIB) – Court/moyen terme

- Conversion canal/énergie
  - Table de gain & offset
  - Table bornes des bandes d'énergie
- Table/Carte d'uniformité imagerie
- Table des seuils ASIC – seuil bas (SBN)
- Table des pixels bruyants
- Table des pixels morts

### Calibration (CALIB) – Long terme

- Coefficients correction biais fixes imagerie
- Corrections déformations thermo-élastiques
- Matrice de non-alignement ("misalignment")

- *Response Matrix Files* : Réponse spectrale DPIX
- *Ancillary Response File*: Réponse masque codé
- *Instrument Response File* : Réponse angulaire

## ● Autres fonctions

- Inter calibration Eclairs / GRM / MXT / instruments HE
- Bruit de fond sur l'orbite
- Correction du temps mort

## ● Configurations (UGTS)

### Surveillance trigger scientifique (TRIG)

- Table des seuils
- Catalogue de sources

## ● Gestion entrée-sortie SAA

### Contours de la SAA (MSAA)

## ● Configurations (DPIX & UGTS)

### Maintenance logiciel bord (MSOFTV)

- *Update Software* (trigger + UGTS)

## ● Support BAs (BASUP)

### Liste des IS Eclairs pour support BAs

## ● Documentation

- à destination de SVOM

## Statut/ Etat des lieux/Objectifs 2019

- SAA (Stage ingénieur 2018, implémenté, fourniture des premières cartes sous forme de « polygones » au FPOC pour analyse système)
- SBN (Stage M2 LUPM en 2018 + calibration proto (JP Dezalay):
  - création de la première table de seuils SBN (seuil @ 4 keV), testé sur le proto et en cours d'analyse)
- Gain & offset (Stage M2 IRAP 2019 + calibration proto (JP Dezalay))
- Gestion des pixels morts & bruyants (calibration proto (O Godet & A Bajat))
- Réponses spectrale & temporelle:
  - Travaux de A Bajat: A partir de données expérimentales & simulation avec « ElSimulator »
    - 1<sup>er</sup> modèle physique de la réponse, modèle de temps morts validé sur le proto.
- Travaux en interface avec l'équipe DPIX, calibration et l'EIC
  - Environnement de simulation GEANT4 des différents dispositifs expérimentaux testés au sol (du proto-DPIX à ECLAIRS) – version proto-DPIX existante
  - Préparation des essais et support aux analyses
- Simulateurs bruit dynamique





# Développement sur le site web... ...en 2018

- ✓ Mise en place de la base du site
- ✓ Inscription
- ✓ Gestion des utilisateurs et droits
- ✗ Gestion des astreintes
- ✗ Visualisation des HKs (partiel)
- ✗ Base de données interne à l'EIC (partiel mais bien avancé)
- ✓ Plateforme d'intégration continue et documentation
- ✓ Déploiement sur système automatique via docker



 [add description](#)

All 

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		<a href="#">EIC_WEBSITE</a>	16 days - <a href="#">#3</a>	N/A	52 sec	
		<a href="#">EIC_WEBSITE_DEPLOY</a>	16 days - <a href="#">#15</a>	3 mo 22 days - <a href="#">#12</a>	3 min 52 sec	
		<a href="#">EIC_WEBSITE_TEST</a>	5 mo 10 days - <a href="#">#14</a>	5 mo 12 days - <a href="#">#11</a>	28 sec	
		<a href="#">Pipeline-bricks-targz-exporter</a>	7 days 19 hr - <a href="#">#126</a>	7 mo 1 day - <a href="#">#8</a>	11 sec	
		<a href="#">PipelineBricks</a>	7 mo 0 days - <a href="#">#4</a>	N/A	19 sec	
		<a href="#">PythonTestJenkins</a>	5 mo 13 days - <a href="#">#13</a>	7 mo 15 days - <a href="#">#9</a>	10 sec	

Quality Gate **Passed**

Bugs Vulnerabilities

Leak Period: since previous version started 4 months ago

0 <sup>A</sup> Bugs	0 <sup>A</sup> Vulnerabilities	0 <sup>A</sup> New Bugs	0 <sup>A</sup> New Vulnerabilities
------------------------	-----------------------------------	----------------------------	---------------------------------------

Code Smells

3h <sup>A</sup> Debt <small>started 4 months ago</small>	10 Code Smells	0 <sup>A</sup> New Debt	0 New Code Smells
--	-------------------	----------------------------	----------------------

Coverage

60.0% Coverage	— Coverage on New Code
-------------------	---------------------------

Duplications

0.0% Duplications	0 Duplicated Blocks	— Duplications on New Code
----------------------	------------------------	-------------------------------

1.3k Lines of Code Python 1.3k

No tags

Activity

January 7, 2019 1.0

October 3, 2018 Project Analyzed Show More

Quality Gate (Default) SonarQube way

Quality Profiles (Python) Sonar way

Key EIC\_WEBSITE

Leak Period: since previous version

Risk

Color: Worse of Reliability Rating and Security Rating Size: Lines of Code

A B C D E



## Table Of Contents

Welcome to EIC's documentation!

- Introduction
- Code
  - Localization
  - Documentation
  - PyBuilder
- Setup
  - Dependencies
  - Start the app for development

Indices and tables

## This Page

Show Source

## Quick search

# Welcome to EIC's documentation!

## Introduction

This is the documentation for the website of the EIC project. The website is developed in Python 3.5 with the framework Flask.

This documentation contains the quick start guide to start developping on the project and the code documentation on the [Module Index](#).

## Code

## Localization

The website code is in english however it can be rendered in both french and english. To be able to do that, [Babel](#) and [Flask-Babel](#) packages are used.

Two bash scripts script are used to simplify the use of Babel: `update_lang.sh` and `complil_lang.sh`

- *update\_lang.sh* will extract all the uses of Babel inside the application (both the source code and the template) and create a file referencing them. After that update the localizations files are located in the translations folder. After the use of this script, an human intervention may be needed to update the translation.
- *compil\_lang.sh* will use the translations then compile them into a Babel file. It need to be called each time a new translation is done.

## Documentation

The main documentation for the EIC website is written using [Sphinx](#). It will parse the doc inside the Python file then render it in HTML.

The configuration files are located in the docs folder. The documentation need to be build. This will be done with the use of PyBuilder. This part is described latter in the document.

## PyBuilder

[PyBuilder](#) is a build tool similar to Maven for Python applications. The `build.py` file located at the root of repository is used to configure it.

Currently it allow to build the documentation. To do that, just use the `pyb` command on the root of the repository.

## Setup

## Dependencies

The dependencies are located in two files:

Quick search

Go

## Python Module Index

e

**e**

[-]

**eic**

- `eic.db`
- `eic.db.account`
- `eic.db.database`
- `eic.db.duty`
- `eic.db.group`
- `eic.db.invitation`
- `eic.duty`
- `eic.duty.calendar`
- `eic.duty.duty`
- `eic.duty.month`
- `eic.duty.months`
- `eic.duty.utils`
- `eic.exception`
- `eic.exception.missingConfigValueError`
- `eic.exception.missingFormValueError`
- `eic.exception.userAlreadyExistError`
- `eic.user`
- `eic.user.group`
- `eic.user.invitation`
- `eic.user.user`
- `eic.utils`
- `eic.utils.config`
- `eic.utils.locale`
- `eic.utils.sendmail`

## eic.duty.duty module

`class eic.duty.duty.Duty(identifier: int, user: eic.user.user.User, start_date: datetime.date, end_date: datetime.date, role: int)`

Bases: `object`

[\[source\]](#)

Class describing a duty.

`static create_in_db(user_id: int, start_date: datetime.date, end_date: datetime.date, role: int)`

[\[source\]](#)

Create the duty the the database.

- Parameters:**
- **user\_id** - The account identifier of the User on duty.
  - **start\_date** - The start date of the duty.
  - **end\_date** - The end date of the duty.
  - **role** - The role of the user in the duty (0 = Primary, 1 = Backup).

**Returns:** Nothing.

`static edit_in_db(duty_id: int, user_id: int, start_date: datetime.date, end_date: datetime.date, role: int)`

[\[source\]](#)

Update the duty with the given identifier with the given values.

- Parameters:**
- **duty\_id** - The identifier of the duty to update.
  - **user\_id** - The user identifier to assign to the duty.
  - **start\_date** - The start date of the duty.
  - **end\_date** - The end date of the duty.
  - **role** - The role of the user in the duty (0 = Primary, 1 = Backup).

**Returns:** Nothing.

`get_formatted_end_date()` → str

[\[source\]](#)

Compute then return the end date formatted as yyyy-mm-dd.

**Returns:** the end date formatted as yyyy-mm-dd str.

`get_formatted_start_date()` → str

[\[source\]](#)

Compute then return the start date formatted as yyyy-mm-dd.

**Returns:** the start date formatted as yyyy-mm-dd str.

`get_user_email()` → str

[\[source\]](#)

Return the email of the user on duty.

**Returns:** the email of the user on duty.

`get_user_name()` → str

[\[source\]](#)

Return the complete name of the user doing the duty.

**Returns:** the complete name of the user doing the duty.

# Développement sur le site web...

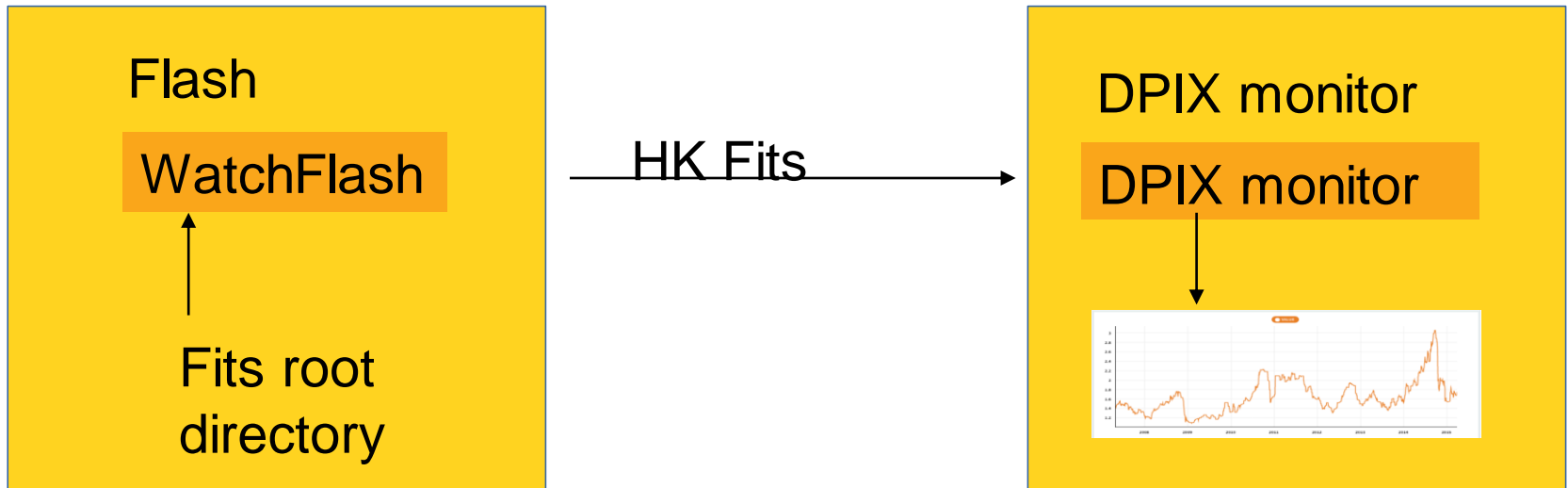
## ... en 2019

- × Finalisation de la visualisation des HKs
  - Gestion des alarmes et rapport d'alarmes
  - Génération des rapports d'états
  - Logging / surveillance
- × Finalisation de la BDD interne à l'EIC
- × Finalisation de divers modules
  - × Besoin de spécifications techniques des interfaces / protocoles d'échange



# EIC proto DPIX monitor

2018

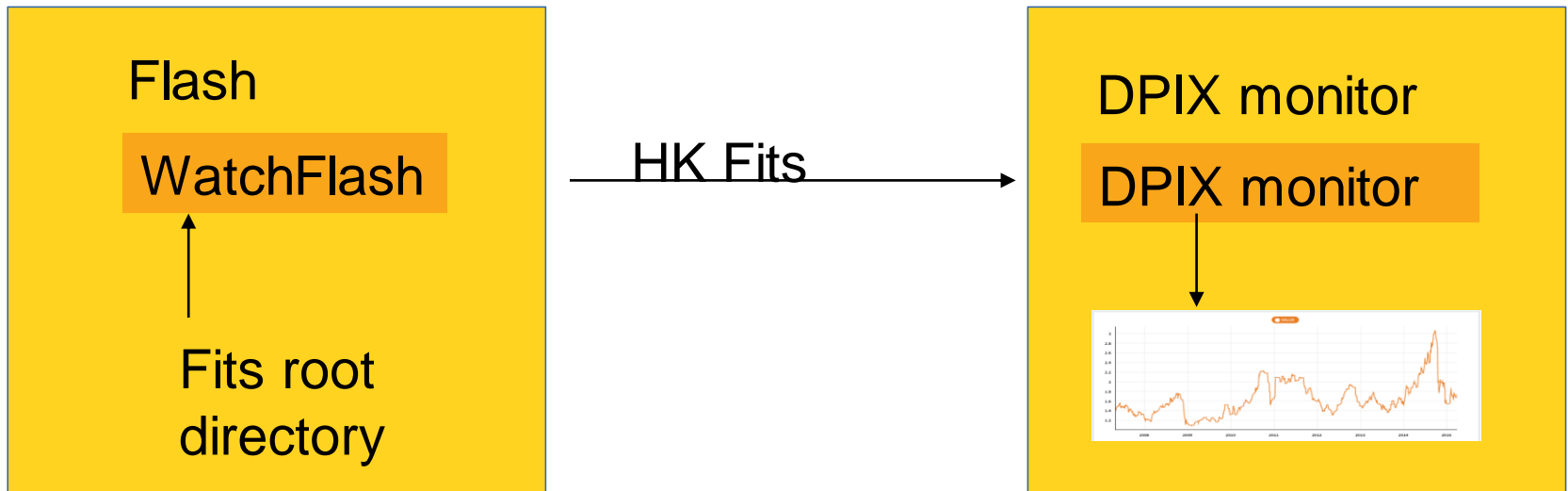


- Les FITS déposés sur le disque sont automatiquement envoyés à l'application de monitor.

- Les FITS recus sont analysés et ingérés dans une base de données locale..

# EIC proto DPIX monitor

2019



- Installation de l'application.

- Interface web du monitoring.

# WebSocket

WS state

Status: ⓘ Event: Aston Martin DB9

WebSocket is opened 🔥

## Apache:

Https, wss : <VirtualHost eic1.irap.omp.eu:443>

Http, ws : <VirtualHost eic1.irap.omp.eu:80>

SSLProxyEngine On

ProxyPass /nats\_pub/ http://127.0.0.1:5010/nats\_pub/

ProxyPassReverse /nats\_pub/ http://127.0.0.1:5010/nats\_pub/

ProxyPass /ws\_pub ws://127.0.0.1:5678/ws\_pub

ProxyPassReverse /ws\_pub ws://127.0.0.1:5678/ws\_pub

http:  
127.0.0.1:5010

ws:  
127.0.0.1:5678

Appli Python

# Environnement de développement



+ MERCI !