La Cantine

Introduction à la Programmation en Flux

« Lazy » flow-based programming :
Utilité, concepts de base et application avec PyF.

Présenté par :
Jonathan Schemoul, consultant JMSI

Qu'est-ce que la programmation en flux ?

- Dans les années 70 : John Paul Morrison au Canada
  - Applications Bancaires
- On traite les données unitairement
  - Elles passent d'un bloc à un autre, formant un flux
  - On transforme les données dans le flux, comme dans une chaine de production.

Qu'est-ce que la programmation en flux ?

Dans la théorie :

- L'ensemble de blocs (composants) constitue un réseau
- Chaque composant a des ports d'entrée et de sortie
- Le réseau est géré par un « scheduler » (ordonnanceur)

  Il gère les communications entre les processus, et vérifie l'état du réseau

- Les données échangées sont des paquets standards, appelés « Information Packets ».

En Python : une chaine de générateurs

Pour utiliser les générateurs en programmation en flux...

On définit des boites, étant chacune un générateur

- On passe l'itérateur à un autre générateur, qui renverra lui aussi un flux
- On dit que l'on « adapte » le flux au fur et à mesure

```
>>> def mon_adaptateur(flux) :
...     for v in flux:
...         print « J'ai reçu », v
...         yield dict(valeur=v,
...                    double=v*2)

>>> it1 = xrange(5) # 0 à 5
>>> it2 = mon_adaptateur(it1)

>>> def mon_consommateur(flux_adapte) :
...     for nv in flux_adapte:
...         print nv['double']
...         yield True # c'est ok.

>>> it3 = mon_consommateur(it2)

>>> for value in it3:
...     if value is True:
...         print 'ok so far'
...     else:
...         print 'not ok'

J'ai reçu 0
0
ok so far
J'ai reçu 1
2
ok so far
[...]
```

Avantages de la prog. en flux avec des générateurs

- Pas de montée en RAM, traitement de gros volumes
- Méthode de programmation facile à apprendre (on prend un item en entrée, on en sort un... ou pas :-)
- Tous les objets python peuvent être utilisés (pas uniquement des packets)
- Pas besoin de threads
- Pas besoin de stackless, utilisation de la librairie standard

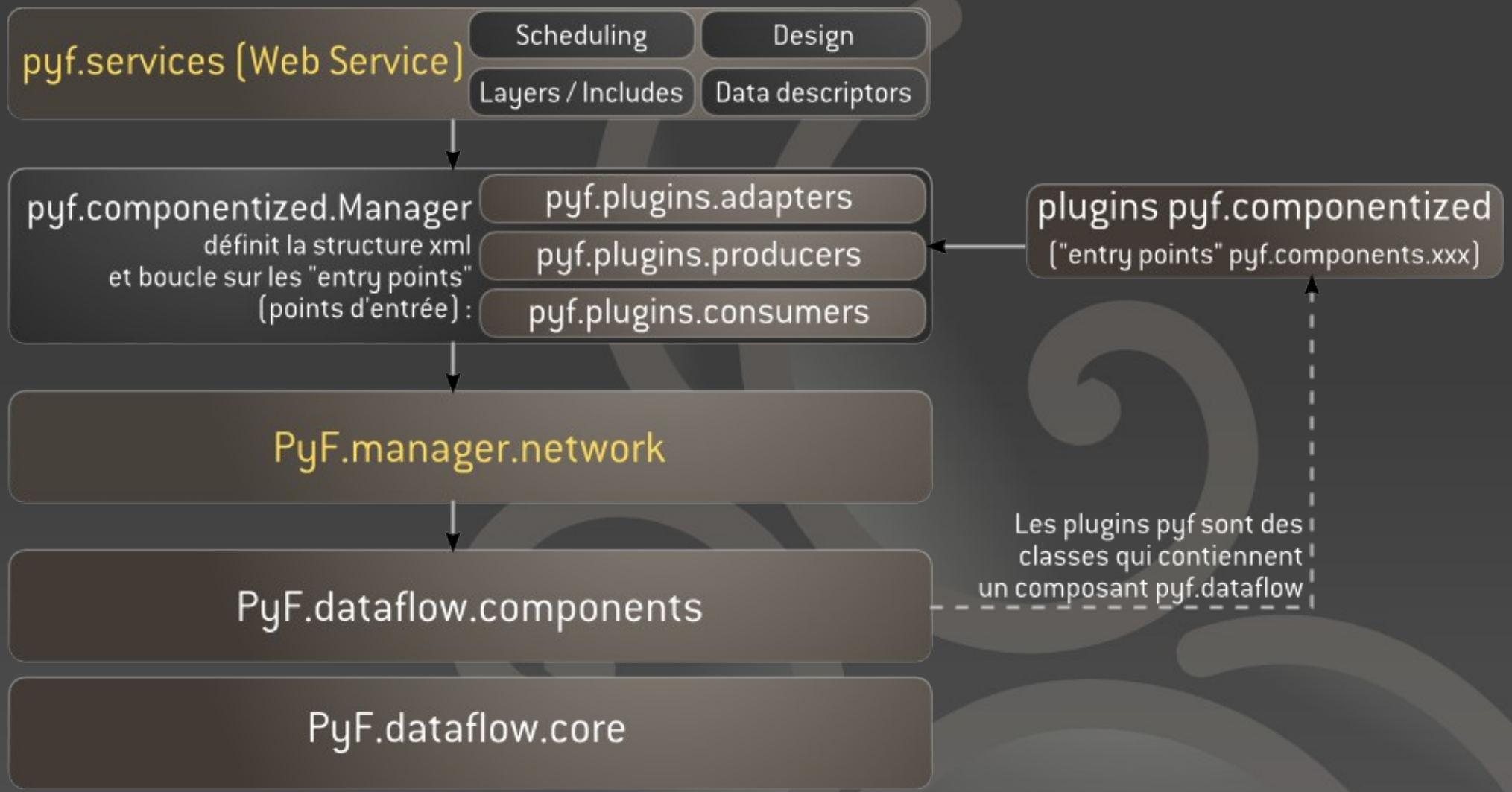
En Python : PyF

- PyF est un framework basé sur un fork de Zflow pour les couches basses (pyf.dataflow.core)

  Fork « ami »

- PyF contient beaucoup plus que la base, et est « batteries included ».

Architecture de PyF



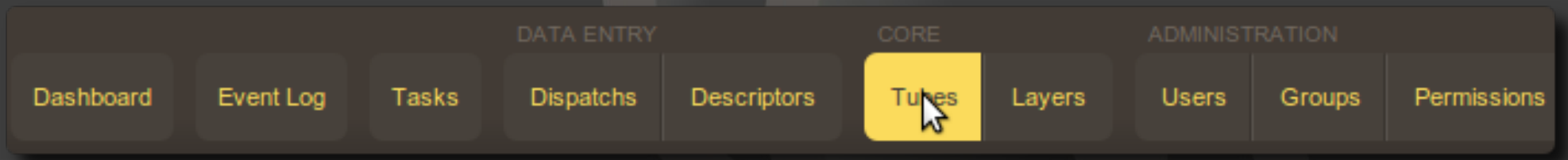
En Python : PyF

- Système de Plugins
  - De nombreux existants
    - pour l'extraction (site web, rss, sql alchemy...),
    - pour l'adaptation,
    - pour l'écriture (csv, xml, fixed-length, xlsx, pdf avec xhtml, odt ou rml, etc.)
- Du code python peut être mis dans le code source d'un tube
- Beaucoup d'outils annexes pour la gestion des flux (designer graphiquer, scheduler, …)

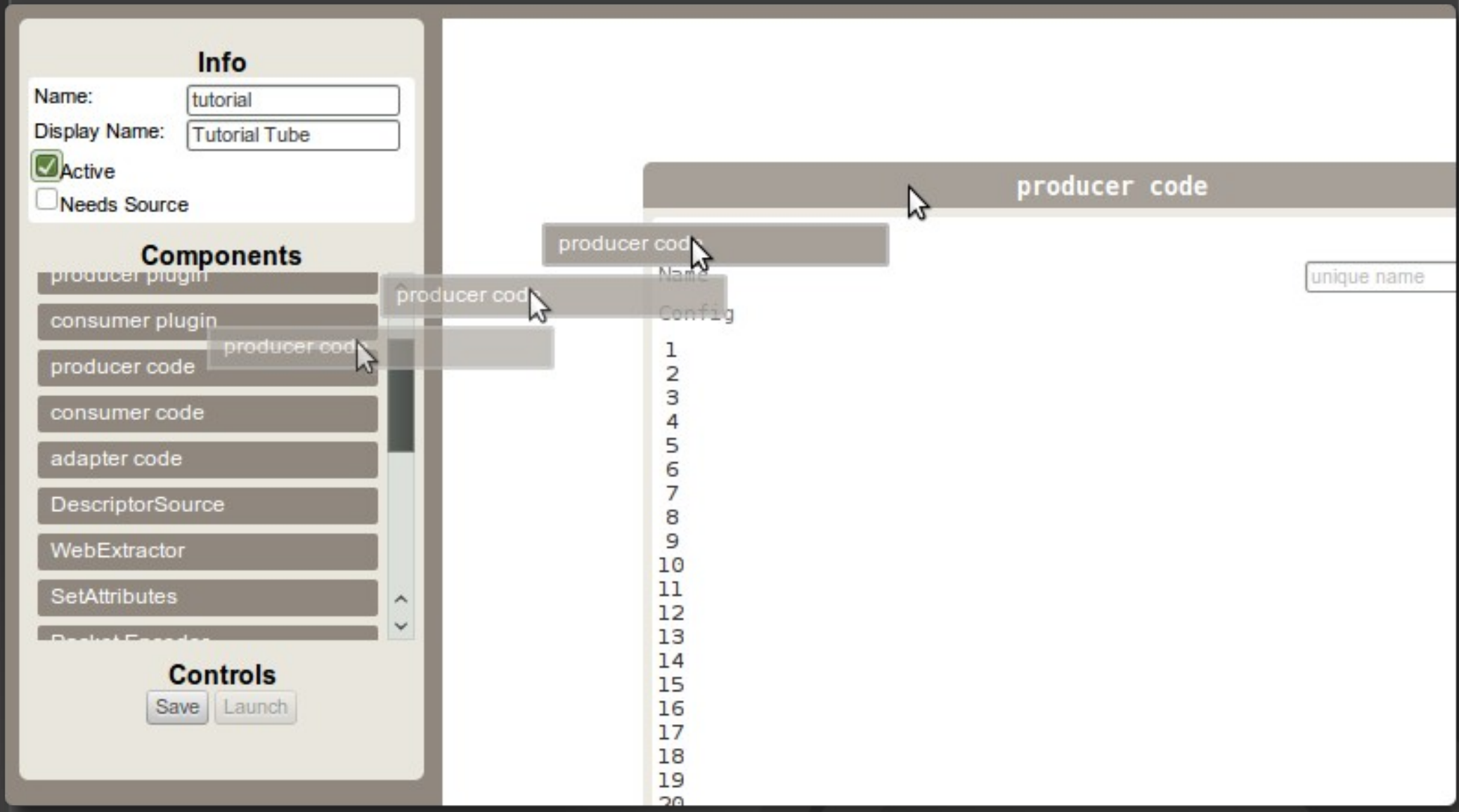
Les avantages

- Votre business c'est vos données
- PyF les traite, rapidement et de façon « scalable »
  - Lecture,
  - Transformation,
  - Adaptation,
  - Reporting...

Le service web



Créer un tube



Du code...

producer code

Joiner Info

Name source1

Config

```
class User(object):
    def __init__(self, nam
        self.name = name
        self.email = email
        self.level = level

def get_source():
    for index in range(0,
        yield User(
                    "John%0
                    "john%0
                    ['high'
```

adapter code

Joiner Info

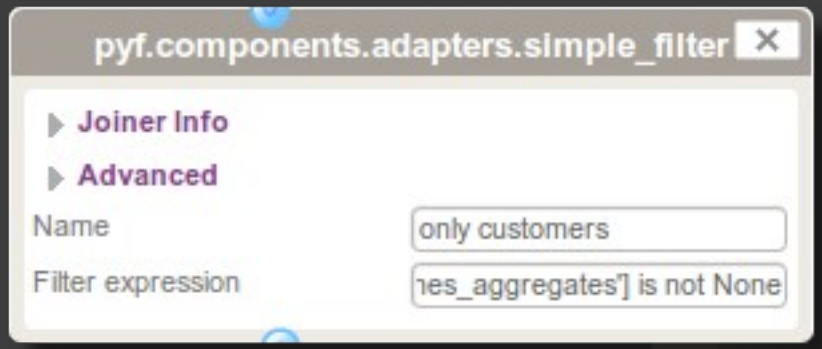
Advanced

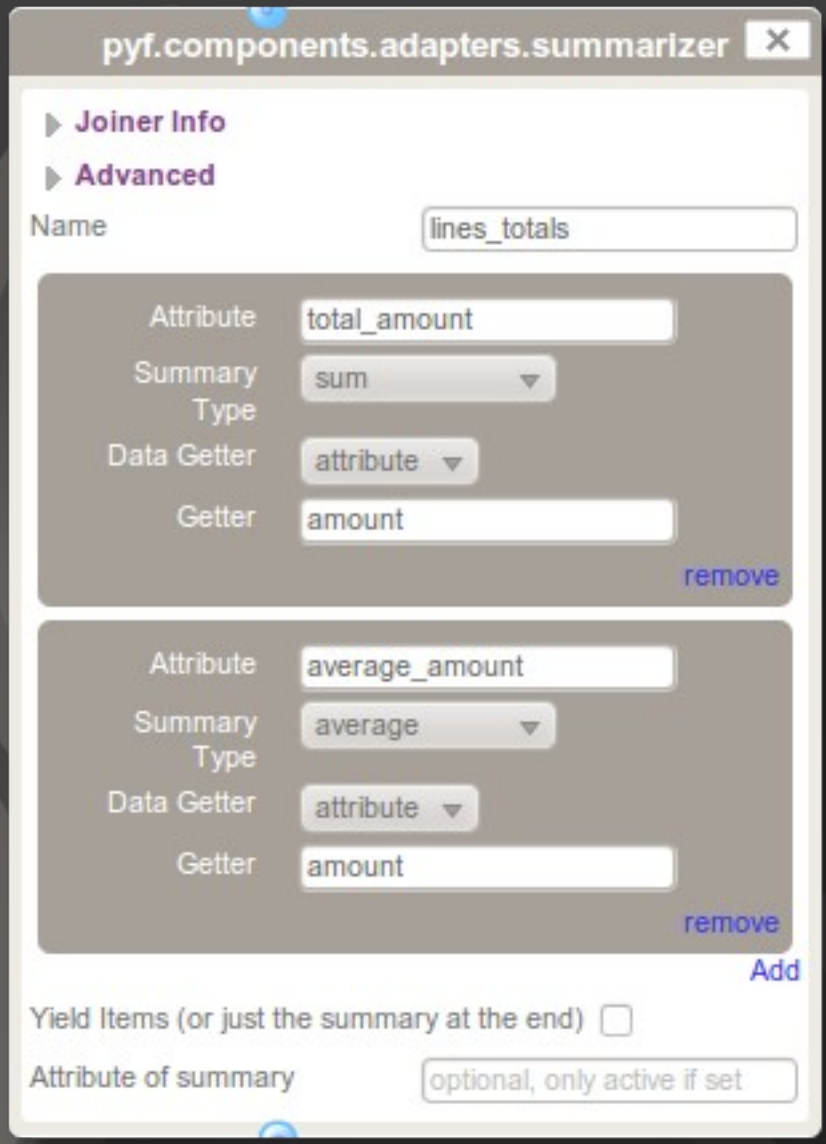
Name lines_aggregates

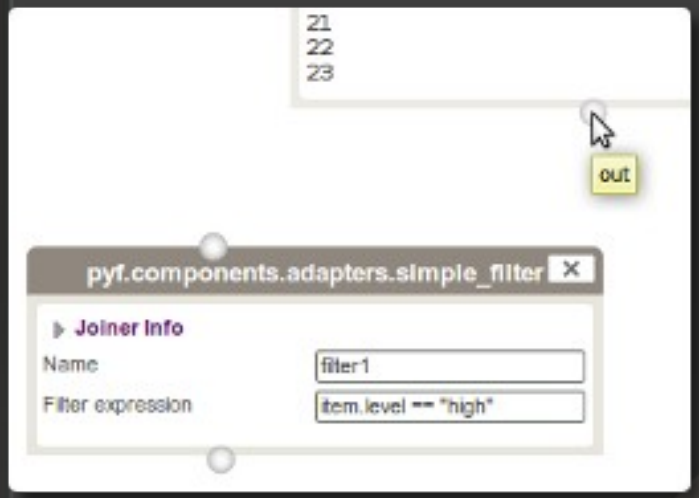
Config Add

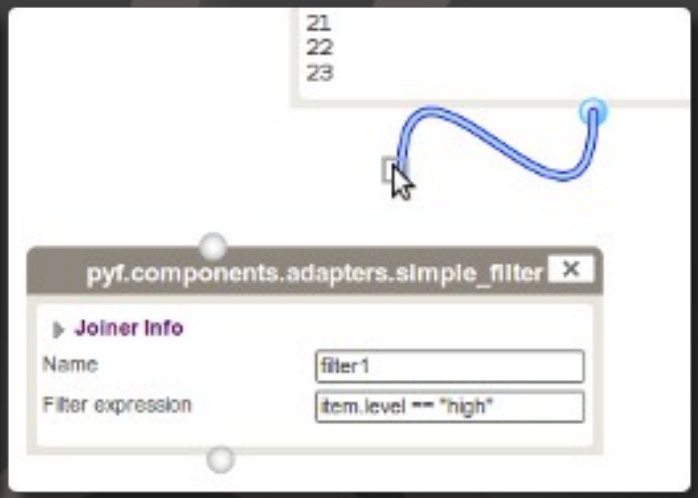
```
from itertools import groupby
from operator import attrgetter
from pyf.transport import Packet

def aggregate_lines(lines):
    for account_code, acc_lines in groupby(lines, key=attrgetter('ac
        yield Packet(dict(account_code=account_code,
                          lines=list(acc_lines)))
```

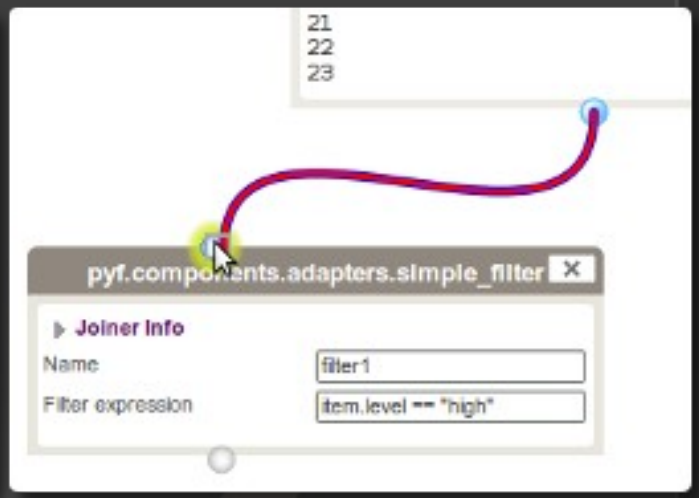
Ou des plugins...

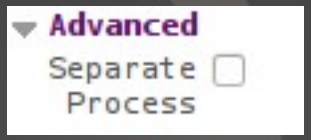


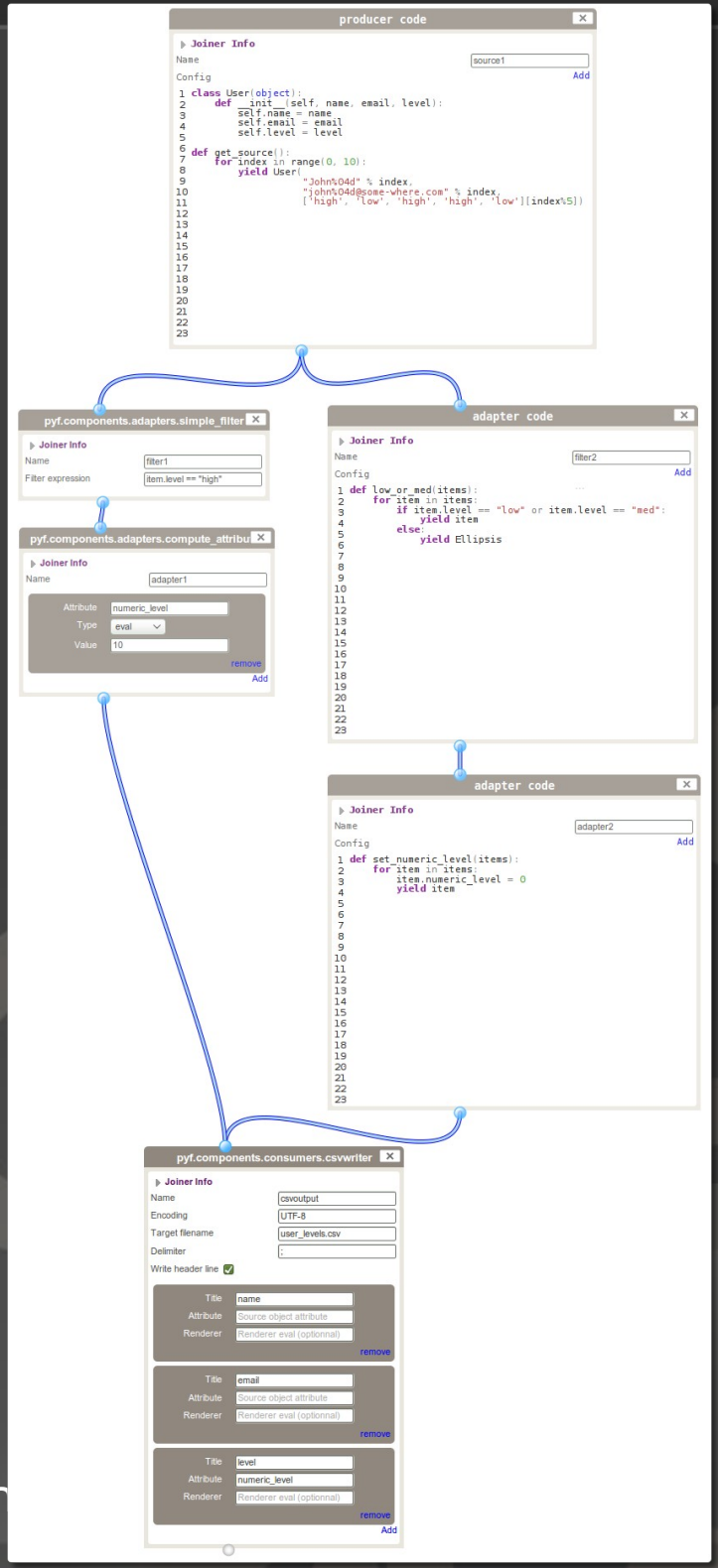



21
22
23
out
pyf.components.adapters.simple_filter
Joiner Info
Name
filter1
Filter expression
item.level == "high"


21
22
23
pyf.components.adapters.simple_filter
Joiner Info
Name
filter1
Filter expression
item.level == "high"

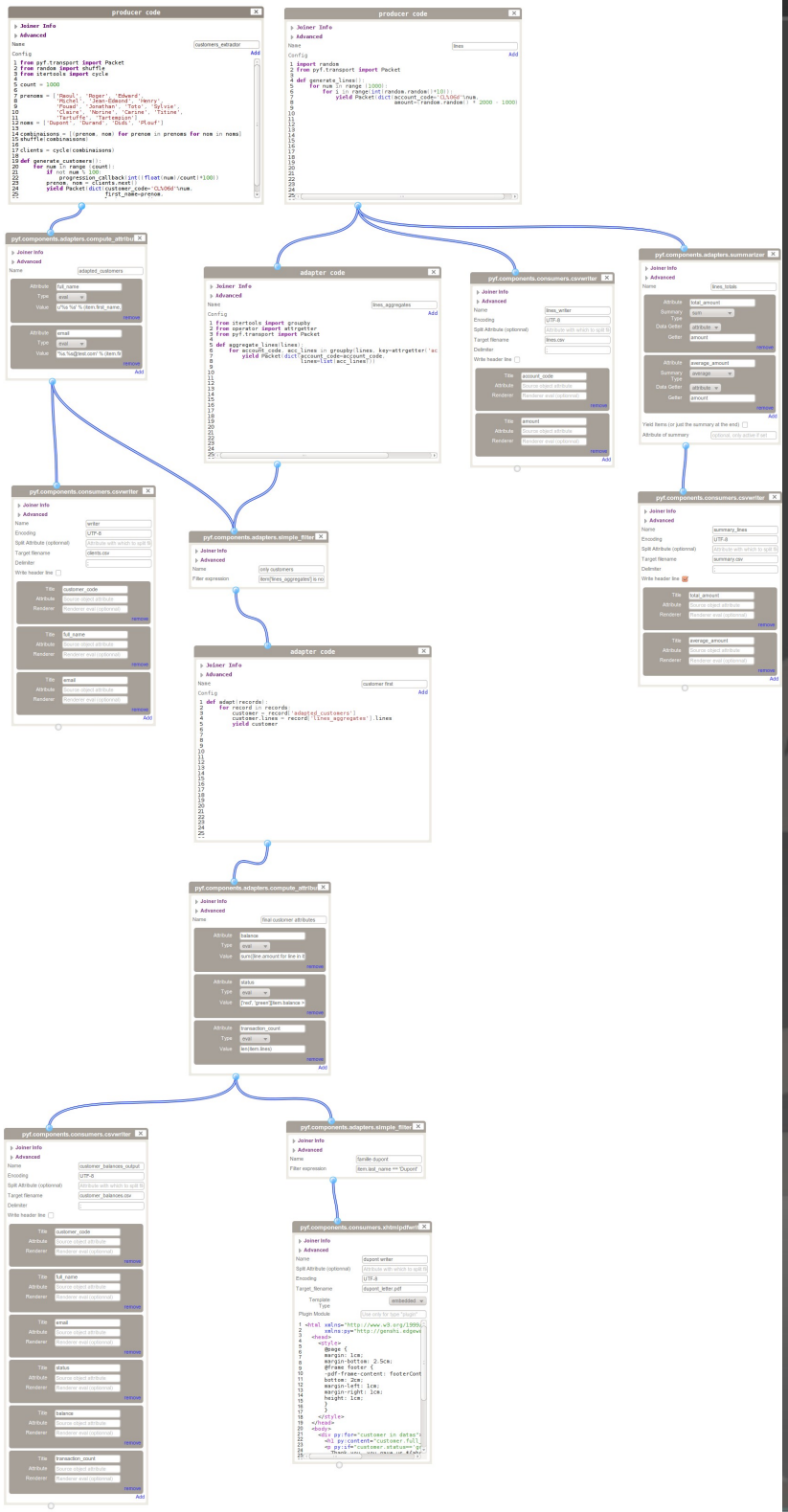

21
22
23
pyf.components.adapters.simple_filter
Joiner Info
Name
filter1
Filter expression
item.level == "high"


Advanced
Separate Process



jmsi Software Consultin

PYthon Flow



jmsi Software Con

PYthon Flow

A bientôt !

- Pour plus d'informations :

  http://www.pyfproject.org/

  http://groups.google.com/group/pyf-users