COLLEGE MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Laboratoire des Sciences du Numérique de Nantes

Évaluation de la performance basée sur les langages exécutables spécifiques au domaine

Soutenance de thèse de doctorat de Nantes Université par

Hiba AJABRI

le 30 avril 2026

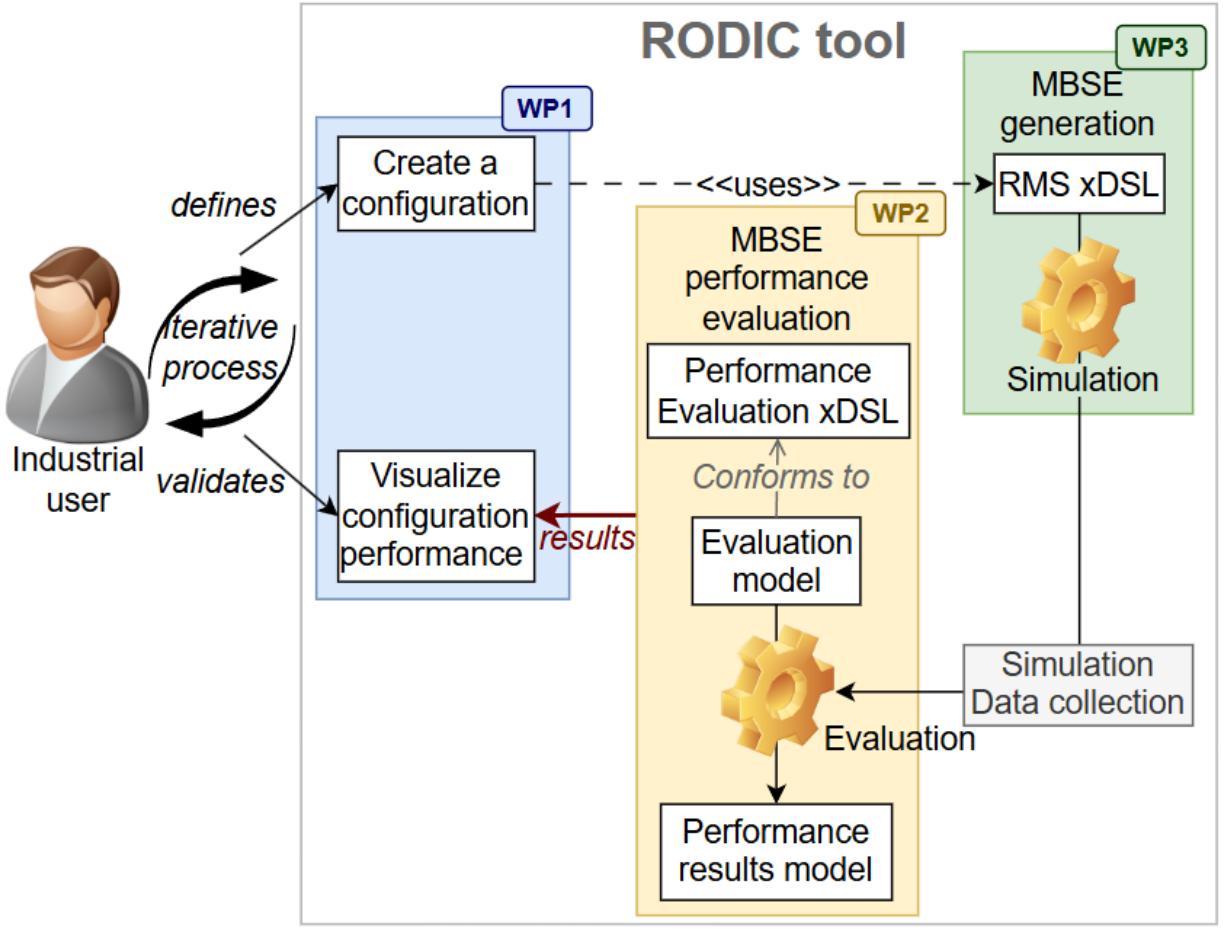
- **Sous la direction de** : Pr. Christian ATTIOGBE, Pr. Pascal BERRUET, Dr. Jean-Marie MOTTU
- **Rapporteures** : Pr. Agnès FRONT, Pr. Christelle URTADO
- **Examinateur** : Pr. Julien DEANTONI **Invité** : Dr. Florent De Lamotte

Contexte – Projet RODIC (2022-2026)

- Projet ANR multidisciplinaire impliquant des compétences en ingénierie industrielle et en ingénierie logicielle.

- Permettre aux utilisateurs industriels de :

  - **Modéliser** et **simuler** des systèmes de production reconfigurables (RMS).
  - **Évaluer la performance de RMS.**
  - Aider à la **prise de décision** pendant la **phase de conception,** par simulation avant déploiement.



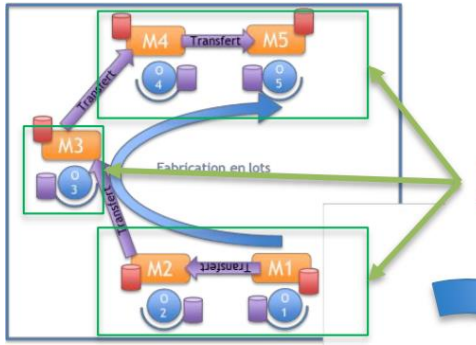
Systèmes de fabrication reconfigurables (RMS)

Lampex système

↓ produit

Lampe portable à dynamo

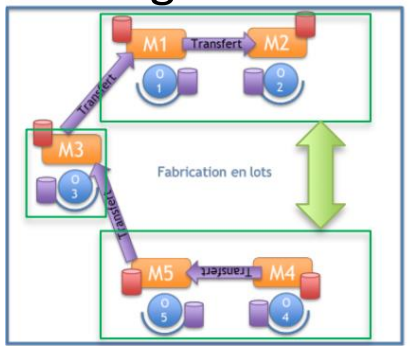
Configuration 1



Modules

Changement de conditions
de production
Arrêt de la production
Modification de la configuration
Reprise de la configuration

Configuration 2



La question de RODIC est comment choisir une bonne configuration cible ?

Introduction : comment permettre l'évaluation précoce de la performance ?

L'évaluation précoce de la performance nécessite :

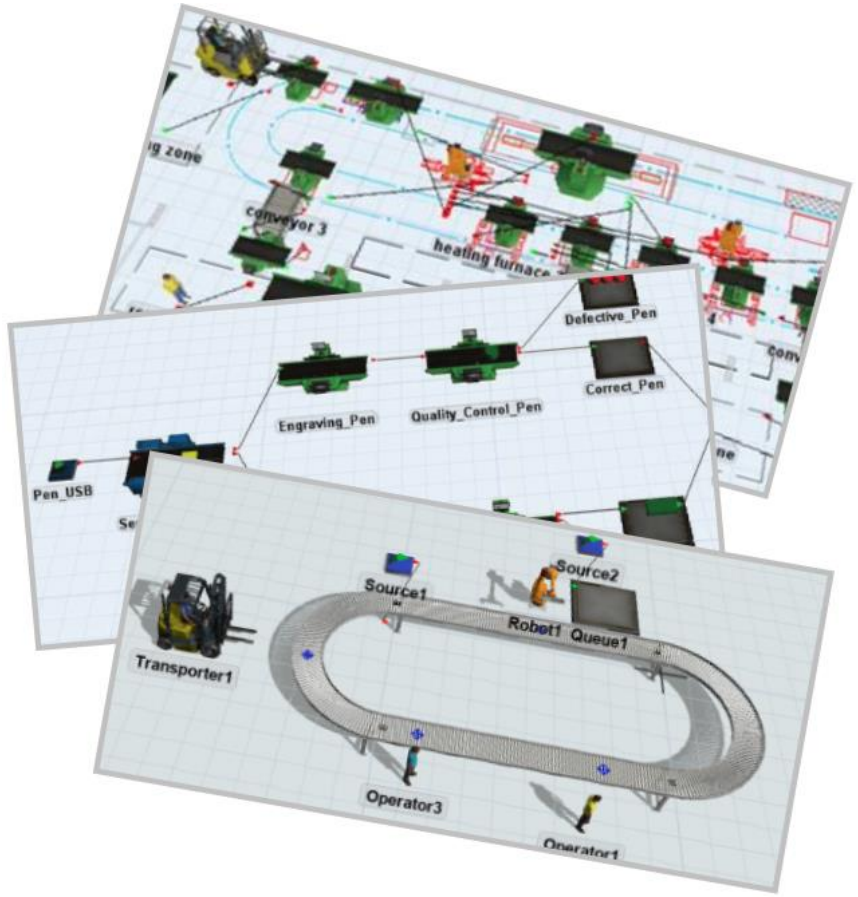
- La **modélisation** du comportement des systèmes.
- Leur **simulation** avant la mise en œuvre réelle.

➔ **FlexSim** existe, mais :

- Complexe,
- Réservé aux experts FlexSim,
- Utilisation chronophage.

➔ Pour répondre à cette problématique :

Des moyens / mécanismes offerts par les langages exécutables spécifiques au domaine.



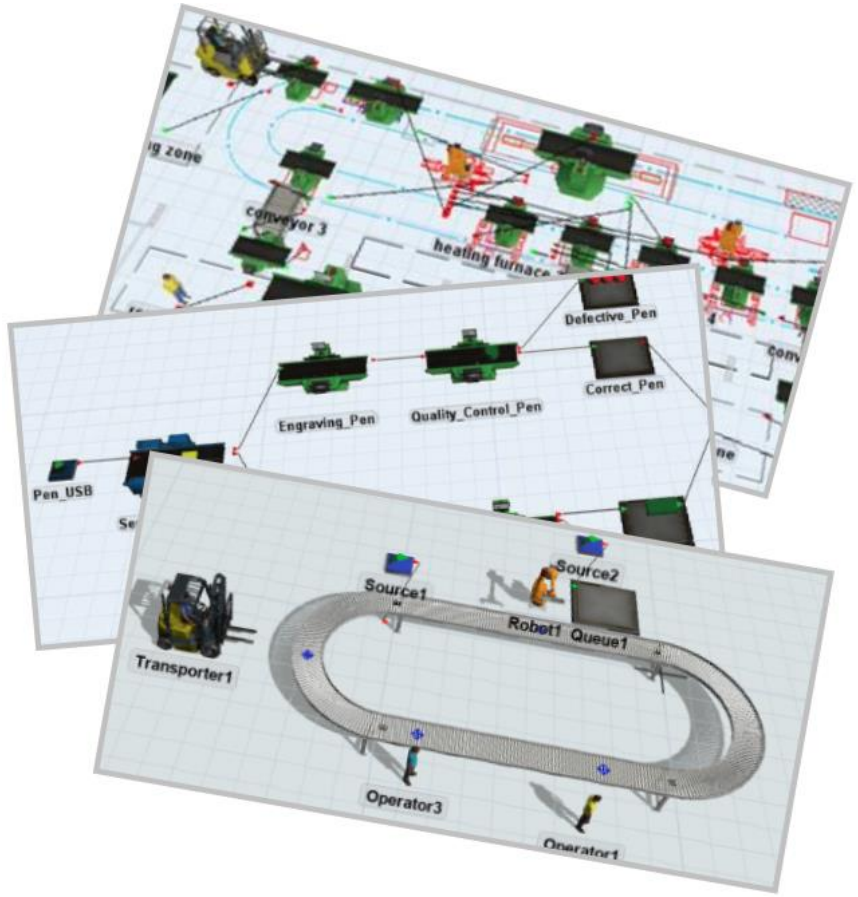
Introduction : comment permettre l'évaluation précoce de la performance ?

L'évaluation précoce de la performance nécessite :

- La **modélisation** du comportement des systèmes.
- Leur **simulation** avant la mise en œuvre réelle.

➔ **FlexSim** existe, mais :

- Complexe,
- Réservé aux experts FlexSim,
- Utilisation chronophage.



➔ Pour répondre à cette problématique :

Des moyens / mécanismes offerts par les langages exécutables spécifiques au domaine.

Introduction : comment permettre l'évaluation précoce de la performance ?

L'évaluation précoce de la performance nécessite :

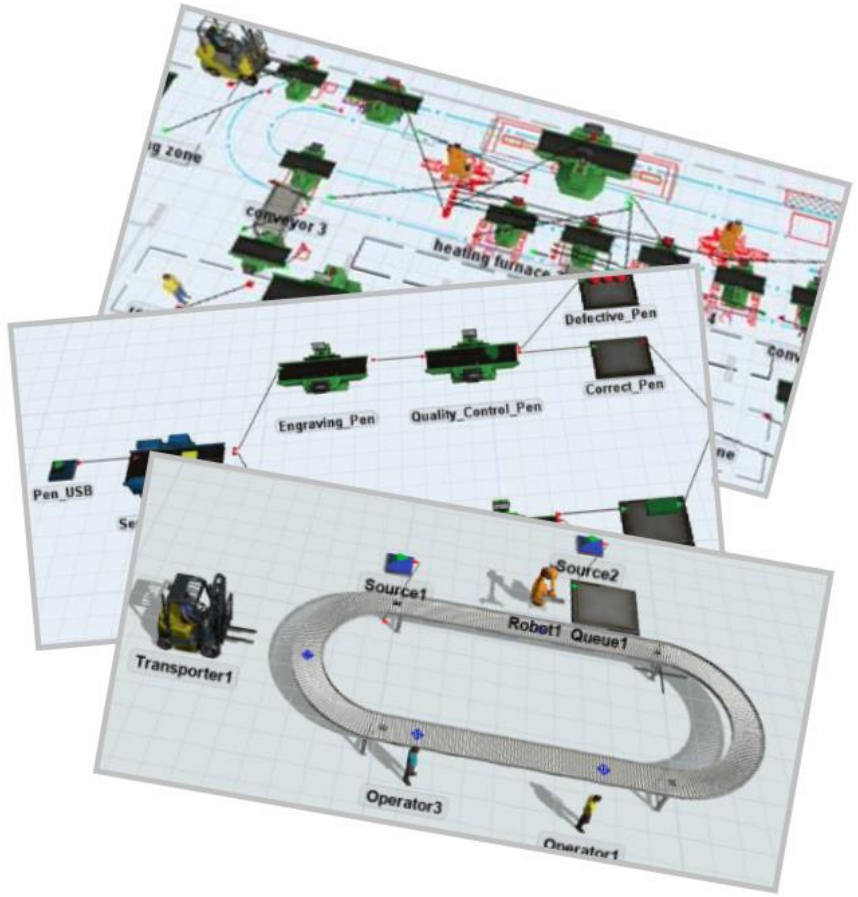
- La **modélisation** du comportement des systèmes.
- Leur **simulation** avant la mise en œuvre réelle.

➔ **FlexSim** existe, mais :

- Complexe,
- Réservé aux experts FlexSim,
- Utilisation chronophage.

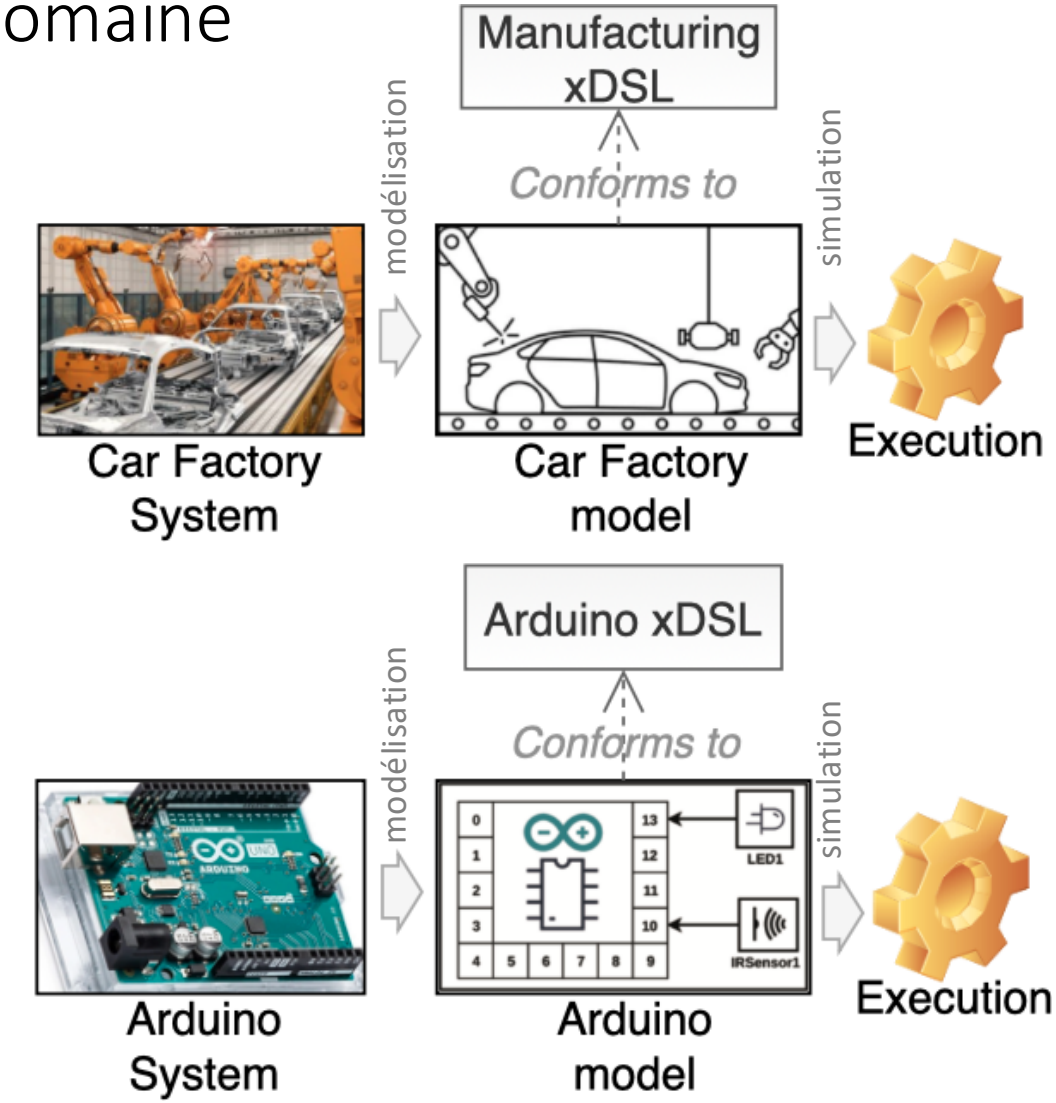
➔ Pour répondre à cette problématique :

Des moyens / mécanismes offerts par les langages exécutables spécifiques au domaine.

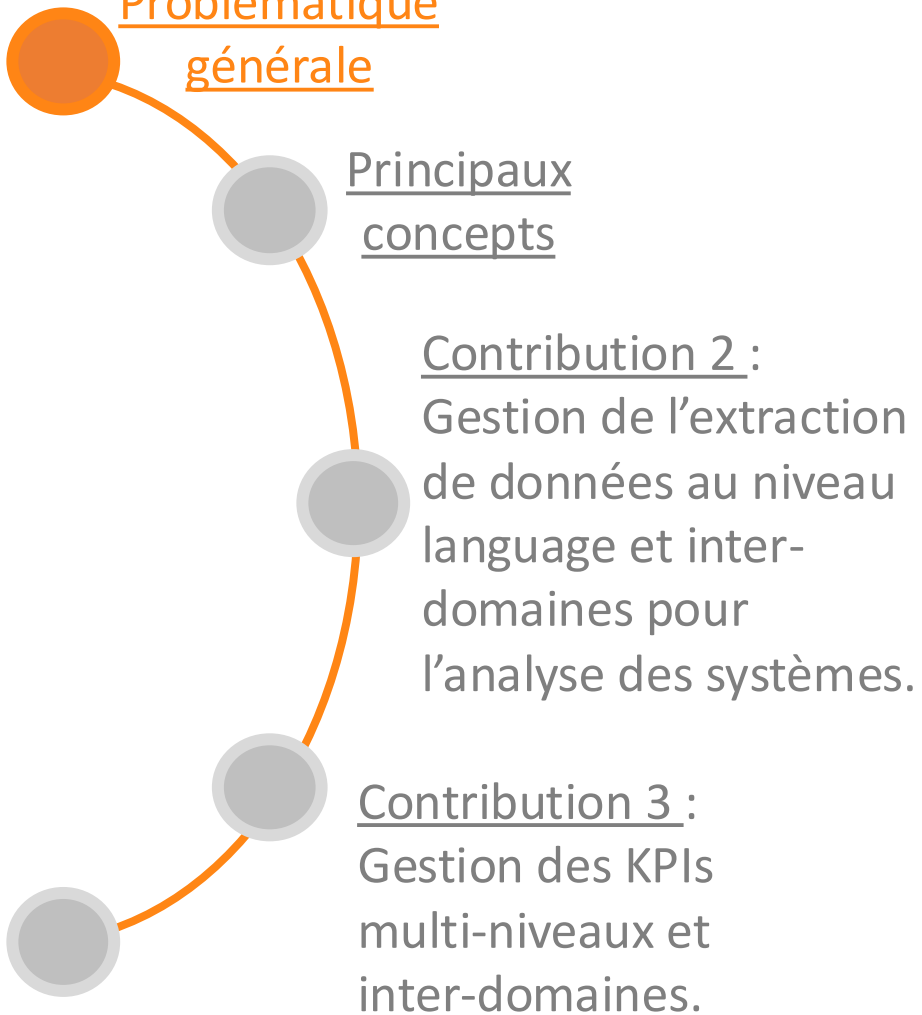


Introduction : évaluation de la performance basée sur les langages exécutables spécifiques au domaine

- **Les langages exécutables spécifiques au domaine (xDSLs)** sont bien adaptés pour modéliser et simuler le système.
- La **simulation du système** génère *des traces d'exécution*.
- Les traces d'exécution vont permettre une **évaluation précoce** des performances.



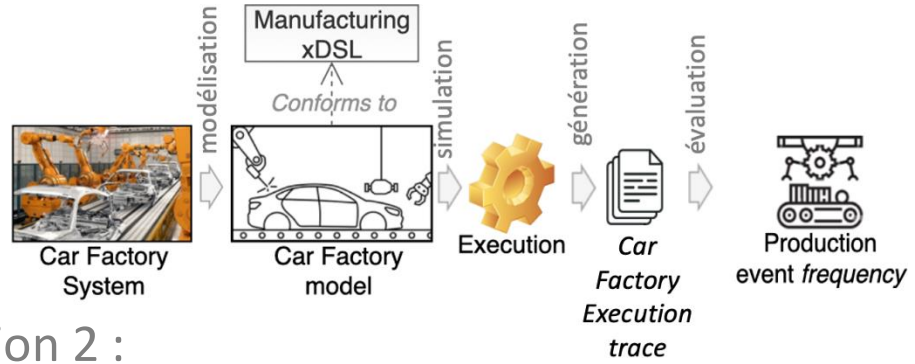
Plan de l’exposé



Plan de l’exposé

Problématique générale

Principaux concepts



Contribution 2 : Gestion de l’extraction de données au niveau language et inter-domaines pour l’analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveaux et inter-domaines.

Conclusion

Plan de l'exposé

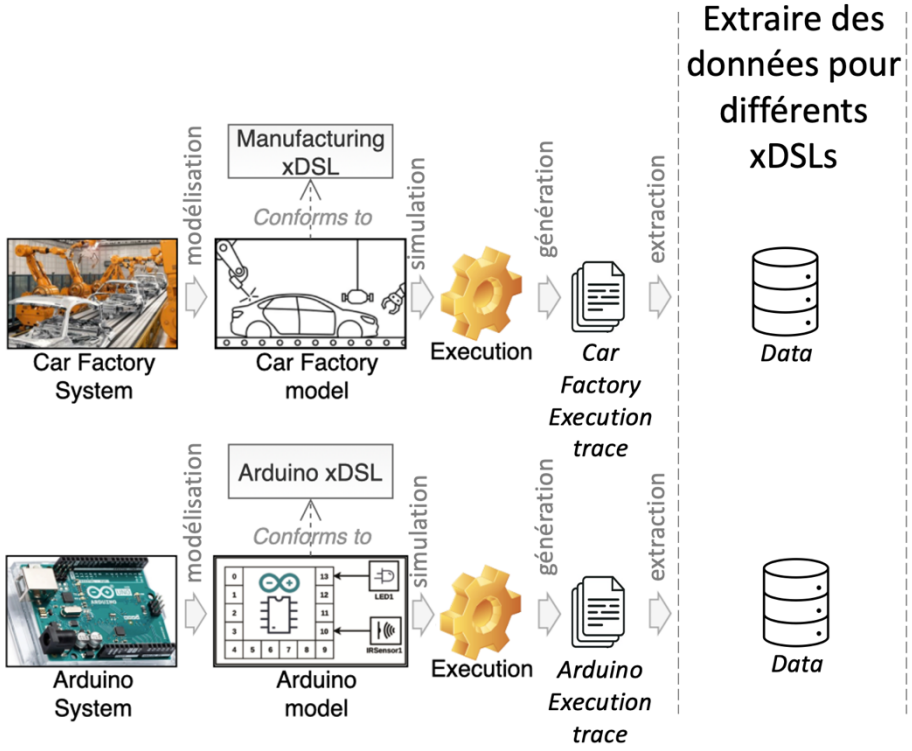
Problématique générale

Principaux concepts

Contribution 2 : Gestion de l'extraction de données au niveau language et inter-domaines pour l'analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveaux et inter-domaines.

Conclusion



Plan de l’exposé

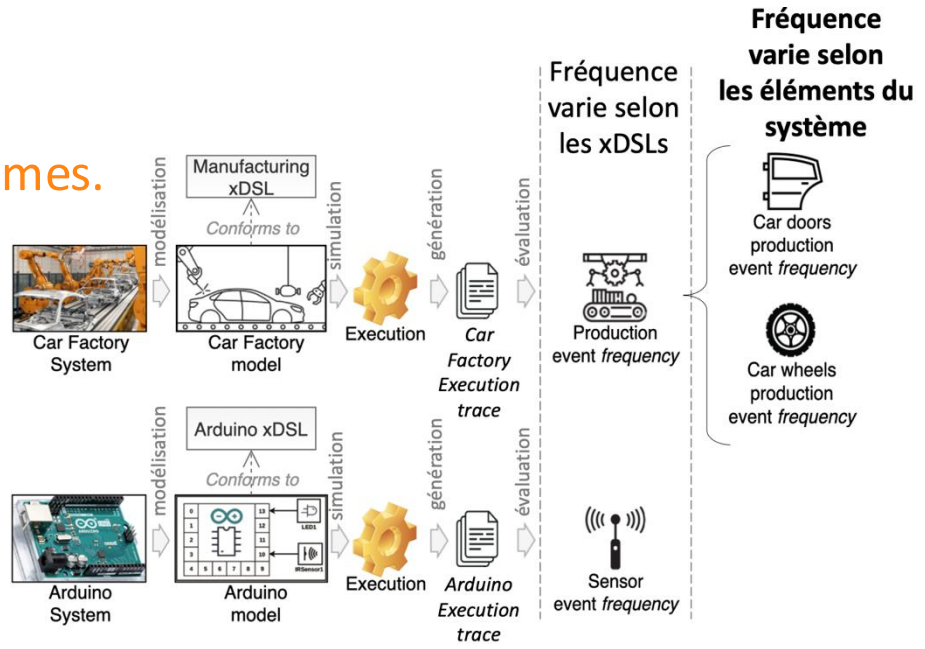
Problématique générale

Principaux concepts

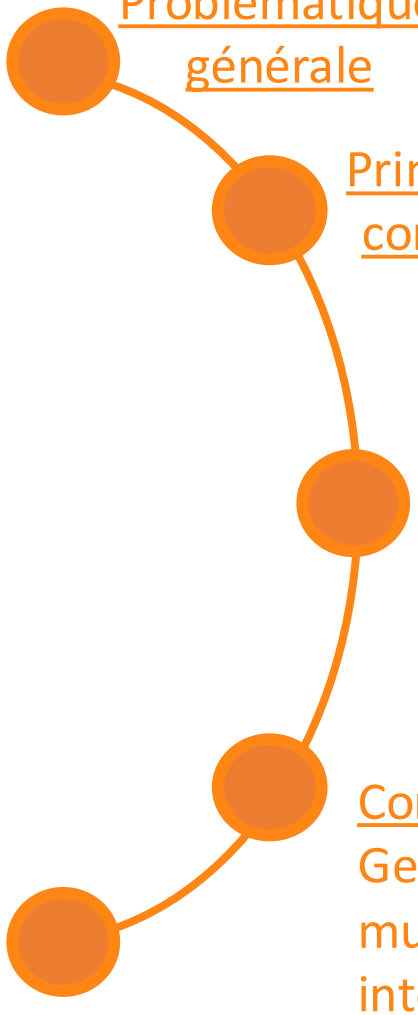
Contribution 2 : Gestion de l’extraction de données au niveau language et inter-domaines pour l’analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveaux et inter-domaines.

Conclusion



Plan de l’exposé



Problématique générale

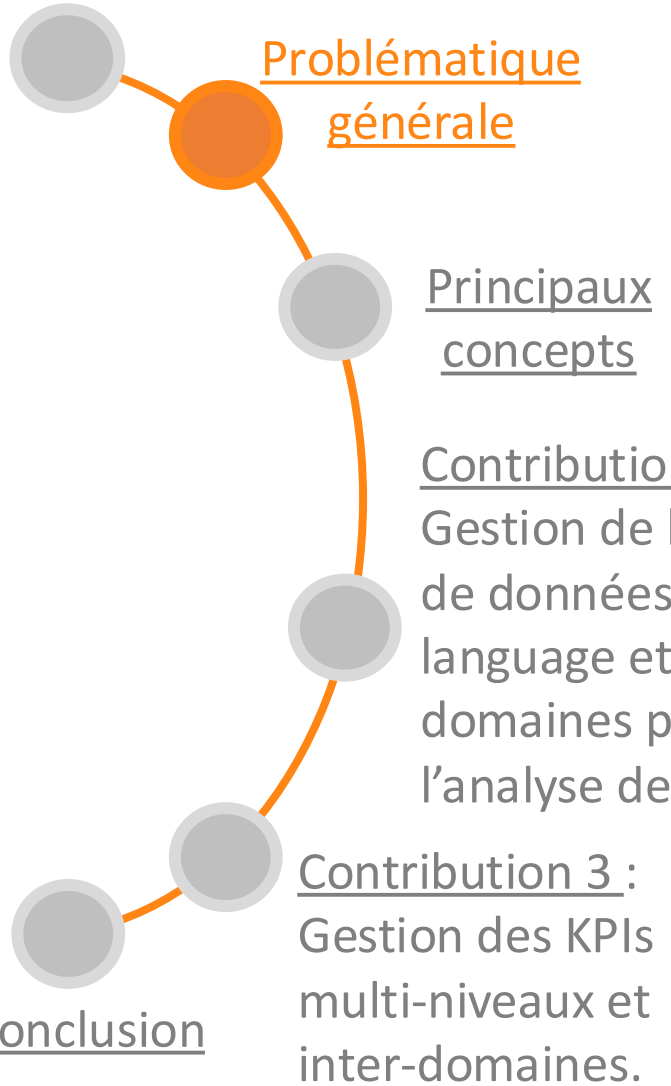
Principaux concepts

Contribution 2 : Gestion de l’extraction de données au niveau language et inter-domaines pour l’analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveaux et inter-domaines.

Conclusion

Plan de l’exposé



L’évaluation de la performance nécessite la mesure des KPIs (1/2)

Indicateurs clés de performance (KPI) :

Ensemble de **métriques utilisées** pour évaluer les performances d’un système.

❑ Exemple : **KPI de *fréquence***

Nombre d’événements sur une période donnée.

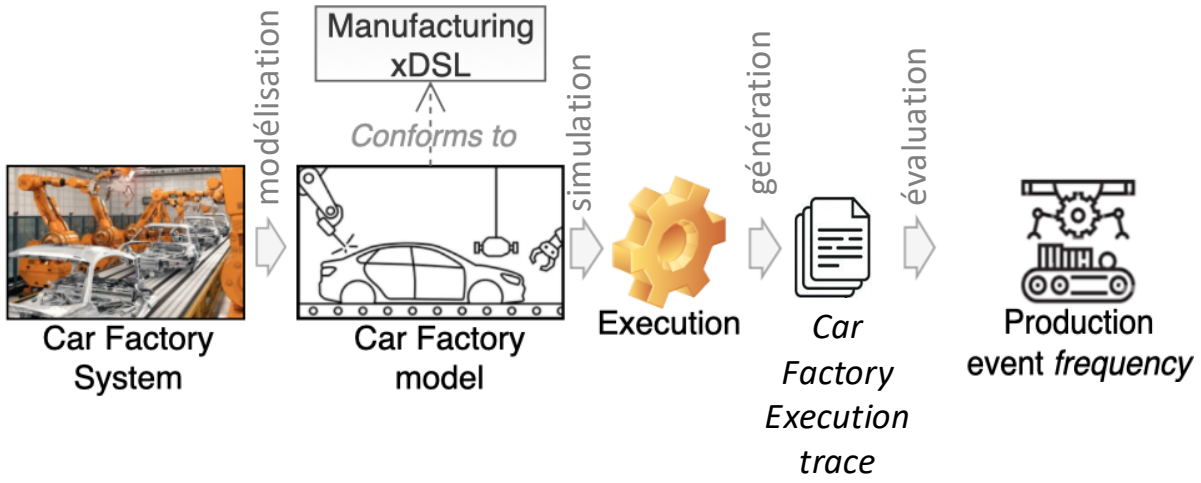
L’évaluation de la performance nécessite la mesure des KPIs (1/2)

Indicateurs clés de performance (KPI) :

Ensemble de **métriques utilisées** pour évaluer les performances d’un système.

❑ Exemple : **KPI de *fréquence***

Nombre d’événements sur une période donnée.



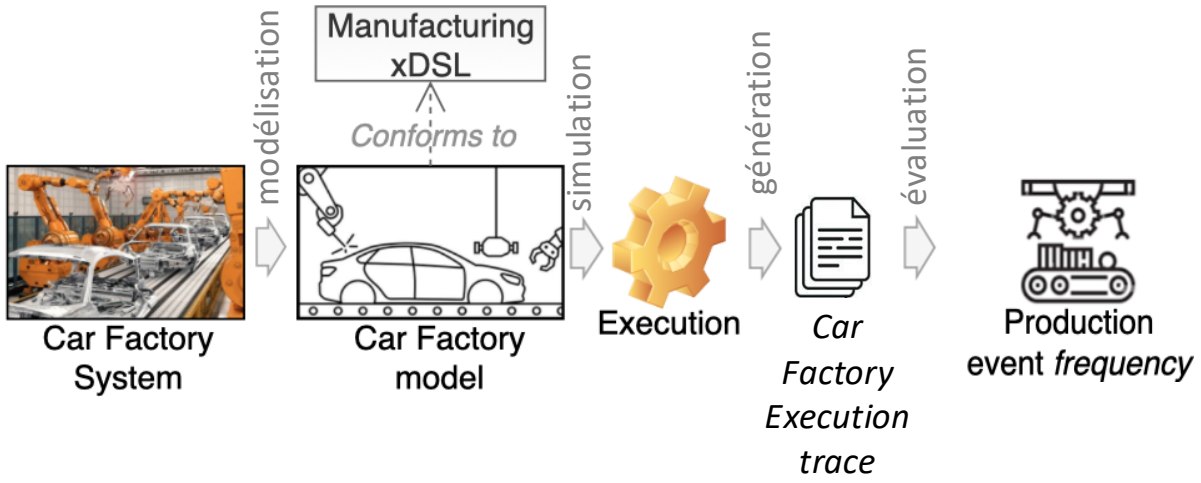
L’évaluation de la performance nécessite la mesure des KPIs (1/2)

Indicateurs clés de performance (KPI) :

Ensemble de **métriques utilisées** pour évaluer les performances d’un système.

❑ Exemple : **KPI de *fréquence***

Nombre d’événements sur une période donnée.



Deux défis principaux :

1. **Extraction des données** depuis la trace
2. **Définition de la fréquence selon le xDSL**

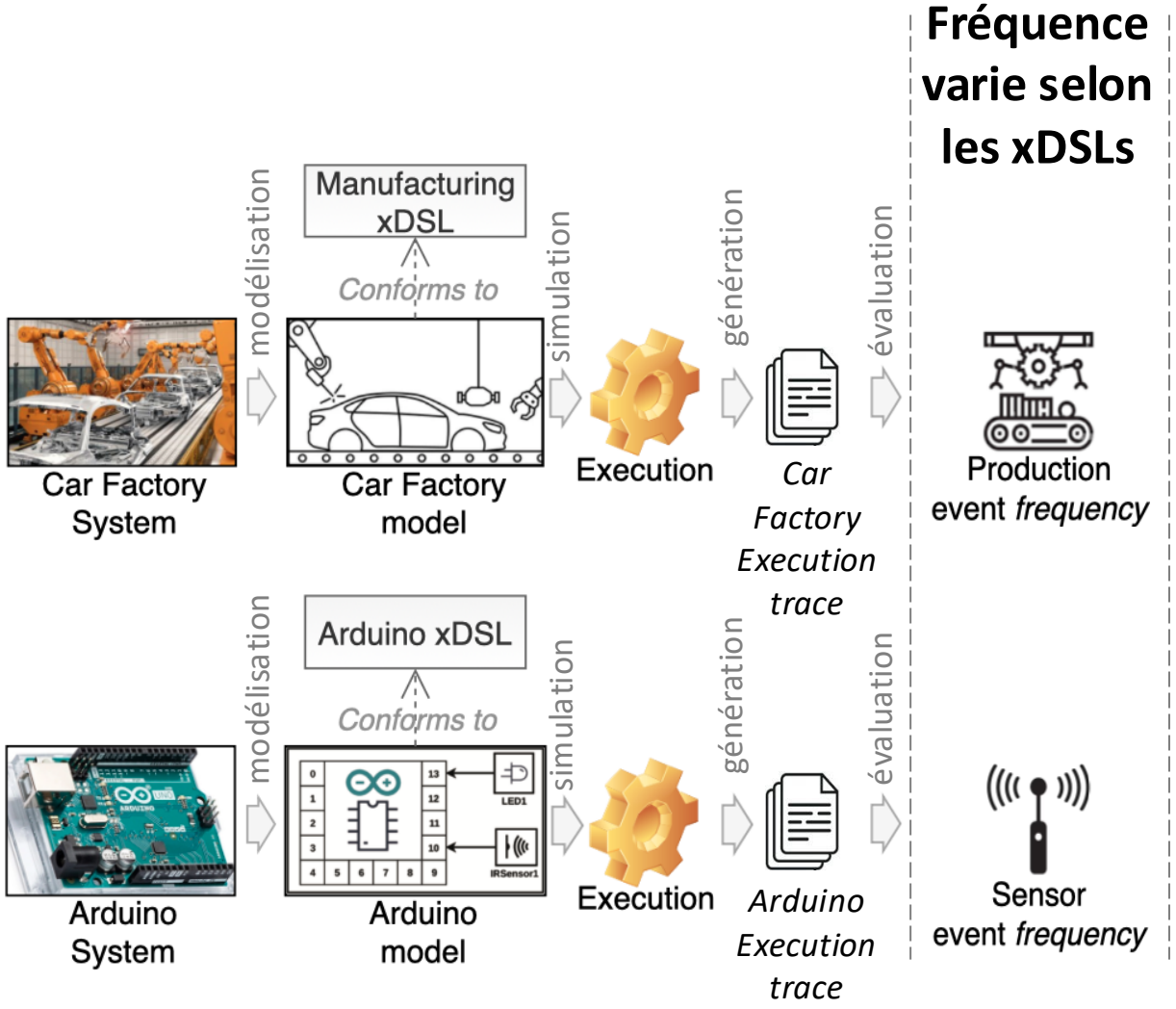
L’évaluation de la performance nécessite la mesure des KPIs (1/2)

Indicateurs clés de performance (KPI) :

Ensemble de **métriques utilisées** pour évaluer les performances d’un système.

❑ Exemple : **KPI de *fréquence***

Nombre d’événements sur une période donnée.



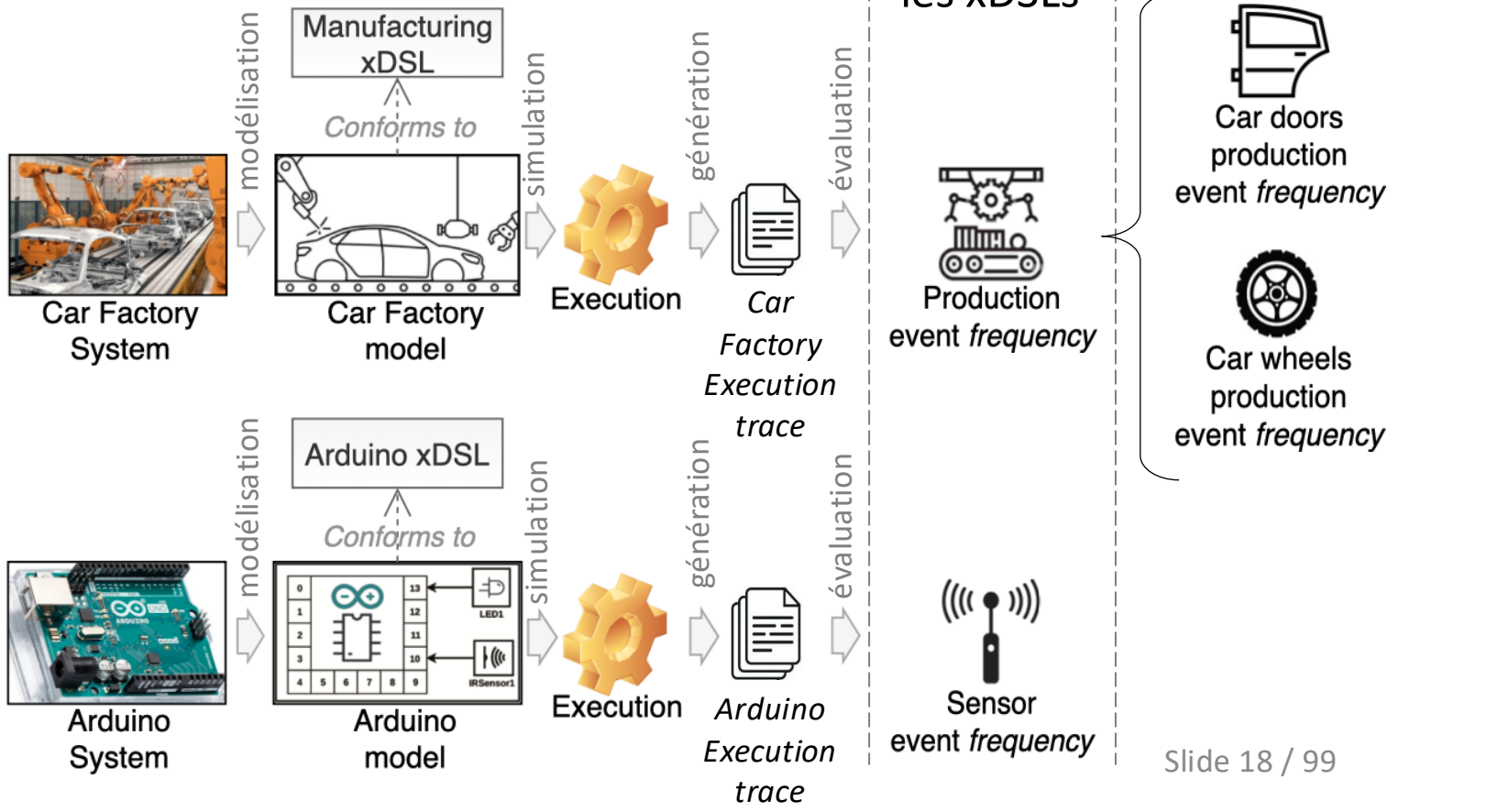
L'évaluation de la performance nécessite la mesure des KPIs (1/2)

Indicateurs clés de performance (KPI) :

Ensemble de **métriques utilisées** pour évaluer les performances d'un système.

❑ Exemple : **KPI de *fréquence***

Nombre d'événements sur une période donnée.



L’évaluation de la performance nécessite la mesure des KPIs (2/2)

→Pour mesurer les KPIs, cela nécessite :

1. La spécification du **xDSL**.
2. La spécification d’un **système**.
3. **L’extraction des** données d’une **trace**.

➢ Ainsi, les KPIs varient selon **les niveaux d’abstraction** et les **domaines hétérogènes**, et **la gestion de cette variabilité est complexe**.

L’évaluation de la performance nécessite la mesure des KPIs (2/2)

➢ Ainsi, les KPIs varient selon **les niveaux d’abstraction** et les **domaines hétérogènes**, et **la gestion de cette variabilité est complexe**.

Extraire des données et mesurer des KPIs nécessitent

(1) implémenter un programme, ou (2) adapter le xDSL considéré :

- Coûteux,
- Chronophage,
- Sujet aux erreurs,
- Peu réutilisable,
- Nécessite de solides compétences en développement logiciel.

Problématique : résumé

1. Concevoir un langage de requête
   - permettant d'exploiter les traces d'exécution , indépendamment du domaine d'application,
   - réutilisable et facile à utiliser par des experts métier.

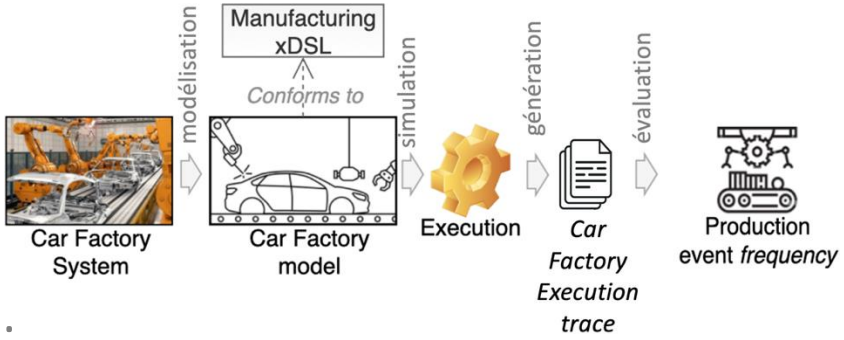
2. Concevoir un langage d'évaluation de performance
   - applicable à plusieurs xDSL de différents domaines, sans nécessité d'extensions des xDSLs,
   - dédié aux experts métier pour **éditer** les formules des KPIs, **appliquer les KPIs** aux concepts du domaine, **configurer les KPIs**, une fois définis au niveau du langage, pour leurs systèmes spécifiques.

Plan de l'exposé

Introduction

Problématique générale

Principaux concepts



Contribution 2 : Gestion de l'extraction de données au niveau language et inter-domaines pour l'analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveaux et inter-domaines.

Conclusion

Système et modèle

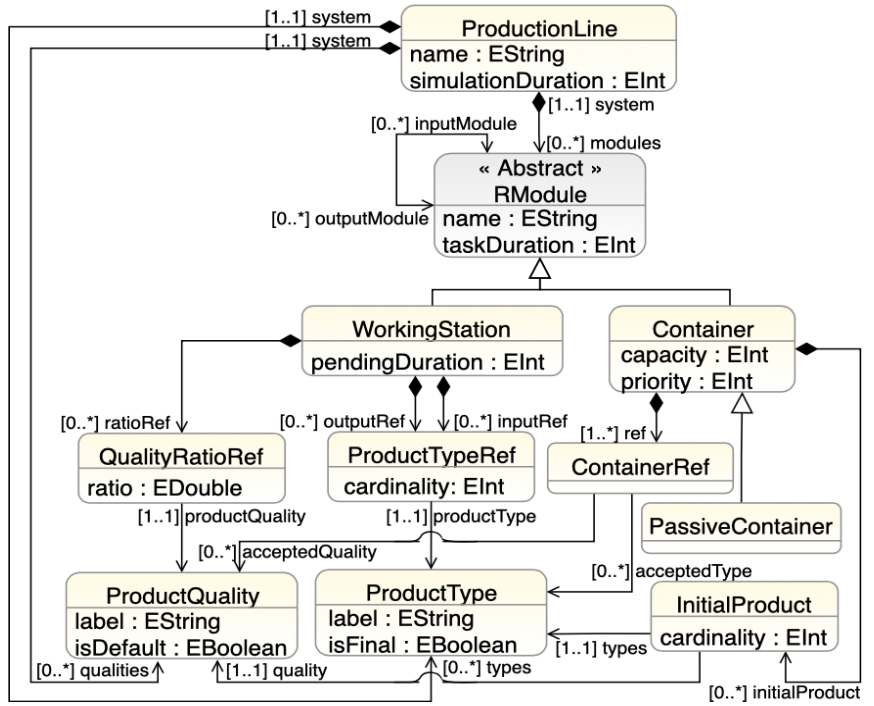
- **Un système est** défini par :
  1. **Éléments,**
  2. **Inter-connectivité,**
  3. **Objectif.**
- Un système peut changer **d'état** pour répondre à des **entrées** et produire des **sorties**, ce qui révèle un **comportement**.

- **Un modèle est** une représentation d'un système, qui dépend **de l'objectif de modélisation.**
- Un modèle est **valide** s'il constitue une représentation fidèle au système modélisé selon un objectif donné.

Comment créer et simuler un modèle de système ?

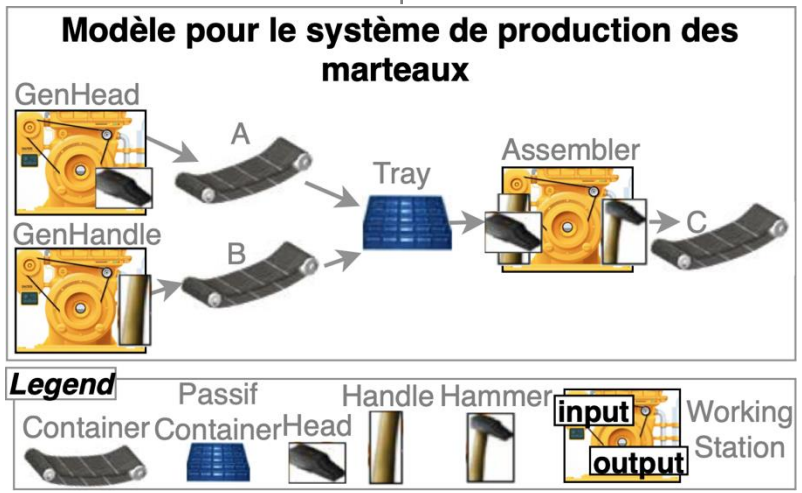
- Un **méta-modèle** (**syntaxe abstraite**) permet de définir un modèle.
- Pour simuler le comportement du système, des *modèles exécutables* sont nécessaires ; Une **sémantique d'exécution** est alors définie pour ces modèles.
- Une **syntaxe concrète** (graphique ou textuelle) est utilisée pour illustrer le modèle.

➢ Executable Domain-Specific Languages (xDSLs)



Syntaxe abstraite du SMS xDSL (méta-modèle)

conforme à

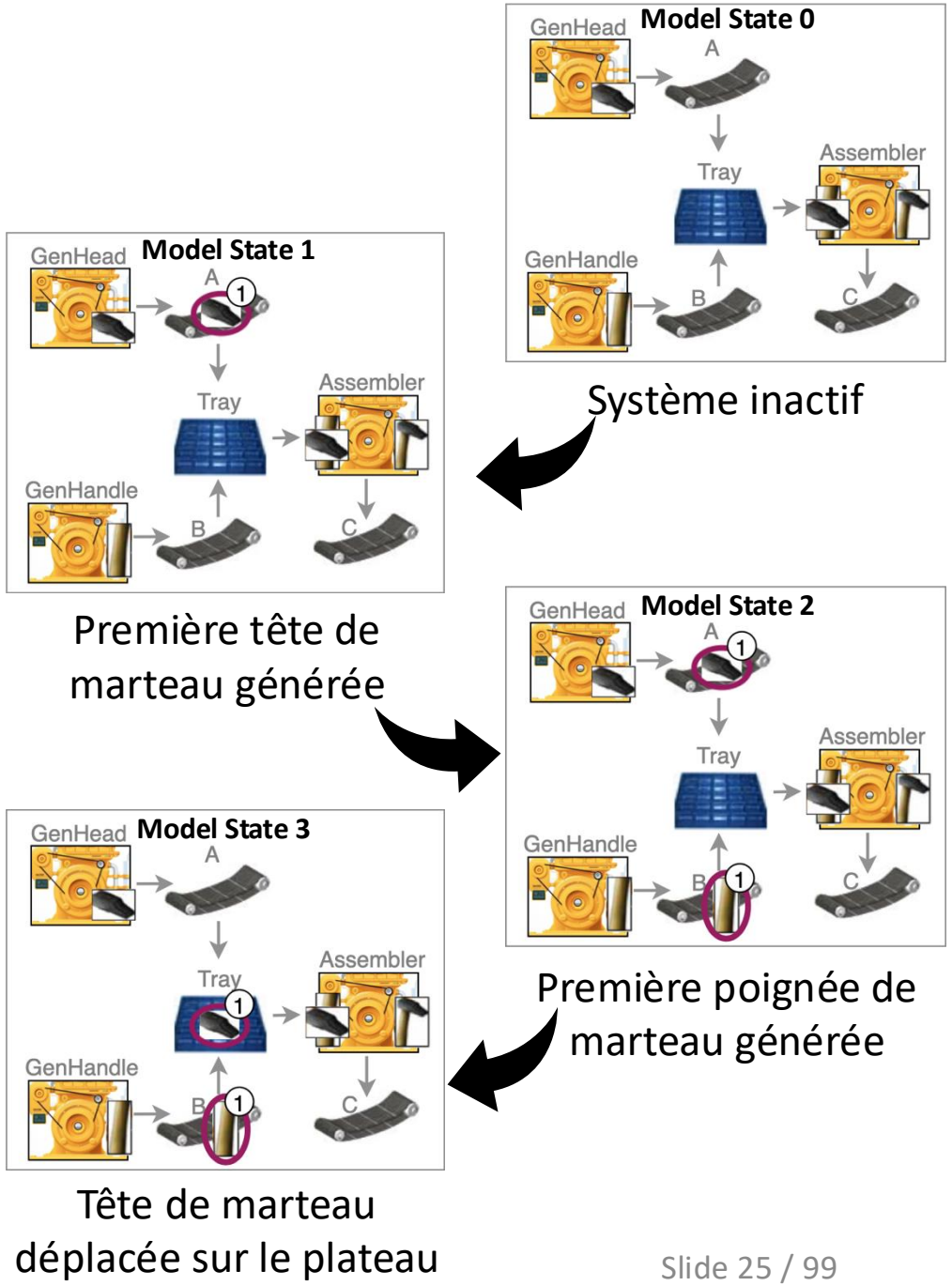


Traces d'exécution

- La **sémantique d'exécution** correspond à des règles d'exécution qui font évoluer le modèle d'un **état** à un autre, permettant ainsi son exécution.

- La trace d'exécution :
  - capture ces états du système pendant l'exécution,
  - permet d'accéder aux **données générées** a posteriori.

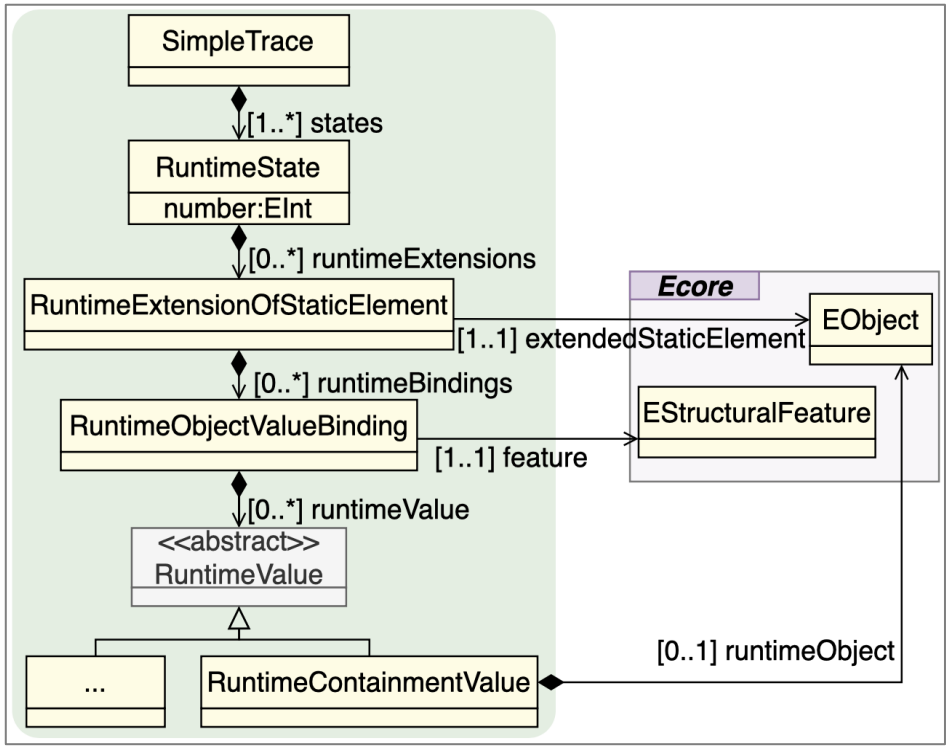
- Ces données permettent l'analyse du système, telle que **l'évaluation de la performance**.



Traces d'exécution - Modèle

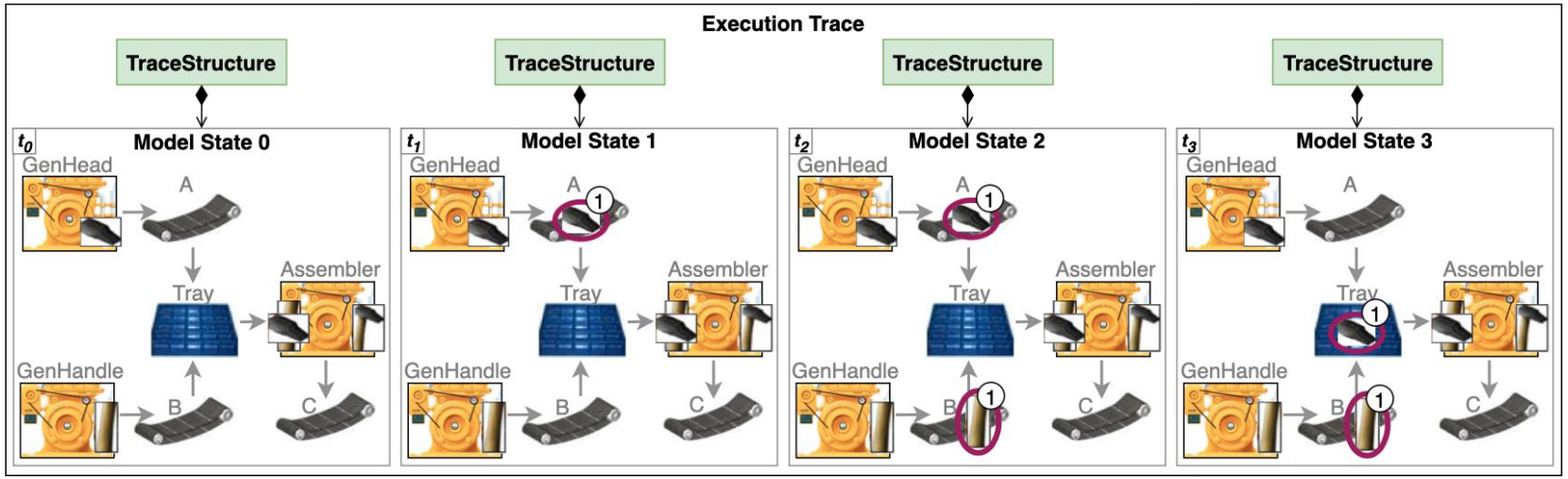
- Dans les approches d'ingénierie dirigée par les modèles, **les traces d'exécution sont des modèles**.

- ***SimpleTrace*** est un exemple de langage permettant de définir une trace d'exécution.

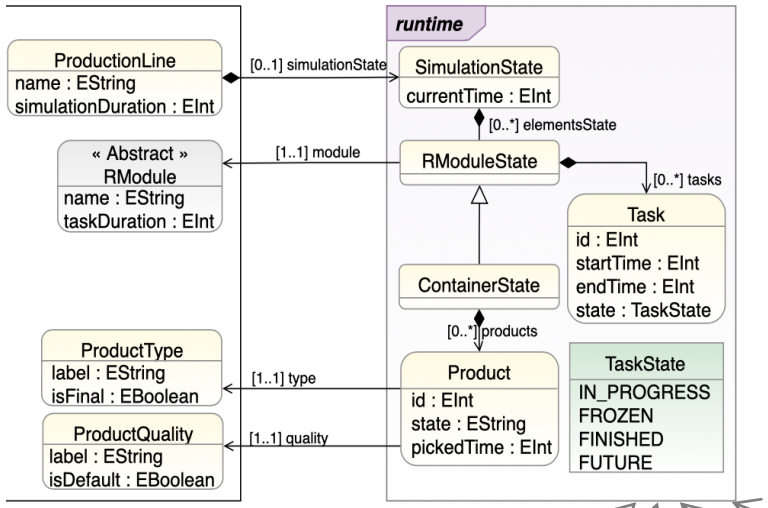


*Extrait du méta modèle **SimpleTrace***

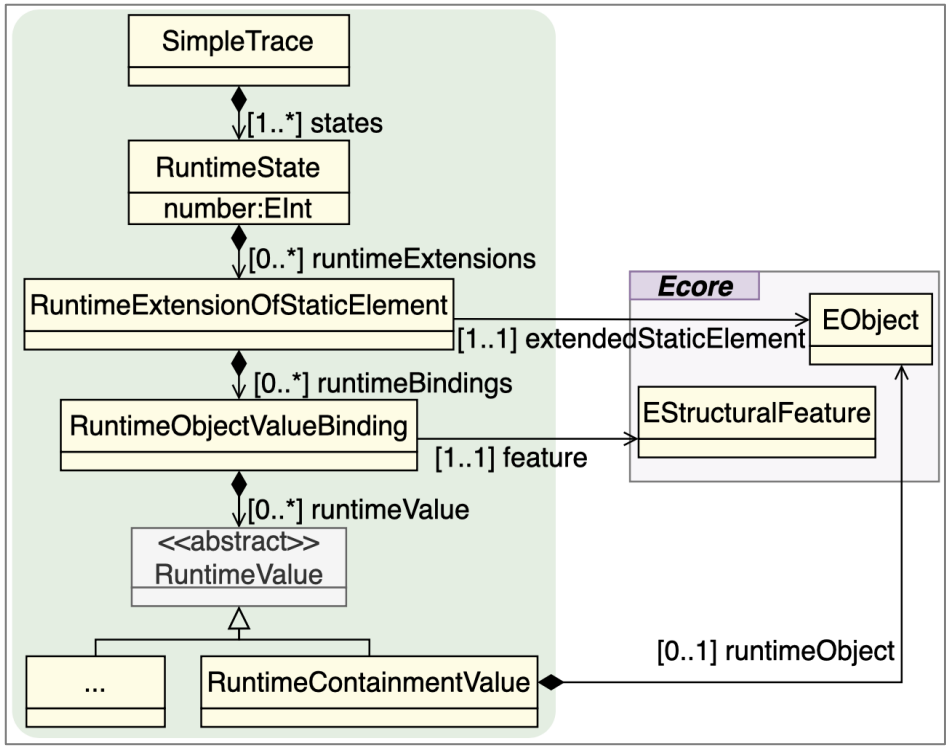
conforme à



Traces d'exécution - Modèle



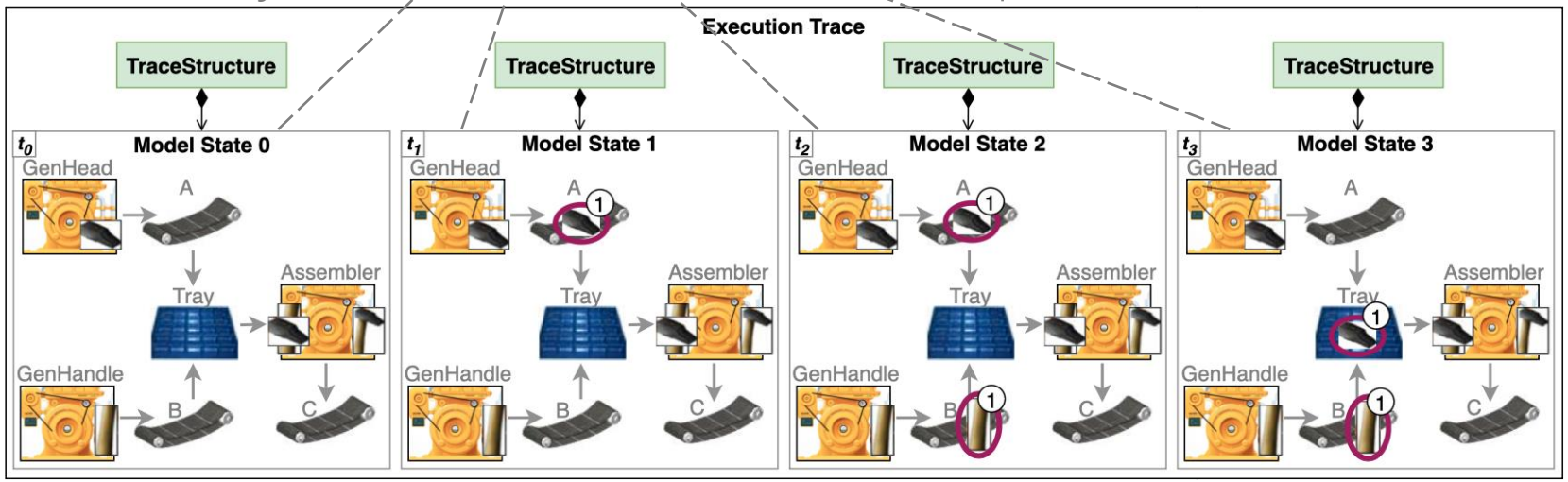
Définition de l'état d'exécution



*Extrait du méta modèle **SimpleTrace***

conforme à

conforme à

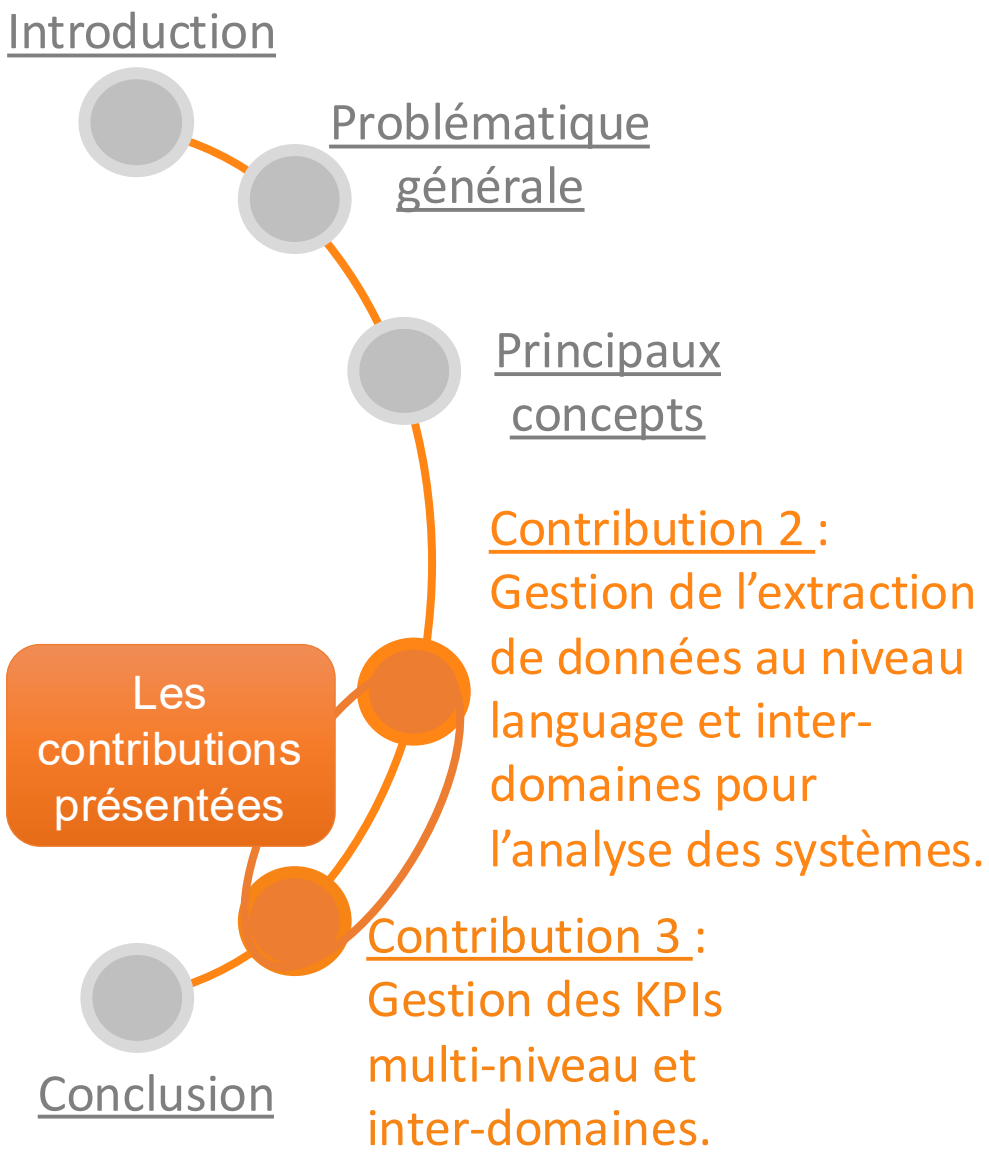


Les indicateurs clés de performance

- Les données de simulation sont extraites des traces d'exécution.
- Des indicateurs clés de performance (KPI) sont calculés à partir de ces données.
- Il existe différentes classifications des KPIs ; nous considérons ici des KPIs dynamiques, qui peuvent être génériques (e.g., débit) ou spécifiques à un domaine (e.g., consommation d'énergie).

KPI	Domaine	Définition	Formule
Débit (Throughput)	Industrie, Réseaux, Santé.	Nombre d'entités livrées sur une période d'exploitation.	Débit = Nombre d'entités / Durée
Consommation d'énergie	Systèmes électroniques	Quantité d'énergie consommée par un dispositif	Consommation d'énergie = Tension * Courant * Temps de fonctionnement

Plan de l’exposé

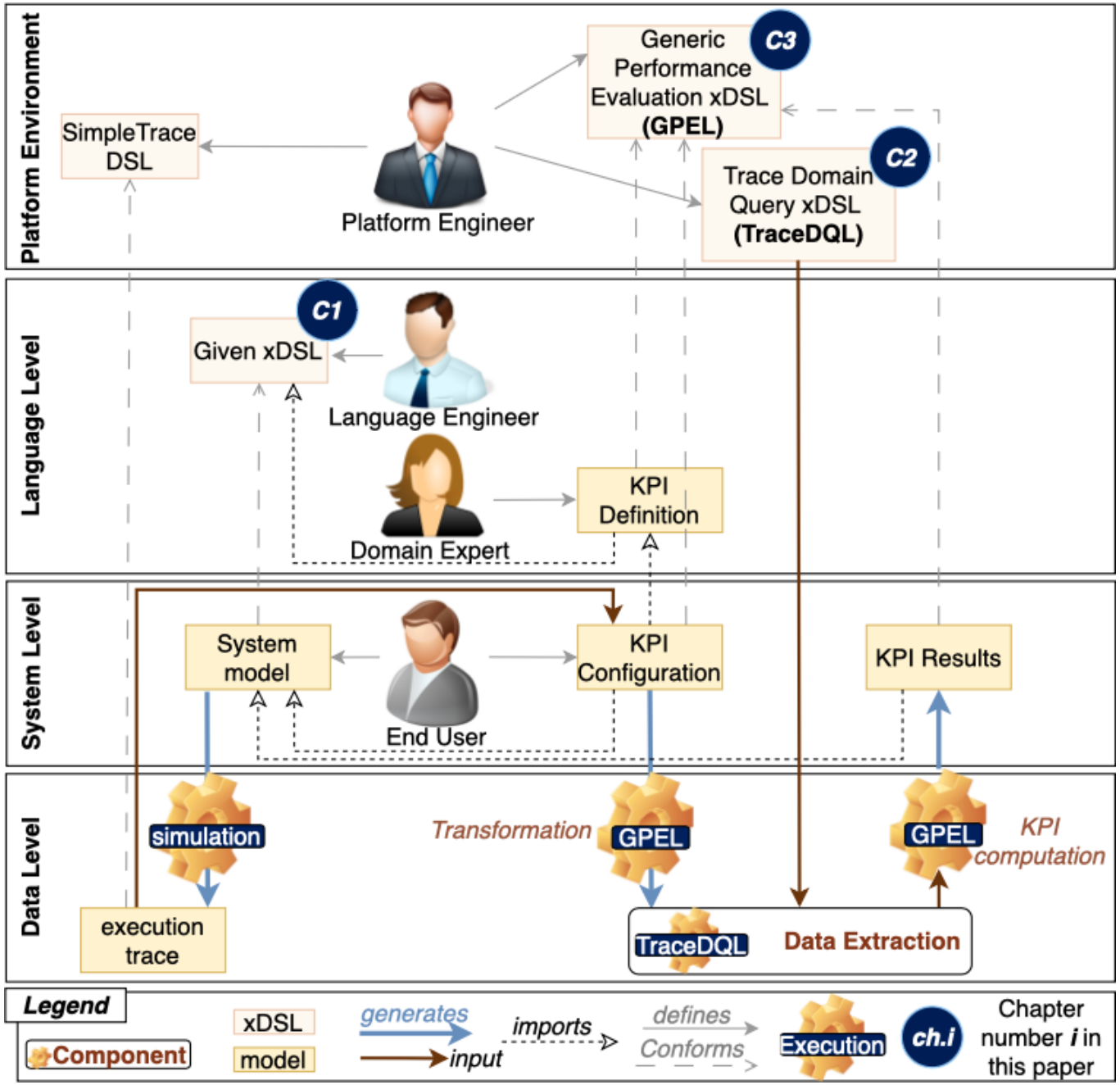


Vue d'ensemble des contributions

- **Contribution 1** : Approche à deux niveaux pour la gestion des KPI au lieu d'une approche à un seul niveau.

Deux contributions (présentées) :

- **Contribution 2 (TraceDQL)** : Approche **générique pour** l'extraction de données (multi-domaines).
- **Contribution 3 (GPEL)** : Approche générique, multi-niveaux pour la **variabilité des** KPIs.

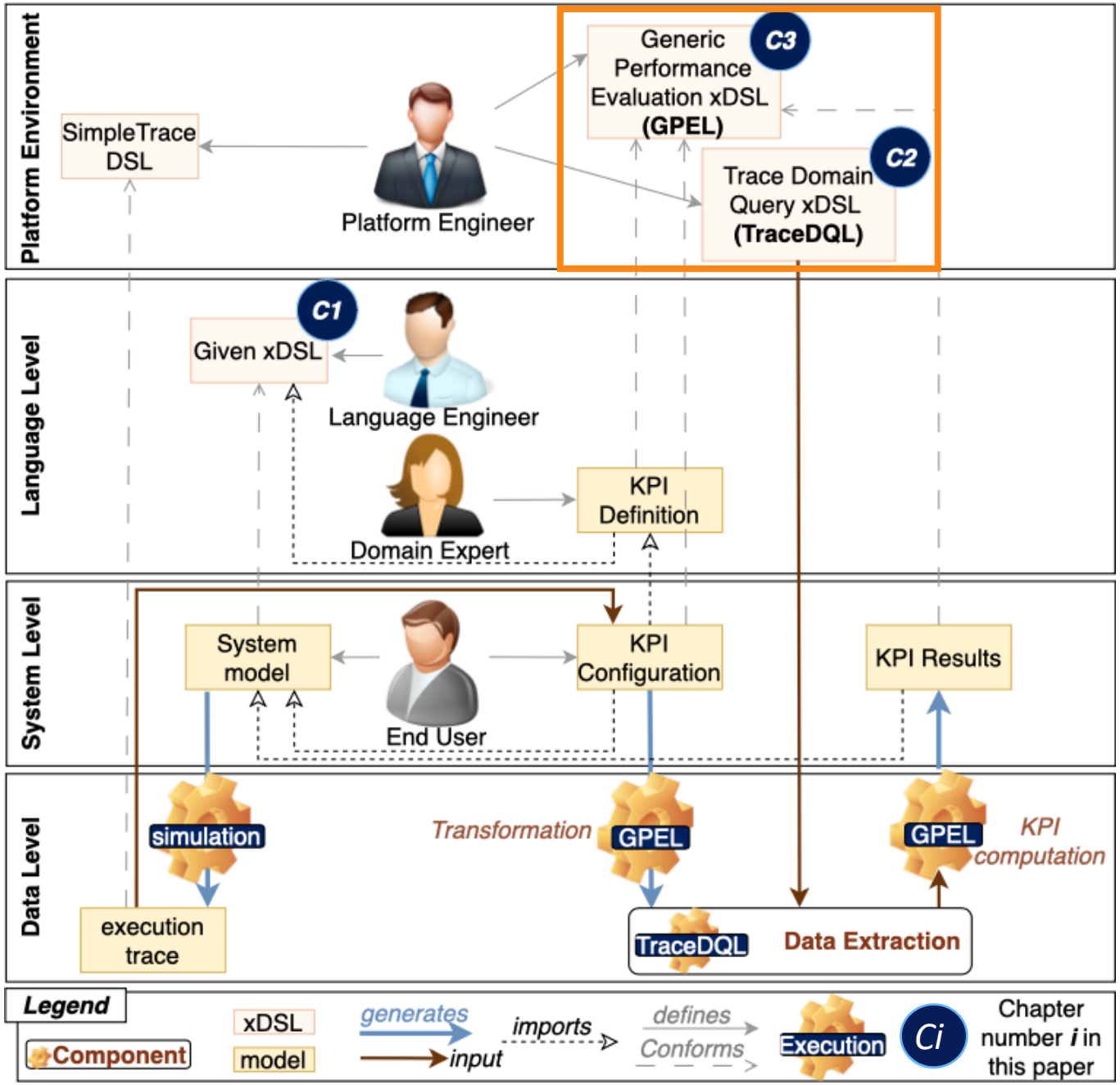


Vue d'ensemble des contributions

- **Contribution 1** : Approche à deux niveaux pour la gestion des KPI au lieu d'une approche à un seul niveau.

Deux contributions (présentées) :

- **Contribution 2 (TraceDQL)** : Approche **générique pour** l'extraction de données (multi-domaines).
- **Contribution 3 (GPEL)** : Approche générique, multi-niveaux pour la **variabilité des** KPIs.



Plan de l'exposé

Introduction

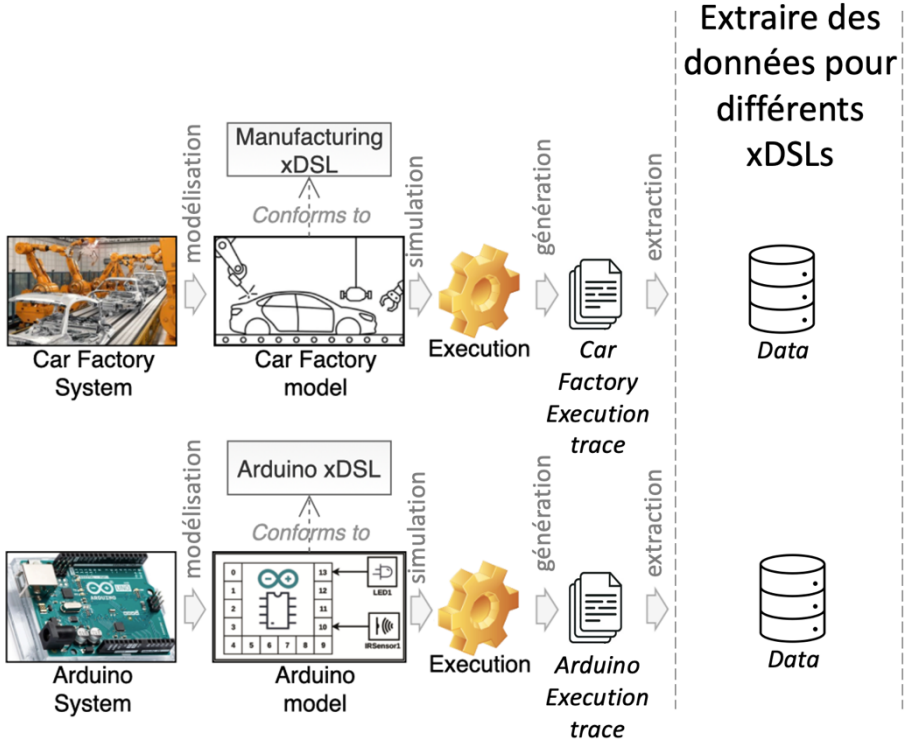
Problématique générale

Principaux concepts

Contribution 2 : Gestion de l'extraction de données au niveau language et inter-domaines pour l'analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveau et inter-domaines.

Conclusion



Langage pour naviguer dans la trace d'exécution des modèles de xDSL

Les langages exécutables spécifiques au domaine (xDSL) permettent l'analyse des systèmes en supportant :

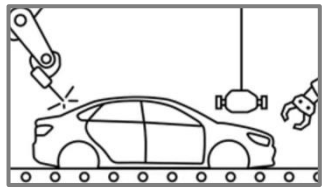
- la modélisation du système,
- la simulation du comportement du système,
- la génération de traces d'exécution.

L'analyse du système évalue l'efficacité du système à travers :

- les tests et la vérification,
- l'évaluation de la performance,
- le diagnostic du système afin de détecter et suivre les problèmes.

Système réel

Modélisation



Modèle du système

Exécution

Génération

Trace d'exécution

Analyse du système

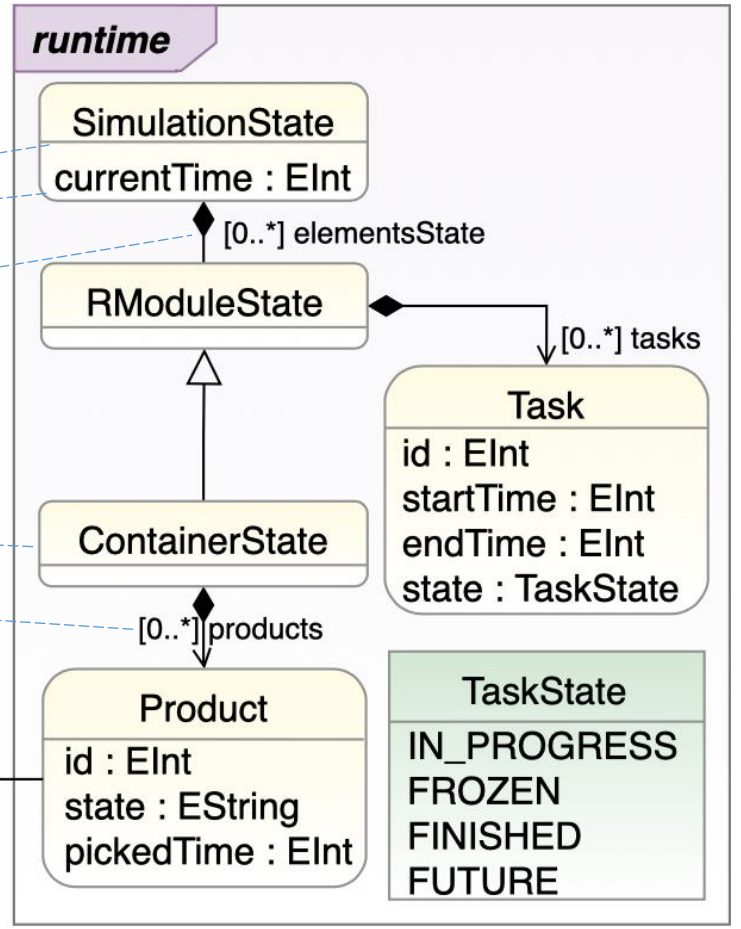
Évaluation de la performance

Analyse : exemple de calcul de KPI implémenté pour le SMS xDSL

Algorithme de calcul du KPI de débit (Throughput)

```
begin
    SMMmodel ← initializeSMMmodel();
    simulation ← executionTrace.states.last.castTo(SimulationState);
    SD ← simulation.currentTime;
    PP ← 0;
    foreach moduleState ∈ simulation.elementsState do
        if moduleState instanceof ContainerState then
            container ← moduleState.castTo(ContainerState);
            foreach p ∈ container.products do
                if p.type.isFinal then
                    PP ← PP + 1;
    end end end end
    storeComputedKPIValue(SMMmodel, PP/SD);
end
```

Navigation dans le méta modèle du SMS xDSL et dans la trace d'exécution



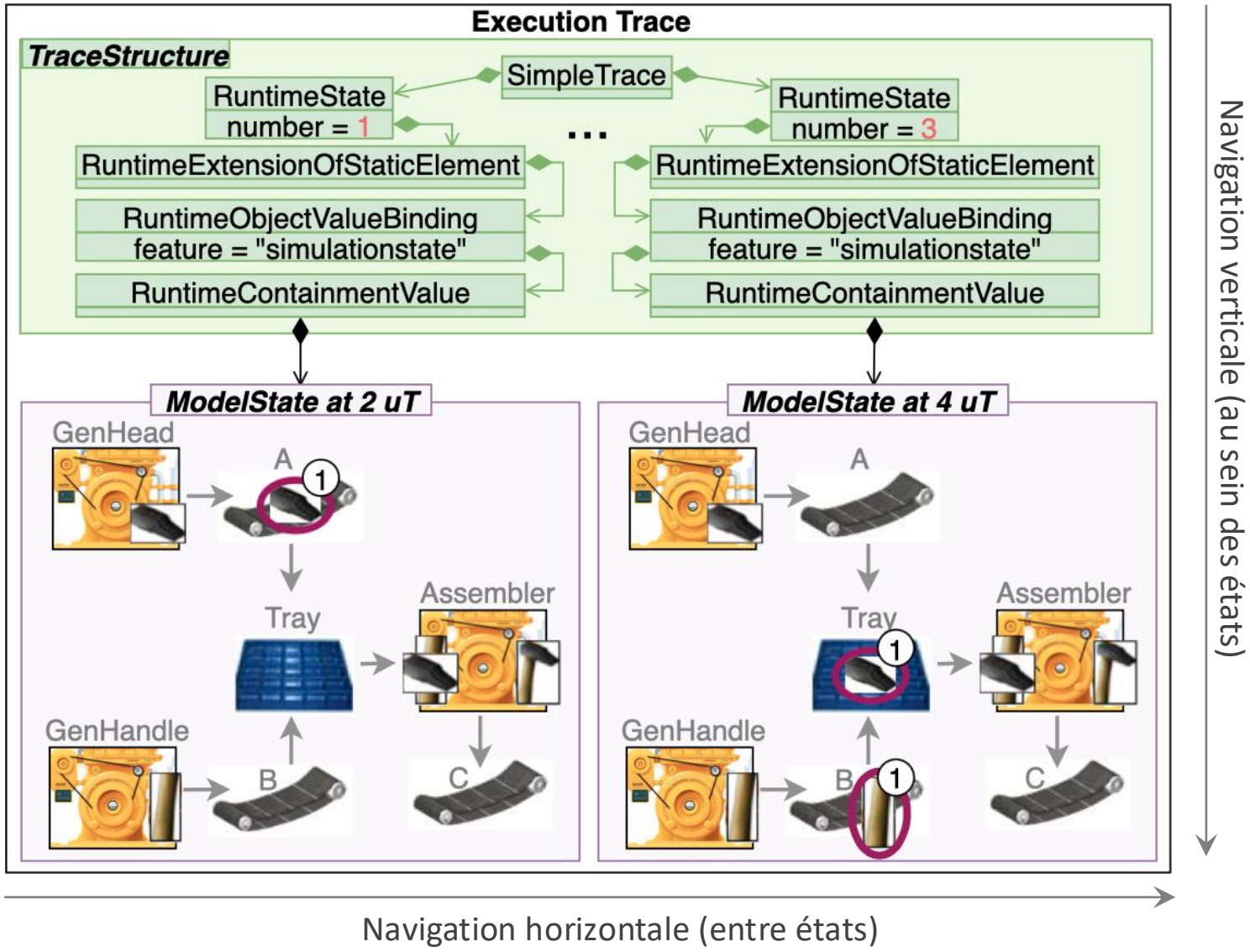
Cet algorithme d'extraction des données de la trace est directement applicable à tout modèle SMS xDSL.

Problème : comment extraire des données à partir d'une trace d'exécution ?

L'écriture de requêtes pour les experts métier est complexe :

- Nécessite une compréhension approfondie de la **structure de la trace**.
- Requiert de bonnes **compétences en programmation**.

La ré-implémentation des requêtes pour chaque nouveau xDSL est coûteuse et chronophage.

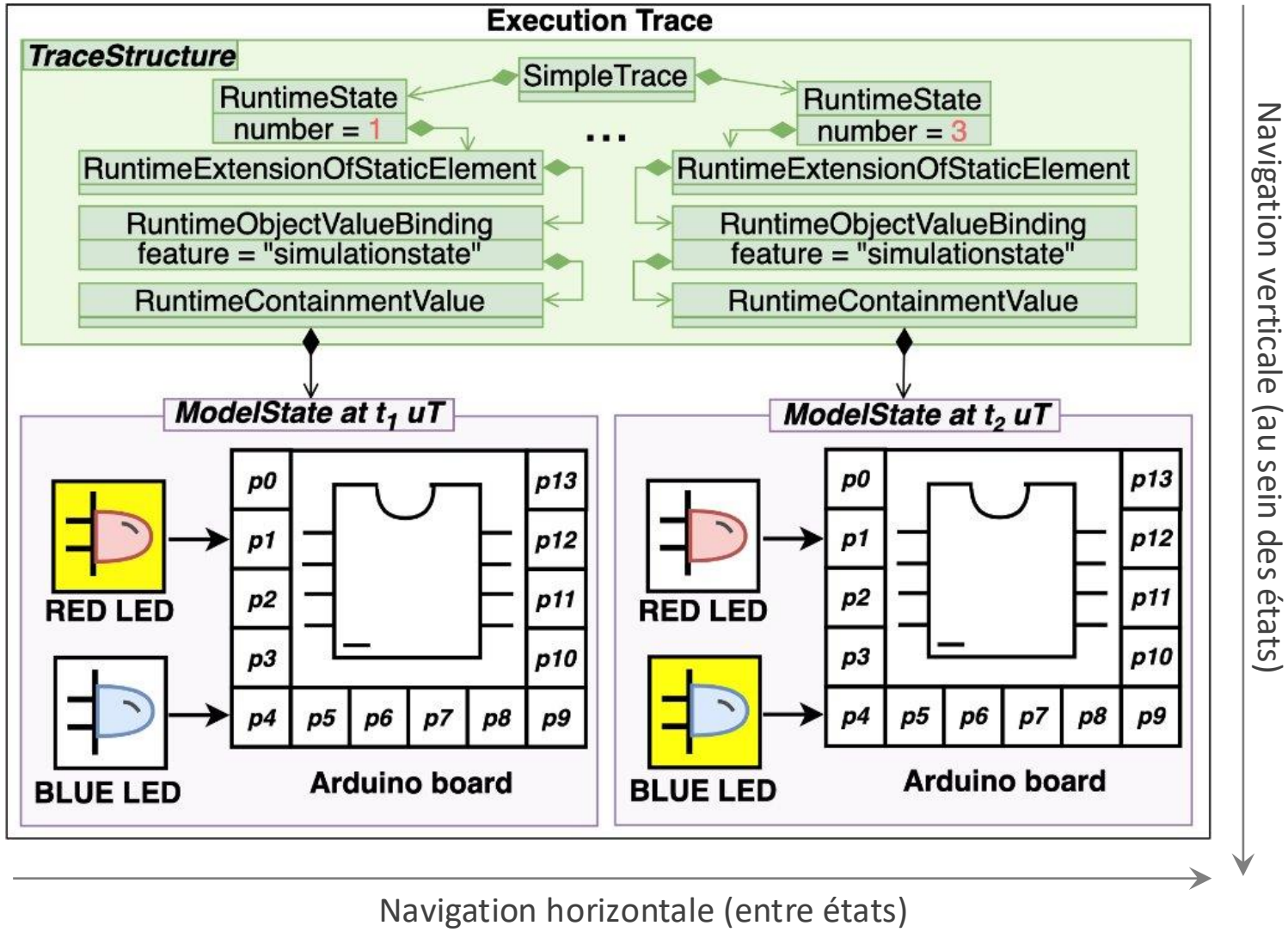


Problème : comment extraire des données à partir d'une trace d'exécution ?

L'écriture de requêtes pour les experts métier est complexe :

- Nécessite une compréhension approfondie de la **structure de la trace**.
- Requiert de bonnes **compétences en programmation**.

La **ré-implémentation** des requêtes pour chaque nouveau xDSL est coûteuse et chronophage.



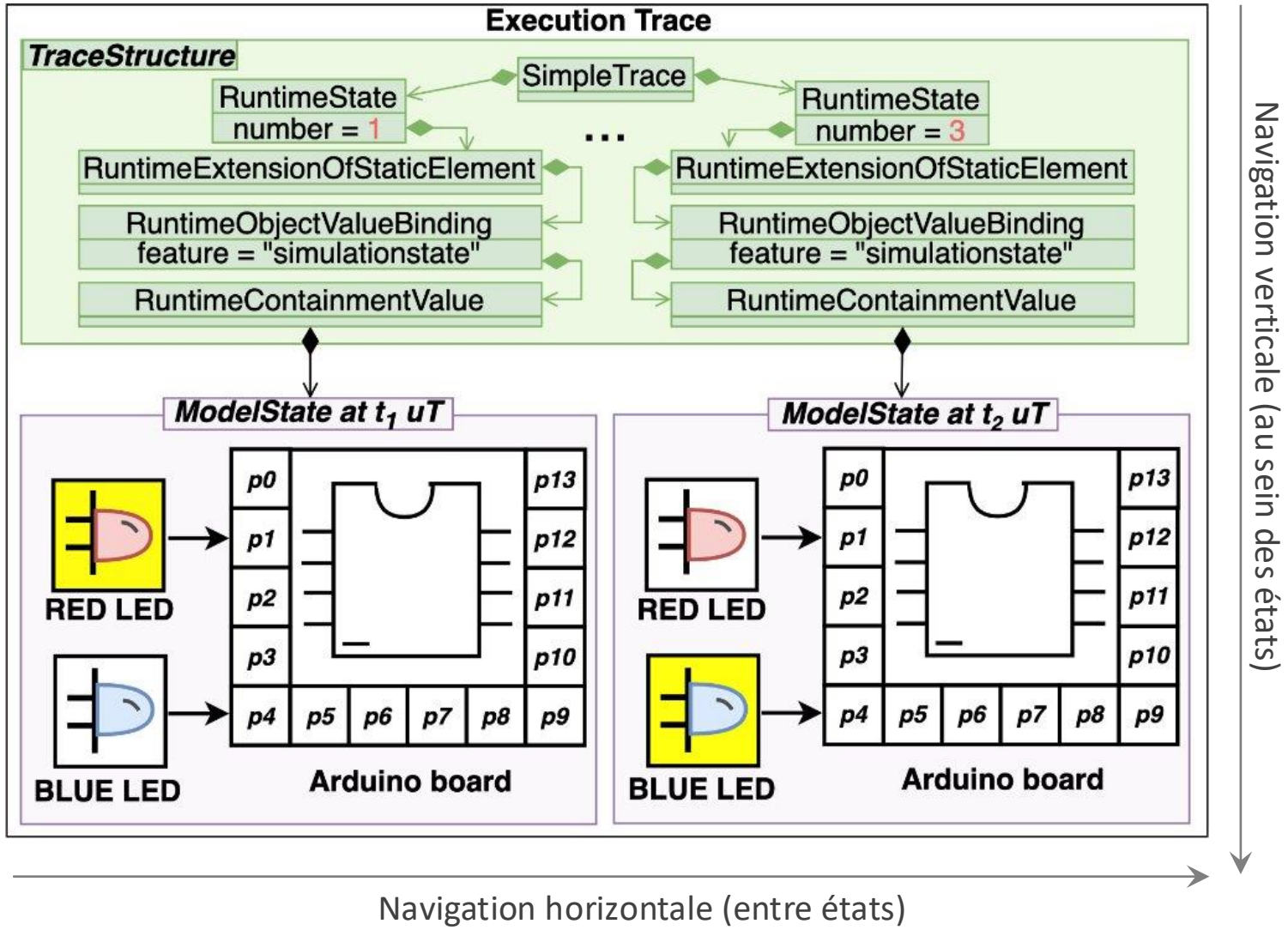
Problème : comment extraire des données à partir d'une trace d'exécution ?

L'écriture de requêtes pour les experts métier est complexe :

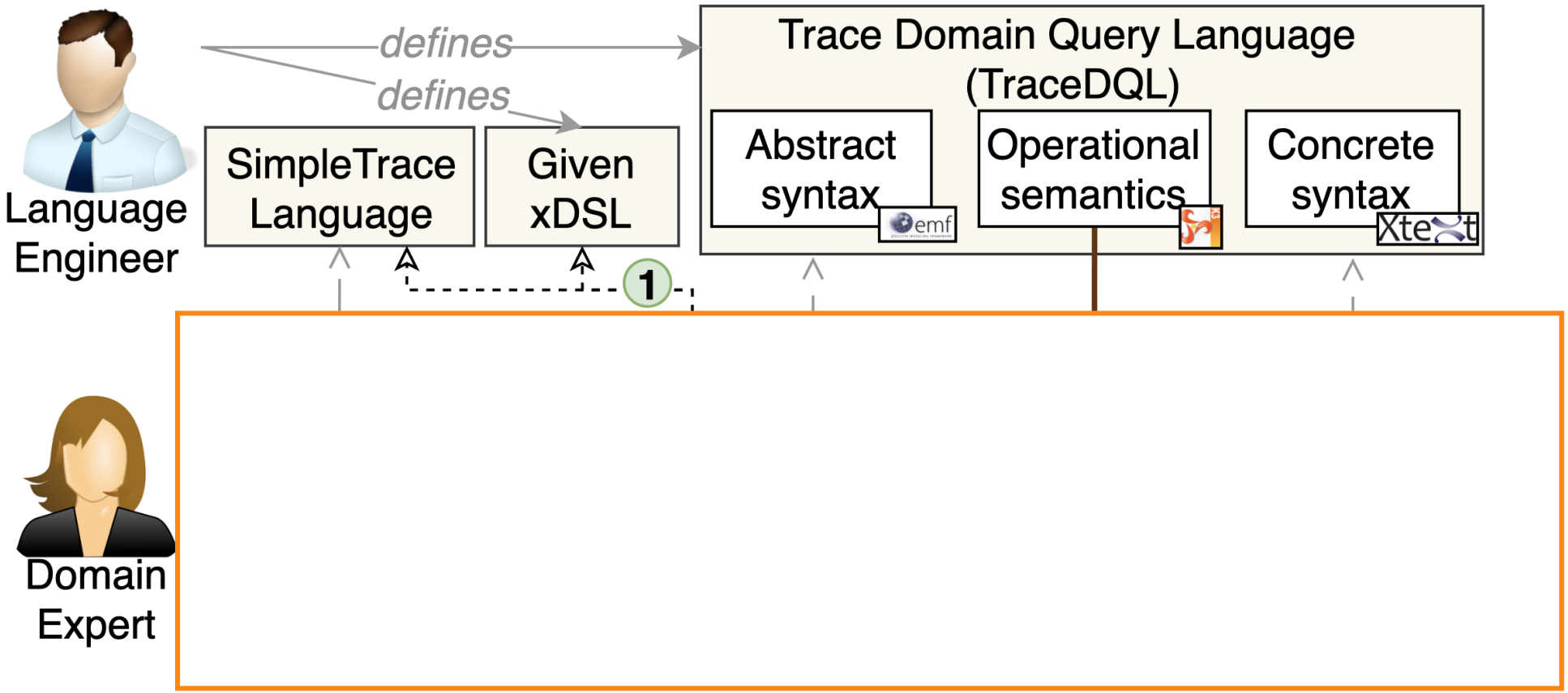
- Nécessite une compréhension approfondie de la **structure de la trace**.
- Requiert de bonnes **compétences en programmation**.

La **ré-implémentation** des requêtes pour chaque nouveau xDSL est coûteuse et chronophage.

➢ **Besoin : Langage de requête générique et permettant la réutilisation pour l'extraction de données de traces (multi-xDSL) ?**

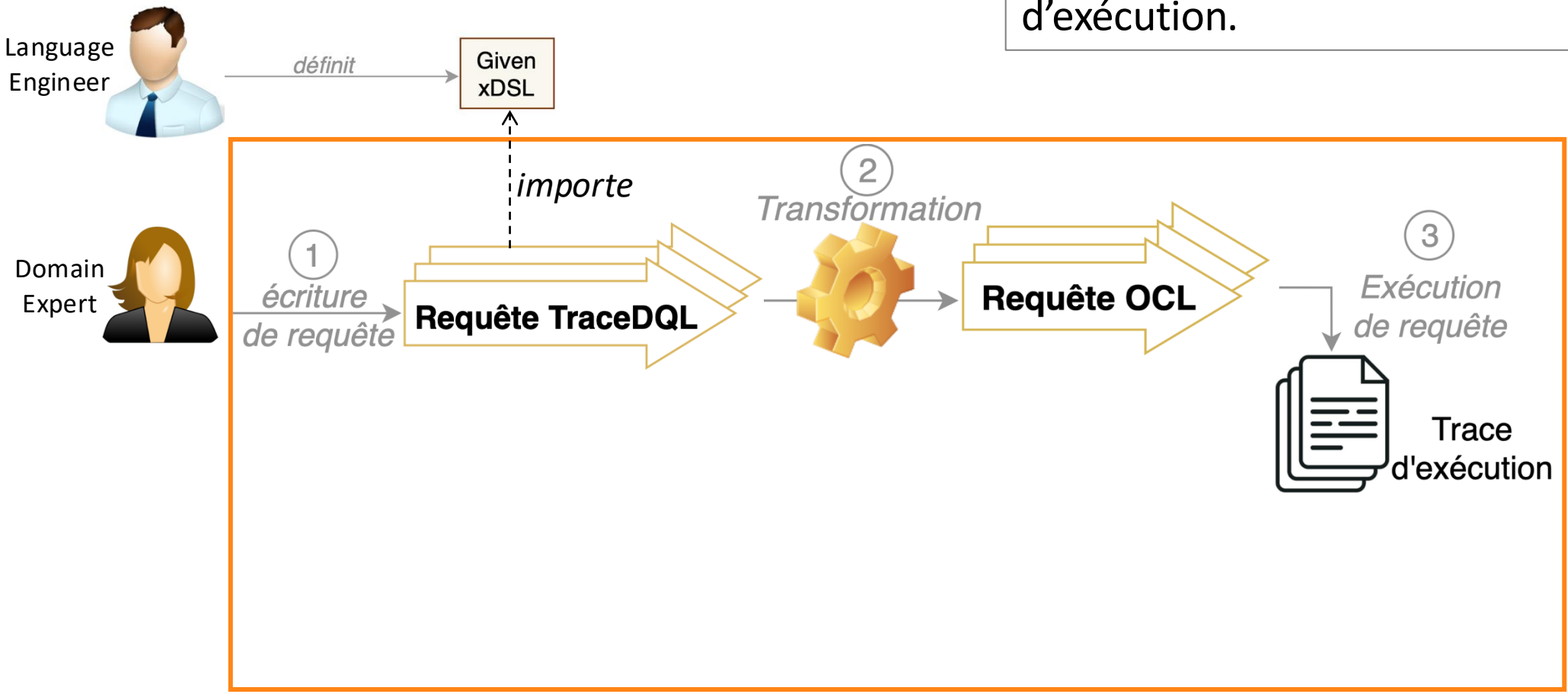


Proposition TraceDQL : vue d'ensemble de l'approche proposée (1/5)



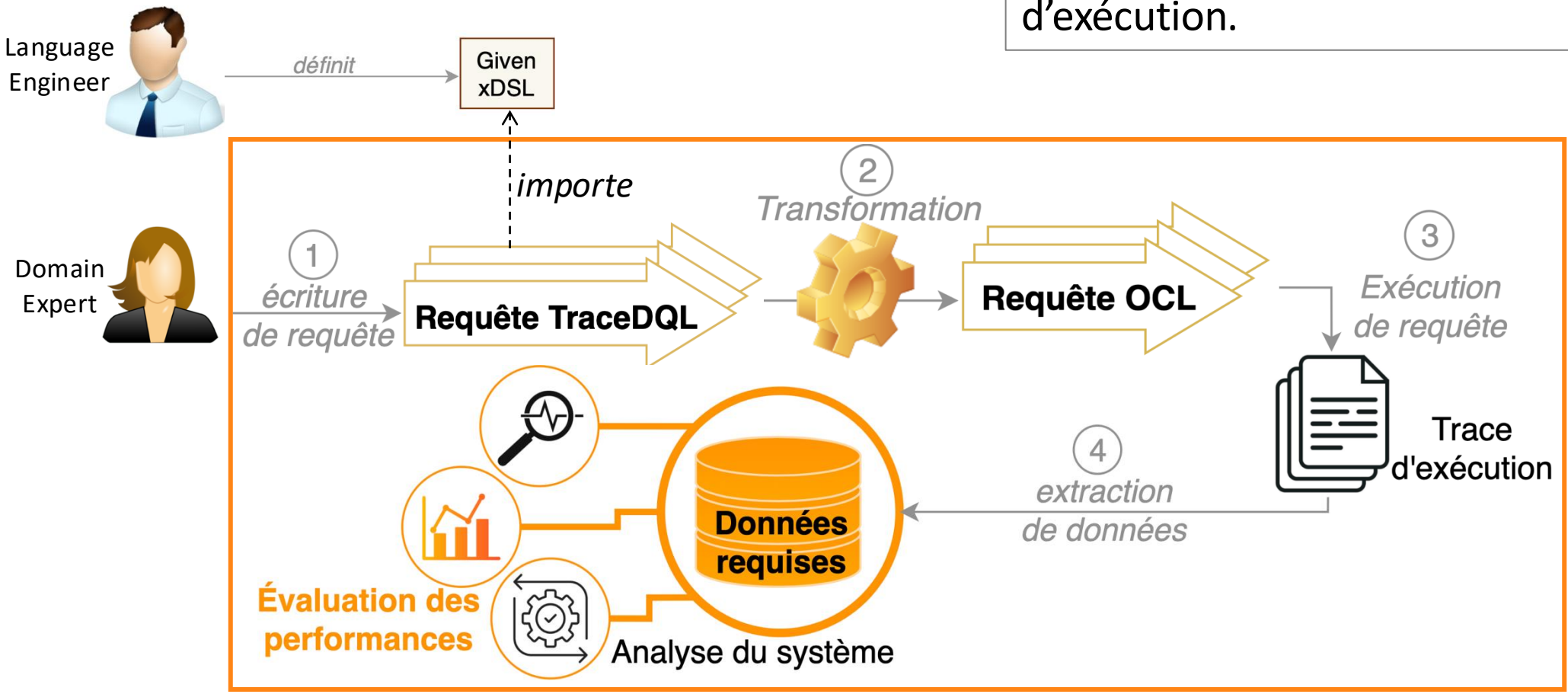
TraceDQL : vue d'ensemble du **processus d'application** (2/5)

Un langage **générique** et **exécutable** qui permet aux **experts du domaine** d'écrire des **requêtes** en utilisant la **terminologie du domaine** afin d'extraire les **données** de la **trace** d'exécution.



TraceDQL : vue d'ensemble du **processus d'application** (2/5)

Un langage **générique** et **exécutable** qui permet aux **experts du domaine** d'écrire des **requêtes** en utilisant la **terminologie du domaine** afin d'extraire les **données** de la **trace** d'exécution.



TraceDQL : exemple d'extraction de données à partir de trace de SMS xDSL (3/5)

L'extraction de cette donnée nécessite de :

a. **Récupérer** les produits à partir de chaque état enregistré.

b. **Filtrer** les produits de tête de marteau.

c. **Supprimer** les doublons.

d. **Compter** les produits de tête de marteau restants.

Experte industrielle

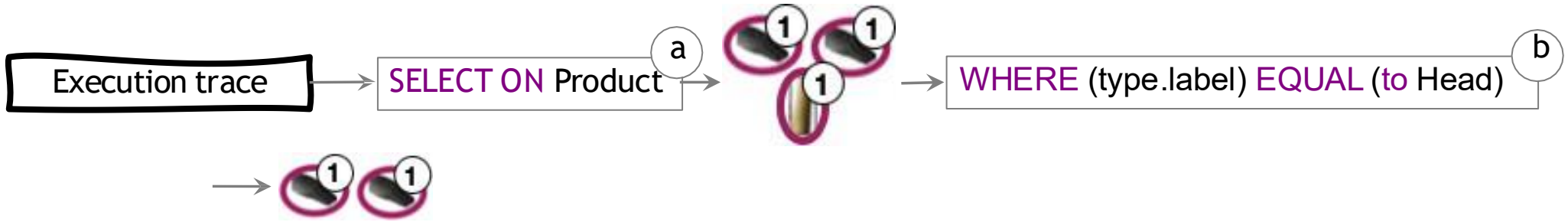
➢ TraceDQL fournit **des instructions** et **des opérateurs** permettant d'extraire cette donnée en utilisant la **terminologie du SMS xDSL**.

TraceDQL : exemple d’extraction de données à partir de trace (3/5)

Instructions TraceDQL pour extraire la donnée demandée.

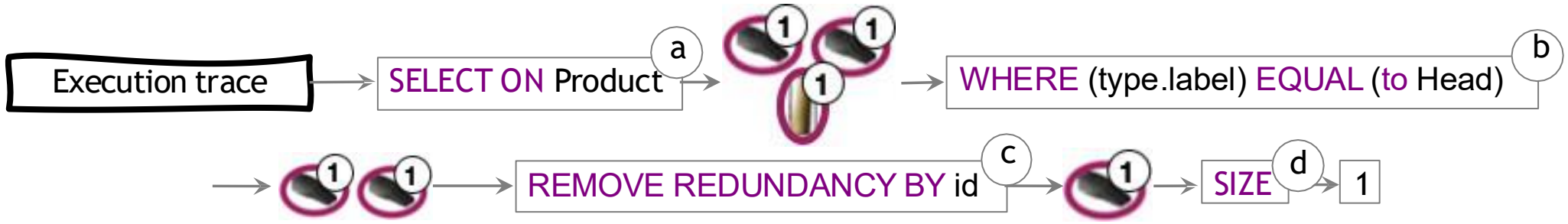
TraceDQL : exemple d'extraction de données à partir de trace (3/5)

Instructions TraceDQL pour extraire la donnée demandée.



TraceDQL : exemple d'extraction de données à partir de trace (3/5)

Instructions TraceDQL pour extraire la donnée demandée.



TraceDQL : exemple d'extraction de données à partir de trace (3/5)

Instructions TraceDQL pour extraire la donnée demandée.

The entire TraceDQL query/model to execute

```
1  import sms.runtime.*

2  Block countProducts ( paraType ) {
3    query query_COUNT {
4          SELECT SIZE ON Product ,
5          WHERE  (type.label) EQUAL ( value paraType)
6          REMOVE REDUNDANCY BY id
7    }
8  }
9  Data ProducedProducts ( paraType :: Head ) = countProducts;
```

TraceDQL key words — SMS xDSL terminology
Hammer SMS model elements — Identifiers

TraceDQL : génération de requêtes OCL (4/5)

Navigation verticale (au sein des états)

```
#OCL query to navigate over the TraceStructure.
self.states
->selectByKind(simple::RuntimeState)
.runtimeExtensions
->selectByKind(simple::
    RuntimeExtensionOfStaticElement)
.runtimeBindings
->selectByKind(simple::RuntimeObjectValueBinding)
.runtimeValue
->selectByKind(simple::RuntimeContainmentValue)
.runtimeObject
```

Requête OCL (1)

```
#OCL query to navigate over the ModelState.
->selectByKind(runtime::SimulationState)
.moduleState
->selectByKind(runtime::ContainerState)
.products
->selectByKind(runtime::Product)
->select(product|product.type.name="Head")
```

Requête OCL (2)

Navigation horizontale (entre les états)

```
#OCL query to remove redundancy of products by id
c->iterate(p;product:Sequence(runtime::Product)=c
   | let idList:Sequence(Integer) = product
   ->iterate(pr;acc:Sequence(Integer)=Sequence{}
     | acc->including(pr.id))
   in if(idList->count(p.id)>1)
      then product->excluding(p)
      else product
      endif )
->size()
```

Requête OCL (3),
où c = Requête OCL (1) + Requête OCL (2)

TraceDQL : réutilisation (5/5)

Réutilisation des résultats des requêtes

```
1  import sms.runtime.*

2  Block subtrace ( timestamp ) {
3    query query_subtrace {
4          SELECT ON SimulationState ,
5          WHERE  (currentTime) LESS_THAN ( value timestamp)
6    }
7  }
8  Block countProducts ( timestamp, paraType ) {
9     query query_COUNT {
10         FROM DATA subtrace[timestamp] ,
11         SELECT SIZE ON Product ,
12         WHERE  (type.label) EQUAL ( value paraType)
13         REMOVE REDUNDANCY BY id
14    }
15 }
16 TransitData subtrace ( ) = subtrace;
17 Data ProducedProducts ( timestamp: 13, paraType :: Head ) = countProducts;
```

TraceDQL key words | SMS xDSL terminology | Hammer SMS model elements | Identifiers

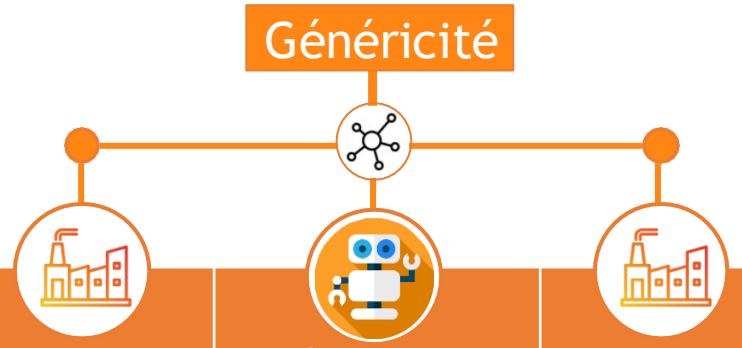
La sortie de la requête ***subtrace*** est réutilisée dans la requête ***countProducts***.

Utilisation de paramètres

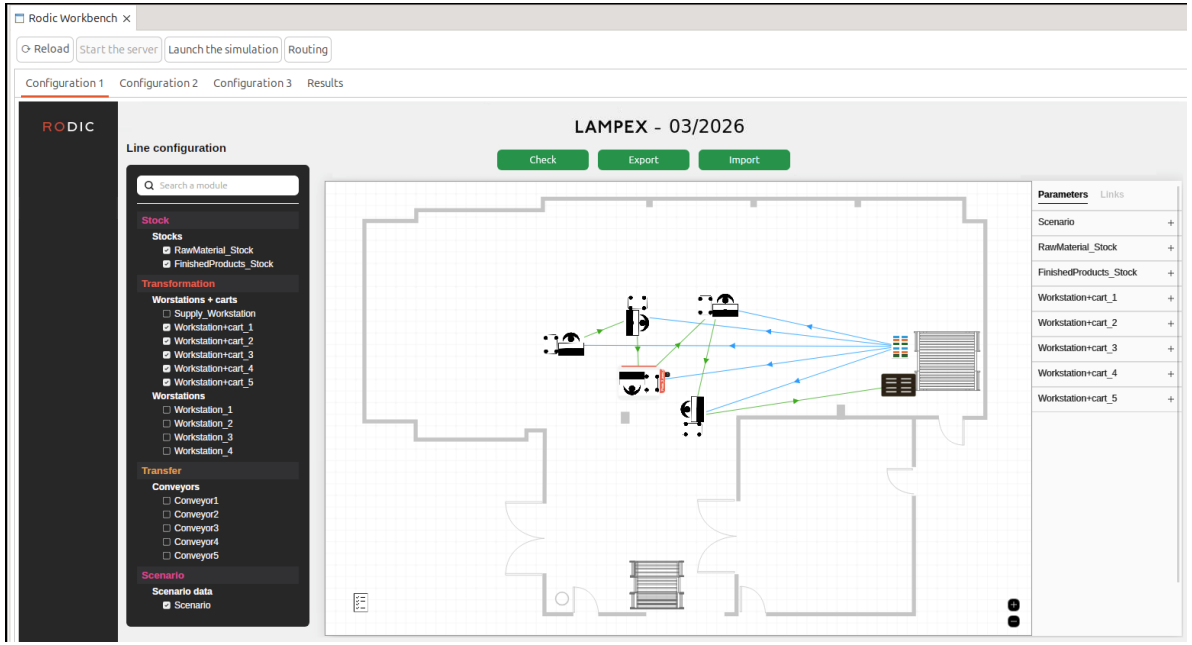
```
Data ProducedProducts ( paraType :: Hammer ) = countProducts;
```

Évaluation de l’approche autour de TraceDQL

La généricité de TraceDQL est expérimentée par son application réussie à trois xDSL différents.



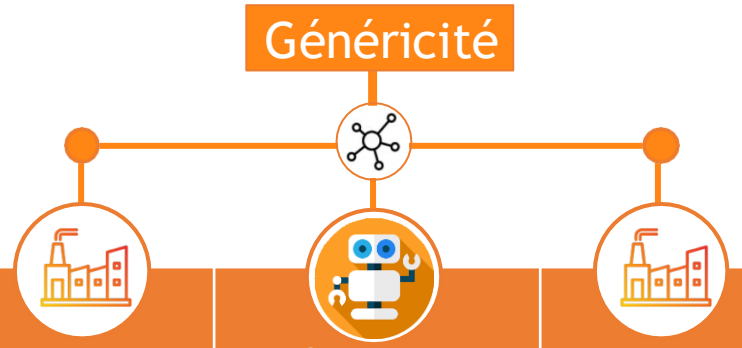
Cas d’étude	SMS xDSL	Arduino xDSL	RMS xDSL
Développeur	Nous-même	Arduino Designer	Projet RODIC
Taille du méta modèle	16 méta classes	59 méta classes	32 méta classes
Modèle	SimpleCase	Blinking LED	Lampex
Taille du modèle	30 instances	18 instances	238 instances



Système Lampex modélisé sur RODIC Studio

Évaluation de l’approche autour de TraceDQL

La généricité de TraceDQL est expérimentée par son application réussie à trois xDSL différents.



Cas d’étude	SMS xDSL	Arduino xDSL	RMS xDSL
Développeur	Nous-même	Arduino Designer	Projet RODIC
Taille du méta modèle	16 méta classes	59 méta classes	32 méta classes
Modèle	SimpleCase	Blinking LED	Lampex
Taille du modèle	30 instances	18 instances	238 instances

TraceDQL permet la réutilisabilité de plusieurs manières.

Utilisation de paramètres

La requête paramétrée ***countProducts*** est réutilisée pour différents types de produits.

Réutilisation des requêtes

Réutilisation des résultats des requêtes

Expérimentations et évaluation de l'approche

Le temps d'exécution des requêtes TraceDQL a été testé sur deux traces d'exécution différentes.

Trace Size (in MB)	Model State Capture	Number of Objects		Single			Nested	Combined	
				S	SC	SR	(4 queries)	Q1	Q2
3.316	500	20350	t_{load}	<1s	<1s	<1s	<1s	<1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	179s	<1s	<1s	<1s
			t_{total}	<1s	<1s	179s	<1s	<1s	
33.820	5000	133975	t_{load}	1s	1s	2s	1s	1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	446s	<1s	<1s	<1s
			t_{total}	1s	1s	450s	1s	<1s	

Expérimentations et évaluation de l'approche

Le temps d'exécution des requêtes TraceDQL a été testé sur deux traces d'exécution différentes.

Trace Size (in MB)	Model State Capture	Number of Objects		Single			Nested	Combined	
				S	SC	SR	(4 queries)	Q1	Q2
3.316	500	20350	t_{load}	<1s	<1s	<1s	<1s	<1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	179s	<1s	<1s	<1s
			t_{total}	<1s	<1s	179s	<1s	<1s	
33.82	5000	133975	t_{load}	1s	1s	2s	1s	1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	446s	<1s	<1s	<1s
			t_{total}	1s	1s	450s	1s	<1s	

Expérimentations et évaluation de l'approche

Le temps d'exécution des requêtes TraceDQL a été testé sur deux traces d'exécution différentes.

Trace Size (in MB)	Model State Capture	Number of Objects		Single			Nested	Combined	
				S	SC	SR	(4 queries)	Q1	Q2
3.316	500	20350	t_{load}	<1s	<1s	<1s	<1s	<1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	179s	<1s	<1s	<1s
			t_{total}	<1s	<1s	179s	<1s	<1s	
33.820	5000	133975	t_{load}	1s	1s	2s	1s	1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	446s	<1s	<1s	<1s
			t_{total}	1s	1s	450s	1s	<1s	

Expérimentations et évaluation de l'approche

Le temps d'exécution des requêtes TraceDQL a été testé sur deux traces d'exécution différentes.

Trace Size (in MB)	Model State Capture	Number of Objects		Single			Nested	Combined	
				S	SC	SR	(4 queries)	Q1	Q2
3.316	500	20350	t_{load}	<1s	<1s	<1s	<1s	<1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	179s	<1s	<1s	<1s
			t_{total}	<1s	<1s	179s	<1s	<1s	
33.820	5000	133975	t_{load}	1s	1s	2s	1s	1s	
			$t_{transform}$	<1s	<1s	<1s	<1s	<1s	<1s
			t_{exec}	<1s	<1s	446s	<1s	<1s	<1s
			t_{total}	1s	1s	450s	1s	<1s	

Plan de l’exposé

Introduction

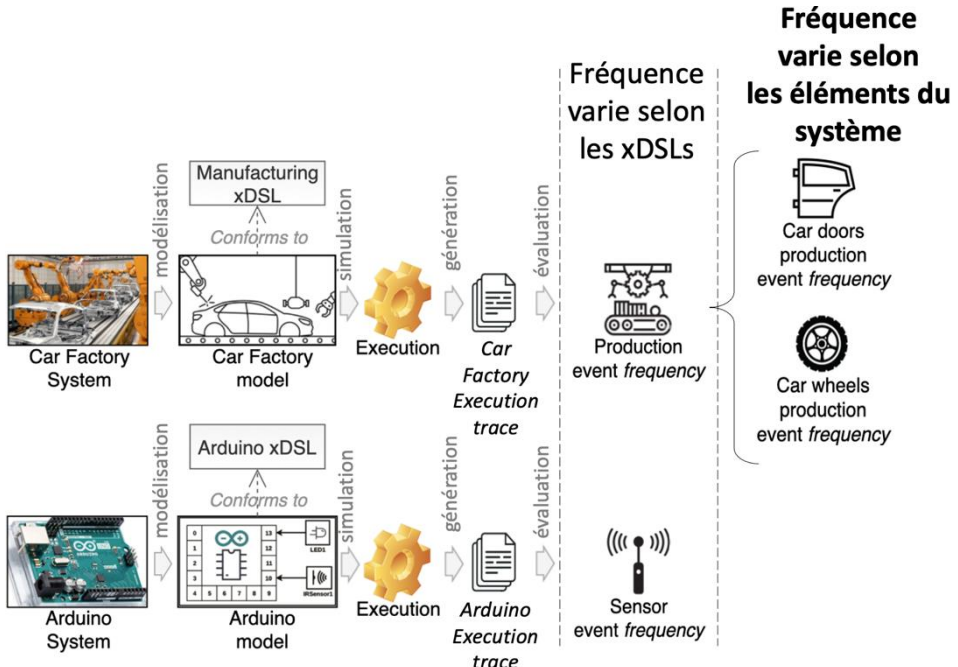
Problématique générale

Principaux concepts

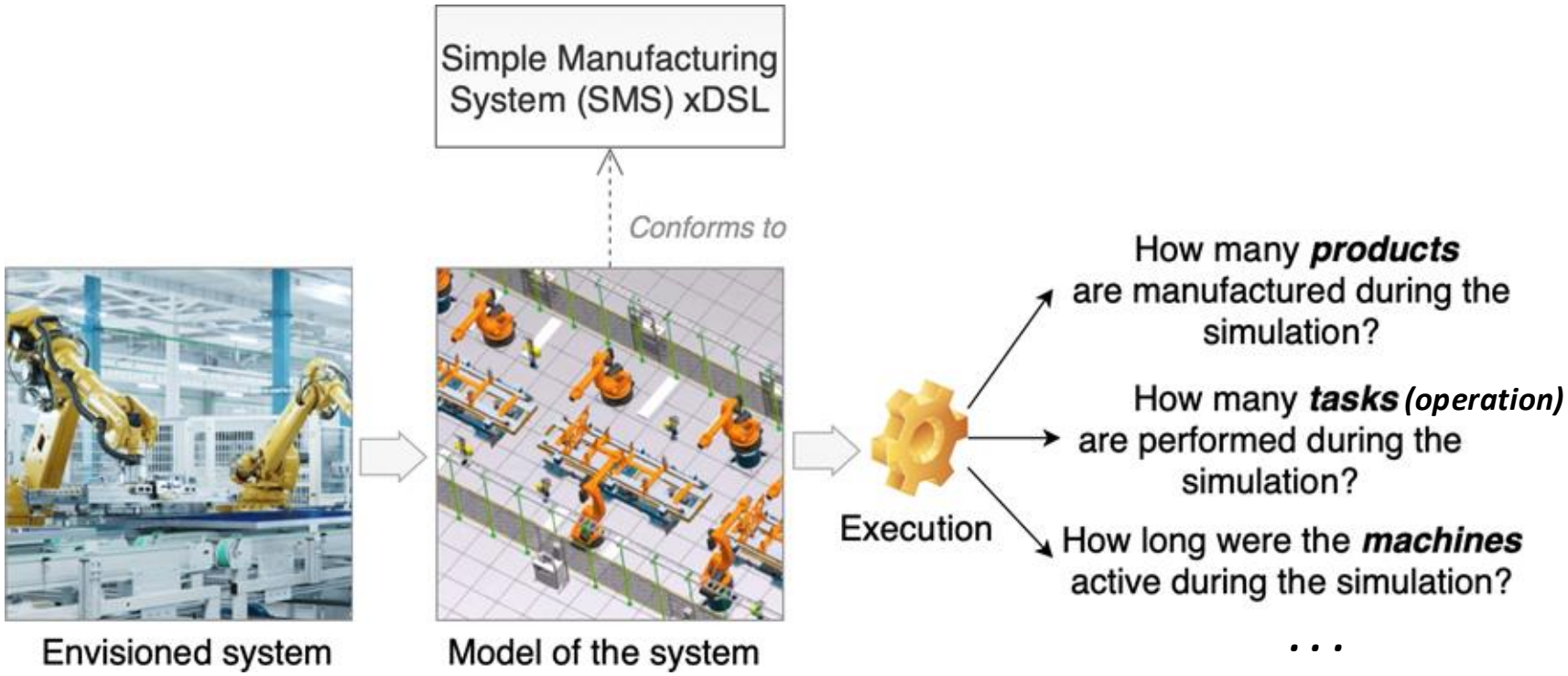
Contribution 2 : Gestion de l’extraction de données au niveau language et inter-domaines pour l’analyse des systèmes.

Contribution 3 : Gestion des KPIs multi-niveaux et inter-domaines.

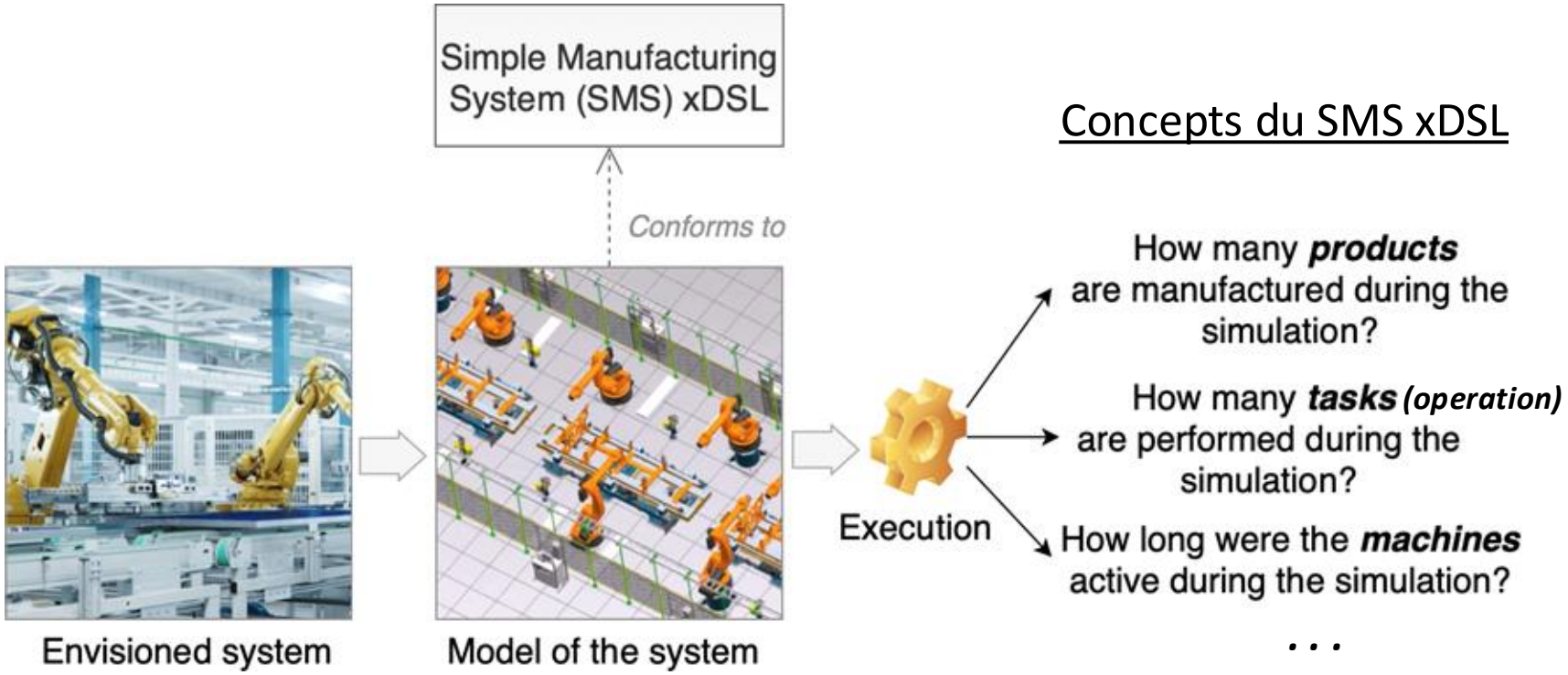
Conclusion



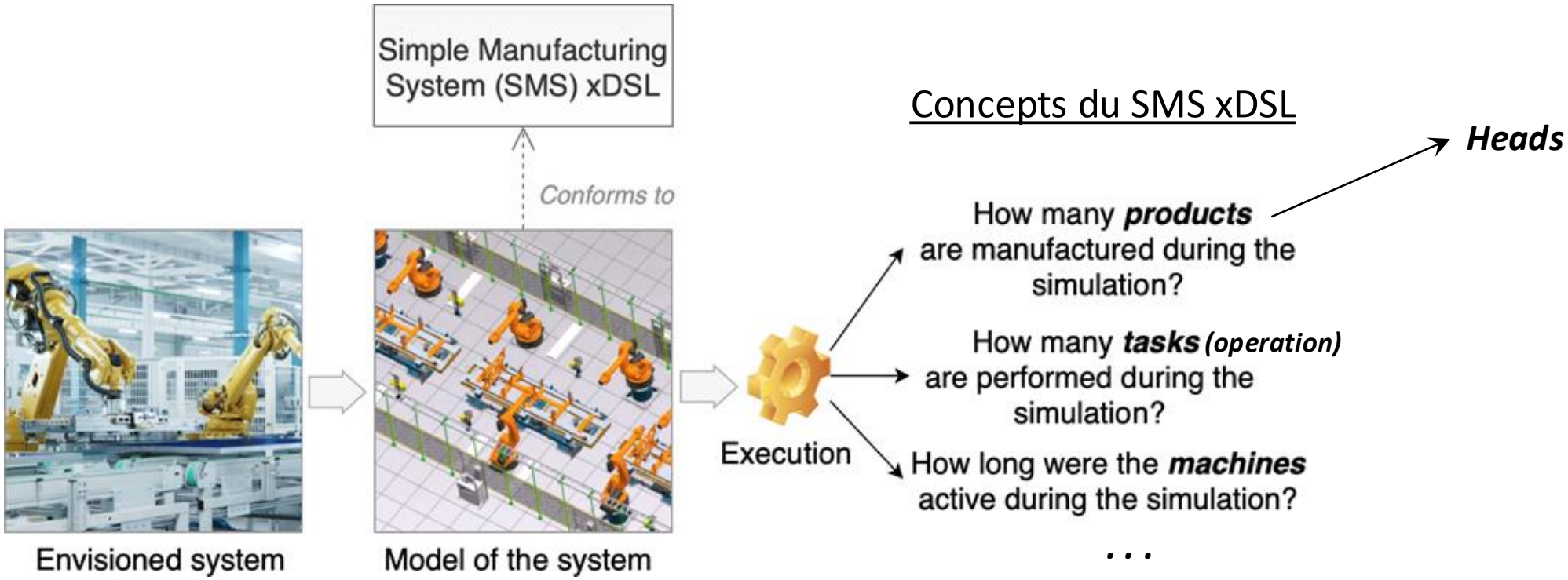
Besoins pour mesurer un KPI



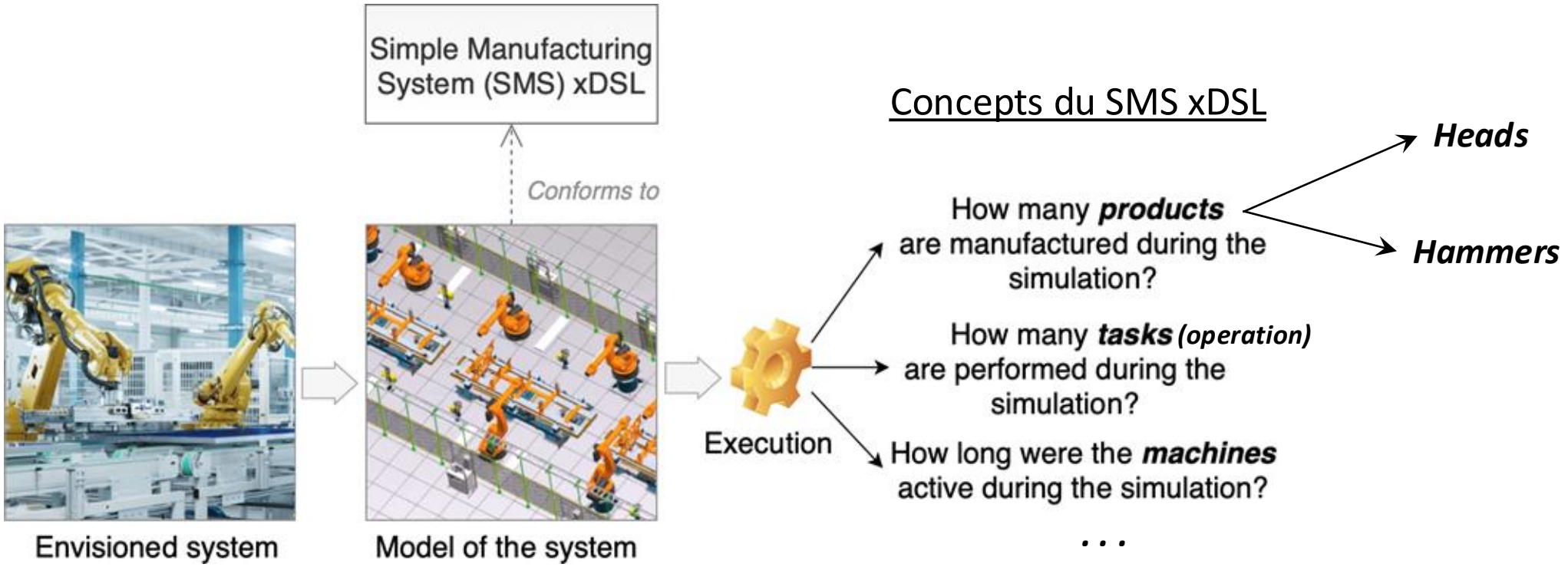
Besoins pour mesurer un KPI



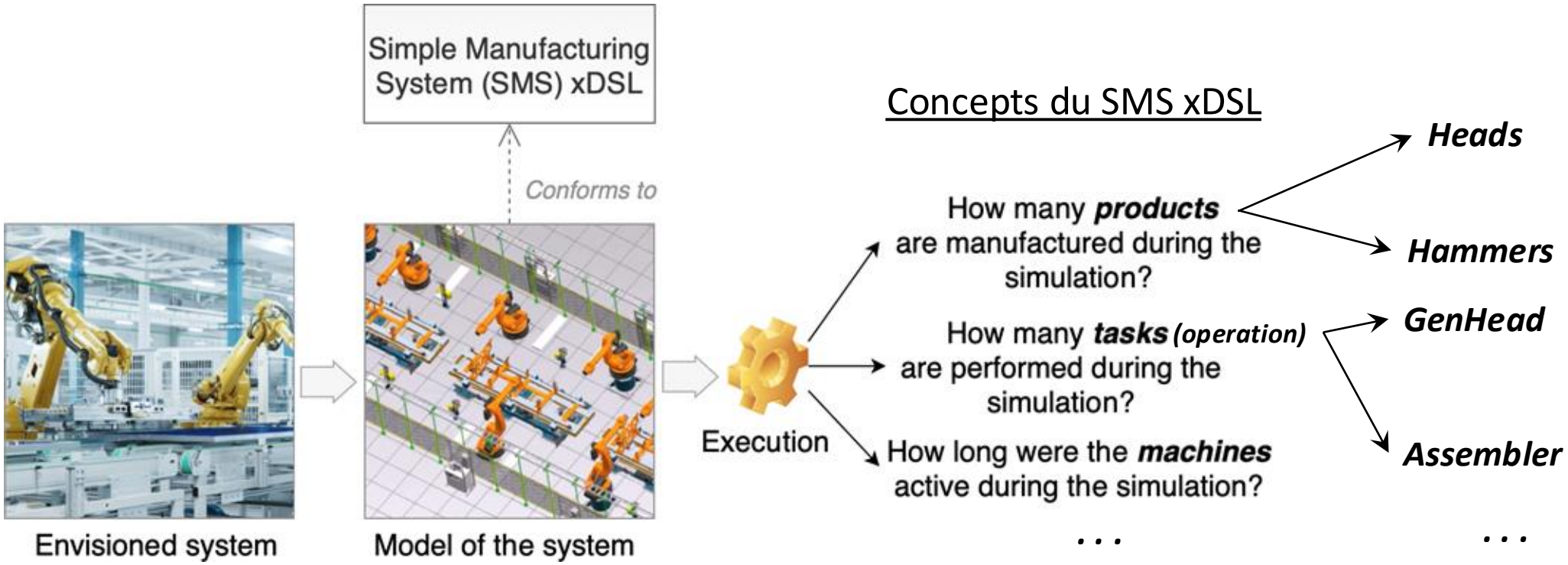
Besoins pour mesurer un KPI



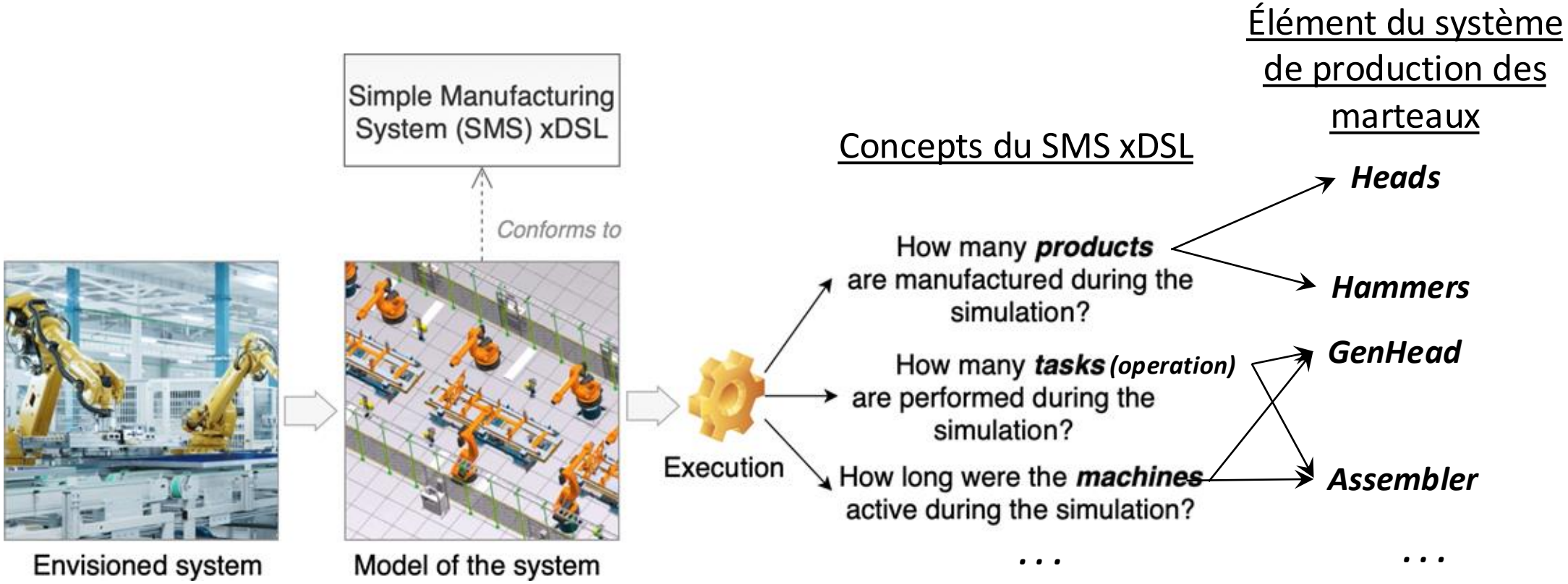
Besoins pour mesurer un KPI



Besoins pour mesurer un KPI



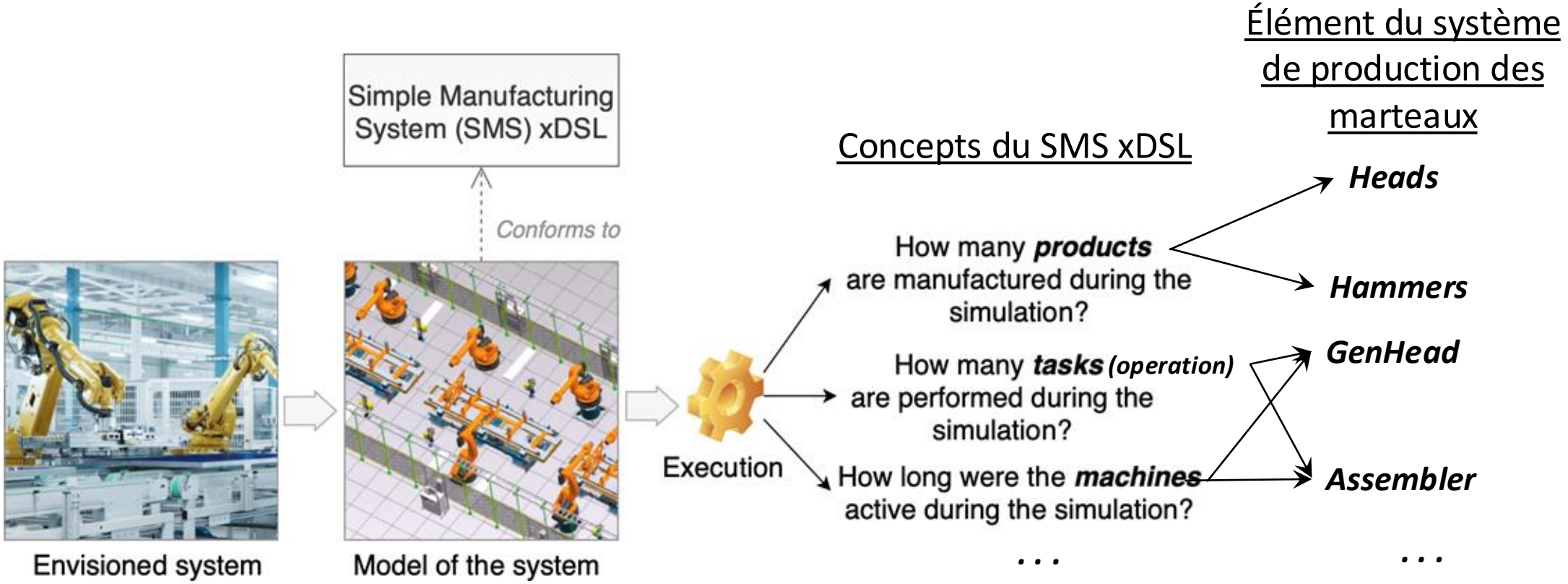
Besoins pour mesurer un KPI



Besoins pour mesurer un KPI

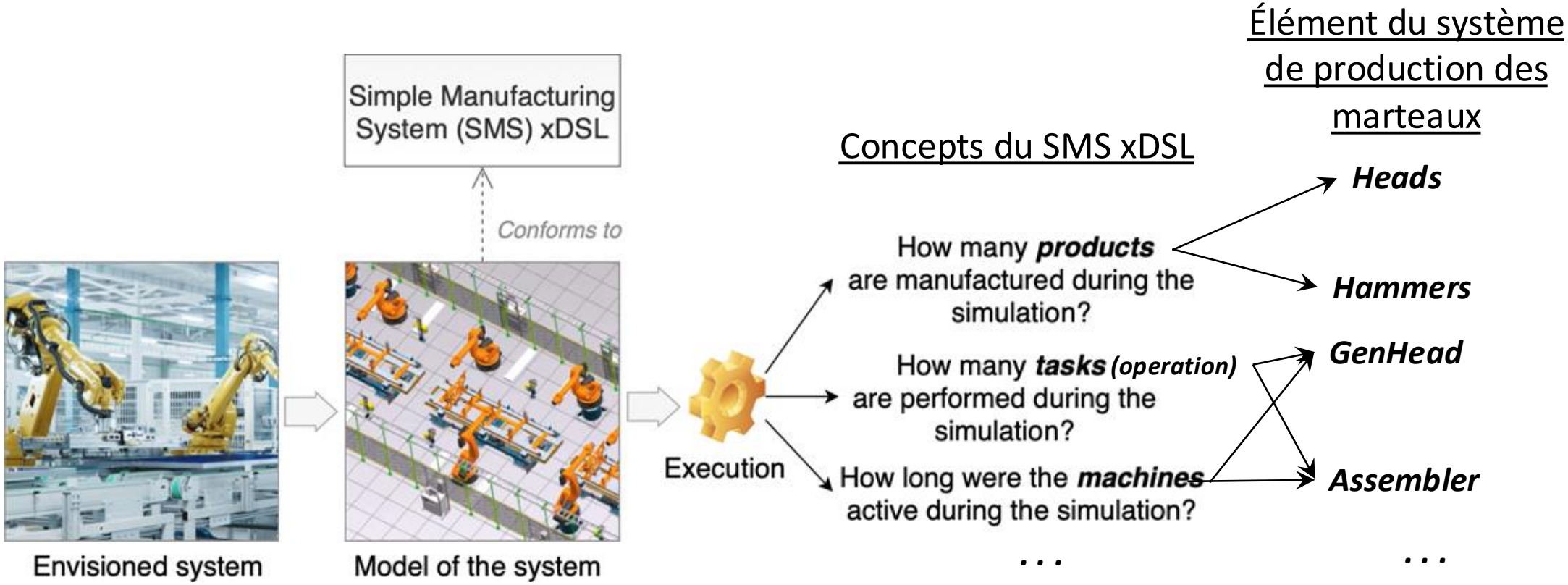
La mesure d'un KPI nécessite :

1. La spécification du langage.
2. La spécification du système.



Besoins pour mesurer un KPI

La mesure d'un KPI nécessite :
1. La spécification du langage.
2. La spécification du système.



➢ **La gestion de la variabilité** des KPI est une tâche difficile.

Décomposition de l’évaluation de performance pour la gestion de la variabilité des KPIs

Trois aspects des KPIs sont distingués pour gérer la variabilité :

Calcul des KPIs

Configuration des KPIs

Définition des KPIs

Problèmes de définition des KPIs (1/2)

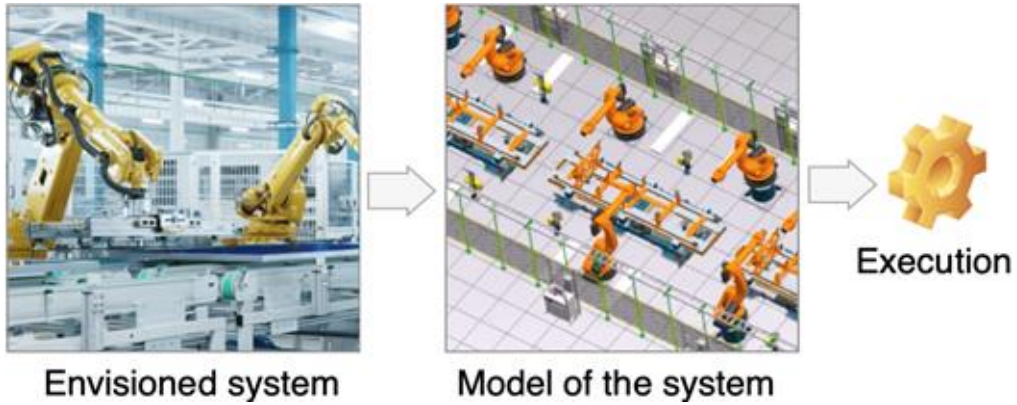
Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.

Problèmes de définition des KPIs (1/2)

Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.



Problèmes de définition des KPIs (1/2)

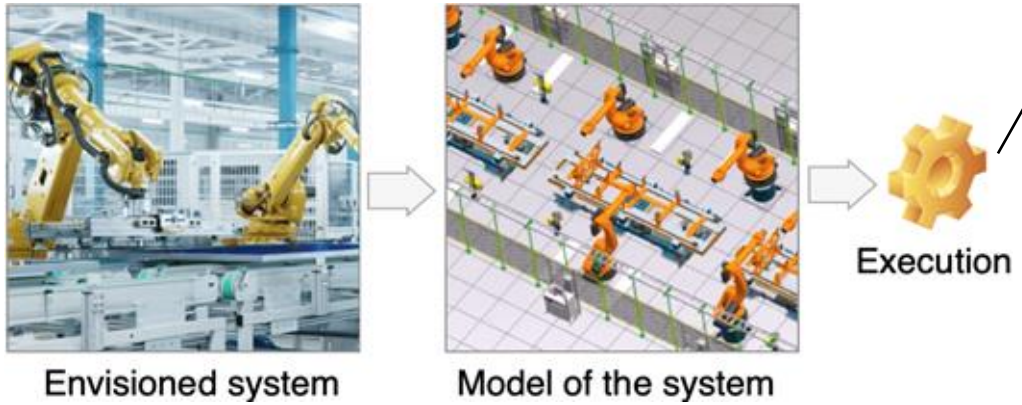
Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.

Implémentation :

*KPI1 = nombre de **heads** / duration*

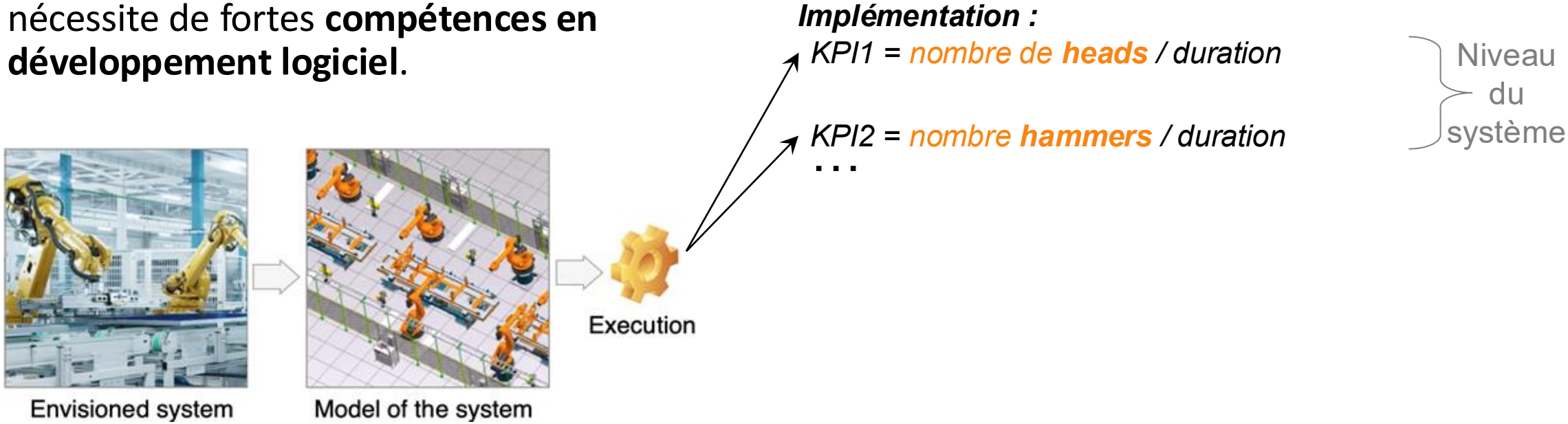
Niveau du système



Problèmes de définition des KPIs (1/2)

Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

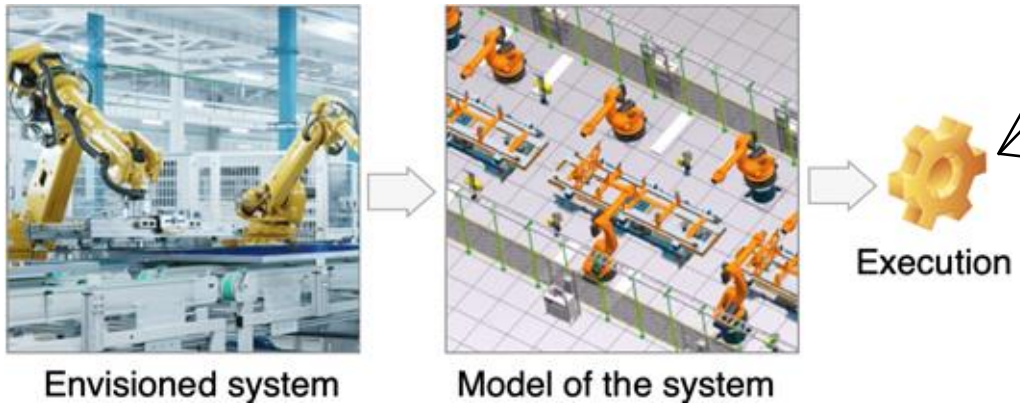
- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.



Problèmes de définition des KPIs (1/2)

Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.



Implémentation :
*KPI1 = nombre de **heads** / duration*

*KPI2 = nombre **hammers** / duration*

...

Implémentation :
*KPI3 = nombre de **produits** / duration*

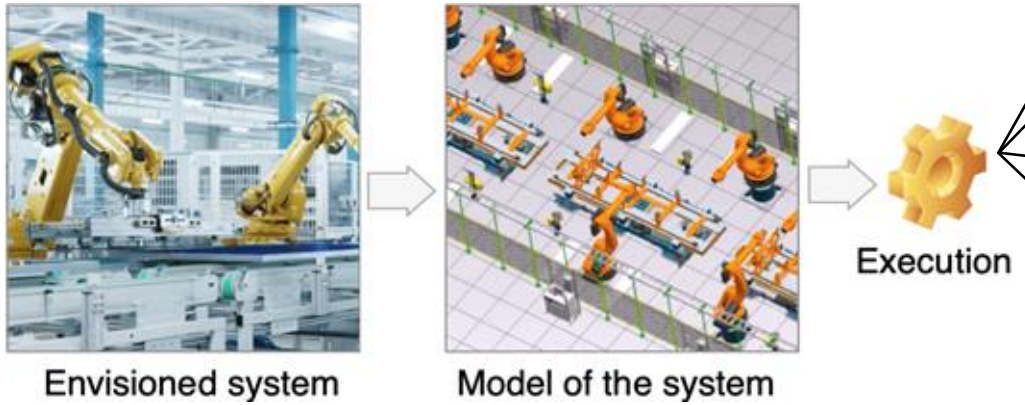
Niveau du système

Niveau du langage

Problèmes de définition des KPIs (1/2)

Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.



Implémentation :

*KPI1 = nombre de **heads** / duration*

*KPI2 = nombre **hammers** / duration*

...

Implémentation :

*KPI3 = nombre de **produits** / duration*

Ré-Implémentation :

*KPI4 = nombre de **tâche** / duration*

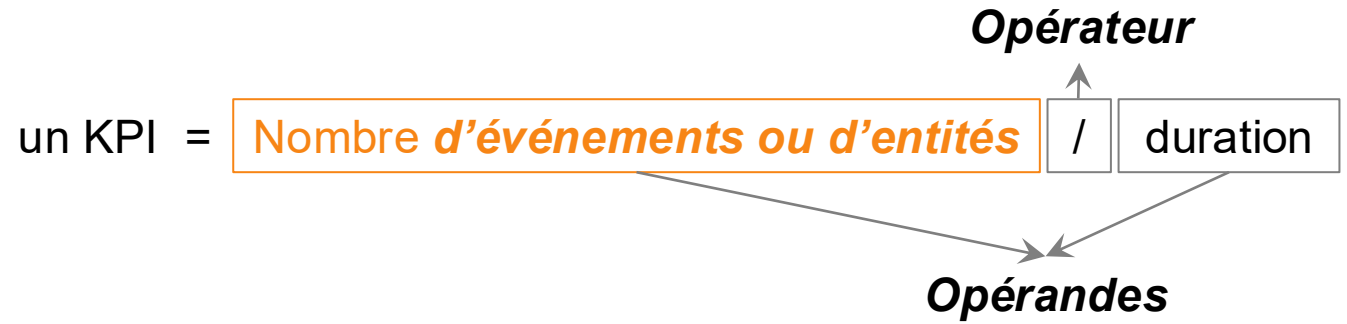
Niveau du système

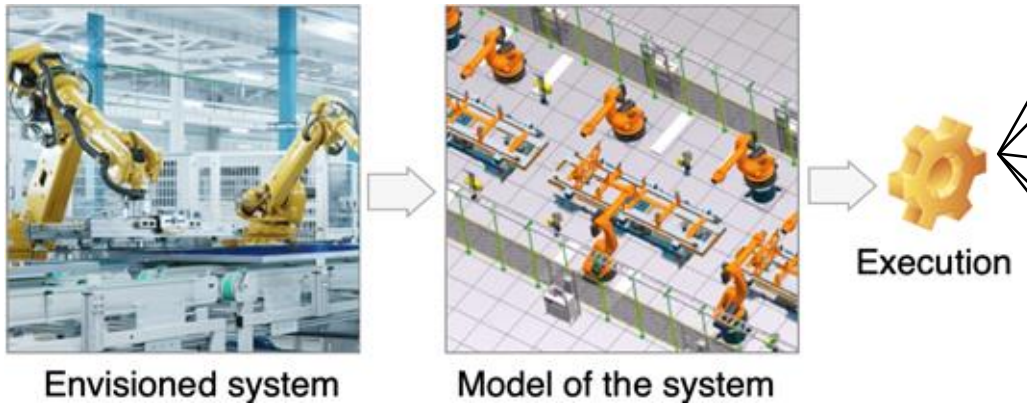
Niveau du langage

Problèmes de définition des KPIs (1/2)

Définir un KPI pour **un concept de langage** ou **un élément de système particulier** :

- conduit à une implémentation **codée en dur** et **non réutilisable,**
- nécessite de fortes **compétences en développement logiciel**.





Implémentation :

*KPI1 = nombre de **heads** / duration*

*KPI2 = nombre **hammers** / duration*

...

Niveau du système

Implémentation :

*KPI3 = nombre de **produits** / duration*

Ré-Implémentation :

*KPI4 = nombre de **tâche** / duration*

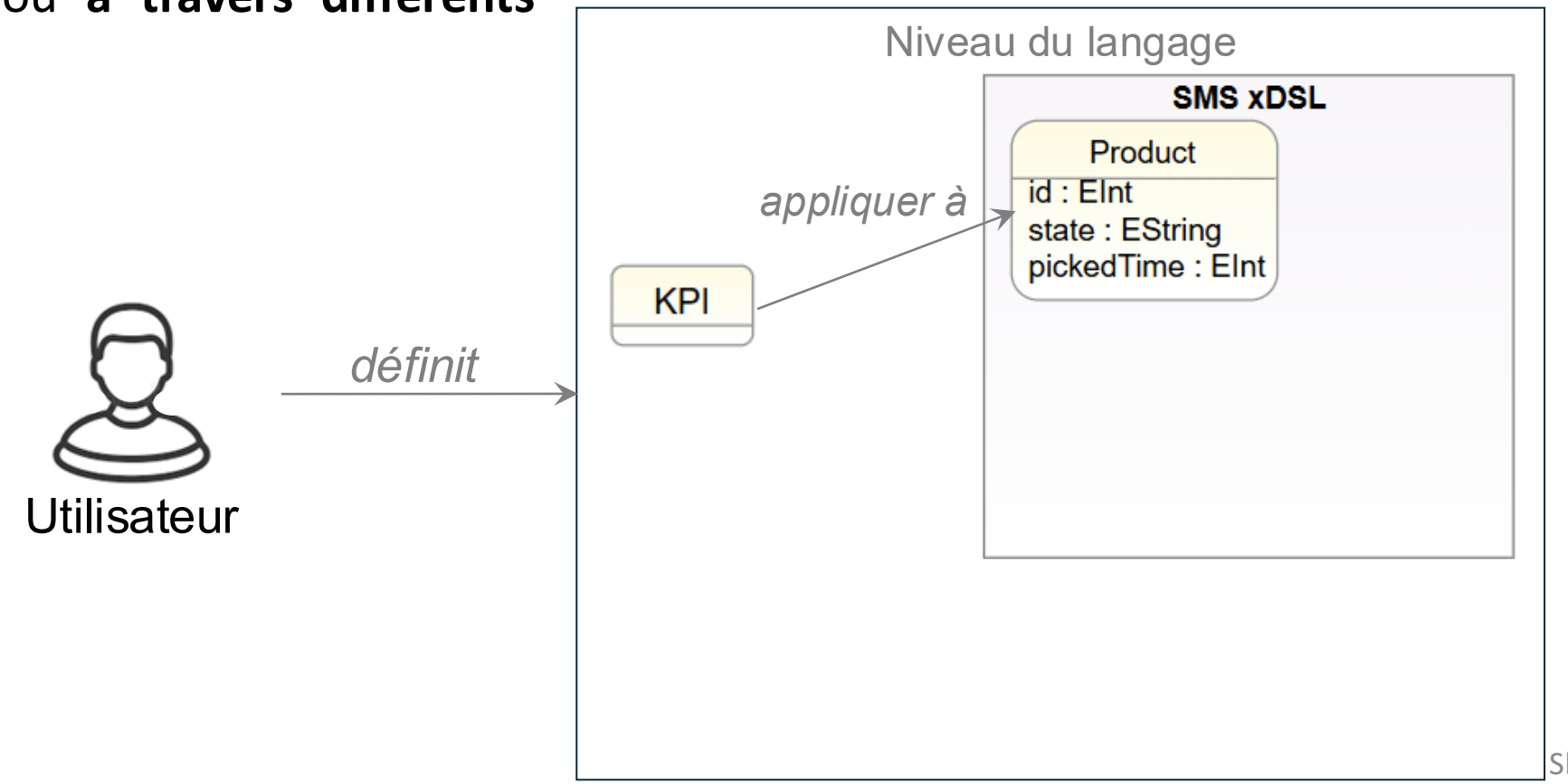
*KPI5 = nombre d'un **concept donné** / duration*

...

Niveau du langage

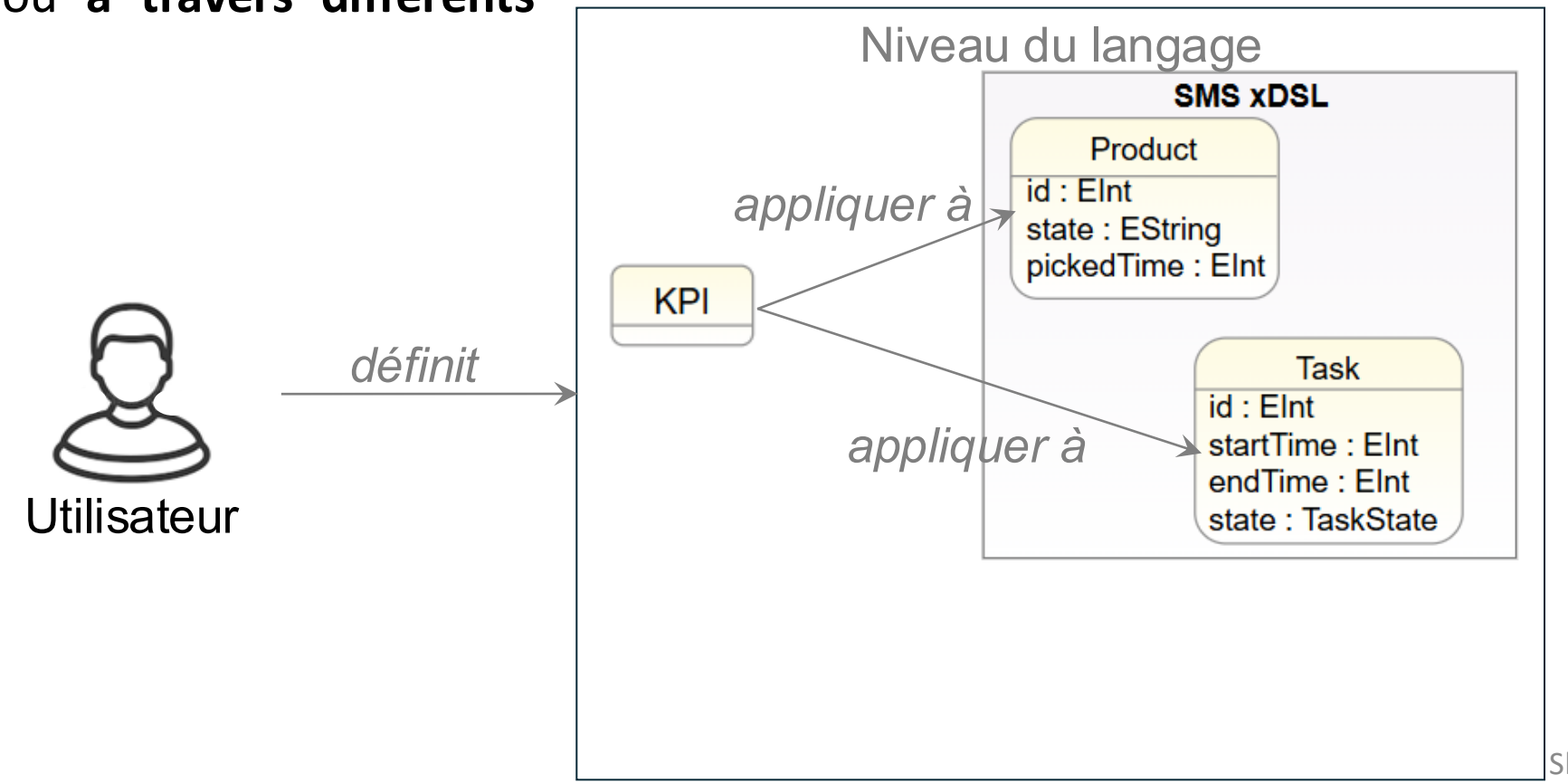
Problème de définition des KPIs (2/2)

Exigence de flexibilité : application des KPI à d'autres concepts **au sein du même xDSL** ou **à travers différents xDSL**.



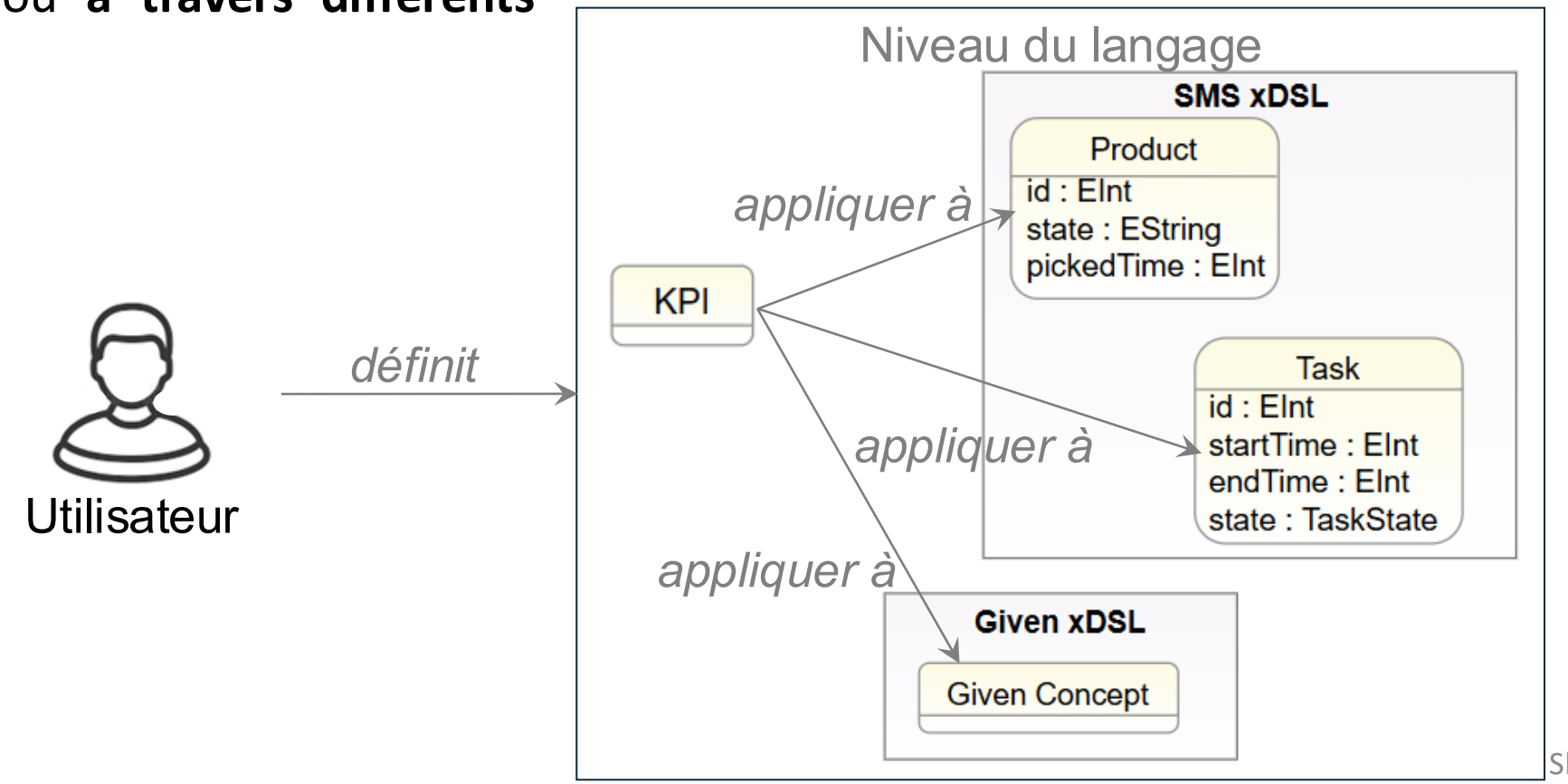
Problème de définition des KPIs (2/2)

Exigence de flexibilité : application des KPI à d'autres concepts **au sein du même xDSL** ou **à travers différents xDSL**.



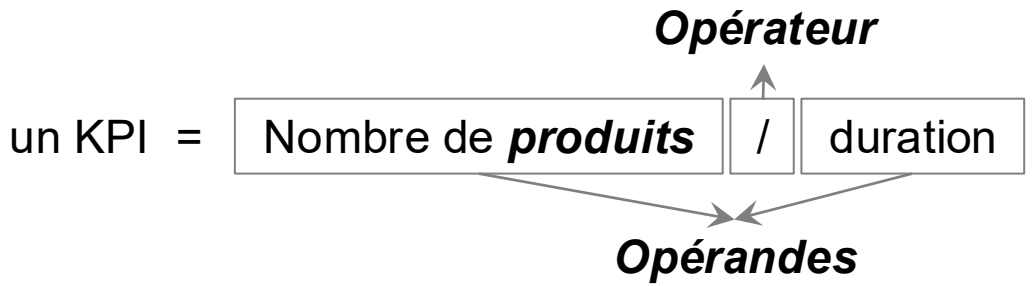
Problème de définition des KPIs (2/2)

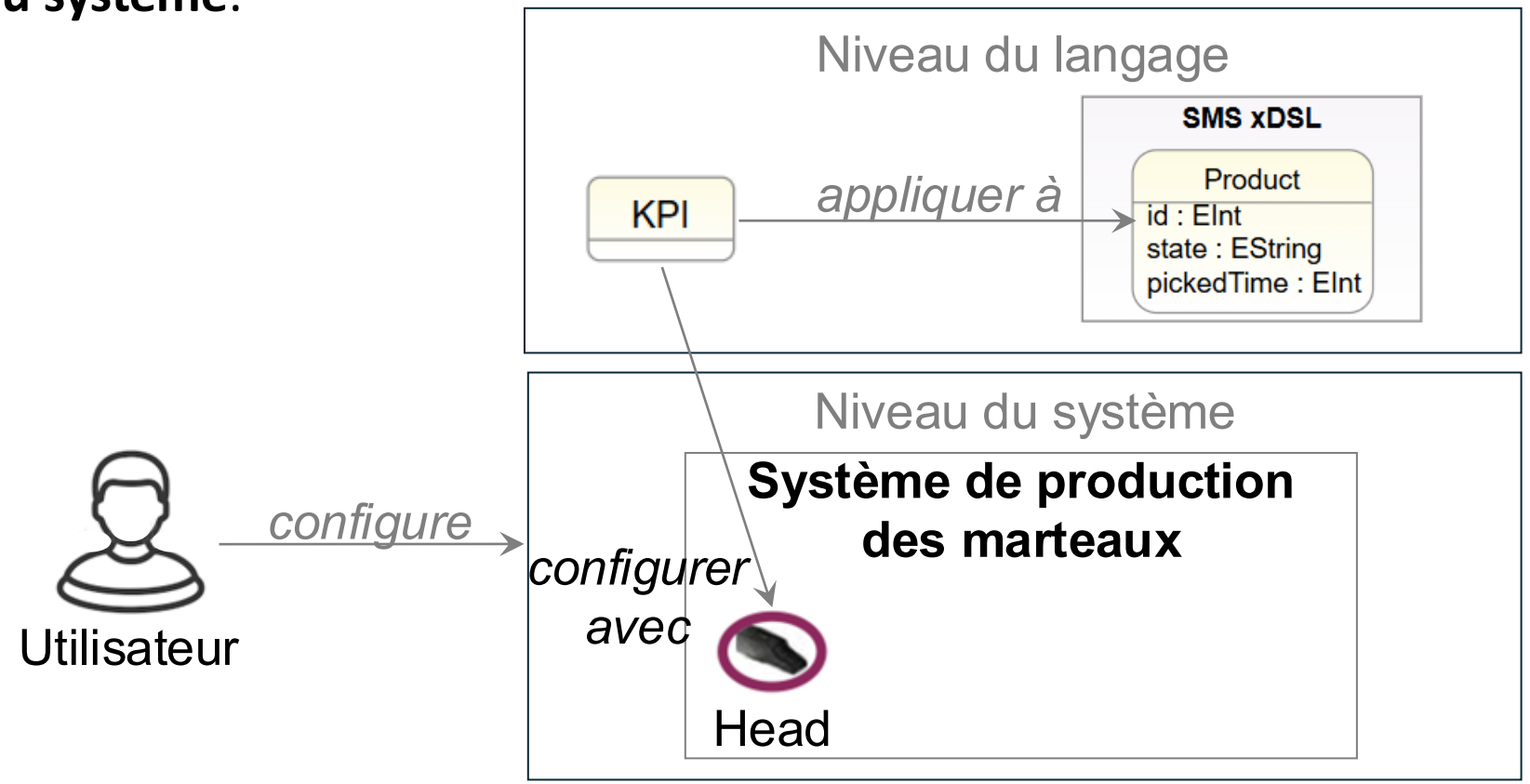
Exigence de flexibilité : application des KPI à d'autres concepts **au sein du même xDSL** ou **à travers différents xDSL**.



Problème de configuration des KPIs

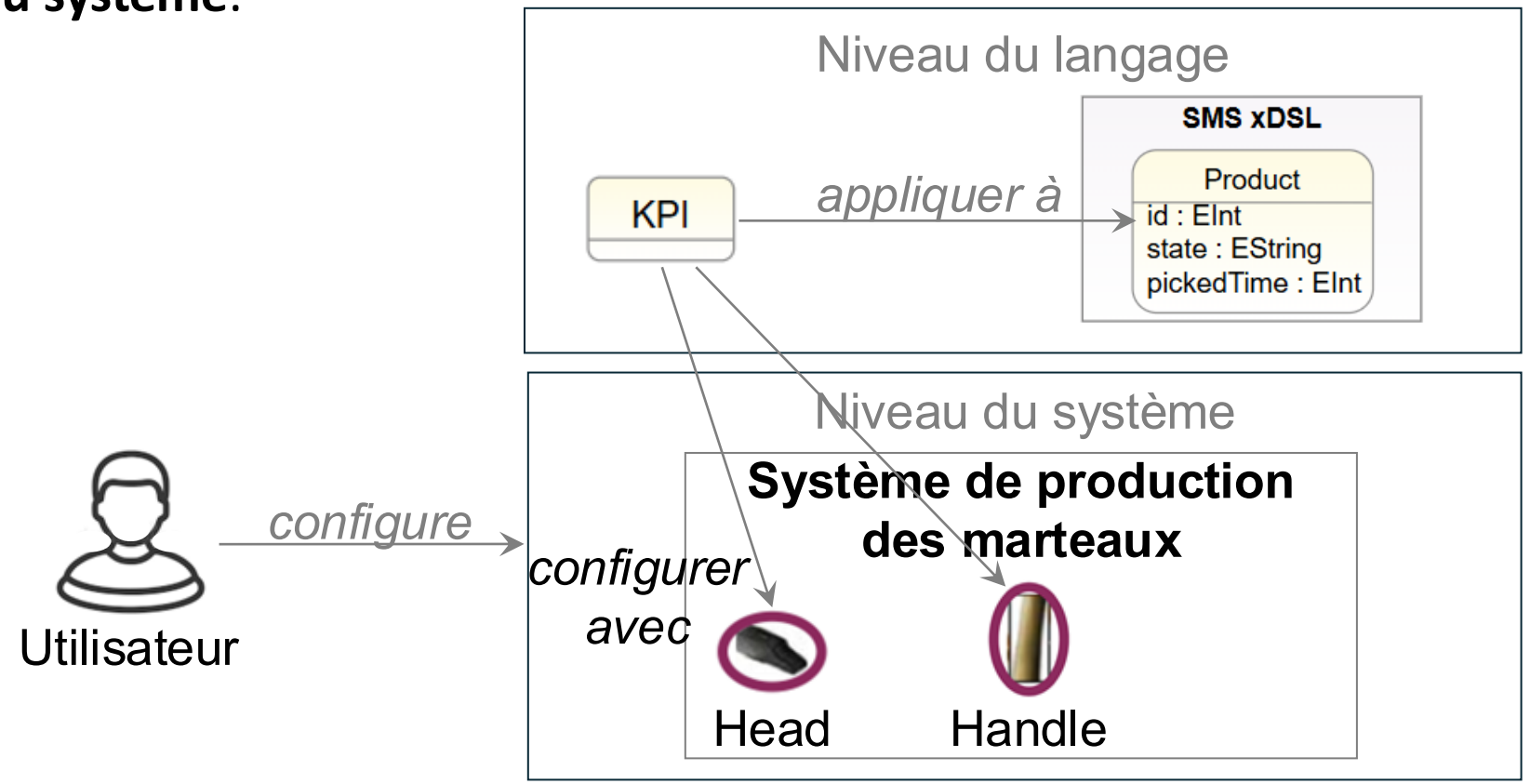
Exigence de configurabilité : configuration du KPI avec les **éléments du système**.





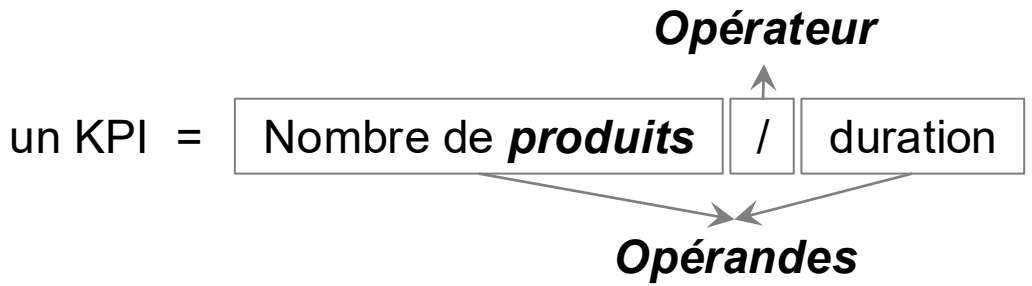
Problème de configuration des KPIs

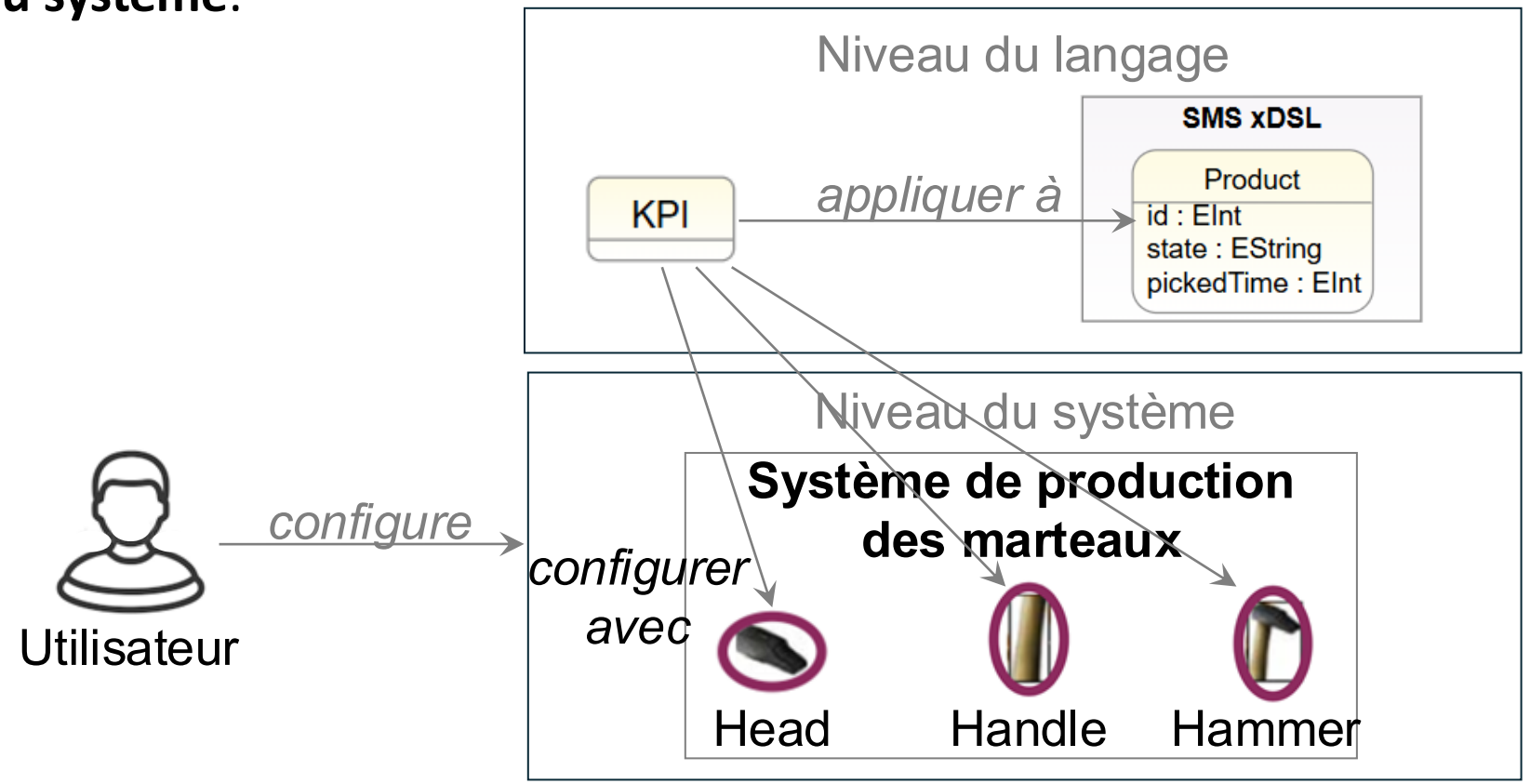
Exigence de configurabilité : configuration du KPI avec les **éléments du système**.



Problème de configuration des KPIs

Exigence de configurabilité : configuration du KPI avec les **éléments du système**.

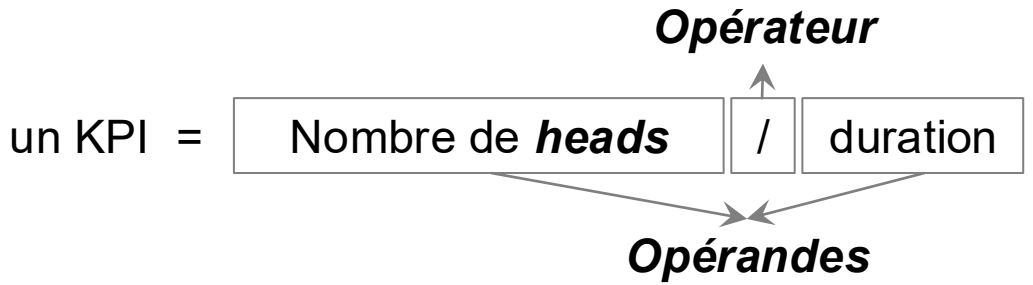


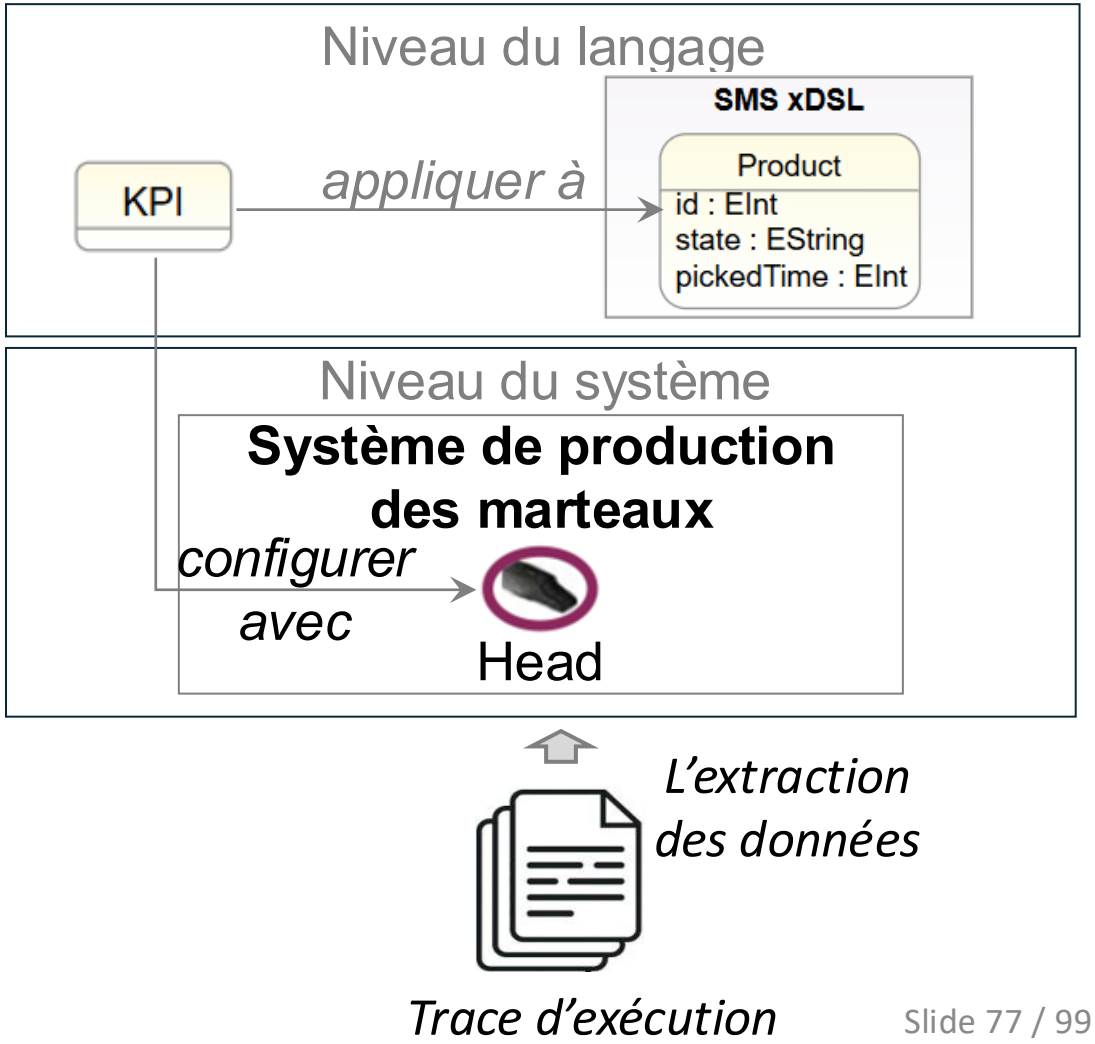


Problème de calcul des KPIs

Le calcul des KPI nécessite **l'implémentation d'un logiciel** d'extraction de données :

- lourd,
- chronophage,
- et nécessitant de solides compétences en développement logiciel.





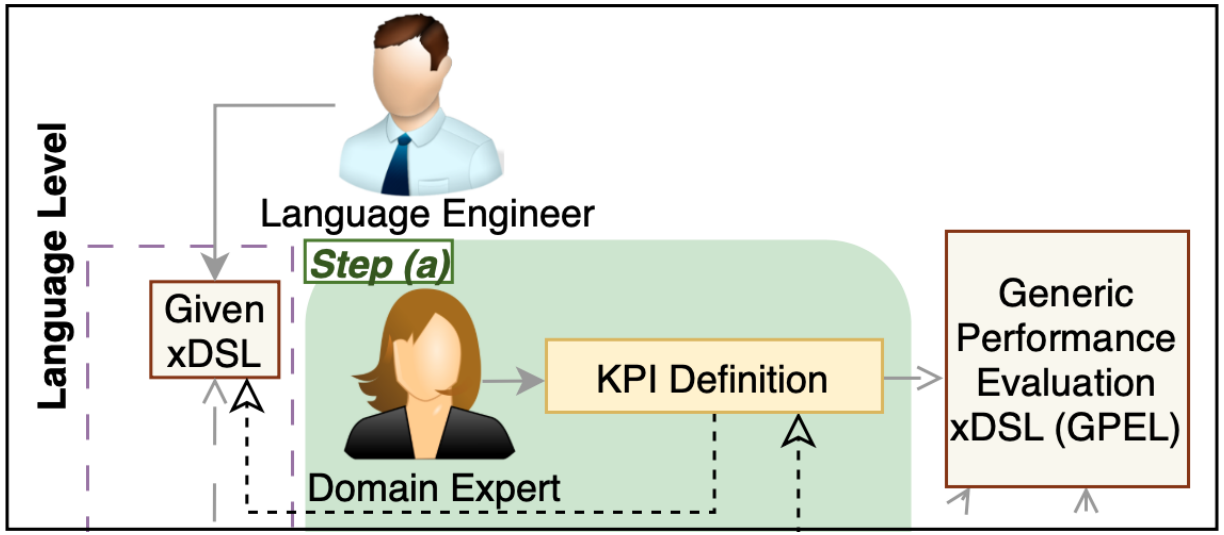
Proposition GPEL : approche générique et multi-niveaux pour l'évaluation de la performance (1/4)

Trois rôles sont impliqués :

- **Ingénieur de langage** : conçoit le xDSL concerné.
- **Expert du domaine** : édite les formules de KPI en utilisant la terminologie du domaine via une approche de correspondance.
- **Utilisateur final** : configure les KPI définis en utilisant les éléments du système.

👉 Les KPIs sont calculés grâce **à la génération et à l'exécution automatiques de requêtes *TraceDQL*** afin d'extraire les données.

Proposition GPEL : approche générique et multi-niveaux pour l'évaluation de la performance (1/4)



Trois rôles sont impliqués :

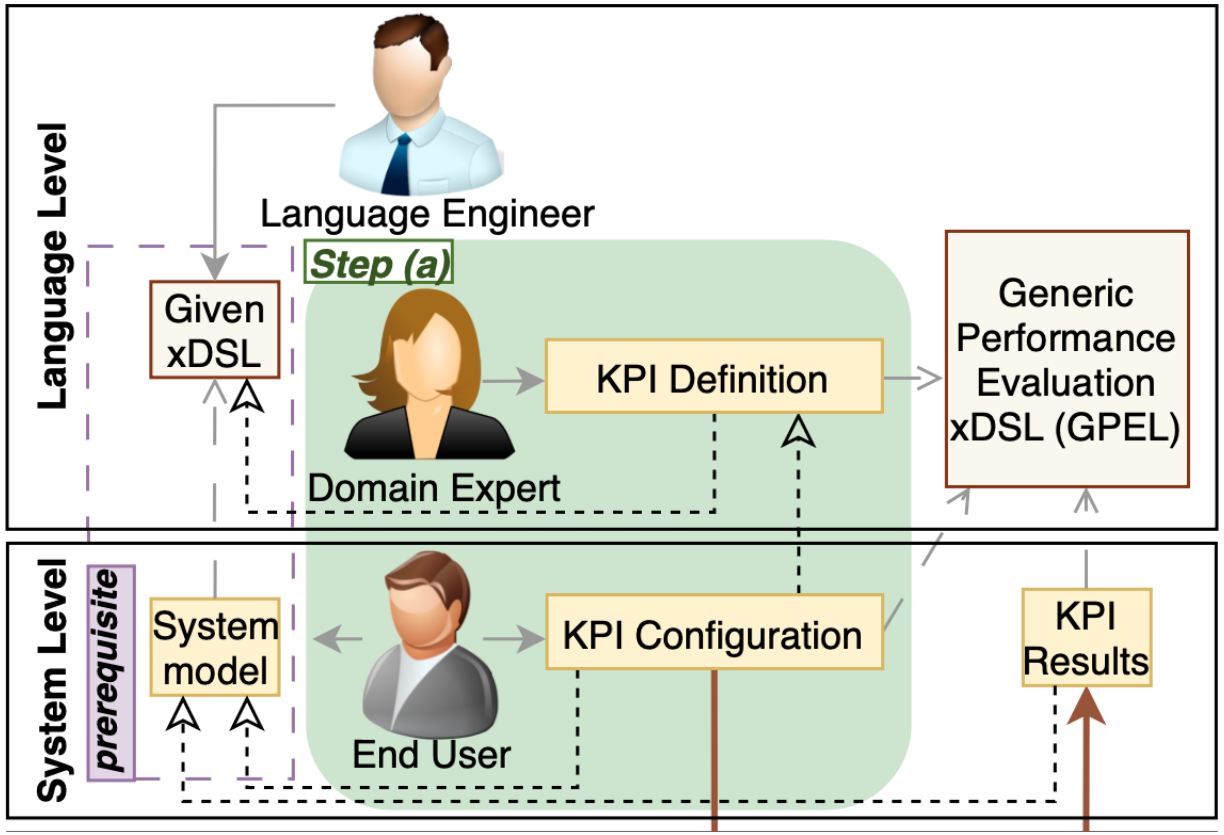
- **Ingénieur de langage** : conçoit le xDSL concerné.
- **Expert du domaine** : édite les formules de KPI en utilisant la terminologie du domaine via une approche de correspondance.
- **Utilisateur final** : configure les KPI définis en utilisant les éléments du système.

👉 Les KPIs sont calculés grâce **à la génération et à l'exécution automatiques de requêtes *TraceDQL*** afin d'extraire les données.

Proposition GPEL : approche générique et multi-niveaux pour l'évaluation de la performance (1/4)

Trois rôles sont impliqués :

- **Ingénieur de langage** : conçoit le xDSL concerné.
- **Expert du domaine** : édite les formules de KPI en utilisant la terminologie du domaine via une approche de correspondance.
- **Utilisateur final** : configure les KPI définis en utilisant les éléments du système.



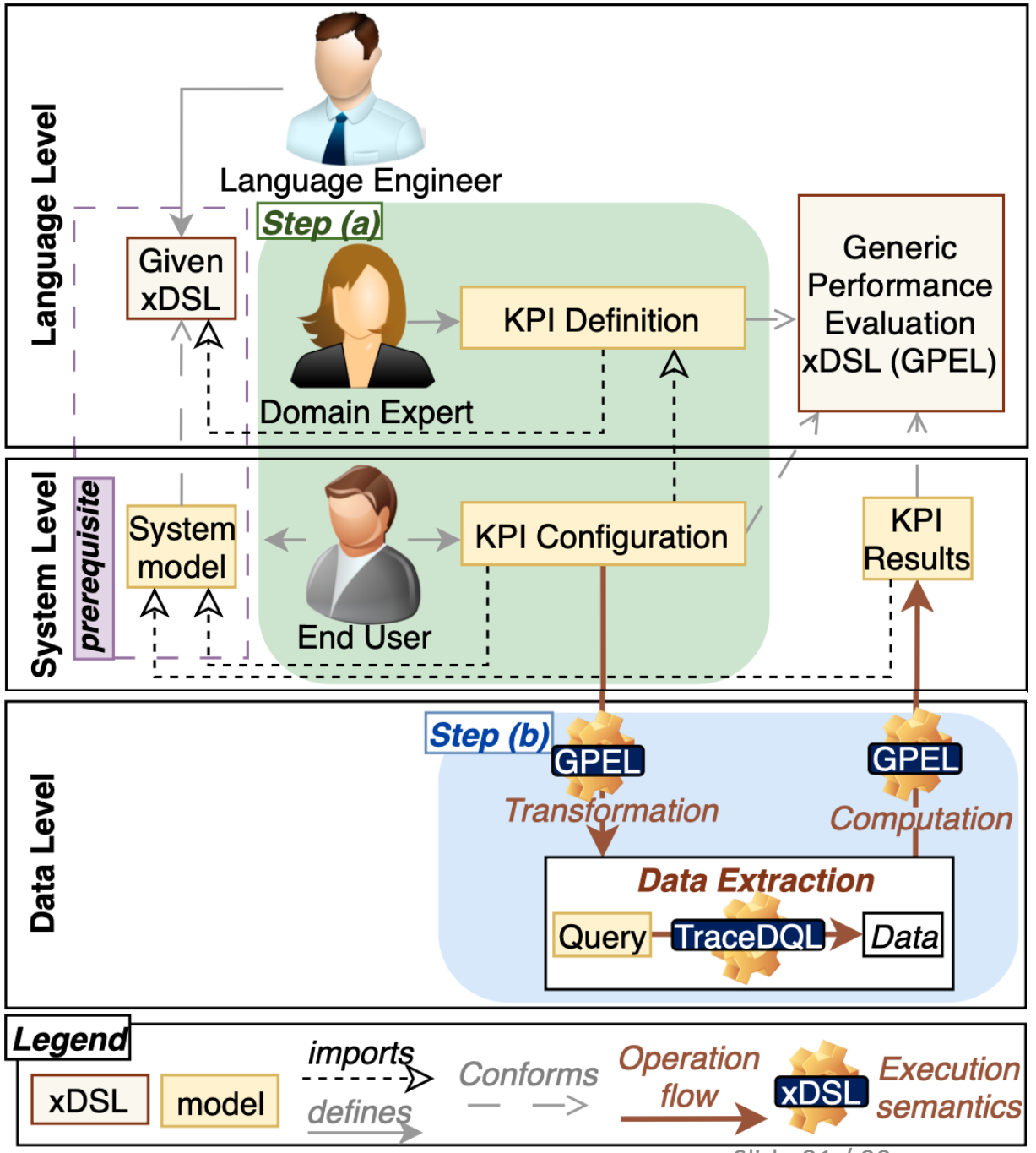
👉 Les KPIs sont calculés grâce **à la génération et à l'exécution automatiques de requêtes *TraceDQL*** afin d'extraire les données.

Proposition GPEL : approche générique et multi-niveaux pour l'évaluation de la performance (1/4)

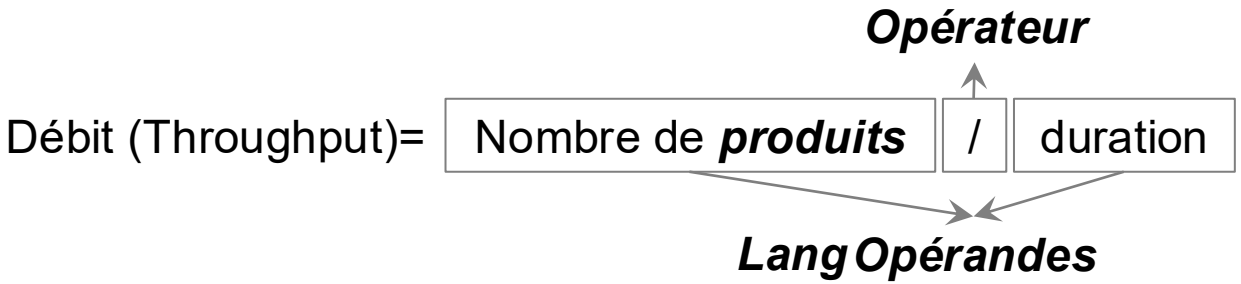
Trois rôles sont impliqués :

- **Ingénieur de langage** : conçoit le xDSL concerné.
- **Expert du domaine** : édite les formules de KPI en utilisant la terminologie du domaine via une approche de correspondance.
- **Utilisateur final** : configure les KPI définis en utilisant les éléments du système.

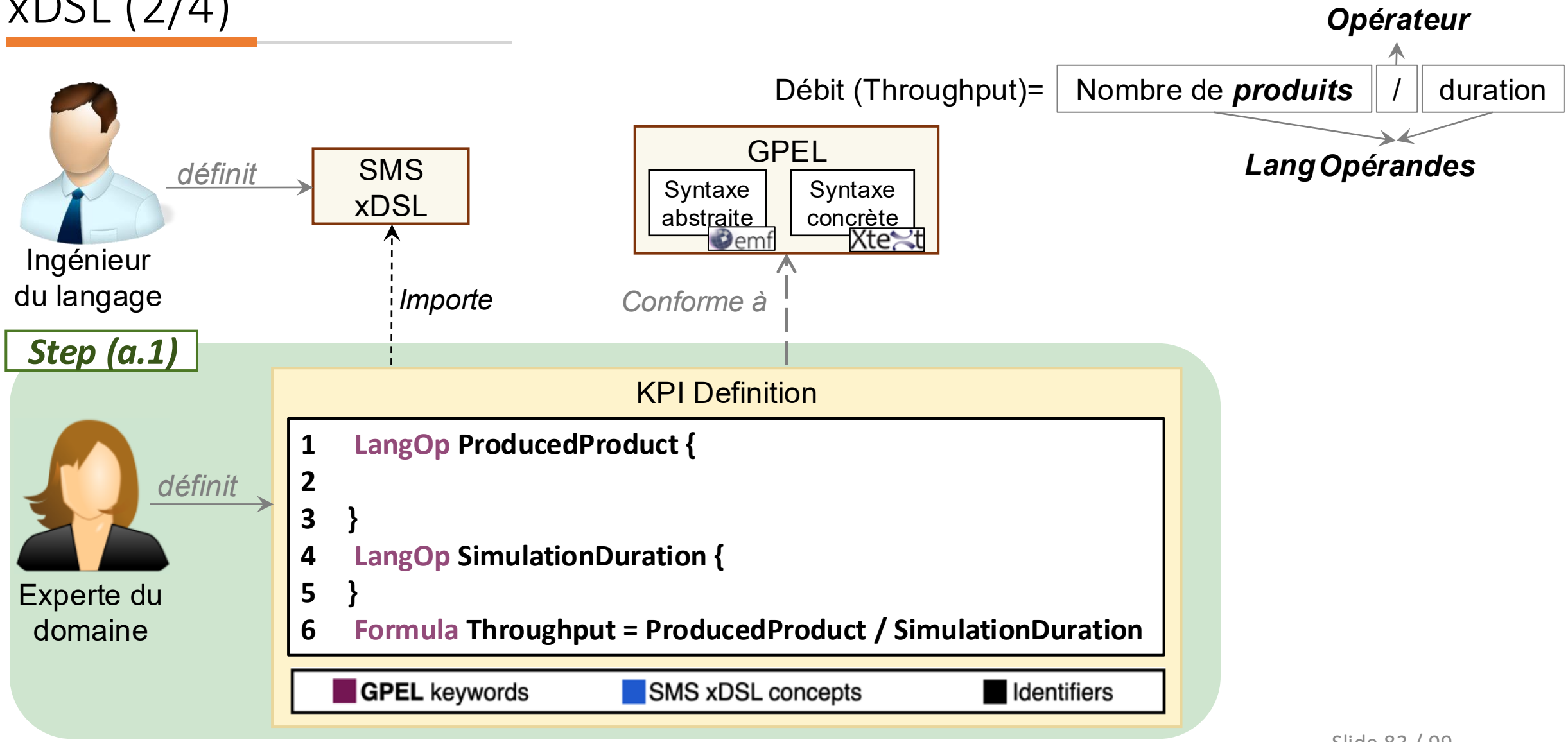
👉 Les KPIs sont calculés grâce **à la génération et à l'exécution automatiques de requêtes *TraceDQL*** afin d'extraire les données.



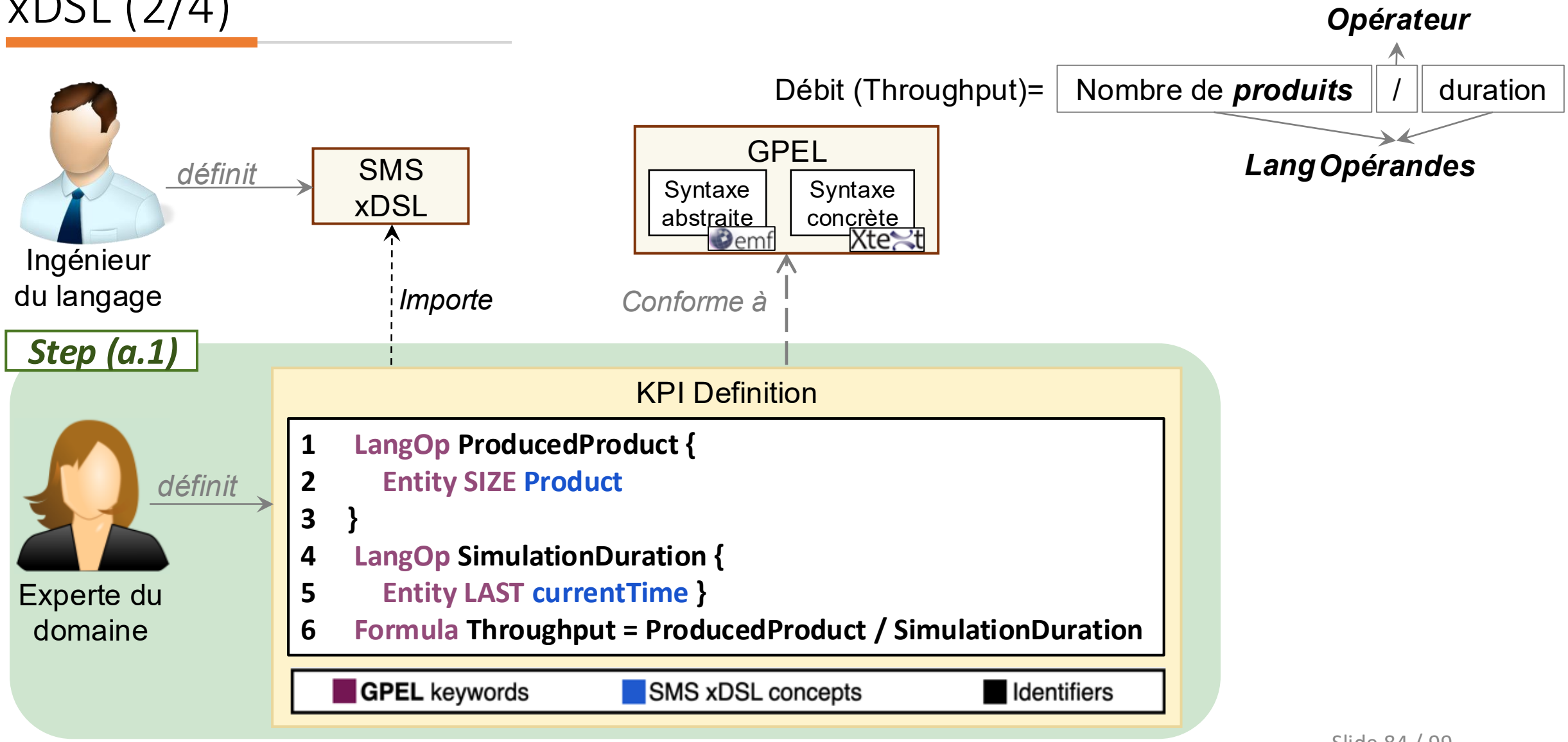
GPEL : exemple de KPI de débit des produits pour le SMS xDSL (2/4)



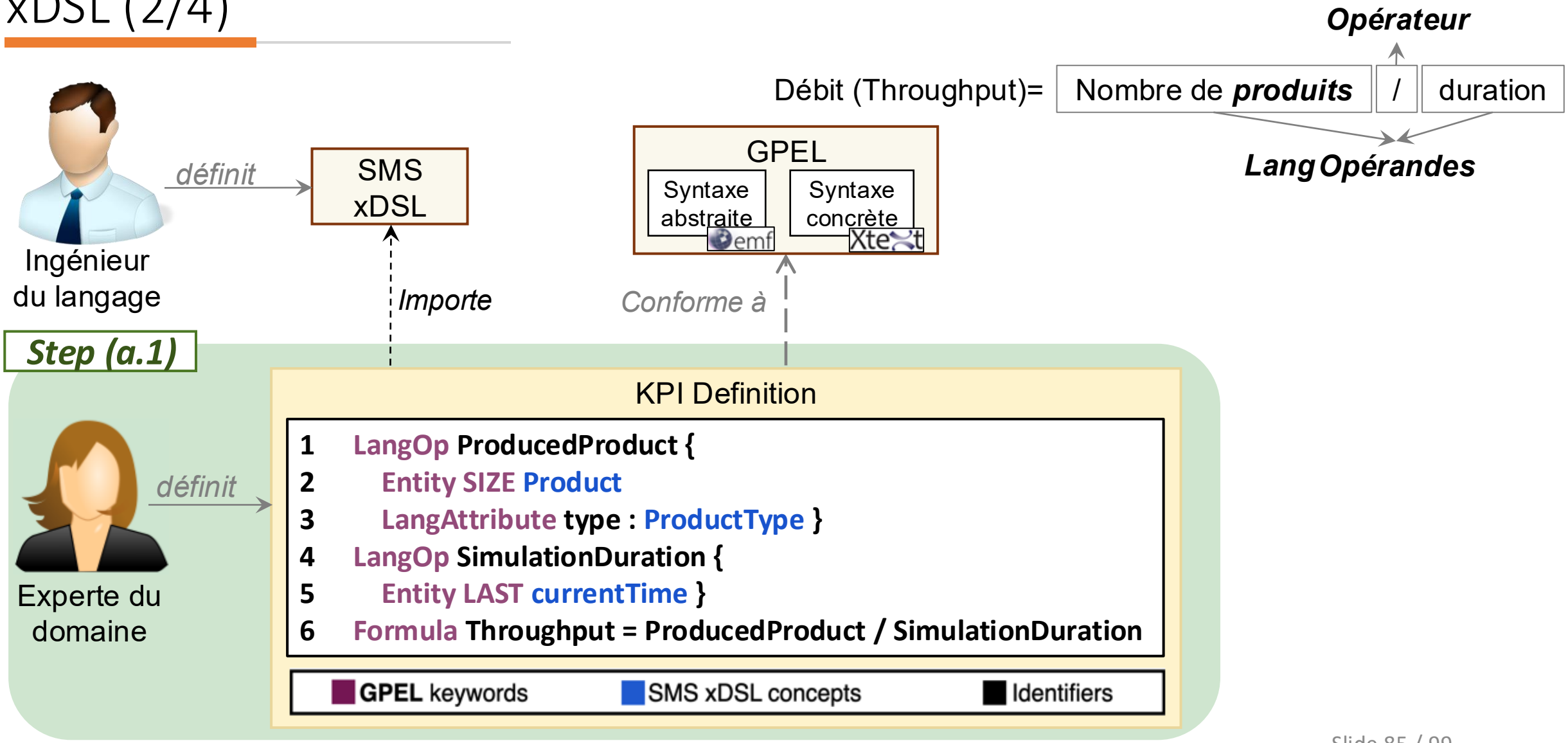
GPEL : exemple de KPI de débit des produits pour le SMS xDSL (2/4)



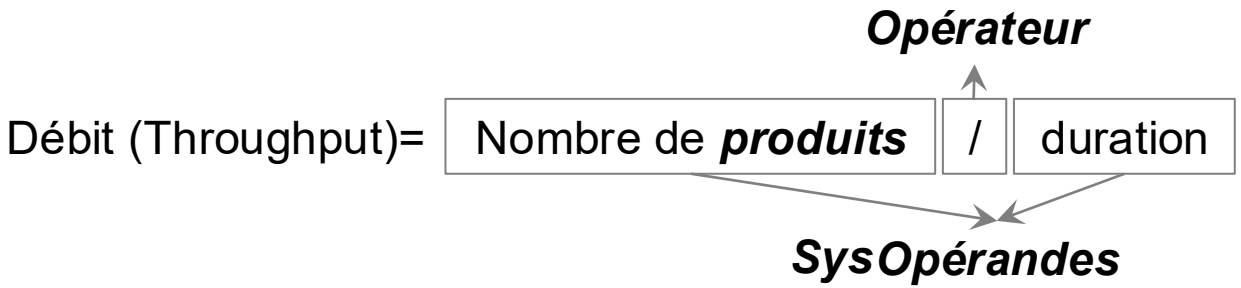
GPEL : exemple de KPI de débit des produits pour le SMS xDSL (2/4)



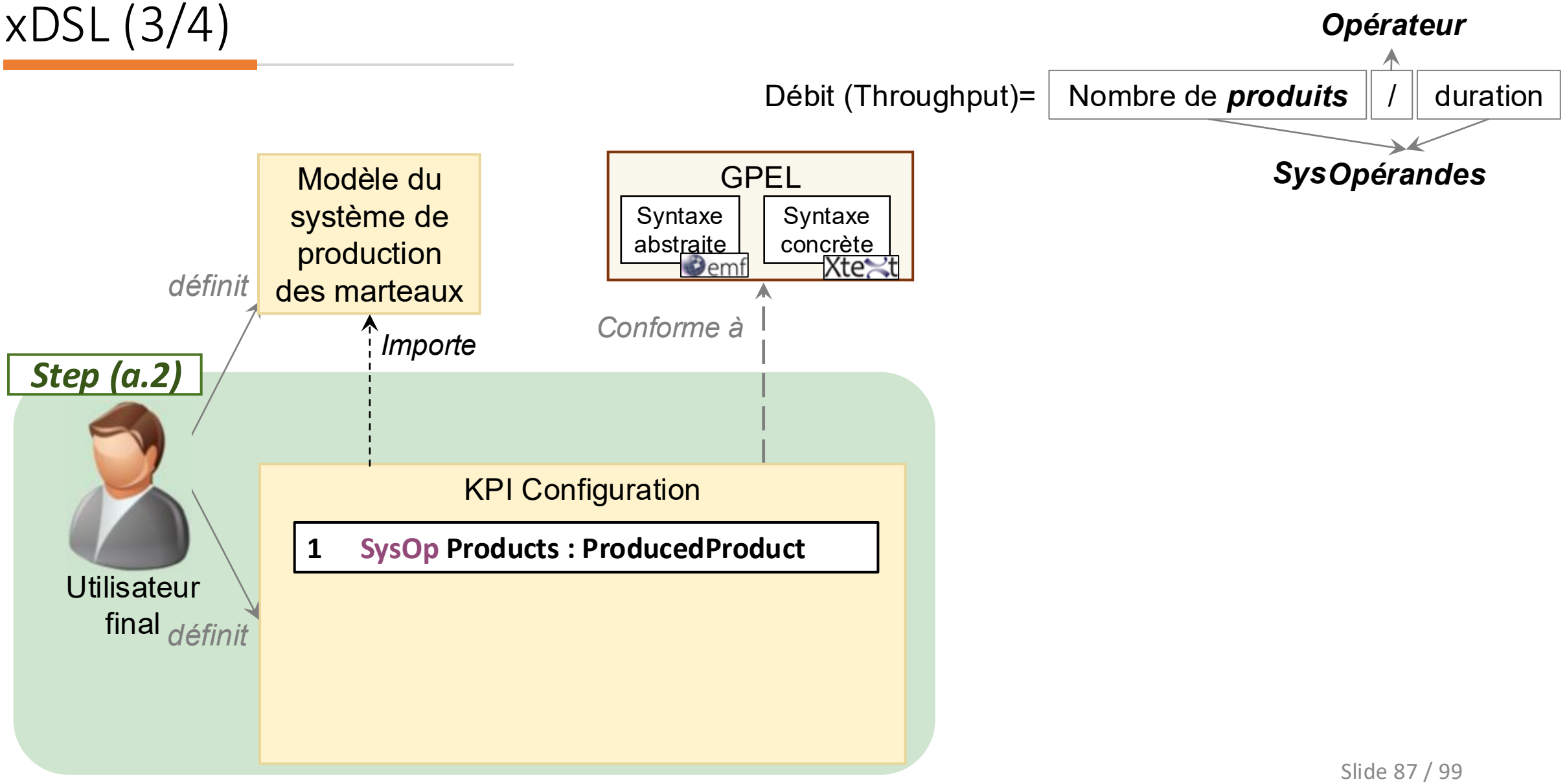
GPEL : exemple de KPI de débit des produits pour le SMS xDSL (2/4)



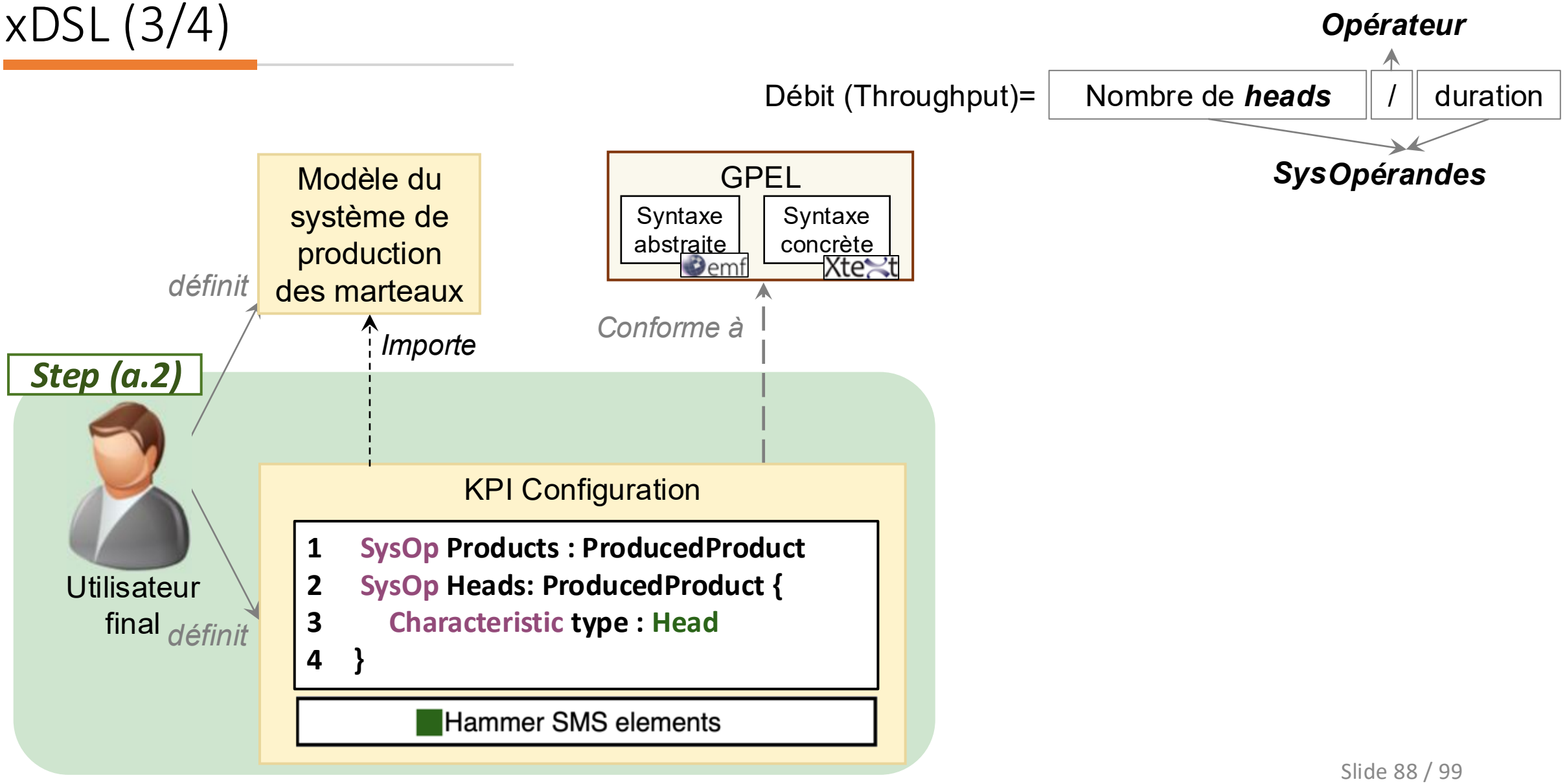
GPEL : exemple de KPI de débit des produits pour le SMS xDSL (3/4)



GPEL : exemple de KPI de débit des produits pour le SMS xDSL (3/4)



GPEL : exemple de KPI de débit des produits pour le SMS xDSL (3/4)



GPEL : extraction de données à l'aide de TraceDQL et des KPIs (4/4)

Step (b)

```
1  SysOp Products : ProducedProduct
```

génère

```
1  Block Products_Block ( ) {
2     query query_filter {
3           SELECT SIZE ON Product ,
4           REMOVE REDUNDANCY BY id IN Product } }
5  Data data_Products ( ) = Products_Block ;
```

```
1  SysOp Heads : ProducedProduct {
2     Characteristic type : Head
3  }
```

génère

```
1  Block Heads_Block ( ProductTypeID ) {
2     query query_filter {
3           SELECT SIZE ON Product ,
4           WHERE ( type . name ) EQUAL ( value ProductTypeID ) ,
5           REMOVE REDUNDANCY BY id IN Product } }
6  Data data_Heads_Head(ProductTypeID :: Head)=Heads_Block ;
```

GPEL : extraction de données à l'aide de TraceDQL et des KPIs (4/4)

Step (b)

```
1  SysOp Products : ProducedProduct
```

Génère

```
1  Block Products_Block ( ) {
2     query query_filter {
3           SELECT SIZE ON Product ,
4           REMOVE REDUNDANCY BY id IN Product } }
5  Data data_Products ( ) = Products_Block ;
```

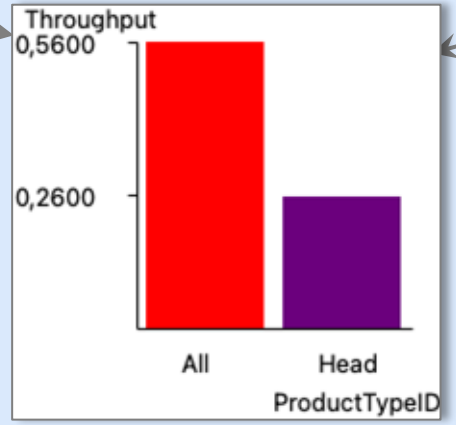
résultats (cf. TraceDQL)

```
1  SysOp Heads : ProducedProduct {
2     Characteristic type : Head
3  }
```

génère

```
1  Block Heads_Block ( ProductTypeID ) {
2     query query_filter {
3           SELECT SIZE ON Product ,
4           WHERE ( type . name ) EQUAL ( value ProductTypeID ) ,
5           REMOVE REDUNDANCY BY id IN Product } }
6  Data data_Heads_Head(ProductTypeID :: Head)=Heads_Block ;
```

résultats



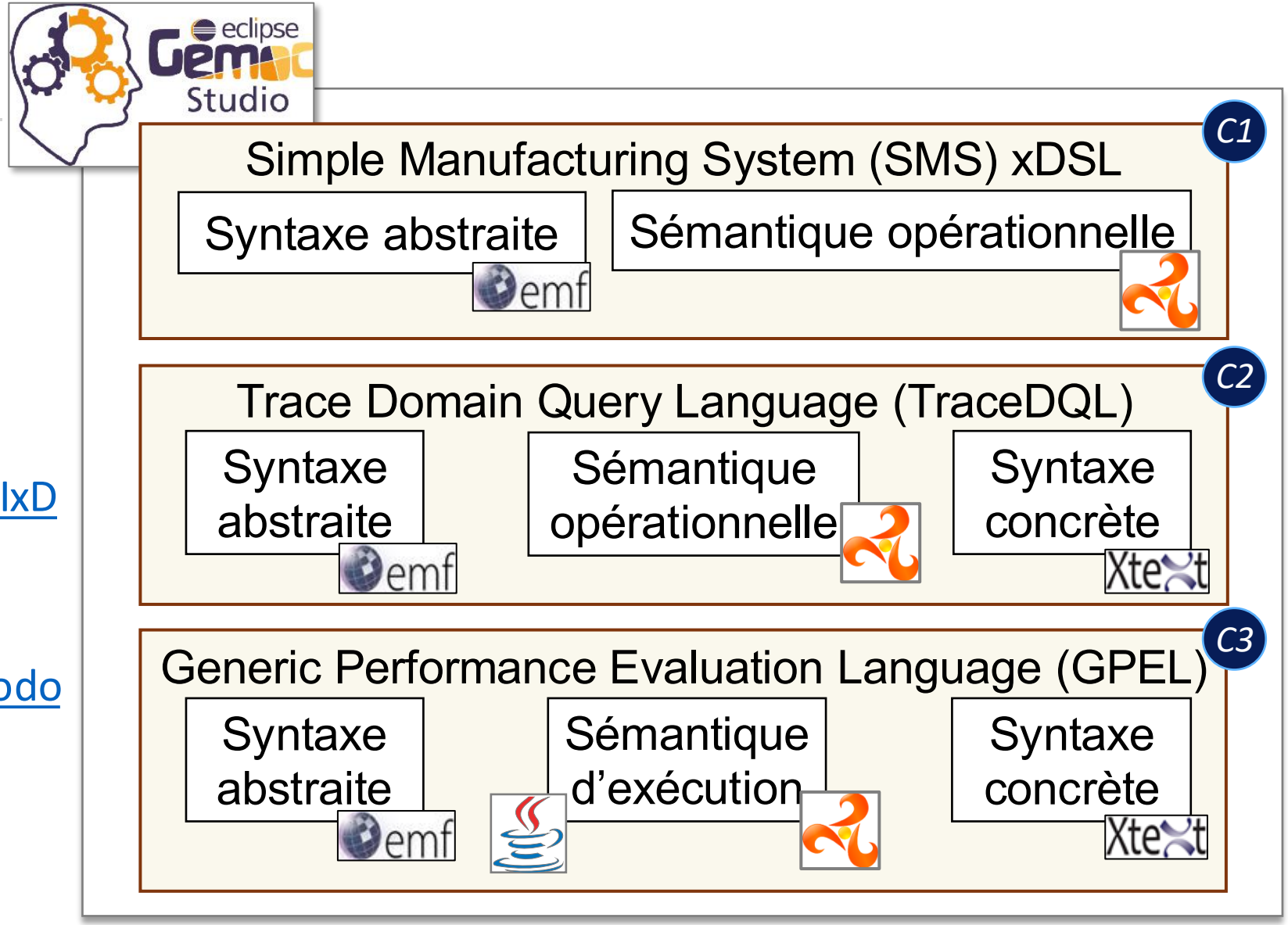
Expérimentation et évaluation de l'approche

Case Study	Domain	xDSL	Size MM	Model	Size M	Computed KPIs
1	Manufacturing	SMS xDSL	16	Hammer Factory model	16	- Throughput - Machine Utilization - Machine Idle Rate - Average Lead Time - Lead Time
				Simple Case model	30	
2		RMS xDSL	52	Lampex model	254	
3	Electronic	Arduino xDSL	59	Blinking LED model	18	- LED Utilization - LED Idle Rate - LED Energy Consumption - System Energy Consumption

Implémentation

Le code est disponible sur GitLab/Zenodo :

- https://gitlab.univ-nantes.fr/rodic/evaluationKPIxDSL
- https://doi.org/10.5281/zenodo.19320717



Définition des KPIs pour SMS xDSL

Données extraites

Association des concepts à des KPIs

Édition de formule KPIs

Trace d'exécution

Configuration des KPIs pour la ligne de production des marteaux

Configuration des KPIs pour *Simple Case Study*

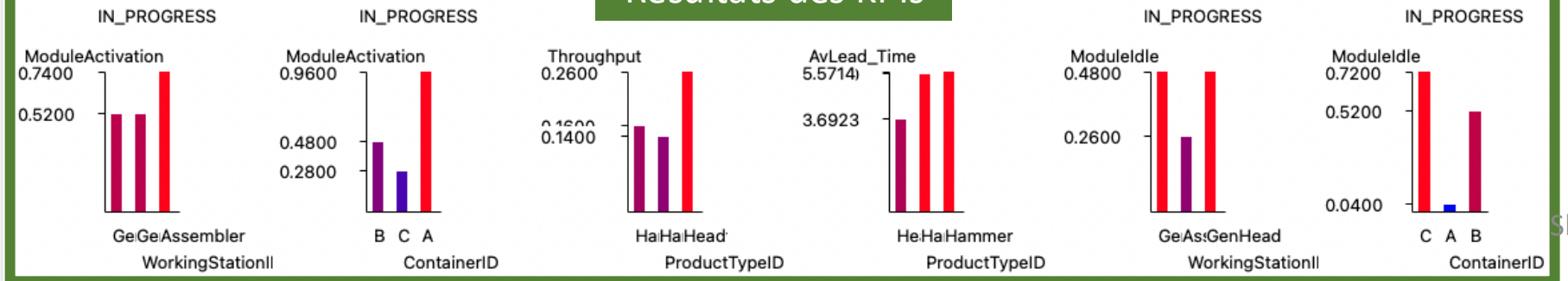
Requêtes *TraceDOL* générées

Résultats des KPIs

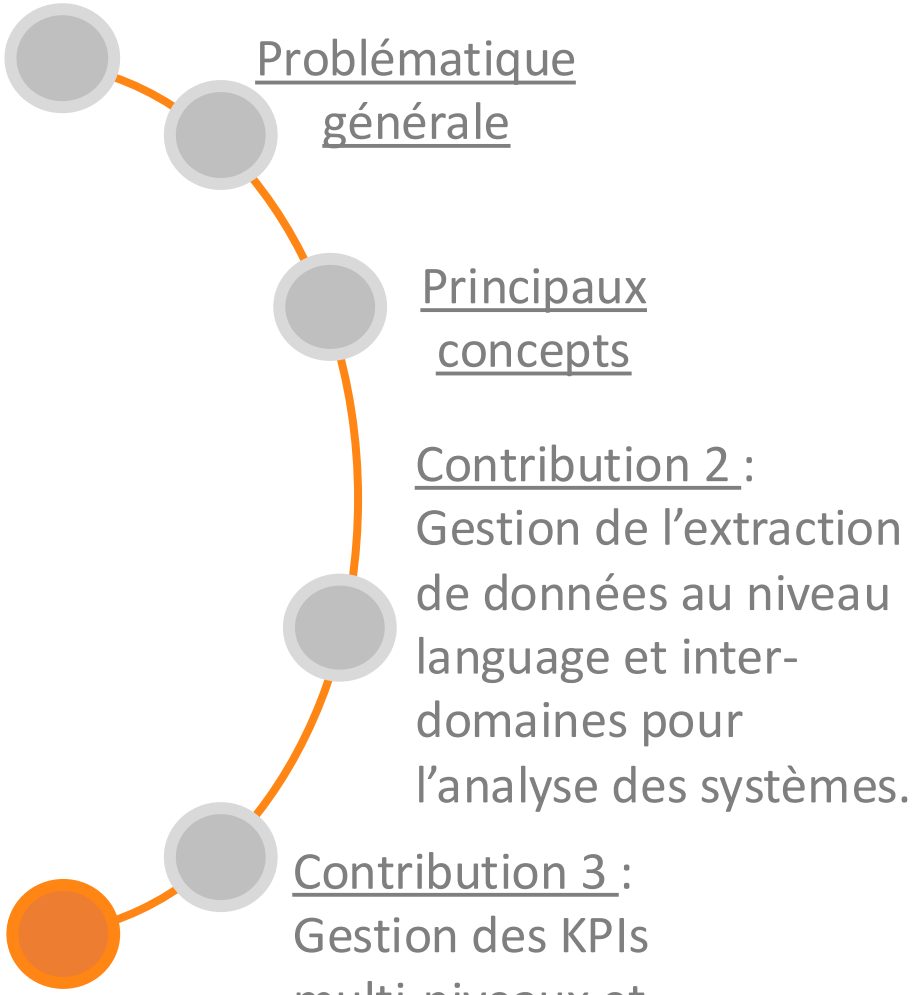
```
KPIDefinition SmsKPIDefinition{

    LangOp ProducedProduct{
        Entity SIZE Product
        LangTime : Entity SimulationState.currentTime
        LangContainer container : ContainerState IN RModule //RModule Container
        LangAttribute type : ProductType
        LangAttribute id : Product.id
    }

    LangOp ModuleOperation{
        LangTime : Entity SimulationState.currentTime
        LangContainer module : RModuleState IN  RModule //WorkingStation
        LangInterval : ORDERED SimulationState.currentTime DISTINCT Task.state
    }

    LangOp SimulationDuration{
        Entity LAST SimulationState.currentTime
    }

    LangOp ProductDistructionTime{
        Entity LAST SimulationState.currentTime
        LangAttribute pp : ProducedProduct
    }

    LangOp ProductConstructionTime{
        Entity FIRST SimulationState.currentTime
        LangAttribute pp : ProducedProduct
    }

    Formula Throughput = ProducedProduct / SimulationDuration
    Formula ModuleActivation = ModuleOperation / SimulationDuration
    Formula ModuleIdle = 1 - Formula ModuleActivation
    Formula AvLead_Time IDENTICAL = (ProductDistructionTime - ProductConstructionTime)/ ProducedProduct
}
```

```
SysOp SD : SimulationDuration

/** Throughput **/

SysOp Heads : ProducedProduct{
    Characteristic  type : Head
}

SysOp Handles : ProducedProduct{
    Characteristic  type : Handle
}

SysOp Hammers : ProducedProduct{
    Characteristic  type : Hammer
}
```

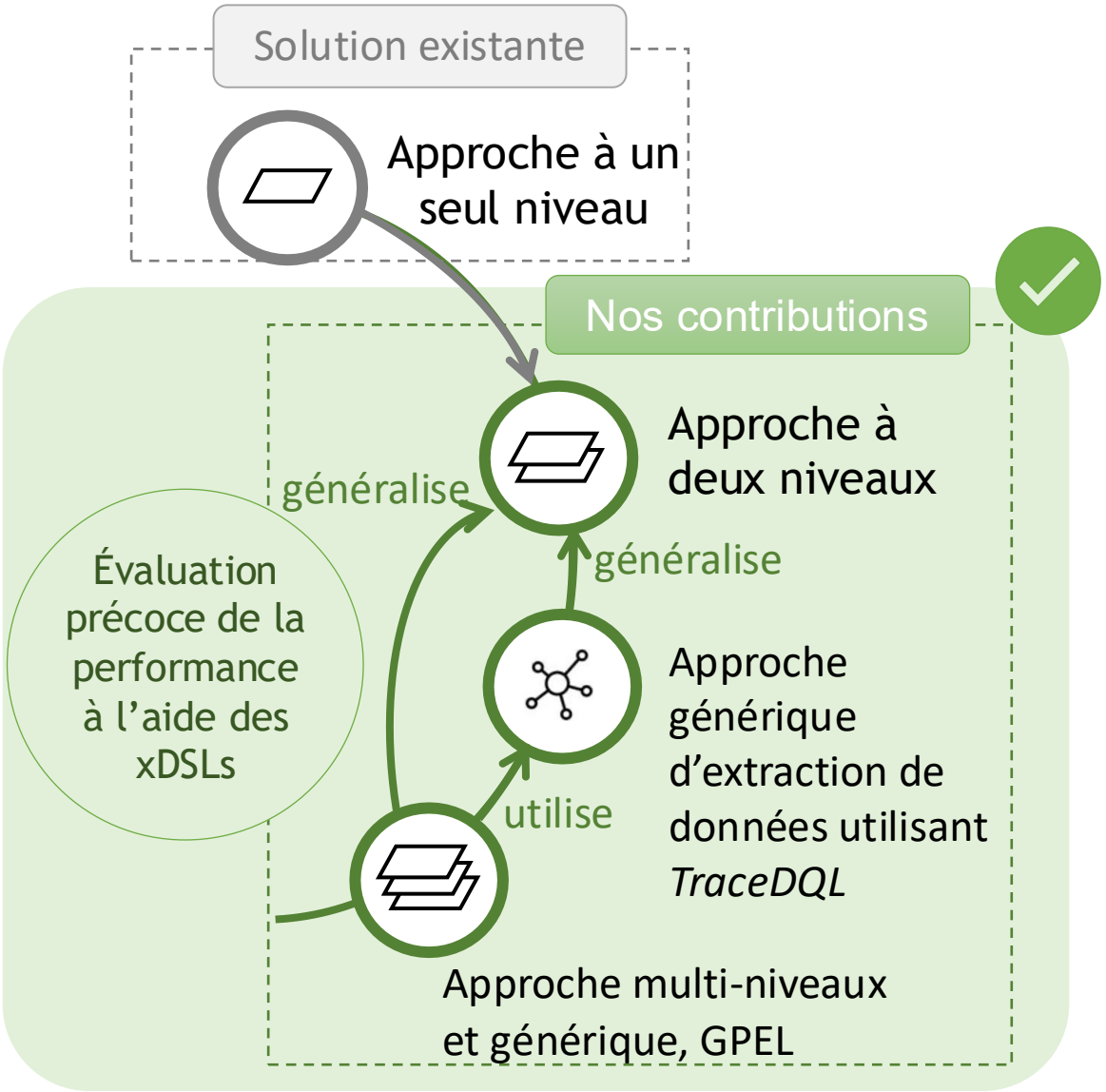
```
SysOp SD : SimulationDuration

/** Throughput **/

SysOp All_PP : ProducedProduct

SysOp PP_By_Type : ProducedProduct{
    Characteristic  type : RedProduct
}

SysOp PP_By_Type : ProducedProduct{
    Characteristic  type : Product
}

SysOp PP_By_Type : ProducedProduct{
    Characteristic  type : Pallete
}
```



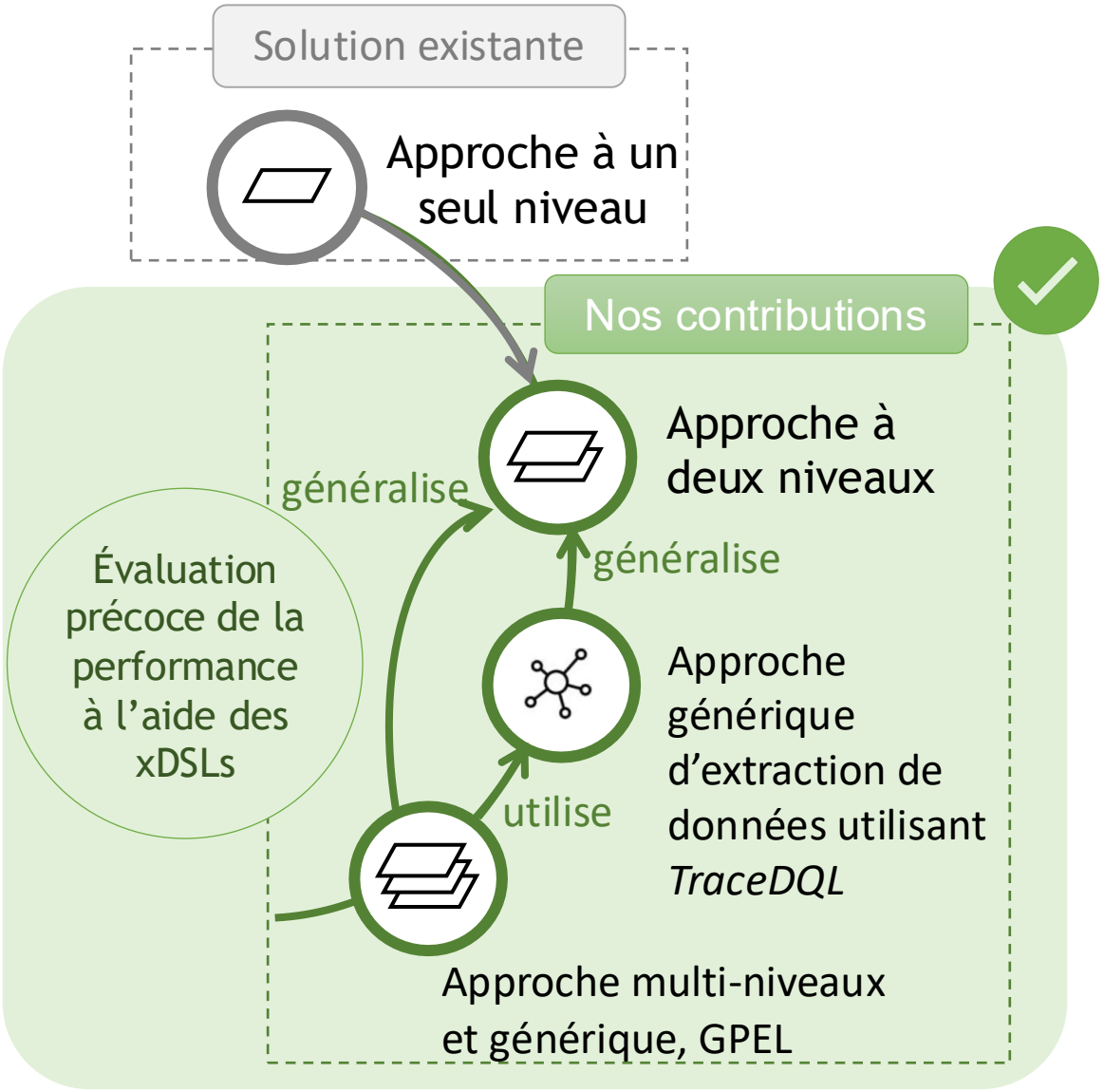
Plan de l’exposé

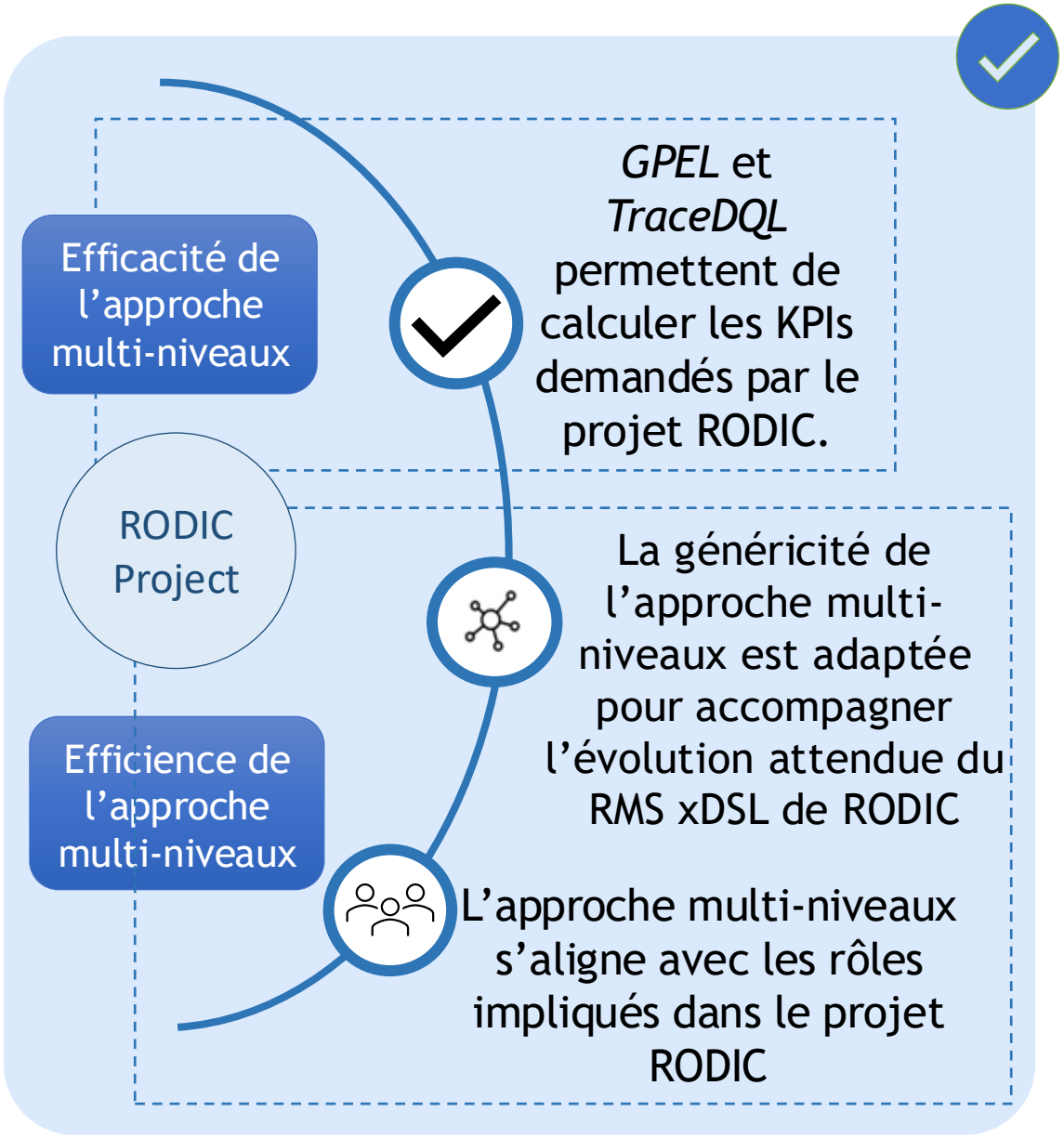


Conclusion et perspectives

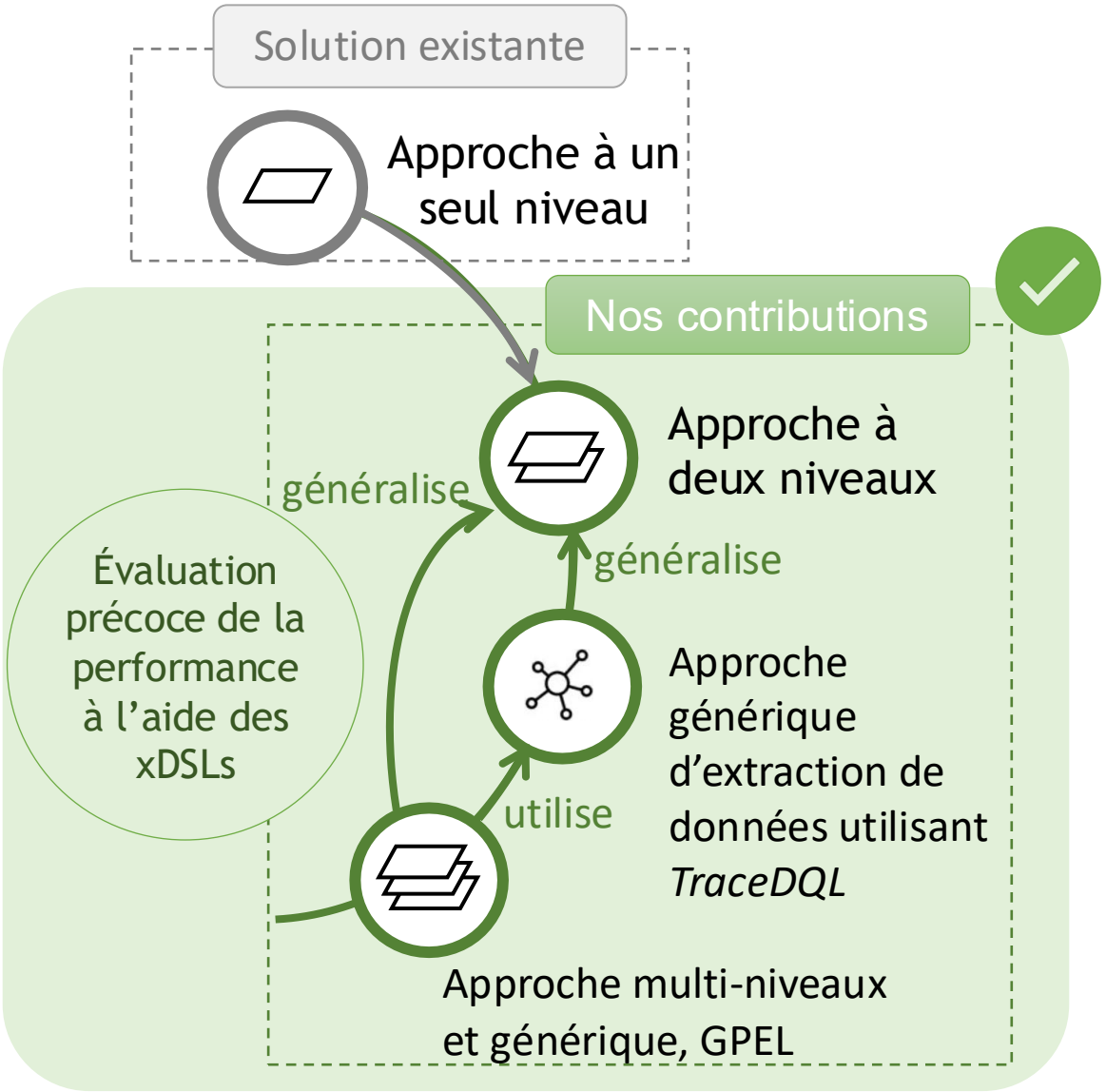


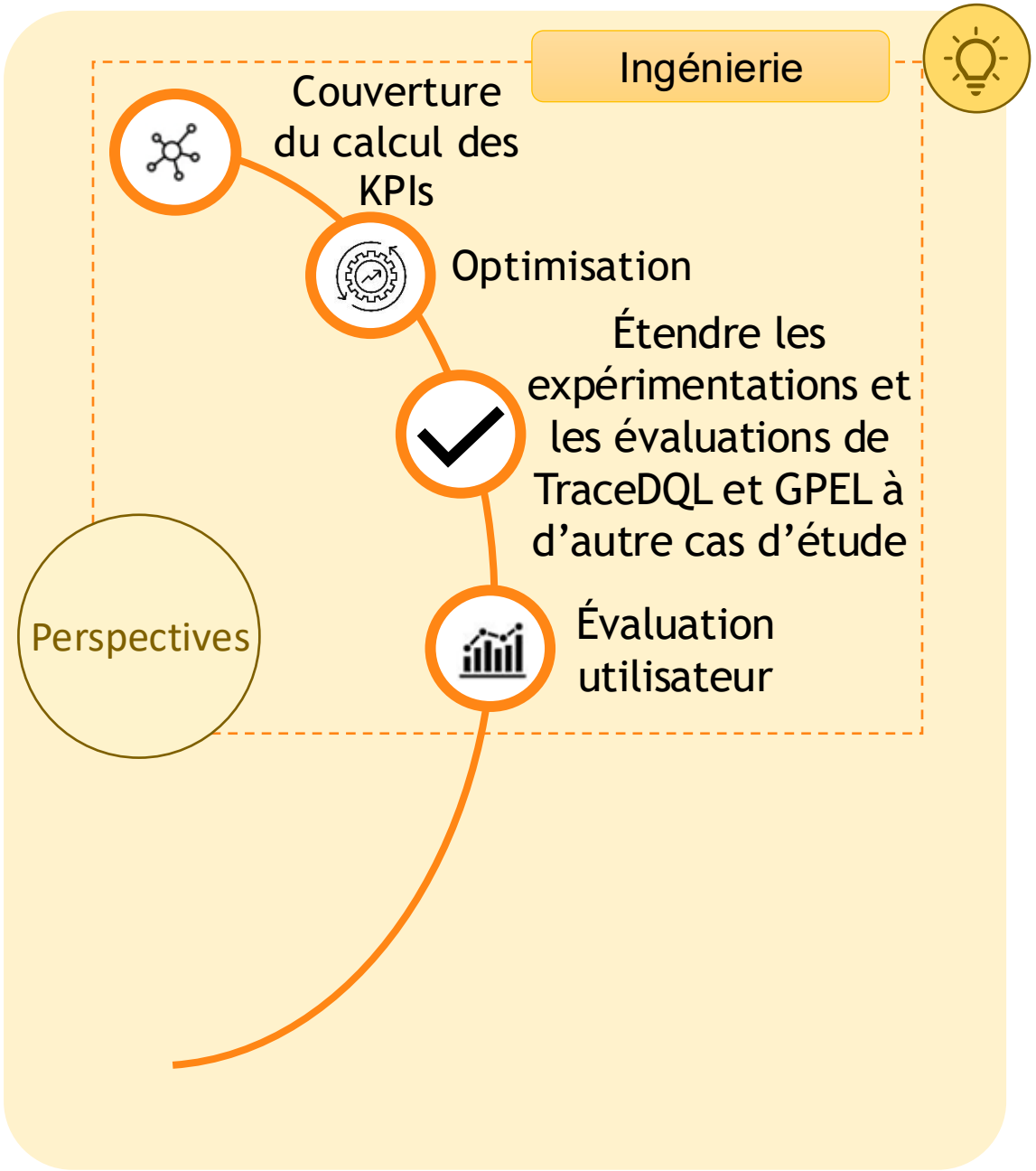
Conclusion et perspectives



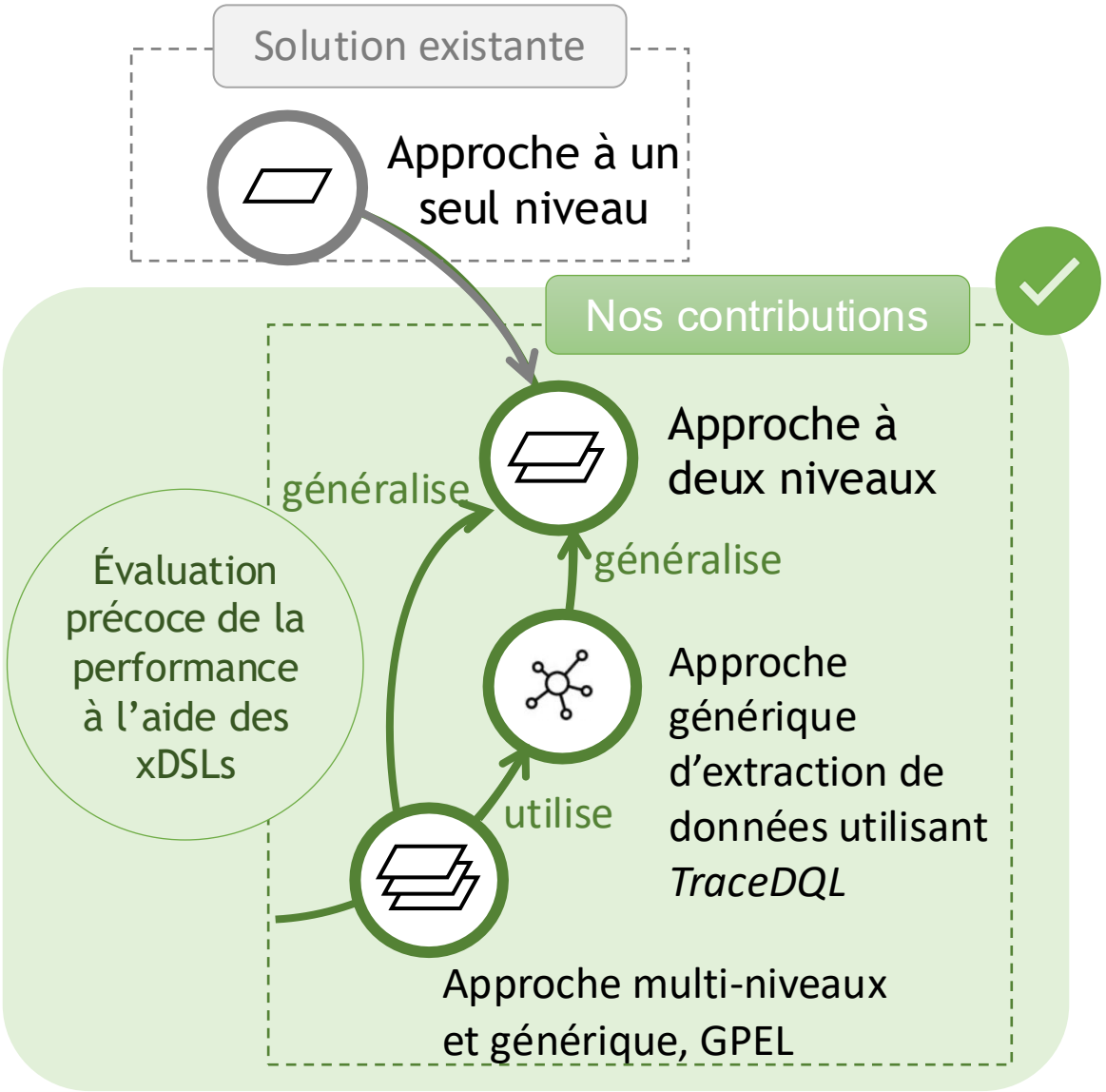


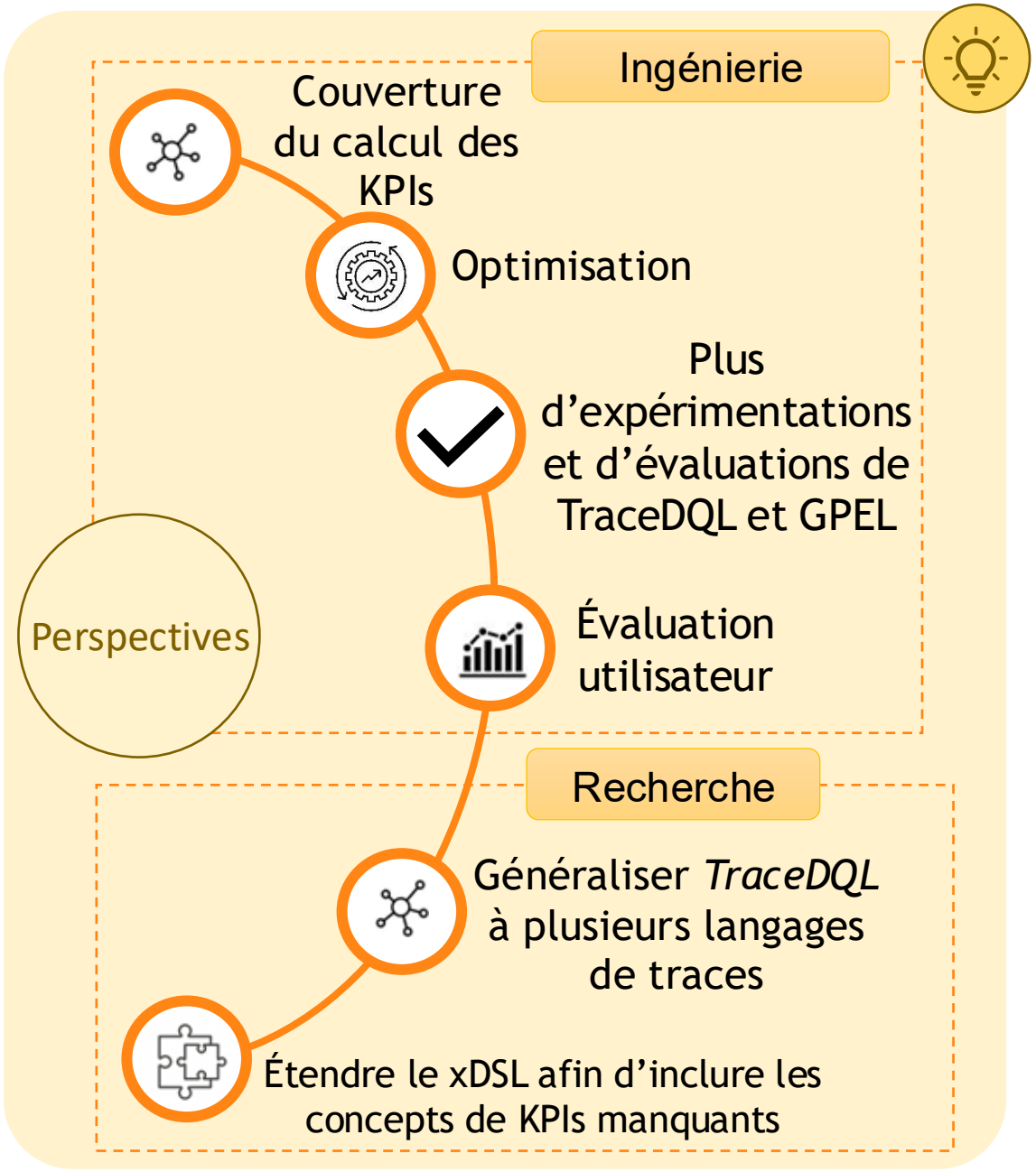
Conclusion et perspectives





Conclusion et perspectives





COLLEGE MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Laboratoire des Sciences du Numérique de Nantes

Évaluation de la performance basée sur les langages exécutables spécifiques au domaine

Soutenance de thèse de doctorat de Nantes Université par

Hiba AJABRI

le 30 avril 2026

- **Sous la direction de** : Pr. Christian ATTIOGBE, Pr. Pascal BERRUET, Dr. Jean-Marie MOTTU
- **Rapporteures** : Pr. Agnès FRONT, Pr. Christelle URTADO
- **Examinateur** : Pr. Julien DEANTONI **Invité** : Dr. Florent De Lamotte

Annexe

Vue d'ensemble de contribution 1

- **Contribution 1** : Approche à deux niveaux pour la gestion des KPI au lieu d'une approche à un seul niveau.