

MAWIE

PAR ILIAS, THOMAS ET ERIC

TBL DES MATIÈRES

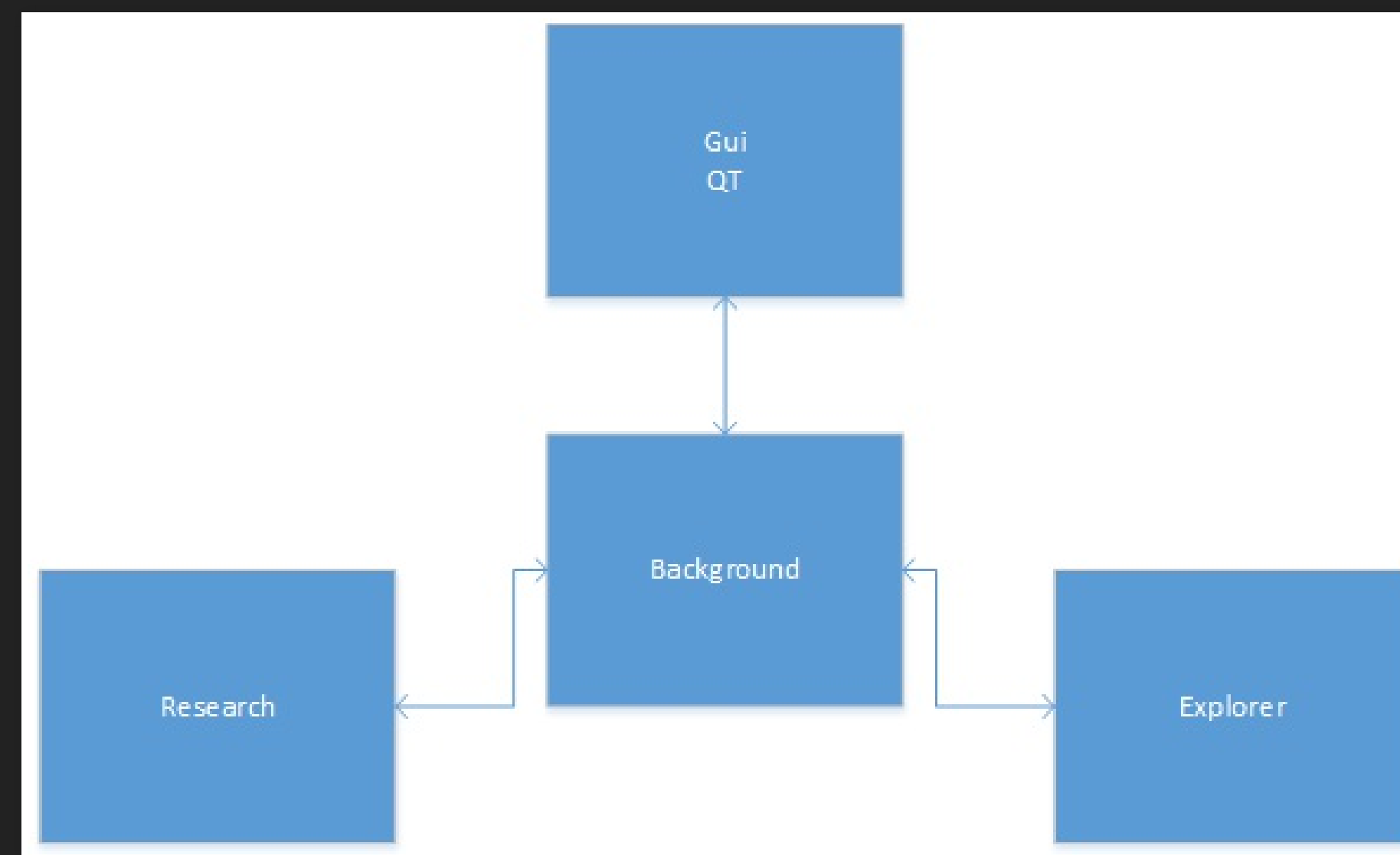
- Technologies
- BDD & Modèles
- Composants
- Explorer
- Search
- Gui
- Event handling
- Bugs restants
- Améliorations
- Conclusion
- Q/A

TECHNOLOGIES

- Python3
- SQLAlchemy (ActiveAlchemy)
- sqlite3
- PyQt5
- Tkinter (déprecié)

On a développé l'application pour qu'elle soit orienté événement (comme une application Android).

COMPOSANTS



SEARCH

Recherche sur les 2 modèles, ou autre.

- Namepsace : mawie.research
- Recherche Simple
- Recherche Avancé sur les modèles par defaults
- Recherche Avancé

##^ Recherche simple

```
from mawie.research.research import Research
from mawie.models.movie import Movie
searchable = Research()
res = searchable.query("Some awesome movie title")
for elem in res:
    assert isinstance(elem, Movie)
    print(elem.title)
```

##^ Recherche Avancé

```
... imports
searchable = Research()
res = searchable.query("Some awesome movie title",["desc","acti
for elem in res:
    assert isinstance(elem, Movie)
    print(elem.title)
```

##^ Recherche Avancé (2)

```
... imports
searchable = Research()
```

```
res = searchable.query({Movie:{"release":{"gte":"2010-01-01",'
for elem in res:
    assert isinstance(elem, Movie)
    print(elem.title)
```

EXPLORER

- Qu'est-ce que c'est
- Namespace : mawie.explorer
- Utilise des Apis (duckduckgo, et imdb)

##^ Utilisation d'un comparateur

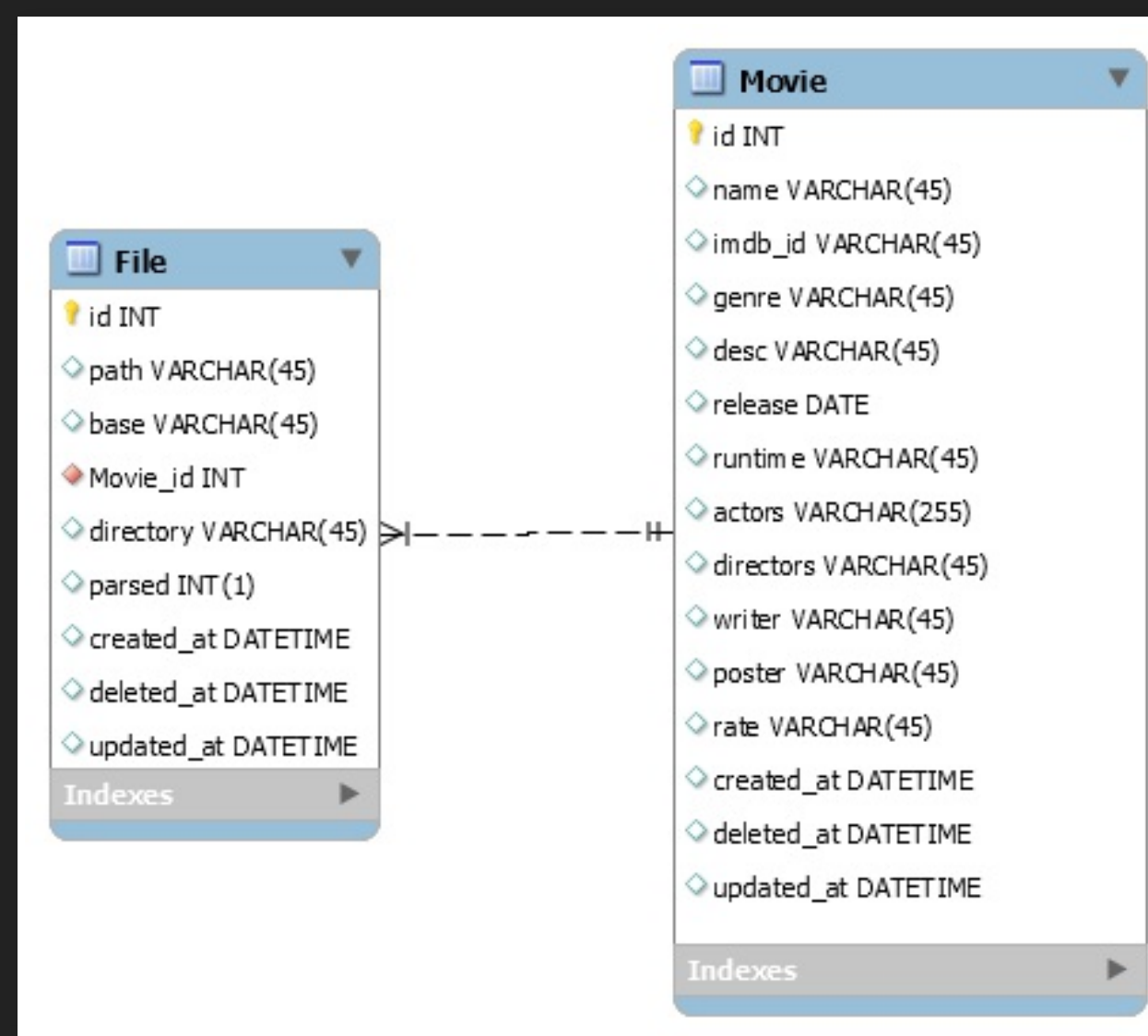
Dans un premier, on a utilisé Levenstein pour comparer le nom du film que nous renvoyait Guessit, et le nom du fichier originel (si taux de ressemblance plus haut que 80% c'est OK)

Cela était due au nom des fichier improbable dans le dossier.

##^ Utilisation de duckduckgo

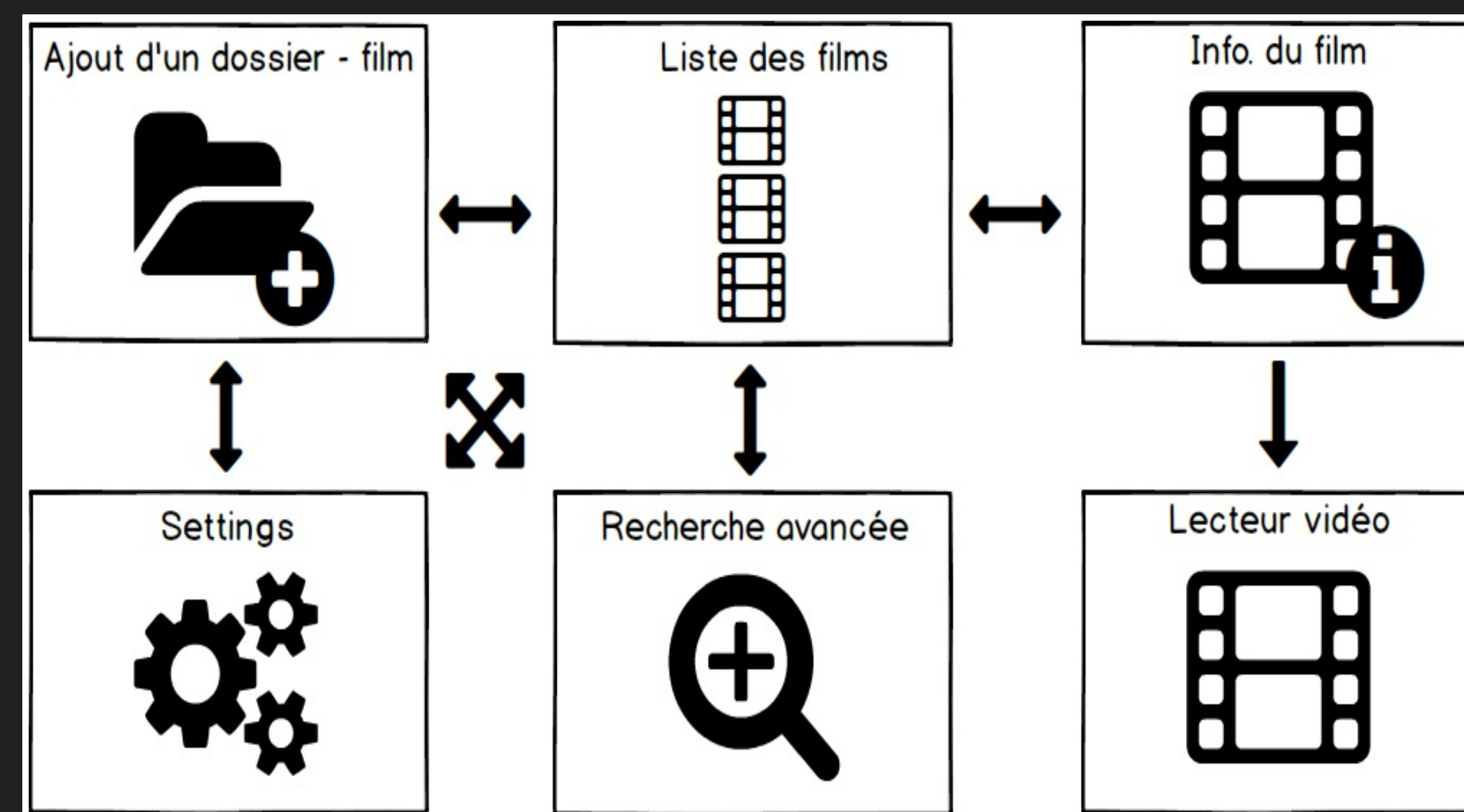
Dans un deuxième temps on a développé une solution plus simple. On recherche le nom du film donnée par Guessit sur duckduckgo. Cela permet de le traduire, et d'avoir beaucoup plus souvent des résultats de recherche cohérent (dépendant du film). Après avoir récupéré le contenu imdb, on fait un test de ressemblance entre le nom Guessit, et le nom retiré IMDB pour vérifier que l'on ait bien trouvé le bon film, puis on l'insère dans la base de données.

BDD & MODÈLES

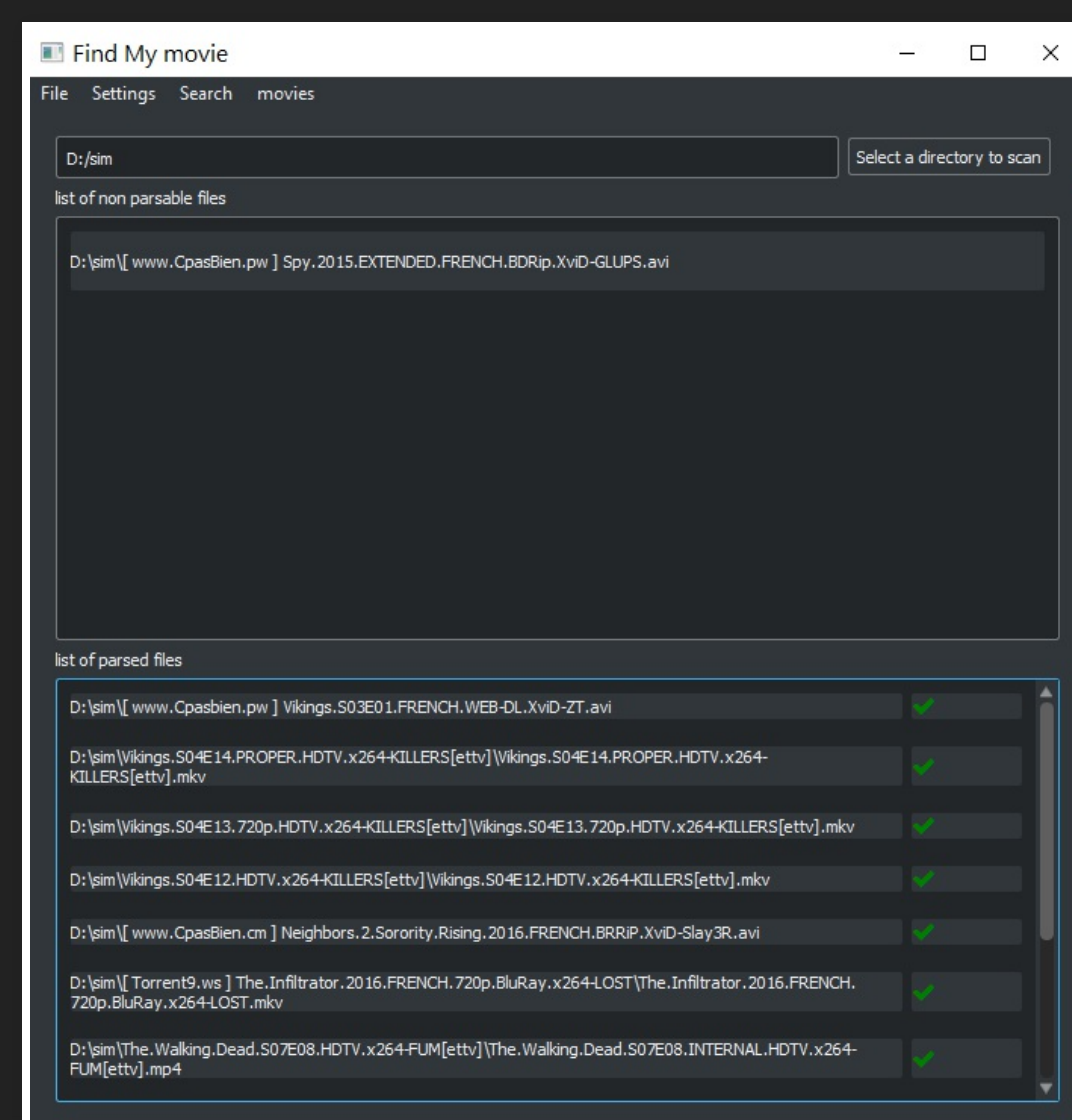


GUI

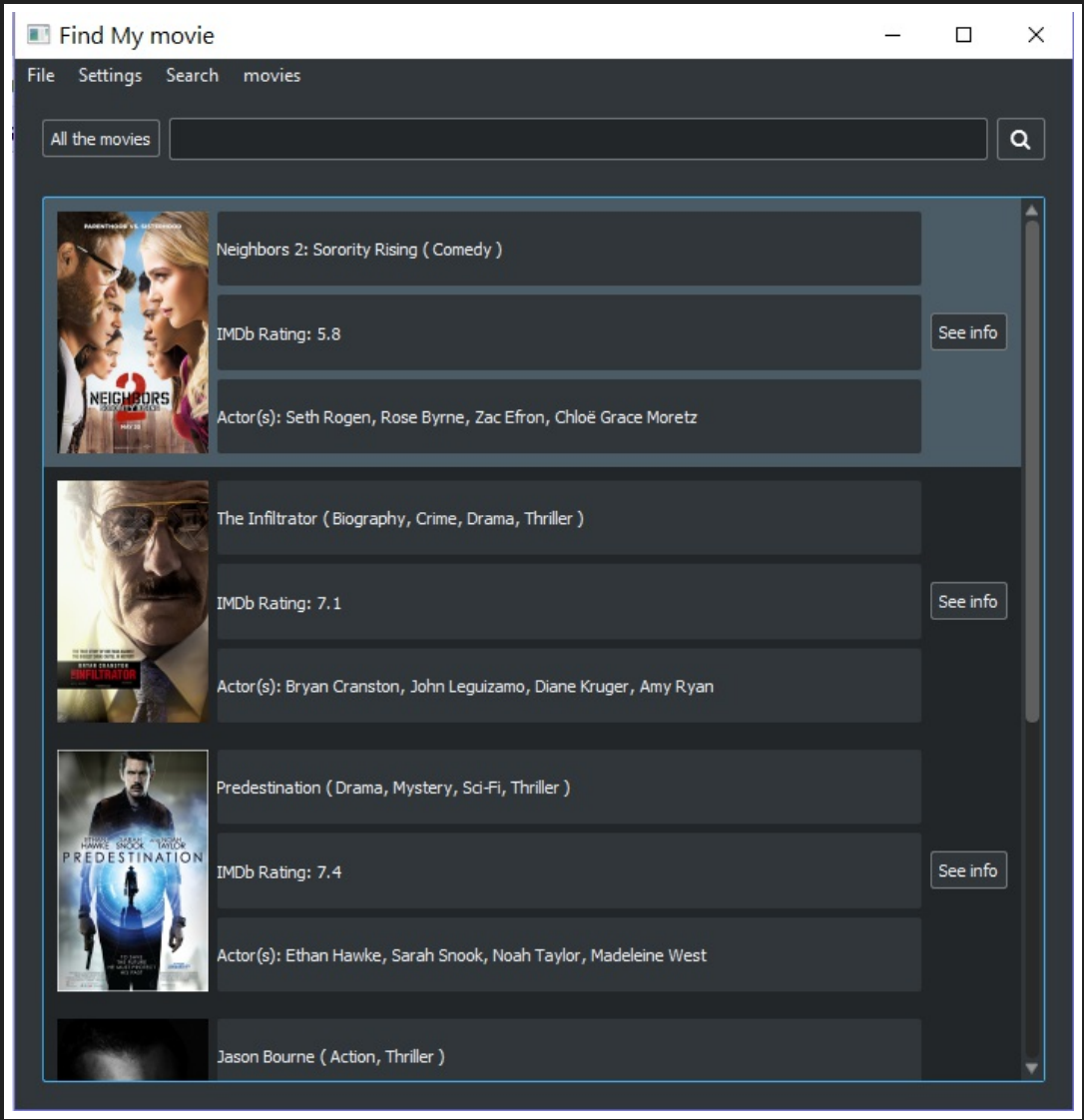
LES DIFFÉRENTES FENÊTRES



##^ ajouter un dossier



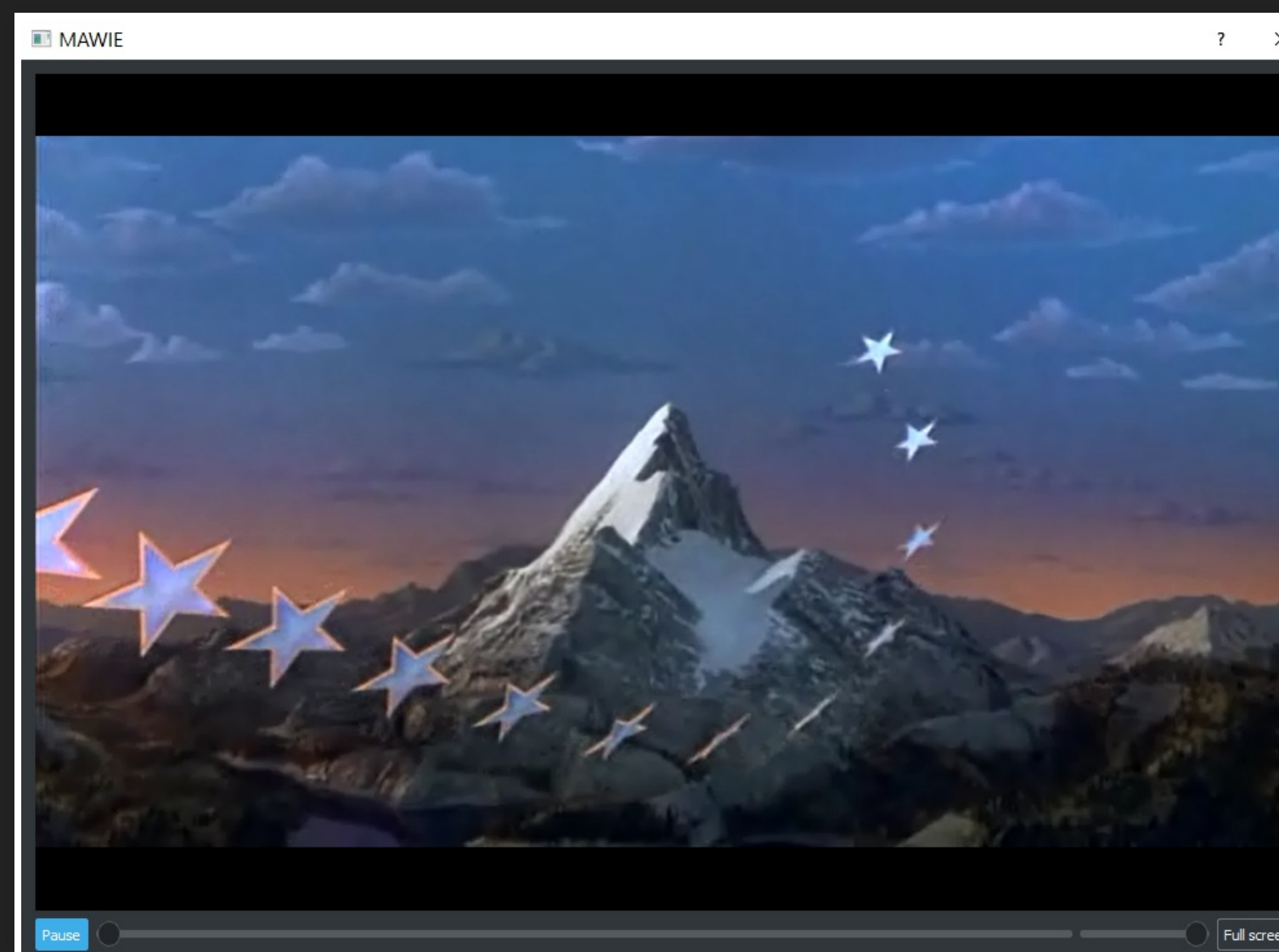
##^ Affichage le contenu d'une recherche



##^ Affichage des informations d'un film



##^ Lire le film



##^ Recherche avancée



GUI : AMÉLIORATIONS

- Gui
- Explorer
- Research
- Système d'événement

##^ Gui

- Améliorer l'ergonomie
- Rendre plus responsive
- Faire une fenêtre adaptée pour les séries
- Ajouter la possibilité de donner un nom de film pour les fichiers qui n'ont pas été parsés
- Indiquer le film comme «viewed» lorsqu'on a - cliquer sur le bouton «play film»

BUGS RESTANTS

- Gui
- Explorer
- Research
- Système d'événement

##^ Gui

- La fenêtre dépasse si l'écran est trop petit
- Pas de gestion des formats non pris en charge par le lecteur media personnalisé
- Sous Linux, le fichier ne se montre pas dans l'explorer

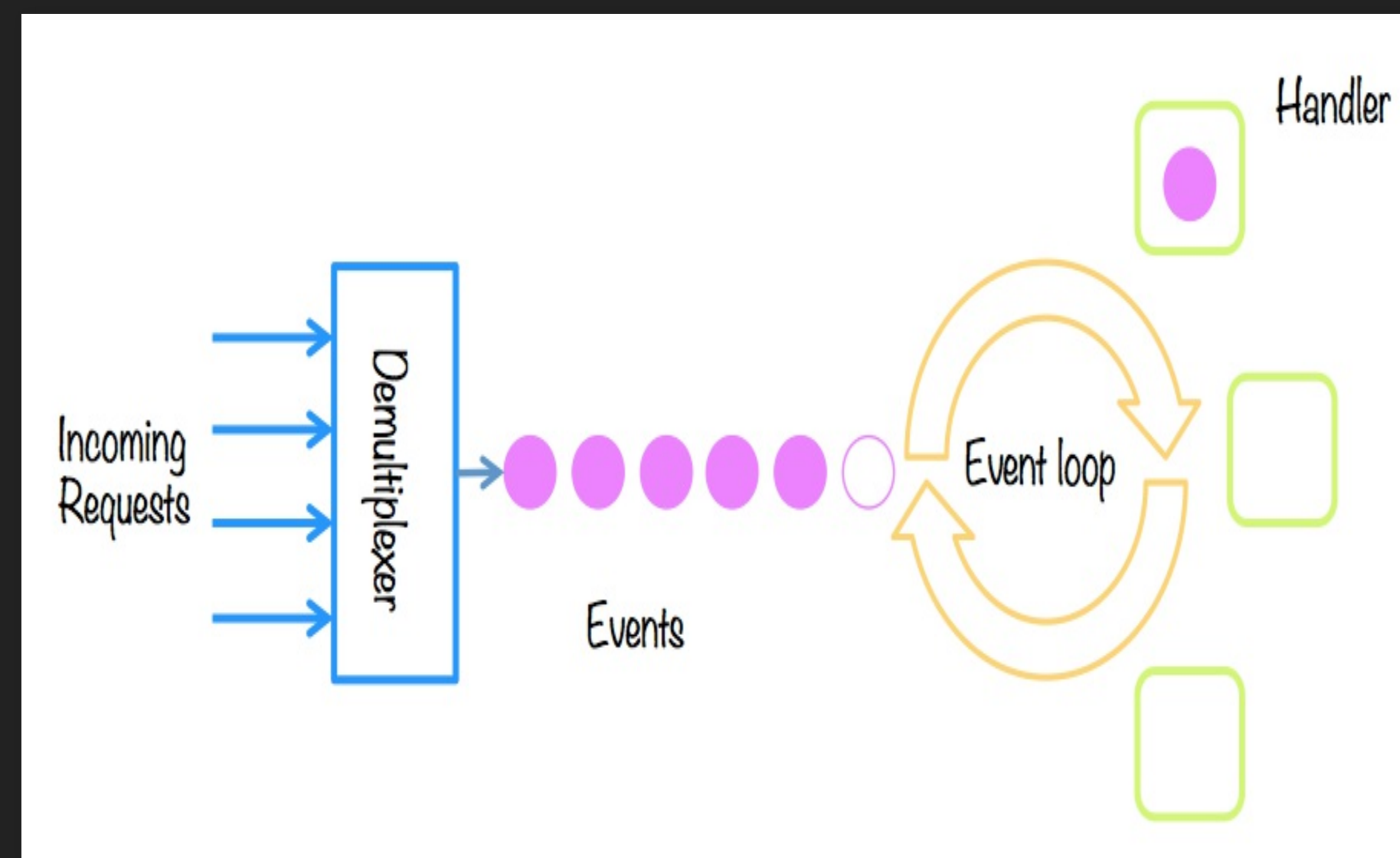
GESTION D'ÉVÉNEMENT

##^ Entre composants QT

On utilise les signaux Qt, et ensuite on communique les données à l'arrière plan avec le système d'événement.

##^ Entre service d'arrière plan

On utilise une solution faite maison qui reprend l'idée d'un MessageBroker (AMQT, ZeroMq, etc...)



Note: On fait tourner un event loop dans le GUI (pour passer les données à l'arrière plan).

Note: On fait tourner un Event loop dans un thread à part pour processor les événements toute les .25s (pour le pas surchargé le thread principale QT!)

Note: Tout ça a cause du GIL