

# VISUALISATION *DYNAMIQUE* DES DONNÉES D'OBSERVATION



Jean-Olivier Irissou (irissou@normalesup.org)<sup>1</sup>, Service d'Observation de l'Observatoire Océanologique de Villefranche-sur-Mer<sup>2</sup>

<sup>1</sup> Sorbonne Universités, UPMC Univ Paris 06, CNRS, Laboratoire d'Océanographie de Villefranche (LOV), 06230 Villefranche-sur-Mer, France

<sup>2</sup> Sorbonne Universités, UPMC Univ Paris 06, CNRS, Observatoire Océanologique de Villefranche-sur-Mer (OOV), 06230 Villefranche-sur-Mer, France

## RÉSUMÉ

La première étape de l'utilisation des données d'observation est leur représentation **graphique** : elle permet un contrôle qualité minimum et l'identification visuelle de signaux qui devront ensuite être analysés statistiquement. L'Observatoire Océanologique de Villefranche-sur-Mer a mis en place une **architecture de traitement** et de **visualisation** des données en temps réel, en utilisant des outils OpenSource et relativement faciles d'utilisation, qui permet d'atteindre ces objectifs.

Je présente ici l'**architecture générale** de traitement de données mise en place à Villefranche, qui pourrait facilement être répliquée ailleurs, ainsi que l'**outil de visualisation** des données via des pages web **dynamiques**.

## PRINCIPE

Les données récoltées sont de natures et de **formats divers**. Les fichiers bruts sont envoyés sur un **serveur central** par chaque site d'observation. Toutes les nuits, des **scripts ad-hoc lisent** l'ensemble des fichiers bruts, **traduisent** les noms de variables et les éventuels codes qualités selon un **standard homogène**, font un premier **contrôle qualité** automatique des variables, **dérivent** des variables annexes et **re-formatent** les données. Celles-ci sont écrites dans des formats d'échange et des formats d'archivage.

Après ce traitement, les données sont lues et mises en mémoire par des **applications web dites réactives**, c'est à dire qui recalculent leurs sorties (graphiques) en fonction des choix des utilisateurs. Les opérateurs peuvent alors **explorer** puis **corriger** ou **qualifier** les données brutes. Le résultat des corrections sera visible dès le lendemain.

## OUTILS

Le serveur central est un serveur **linux Ubuntu 16.04**, accessible par **SSH** et partage **Samba** (lecteur réseau Windows).

La programmation du lancement des scripts est faite par **cron**.

Les scripts de traitement sont écrits en **R**, à l'exception du traitement CTD qui fait appel à des batchs pour **SBE Data Processing**.

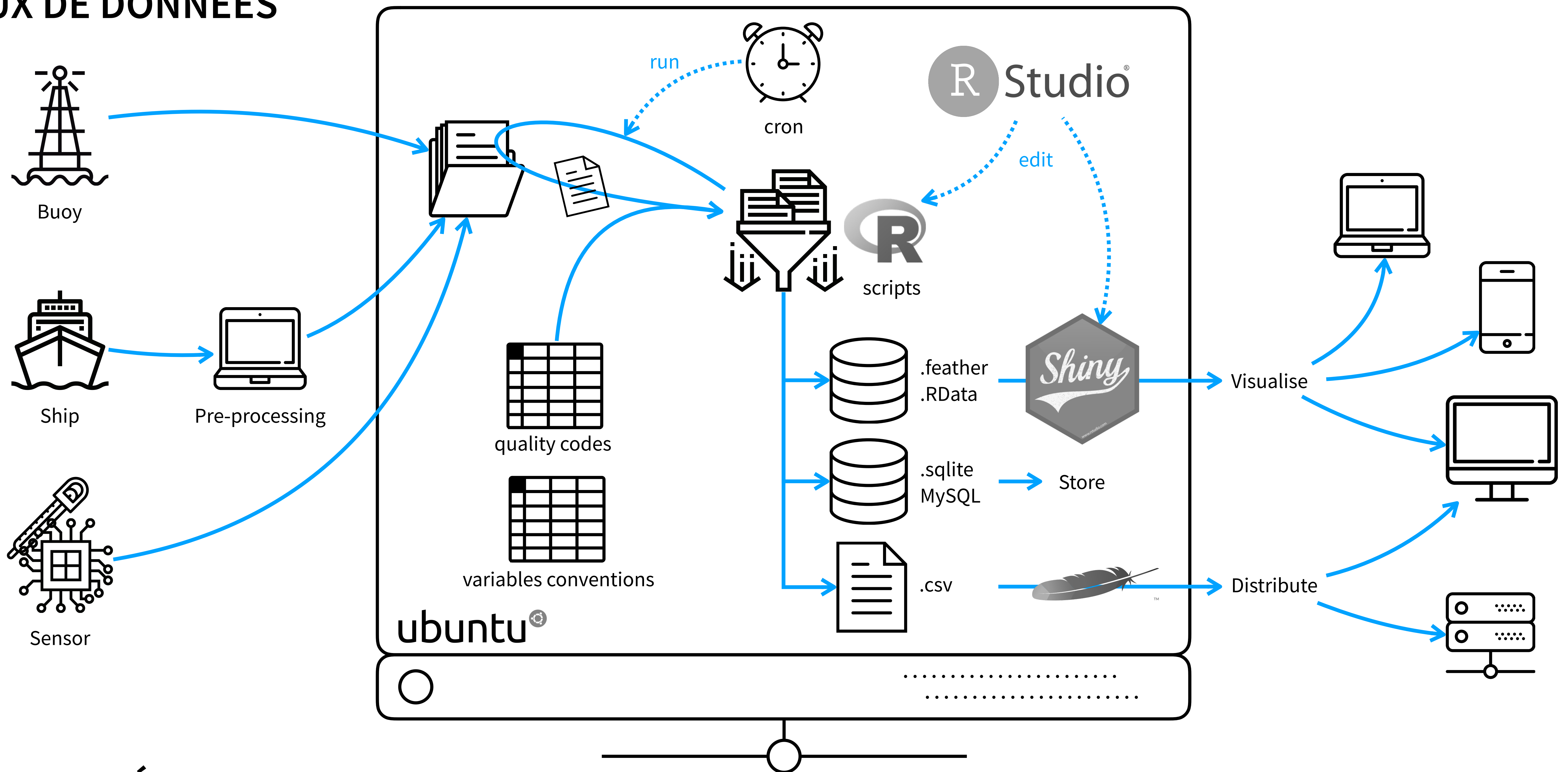
Les données sont écrites en **.feather** (format d'échange entre python et R) ou **.RData** pour réutilisation + intégrées à une base **MySQL**, écrites en **.sqlite** et **.csv** pour archivage.

Les applications web de visualisation des données sont écrites en **R** avec le package **shiny**.

Les scripts et applications peuvent être édités à distance par chaque utilisateur via **RStudio Server**.

Certains fichiers de données sont distribués sur le web via un serveur **Apache**.

## FLUX DE DONNÉES



## RÉACTIVITÉ

Start server

global.R

- load functions
- load data in memory

Allez voir des exemples sur <http://www.obs-vlfr.fr/data/view/>

Connect to app

Demo EvolEco

Select a date range: 2017-01-01 to 2017-09-30

Choose a colour for the plot: black, red, blue

Change dates

Demo EvolEco

Select a date range: 2017-01-01 to 2017-09-30

Choose a colour for the plot: black, red, blue

Change colour

Demo EvolEco

Select a date range: 2017-01-01 to 2017-09-30

Choose a colour for the plot: black, red, blue

ui.R

- prepare user interface 1

ui.R

- prepare user interface

ui.R

- prepare user interface

server.R

- select data 3
- prepare plot 4
- make plot 2, 5

server.R

- select data 2
- prepare plot
- make plot 1, 3

server.R

- select data
- prepare plot 2
- make plot 1, 3