# MooseCritics, outil d’analyse de règles architecturales

Présenté par Romain Degrave

Université de Lille

RMod

Inria

M. Romain Rouvoy

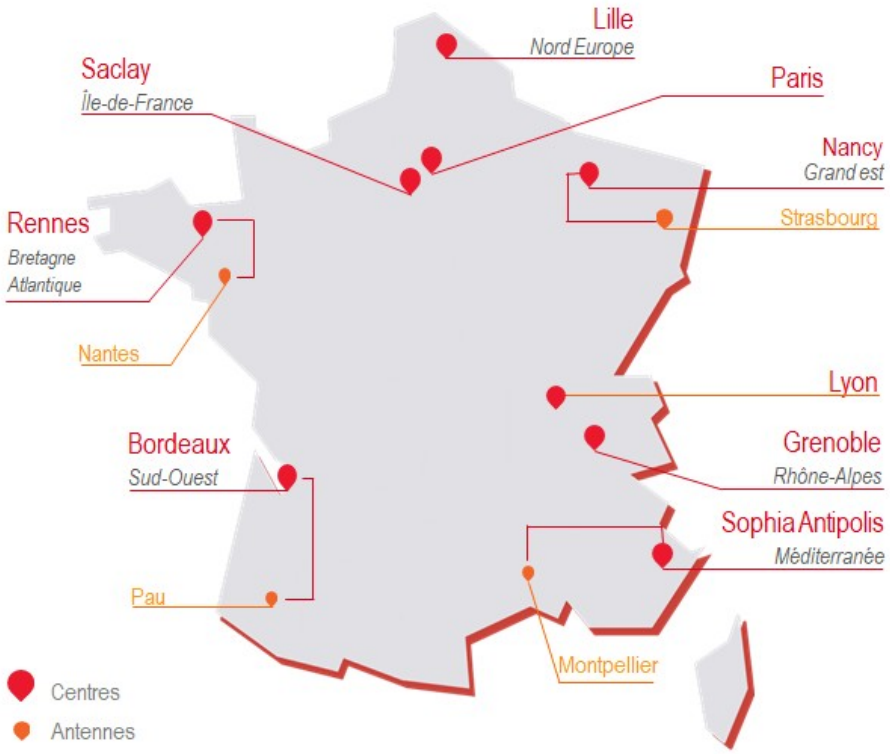
Mme. Anne Etien

# Inria

- 10 centres

- 200 équipes-projets

- 3900 scientifiques

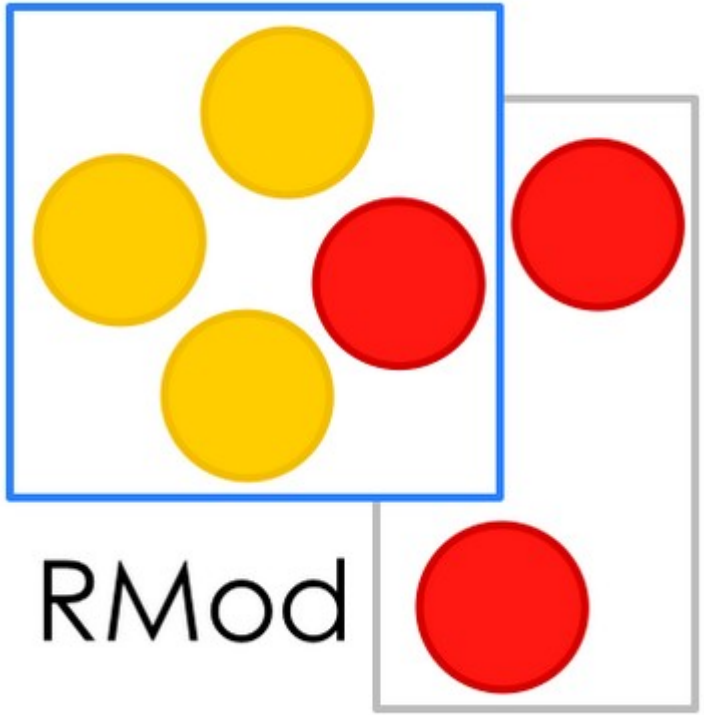


# RMod

- Spécialisée en Génie Logiciel

- Analyse (métriques, visualisations)

- Définition de constructeurs (traits)

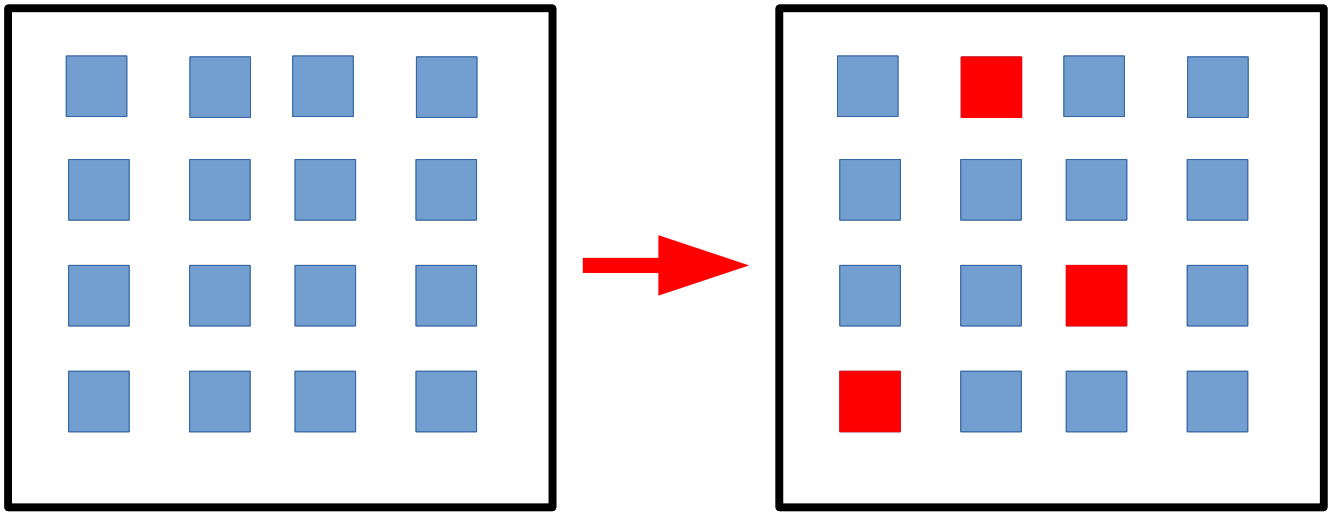


# Pharo / Moose

- Dérivé de Smalltalk
- Simple, interactif
- Moose : dédié à l’analyse


# Analyse des règles architecturales





# MooseCritics

- Outil intégré à Moose

- Écrire des règles

- Exécuter les règles sur le logiciel

- Récupérer les violations

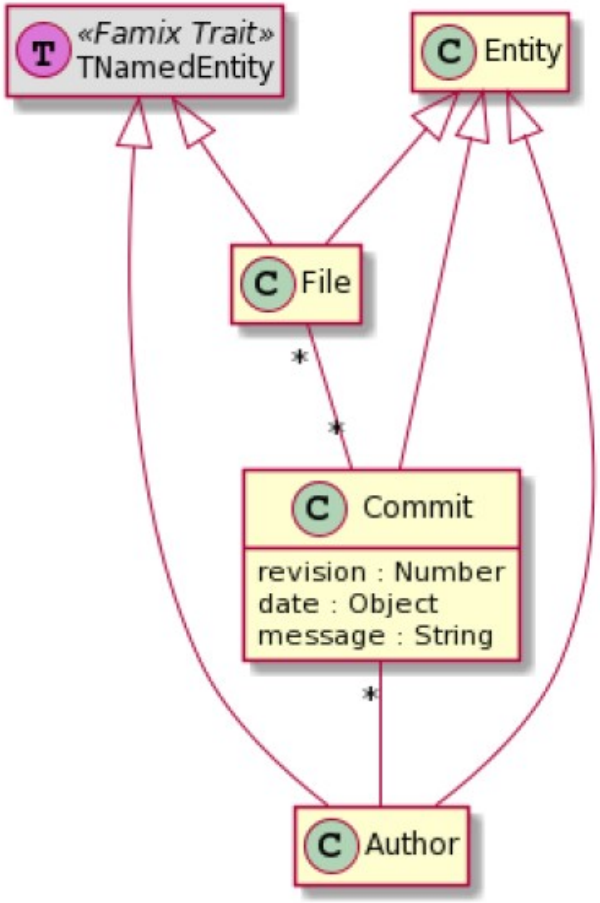
# Plan

- Méta-modèle et MooseIDE
- Contributions
  - Présentation générale
  - Intégration dans MooseIDE
  - Utilisation du Queries Browser
  - Représentation et traitement des règles
  - MooseCritics en application
- Conclusion

- **Méta-modèle et MooseIDE**
- Contributions
  - Présentation générale
  - Intégration dans MooseIDE
  - Utilisation du Queries Browser
  - Représentation et traitement des règles
  - MooseCritics en application
- Conclusion

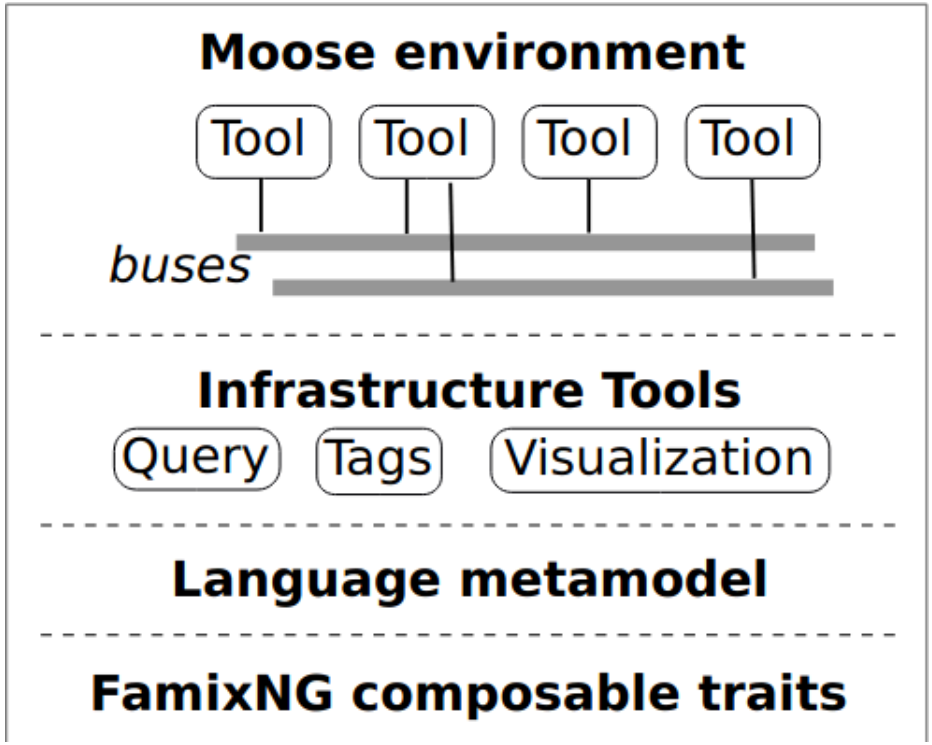
# Méta-modèles Famix

- Modèle : représente le logiciel
- Méta-modèle : structure du modèle
- Composé grâce aux traits
- Outils « universels »




*Source : Modular Moose: A new generation of software reengineering platform, (ICSR'20), Nicolas Anquetil*

# MooseIDE



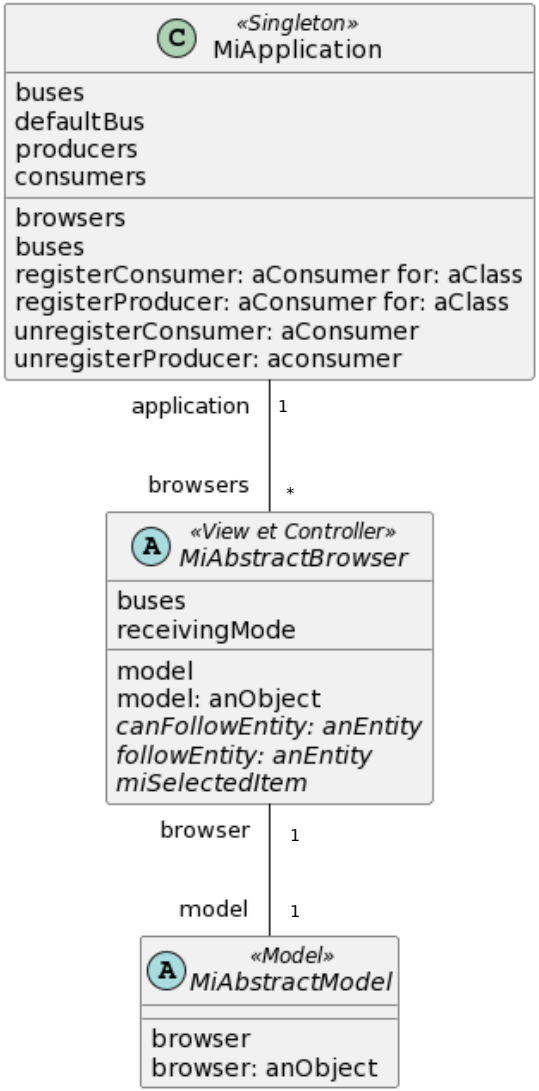

*Source : Modular Moose: A new generation software reverse engineering environment, (ICSR'20), Nicolas Anquetil*

# Architecture d’un browser

- Lier les outils
(bus, *producer / consumer*)

- Fonctionnement général
  de l’outil

- Gère données et traitements



- Méta-modèle et MooseIDE
- Contributions
  - **Présentation générale**
  - Intégration dans MooseIDE
  - Utilisation du Queries Browser
  - Représentation et traitement des règles
  - MooseCritics en application
- Conclusion

# Fenêtre principale

Model Critic Browser

Follow | Freeze | Highlight | Bus: Default | Inspect | Propagate | Help

**Entities**

- 9 AnnotationTypeAttributes(9 FamixStAnnotationTypeAttribute)
- 13 AnnotationTypes(13 FamixStAnnotationType)
- 70 Attributes(70 FamixStAttribute)
- 26 Classes(26 FamixStClass)
- 1 GlobalVariable(1 FamixStGlobalVariable)
- 171 ImplicitVariables(171 FamixStImplicitVariable)
- 149 LocalVariables(149 FamixStLocalVariable)
- 253 Methods(253 FamixStMethod)
- 2 Namespaces(2 FamixStNamespace)
- 6 Packages(6 FamixStPackage)
- 110 Parameters(110 FamixStParameter)

**Rules**

Une classe ne doit pas dépasser les 20 méthodes (2)

**Entities requested**

- 2 Classes
  - MiCriticBrowser -> Une classe ne doit pas dépasser les 20 méthodes
  - FamixCBRule -> Une classe ne doit pas dépasser les 20 méthodes

Inspect model | Import rules | Export rules | Add rules | Run

# Représentation des règles

Rule Editor

Pharo Code  Queries Browser

Rule name :

Services name must end with 'Intf' suffix

Input for code query :

```
1 [ :entity |
2     (entity isInterface and: [
3         entity superclassHierarchy anySatisfy: [ :class |
4             class name = 'Remote' ] ]) and: [
5         (entity name endsWith: 'Intf') not ] ]
```

Summary of the rule :

The name of a service must end with the suffix "Intf".

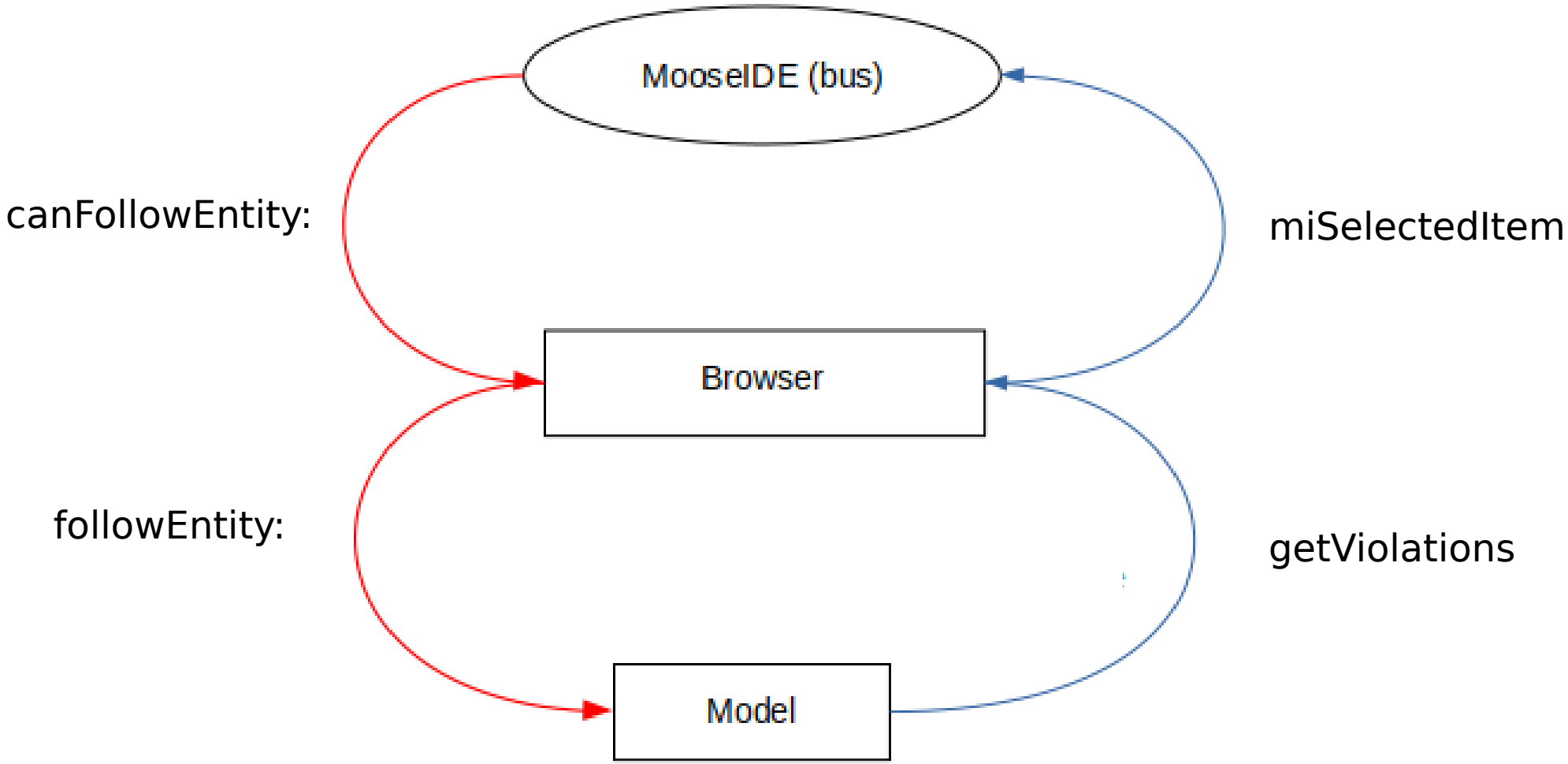
Edit rule  × Cancel edit

Contexte de la règle

Condition

- Méta-modèle et MooseIDE
- Contributions
  - Présentation générale
  - **Intégration dans MooseIDE**
  - Utilisation du Queries Browser
  - Représentation et traitement des règles
  - MooseCritics en application
- Conclusion

# Intégration dans MooseIDE



- Moose : IDE et méta-modèle
- Contributions
  - Présentation générale
  - Intégration dans MooseIDE
  - **Utilisation du Query Browser**
  - Représentation et traitement des règles
  - MooseCritics en application
- Conclusion

# Producteur / Consommateur

Enregistrement comme consommateur

MiCriticBrowser>>initialize

Comment | MiCriticBrowser | initialize | UML-Class | Inst. side methc

```
initialize
    super initialize.
    application registerConsumer: addRulesPresenter for: FQAbstractQuery.
    application registerConsumer: rulesEditorPresenter for: FQAbstractQuery
```

MiCriticBrowserAddRulesPresenter>>consume:

Comment | MiCriticBrowser | consume: | UML-Class | Inst. side methc

```
consume: aQuery
    queryPresenter queryDroplist items: self fetchQueries
```

Récupération des requêtes

- Méta-modèle et MooseIDE
- Contributions
  - Présentation générale
  - Intégration dans MooseIDE
  - Utilisation du Queries Browser
  - **Représentation et traitement des règles**
  - MooseCritics en application
- Conclusion

# Méta-modèle des règles

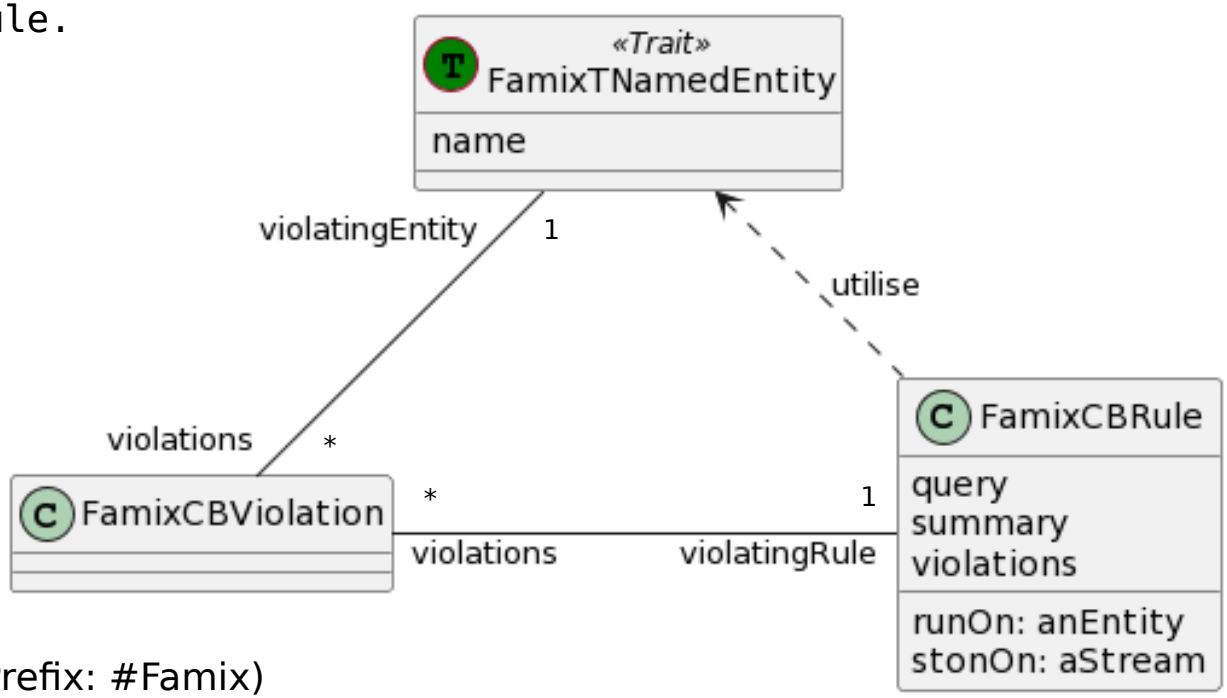
```
violation := builder newClassNamed: #Violation.
rule := builder newClassNamed: #Rule.

"Inheritance"
rule --|> #TNamedEntity.

"Properties"
rule property: #summary type: #String.

"Associations"
(violation property: #violatedRule)
          *-
(rule property: #violations)

(violation property: #violatingEntity)
          *-
((self remoteTrait: #TNamedEntity withPrefix: #Famix)
property: #violations).
```



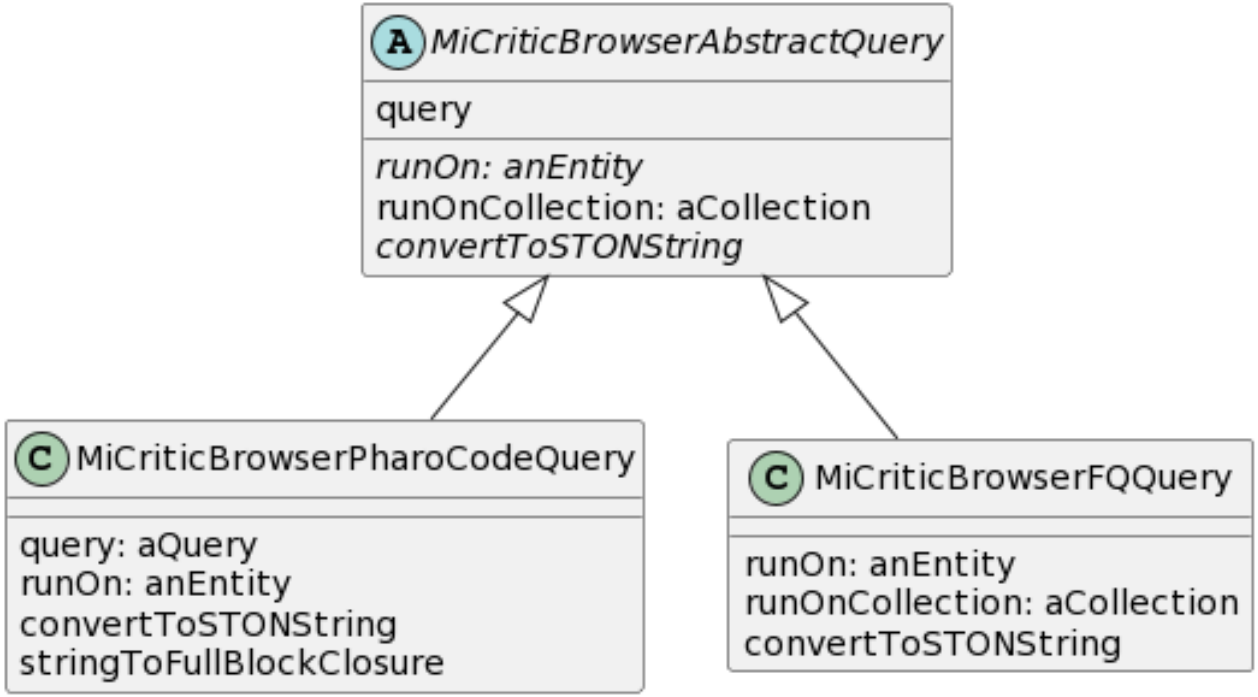
# Exécution des règles

- Requête booléenne

- Permet d'autres « syntaxes »

- Polymorphisme

- Si résultat = *true* → Violation



- Méta-modèle et MooseIDE
- Contributions
  - Présentation générale
  - Intégration dans MooseIDE
  - Utilisation du Queries Browser
  - Représentation et traitement des règles
  - **MooseCritics en application**
- Conclusion

# Architecture

- Interface, étend Remote

Services

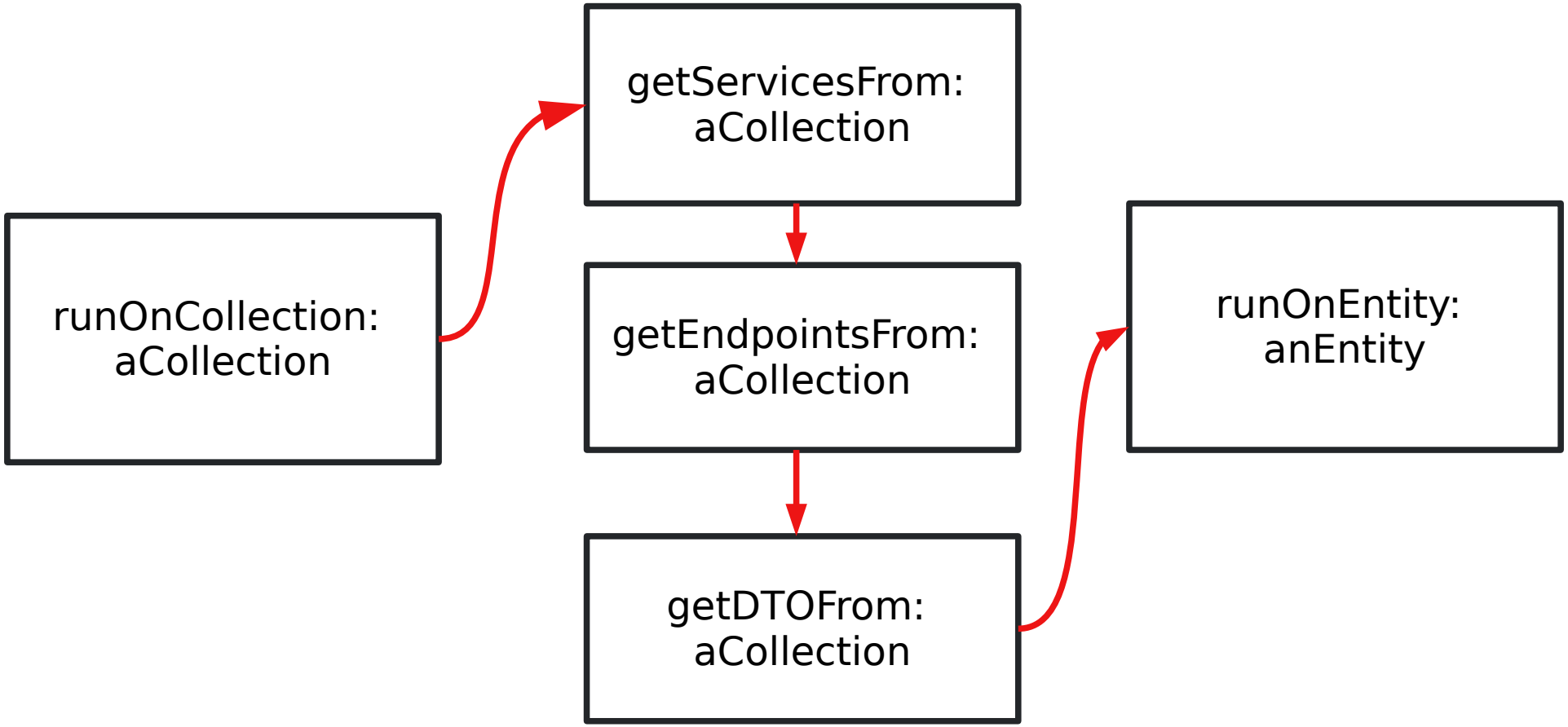
- Méthodes de Services

Endpoints

- Paramètres, return d’endpoints
- Attributs de DTO = DTO (récursion)

DTO

# Execution de règle DTO



- Méta-modèle et MooseIDE
- Contributions
  - Présentation générale
  - Intégration dans MooseIDE
  - Utilisation du Queries Browser
  - Représentation et traitement des règles
  - MooseCritics en application
- **Conclusion**

# Bilan professionnel

Rules

- Services must have an implementation (2)
- Services name must end with 'Intf' suffix (2)
- Services implementations must have the same name, except an 'Impl' suffix (3)
- A DTO must have an empty public constructor (292)
- A DTO must have a getter and setter for every attribute (79)
- A DTO can't have a getter without access (113)
- A DTO must inherit from a particular class (56)
- Subclasses of EnumerationAbstract must implement getAll (90)
- Post-migration invalid DTOs (13)

- Utile pour des entreprises

- Suite : Amélioration de l’intégration

# Bilan personnel

- Design Pattern, TDD

- Communication avec professionnels