

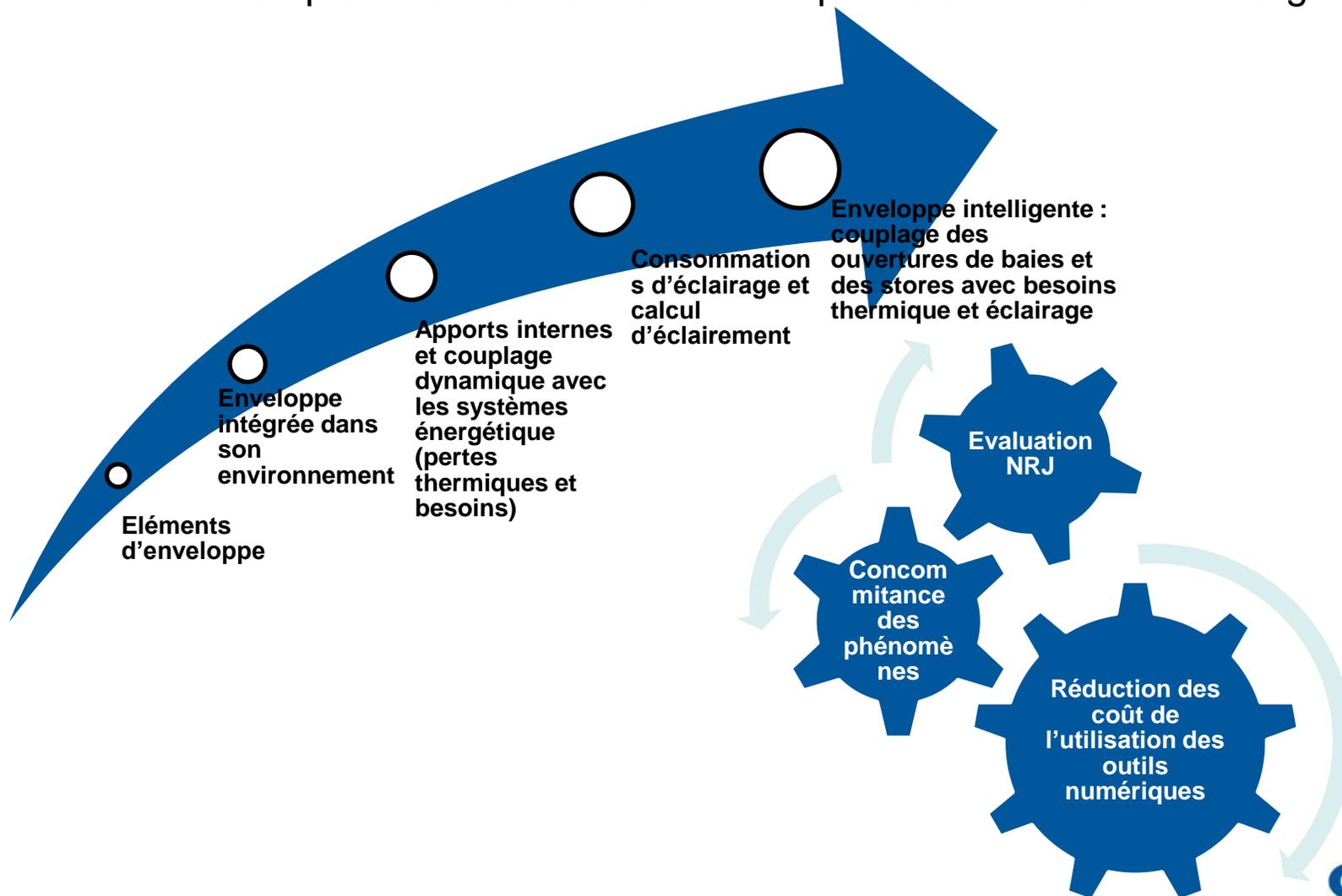


Interopérabilité d'outils métier

Benjamin Haas

En route vers des outils intégrés et évolutifs multi-métier

Exemple : De la simulation thermique vers la simulation énergétique





Aujourd'hui ...

Mise à disposition d'outils de plus en plus complexe à l'intérieur,
de plus en plus simple d'utilisation

Intégrabilité interactive dans les logiciels du commerce

Outils enrichissable sans rentrer dans le câblage, sans connaissance du code
source

Couplage avec des bases de données reconnues

... et demain



Outils
éclairage,
etc ...

Outils
acoustiques

Outils
thermico/NRJ

Outils
environnementaux

elodie

Outils multi-
métier

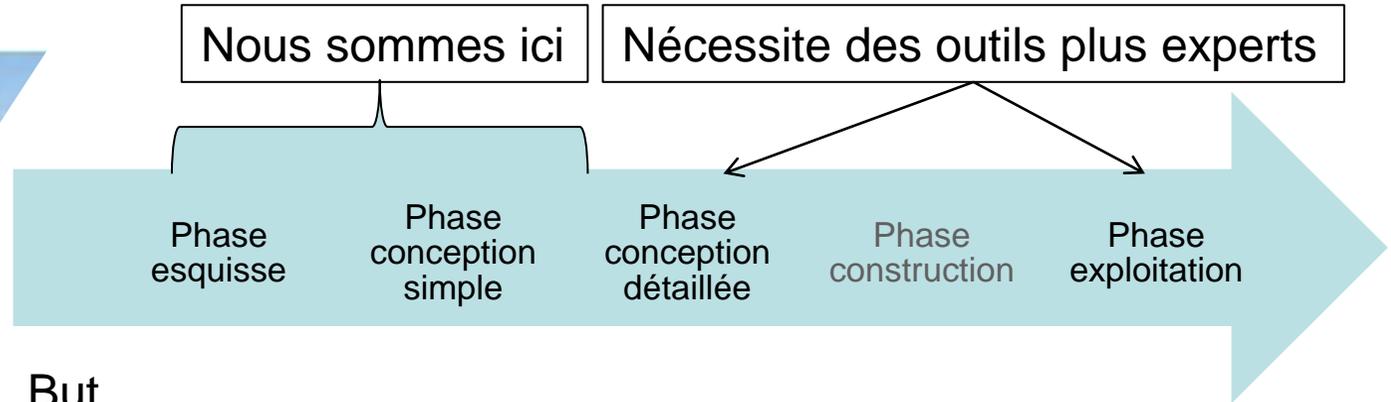
CSTB

Phanie

CSTB

Ometh

Pour qui ? Pour quoi ?



But

1. Object de confort
2. Objectif de performances/impacts
3. Objectif de coût

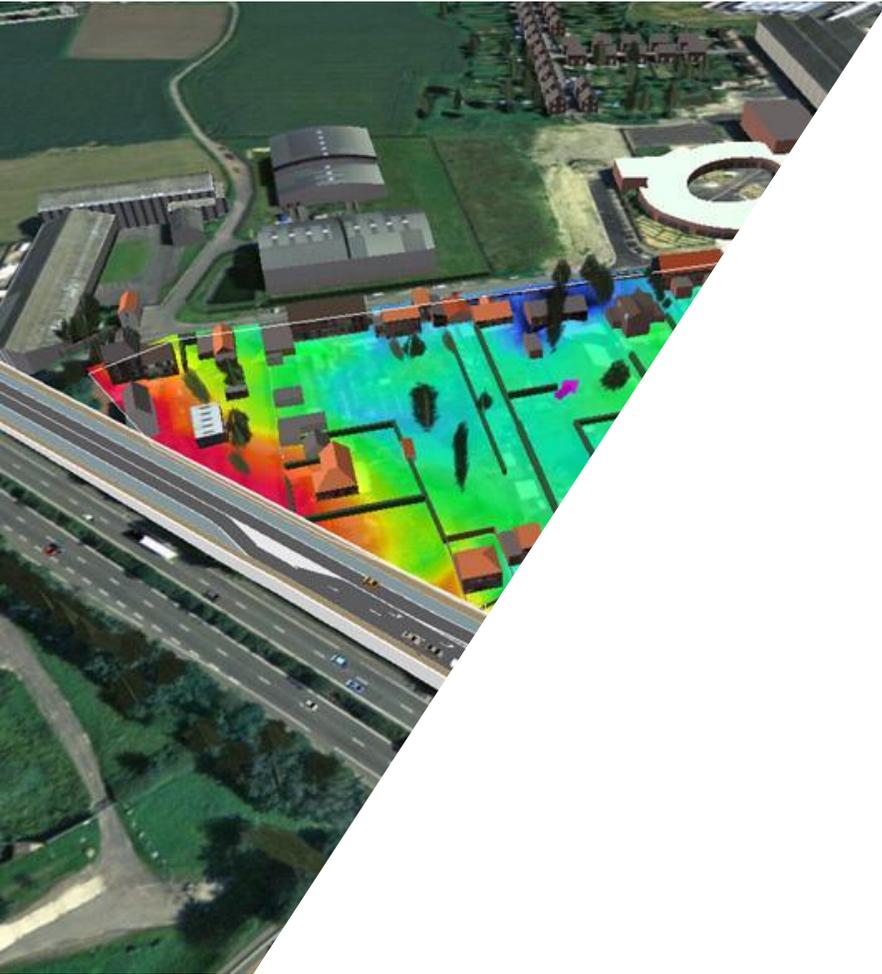
Quels leviers?

1. Dimensionnement de l'enveloppe
2. Choix des matériaux
3. Dimensionnement et choix des systèmes
4. ...

Quels utilisateurs ?

1. Les bureaux d'étude
2. Les industriels

Sommaire



1. **L'interopérabilité des données utilisateurs**
2. **L'interopérabilité dynamique des différents métiers**
3. **La versatilité des modules**
4. **(non traité) La connexion avec la maquette numérique**
5. **(non traité) La connexion avec des bases de données produits**

1. L'interopérabilité des données utilisateurs



Objectifs :

1. Le moins de ressaisies possible entre métier
2. Garantie de cohérence



SIMBIO Prototype Tool
Outil pédagogique

Résultats:

- Une première structure de donnée multi-métier dédiée bâti
- Une IHM graphique permettant de sélectionner tout ou partie des métiers
- Les données redondantes entre métier sont automatiquement recalculées
- Les 4 moteurs sont lancés au travers de 4 Web Service

1. L'interopérabilité des données utilisateurs : SIMBIO

Principal verrou : les différentes échelles

Nom de baie	Type ouvrant	U horiz [W/m²K]	Sw (-)	TI (-)	Vert N	Vert E	Vert S	Vert O	45° N	45° E	45° S	45° O	Horiz
Baie 1	coulisse	3	0,3	0,8	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	avec P.M.	6,6	6,3	0,23	0,6	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	sans P.M.	8	5,27	0,5	0,5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	avec P.M.	4,01	4,12	0,6	0,583	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Points d'amélioration:

- Ergonomie de l'interface
- Ajout des systèmes
- Compatibilité format de données existant
- Connexion au BIM et aux bases de données

2. Interopérabilité dynamique entre les différents métier



Objectifs 2013/2014:

Faire communiquer les moteurs spécifique métier entre chaque itération

Principaux verrous :

a) Redondances de modélisations

- a) Occupation
- b) Climat
- c) Gestion/régulation
- d) ...

b) Difficultés informatiques

- a) Cœur de calcul pilotables VS. Logiciels
- b) Langages (moins de mesure)

3. Versatilité des modules : l'approche par extensions dynamiques



Objectifs :

Permettre des développements de modules tiers, Compatibles, facilement branchables, éventuellement boîte noire (garantie de la PI)

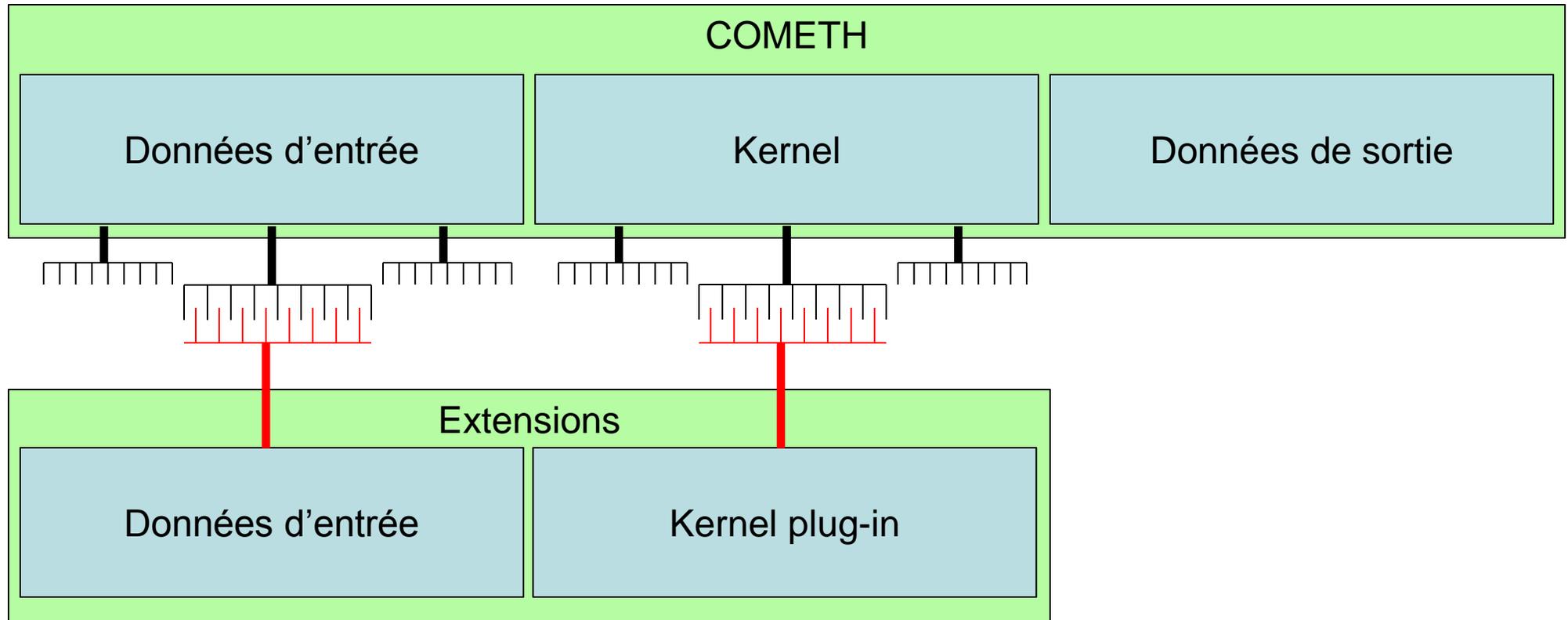
Principaux verrous :

Plus aucun en modélisation causale : en production pour la procédure Titre V

Comment faire ?:

- Définition d'une ontologie de « systèmes » par des Interfaces de variables physiques échanges
- Couplage faible plutôt que fort

3. Versatilité des modules : l'approche extensions dynamiques



3. Versatilité des modules : l'approche plug-in



Point de vue du développeur d'extension:

- Inutile de connaître de s'appropriier le code source du moteur
- L'interface avec le moteur est définie de manière univoque
- Le moteur s'occupe lui-même de faire le branchement du plug-in
- Responsable de la maintenance de son module

Point de vue du développeur du moteur:

- Intégrité du code source du moteur garantie
- Possibilité de livrer ou pas le plug-in
- Création d'une communauté

Point de vue de l'utilisateur du logiciel:

- Totalement transparent : le plug-in apparaît comme un ajout au jeu de données

3. Versatilité des modules : l'approche plug-in



Solution technique:

- Environnement .Net
- Compatible *sans recompilation* Mac/Linux/Windows
- Interface en code objet C#
- Interfaçage logiciel garantie en
 - C/C++
 - Langages COM Windows
 - Python
 - JAVA ?

Aujourd'hui, plusieurs extensions dynamiques en production chez les clients



Les mises en application actuelle

Les outils basés sur COMETH, donc la RT2012 et la procédure Titre V pour une application réglementaire

- **Simulation énergétique intégrée**
- **Plug-in de modules (appliqués aux Titre V)**
- **Plug-out limité**

Les outils de validation du corpus de normes européennes

- **Vers une simulation énergétique intégrée**
- **Architecture totalement plug-in en phase avec les normes**

Les projets SIMBIO et ELODIE2.0 + HOLISTEEC/COSIMPFI(?) et MAEVIA

- **Vers le multi-métier**



Conclusion

Le temps de la recherche

- **Définition des interfaces de données échangées**
- **Travail sur les jeux de données**
- **Réponses techniques informatiques le plus neutre possible**

Le temps de la mise en production

- **Mise en place d'une communauté**
- **Montée en compétence de la communauté**
- **Ajustements inévitables suite aux retours de la communauté**

Quand l'outil structure le développement scientifique

Conclusion 2

Même interopérable, même multi-métier, les hypothèses nécessaires à la mise en place des plugins restreignent le périmètre de l'outil

C'est le prix à payer pour la facilité de développement de modules tiers

Les outils visés ici valident les grandes orientations des projets. Les projets à la plus forte ajoutée devront s'appuyer sur des outils plus évolués

La facilité d'utilisation viendra en grande partie de la mise à disposition des utilisateurs des données (bases de données et maquette numérique)

