

<http://aflak.jp>

aflak  
アフラーク

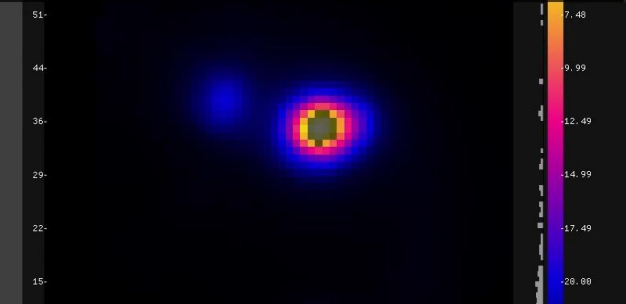
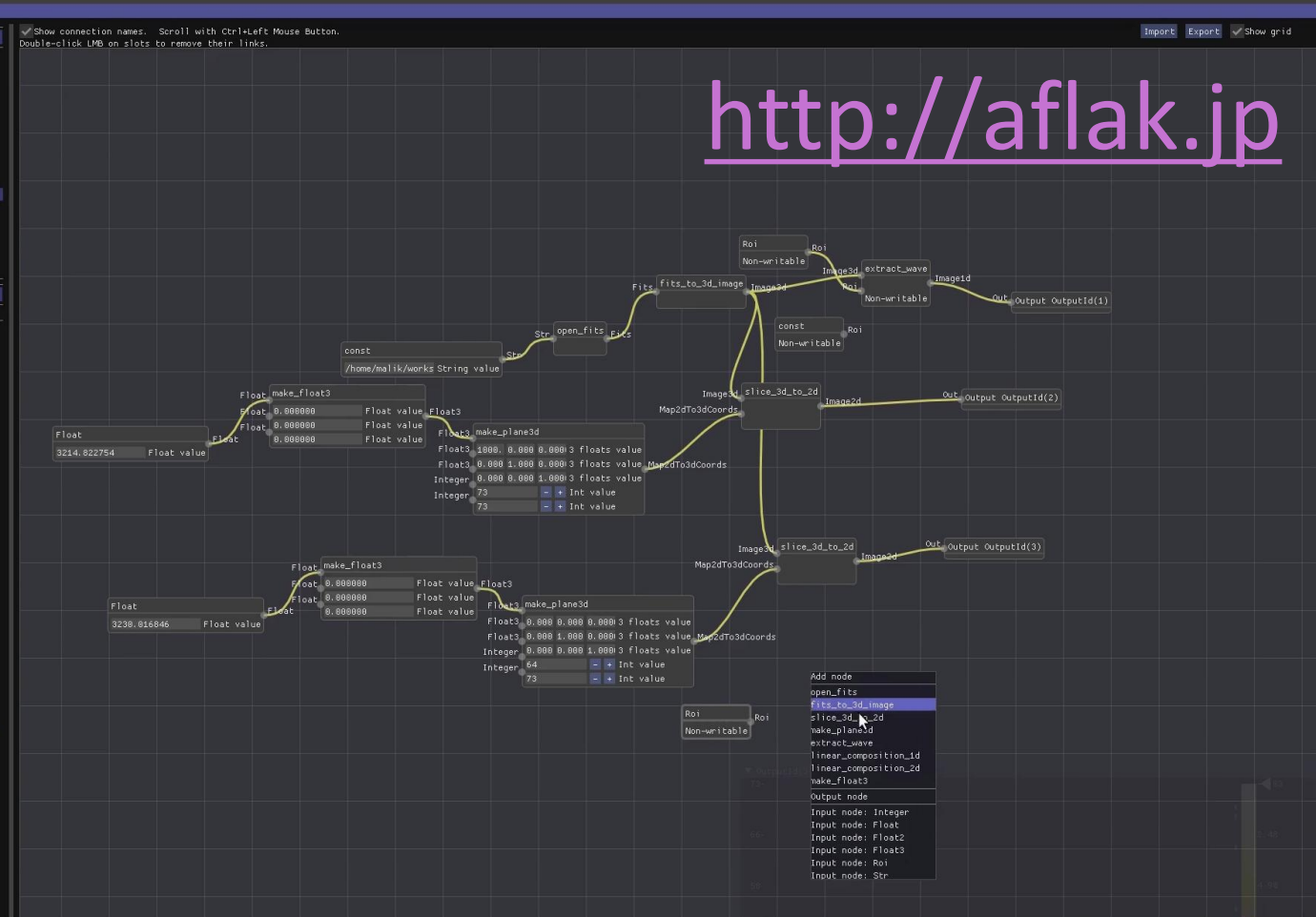
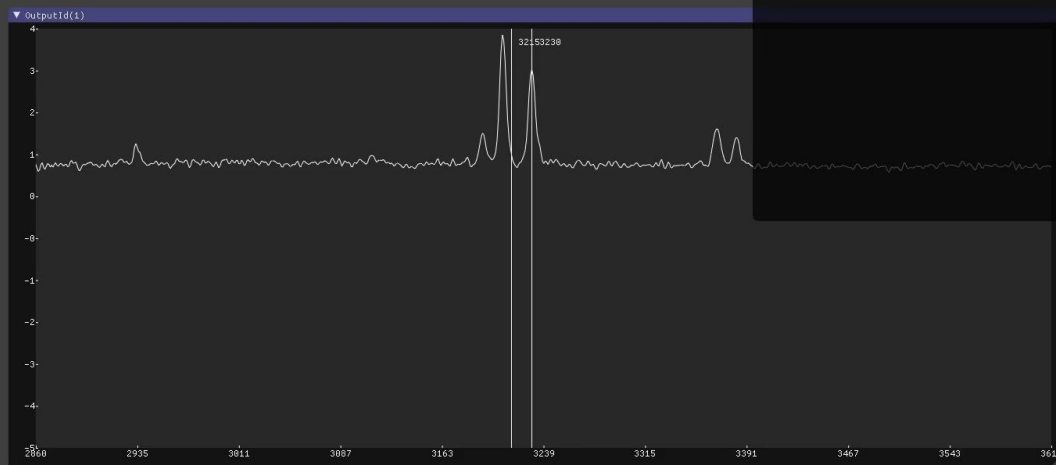
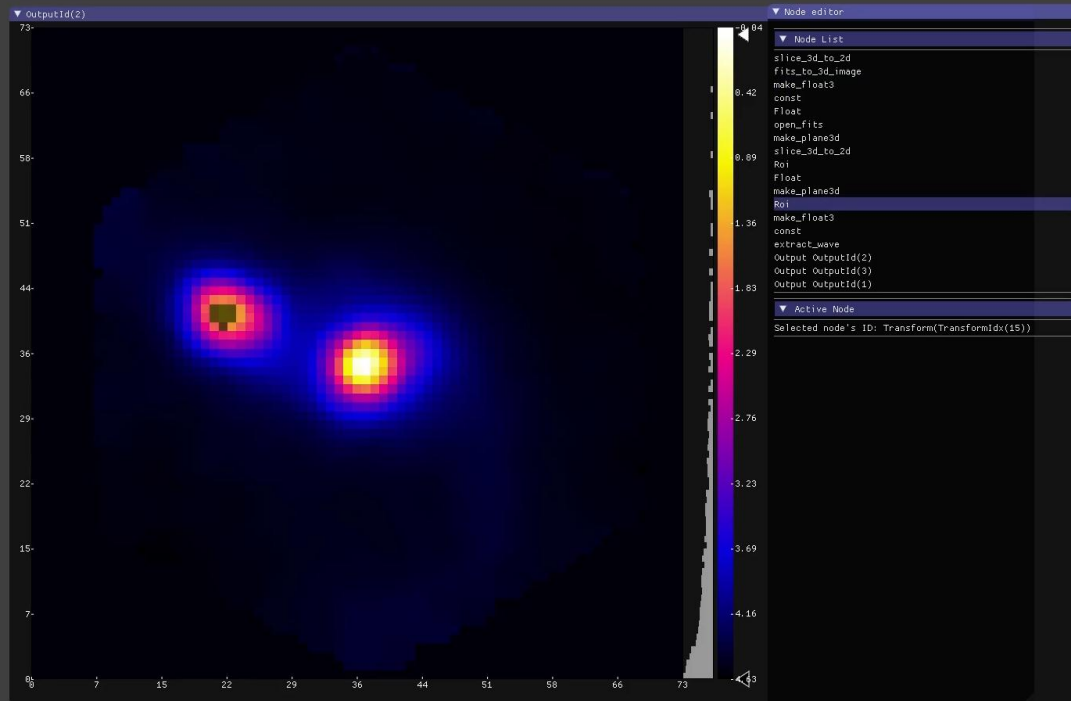
# Advanced Framework for Learning Astrophysical Knowledge

分光データ解析用ビジュアルプログラミング言語

木本 真理究 (Malik Olivier Boussejra)<sup>1</sup>・竹川 俊也<sup>2</sup>・内木 陸雄<sup>1</sup>・松林 和也<sup>3</sup>・竹島 由里子<sup>4</sup>・植村 誠<sup>5</sup>・藤代 一成<sup>1</sup>

<sup>1</sup>慶應義塾大学 <sup>2</sup>国立天文台 <sup>3</sup>京都大学 <sup>4</sup>東京工科大学 <sup>5</sup>広島大学

[malik@boussejra.com](mailto:malik@boussejra.com)



QUICK DEMO: <https://vimeo.com/290328343>

# RUST

[Documentation](#)[Install](#)[Community](#)[Contribute](#)

**Rust** is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

[Install Rust 1.30.1](#)

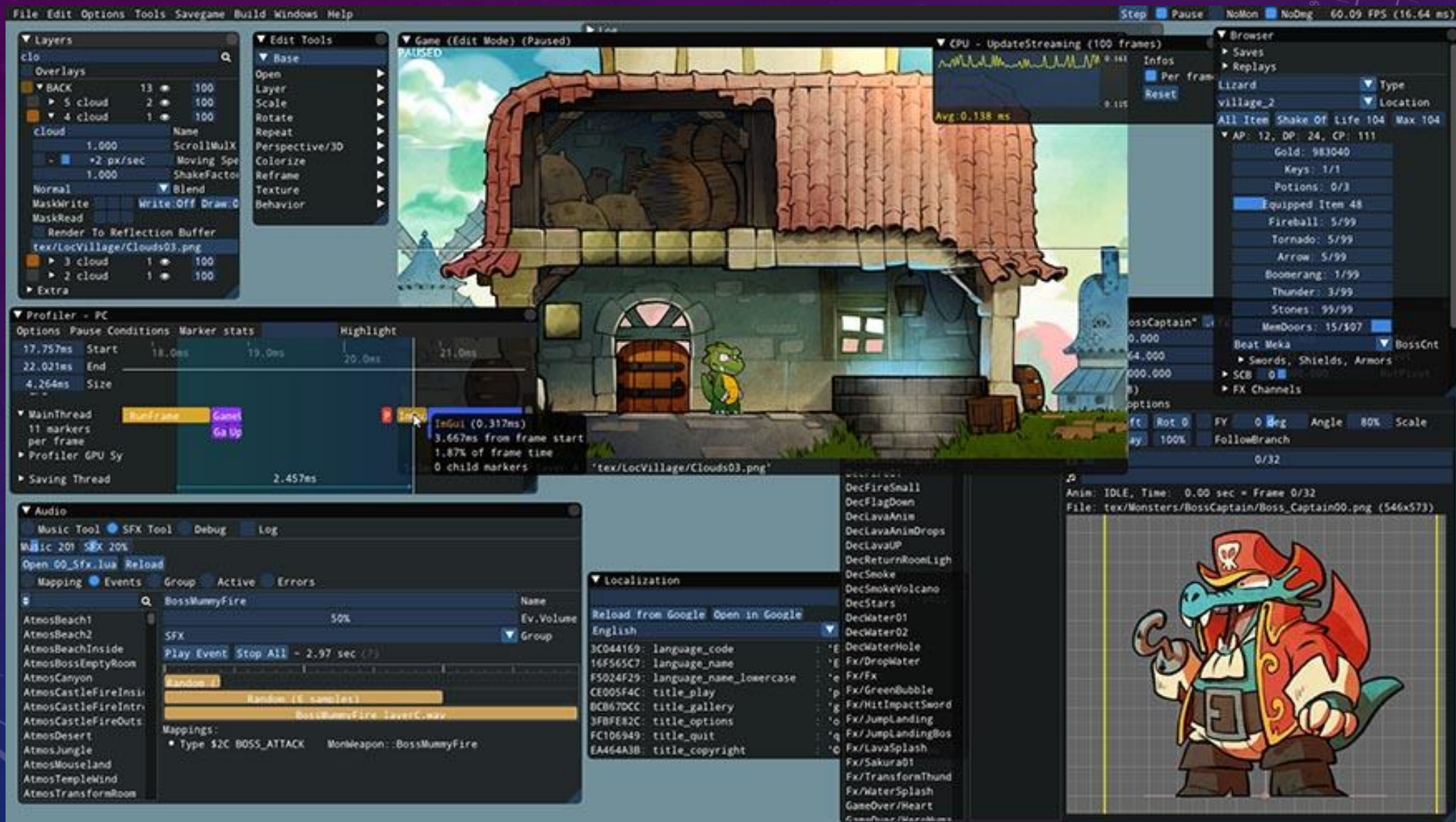
November 8, 2018

<https://www.rust-lang.org/>



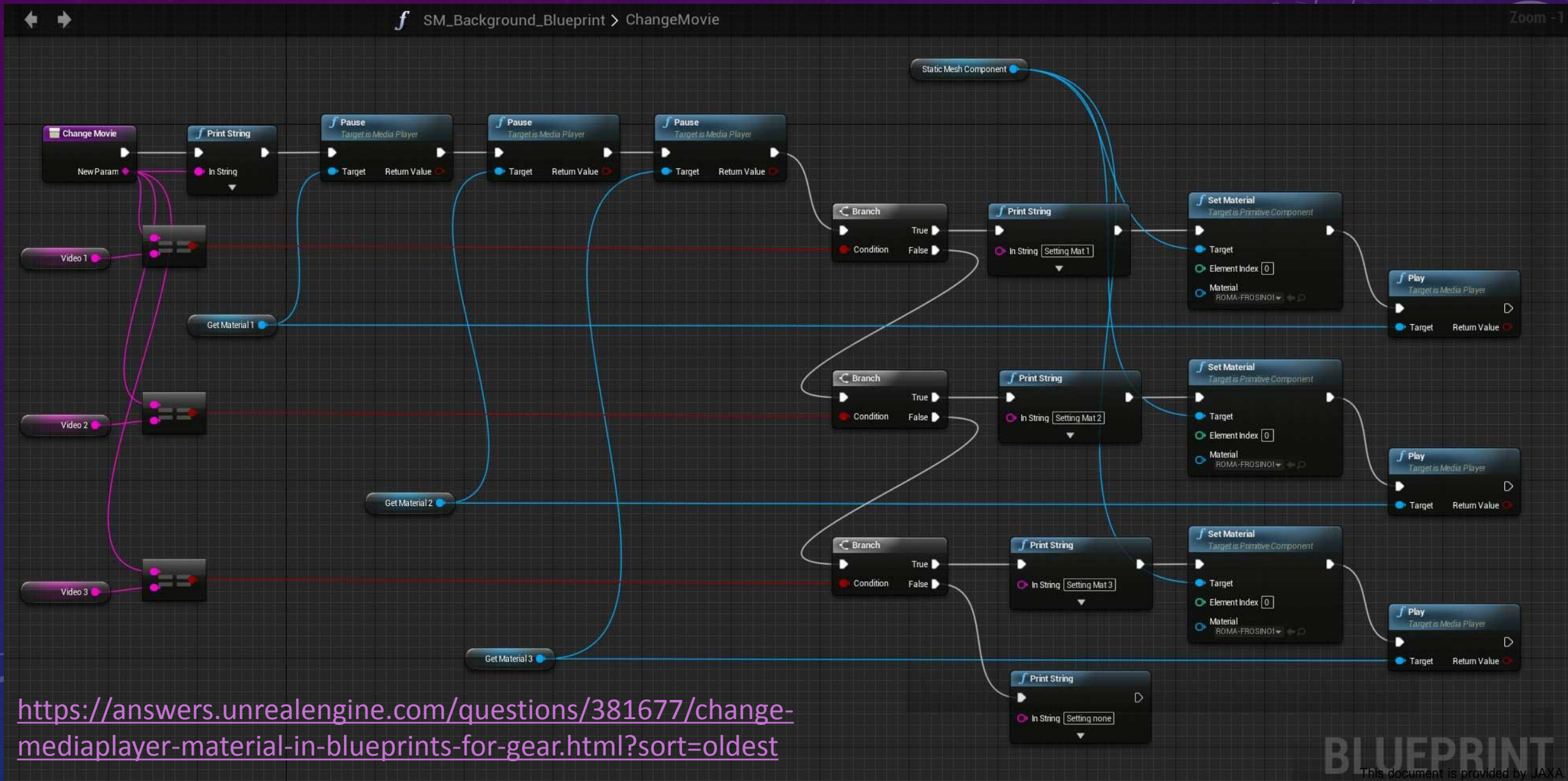
# DEAR IMGUI (C++)

<https://github.com/ocornut/imgui>





# BLUEPRINTS – UNREAL ENGINE

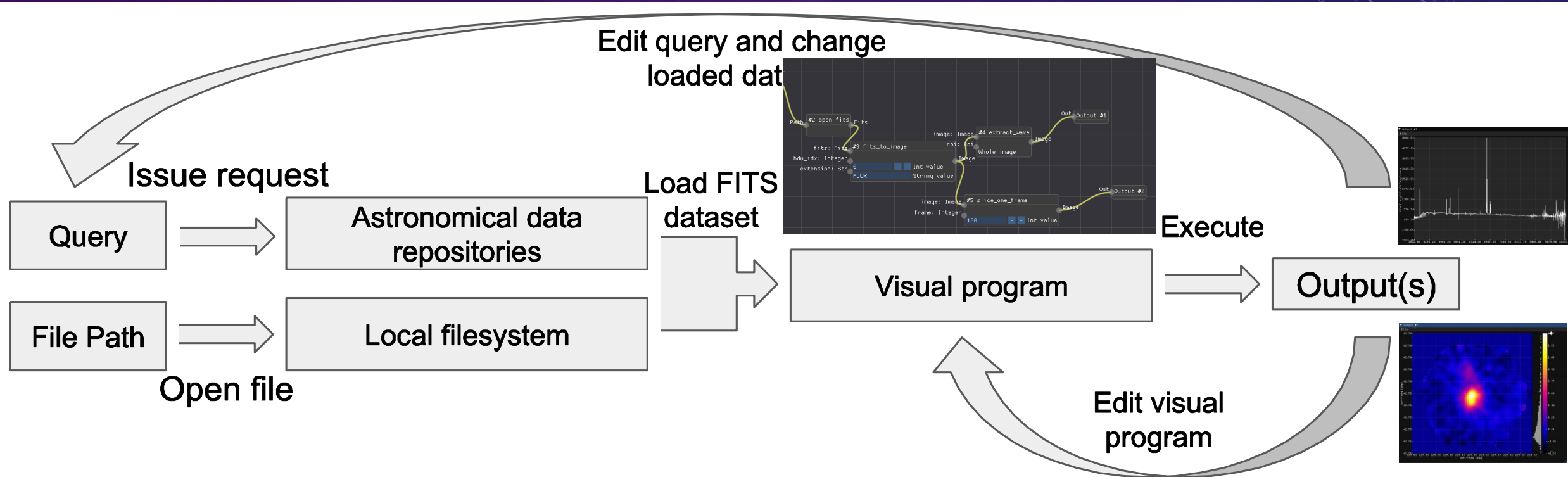


<https://answers.unrealengine.com/questions/381677/change-mediaplayer-material-in-blueprints-for-gear.html?sort=oldest>

# 経緯

- February 2018 – START
- October 2018 – SciVis – aflak: Pluggable Visual Programming Environment with Quick Feedback Loop Tuned for Multi-Spectral Astrophysical Observations in Proceedings of IEEE Scientific Visualization Conference
- November 2018 – ADASS – aflak: Visual Programming Environment with Quick Feedback Loop, Tuned for Multi-Spectral Astrophysical Observations, in Proceedings of ADASS 2018
  - VO standards are important
- April 2019 – PacificVAST – aflak: Visual Programming Environment Enabling End-to-End Provenance Management for the Analysis of Astronomical Datasets [TBP]
- aflak: モジュール可視化環境による等価幅マップの生成 日本天文学会 2019 年春季年会 [TBP]

# WORKFLOW



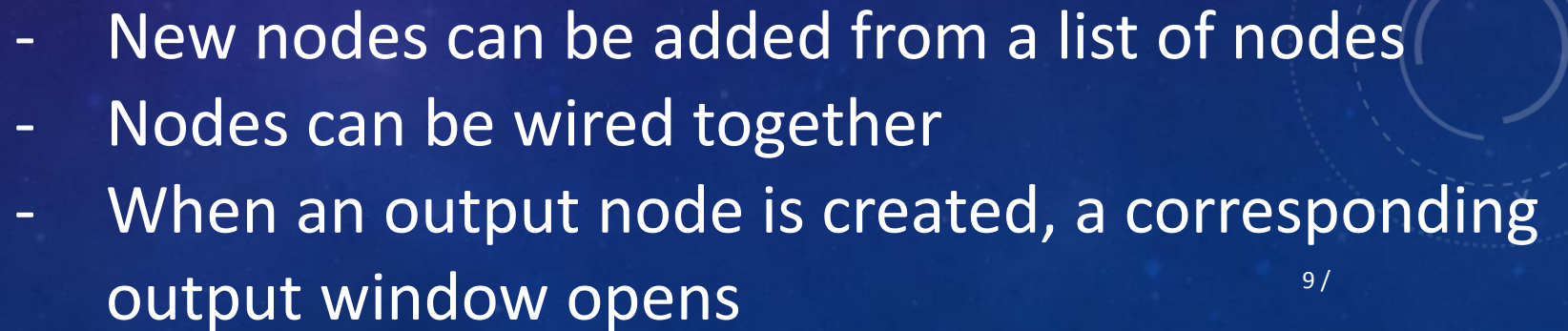


<http://aflak.jp>



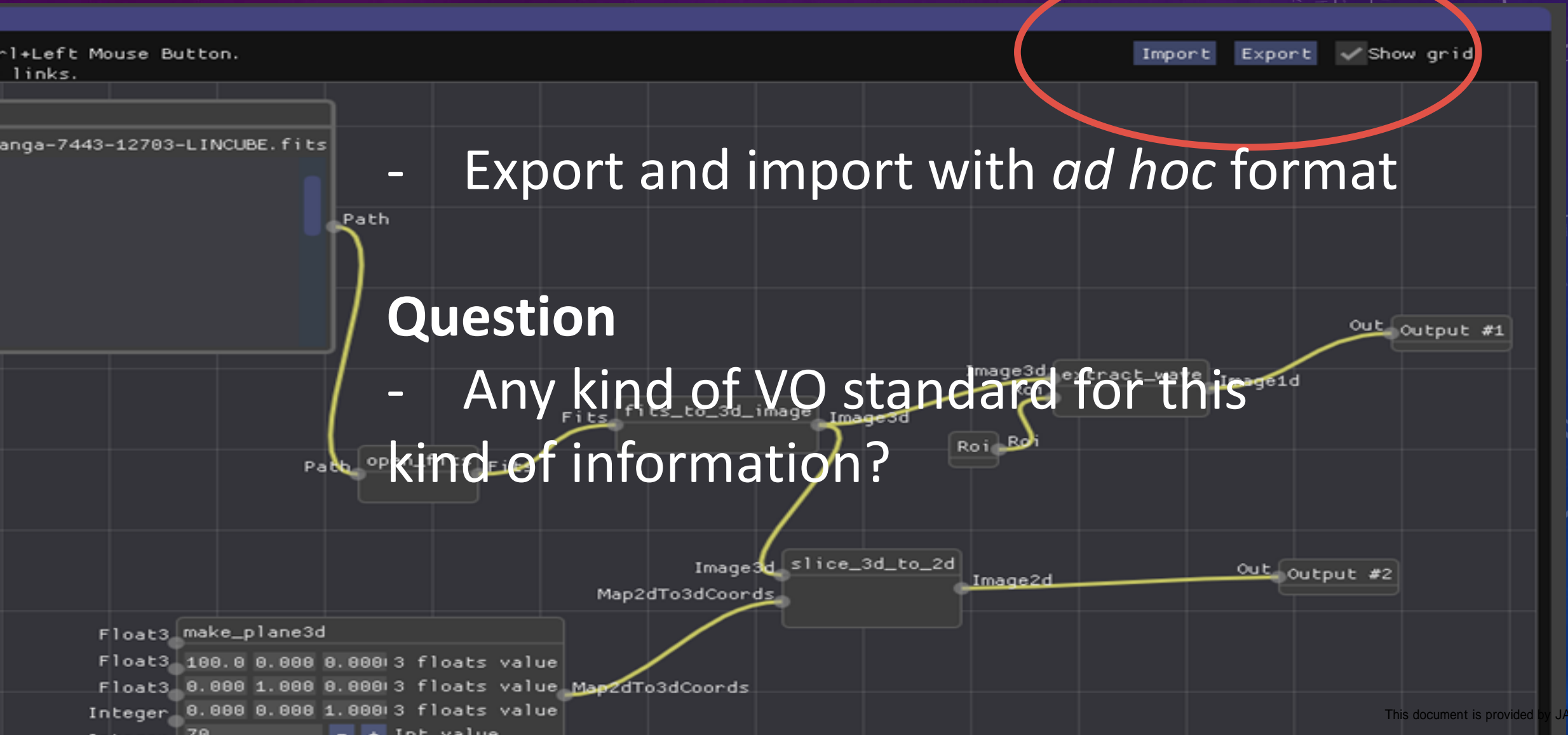


<http://aflak.jp>



# IMPORT / EXPORT OF NODE PROGRAM

<http://aflak.jp>

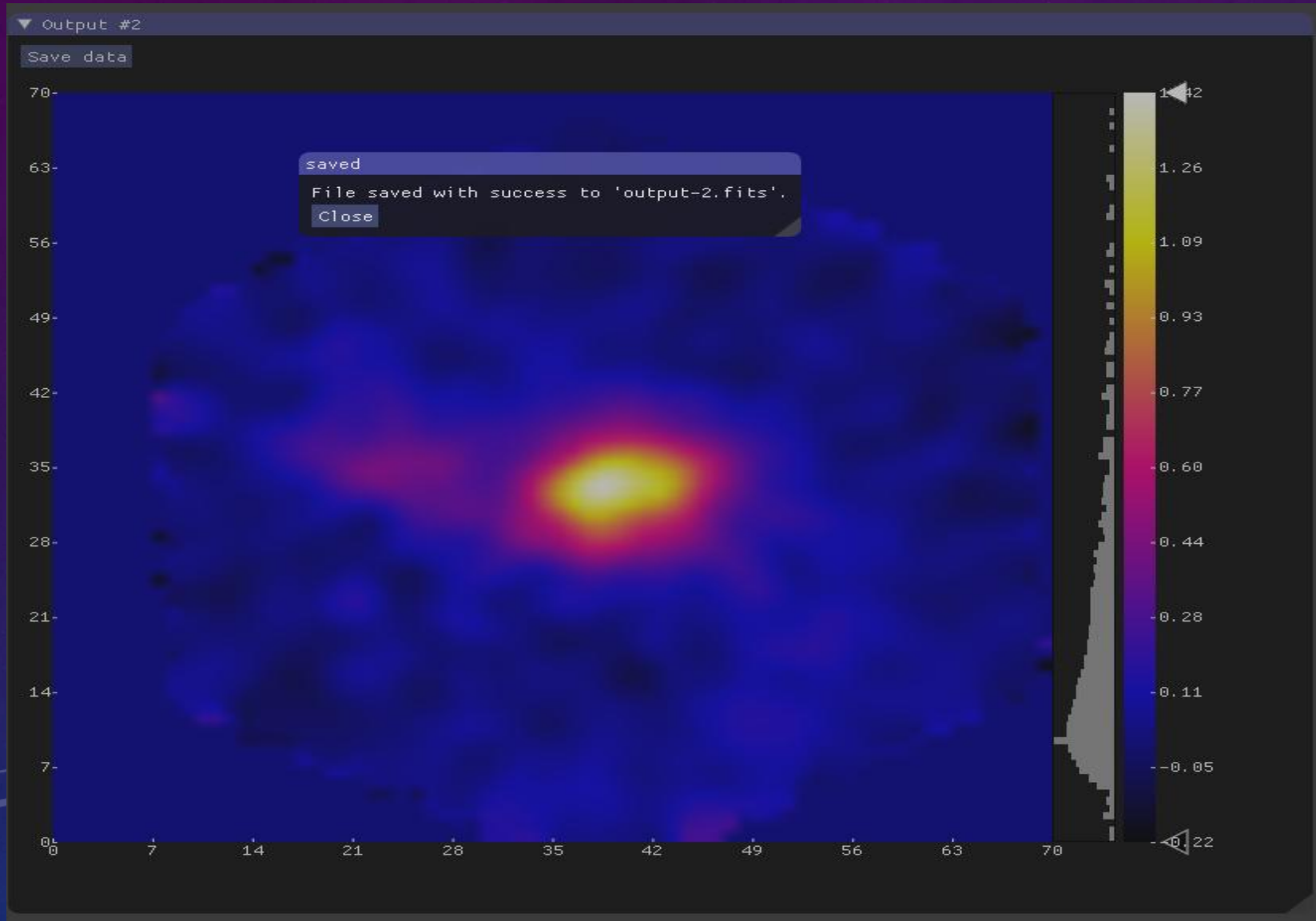


- Export and import with *ad hoc* format

## Question

- Any kind of VO standard for this kind of information?

SAVE OUTPUT AS FITS FILE CONTAINING  
END-TO-END LINEAGE OF HOW THE DATA WAS CREATED



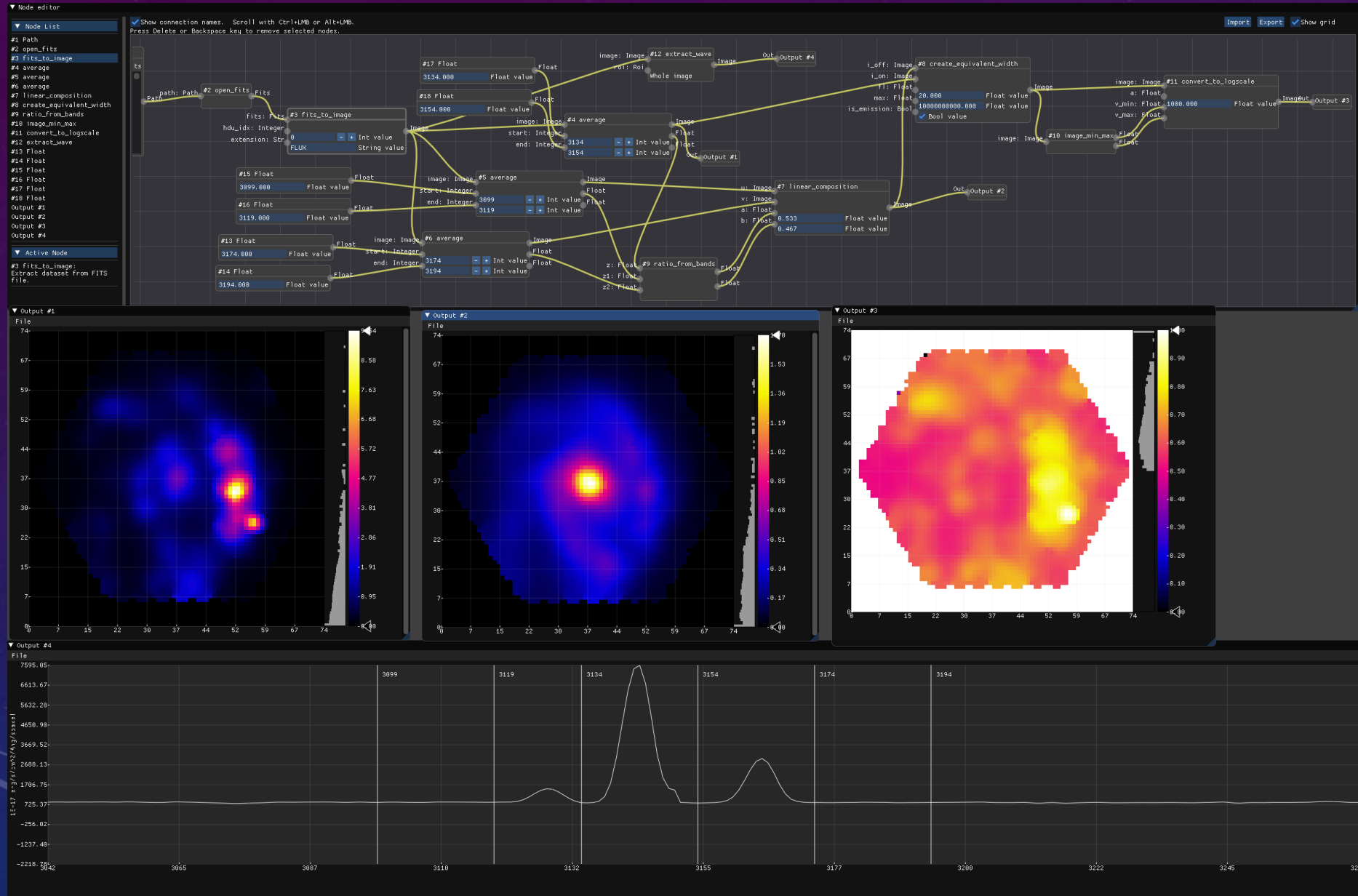
**Question**

Any kind of  
“standard”  
procedure  
for recording  
data provenance?

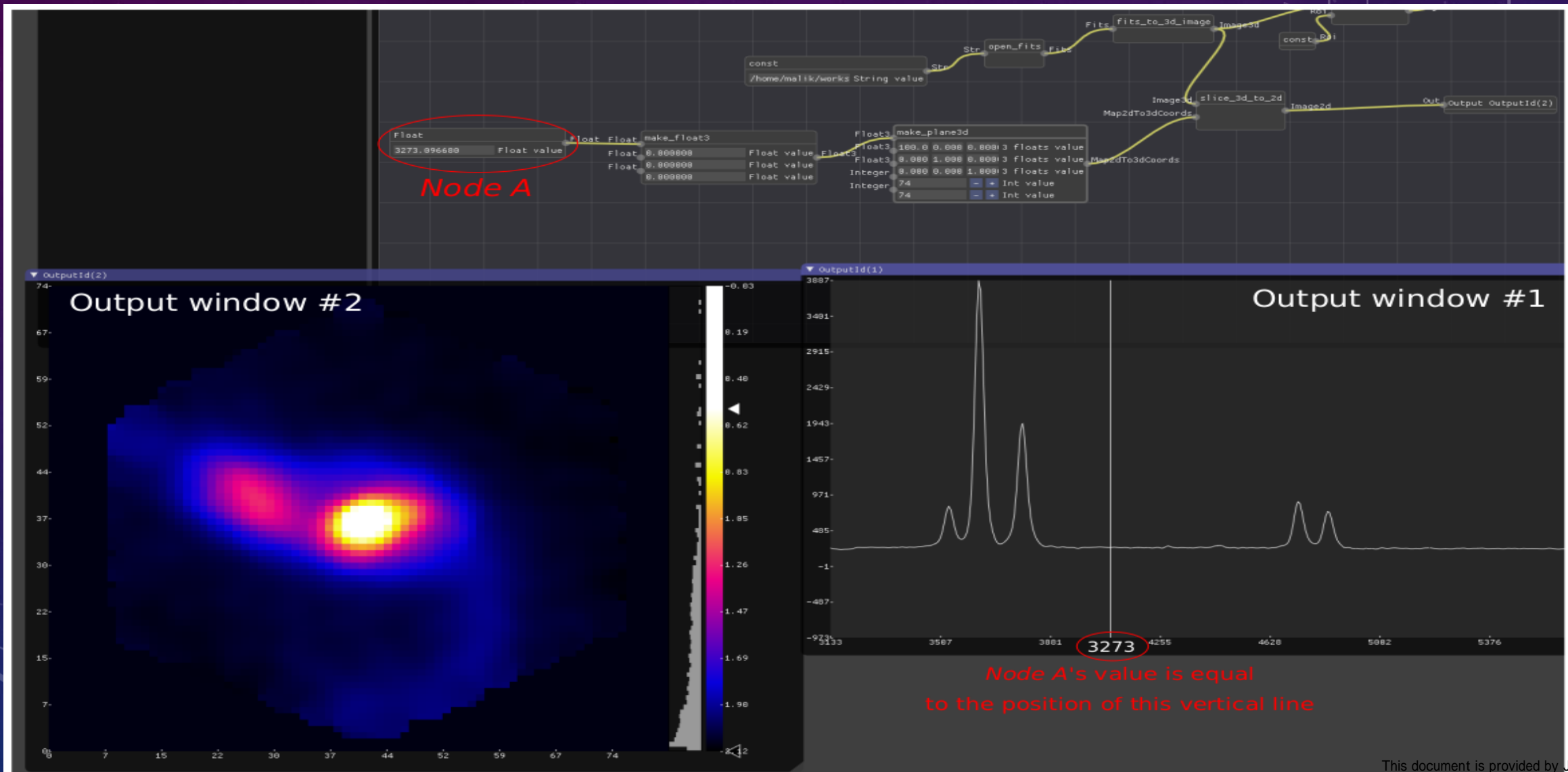
Embed node graph  
into FITS header



# CASE STUDY: EXTRACTION OF EQUIVALENT WIDTH

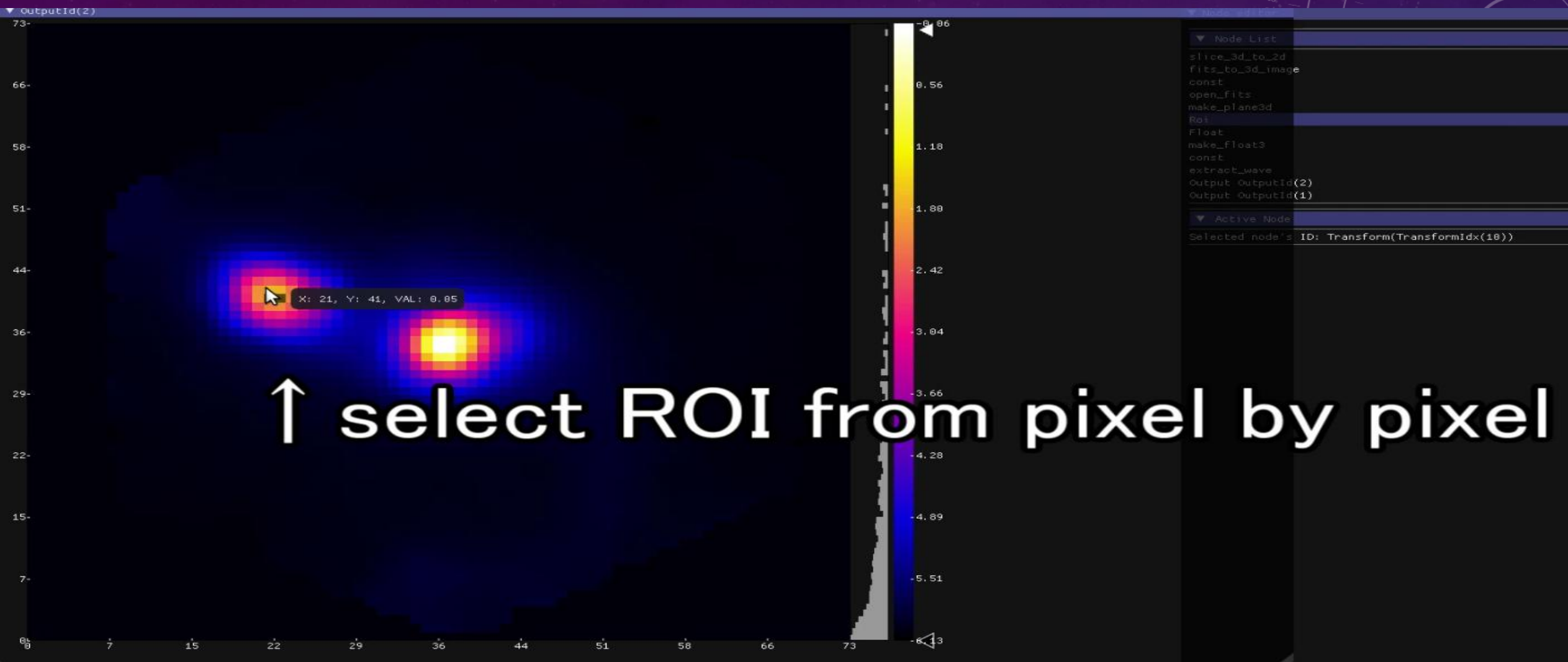


# DOUBLE-FEEDBACK BETWEEN NODE EDITOR'S VARIABLES AND VISUALIZATION OUTPUT



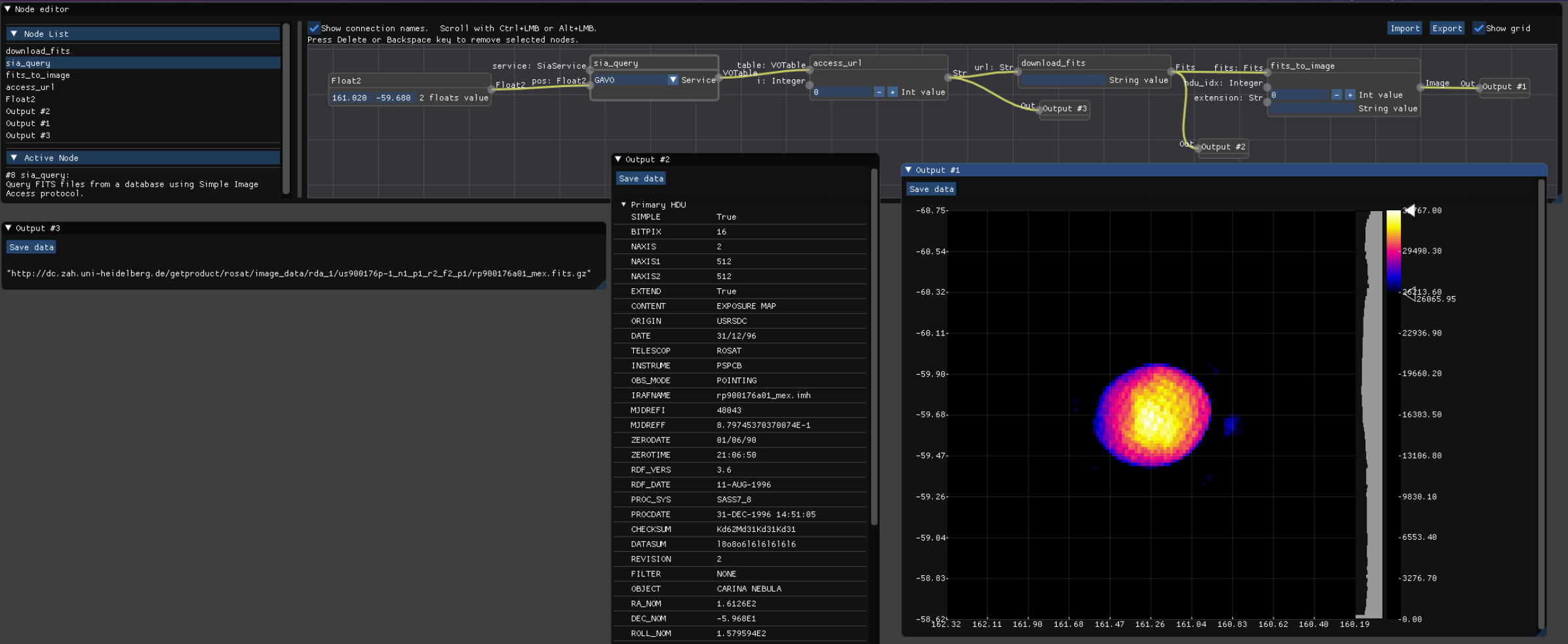
# SELECTING REGION OF INTEREST

<http://aflak.jp>





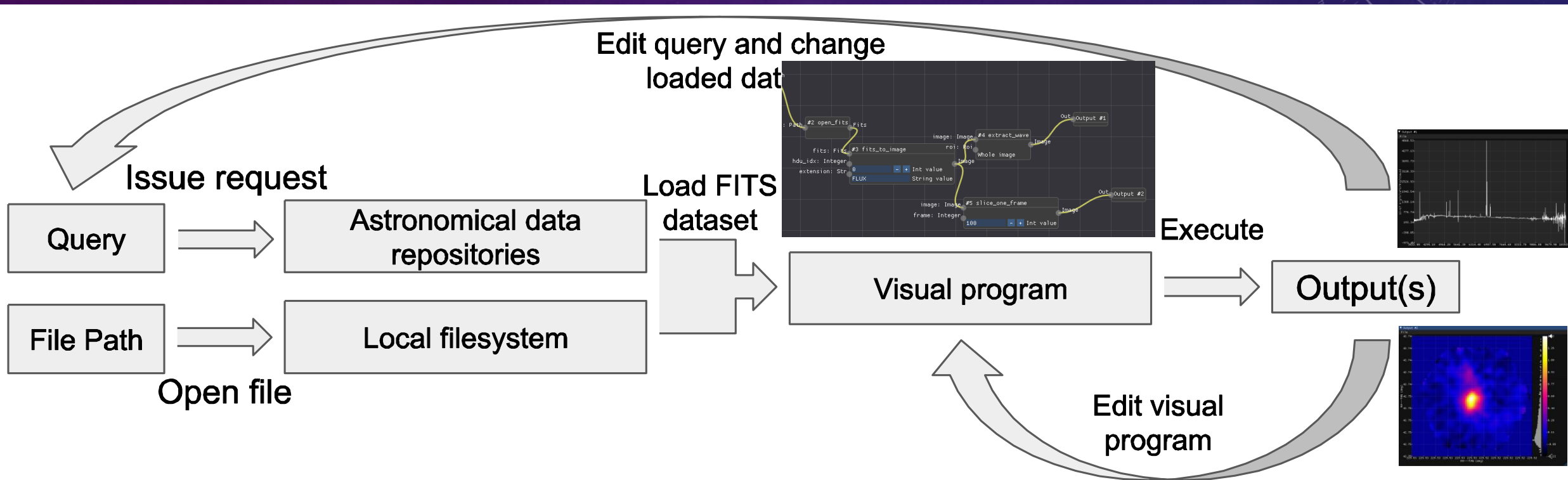
# QUERY DATA REPOSITORIES USING VO STANDARDS



# USE CASES

- Analysis that requires gradual and manual fiddling with many parameters  
aflak aims at putting the astronomer in the loop
- Denoising, preprocessing?

# WORKFLOW: ASTRONOMER IN THE LOOP






# PRIMITIVES

<http://aflak.jp>

- A set of algorithms applying transformations on datasets
- New nodes can be created
  - by combination of existing nodes (macro)
  - by direct implementation (currently Rust only, but C or Python solutions are explored)


# FITS AND VO LIBRARY IN RUST: FITRS / VO

<https://github.com/malikolivier/fitrs>

**crates.io**  
Rust Package Registry

Click or press 'S' to search...


[Browse All Crates](#) | [Docs](#) | [Log in with GitHub](#)

 **fitrs** 0.3.1

[Homepage](#) [Documentation](#) [Repository](#) [Dependent crates](#)

Cargo.toml

```
fitrs = "0.3.1"
```



Last Updated  
**a day ago**

Crate Size  
**33.42 kB**

**FITRS**

build passing

crates.io v0.3.1

docs 0.3.1

**Authors**

# FUTURE WORK

- <https://github.com/aflak-vis/aflak/issues>
- Batch processing
- Full macro support (Sit back! Writing the code as we speak, will be released very soon!)
- Full WCS support . Currently only partial and most probably buggy support is implemented.
- VO standards for communication with Aladin / Topcat / etc.
- Node primitives implementable in languages other than Rust (e.g. Python, C)
- Have more primitives included by default



# ONE (TWO) –COMMAND INSTALL!

<http://aflak.jp>

```
$ curl https://sh.rustup.rs -sSf | sh
```

```
$ cargo install --git https://github.com/aflak-vis/aflak aflak
```

- Currently supported OSes:
  - Debian 9.X
  - Ubuntu 18.04
  - macOS
  - Windows
- Run on normal laptop. RAM requirements depend on the open datasets. 4GB or more is advised. A wide screen (2K or more) is better.
- Bug report / Feature requests / Comment / Anything

<https://github.com/aflak-vis/aflak/issues/>