

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université de Jijel**  
**Faculté des Sciences exactes et de l'informatique**  
**Département d'informatique**



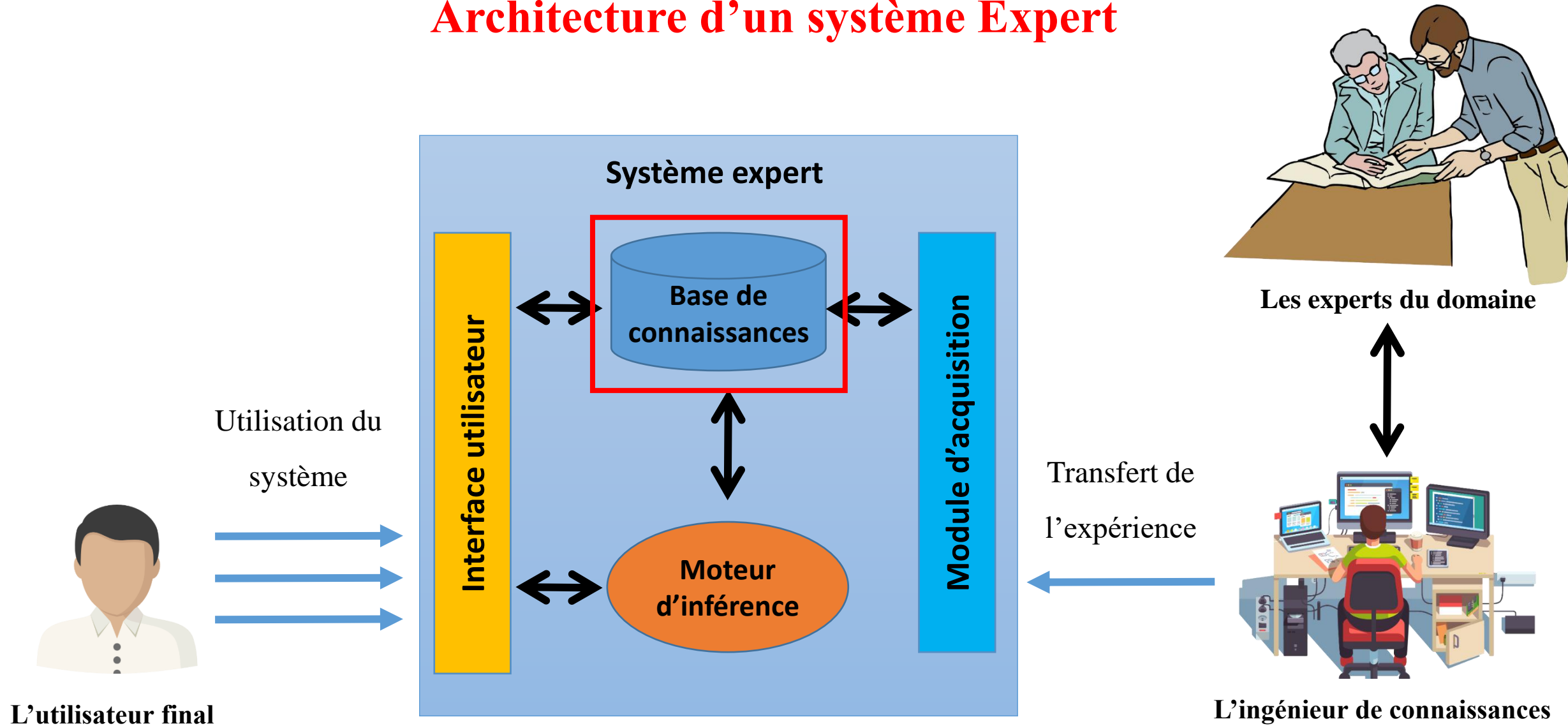
**– Module –**  
**Systemes Experts**

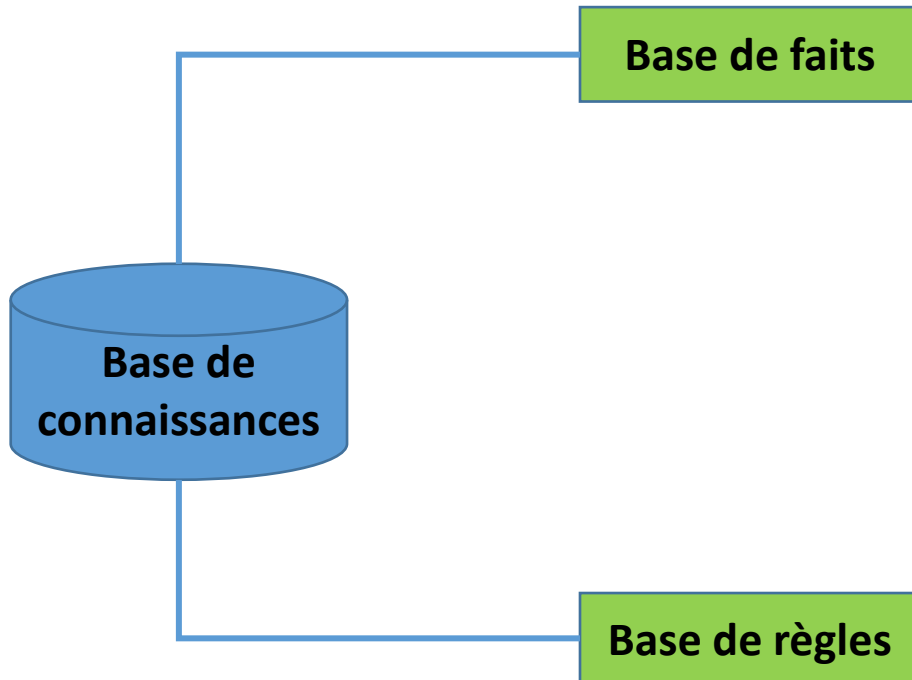
**Master 1 : SIAD**

**Enseignant du module : Dr. Hemza FICEL**

**Contact: [hemza.ficel@univ-jijel.dz](mailto:hemza.ficel@univ-jijel.dz)**

## Architecture d'un système Expert





- ✚ Conserve la connaissance sur l'**étude en cours** (constats, réalité).
  - ✚ Son état évolue au cours de l'expertise.
  - ✚ Considérée comme un espace de travail temporaire
  - ✚ Exemple : Ali est un étudiant algérien.
- 
- ✚ Conserve la connaissance sur le **domaine**.
  - ✚ Les règles établissent des relations entre les faits.
  - ✚ Représentation la plus utilisée : **Si** (prémisse) **Alors** (action)
  - ✚ La **prémisse** est la **condition** d'application de la règle (**partie déclencheur**)
  - ✚ La **conclusion** est le **résultat d'exécution** de la règle (**partie effets**).

✚ **Les faits** sont des conditions initiales à partir desquelles d'autres connaissances pourront être utilisées.

✚ Exemple de faits : Ali habite Jijel.

✚ La valeur d'un **fait est toujours vraie** selon une propriété ou des critères donnés.

✚ Les **règles** représentent des **connaissances opératoires** (fournies par l'expert de domaines) qui **permettent de déduire d'autres connaissances (faits)**.

✚ **Une règle** est composée d'une **prémisse qui implique une conclusion**. Les prémisses et les conclusions sont constituées de **faits**.

✚ **Attention à ne pas confondre** une règle de déduction avec une règle d'inférence

✚ **Une règle de déduction** correspond à une connaissance opératoire qui fait partie de la base de connaissance du système expert. Elle permet d'établir des conclusions à partir de prémisses connues en utilisant une structure logique.

**Exemple : SI (A et B ) alors C.**

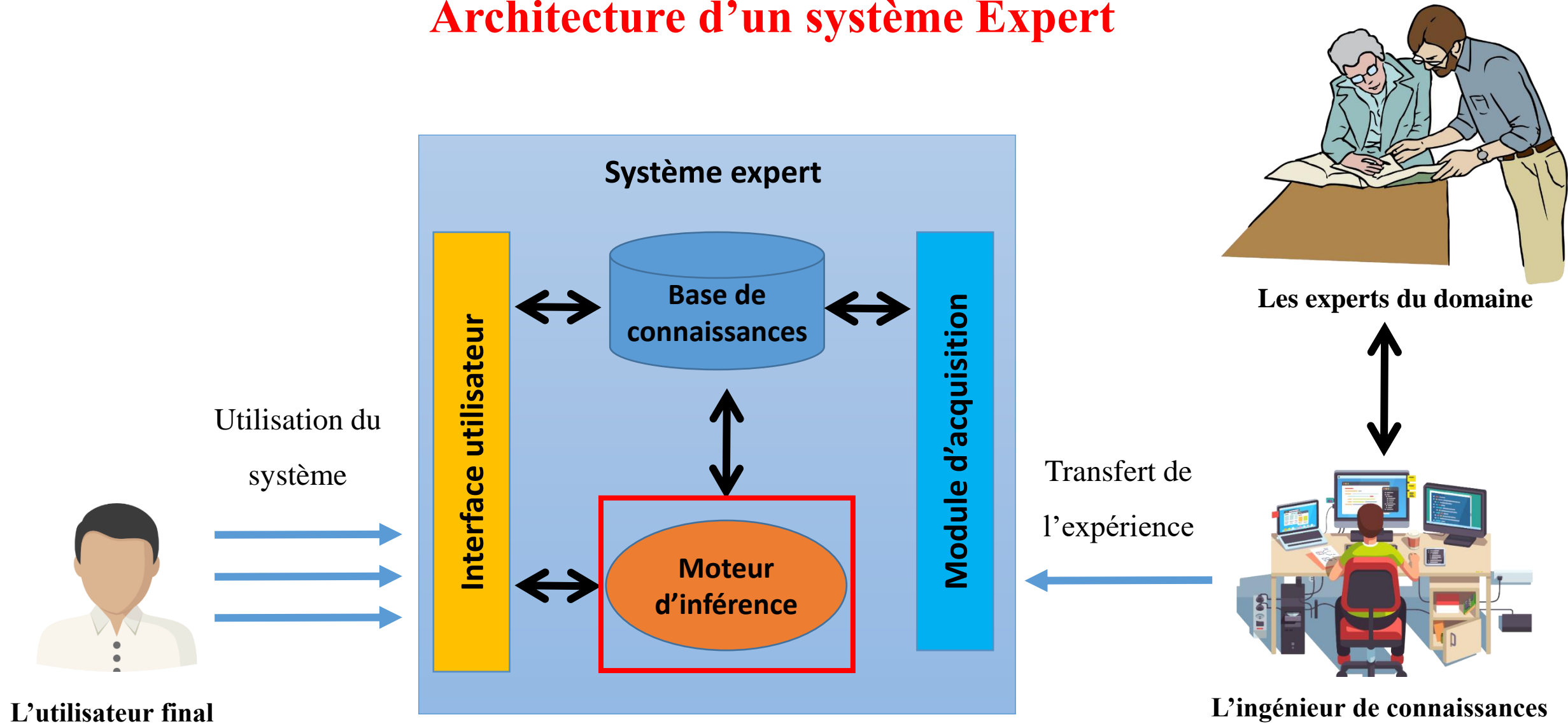
✚ **Une règle de d'inférence** correspond à un mécanisme de raisonnement externe à la base de connaissance du système expert. Elle permet de faire des inférences à partir de faits connus et établir de nouveaux faits.

**Exemple : Le modus Ponens**

**SI (P est vrai et  $P \rightarrow Q$  ) alors Q est vrai.**

# Chapitre 3 – Moteur d'Inférence

## Architecture d'un système Expert



Un « **moteur d'Inférence** », c'est quoi exactement ?



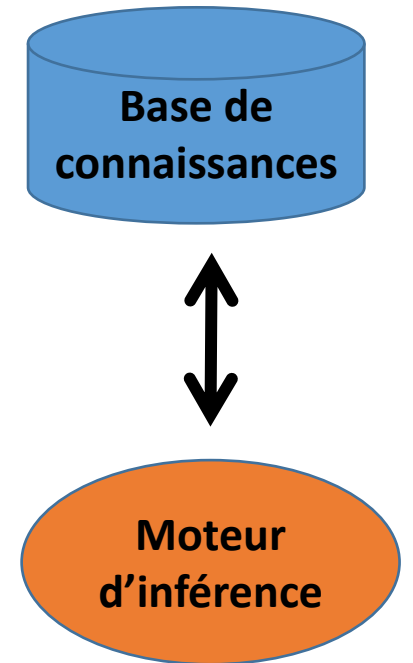
## Définition

✚ Inférer : **tirer une conséquence** de quelque chose, **conclure, induire** quelque chose de quelque chose. **Dictionnaire Larousse.**

✚ Le moteur d'inférence : est un programme qui **utilise les connaissances** stockées dans la base de connaissances **pour résoudre un problème** posé par un utilisateur.

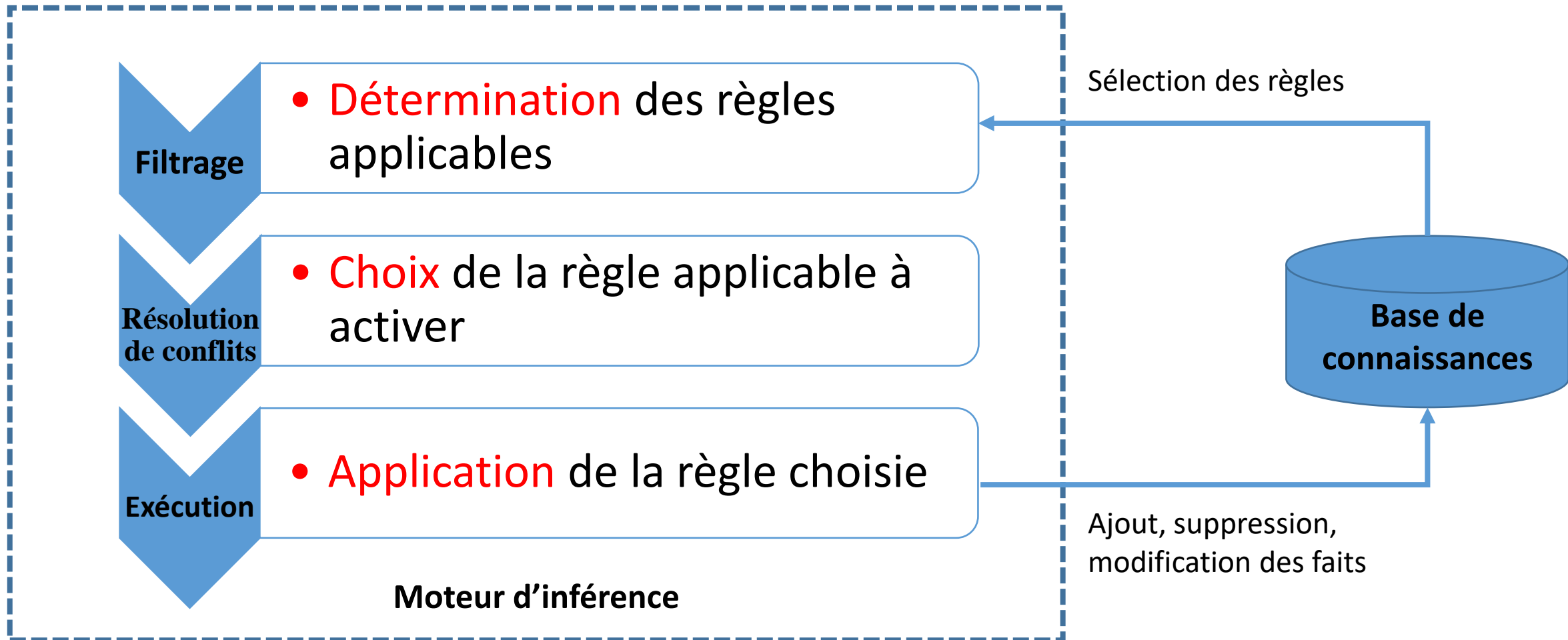
✚ Le moteur d'inférence **applique les règles aux faits** (réalité, hypothèses) de départ **pour déduire de nouveaux faits** afin d'atteindre un objectif.

✚ Le moteur d'inférence **simule l'esprit logique** de l'expert humain en **raisonnant sur des vérités connus pour en sortir de nouvelles vérités.**

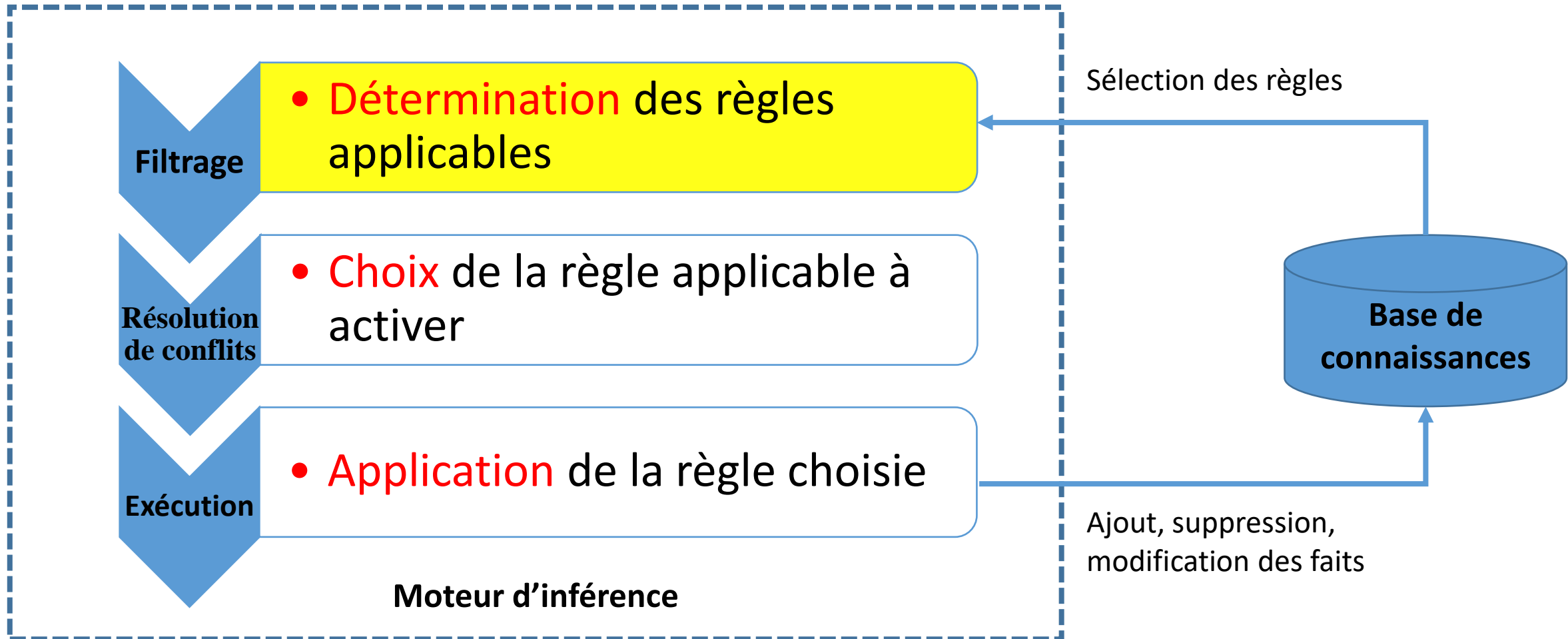


Le moteur d'inférence suit **un processus logique** pour résoudre les problèmes en utilisant les règles et les faits disponibles dans la base de connaissances ...

## Cycle de base d'un moteur d'inférence



## Cycle de base d'un moteur d'inférence



## Cycle de base d'un moteur d'inférence

### Etape de filtrage

✚ **Détermination des règles applicables** : le moteur d'inférence commence par examiner toutes les règles de la base de connaissances pour trouver celles qui sont **applicables** aux faits disponibles. Ainsi, les **règles candidates** sont sélectionnées lors de cette première étape (**étape de filtrage**) du cycle de base d'un moteur d'inférence.



✚ **Règle candidate** : est une règle dont le déclencheur (la partie conditionnelle) s'apparie avec les faits présents dans la base de connaissances (**toutes les conditions de la règle sont satisfaites**). Cela veut dire que la règle est **potentiellement applicable** à la situation actuelle.

## Cycle de base d'un moteur d'inférence

### Etape de filtrage

✚ Pour chaque règle, le moteur d'inférence vérifie si les **prémises de la règle sont satisfaites par les faits actuels**. Si c'est le cas, la règle est considérée comme applicable.

#### Exemple : Système expert pour la détection de fraudes

Base de règles	Base de faits ( <b>Utilisateur 1</b> )
<b>R1:</b> Si <b>Achats &gt; 10</b> et <b>Session &lt; 5 minutes</b> , alors <b>signaler la transaction comme suspecte</b> .	Achats = 15; Session = 3 minutes ; Montant = 2000 €
<b>R2 :</b> Si <b>Achats = 1</b> et <b>Montant &gt; 5000 €</b> , alors <b>demander une vérification manuelle d'une possible fraude</b> .	

**R1, est-elle applicable ? / Tous les faits de R1 sont-ils dans la base des faits ?**



**R2, est-elle applicable ? / Tous les faits de R2 sont-ils dans la base des faits ?**



## Cycle de base d'un moteur d'inférence

### Etape de filtrage

✚ Pour chaque règle, le moteur d'inférence vérifie si les **prémises de la règle** sont satisfaites par les faits actuels. Si c'est le cas, la règle est considérée comme applicable.

#### Exemple : Système expert pour la détection de fraudes

Base de règles	Base de faits ( <b>Utilisateur 2</b> )
<b>R1:</b> Si <b>Achats &gt; 10</b> et <b>Session &lt; 5 minutes</b> , alors <b>signaler la transaction</b> comme suspecte.	Achats = 1; Session = 4 minutes ; Montant = 9000 €
<b>R2 :</b> Si <b>Achats = 1</b> et <b>Montant &gt; 5000 €</b> , alors <b>demander une vérification manuelle</b> d'une possible fraude.	

**R1, est-elle applicable ? / Tous les faits de R1 sont-ils dans la base des faits ?**



**R2, est-elle applicable ? / Tous les faits de R2 sont-ils dans la base des faits ?**



On peut avoir plusieurs règles applicables à la fois.

**Quelle règle à activer ?**



**Situation de conflits**



## Cycle de base d'un moteur d'inférence

Exemple d'une situation de conflits dans un système expert pour la détection de fraudes

Base de règles	Base de faits ( <b>Utilisateur 3</b> )
R1: Si <b>Achats &gt; 10</b> et <b>Session &lt; 5 minutes</b> , alors signaler la transaction comme suspecte.	Achats = 1 ; Session = 2 minutes ; Montant = 12000 €
R2 : Si <b>Achats = 1</b> et <b>Montant &gt; 4000 €</b> , alors demander une vérification manuelle d'une possible fraude.	
R3 : Si <b>Session &lt; 5 minutes</b> et <b>Montant &gt; 8000 €</b> , alors bloquer immédiatement l'accès au compte de l'utilisateur.	

R1, est-elle applicable ? / Tous les faits de R1 sont-ils dans la base des faits ?



R2, est-elle applicable ? / Tous les faits de R2 sont-ils dans la base des faits ?

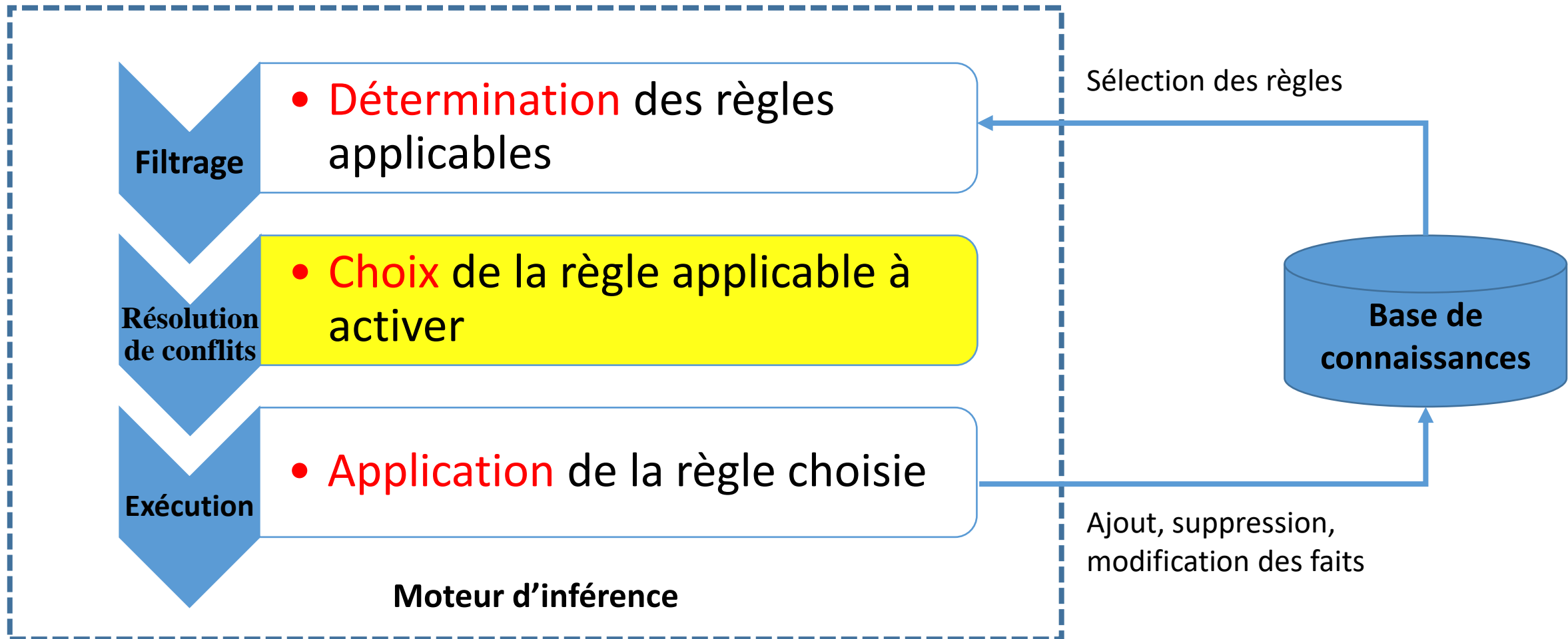


R3, est-elle applicable ? / Tous les faits de R3 sont-ils dans la base des faits ?



Comment choisir la **règle à activer dans cette situation ?**

## Cycle de base d'un moteur d'inférence



## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

✚ Dans un système expert, **une situation de conflits** peuvent survenir lorsque plusieurs règles sont applicables pour un même ensemble de faits. **Dans cette situation une seule règle peut être déclenchée à la fois !**



✚ Le système expert doit donc mettre en place des **stratégies de résolution de conflits** qui permettent de **choisir la règle applicable à activer (étape de résolution de conflits)**

## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

✚ Une **stratégie de résolution de conflits** comporte un ensemble de règles que le système applique pour résoudre des situations **où des règles contradictoires s'appliquent pour un même ensemble de faits.**

✚ Il existe plusieurs stratégies de résolution de conflits :

- Au hasard.
- Stratégie de priorité ou de confiance.
- A partir des méta-règles (règles de contrôle)
- ...

## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

✚ **Stratégie de priorité ou de confiance** : en attribuant des **coefficients de priorité/confiance** aux règles en fonction de leur **fiabilité ou importance**, cette stratégie assure que **les conditions les plus graves/importantes sont traitées en priorité**.

**Exemple : Système expert d'aide au diagnostic médical**

Base de règles	Base de faits ( <b>Utilisateur 1</b> )
<b>R1</b> : Si <b>douleur thoracique</b> et <b>fréquence cardiaque rapide</b> , alors <b>crise cardiaque</b> .	<b>Douleur thoracique,</b> <b>Fréquence cardiaque rapide,</b> <b>Fièvre</b>
<b>R2</b> : Si <b>fièvre</b> et <b>douleurs articulaires</b> , alors <b>grippe</b> .	<b>Douleurs articulaires</b>

**R1, est-elle applicable ? / Tous les faits de R1 sont-ils dans la base des faits ?**



**R2, est-elle applicable ? / Tous les faits de R2 sont-ils dans la base des faits ?**



## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

✚ **Stratégie de priorité ou de confiance** : en attribuant des **coefficients de priorité/confiance** aux règles en fonction de leur **fiabilité ou importance**, cette stratégie assure que **les conditions les plus graves/importantes sont traitées en priorité**.

**Exemple : Système expert d'aide au diagnostic médical**

Base de règles	Base de faits ( <b>Utilisateur 1</b> )
<b>R1</b> : Si <b>douleur thoracique</b> et <b>fréquence cardiaque rapide</b> , alors <b>crise cardiaque</b> .	<b>Douleur thoracique,</b> <b>Fréquence cardiaque rapide,</b> <b>Fièvre</b>
<b>R2</b> : Si <b>fièvre</b> et <b>douleurs articulaires</b> , alors <b>grippe</b> .	<b>Douleurs articulaires</b>

**R1** indique **une condition potentiellement mortelle (crise cardiaque)** => **R1** doit avoir **une priorité plus élevée que R2** qui indique une condition moins grave (grippe).

## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

- ✚ **Méta-règles de résolution de conflit** : sont des règles introduites par les experts pour déterminer comment un système expert doit gérer les situations de conflits.
- ✚ P. ex. **La règle de précedence (critère de l'ordre)** : on choisit la règle applicable qui apparaît en premier dans la base de connaissances sera sélectionné.

**SI R1 et R2 ALORS R1.**

**SI R5 et R8 ALORS R5.**



## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

- ✚ **Méta-règles de résolution de conflit** : sont des règles introduites par les experts pour déterminer comment un système expert doit gérer les situations de conflits.
- ✚ P. ex. **La règle de spécificité** : on choisit la règle applicable la plus spécifique dans sa description.

Base de règles	Base de faits
R1: Si <b>personne atteinte de diabète</b> , alors les fruits sont un bon choix.	Personne atteinte de diabète
R2 : Si <b>personne atteinte de diabète</b> , alors les pommes sont un bon choix.	

R1 et R2 sont applicables, mais R2 **est plus spécifique que** R1 => R2 sera sélectionnée

## Cycle de base d'un moteur d'inférence

### Etape de résolution de conflits

✚ **Méta-règles de résolution de conflit** : sont des règles introduites par les experts pour déterminer comment un système expert doit gérer les situations de conflits.

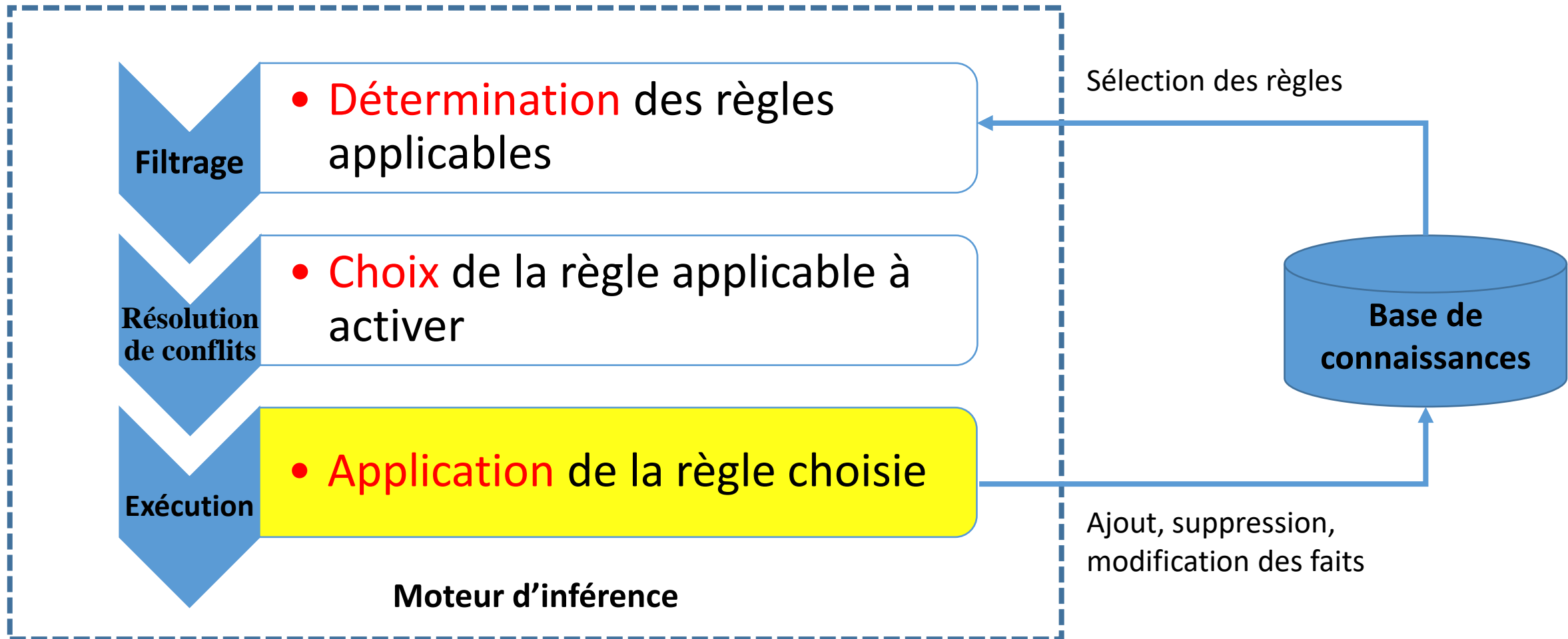
✚ Autres règles:

➤ **La récence** : on choisit la règle applicable qui utilise les faits les plus récemment déduits.

➤ **La fréquence** : on choisit la règle applicable qui a été appliquée le moins souvent.

➤ ...

## Cycle de base d'un moteur d'inférence



## Cycle de base d'un moteur d'inférence

### Etape d'exécution

- ✚ Il existe plusieurs types de moteurs d'inférence selon **la logique formelle** utilisée et **le mode du raisonnement** adopté.
- ✚ **Type de la logique formelle :**
  - **Un moteur d'inférence d'ordre 0** est basé sur la logique des propositions, qui ne manipule que des valeurs booléennes (vrai ou faux).
  - **Moteur d'inférence d'ordre 0+** est basé sur l'extension de l'ordre 0 avec des prédicats, des constantes et des fonctions, **mais sans l'utilisation de quantificateurs**.
  - **Moteur d'inférence d'ordre 1** est basé sur la logique des prédicats (prédicats, fonctions, quantificateurs) pour permettre le raisonnement sur des situations plus complexes.
  - ....

## Cycle de base d'un moteur d'inférence

### Etape d'exécution

- Il existe plusieurs types de moteurs d'inférence selon **la logique formelle** utilisée et **le mode du raisonnement** adopté.

Type de moteur d'inférence	Variables	Quantificateurs	Expressivité
<b>Ordre 0</b>	✗	✗	Basique (propositions)
<b>Ordre 0+</b>	☑	✗	Moyenne (prédicats sans généralisation)
<b>Ordre 1</b>	☑	☑	Élevée (prédicats avec généralisation)

## Cycle de base d'un moteur d'inférence

### Etape d'exécution

- ✚ Il existe plusieurs types de moteurs d'inférence selon **la logique formelle** utilisée et **le mode du raisonnement** adopté.
- ✚ **Mode du raisonnement :**
  - **Le chaînage avant (forward chaining)** : un mécanisme d'inférence **guidé par les faits**
  - **Le chaînage arrière (backward chaining)** : un mécanisme d'inférence **guidé par les buts**
  - **Le chaînage mixte (mixed chaining)** : un mécanisme d'inférence qui combine le chaînage avant et le chaînage arrière.

## Modes de raisonnement



Nous faisons l'hypothèse selon laquelle : « une règle dans la base de règles se met sous la forme **d'une règle de production** »:

**SI** condition **Alors** action

La partie gauche est une **premise**

La partie droite est une **conclusion**.

**SI** l'animal vole et l'animal pond des œufs **ALORS** l'animal est un oiseau

## Modes de raisonnement

Démarche 1: avec **les ingrédients (les faits)** qu'est-ce que je peux faire ?



## Chânage-avant

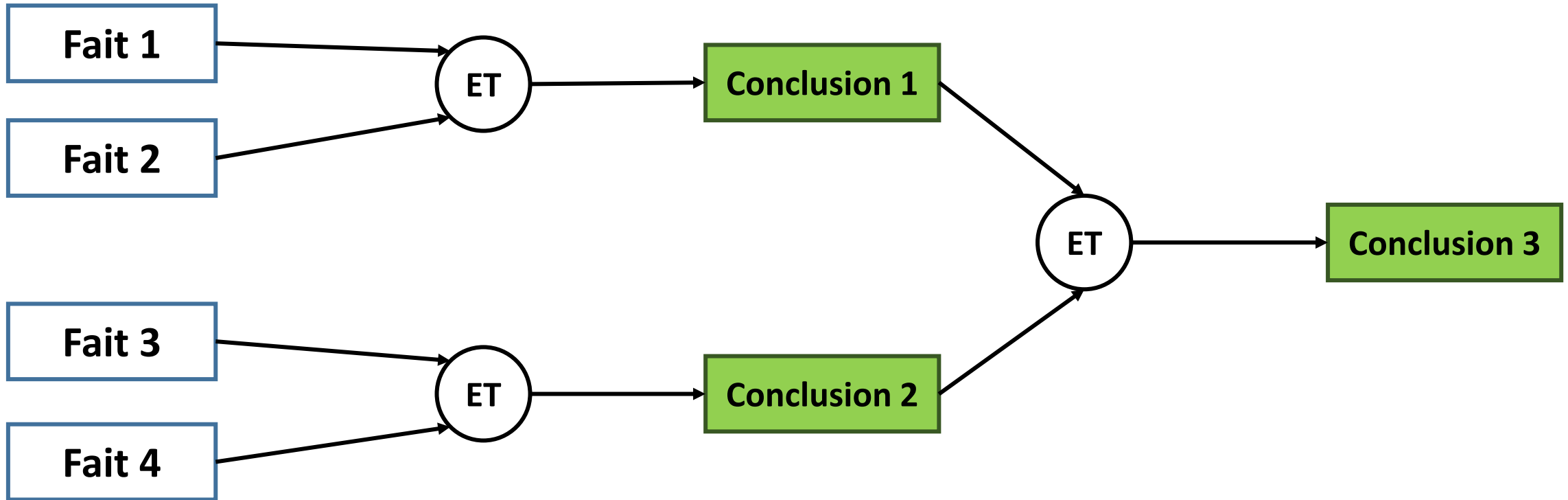


Un mécanisme d'inférence qui applique **les règles d'inférences** sur **les connaissances disponibles** (faits/réalité) pour extraire de nouvelles connaissances jusqu'à ce que **un but soit atteint**.



## Modes de raisonnement

### Chaînage-avant



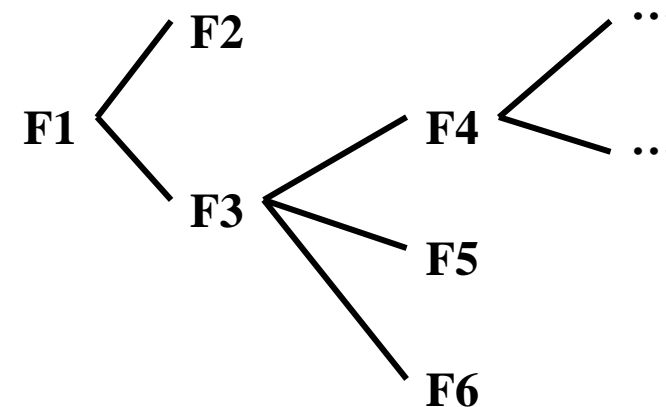
## Modes de raisonnement

### Chaînage-avant

- ✚ Un mécanisme d'inférence **guidé par les faits** : il applique **les règles d'inférences** sur **les connaissances disponibles** (les faits/réalités) pour extraire de nouvelles connaissances afin d'atteindre **un but précis** (il va des faits vers les buts).
- ✚ La traduction du **modus ponens** (raisonnement direct/implication) : **SI (F1 → F2 et F1 est vrai ) alors F2 est vrai.**

La règle                      Le fait

- ✚ Le chaînage-avant traduit un raisonnement **déductif** : de F1 sont déduits F2 et F3, de F3 sont déduits F4, F5 et F6, etc.



## Modes de raisonnement

### Chaînage-avant

Les règles applicables sont celles dont toutes les prémisses (partie gauche) sont satisfaites.

#### Algorithme Chaînage-avant

Entrée : BF = base de faits; BR = base de règles; FD = fait à déduire

#### Début

→ Déterminer les règles applicables (**Filtrage**)

**TQ** (FD n'est pas dans BF et il existe une règle applicable) **Faire**

Choisir une règle R applicable dans BR (**Résolution des conflits**)

Appliquer la règle choisie R (**Exécution**)

$BR = BR - R$  (**Désactivation de la règle**)

$BF = BF + \text{Conclusion (R)}$

Modifier l'ensemble des règles applicables

#### FTQ

**Si** (FD appartient à BF) **Alors**

FD déduit

#### Sinon

FD ne peut pas être déduit

#### FSi

#### Fin

## Modes de raisonnement

### Chaînage-avant

### Quand s'arrêter ?

✚ Le but souhaité est-il démontré ? : cet algorithme suppose que l'utilisateur du moteur veut **obtenir une proposition particulière** (but du problème). Si ce fait vient d'être obtenu, il est inutile de poursuivre le travail. Exemple : **Peut-on obtenir une tarte aux cerises avec les ingrédients disponibles ?**

✚ Dans le cas où **aucun but particulier n'est demandé**, le moteur fonctionne jusqu'au moment où aucune règle n'est applicable (condition d'arrêt). On dit alors que **le moteur fonctionne par saturation**. Exemple : **Qu'est-ce que je peux faire avec les ingrédients (faits) disponibles ?**

## Modes de raisonnement

Le système expert « pâtisserie » :

Base de règles	Base de faits
<b>R1</b> : Si <b>farine</b> et <b>beurre</b> et <b>œufs</b> et <b>sel</b> alors <b>pâte</b>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel
<b>R2</b> : Si <b>pommes</b> et <b>sucre</b> alors <b>pommes sucrées</b>	
<b>R3</b> : Si <b>pommes sucrées</b> et <b>pâte</b> alors <b>tarte aux pommes</b>	
<b>R4</b> : Si <b>abricots</b> et <b>pâte</b> alors <b>tarte aux abricots</b>	
<b>R5</b> : Si <b>poires</b> et <b>pâte</b> alors <b>tarte aux poires</b>	
<b>R6</b> : Si <b>cerises</b> et <b>pâte</b> alors <b>tarte aux cerises</b>	

Comment fonctionne en pratique un système expert ?

## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<b>R1</b> : Si farine et beurre et œufs et sel alors pâte	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel
<b>R2</b> : Si pommes et sucre alors pommes sucrées	
<b>R3</b> : Si pommes sucrées et pâte alors tarte aux pommes	
<b>R4</b> : Si abricots et pâte alors tarte aux abricots	
<b>R5</b> : Si poires et pâte alors tarte aux poires	
<b>R6</b> : Si cerises et pâte alors tarte aux cerises	

Résolution de conflits : la règle de précedence (ordre d'écriture des règles)

**R1, est-elle applicable ? / Tous les faits de R1 sont-ils dans la base des faits ?**

**Oui alors** : le fait **pâte (conclusion de R1)** est donc rajouté dans la base des faits et **R1 est désactivée** (une règle peut seulement être activée une fois)

## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<b>R1</b> : Si <del>farine</del> et <del>beurre</del> et <del>œufs</del> et <del>sel</del> alors <del>pâte</del>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel, <b>pâte</b>
<b>R2</b> : Si <b>pommes</b> et <b>sucre</b> alors <b>pommes sucrées</b>	
<b>R3</b> : Si <b>pommes sucrées</b> et <b>pâte</b> alors <b>tarte aux pommes</b>	
<b>R4</b> : Si <b>abricots</b> et <b>pâte</b> alors <b>tarte aux abricots</b>	
<b>R5</b> : Si <b>poires</b> et <b>pâte</b> alors <b>tarte aux poires</b>	
<b>R6</b> : Si <b>cerises</b> et <b>pâte</b> alors <b>tarte aux cerises</b>	

Résolution de conflits : la règle de précedence (ordre d'écriture des règles)

**R2, est-elle applicable ? / Tous les faits de R2 sont-ils dans la base des faits ?**

**Oui alors** : le fait **pommes sucrées** (conclusion de **R2**) est donc rajouté dans la base des faits et **R2** est désactivée.

## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<b>R1</b> : Si <del>farine</del> et <del>beurre</del> et <del>œufs</del> et <del>sel</del> alors <del>pâte</del>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel, <b>pâte</b> <b>pommes sucrées</b>
<b>R2</b> : Si <del>pommes</del> et <del>sucré</del> alors <del>pommes sucrées</del>	
<b>R3</b> : Si <b>pommes sucrées</b> et <b>pâte</b> alors <b>tarte aux pommes</b>	
<b>R4</b> : Si <b>abricots</b> et <b>pâte</b> alors <b>tarte aux abricots</b>	
<b>R5</b> : Si <b>poires</b> et <b>pâte</b> alors <b>tarte aux poires</b>	
<b>R6</b> : Si <b>cerises</b> et <b>pâte</b> alors <b>tarte aux cerises</b>	

Résolution de conflits : la règle de précedence (ordre d'écriture des règles)

**R3, est-elle applicable ? / Tous les faits de R3 sont-ils dans la base des faits ?**

**Oui alors** : le fait **tarte aux pommes (conclusion de R3)** est donc rajouté dans la base des faits et **R3** est désactivée.



## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<b>R1</b> : Si <del>farine</del> et <del>beurre</del> et <del>œufs</del> et <del>sel</del> alors <b>pâte</b>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel, <b>pâte</b> <b>pommes sucrées, tarte aux pommes</b>
<b>R2</b> : Si <del> pommes</del> et <del>sucre</del> alors <b> pommes sucrées</b>	
<b>R3</b> : Si <del> pommes sucrées</del> et <del> pâte</del> alors <b> tarte aux pommes</b>	
<b>R4</b> : Si <b> abricots</b> et <b> pâte</b> alors <b> tarte aux abricots</b>	
<b>R5</b> : Si <b> poires</b> et <b> pâte</b> alors <b> tarte aux poires</b>	
<b>R6</b> : Si <b> cerises</b> et <b> pâte</b> alors <b> tarte aux cerises</b>	

Résolution de conflits : la règle de précedence (ordre d'écriture des règles)

**R4, est-elle applicable ? / Tous les faits de R4 sont-ils dans la base des faits ?**

**Oui alors** : le fait **tarte aux abricots (conclusion de R4)** est donc rajouté dans la base des faits et **R4** est désactivée.

## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<del>R1</del> : Si <del>farine</del> et <del>beurre</del> et <del>œufs</del> et <del>sel</del> alors <del>pâte</del>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel, <b>pâte</b> <b>pommes sucrées, tarte aux pommes</b> <b>tarte aux abricots</b>
<del>R2</del> : Si <del>pommes</del> et <del>sucré</del> alors <del>pommes sucrées</del>	
<del>R3</del> : Si <del>pommes sucrées</del> et <del>pâte</del> alors <del>tarte aux pommes</del>	
<del>R4</del> : Si <del>abricots</del> et <del>pâte</del> alors <del>tarte aux abricots</del>	
R5 : Si <b>poires</b> et <b>pâte</b> alors <b>tarte aux poires</b>	
R6 : Si <b>cerises</b> et <b>pâte</b> alors <b>tarte aux cerises</b>	

Résolution de conflits : la règle de précedence (ordre d'écriture des règles)

**R5, est-elle applicable ? / Tous les faits de R5 sont-ils dans la base des faits ?**

**Oui alors** : le fait **tarte aux poires (conclusion de R5)** est donc rajouté dans la base des faits et **R5** est désactivée.

## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<b>R1</b> : Si <del>farine</del> et <del>beurre</del> et <del>œufs</del> et <del>sel</del> alors <del>pâte</del>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel, <b>pâte</b> <b>pommes sucrées, tarte aux pommes</b> <b>tarte aux abricots, tarte aux poires</b>
<b>R2</b> : Si <del>pommes</del> et <del>sucré</del> alors <del>pommes sucrées</del>	
<b>R3</b> : Si <del>pommes sucrées</del> et <del>pâte</del> alors <del>tarte aux pommes</del>	
<b>R4</b> : Si <del>abricots</del> et <del>pâte</del> alors <del>tarte aux abricots</del>	
<b>R5</b> : Si <del>poires</del> et <del>pâte</del> alors <del>tarte aux poires</del>	
<b>R6</b> : Si <del>cerises</del> et <del>pâte</del> alors <del>tarte aux cerises</del>	

Résolution de conflits : la règle de précedence (ordre d'écriture des règles)

**R6, est-elle applicable ? / Tous les faits de R6 sont-ils dans la base des faits ?**

**Non** (le fait "cerises" n'existe pas dans la BF) alors Le mécanisme ne peut plus effectuer de déductions et **le moteur d'inférence s'arrête**. La BF est saturée.

## Modes de raisonnement

### Chaînage-avant

Base de règles	Base de faits
<b>R1</b> : Si <del>farine</del> et <del>beurre</del> et <del>œufs</del> et <del>sel</del> alors <b>pâte</b>	Œufs, Pommes , Poires Abricots, Farine Beurre, Sucre Sel, <b>pâte</b> <b>pommes sucrées, tarte aux pommes</b> <b>tarte aux abricots, tarte aux poires</b>
<b>R2</b> : Si <del>potommes</del> et <del>sucre</del> alors <b>potommes sucrées</b>	
<b>R3</b> : Si <del>potommes sucrées</del> et <del>pâte</del> alors <b>tarte aux pommes</b>	
<b>R4</b> : Si <del>abricots</del> et <del>pâte</del> alors <b>tarte aux abricots</b>	
<b>R5</b> : Si <del>poires</del> et <del>pâte</del> alors <b>tarte aux poires</b>	
<b>R6</b> : Si <del>cerises</del> et <del>pâte</del> alors <b>tarte aux cerises</b>	

Avec **les ingrédients (faits)** qu'est-ce que je peux faire ?

Peut-on obtenir **une tarte aux cerises (but)** avec les ingrédients  
(faits) disponibles ?

## Modes de raisonnement

### Chaînage-avant

✚ **Avantage** : la réception de nouvelles données peut déclencher de nouvelles inférences, ce qui rend le moteur plus dynamique.

✚ **Inconvénients** :

- Il déclenche toutes les règles, même celles sans intérêt ;
- la base de faits doit contenir suffisamment de faits;
- Il est facilement explosif, ce qui risque de provoquer une explosion combinatoire.

## Exercice

**Jijel** est une ville au bord de la mer méditerranée qui offre de superbes espaces verdoyants.

Avec ces avenues larges, ces grottes touristiques, et les plages de la corniche, elle demeure une véritable curiosité pour les touristes.

**Jijel mérite-t-elle d'être visitée ?**

Base de règles	Base de faits
R1 : Si belle ville et bon restaurants alors ville méritant une visite.	
R2 : Si ville historique alors ville méritant une visite.	
R2 : Si ville maritime et ville touristique alors ville méritant une visite.	
R3 : Si ville maritime et belle ville alors ville méritant une visite.	
R4 : Si végétation abondante et ville maritime alors belle ville	
R5 : Si tradition culinaire et restaurants 3 étoiles alors bons restaurants.	
R6 : Si musées et ville ancienne alors ville historique.	
R7 : Si parc animalier et grottes touristiques alors ville touristique.	
R8 : Si monuments célèbres et moyens de transport alors ville touristique.	
R9 : Si espaces verdoyants et avenues larges alors végétation abondante.	
R10 : Si corniche et bord de mer alors ville maritime.	