



# OpenFLUID

Software Environment  
for Spatial Modelling in Landscapes

## Une plateforme logicielle pour la modélisation spatio-temporelle des paysages

Equipe OpenFLUID

*LISAH - Laboratoire d'étude des Interactions Sol-Agrosystème-Hydrosystème*

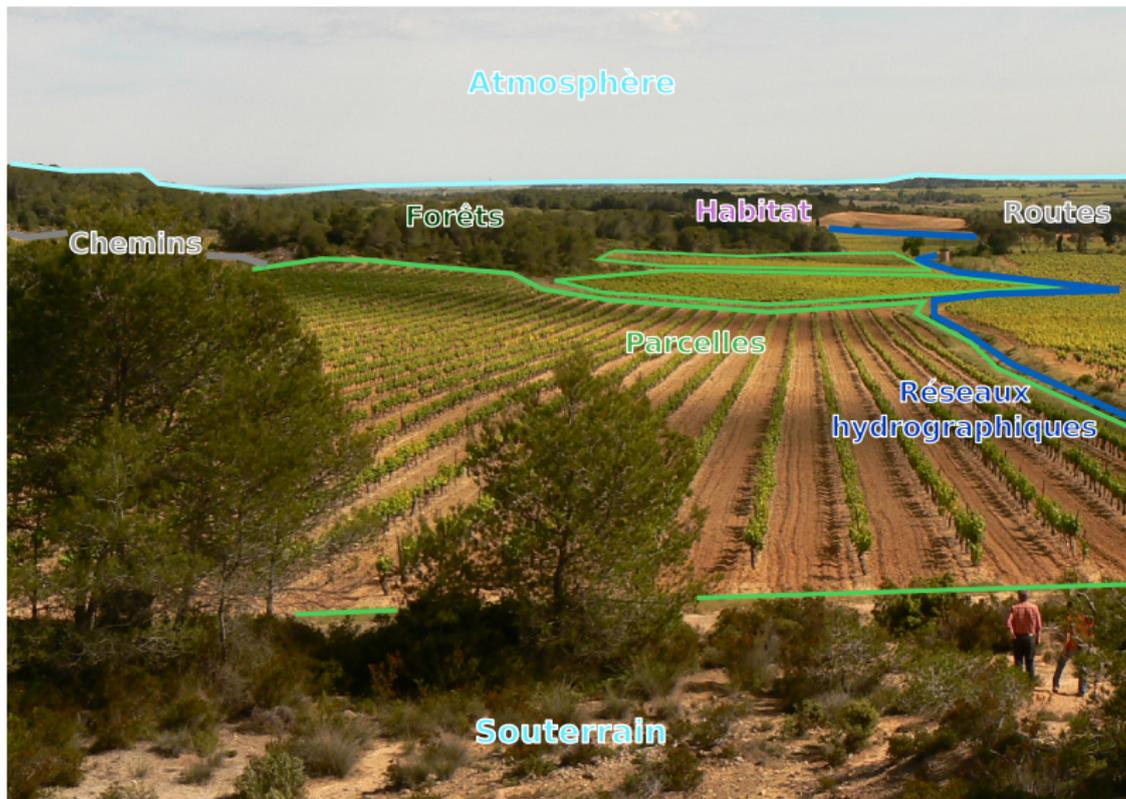


# Panorama

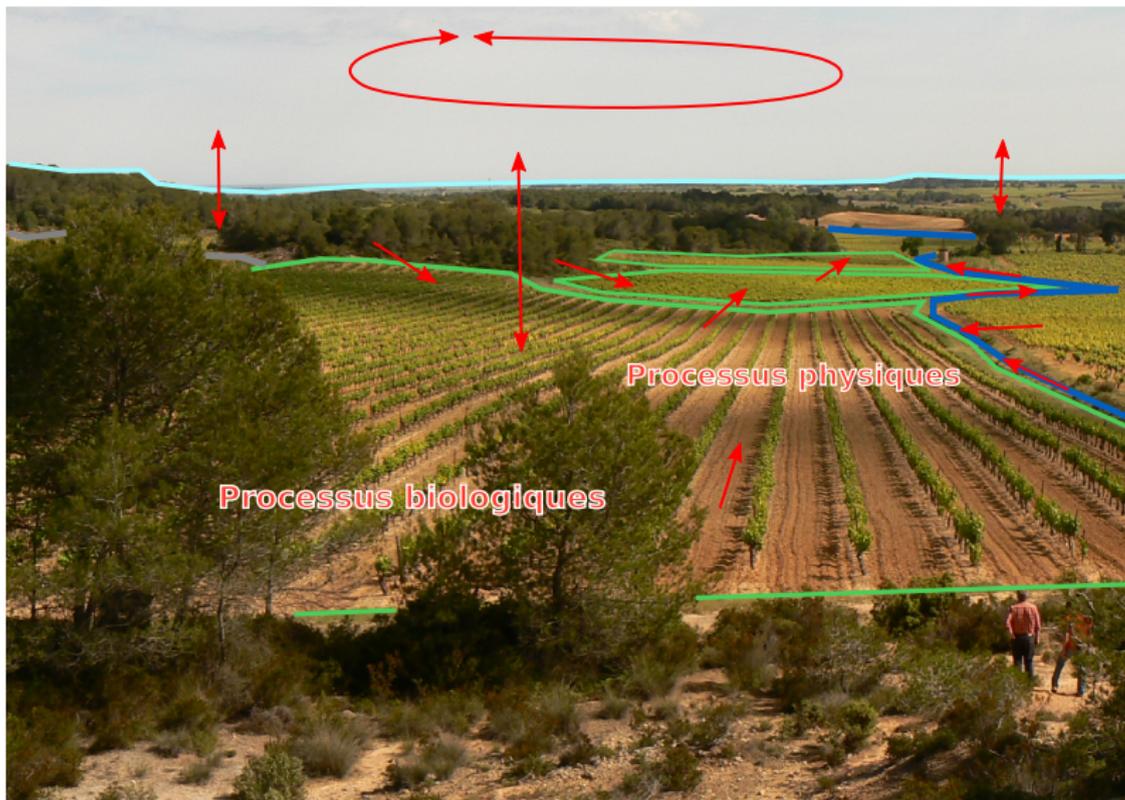
# Le paysage



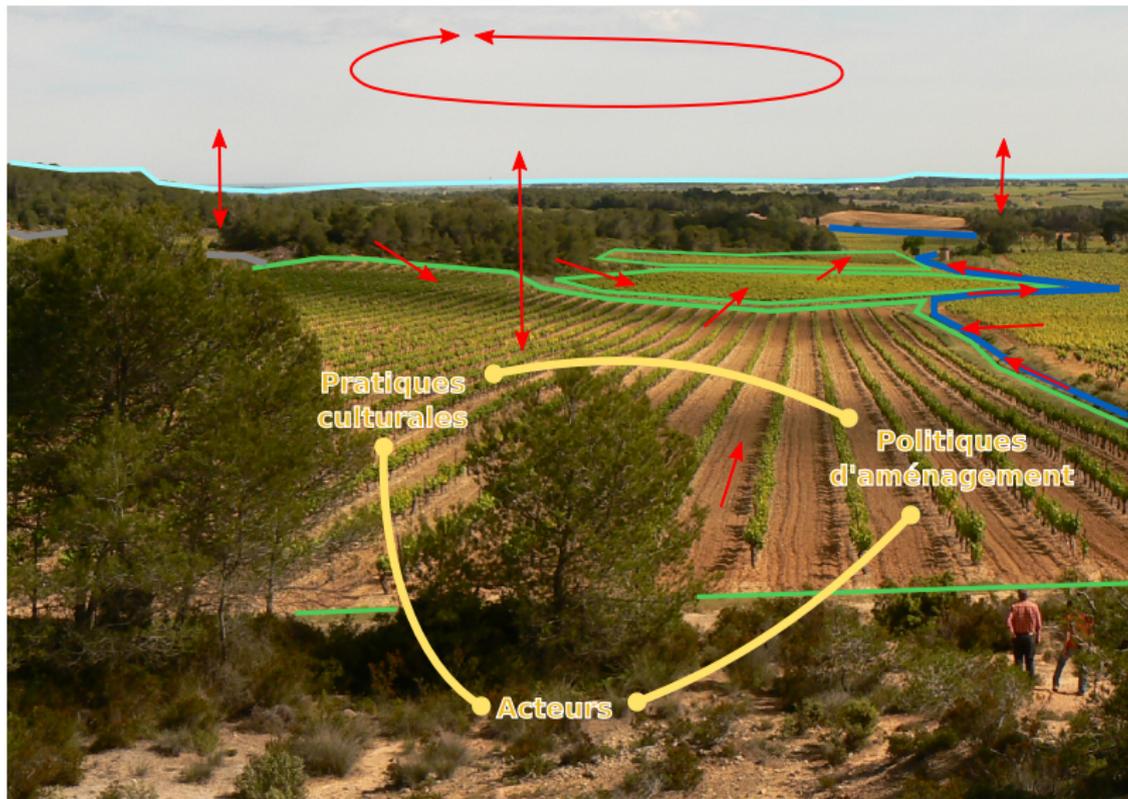
# Le paysage



# Le paysage



# Le paysage



# Etude du paysage

Le paysage est un **système complexe spatialisé, multifonctionnel**, où de nombreux **objets et processus sont en fortes interactions**.  
(agrosystème, écosystème, agroécosystème)

Etude de l'organisation et du fonctionnement du paysage  
⇒ modélisation **pluridisciplinaire** de la **structure** et de la **dynamique** du système

Selon les objectifs:

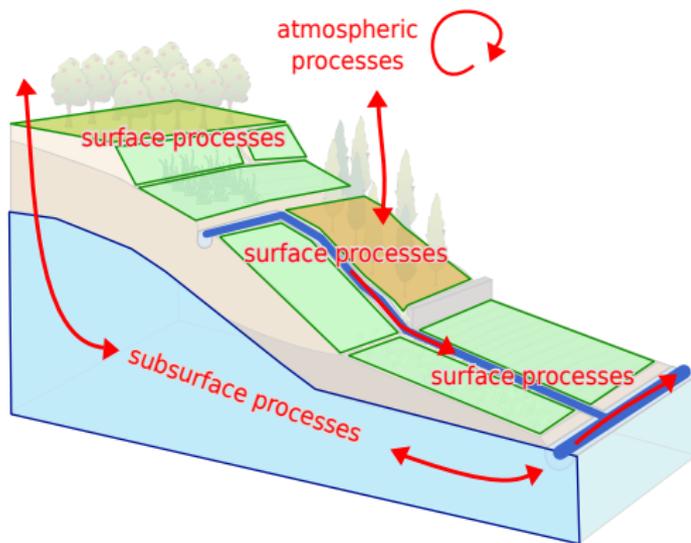
- choix des échelles,
- détermination des éléments pertinents dans le paysage, et des relations entre ces éléments,
- identification des processus à prendre en compte,
- ...

# Modélisation du fonctionnement du paysage

Représentation **numérique**  
du paysage

+

Modélisation des **processus**  
en **interaction**



# Modélisation du fonctionnement du paysage

Représentation **numérique**  
du paysage

+

Modélisation des **processus**  
en **interaction**

- **éléments du paysage**  
(parcelles, routes, fossés, nappes, . . .)
- **propriétés** de ces éléments  
(géométrie, propriétés physiques, . . .)
- **relations/connexions** entre ces éléments (topologiques et hiérarchiques)



- **dynamiques locales et spatiales** des processus  
(transferts, évolutions, décisions, . . .)
- **couplage** entre les processus (interactions, rétroactions)

# Vers une plateforme de modélisation. . .

Début 90's: développement de modèles spatio-temporels

1995: développement du modèle MHYDAS au LISAH

2000-2005: couplages de plus en plus nombreux, "explosion" des codes sources de modèles

2005-2006: réflexion pour le développement d'une **plateforme de modélisation**:

⇒ moderniser l'infrastructure de modélisation, normaliser les développements, se doter d'un socle commun pour la modélisation

Août 2006: première ligne de code OpenFLUID

Juillet 2007: première formation OpenFLUID (interne au LISAH)

# Fondements d'OpenFLUID

Développer et mettre à disposition une **plateforme logicielle ouverte** pour la **modélisation spatialisée dans les paysages**

- Intégration de représentations numériques de l'espace
- Branchement de modèles, couplage automatique

Proposer un **socle collaboratif** pour des **approches pluri-disciplinaires** de la modélisation des paysages

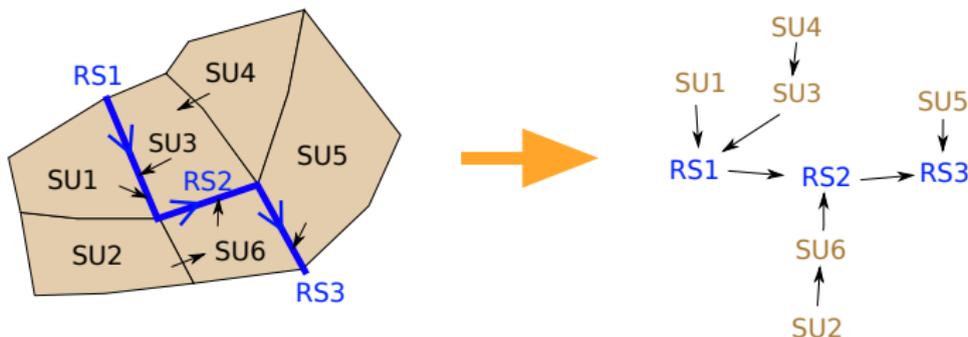
Accompagner les **utilisateurs** dans le **développement**, la **capitalisation**, le **partage**, le **couplage** de modèles spatiaux



# Représentation de l'espace sous OpenFLUID

L'espace est représenté par OpenFLUID sous la forme d'un **graphe orienté**

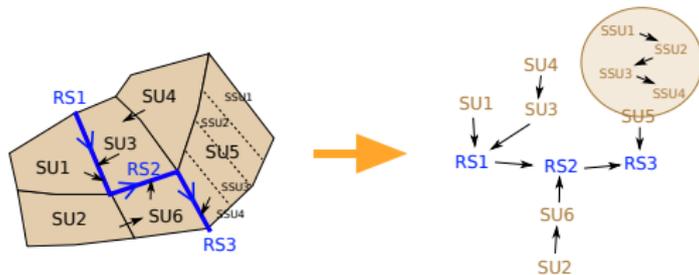
- Les noeuds sont les **unités spatiales** composant l'espace, rangés par **classes d'unités spatiales**
- Les arcs orientés sont les **relations** entre les unités spatiales
- Chaque noeud porte des **attributs propres** à l'unité spatiale qu'il représente



Nombreux **algorithmes** disponibles sur les graphes  
(calculs de connectivité, parcours, parallélisation, ...)

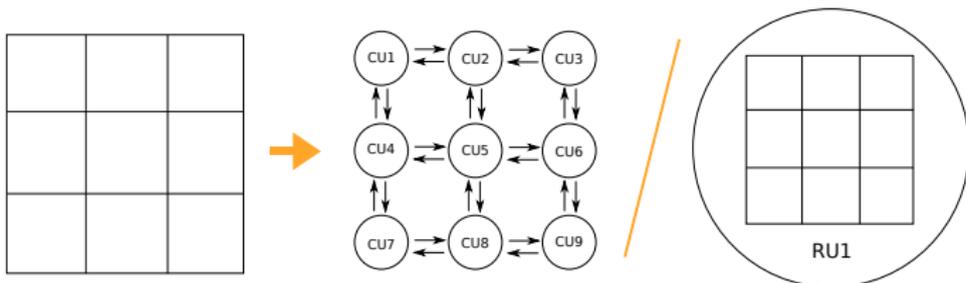
# Représentation de l'espace sous OpenFLUID

Un noeud peut contenir un graphe pour des approches **multi-échelles**



Pour une modélisation orientée **raster**, différentes approches sont possibles:

- 1 unité spatiale par cellule du raster
- 1 unité spatiale portant une **matrice correspondant au raster**



# Représentation de la dynamique sous OpenFLUID

La dynamique est représentée par des **simulateurs** OpenFLUID  
= **codes de calcul** développés à partir de modèles mathématiques

- Il simule un ou plusieurs processus spatiaux (transferts, évolutions, ...)
- Il fait **évoluer les états** des unités spatiales
- Il peut utiliser l'ensemble des informations de simulation disponibles pour ses calculs (attributs spatiaux, variables, ...)

Un simulateur est construit sous la forme  
d'un **plugin** pour la plateforme OpenFLUID:

- développé "from scratch"
- **encapsulation d'un code de calcul** existant



Possibilité de **parallélisation automatique des calculs**, basée sur la structure du graphe d'espace

# Représentation de la dynamique sous OpenFLUID

## Signature des simulateurs et couplage

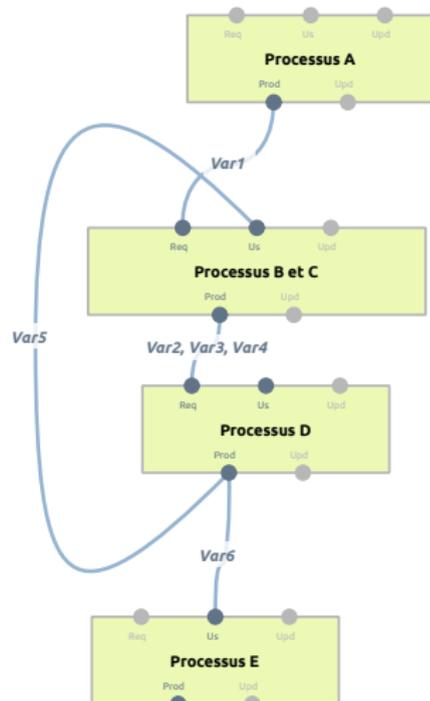
Chaque simulateur déclare son **comportement** au travers de sa **signature**:

- **Variables** requises et/ou utilisées en entrée, produites et/ou mises à jour en sortie
- **Attributs spatiaux** requis ou utilisés
- Pas de temps, modifications spatiales possibles, ...

ainsi que des **métadonnées**

- Nom de l'auteur, description, version ...

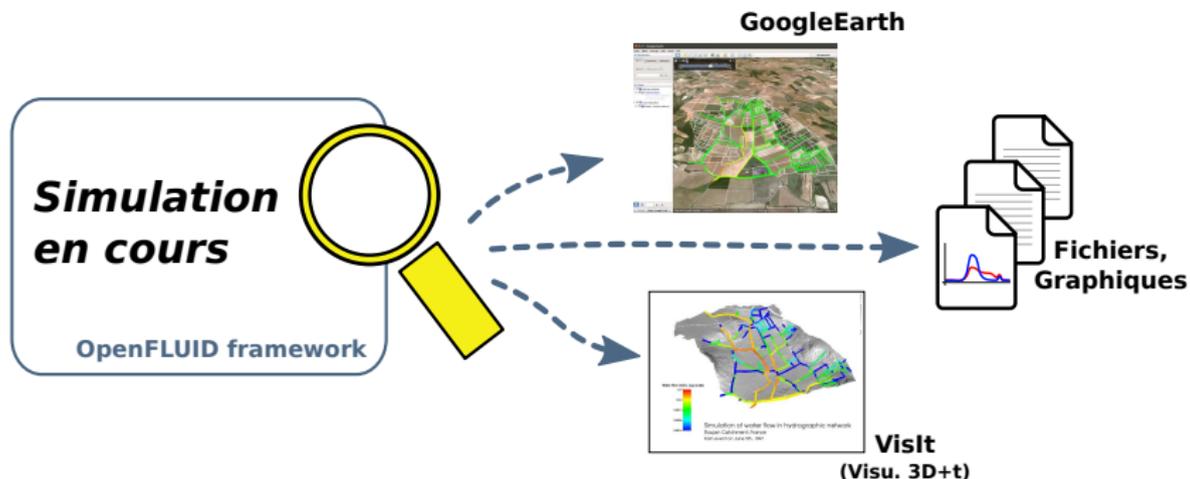
Le couplage de simulateurs est **réalisé et vérifié automatiquement** par la plateforme à partir des signatures des simulateurs mobilisés



# Observation des simulations OpenFLUID

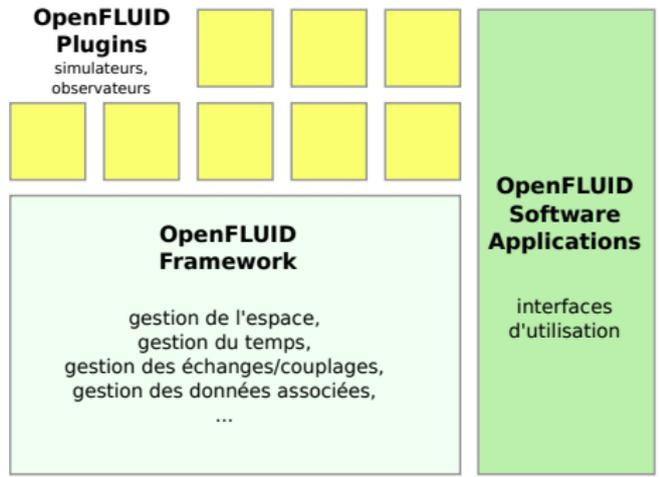
Les données de simulation peuvent être observées tout au long des simulations via des **observateurs** (plug-ins)

- export de données, contrôles, ...
- une simulation peut comporter de 0 à n observateurs qui composent le **monitoring** de simulation



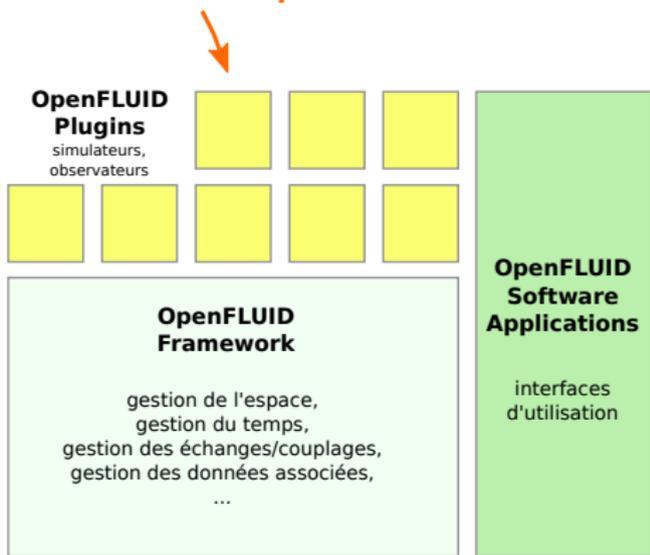
# Plateforme logicielle

# Infrastructure logicielle OpenFLUID



# Infrastructure logicielle OpenFLUID

*Connaissance scientifique  
et thématique*



*Mise en application,  
jeux de données*

*Ingénierie logicielle  
pour la modélisation spatio-temporelle*

# Utilisation en ligne de commande

Appli. **openfluid** en ligne de commande  
⇒ console, cluster de calcul, scripting, ...

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
* Building spatial domain... [OK]
* Building model... [OK]
* Building monitoring... [OK]
* Initializing parameters... [OK]
* Preparing data... [OK]
* Checking consistency... [OK]

Spatial domain, 629 units :
- GU, 25 units
- RS, 372 units
- SU, 232 units

Simulation from 1997-06-05 04:30:00 to 1997-06-05 16:00:00
Default DeltaT is 60 seconds

Size of buffers for variables is set to 692 (automatically computed)

**** Running simulation ****

Initialize... [OK]

Progress      Real time      Status
-----
0.14%        1997-06-05 04:31:00 [OK]
0.29%        1997-06-05 04:32:00 [OK]
0.43%        1997-06-05 04:33:00 [OK]
0.58%        1997-06-05 04:34:00 [OK]
0.72%        1997-06-05 04:35:00 [OK]
0.87%        1997-06-05 04:36:00 [OK]
1.01%        1997-06-05 04:37:00 [OK]
1.16%        1997-06-05 04:38:00 [OK]
1.30%        1997-06-05 04:39:00 [OK]
```

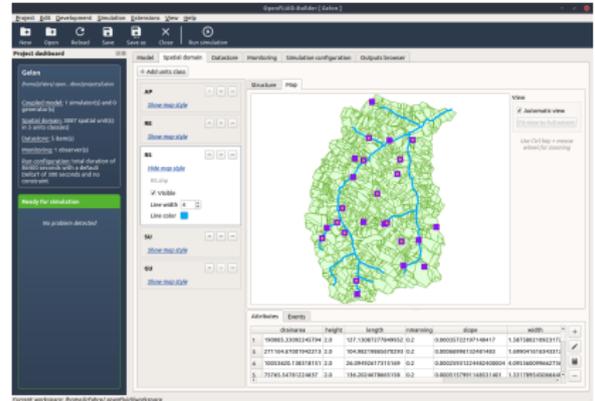
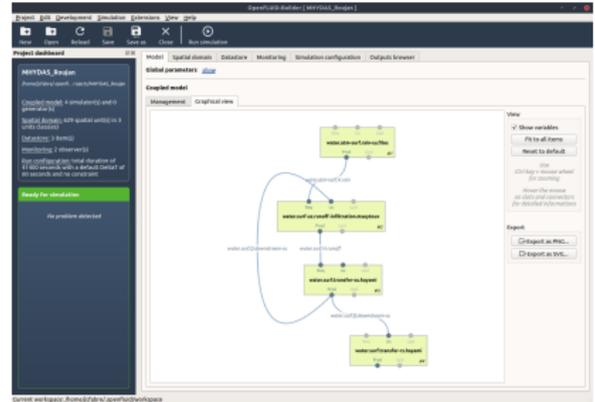
# Utilisation en interface graphique

## Appli. OpenFLUID-Builder

- Préparation des données
- Paramétrage des simulations
- Exécution des simulations
- Exploitation des résultats

## Extensible par ajout de plugins Builder-extensions:

- import de données,
- aide au paramétrage,
- visualisation,
- ...



# OpenFLUID en environnement externe

## bindings

Package **ROpenFLUID**: pilotage d'OpenFLUID depuis l'environnement GNU R

- Paramétrage, exécution de simulations, exploitation des résultats

⇒ Profiter des fonctionnalités d'**exploration de simulations** sous R

## Lancement d'une simulation sous R

```
library('ROpenFLUID')  
ofsim = OpenFLUID.loadDataset('/path/to/dataset')  
OpenFLUID.runSimulation(ofsim)  
data = OpenFLUID.loadResult(ofsim, 'SU', 15, 'water.surf.Q.downstream')
```

Module **PyOpenFLUID** : pilotage d'OpenFLUID depuis le langage Python

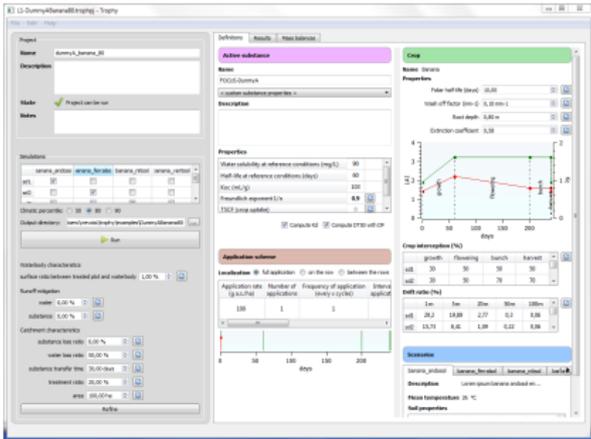
Module **OpenFLUIDjs** : pilotage d'OpenFLUID depuis le langage Javascript

# OpenFLUID en environnement externe embedding

Framework OpenFLUID **embarqué au coeur d'outils indépendants** en tant que moteur de simulation

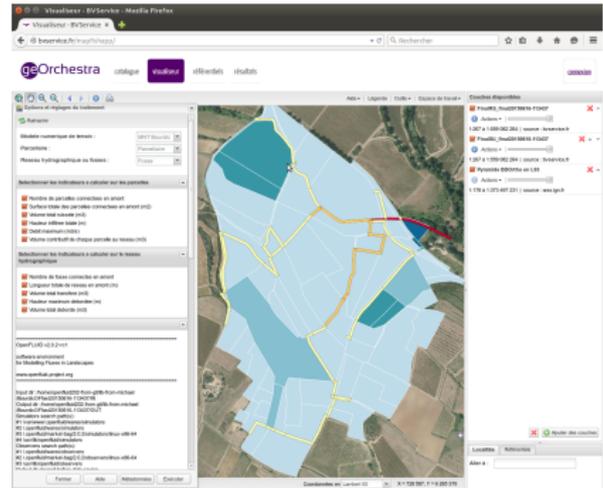
## Trophy (ANSES)

outil de validation de molécules pesticides en milieu tropical

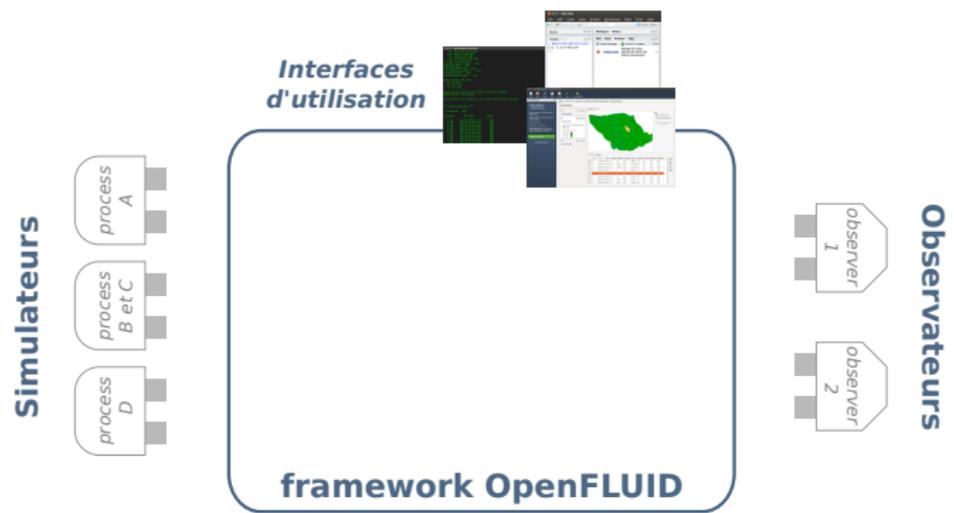


## BVservice (AFB)

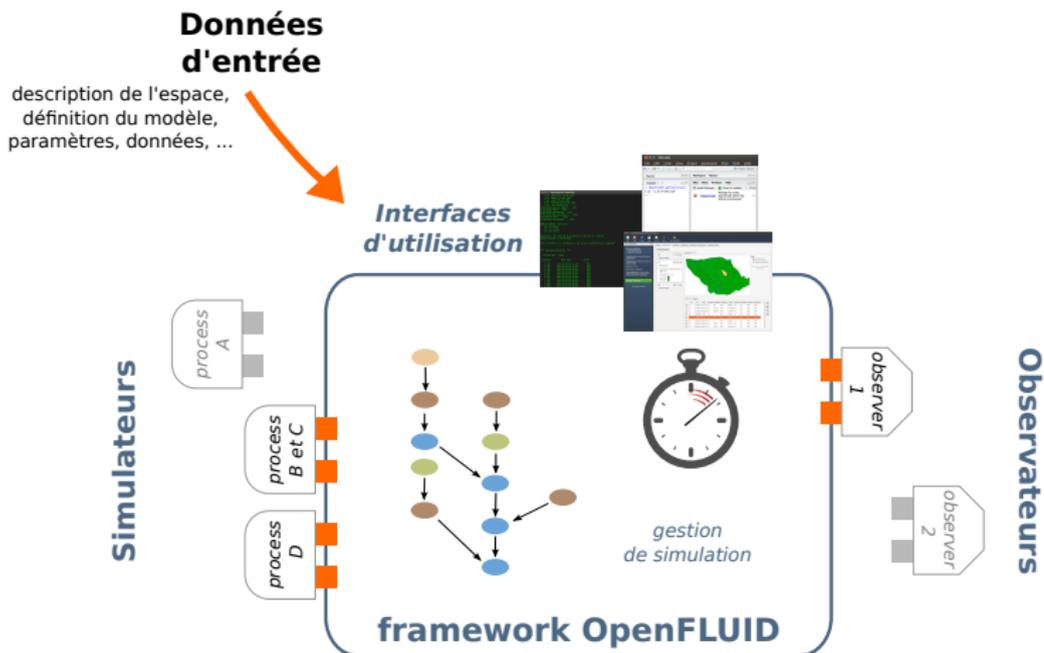
outil en ligne de diagnostic BV



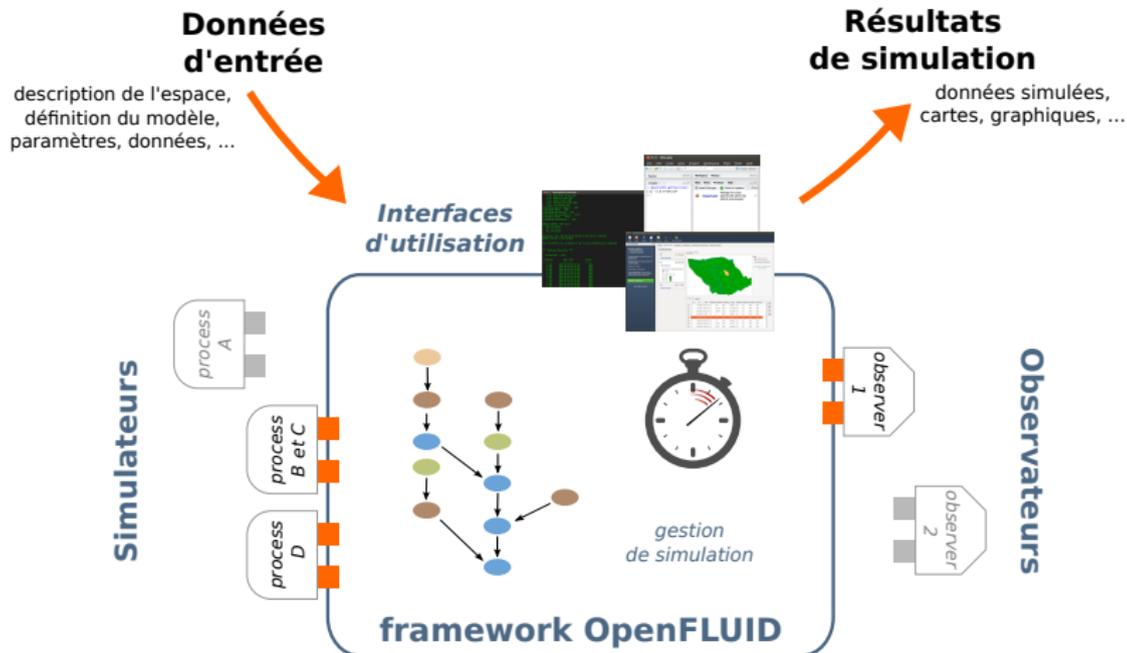
# Démarche générale de simulation



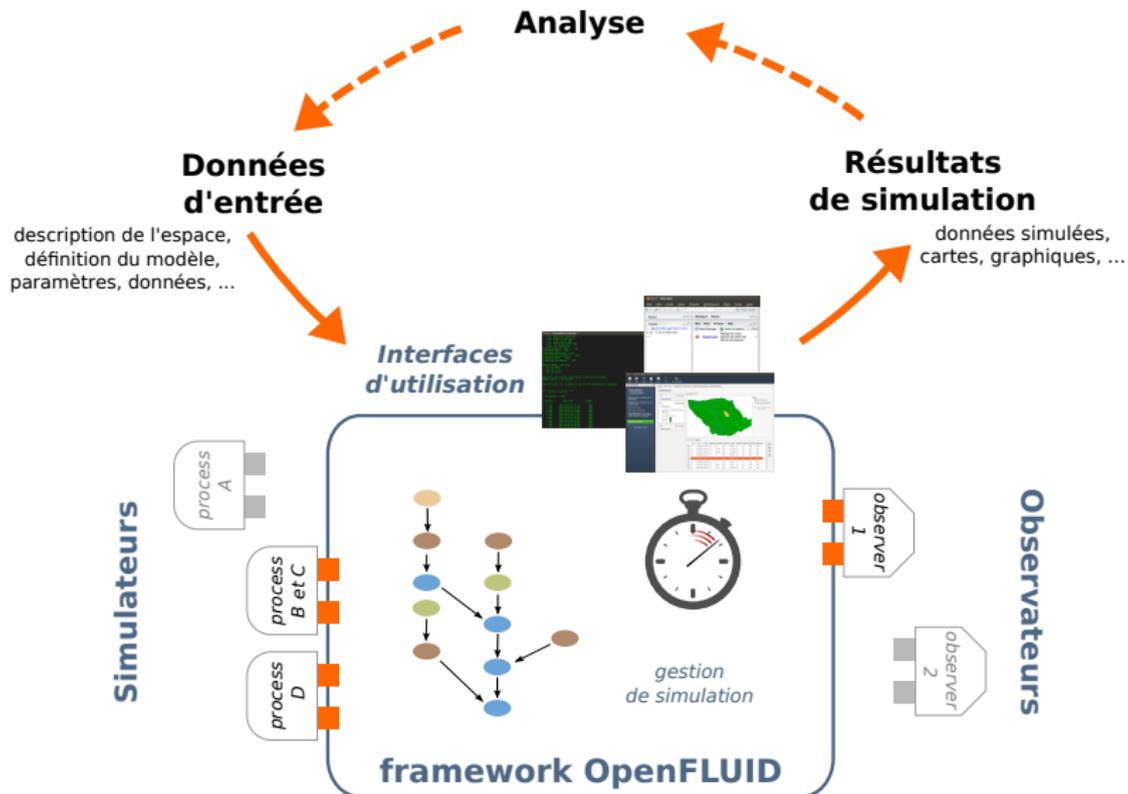
# Démarche générale de simulation



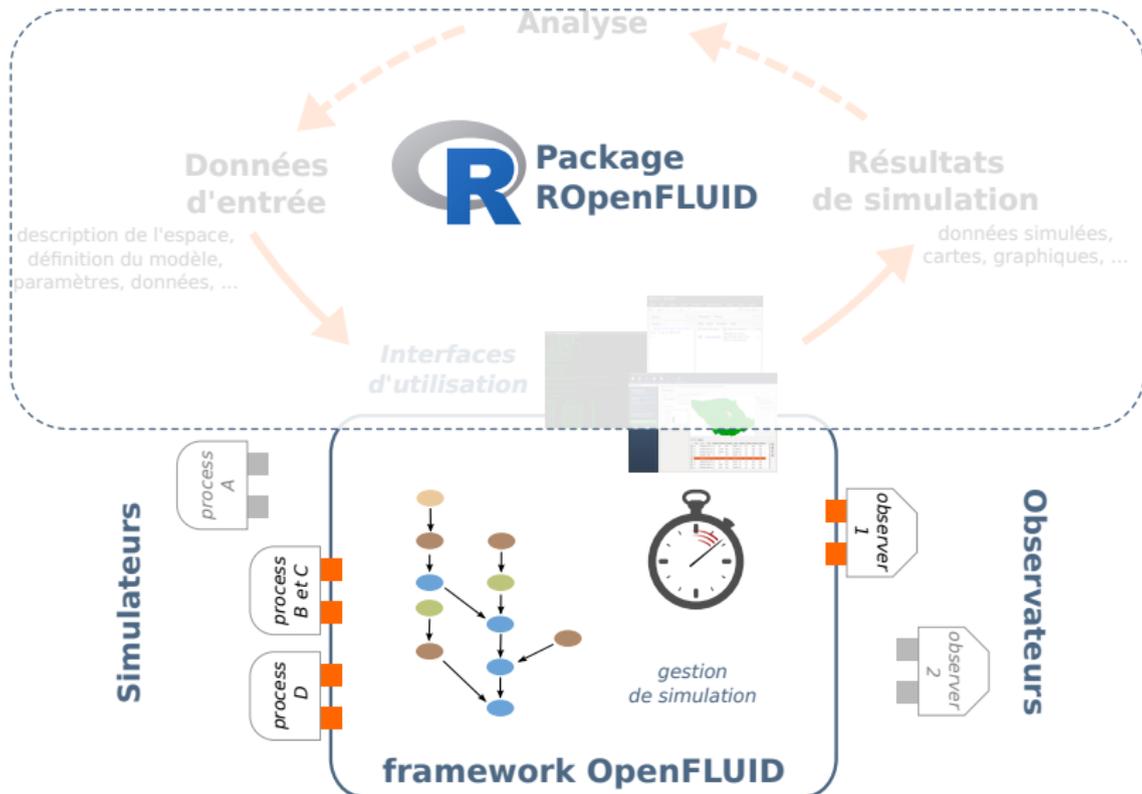
# Démarche générale de simulation



# Démarche générale de simulation



# Démarche générale de simulation

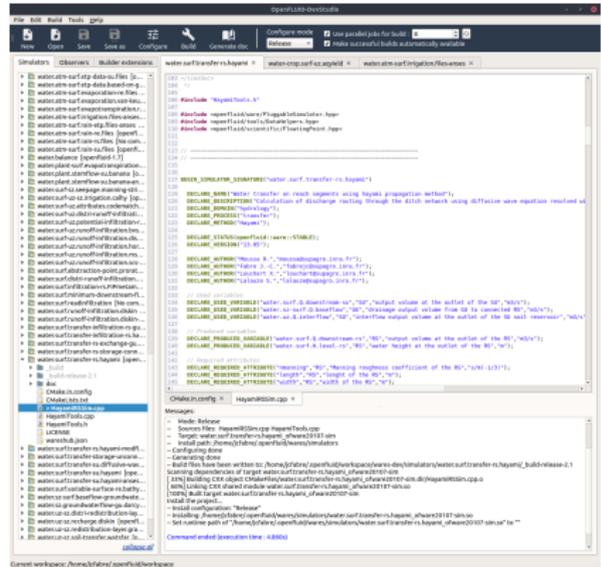


# Interface de développement

simulateurs, observateurs, builder-extensions

## Appli. OpenFLUID-DevStudio

- Génération automatique du code source (sans code de calcul)
- Assistance au développement
- Automatisation de la construction
- Génération de la **documentation scientifique des simulateurs** en PDF (avec possibilité de syntaxe  $\text{\LaTeX}$ )
- ...



## Langages pour les simulateurs:

- nativement C++, C, Fortran, ...
- par intégration pour R, Python, ...

# OpenFLUID-WaresHub

Espace en ligne pour la **capitalisation** et le **partage** de simulateurs, observateurs, extensions graphiques

- gestion du **code source** (git)
- gestion des **tâches** (bugs, évolutions, review)
- gestion des **droits d'accès**
- mise à disposition de la **documentation associée**
- **méta-information** : licence, contributeurs, contact(s), ...

⇒ **Automatisation de traitements** (tests, packaging, ...)

Déployable par projet, groupe de travail, labo, ...

server  
OpenFLUID-WaresHub



Devel. Environment  
local



# Exemple de dépôt de modèles partagés

The screenshot shows the LISAH WaresHub interface in a Mozilla Firefox browser. The page title is "LISAH WaresHub" and the user is logged in as "fabrejc". The main content area features a navigation bar with three buttons: "Simulators" (85), "Observers" (1), and "Builder-extensions" (1). Below this is a search bar with the text "Searched terms" and a "Search" button. The main content is a table listing simulators with columns for ID, Doc, and OpenFLUID compatibility.

| ID   | Doc   | OpenFLUID compatibility |
|--|---|-------------------------|
| <a href="#">energy.atm-surf.daily-min-max-temperature-data.files</a> | Compute daily min and max temperature data from files<br>📄 📄 📄                      | ★ 1.7 and previous      |
| <a href="#">energy.atm-surf.global-radiation-data.files</a>          | Compute Global Radiation (GR) data from files<br>📄 📄 📄                              | ★ 2.0 and previous      |
| <a href="#">energy.atm-surf.temperature-data.files</a>               | Generate temperature data from files<br>📄 📄 📄                                       | ★ 2.0 and previous      |
| <a href="#">energy.surf.rainfall-kinetic.zanchi-torri</a>            | Compute the kinetic energy of rainfall from Zanchi and Torri (1980) method<br>📄 📄 📄 | ★ 2.0 and previous      |
| <a href="#">erosion.surf.attenuation.-</a>                           |   | ★ 2.0 and previous      |
| <a href="#">erosion.surf.detachment-rs.-</a>                         |   | ★ 2.0 and previous      |

# Exemple de dépôt de modèles partagés

Wareshub - Mozilla Firefox

Wareshub

https://coding.umr-lisah.fr/lisah-wareshub/report/index.php?waretypes=

LISAH WaresHub

fabrejc Sign out

/ simulators / water.surf.transfer-rs.hayami

Discharge routing through the ditch network using diffusive wave equation resolved with Hayami method and discharge conversion to water height

Documentation: PDF

OpenFLUID version(s): 2.0 and 1.7, 1.6, 1.5, 1.4.2, 1.4.1, 1.4.0

4 contributors:

- Jean-Christophe Fabre  
32 commit(s)
- Simon Lalauze  
23 commit(s)
- Xavier Louchart  
2 commit(s)
- Aline Libres  
1 commit(s)

git access

URL:  
`https://fabrejc@coding.umr-lisah.fr/lisah-wareshub/git/s`

- Read access for all users
- Write access for chataign, crevoisi, fabrejc, rabotin

Branch: openfluid-2.0

General information | Commits history | Issues

#water #surface #network #transfer #hayami

Development status: stable

License: GPL-3.0

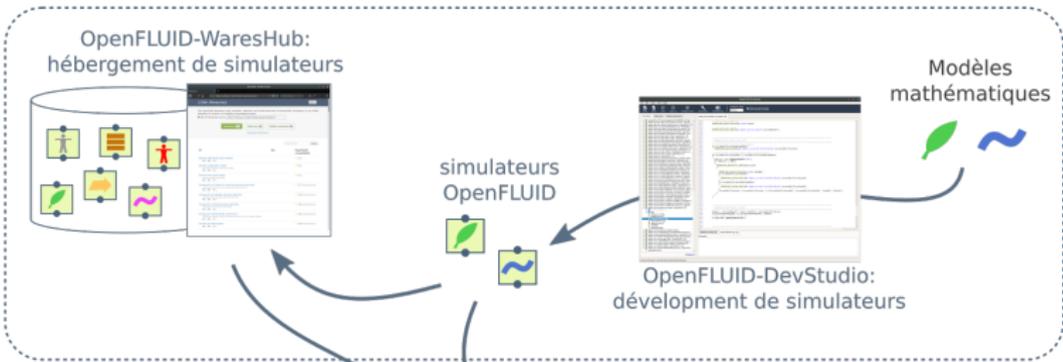
Contacts:  
[contact@openfluid-project.org](mailto:contact@openfluid-project.org)

LISAH-WaresHub - contact

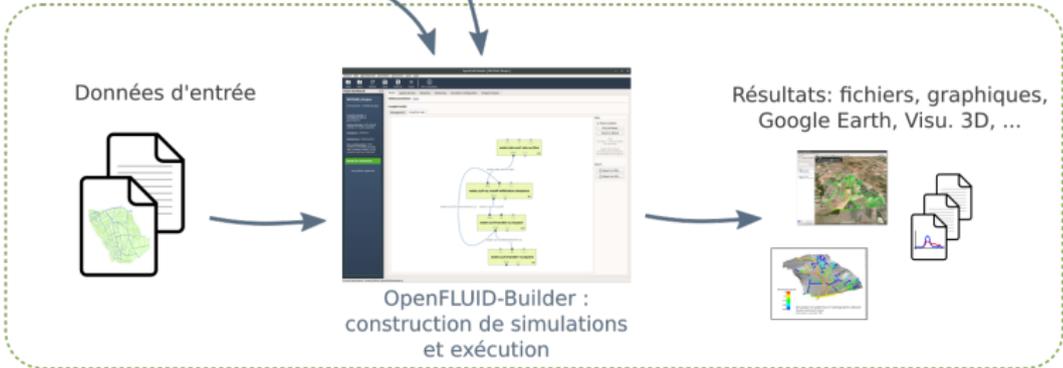
# Cycle de vies des simulateurs

développement → capitalisation → (ré)utilisation

**Développement  
de simulateurs**



**Paramétrage  
et simulation**





# Hydrologie de surface intra-parcellaire

Parcelle AW6 - Roujan (*EU Life-Aware, X. Louchart*)



## Parcelle

- 1200  $m^2$ , 1070 unités spatiales
- 4 simulateurs

## Simulation des chemins de l'eau en intra-parcellaire

- sur 1 heure
- pas de temps : 10 s

# Transfert en réseau hydrographique

## Bassin versant de Roujan (*R. Moussa*)

### Bassin versant de Roujan :

- $0.91 \text{ km}^2$
- ~600 unités spatiales

### Simulation :

- sur 7 heures
- pas de temps : 60 s

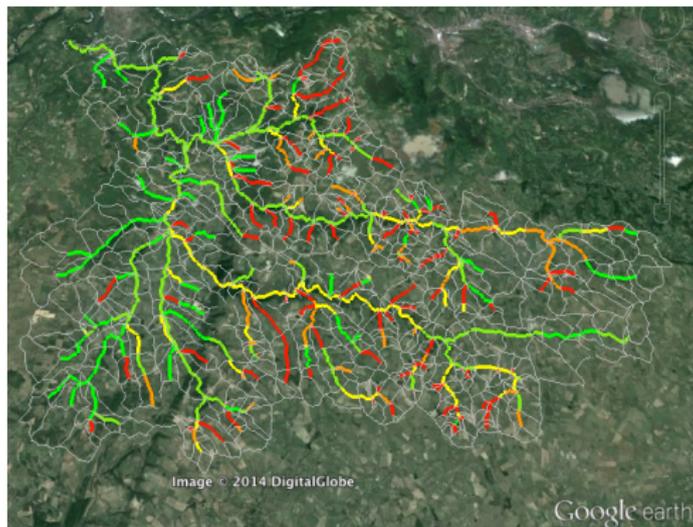


# Impact hydrologique de retenues et prélèvements

Diège (collab. LISAH - BE CEREG, transferts de technologie et de compétences)

## Evaluation de l'impact hydrologique de retenues en tenant compte des prélèvements pratiqués

- 165km<sup>2</sup>, 1115 unités spatiales
- 8 simulateurs couplés
- simulations sur 30 ans,  $\Delta t = 1$  journée
- différents scénarios d'aménagement



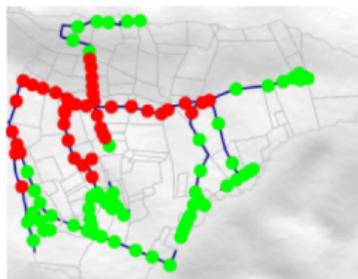
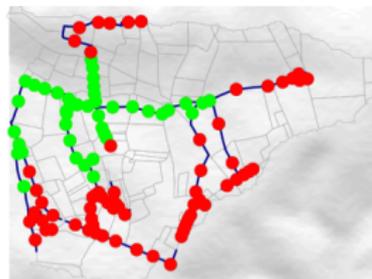


# Limitation de la contamination des masses d'eau

Ruiné (ONEMA FIP, LISAH)

Méthodologie de **diagnostic et de gestion**  
des réseaux de **fossés agricoles infiltrants**  
pour la **limitation de la contamination des masses d'eau**  
par les pesticides

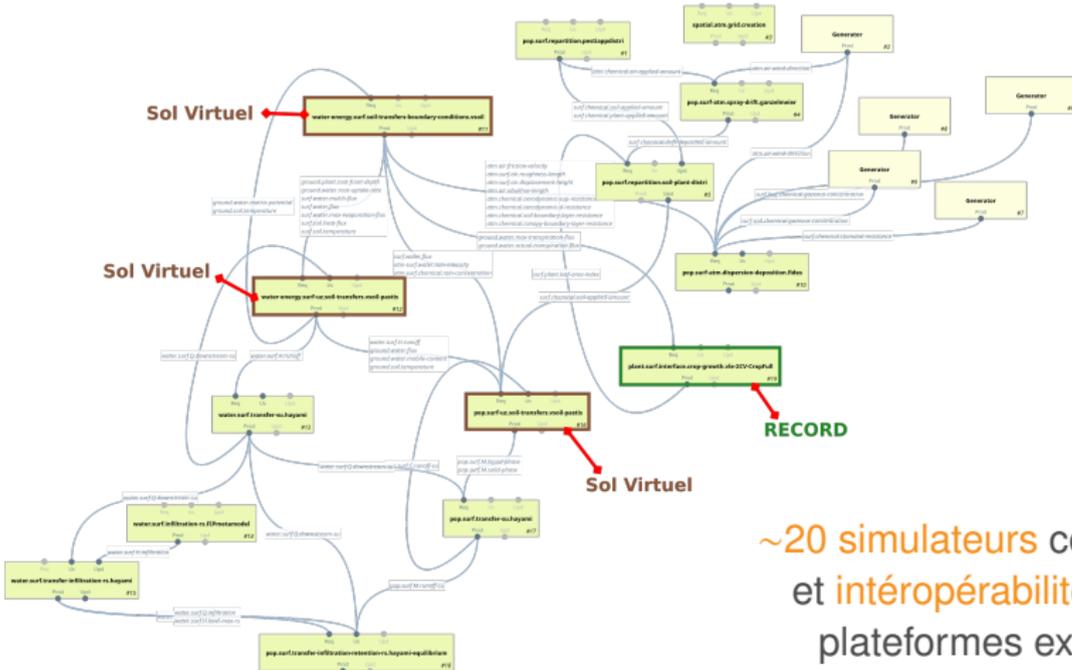
*Ex: Rétention des pesticides selon gestion spatiale de l'entretien des fossés*



- fauche/brûlis
- curage

# Modélisation Intégrée des Pesticides dans les Paysages

(EA MIPP, collab. ECOSYS, ISPA, ITAP, LISAH, PIAF, OpenFLUID, RECORD, SoIVirtuel)



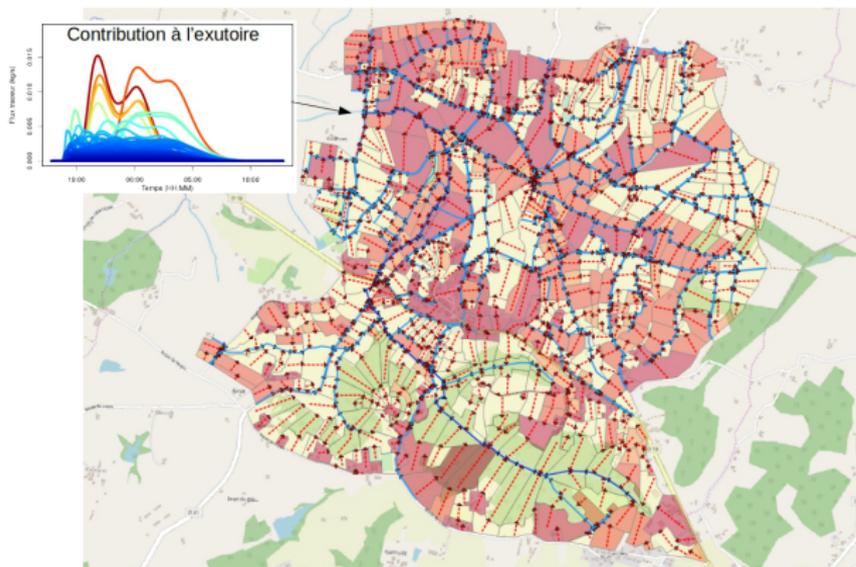
~20 simulateurs couplés  
et intéropérabilité avec  
plateformes externes

# Vulnérabilité des territoires aux pollutions diffuses

Doazit (BE Envilys)

Evaluation de la **contribution hydrologique de chaque parcelle**  
sur les écoulements dans le réseau hydrographique

Traceur pour l'évaluation des pollutions diffuses (nitrates ou pesticides)

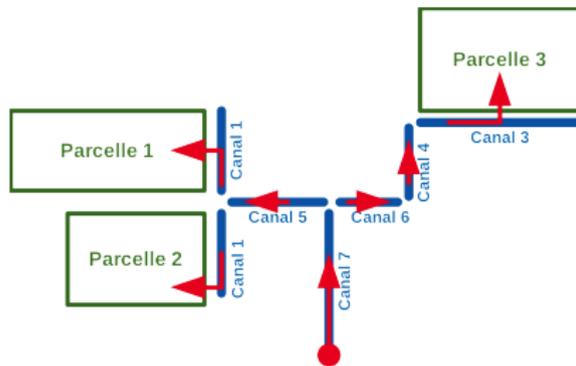


- 905 ha, 62.2 km de réseau
- 935 parcelles, 754 tronçons de réseau
- 6 simulateurs couplés
- simulations sur 24 h,  $\Delta t = 2$  min

# Efficience de l'irrigation au calan

Domaine du Merle, Plaine de la Crau (Stage S.Roman, collab. G-Eau)

Diagnostic de l'efficience des tours d'irrigation par simulation  
débits et pertes dans le réseau, humidité des parcelles, croissance des cultures



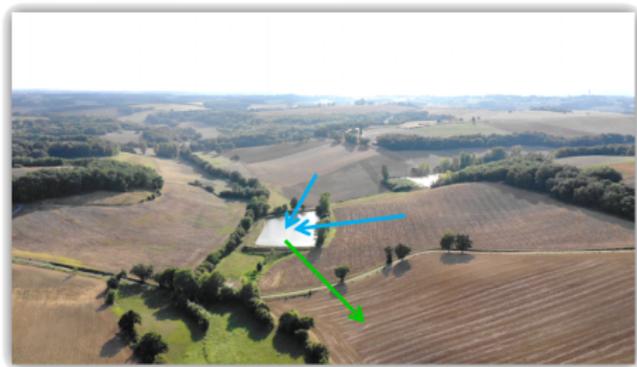
- Branche Nord n°6
- Réseau : 2400 m, débit de 150 l/s
- Parcellaire : 37 ha, foin
- Calendriers d'irrigation

Développement du modèle Calhy  
à partir de concepts d'Optirrig (ex-PILOTE, module foin Crau)  
intégrés dans 2 simulateurs OpenFLUID

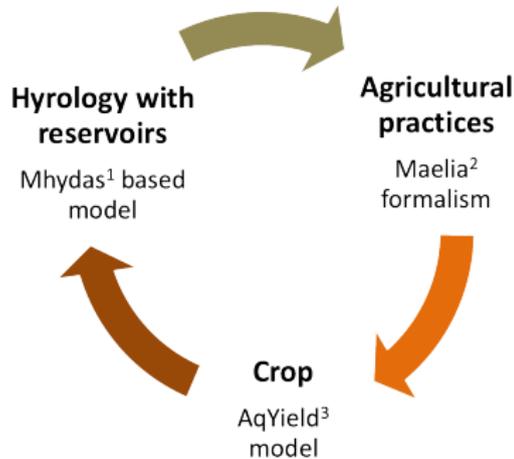
# Impact cumulé des retenues

(AFB Retenue, thèse N. Lebon, collab. LISAH - AGIR)

## DAHM-Reservoir : a Distributed Agro-Hydrological Model for Reservoir



→ lien topologique  
→ lien agronomique

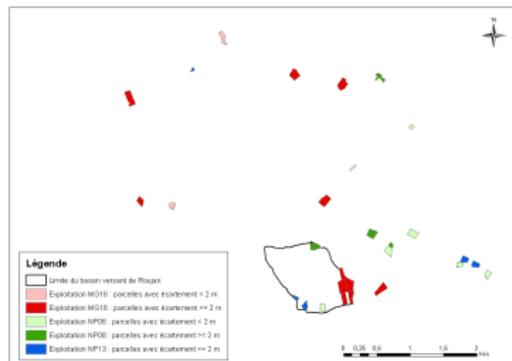
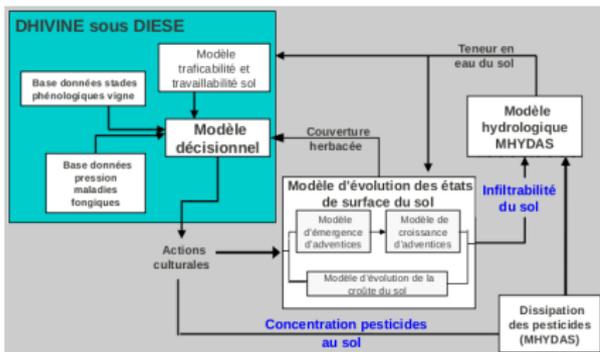


<sup>1</sup> Moussa et al., 2002; <sup>2</sup> Therond, 2018; <sup>3</sup> Constantin et al., 2015

# Couplage hydrologique - décision agronomique

La Peyne - (thèse N.Paré, collab. BIA Toulouse, ANR GeDuQuE, ANR SP3A)

Construction d'un **modèle couplé pression-impact** pour l'**expérimentation virtuelle** de **pratiques culturales** à l'échelle de petits bassins versants



**Encapsulation** du framework DIESE (Rellier et al., BIA Toulouse) dans un simulateur OpenFLUID

Couplages de 15 simulateurs (biophysique + décision)

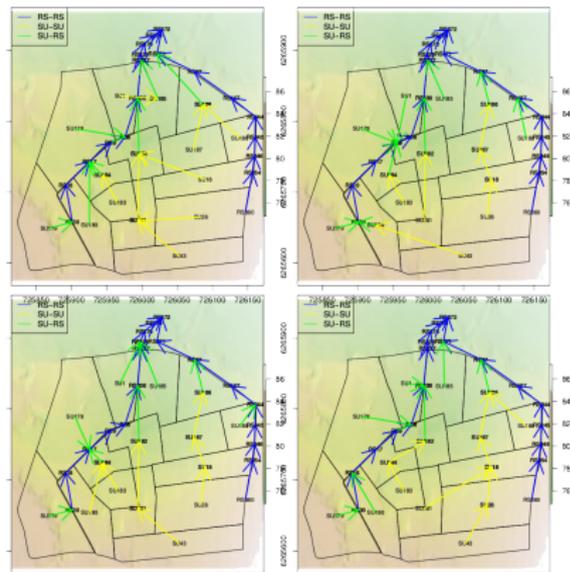
Génération dynamique d'**événements discrets** au cours de la simulation

# Analyse de sensibilité de paramètres spatiaux

Roujan (Projet ASM - Département EA INRA)

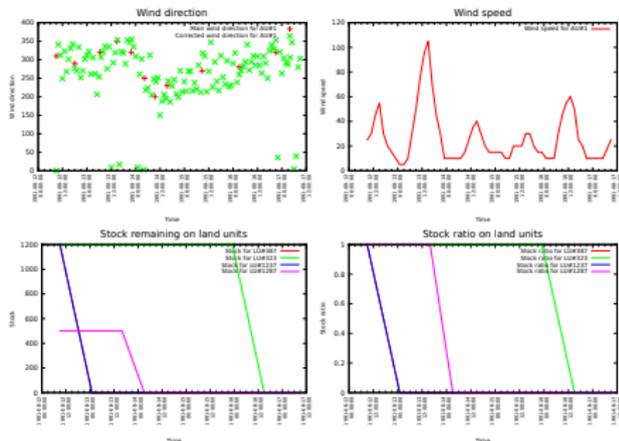
## Evaluation de l'impact de l'incertitude du MNT sur la connectivité topologique

- 100 cartes générées
- Simulations par le modèle MHYDAS
- Sensibilité du débit max et du volume au facteur connectivité en différents points du réseau hydrographique



# Propagation d'incendie

Haute vallée de l'Orb - (Exemple 'proof of concept')



Modèle simplifié de propagation d'incendie

Simulation sur grille (1516 unités spatiales, ~9500ha),  
Couplage de 2 simulateurs

Recalcul en cours de simulation du graphe de connexions spatiales en fonction  
des directions de vent

# Pour approfondir

# Types de données OpenFLUID

Les données manipulées par OpenFLUID durant la simulation sont typées

Différents types de données sont définis

Types simples:

- BooleanValue : nombre **booléen** (0 ou 1)
- IntegerValue : nombre **entier** (1,18, 56325874, ...)
- DoubleValue : nombre **réel en double précision** (1.52, 0.000025, ...)
- StringValue : **chaîne de caractères**

Types composés:

- VectorValue : **vecteur** (1D) de nombres en **double précision**
- MatrixValue : **matrice** (2D) de nombres en **double précision**
- MapValue : **liste clé-valeur** de tout autre type de données

# Variables

Les variables d'état sont **centrales dans le couplage** car **échangées entre simulateurs** tout au long de la simulation

- elles sont **attachées aux unités spatiales** via le graphe d'espace
- les **valeurs successives sont associées à un index de temps** dans la simulation (instant de production)
- les valeurs des variables sont ajoutées/modifiées par les simulateurs

Une variable d'état est présente sur toutes les unités spatiales d'une même classe d'unité

Les variables d'état peuvent être de n'importe quel type  
OpenFLUID

# Attributs spatiaux

Les attributs spatiaux sont des **données attachées aux unités spatiales**.

- *morphologie : surface, longueur, largeur, profondeur, ...*
- *propriétés physiques : conductivité, teneur en eau,...*
- *coefficients : manning, ...*
- *paramètres descriptifs : occupation du sol, ...*

Ils sont stockés sous la forme de chaînes de caractères (StringValue), et peuvent être convertis vers n'importe quel autre type OpenFLUID

Ils ne peuvent **plus être modifiés** une fois la simulation lancée.

## Paramètres de simulateurs et d'observateurs

Les **paramètres des simulateurs et des observateurs** sont des paramètres qui s'appliquent à un simulateur ou à un observateur en particulier.

Ils sont stockés sous la forme de chaînes de caractères (StringValue), et peuvent être convertis vers n'importe quel autre type OpenFLUID.

Il est possible de définir des **valeurs globales** pour des paramètres qui s'appliqueraient à plusieurs simulateurs ou observateurs.

## Evènements discrets

Les évènements discrets surviennent à un instant précis sur une unité spatiale donnée

- *opérations culturales : labours, épandage de produits, ...*
- *changement d'occupation du sol, rotation de culture, ...*
- *aménagements : curage d'un fossé, ...*

Ils portent des informations qui peuvent être traitées par les simulateurs.

Ces informations sont stockées sous la forme de chaînes de caractères (StringValue), et peuvent être converties vers n'importe quel autre type OpenFLUID

Ils peuvent être connus *à priori* sous la forme d'un **calendrier**, ou **générés** par les simulateurs **au cours de la simulation**.

## Banque de données intégrée : Datastore

Le datastore permet d'intégrer des **données non structurées**, en complément des données standards.

Actuellement, le datastore permet d'intégrer des données de type couche géographique **raster** ou **vecteur**.

A terme, intégration de nouvelles sources de données : liens BDs, chroniques, ...

Les données du datastore ne font pas partie des données de simulation, mais viennent en complément.

## Déroulement d'une simulation

Une simulation est définie par une **période de simulation** avec une date de début et une date de fin.

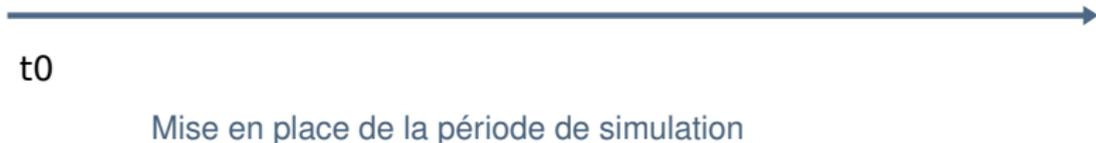
Un **deltat par défaut** est défini pour les simulateurs.

Une **contrainte de planification** peut être appliquée aux simulateurs

- Aucune contrainte
- La planification des simulateurs est **vérifiée** et doit être égale au deltat par défaut
- La planification des simulateurs est **forcée** pour être égale au deltat par défaut (**risqué!**)

## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



# Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



Positionnement de la simulation à  $t_0$



## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



Positionnement de simA pour sa prochaine exécution

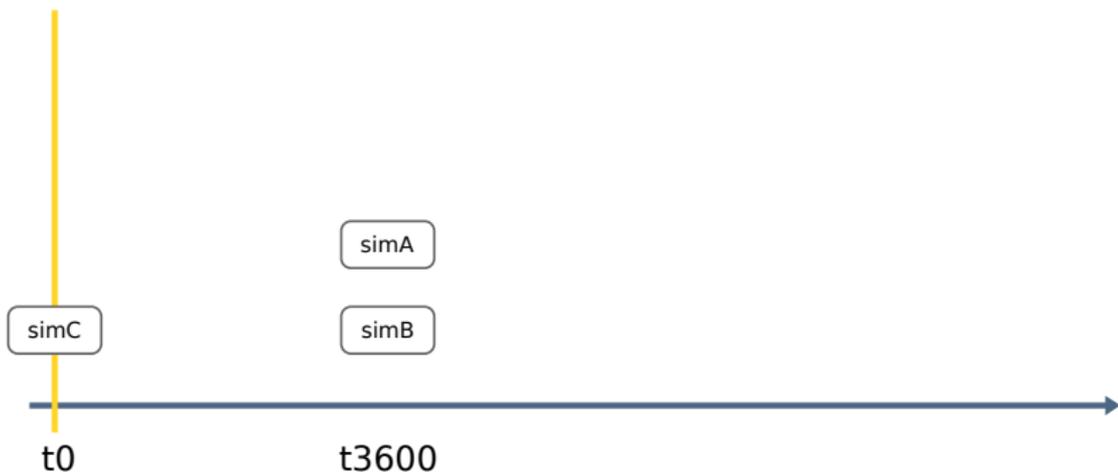
## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



# Déroulement d'une simulation

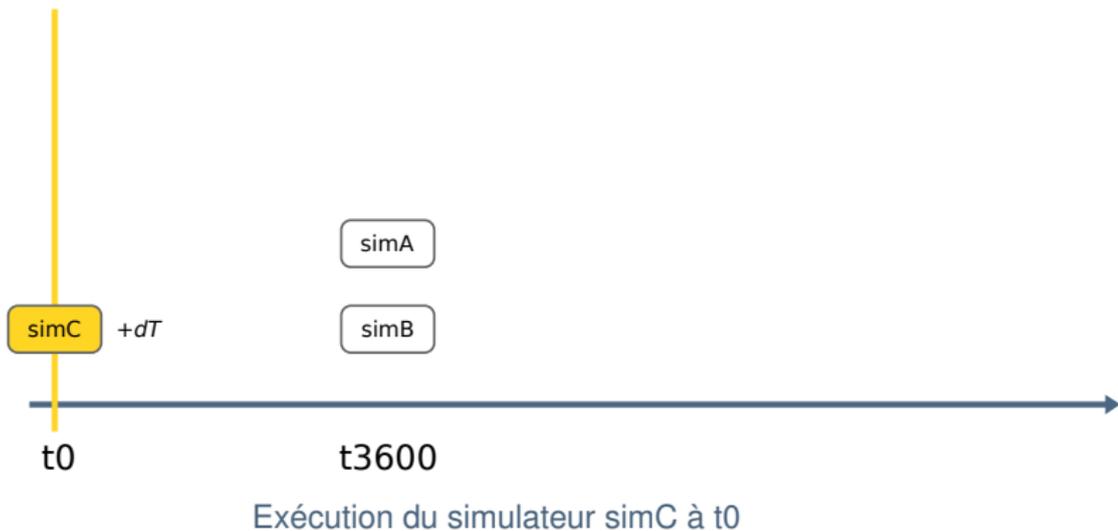
Exemple de simulation comprenant 3 simulateurs, avec un deltat (dt) par défaut de 1h00 (3600s), sans contrainte de planification.



Positionnement de simB pour sa prochaine exécution (réagencement)

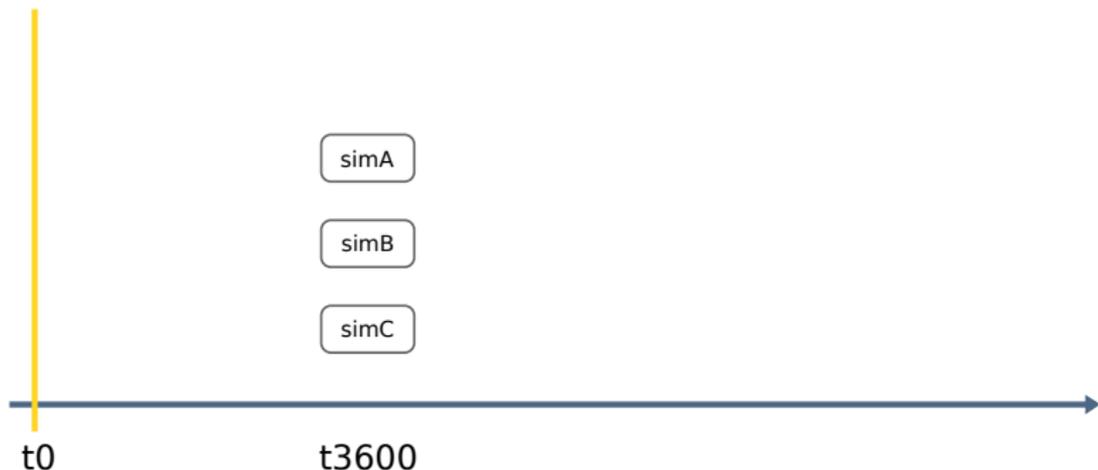
# Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**, avec un **deltat (dt)** par défaut de **1h00** (3600s), sans contrainte de planification.



## Déroulement d'une simulation

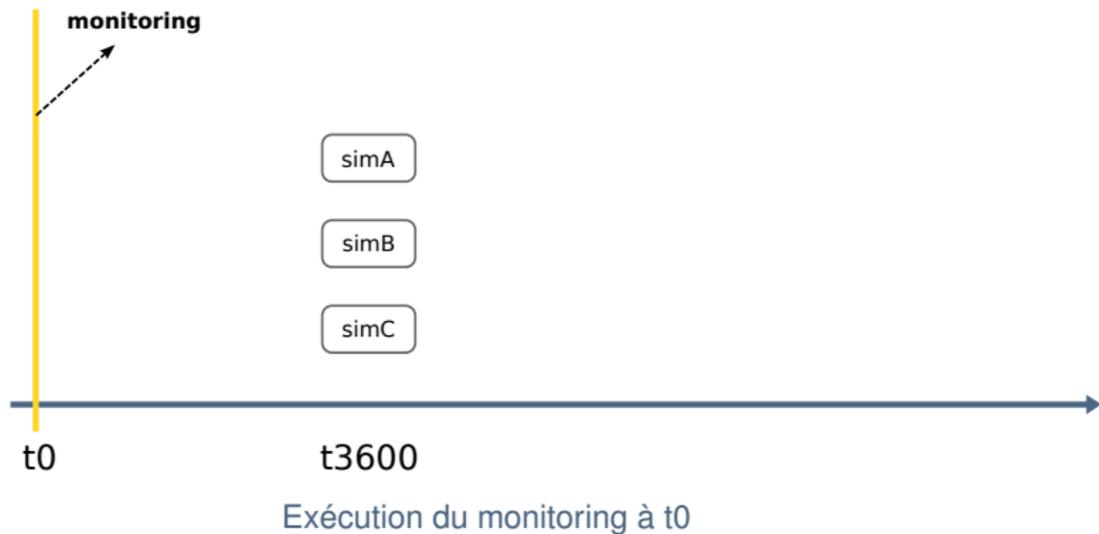
Exemple de simulation comprenant **3 simulateurs**,  
avec un  $\text{deltat (dt)}$  par défaut de **1h00** (3600s),  
sans contrainte de planification.



Positionnement de simC pour sa prochaine exécution

## Déroulement d'une simulation

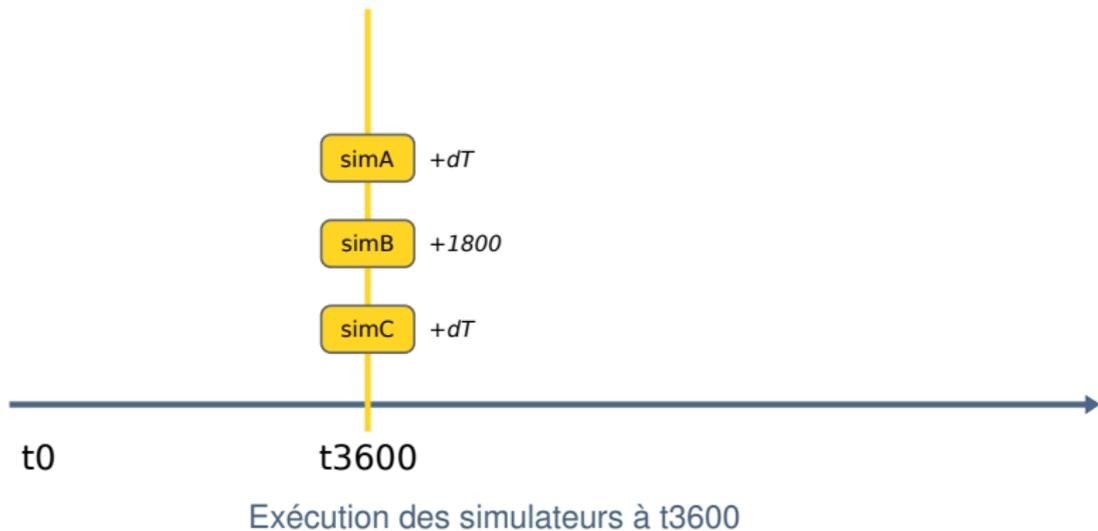
Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.





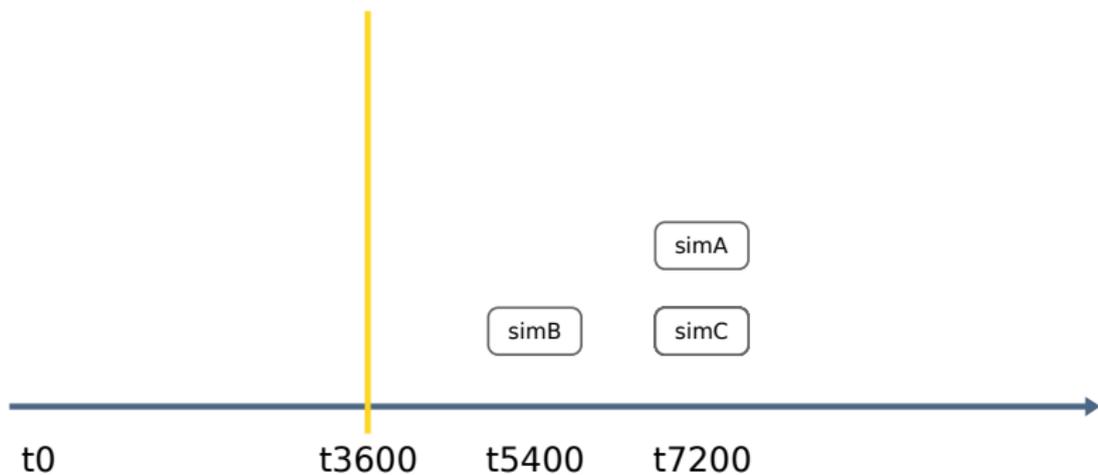
## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



## Déroulement d'une simulation

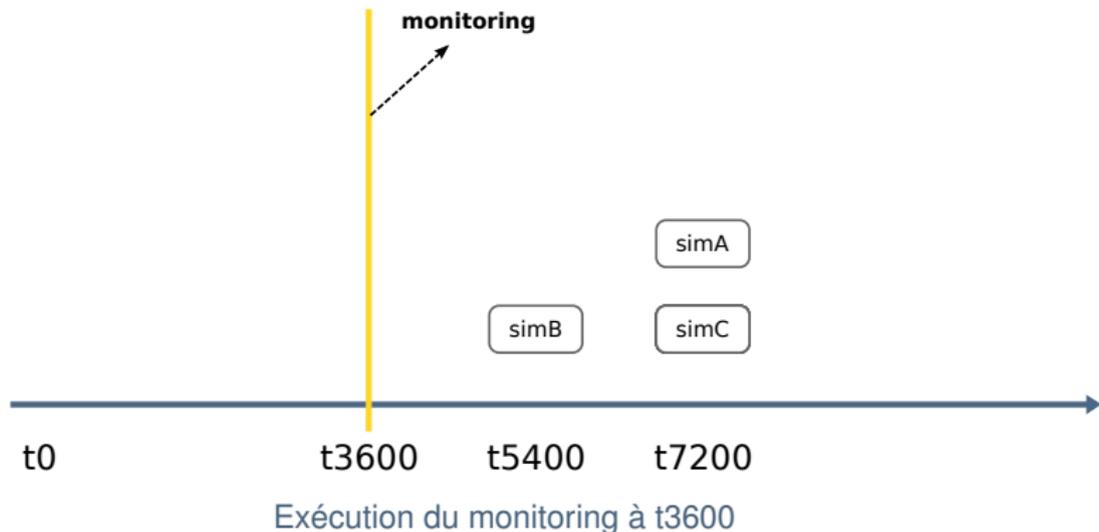
Exemple de simulation comprenant **3 simulateurs**, avec un **deltat (dt)** par défaut de **1h00 (3600s)**, sans contrainte de planification.



Positionnement des simulateurs pour leurs prochaines exécutions

## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**, avec un deltat (dt) par défaut de **1h00** (3600s), sans contrainte de planification.

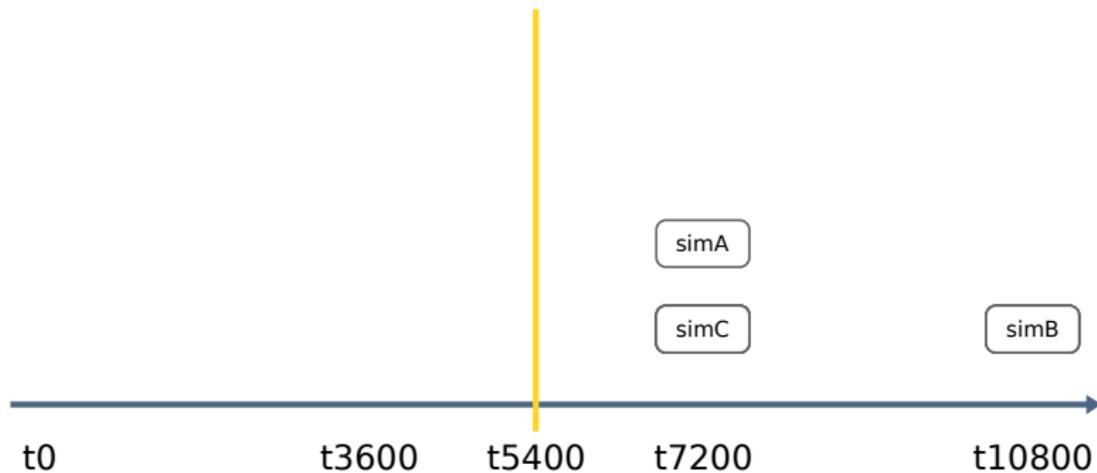






## Déroulement d'une simulation

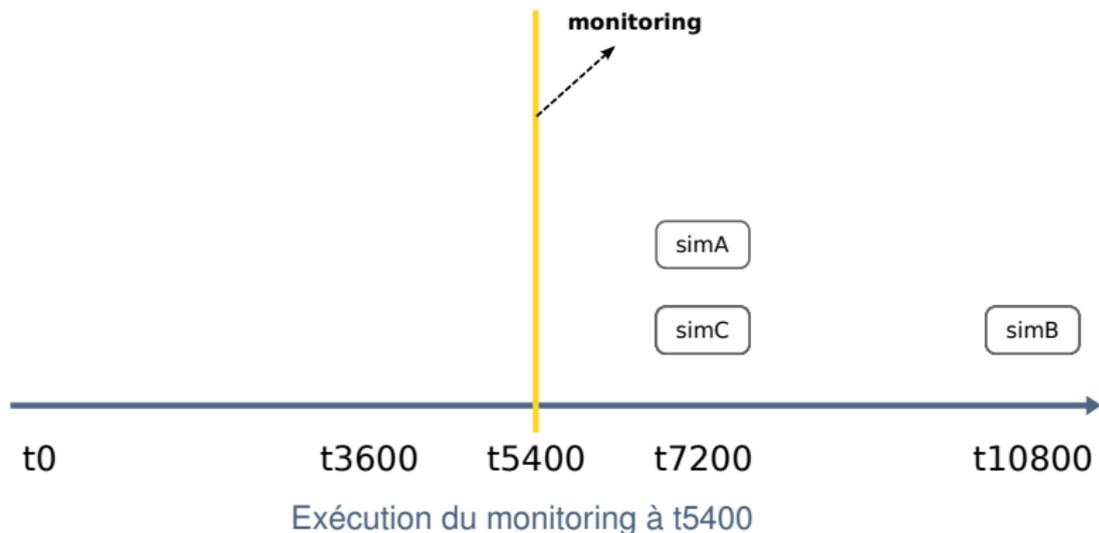
Exemple de simulation comprenant **3 simulateurs**,  
avec un deltat (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



Positionnement de simB pour sa prochaine exécution

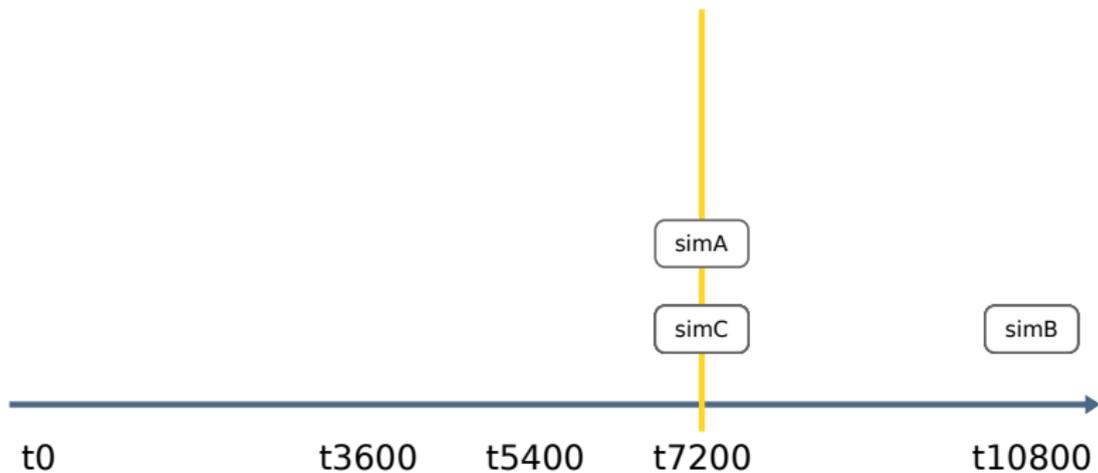
# Déroulement d'une simulation

Exemple de simulation comprenant 3 simulateurs, avec un deltat (dt) par défaut de 1h00 (3600s), sans contrainte de planification.



# Déroulement d'une simulation

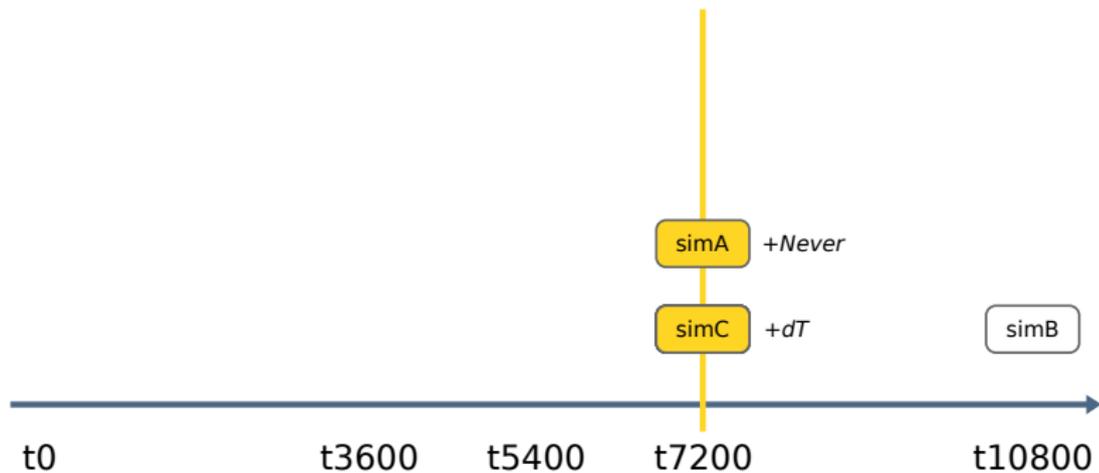
Exemple de simulation comprenant 3 simulateurs, avec un deltat (dt) par défaut de 1h00 (3600s), sans contrainte de planification.



Positionnement de la simulation au prochain index de temps ( $t_{7200}$ )

## Déroulement d'une simulation

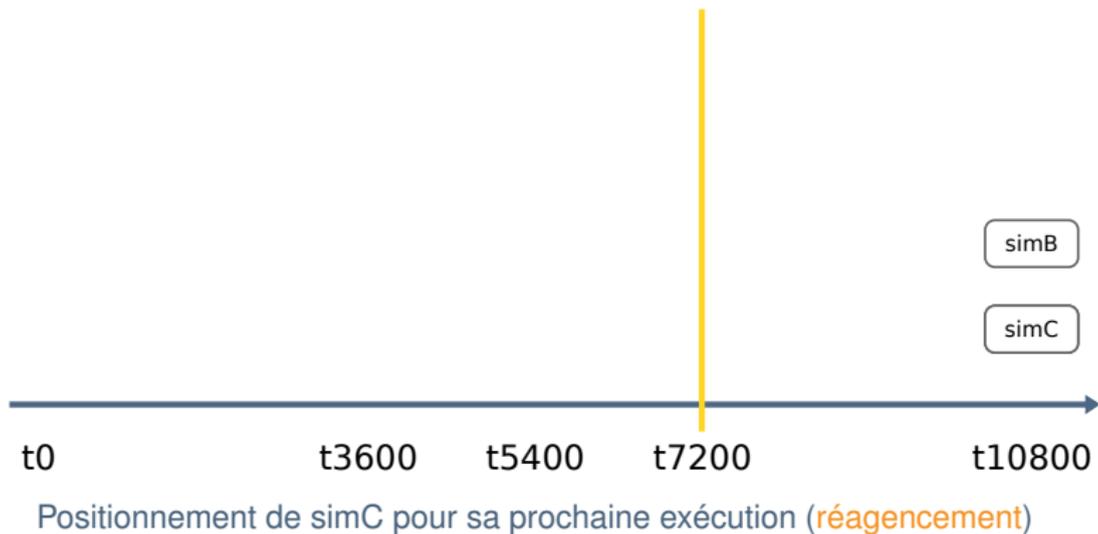
Exemple de simulation comprenant **3 simulateurs**,  
avec un **deltat (dt)** par défaut de **1h00 (3600s)**,  
sans contrainte de planification.



Exécution des simulateurs simA et simC à  $t_{7200}$

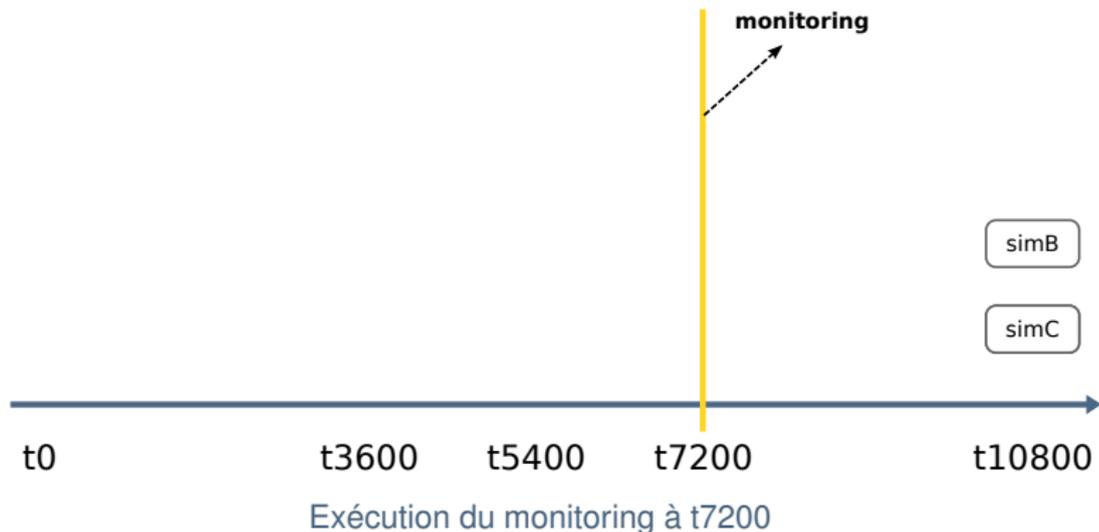
# Déroulement d'une simulation

Exemple de simulation comprenant 3 simulateurs, avec un  $\Delta t$  (dt) par défaut de 1h00 (3600s), sans contrainte de planification.



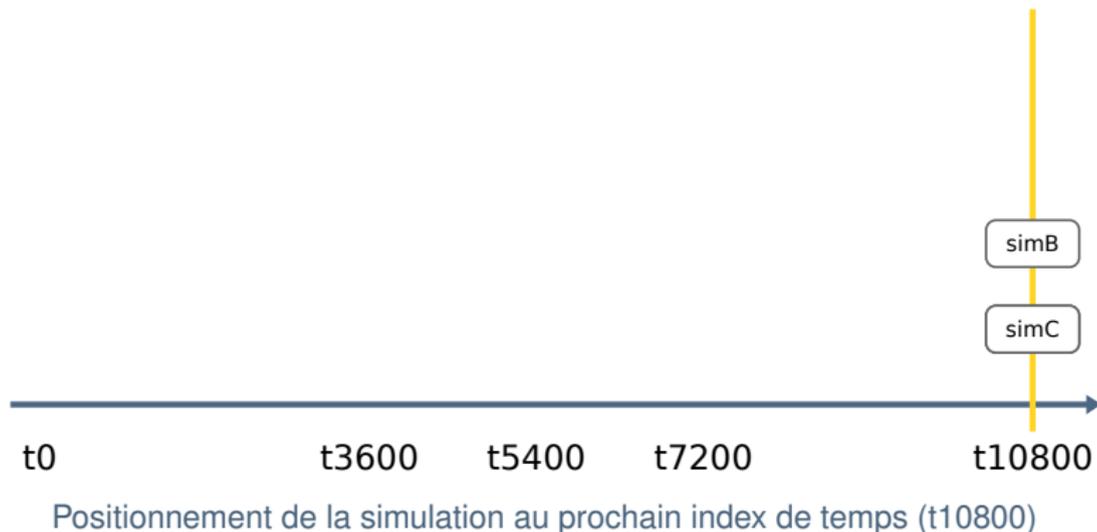
## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un  $\text{deltat (dt)}$  par défaut de **1h00** (3600s),  
sans contrainte de planification.



## Déroulement d'une simulation

Exemple de simulation comprenant **3 simulateurs**,  
avec un  $\text{deltat}$  (dt) par défaut de **1h00** (3600s),  
sans contrainte de planification.



# Générateurs de valeurs de variables

Les générateurs produisent des **valeurs pour des variables**  
Ils sont intégrés au framework OpenFLUID, et ne nécessitent pas de simulateurs particuliers

- Générateur de valeurs **constantes** (fixed)
- Générateur de valeurs **aléatoires** (random)
- Générateur de valeurs **à partir d'un fichier**, avec **distribution spatiale** du contenu (inject)
- Générateur de valeurs **à partir d'un fichier**, avec **interpolation temporelle** et **distribution spatiale** du contenu (interp)

# Cohérence du modèle couplé

Au travers de la signature de chaque simulateur, la **cohérence du modèle** couplé peut être vérifiée.

## Cohérence spatio-temporelle des **variables**

- Une variable requise par un simulateur sur une classe d'unité  $u$  doit être produite par un autre simulateur sur l'ensemble des unités spatiales de la classe  $u$
- Une même variable ne peut être produite que par un seul simulateur

## Cohérence spatiale des **attributs spatiaux**

- Un attribut requis sur une classe d'unité  $u$  doit être présent sur l'ensemble des unités spatiales de la classe  $u$

La cohérence du modèle couplé est **automatiquement vérifiée par le framework OpenFLUID** à la mise en place du modèle couplé

# Dossiers utilisés par OpenFLUID

## Localisation du dossier personnel OpenFLUID:

- Linux/Unix: `/home/<user>/.openfluid`
- Windows: `C:\Users\<user>\openfluid`

## Structure interne du dossier personnel OpenFLUID:

```
/examples ← exemples fournis
/network
openfluid.conf ← fichier de configuration OpenFLUID (utilisateur)
/wares
  /builderexts
  /observers
  /simulators ← simulateurs construits (utilisateur)
/workspace
  /projects ← projets (utilisateur)
  /waresdev
    /builderexts
    /observers
    /simulators ← codes sources des simulateurs (utilisateur)
```

# Structure d'un projet OpenFLUID

## Structure interne d'un projet OpenFLUID:

```
/IN ← jeu de données d'entrée  
  datastore.fluidx ← définition du datastore  
  domain.fluidx ← définition du domaine spatial  
  model.fluidx ← définition du modèle  
  monitoring.fluidx ← définition du monitoring  
  run.fluidx ← configuration d'exécution  
/OUT ← résultats de simulation  
openfluid-project.conf ← fichier de configuration du projet
```

- Le dossier IN peut comporter d'**autres fichiers et sous-dossiers** utilisés par des simulateurs ou observateurs
- Un même projet OpenFLUID peut **indifféremment** être exécuté en ligne de commande, dans l'interface graphique OpenFLUID-Builder, ou encore depuis R avec ROpenFLUID

# Fonctionnalités complémentaires

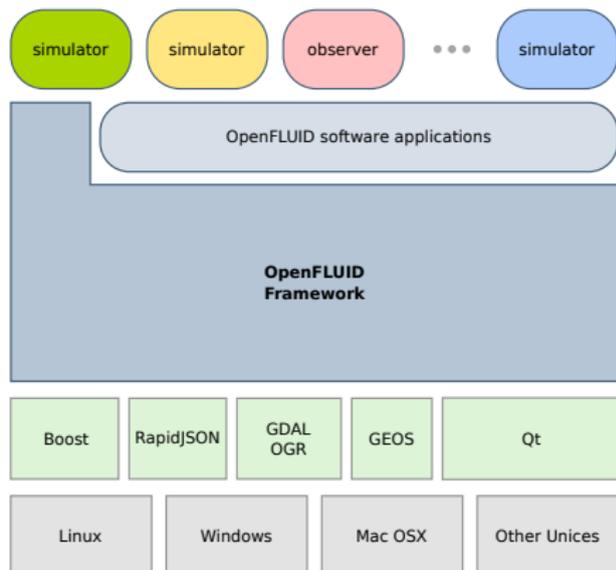
- Parallélisation automatisée des calculs des simulateurs  
(basée sur l'indépendance entre unités spatiales)
- Gestion des messages d'avertissements et d'erreurs
- Profilage de simulation
- Générateur de documentation scientifique à partir des simulateurs
- ...



# Architecture logicielle

## Framework et applications:

- Architecture **objet en C++**
- **API** complète
- Basé sur des bibliothèques de référence dans le domaine
- ~212000 lignes de codes +  
~43000 lignes de commentaires/doc
- Intégration continue
- Tests complets toutes les nuits
- Une release tous les 3 à 6 mois
- Disponible sous Linux, Mac et Windows



# Développements en cours et à venir

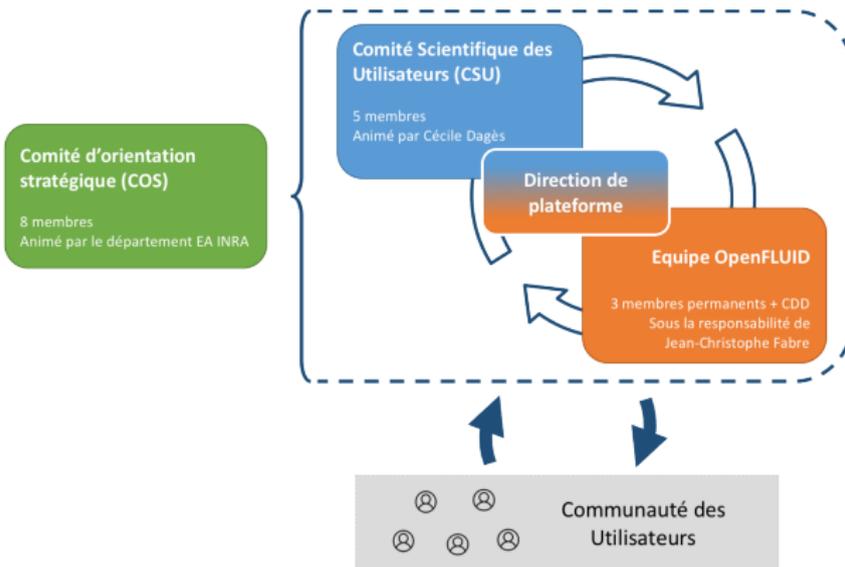
## A court terme (planifiés)

- Intégration native des données géographiques en entrée (import)
- Bibliothèque en ligne de fragment de codes sources de modèles
- Schémas de couplage spatio-temporels itératifs
- Protocole FluidHub pour la mise en place de web services autour d'OpenFLUID
- OpenFLUID dans le cloud (webservices interconnectés et stockage en ligne)
- Ajouts de nouveaux exemples livrés à l'installation
- ...

## A moyen et long terme :

- Attributs sur connexions spatiales, Process order multiples, ...

# Gouvernance



**Actif stratégique** de l'INRA depuis 2014  
soutenu par le département Environnement & Agronomie

Equipe et COS créés en 2015, CSU en 2018

# Equipe OpenFLUID



**David Crevoisier**

Modélisation numérique



**Jean-Christophe Fabre**

Gestion de projet  
Développement logiciel



**Armel Thöni**

Développement logiciel



+ Guest star

**Cécile Dagès**

Animatrice du comité des utilisateurs

# Pour citer OpenFLUID



J.-C. Fabre, M. Rabotin, D. Crevoisier, A. Libres, C. Dagès, R. Moussa, Ph. Lagacherie, D. Raclot, and M. Voltz.

OpenFLUID: an open-source software environment for modelling fluxes in landscapes.

In *Vol. 15, EGU2013-8821-1, EGU General Assembly 2013, Vienna (Austria), 7-12 avril 2013.*



J.C. Fabre, X. Louchart, R. Moussa, C. Dagès, F. Colin, M. Rabotin, D. Raclot, P. Lagacherie, and Voltz M.

OpenFLUID: a software environment for modelling fluxes in landscapes.

In *LANDMOD2010, INRA, CIRAD, page 13pp, Montpellier, France, 2010. Quae.*



M. Rabotin, J.-C. Fabre, A. Libres, P. Lagacherie, D. Crevoisier, and R. Moussa.

Using graph approach for managing connectivity in integrative landscape modelling.

In *Vol. 15, EGU2013-8851, EGU General Assembly 2013, Vienna (Austria), 7-12 avril 2013.*



# OpenFLUID sur les réseaux sociaux



@OpenFLUID

<https://twitter.com/OpenFLUID>



YouTube

<https://www.youtube.com/user/openfluidproject>



# Statut juridique

OpenFLUID est disponible sous **licence libre** ou **licence dédiée**

OpenFLUID est propriété de l'**INRA** et de **Montpellier SupAgro**

**OpenFLUID®** est une marque déposée pour la zone Europe

⇒ **Pas de restriction à un usage "fair" d'OpenFLUID en contexte académique/recherche**

⇒ De multiple possibilités dans le cadre de partenariat avec le privé (pour R&D, transfert, industrialisation de modèles, ...)

# Pour la formation

Version utilisée pour la formation :

OpenFLUID 2.1.8

A noter:

- version stable
- documentation disponible sur <https://community.openfluid-project.org/start/manuals/>
- supports de formation disponibles sur <https://community.openfluid-project.org/start/trainings/>