Thibault Hallouin
Hydro(geo)logical Modeller
BRGM (French Geological Survey)

unifhy

A community model for the terrestrial water cycle

evalhyd

A polyglot tool for the evaluation of streamflow predictions

# A community model for the terrestrial water cycle

## Overview of the Python package **unifhy**

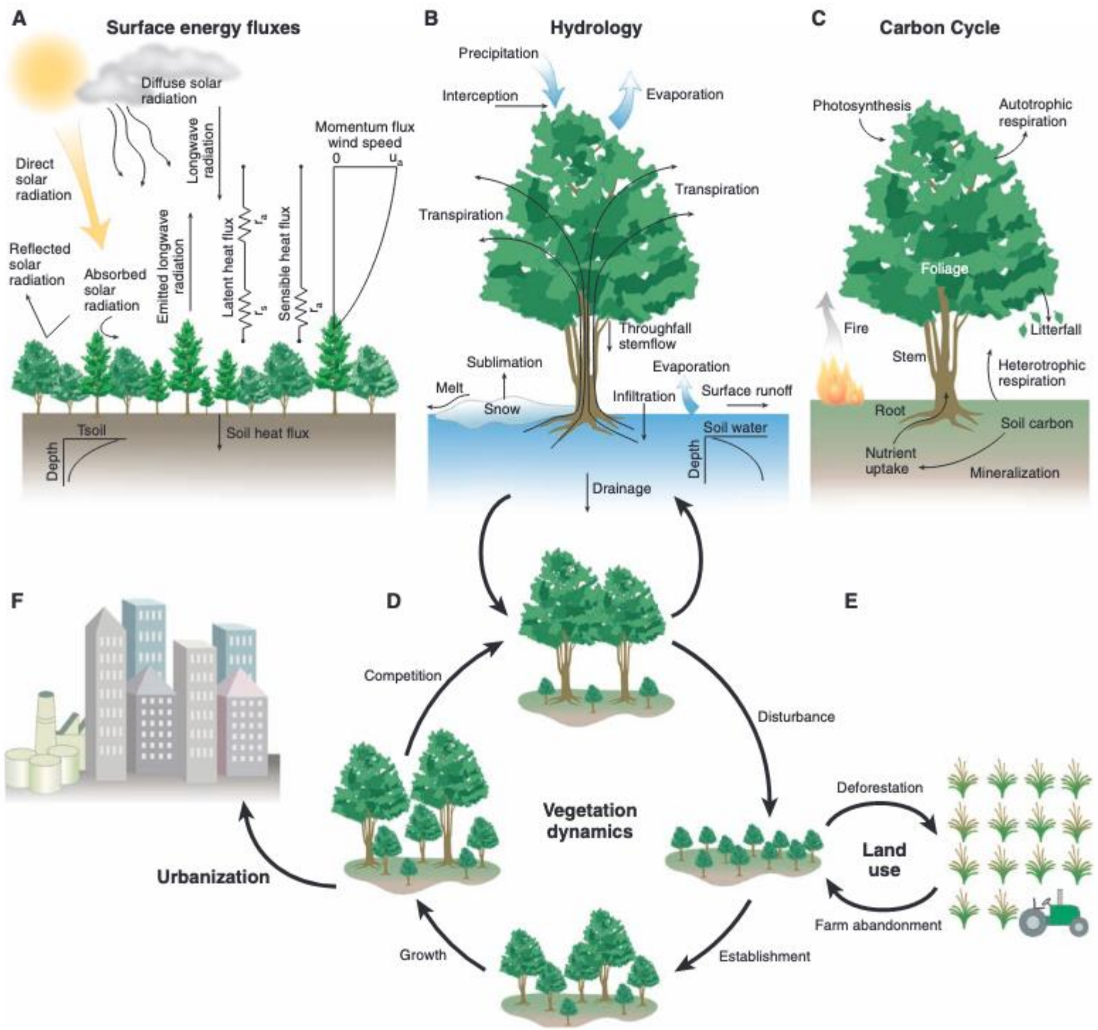
Thibault Hallouin and many others*

t.hallouin@brgm.fr

Hydro-JULES Online Course
9th July 2024

Hydro-JULES
UK Centre for Ecology & Hydrology
BGS British Geological Survey
National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

# What is a Land System Model (LSM)?




A Land System Model (LSM) simulates the biophysical processes involved in the exchanges of energy, water, and carbon with the Earth's atmosphere.

It considers the Earth's surface (i.e. vegetation, land water, land ice, land snow, urban fabric, etc.) as well as the Earth's subsurface (soil, moisture, etc.).

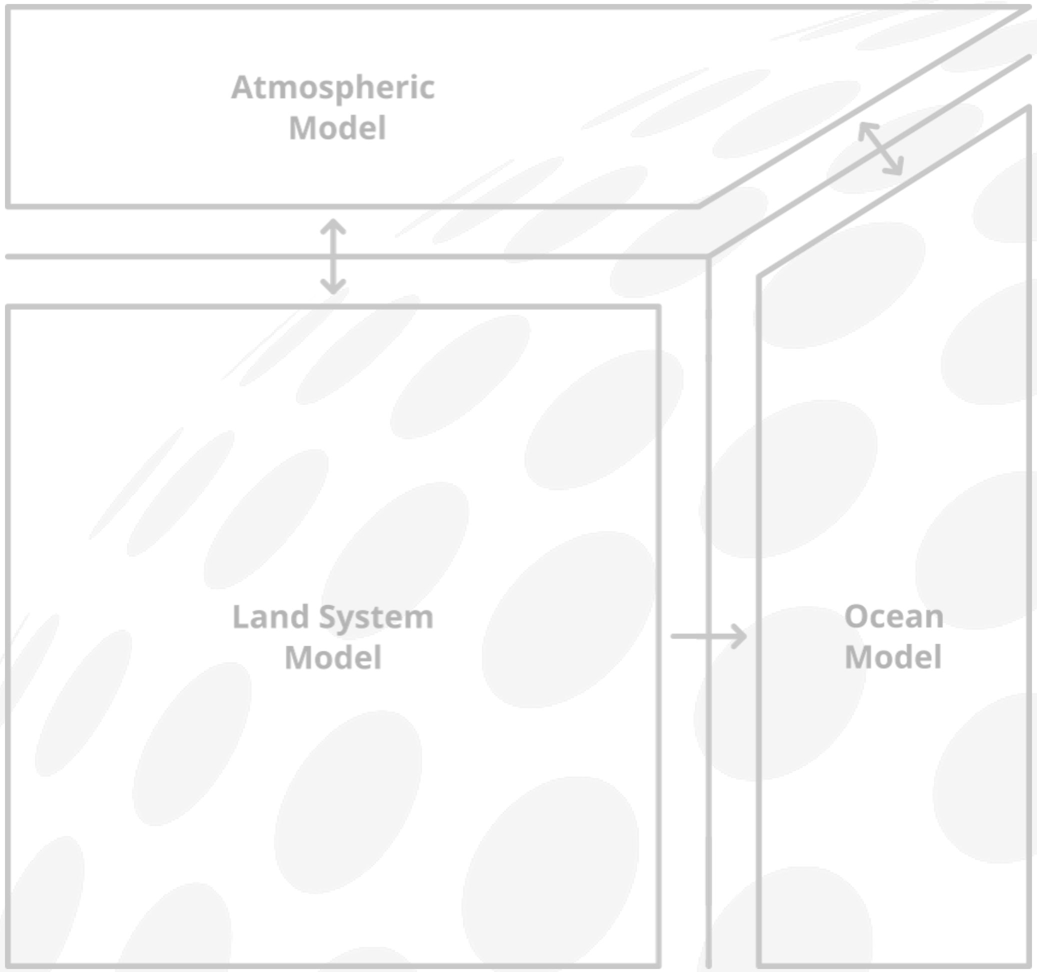
See figure below for an overview of the processes LSMs typical consider.

Bonan, G. B. (2008). *Forests and Climate Change: Forcings, Feedbacks, and the Climate Benefits of Forests*. Science, 320(5882), 1444–1449. https://doi.org/10.1126/science.1155121

# Current limitations in the modelling of hydrological processes in LSMs

Because Land System Models (LSM) were historically developed as a lower boundary condition to atmospheric models for Numerical Weather Predictions and Climate Modelling, they present limitations in the representation of the hydrological cycle:

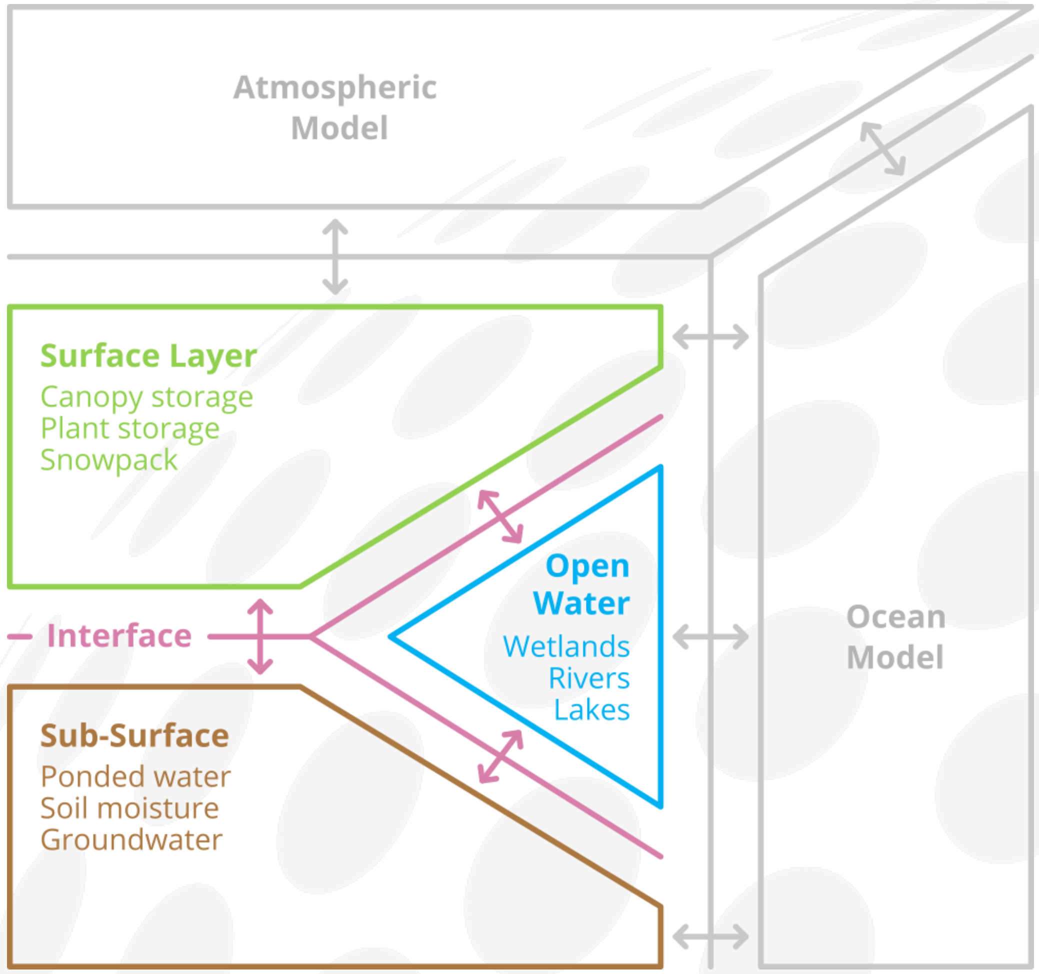
- inherited the resolution of the atmospheric grid
  - ↘ land heterogeneity (e.g. different vegetation types) within the atmospheric grid accommodated for by using implicit sub-grid approaches
- inherited a column-based approach (i.e. vertical exchanges)
  - ↘ absence of lateral hydrological connectivity between grid elements
- overlooked the two-way interaction between the land and the ocean
  - ↘ absence of ocean feedbacks such as tides, and storm surges



# Hydro-JULES contributions to these limitations

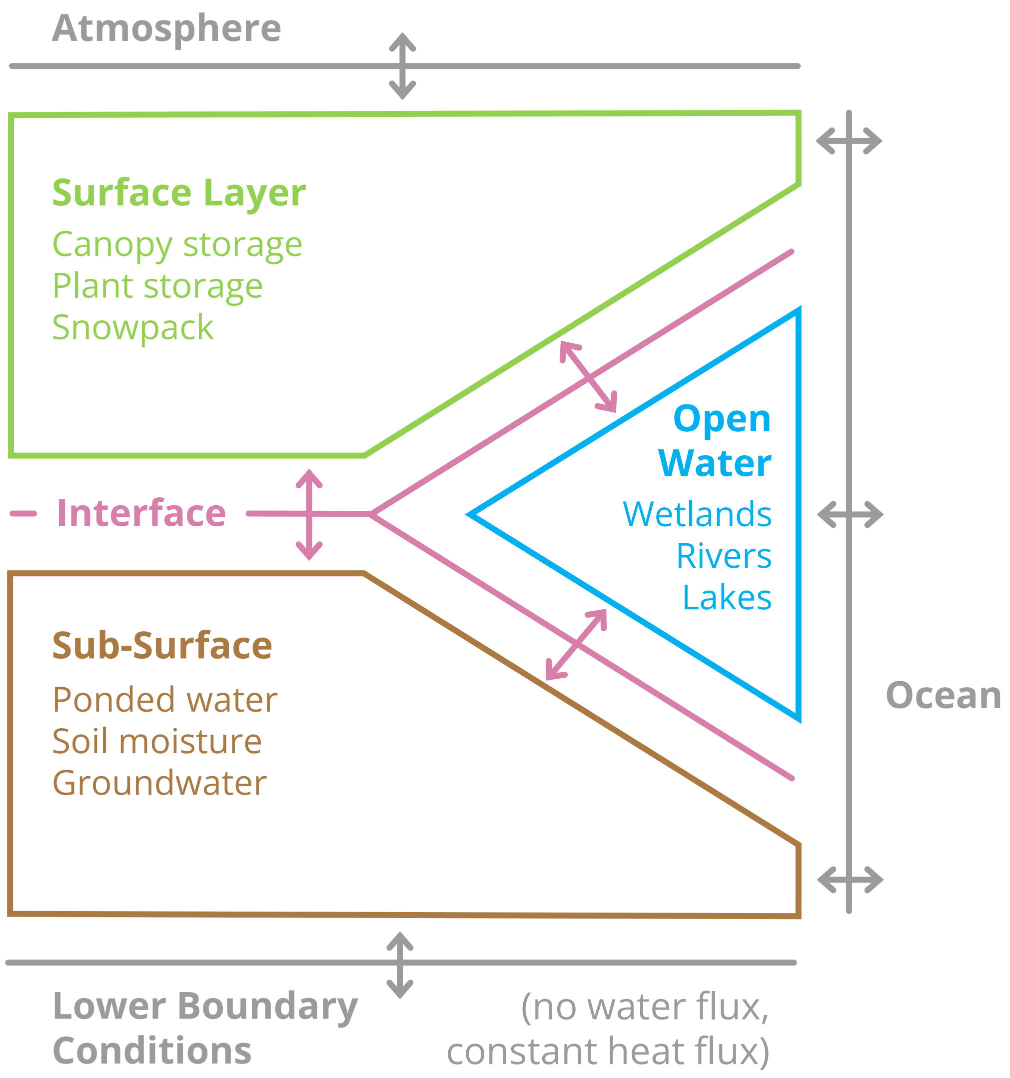
Hydro-JULES aims to tackle these limitations by developing:

- a new modelling framework for the terrestrial water cycle
  - ↘ a modular representation of the water cycle
  - ↘ interchangeable modules (referred to as components)
  - ↘ possible two-way communication with other models (atmosphere, ocean)
- a library of components, including:
  - ↘ a modular version of JULES
  - ↘ new groundwater models
  - ↘ CaMa-Flood



Conceptually,

# A modular representation of the terrestrial water cycle

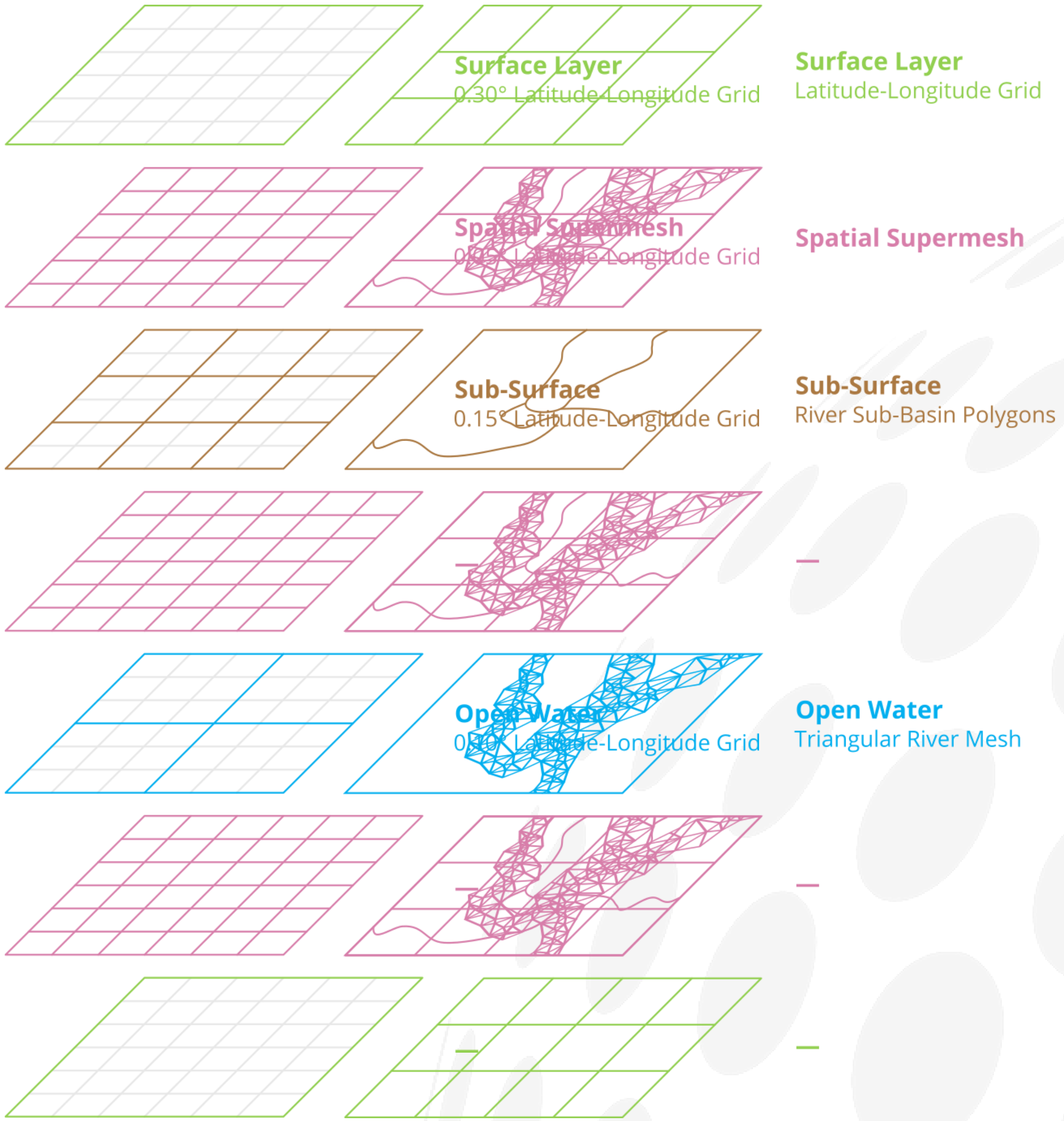



- subdivide the land system into components whose resolutions are adapted to the equations they aim to solve:
  - spatial resolution can be adapted to the dominant structures of heterogeneity
  - temporal resolution can be adapted to the timescale of the dominant processes
- for each component, all processes governing the energy, water and biogeochemical cycles are treated within a common numerical framework
- each scientific community (and their respective models and expertise) should map onto one or more components

# Flexibility in the resolutions of the components

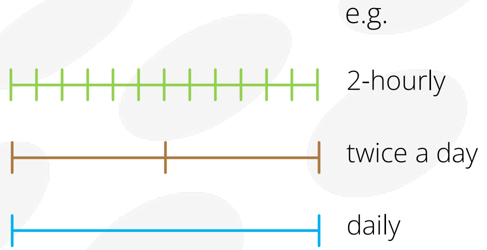
## Space

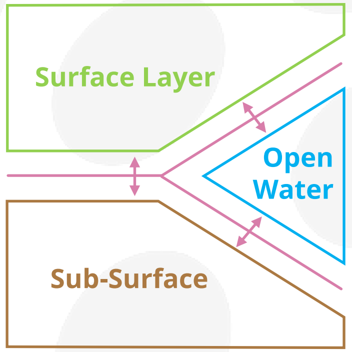
- each component can feature its own spatial discretization
- a spatial "supermesh" is determined from the resolution of the components to preserve the continuity equations
- the grid cells/polygons can communicate with their neighbours (i.e. lateral flow, not only vertical)




## Time

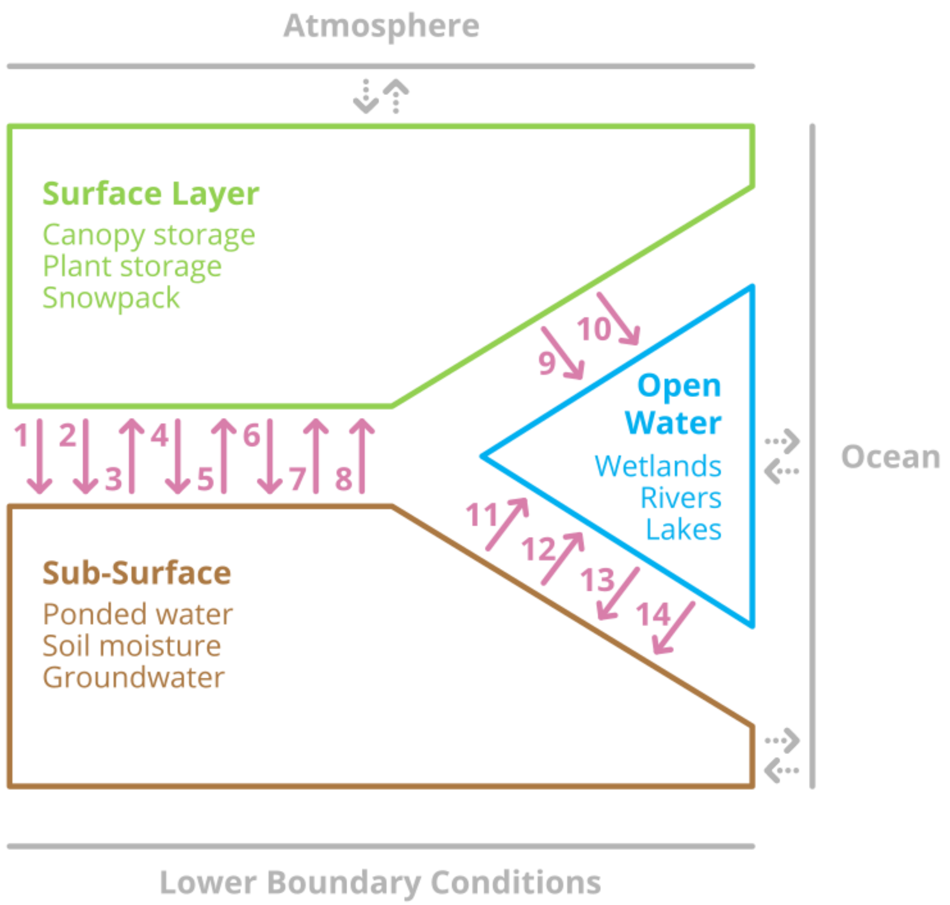
- each component can feature its own regular time-stepping scheme



# A fixed interface of transfers between components

*With an initial focus on the water cycle*




- each component must comply with a fixed interface (i.e. information to produce, information to incorporate)

**Transfers**

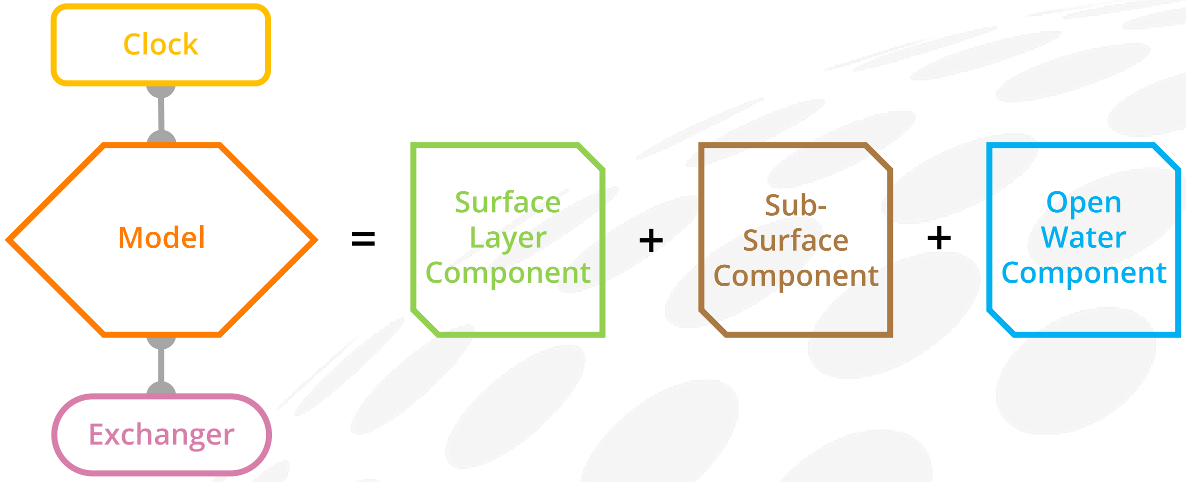
1 - Canopy liquid throughfall and snow melt flux
2 - Transpiration flux from root uptake
3 - Soil water stress for transpiration
4 - Direct water stress for direct soil evaporation
5 - Soil water stress for direct soil evaporation
6 - Water evaporation flux from standing water
7 - Standing water area fraction
8 - Total water area fraction
9 - Water evaporation flux from open water
10 - Direct throughfall flux
11 - Surface runoff flux delivered to rivers
12 - Net groundwater flux to rivers
13 - Open water area fraction
14 - Open water surface height

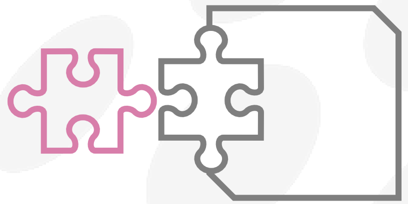
Technically,

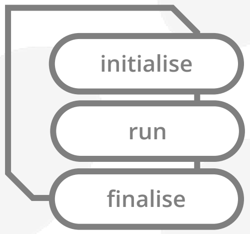
# A Python package as a first implementation of the framework blueprint

Core concepts of the Python implementation

- each **Component** is an object (of 3 different types: **SurfaceLayer**, **SubSurface**, **OpenWater**)
- A (Land System) **Model** object is allowing the communication between each **Component**, it namely:
    - features an **Exchanger** (responsible for the transfer of information and the potential remapping)
    - features a **Clock** (responsible for the time iteration)
- each **Component** must comply with the fixed interface of information going through the **Exchanger** (inward, outward)
- each **Component** must be implemented following the "initialise-run-finalise" paradigm



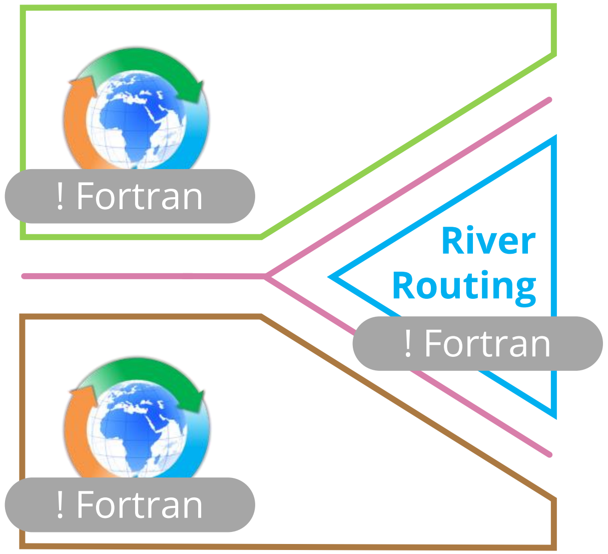


# Supporting Component's initialise-run-finalise to be in Python-C-Fortran

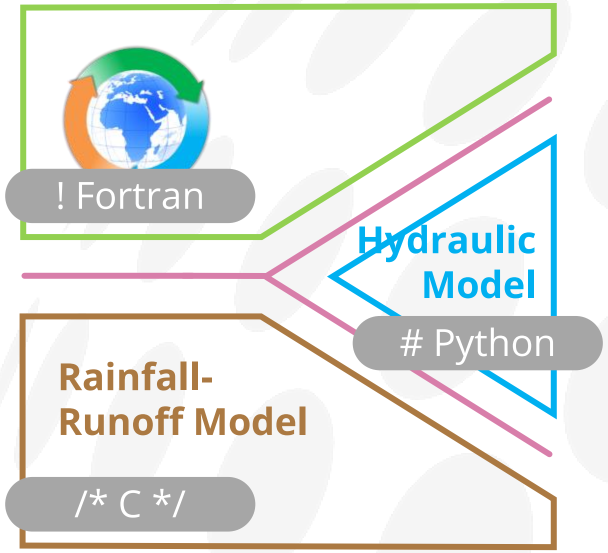
**Three languages are supported by the Python implementation:**

- Python (trivial)
- Fortran
- C/C++

**example 1**
running the framework with all the components from JULES




**example 2**
combining JULES with a rainfall-runoff model and a hydraulic model



# Summary

Hydro-JULES is developing:

- a new blueprint subdividing the terrestrial water cycle into components whose spatial and temporal scales are adapted to the equations of the dominant hydrological processes they are trying to solve
- a first implementation of this blueprint as a Python package (with possibility for Fortran and/or C extensions)
- the complete incorporation of the LSM model JULES in the framework as distinct components
- a framework that can communicate in a two-way fashion with atmospheric and ocean models
- a framework that supports and promotes the development and comparison of model components

github.com/unifhy-org

twitter.com/HydroJules

*Acknowledging and thanking the "many others":

Eleanor Blyth (UK CEH)
Doug Clark (UK CEH)
Simon Dadson (UK CEH)
Rich Ellis (UK CEH)
Andrew Hughes (BGS)
Bryan Lawrence (NCAS)
Grenville Lister (NCAS)
Jan Polcher (CNRS-LMD)

(by alphabetical order)

Hydro-JULES

UK Centre for Ecology & Hydrology

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

British Geological Survey

# A polyglot tool for the evaluation of deterministic and probabilistic streamflow predictions

Thibault Hallouin[1,*], François Bourgin[1], Charles Perrin[1], Maria-Helena Ramos[1], Vazken Andréassian[1]

[1]INRAE, HYCAR, France

[*]now at BRGM (French Geological Survey), France

t.hallouin@brgm.fr

With the financial support of

MINISTÈRES TRANSITION ÉCOLOGIQUE COHÉSION DES TERRITOIRES TRANSITION ÉNERGÉTIQUE MER

Liberté Égalité Fraternité

RÉPUBLIQUE FRANÇAISE

Liberté Égalité Fraternité

INRAE

# Background

## Current situation

- The action of « validation » or « verification » or « evaluation » is routinely done by hydrological practitioners
- The metrics used are largely the same *(e.g. MAE, NSE, KGE, Brier, CRPS)*
- Yet, the tools used to compute them are seldom the same *(e.g. EVS, xskillscore in Python, scoringRules in R)*, and do not feature the same metrics
- This leads to difficulties in reaching reproducibility in hydrological science *(because of discrepancies in metric computation, or different pre- and post-processing)*

## Proposed solution

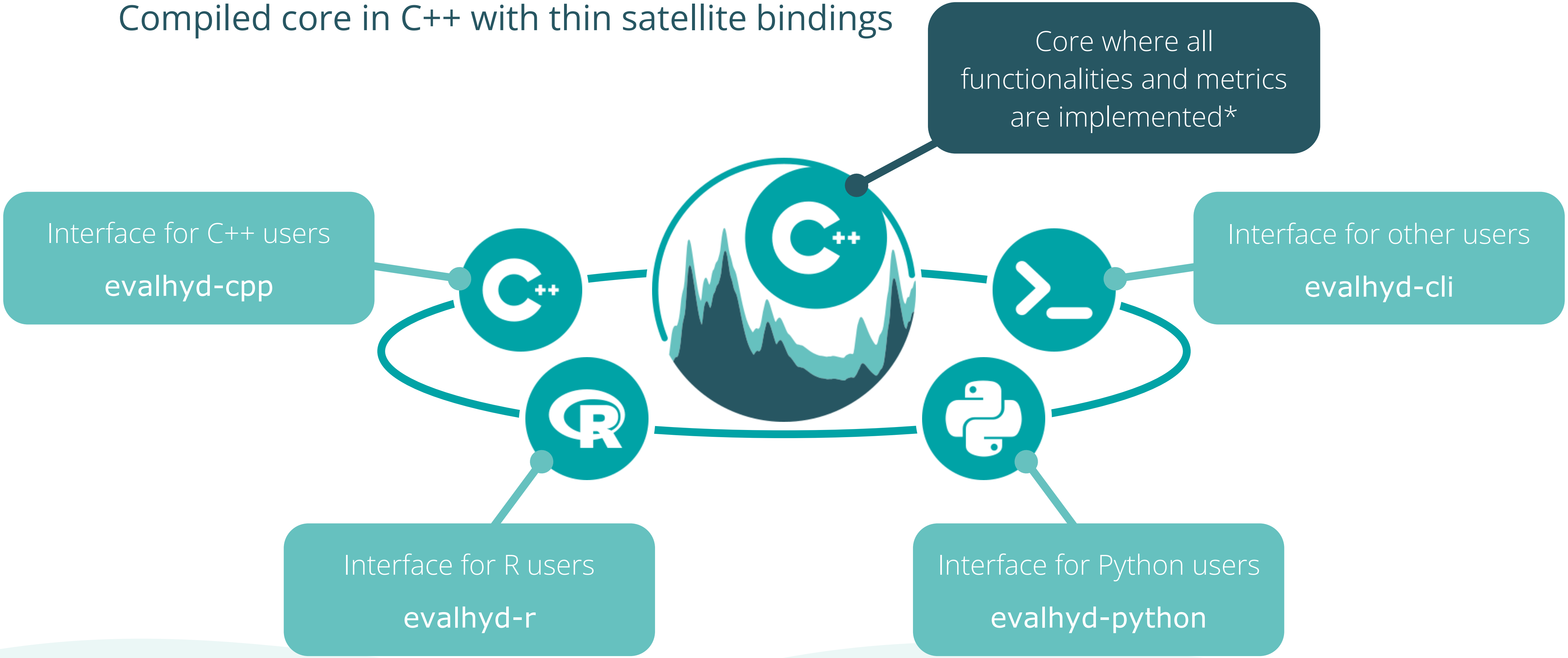
The development of new evaluation tool called **evalhyd** tailored to hydrologists that is:

- **Polyglot** i.e. usable in a variety of programming languages to be integrated in existing workflows and accessible to as many as possible
- **Efficient** i.e. relying on a compiled library to be applicable in a variety of tasks *(e.g. optimisation, analysis)*
- **Open** i.e. open access and open source for sharability and transparency purposes

evalhyd

# Design principles: efficient and polyglot

Compiled core in C++ with thin satellite bindings




*Thank you to Antoine Prouvost and Johan Mabille from QuantStack for the help with **xtensor**

# Design principles: simple two entry-point

Harmonised interfaces across the software stack

## Deterministic

```cpp
auto res = evalhyd::evald(obs, prd, {"NSE"});
```

```r
res ← evalhyd.evald(obs, prd, c("NSE"))
```

```python
res = evalhyd.evald(obs, prd, ["NSE"])
```

```shell
./evalhyd evald "./obs.csv" "./prd.csv" "NSE"
```

## Probabilistic

```cpp
auto res = evalhyd::evalp(
    obs, prd, {"BS"}, q_thr=thr, events="high"
);
```

```r
res ← evalhyd::evalp(
    obs, prd, c("BS"), q_thr=thr, events="high"
)
```

```python
res = evalhyd.evalp(
    obs, prd, ["BS"], q_thr=thr, events="high"
)
```

```shell
./evalhyd evalp \
    "./obs/" "./prd/" "BS" \
    --thr "./thr/" --events "high"
```

# Design principles: multi-dimensional

## Deterministic

- 2-dimensional philosophy:
  - Observations {1, time}
  - Predictions {series, time}
- To deal with multiple predictions, e.g. Monte Carlo experiments, or multi-model approaches

## Probabilistic

- 4-dimensional philosophy:
  - Observations {sites, time}
  - Predictions {sites, leadtimes, members, time}
- To process multiple sites and multiples lead times at the same time
- To make it possible to compute multivariate metrics (e.g. energy score)

# Available metrics

## Deterministic

| | |
|---|---|
| Mean absolute error | **MAE** |
| Mean absolute relative error | **MARE** |
| Mean square error | **MSE** |
| Root mean square error | **RMSE** |
| Nash-Sutcliffe efficiency | **NSE** |
| Kling-Gupta efficiency | **KGE**, **KGE_D**, **KGE_PRIME**, **KGE_PRIME_D**, **KGENP**, **KGENP_D** |
| Contingency table | **CONT_TBL** |

## Probabilistic

| | |
|---|---|
| Brier scores and their decompositions | **BS**, **BSS**, **BS_CRD**, **BS_LBD** |
| Reliability diagram | **REL_DIAG** |
| Continuous rank probability score | **CRPS_FROM_BS**, **CRPS_FROM_QS**, **CRPS_FROM_ECDF** |
| Contingency table | **POD**, **POFD**, **CSI**, **FAR**, **ROCSS** |
| Quantile scores | **QS** |
| Rank scores | **RANK_HIST**, **DS**, **AS** |
| Interval scores | **CR**, **AW**, **AWN**, **WS** |

# Functionalities

## Masking

Perform temporal and conditional subsets to focus on sub-periods of the whole record.

*e.g. "consider periods where streamflow observations are greater than their median" or "consider period from 21st time step to 53rd time step"*

## Memoisation

Avoid recomputing the same variable or the same time step twice to save time and memory.

*e.g. NSE and KGE both require to compute the quadratic error between the observations and their arithmetic mean*

## Bootstrapping

Estimate the sampling uncertainty using a non-overlapping block bootstrapping method.

*e.g. "draw N samples of L hydrological years (blocks) and summarise sampled distribution as percentiles"*

## Handling of missing data

Eliminate (i.e. pairwise delete) time steps with missing data flagged as "not a number".

*e.g. to ignore time steps with missing observations or to ignore time steps with no predictions*

# Thank you

https://hydrogr.github.io/evalhyd

thibault.hallouin@inrae.fr

Geosci. Model Dev., 15, 9177–9196, 2022
https://doi.org/10.5194/gmd-15-9177-2022


Geoscientific Model Development
Open Access

# UniFHy v0.1.1: a community modelling framework for the terrestrial water cycle in Python

**Thibault Hallouin**[1,2,a], **Richard J. Ellis**[3], **Douglas B. Clark**[3], **Simon J. Dadson**[3,4], **Andrew G. Hughes**[5], **Bryan N. Lawrence**[1,2,6], **Grenville M. S. Lister**[1,2], **and Jan Polcher**[7]

[1]National Centre for Atmospheric Science, Reading, UK
[2]Department of Meteorology, University of Reading, Reading, UK
[3]UK Centre for Ecology & Hydrology, Wallingford, UK
[4]School of Geography and the Environment, University of Oxford, Oxford, UK
[5]British Geological Survey, Keyworth, UK
[6]Department of Computer Science, University of Reading, Reading, UK
[7]Laboratoire de Météorologie Dynamique, IPSL, CNRS, Palaiseau, France
[a]now at: HYCAR Research Unit, INRAE, Université Paris-Saclay, Antony, France

**Correspondence:** Thibault Hallouin (th.thibault.hallouin@gmail.com)

Received: 15 December 2021 – Discussion started: 16 December 2021
Revised: 3 October 2022 – Accepted: 6 December 2022 – Published: 23 December 2022

**DOI:** 10.5194/gmd-15-9177-2022

unifhy

Geosci. Model Dev., 17, 4561–4578, 2024
https://doi.org/10.5194/gmd-17-4561-2024


Geoscientific Model Development
Open Access

# EvalHyd v0.1.2: a polyglot tool for the evaluation of deterministic and probabilistic streamflow predictions

**Thibault Hallouin**[1,a], **François Bourgin**[1], **Charles Perrin**[1], **Maria-Helena Ramos**[1], **and Vazken Andréassian**[1]

[1]Université Paris-Saclay, INRAE, HYCAR, Antony, France
[a]now at: BRGM (French Geological Survey), Water Resources Unit, Orléans, France

**Correspondence:** Thibault Hallouin (thibault.hallouin@inrae.fr)

Received: 28 June 2023 – Discussion started: 29 June 2023
Revised: 25 April 2024 – Accepted: 27 April 2024 – Published: 10 June 2024

**DOI:** 10.5194/gmd-17-4561-2024

evalhyd