

Ce document :

- ☞ doit être ouvert avec un lecteur PDF supportant les liens hypertextes ;
- ☞ est amené à évoluer, les modifications pouvant être suivies sur son **dépôt Github**.

# Table des matières

|         |   |    |
|---------|---|----|
| I       | Principes généraux du modèle conceptuel e-ReColNat augmenté | 2  |
| 1.1     | Introduction  | 3  |
| 1.2     | Terminologie  | 3  |
| 1.2.1   | Le modèle « e-RecolNat+ »                                   | 3  |
| 1.2.2   | La base « e-ReColNat+ »                                     | 4  |
| 1.3     | Les entités de base du modèle e-RecolNat+                   | 4  |
| 1.3.1   | Entité abstraite ReColnat (EAR)                             | 4  |
| 1.3.2   | Entité scientifique abstraite (ESA)                         | 5  |
| 1.3.3   | Situation   | 6  |
| 1.3.4   | Sous-types d'EARs   | 7  |
| 1.3.4.1 | Entité feuille abstraite ReColnat (EFAR)                    | 7  |
| 1.3.4.2 | Entité composite abstraite ReColnat (ECAR)                  | 7  |
| 1.4     | Le modèle de tags   | 7  |
| 1.5     | Le cas des discussions                                      | 8  |
| 1.5.1   | Discussions ciblées & discussions générales                 | 8  |
| 1.5.2   | Le modèle de discussions                                    | 9  |
| II      | Les APIs de l'écosystème e-RecolNat                         | 10 |
| 2.1     | Principes généraux des API                                  | 11 |
| 2.1.1   | Proposition de principe général                             | 11 |
| 2.1.2   | Avantages futurs en terme de gestion de projet              | 11 |
| 2.2     | Les APIs « tags » et « discussions »                        | 12 |

# I

## PRINCIPES GÉNÉRAUX DU MODÈLE CONCEPTUEL E-RECOLNAT AUGMENTÉ

# 1.1 Introduction

Cette partie présente les fondements et les entités de base du modèle conceptuel e-ReColNat augmenté afin de préciser dans quel cadre logique nous pensons les rapports entre les modules dont le développement nous incombe et les autres modules du portail coopératif. Les informations sur le modèle exposées dans ce document sont donc partielles, et insistent sur l'esprit d'ensemble plus que sur le détail des diverses classes de sous entités métier.

## 1.2 Terminologie

### 1.2.1 Le modèle « e-RecolNat+ »

Le Modèle conceptuel « e-ReColNat augmenté » — dont le nom est initialement apparu sous les formes « modèle e-ReColNat+ » ou « modèle augmenté », confirmées au fil de plusieurs réunions avec Julien Husson — inventorie et organise l'ensemble des concepts, propriétés et relations sur lesquels reposent les modules développés par le laboratoire DICEN/WP5 (collaboratoire, visite virtuelle, portail, et toutes les interfaces de contribution et de navigation documentaires et sociales qui les constituent ou qu'ils supposent). Le prédicat « augmenté » dénote que ce modèle conceptuel a pour enjeu de fournir des strates d'enrichissements supplémentaires aux entités botaniques identifiées dans le modèle de données du WP2 — nourrie notamment par l'IPT du GBIF —, aux objets des Herbonautes v2, et plus généralement, aux objets métier de l'ensemble des applications liées à e-ReColNat. Les « augmentations » afférentes peuvent être d'ordre documentaire et social. Il contient toutefois des entités qui lui sont propres ; pour exemples, les tags, les discussions, ou encore les entités convoquées dans les visites virtuelles, chacun de ces concepts étant liés à une ou plusieurs entités plus directement botaniques issues des modèles métier des applications du Muséum et des autres partenaires. Ce modèle procède, d'une part, de l'analyse des pratiques et discours de botanistes professionnels et amateurs<sup>I</sup>, et d'autre part, d'une analyse plus globale des gestes savants portant sur des documents multimédias hyperliés en contexte de travail coopératif mettant en jeu les compétences scientifiques du laboratoire DICEN.

---

<sup>I</sup> Menée par Lisa Chupin.

## 1.2.2 La base « e-ReColNat+ »

La Base de données « e-ReColNat augmentée » réalise la persistance des objets identifiées par le modèle e-ReColNat+ (les informations constituant leur identité, leurs propriétés intrinsèques, les données qui les définissent et leurs sont directement rattachées, ainsi que les relations qu'elles entretiennent les unes avec les autres). Différentes APIs (d'extraction et d'écriture) seront proposées pour rendre possible l'exploitation de certaines fonctions de cette base au sein de l'ensemble des modules du portail<sup>2</sup>. Voir II.

## 1.3 Les entités de base du modèle e-ReColNat+

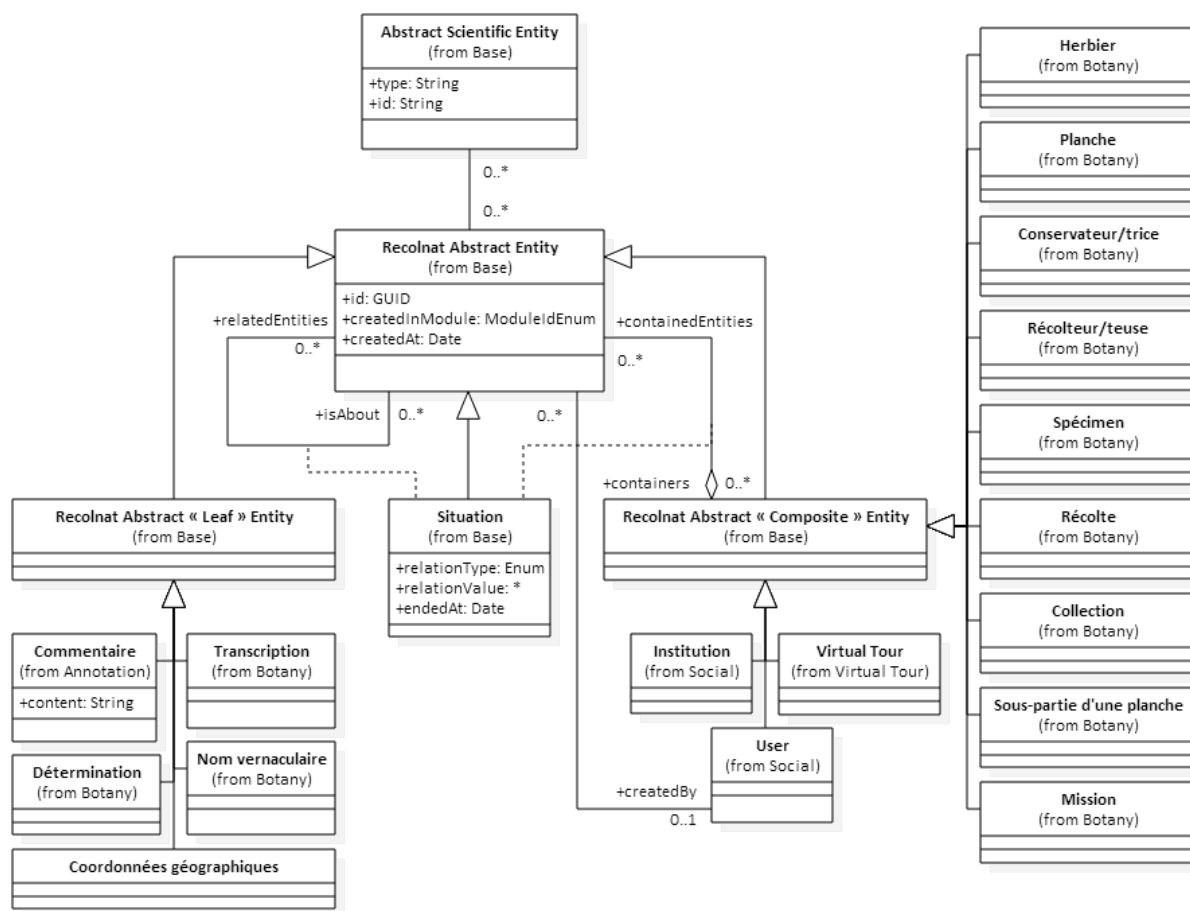
Le modèle e-ReColNat+ repose sur un petit ensemble d'entités abstraites destinées à être déclinées en entités métier et auxquelles elles fournissent des possibilités d'annotation et de structuration socialement et temporellement situées et scientifiquement qualifiées. Sur la figure 3.1, ces entités sont marquées comme issues du groupe **Base**. Les concepts présentés ici, qui sont très abstraits, seront exemplifiés en 1.4 et 1.5.2.

### 1.3.1 Entité abstraite ReColnat (EAR)

Le concept d'entité ReColnat abstraite fournit le type fondamental du modèle e-ReColNat+ dont découle tout objet d'intérêt (à l'exception des entités scientifiques abstraites, voir 1.3.2). Ce type se définit par les caractéristiques intrinsèques et relationnelles suivantes (voir figure 3.1) :

- ✉ Une EAR possède un identifiant e-ReColNat+ qui lui confère une identité stable au sein du système indépendamment de son type métier concret.
- ✉ La création d'une EAR est située : tout objet sait à quel date, par qui (c'est-à-dire par quel utilisateur e-ReColNat) et dans quel module ou lieu du portail (collaboratoire, visite virtuelle, réseau social, Herbonautes, etc.) il a été créé.
- ✉ Une EAR peut être associée à une entité scientifique abstraite (voir 1.3.2), et donc se rapporter à quelque chose qui existe en dehors de son module d'origine voire en dehors du système d'information e-ReColNat.
- ✉ Une EAR peut se rapporter à une ou plusieurs EARs cibles (voir lien réflexif **isAbout/relatedEntities** sur le diagramme de la figure 3.1). Ce lien dénote le

<sup>2</sup> Et peut-être au-delà de ce périmètre, si cela s'avère pertinent.



**Figure 3.1** Diagramme de classes UML simplifié du modèle de base e-ReColNat+

positionnement fondamentalement annotationnel/critique du modèle e-ReColNat+, tout objet pouvant être relié à un autre par une relation hyperdocumentaire située et qualifiée (voir 1.3.3).

## 1.3.2 Entité scientifique abstraite (ESA)

Une entité scientifique abstraite représente un objet existant dans une base de données extérieure (base agrégée du WP2, ou toute autre base scientifique). L'identité d'une ESA se compose de deux propriétés : un **type** renvoyant au type de l'objet dans sa base de données scientifique de provenance, et un identifiant **id** à valeur dans cette base<sup>3</sup>. Tout objet extérieur à la base e-ReColNat+ peut ainsi être désigné de manière univoque.

<sup>3</sup> Il relève de la responsabilité des vues de construire des URLs d'accès effectif aux objets scientifique externes sur la base de ces deux champs.

### 1.3.3 Situation

La classe d'association **Situation** telle qu'elle apparaît sur le diagramme de la figure 3.1 fournit un contexte à la relation réflexive **isAbout** des EARs susmentionnée. Chaque instance de cette relation est ainsi enrichie en premier lieu d'un type (propriété **relationType**), dénotant la sémantique du lien documentaire ou critique qu'elle établit :

- ⌘ lien d'*enrichissement* (exemple : une annotation enrichit l'EAR sur laquelle elle porte) ;
- ⌘ lien d'*approbation* ou à l'inverse de *désaccord* (voir *infra*) ;
- ⌘ lien de *qualification* (exemple : un tag qualifie une EAR, voir 1.4) ;
- ⌘ lien d'*association* (exemple : un extrait de planche délimité dans le collaboratoire peut être associé à des ressources complémentaires telles que des notes ou des photos, ce qui constitue le fondement d'un carnet botanique hyperdocumentaire et hypermédia) ;
- ⌘ etc. (il est possible d'étendre le modèle avec d'autres sémantiques de lien réalisant des opérations critiques non prévues initialement).

Ceci permet de typer les relations entre objets sans sacrifier l'unicité du graphe hyperdocumentaire auquel ils prennent part. De plus, une Situation est également une EAR (au sens d'une relation UML de généralisation), ce qui confère à sa création la possibilité d'être datée et signée (par un utilisateur identifié).

Par ailleurs, en temps qu'EAR, une Situation jouit d'une identité propre qui lui permet à son tour d'être prise pour objet par la relation **isAbout**. Cette capacité peut être utilisée pour représenter le souhait d'un utilisateur de qualifier l'action d'un autre utilisateur portant sur la mise en relation de deux EARs. Ceci appelle un exemple. Soit un utilisateur  $U_1$  — un aimable contributeur — qui propose une transcription  $T_1$  pour un champ  $C_1$  de l'étiquette  $E_1$  d'une planche  $P_1$ . La réalisation de cette action générera une Situation  $S_1$  situant cette action dans le temps (date de création), dans l'espace social (l'auteur  $U_1$ ) et dans l'espace documentaire (mise en relation de  $T_1$  à  $C_1$  via un lien de type « enrichissement »<sup>4</sup>). La Situation  $S_1$  jouissant d'une identité propre, il devient possible à un autre utilisateur  $U_2$  — un éminent botaniste reconnu comme tel par le composant d'authentification/habilitation — de donner son avis sur celle-ci, par l'intermédiaire d'un lien de type « Approbation ». La contribution  $T_1$  d' $U_1$  est ainsi marquée comme contribution de premier plan (car reconnue par un chercheur « habilité »), ce qui peut constituer un critère de sélection intéressant lorsqu'il est question de compléter une base de données scientifique à partir des données contributives d'e-ReColNat+.

Plus généralement, le couplage des Situations et des liens typés inter-EARs permet de constituer un réseau critique complexe.

<sup>4</sup> Une sémantique plus précise peut être admise, comme par exemple « Transcription », étant donné qu'il s'agit là d'un concept métier omniprésent.

## I.3.4 Sous-types d'EARs

Outre les Situations, qui ont un statut particulier du fait de leur fonction d'opérateurs de contextualisation de l'ensemble des opérations de mise en relation permises par le système, les EARs se déclinent en deux sous-types principaux : les entités feuilles abstraites Reclnat et les entités composites abstraites Reclnat. Ces entités, présentées *infra*, renvoient au **patron de conception « composite »**.

### I.3.4.1 Entité feuille abstraite Reclnat (EFAR)

Une entité feuille abstraite Reclnat représente une information monadique. Le diagramme de la figure 3.1 en donne les exemples suivants : commentaire simple, transcription, détermination, nom vernaculaire, coordonnées géographiques. Dans tous ces cas, le contenu de l'EFAR est indivisible, n'admet aucune sous-partie dotée d'une identité propre et jouissant d'un certain degré d'autonomie.

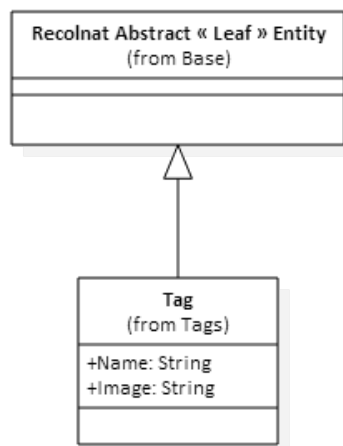
### I.3.4.2 Entité composite abstraite Reclnat (ECAR)

Une entité composite abstraite Reclnat admet un ensemble d'EAR définies comme ses sous-parties. Les types concrets métier du diagramme de la figure 3.1 héritant d'ECAR illustrent tous cette propriété structurelle : un herbier *contient* des planches, une planche *se compose de* sous-parties d'intérêt (étiquette, feuille identifiée, etc.), un/e conservateur/trice *a la responsabilité* d'un ou plusieurs herbiers, une récolte *contient* des spécimens, une collection *contient* des herbiers/planches, etc. Par ailleurs, à l'instar de la relation **isAbout**, le lien d'agrégation **containedEntities** entre une ECAR et un ensemble d'EARs implique une EAR Situation en tant que classe d'association. Ceci permet la contextualisation et la critique de la création des relations d'appartenance (voir I.3.3).

## I.4 Le modèle de tags

Le caractère générique des entités de base du modèle présentées en I.3 ainsi que l'expressivité des relations qu'elles entretiennent rendent possible la génération aisée de modèles métier « partiels ». Par la simple dérivation du concept de tag de celui d'EFAR, nous obtenons un puissant modèle de tagging :

- ⌘ Tout d'abord, un tag se définit intrinsèquement par un nom, et éventuellement une image.
- ⌘ En tant qu'EFAR, un tag peut être mis en relation à une EAR via la relation **isAbout**. Le lien réalisé est alors de type « qualification » (voir I.3.3).



**Figure 4.1** Diagramme de classes UML partiel du modèle de tags

- ☞ Chaque association d'un tag à une EAR est contextualisé par une Situation, ce qui permet de connaître pour un couple <Tag, EAR> donné le nombre et l'identité des utilisateurs ayant réalisé/confirmé l'association afférente ainsi que les dates de ces actions. Cette possibilité du modèle est déterminante pour bâtir des interfaces de statistique d'usage des ressources de classement au sein d'une communauté.
- ☞ En tant qu'EAR, un tag peut être mis en relation à n'importe quelle autre EAR sous quelque modalité que ce soit. Pour exemple, il est possible d'associer des tags à un utilisateur sous la modalité relationnelle « favoris », ce qui, une fois les fonctions adéquates implémentées, permettrait à tout utilisateur de sélectionner ses tags les plus utilisés sur sa page personnelle.
- ☞ Remarquons enfin, un peu spéculativement, qu'en tant qu'EAR, un tag peut être taggé, ce qui ouvre le champ à des interfaces d'administration des tags communautaires via des tags « administrateurs ».

## 1.5 Le cas des discussions

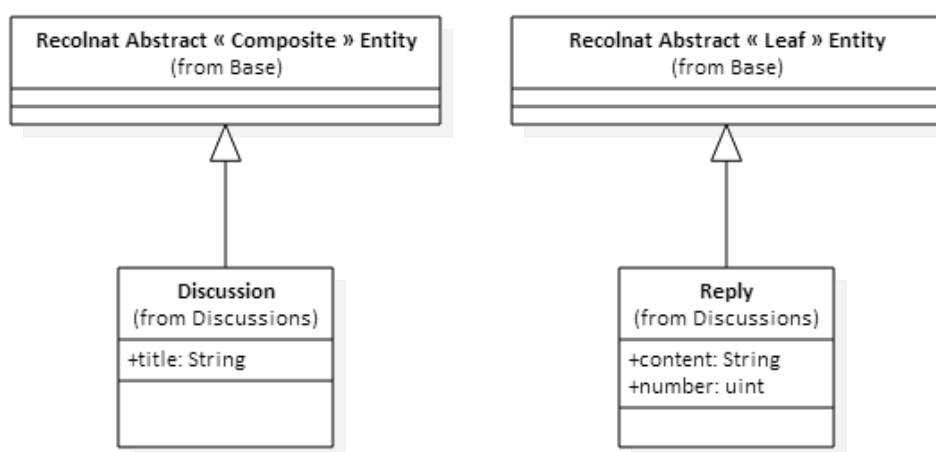
### 1.5.1 Discussions ciblées & discussions générales

Nous proposons d'opérer une distinction entre, d'une part, les *discussions générales* abordant des points relatifs à la vie de la communauté, portant sur le fonctionnement d'un module, et d'autre part, les discussions de portée scientifique à propos d'un objet documentaire ou d'un concept scientifique identifié. À la lumière de cette distinction, nous



préconisons que les discussions générales se tiennent dans des composants de type forums classiques (nul besoin d'un développement spécifique pour ces fonctions standards d'animation de communauté). À l'inverse, les discussions portant sur un objet botanique précis s'apparentent à des formes d'annotations dialogiques dont la valeur scientifique est indiscutable. Le modèle proposé *infra* répond à cette exigence.

## 1.5.2 Le modèle de discussions



**Figure 5.1** Diagramme de classes  
UML partiel du modèle de discussions

Dans les termes du modèle e-ReColNat+, une discussion est un cas particulier d'ECAR, ce qui lui permet d'agréger comme ses sous-partie un ensemble de réponses, qui sont des EFARs. Étudions les possibilités ouvertes par cette modélisation :

- ☞ En tant qu'EAR, les discussions comme les réponses possèdent un/e utilisateur/trice créateur/trice, une date de création, et gardent trace du module ayant accueilli leur création. Ce dernier point permet non seulement d'amorcer une discussion portant sur un objet botanique dans un module et de la poursuivre dans un autre, mais, sur chaque écran de quelque module que ce soit représentant cette discussion, chaque réponse peut indiquer le module dans lequel elle a été formulée.
- ☞ En tant qu'EAR et par la relation **isAbout**, une discussion peut porter (et dans les faits, porte effectivement) sur une autre EAR (par exemple, une entité botanique).
- ☞ Les discussions étant rattachées à des types concrets métier (planche, spécimen, étiquette, récolteur/trice, mission, etc.), le modèle rend possible des opérations de recherche de discussions par type d'entité, et donc des écrans dans les applications clientes de type « Les dernières discussions portant sur les planches ».

## **II**

# **LES APIS DE L'ÉCOSYSTÈME E-RECOLNAT**

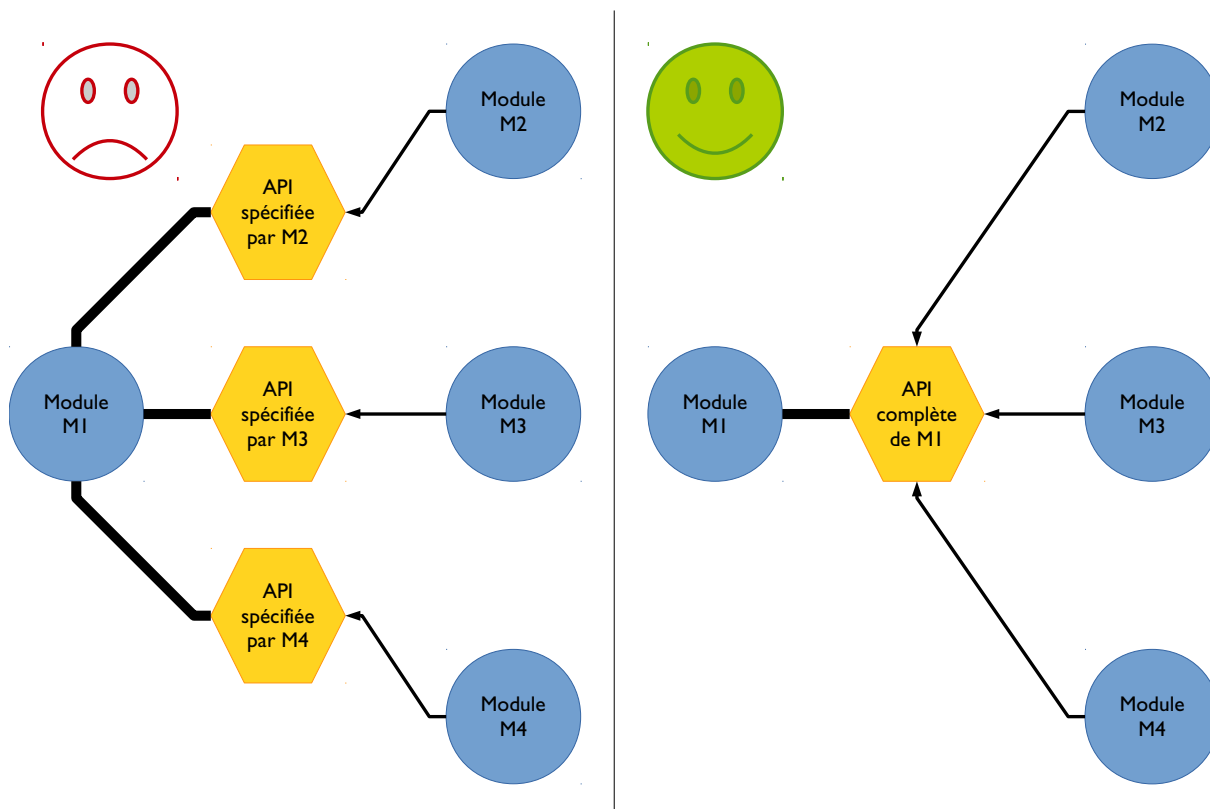
## 2.1 Principes généraux des API

### 2.1.1 Proposition de principe général

Les différents modules du portail peuvent chacun renvoyer à des acteurs, pratiques et enjeux spécifiques, dépendre de bases de données qui leurs sont propres ou encore être écrits dans des langages différents. Cette hétérogénéité fonctionnelle et technique ne doit cependant pas obombrer le fait que les modules sont amenés à manipuler les mêmes concepts et objets documentaires, sociaux et scientifiques (planche, spécimen, détermination, collection, unité de transcription, récolteur/teuse, utilisateur/trice, sous-partie d'intérêt d'une planche, etc.). Ainsi, s'il est prévu que les modules déploient des outils métier particuliers justifiant leurs existences autonomes, il est cependant vital qu'ils puissent consommer les données produites par leurs pairs. Au surplus, la couverture fonctionnelle d'un module pourra être naturellement amenée à évoluer durant la vie du projet (pour hypothèse : l'utilité d'une extension des Herbonautes pourrait être révélée d'ici un ou deux ans suite à une première série de tests, laquelle conduirait à la collecte d'un nouveau type de données dont l'exploitation serait également pertinente dans les autres modules). Pour ces raisons, nous préconisons que chaque module propose une *API exposant l'intégralité des données qu'il permet de récolter*. Ces APIs ne devraient pas être conçues en s'adaptant aux spécifications émanant des autres modules dénotant les besoins respectifs de ceux-ci à un temps donné, mais devraient plus simplement poursuivre le but d'exposer des données « brutes » structurées dans les termes métier des modules auxquels elles sont rattachées, sans présupposer des usages et attentes possibles en d'autres lieux du portail (voir la figure [I.1](#)).

### 2.1.2 Avantages futurs en terme de gestion de projet

Ce découplage des APIs des modules fournisseurs de données par rapport aux besoins des modules tiers consommateurs de données possède des vertus sur le plan de la gestion de projet. Ainsi, si un module fournisseur M<sub>1</sub> attend qu'un module consommateur M<sub>2</sub> spécifie les termes et structures sous lesquels il souhaite pouvoir accéder à un sous-ensemble des données de M<sub>1</sub>, l'API proposée par M<sub>1</sub> pourra se trouver inadéquate pour une future demande identique émanant d'un autre module M<sub>3</sub>. Dans ce cas, l'API définie par M<sub>1</sub> pour M<sub>2</sub> posera des problèmes assimilable à une dette technique pour toute extension à d'autres modules consommateurs. Si au contraire les modules fournisseurs exposent chacun une API complète « brute », la gestion des APIs ne constitue plus un ensemble de sous-projets bloquants transversaux supposant une co-validation de plusieurs partenaires, mais une simple extension fonctionnelle de chacun des modules, une simple modalité supplémentaire des fonctions relevant des aspects « *retrieval* » des opérations « CRUD ». Les modules développés par DICEN reposeront exclusivement sur



**Figure 1.1** Une API complète par module

ce principe. Les clients Web légers du collaboratoire et de la visite virtuelle seront alors les premiers « modules » consommateurs des APIs définies par le composant serveur exposant la base augmentée (voir [1.2.2](#)).

Remarquons que ces APIs complètes « brutes » devront s'accompagner d'une documentation publique à destination des développeurs de l'écosystème e-ReColNat précisant la sémantique et la syntaxe des ressources structurées qu'elles permettent de récupérer (voir *infra*).

Enfin, sur le plan scientifique, concevoir chaque module comme un fournisseur de données complet autorise à découvrir et inventer de nouveaux usages des données en cours de route. À ce jour, nous avons identifié des usages intéressants sur le plan communautaire ou scientifique à la quasi totalité des données produites par les Herbonautes au sein du collaboratoire.

## 2.2 Les APIs « tags » et « discussions »

Note : afin de spécifier nos APIs REST, nous utilisons le langage de description **RAML**. Nous préconisons son utilisation pour toute tâche similaire.

> [Dépôt Github des APIs e-ReColNat+ <](#)