



**HAL**  
open science

## CubicWeb as a Service: un service pour la publication sur le Web de données liées

Fabien Amarger, Nicolas Chauvat, Élodie Thieblin

► **To cite this version:**

Fabien Amarger, Nicolas Chauvat, Élodie Thieblin. CubicWeb as a Service: un service pour la publication sur le Web de données liées. 34es Journées francophones d'Ingénierie des Connaissances (IC 2023) @ Plate-Forme Intelligence Artificielle (PFIA 2023), Jul 2023, Strasbourg, France. hal-04252319

**HAL Id: hal-04252319**

**<https://hal.science/hal-04252319v1>**

Submitted on 20 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# CubicWeb as a Service : un service pour la publication sur le Web de données liées

Fabien Amarger<sup>1</sup>, Nicolas Chauvat<sup>1</sup>, Élodie Thieblin<sup>1</sup>  
<sup>1</sup> Logilab, 104 avenue Blanqui, Paris

prénom.nom@logilab.fr

## Résumé

*La publication de données RDF, suivant les 5 étoiles du Linked Open Data, devient de plus en plus courante. Ces données sont généralement publiées sous forme de fichiers dump sur des portails, ou rendues interrogeables en SPARQL.*

*Il est moins commun d'accéder aux données via une interface web et grâce au mécanisme de négociation de contenu du protocole HTTP, probablement car cela demande des compétences techniques supplémentaires pour déployer et maintenir ces formes d'accès.*

*Dans cet article, nous décrivons nos travaux sur CubicWeb et OWL2YAMS qui, ensemble, permettent de générer, à partir d'une ontologie OWL et de données RDF, une application Web offrant la négociation de contenu, une interface de consultation et une interface de gestion des données.*

*Nous y ajoutons l'automatisation du déploiement, pour qu'il devienne possible, via un formulaire Web, de partir de données de qualité en RDF et d'obtenir en quelques clics une application Web complète pour gérer, publier et visualiser ces données.*

*CubicWeb-as-a-Service a pour but de rendre toujours plus aisée la publication de données sur le Web en respectant les standards du W3C.*

## Mots-clés

*Web de données liées, publication de données, RDF, CubicWeb*

## 1 Introduction

De plus en plus de structures adoptent les principes des données ouvertes et liées (5 étoiles du Linked Open Data) en traduisant leurs données en RDF. Toutefois, la publication de ces données de bonne qualité se fait par un dump RDF (un fichier contenant la totalité des triplets RDF) ou par téléversement des triplets RDF dans un entrepôt RDF interrogeable en SPARQL. La négociation de contenu (API native du Web) est plus coûteuse à mettre en place, car elle requiert de configurer le serveur sur lequel seront servies les données.

Une fois les données en RDF publiées, elles sont généralement plus difficile à modifier directement en RDF que dans l'outil au sein duquel elles ont été produites et qui offre une interface spécifiquement adaptée.

CubicWeb et OWL2YAMS présentés dans des articles précédents[1, 2] permettent de prendre des données en RDF et de créer une application Web de gestion et publication des données, avec négociation de contenu selon le protocole HTTP.

La question de l'hébergement d'une l'application CubicWeb pose un problème pour les personnes et les institutions avec peu de compétences techniques.

CubicWeb-as-a-Service (CWaaS) est un outil en ligne qui automatise la création et l'hébergement d'une application CubicWeb à partir d'une ontologie OWL et des données RDF qui l'instancient.

Cet outil a pour objectif de rendre plus accessible la publication et la gestion de données sur le Web en respectant les standards du W3C.

## 2 CubicWeb et OWL2YAMS

CubicWeb[1] est un logiciel libre écrit en Python, conçu pour faciliter le développement et le déploiement d'applications qui reprennent les concepts essentiels du Web Sémantique. Avec CubicWeb, il est aisé de gérer et de rendre accessibles des données qui suivent un modèle préalablement défini.

CubicWeb utilise le formalisme YAMS (Yet Another Magic Schema<sup>1</sup>) pour représenter le modèle de données de façon explicite et stocker les données suivant ce modèle dans une base de données relationnelle, en profitant d'une gestion fine des permissions.

CubicWeb permet de configurer l'export RDF des données stockées et de les rendre accessibles via la négociation de contenu du protocole HTTP.

Des interfaces Web de gestion et de consultation des données génériques sont disponibles dans CubicWeb.

CubicWeb est utilisé dans des applications de grande envergure comme Data.BnF<sup>2</sup> ou FranceArchives<sup>3</sup>.

OWL2YAMS[2] permet de créer une application CubicWeb à partir d'une ontologie OWL et de la peupler avec des données RDF décrites par ladite ontologie.

À eux deux, ces outils facilitent la création d'une application Web de publication, consultation et gestion des don-

1. <https://forge.extranet.logilab.fr/open-source/yams>

2. [data.bnf.fr/](https://data.bnf.fr/)

3. <https://francearchives.gouv.fr/>

nées à partir de données en RDF, mais ils n'adressent pas la question de l'hébergement.

### 3 Déploiement automatique

Nous avons appelé notre approche **CubicWeb-as-a-Service** (CWaaS) car elle offre la possibilité de déployer une instance de CubicWeb automatiquement. L'idée principale est de proposer une mini-application Web, dans laquelle téléverser une ontologie OWL et des données en RDF respectant cette ontologie, pour qu'une application CubicWeb soit créée et déployée automatiquement. Ce service est donc à la portée de toutes et tous et permettrait de faciliter l'appropriation des technologies du Web Sémantique en les rendant extrêmement simple d'utilisation.

Sur la figure 1 est présentée l'interface d'accueil de CWaaS, sur laquelle ne figurent que trois champs dans le but de la rendre la plus simple possible.

Le premier champ est pour le nom de l'application (ce nom sera aussi utilisé pour générer l'URL à partir de laquelle l'application sera disponible).

Le deuxième champ permet de téléverser l'ontologie que nous souhaitons utiliser pour la création de l'application CubicWeb (dans cet exemple, nous utilisons l'ontologie SKOS exprimée dans la syntaxe N3).

Le troisième champ permet de téléverser un fichier RDF contenant les données qui seront automatiquement chargées dans cette application une fois qu'elle aura été déployée (dans cet exemple, un thésaurus exprimé en SKOS avec une syntaxe XML).

Une fois que le bouton "Create instance!" est actionné, l'ontologie est interprétée pour générer un cube CubicWeb en utilisant OWL2YAMS comme expliqué dans [2].

Le composant applicatif (cube) produit par OWL2YAMS est ensuite envoyé dans un nouvel entrepôt de code automatiquement créé dans notre forge GitLab<sup>4</sup>. Nous pourrions ensuite pousser plus loin le développement de ce cube pour ajouter des fonctionnalités ou modifier les permissions par exemple (c.f. [1] pour des détails) et en employant pour cela les outils et processus habituels de développement logiciel. Nous utilisons le système de déploiement continu intégré à GitLab pour construire l'image Docker<sup>5</sup> contenant l'application créée à partir de notre cube, puis pour déployer automatiquement cette image Docker sur un *cluster* Kubernetes<sup>6</sup>.

Pour suivre l'évolution de la création de l'image et du déploiement, les états d'avancement des jobs de déploiement continu sont affichés sur l'interface de CWaaS, comme nous pouvons le voir sur la figure 2.

Sur cette même figure, nous pouvons observer plusieurs liens. Tout d'abord le lien vers le dépôt de code de l'application générée. Ensuite, un lien pour vérifier l'état d'avancement des tâches du déploiement continu directement sur l'entrepôt de code. Et enfin le lien vers l'instance de l'application qui sera valide lorsqu'elle sera déployée.

4. <https://docs.gitlab.com/>

5. <https://www.docker.com/>

6. <https://kubernetes.io/fr/>

Lorsque l'étape de déploiement est terminée, nous pouvons alors accéder à l'application elle-même, comme nous pouvons le voir sur la figure 3.

Nous observons sur cette figure que l'application déployée a bien intégré les classes de l'ontologie SKOS en chargeant les données (il y a ici 506 instances de `skos:Concept` dans l'application).

L'application est à ce stade disponible, avec les données chargées. Il est possible d'utiliser l'interface d'administration pour modifier les données ou la configuration du serveur. Si une modification est apportée au code dans le dépôt de code, le processus d'intégration continue sera relancé et mettra à jour automatiquement l'application déployée. Il devient donc assez simple, même pour des développeurs ou développeuses, de faire évoluer cette application pour ajouter de nouvelles fonctionnalités, notamment en ajoutant des interfaces personnalisées pour afficher les données de la manière qui convient le mieux.

Dans le cadre du projet CubicWeb, nous sommes en train de travailler sur une nouvelle interface d'administration générique qui s'appuie sur la bibliothèque React-Admin<sup>7</sup>, illustrée par la figure 4.

### 4 Conclusion et perspectives

Nous présentons ici notre approche *CubicWeb-as-a-Service* permettant de créer une instance de CubicWeb et de la déployer très facilement en utilisant une application Web monopage sur laquelle figure un formulaire composé de trois champs. Toute la pile technique utilisée est disponible sous licence libre aux adresses suivantes :

- CubicWeb  
<https://forge.extranet.logilab.fr/cubicweb/cubicweb>
- OWL2YAMS  
<https://forge.extranet.logilab.fr/cubicweb/owl2yams>
- CubicWeb-as-a-Service  
<https://forge.extranet.logilab.fr/cubicweb/cubicweb-as-a-service>

Les autres outils employés sont Docker, Kubernetes, GitLab et GitLab-runner qui sont eux aussi sous licence libre. Plusieurs évolutions de CWaaS sont prévues pour améliorer son utilisation.

Tout d'abord, nous souhaitons rendre plus robuste OWL2YAMS. Effectivement, comme détaillé dans [2], il y a un écart d'expressivité entre la définition du modèle de données dans CubicWeb (YAMS) et OWL-lite. Nous souhaitons améliorer YAMS et OWL2YAMS pour atteindre un niveau d'expressivité similaire à OWL-lite.

L'interrogation des données d'une instance de CubicWeb se fait avec un langage d'interrogation nommé RQL[1]. Bien que ce langage soit assez proche du SPARQL, il existe quelques différences d'expressivité. Nous avons commencé à étudier la possibilité d'interroger une instance CubicWeb en SPARQL, notamment avec l'outil OnTop[3]. Nous avons commencé à développer un cube qui permet

7. <https://marmelab.com/react-admin/>



FIGURE 1 – Interface d'accueil de CWaaS

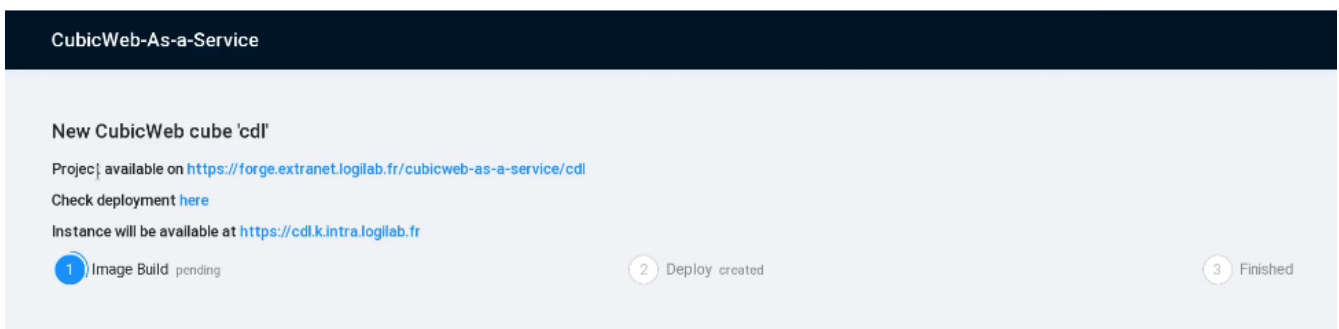


FIGURE 2 – Suivi de l'état d'avancement du déploiement de l'application

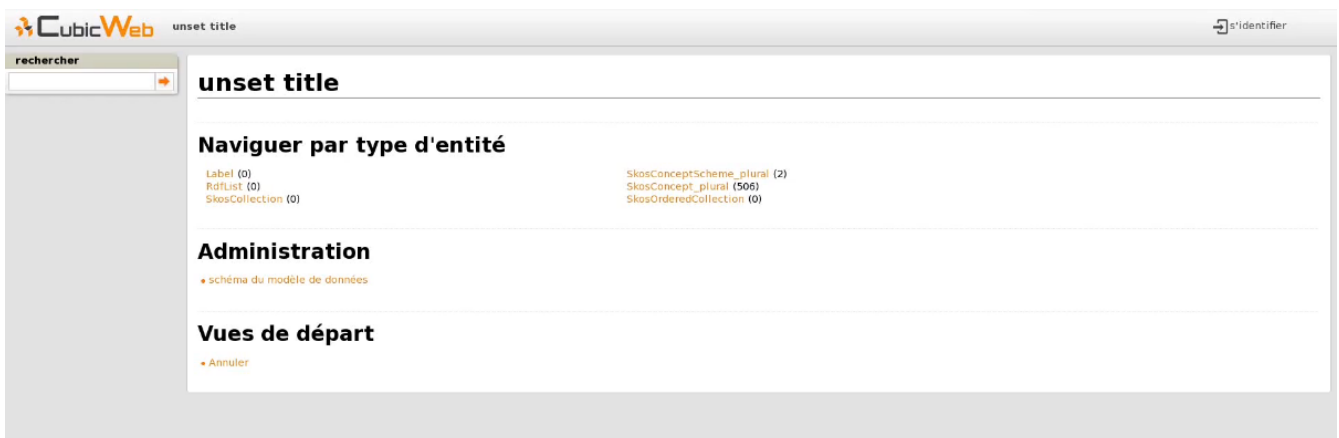


FIGURE 3 – Interface par défaut de CubicWeb

The screenshot shows the CubicWeb administration interface. On the left is a sidebar with navigation links like 'Welcome', 'Auth Token', 'Blog', 'BlogEntry', etc. A search bar and a 'Filters' panel are at the top left. The main area displays a table of blog entries with columns for 'Eid', 'Heading', 'Content', 'Title', 'Content format', 'Creation date', and 'Modification date'. Two entries are visible, one with Eid 25 653 872 and another with Eid 25 283 370.

Eid	Heading	Content	Title	Content format	Creation date	Modification date
25 653 872	Les AFYRo reprennent et le prochain aura lieu dans nos locaux parisiens entre Denfert Rochereau et la Place d'Italie !	Les AFYRo reprennent et le prochain aura lieu dans nos locaux parisiens entre Denfert Rochereau et la Place d'Italie ! Un AFYRo est un événement organisé par l'AFPY (https://www.afpy.org/) - Association Francophone Python - pour regrouper des personnes souhaitant discuter du langage de programmation Python dans un cadre convivial. Après une ou deux présentations (vous pouvez proposer la vôtre) (https://discuss.afpy.org/t/pour-les-parisiens-afpyro-chez-logilab-le-16-mars/1384), nous échangerons autour de quelques pizzas. Le prochain AFYRo sera donc à Logilab, au 104 Boulevard Auguste Blanqui 75013 Paris, le 16 mars 2023 de 19h à 21h et nous offrirons les pizzas. N'hésitez pas à [vous inscrire](https://discuss.afpy.org/t/pour-les-parisiens-afpyro-chez-logilab-le-16-mars/1384) et à passer nous voir !	AFYRo du 16 mars 2023	text/markdown	27/02/2023 18 09 05	27/02/2023 18 09 52
25 283 370	Comme annoncé en novembre dernier, nous allons essayer de diversifier les événements SemWeb Pro. Nous commençons par organiser une journée d'ateliers SemWeb pro à Toulouse.	Comme annoncé en novembre dernier, nous allons essayer de diversifier les événements SemWeb Pro. Nous commençons par organiser une journée d'ateliers SemWeb pro à Toulouse. [Semweb pro à Toulouse](https://2023-tls.semweb.pro/) au sujet du web sémantique et orientée vers les données industrielles. L'idée est de réunir les acteurs industriels locaux ou non pour échanger ensemble sur l'usage des standards du web sémantique, qu'il s'agisse de décrire des installations, des processus ou des équipements, exploiter des réseaux de capteurs, etc. Cette journée aura lieu le 14 Mars à Toulouse, dans l'espace de coworking et de réunion [O'Local](https://www.oloocal.fr/), dans une belle bâtisse typiquement toulousaine, avec comme objectif de favoriser au maximum les interactions. La matinée sera dédiée à des présentations sur le thème du Web Sémantique pour l'industrie. L'après-midi sera l'occasion de mener des ateliers en petits groupes pour prendre le temps d'échanger. Une restitution rapide de s'ateliers clôturera la journée. Tous les détails relatifs à l'inscription sont sur le site [semweb pro](https://2023-tls.semweb.pro). Attention, le nombre de places est limité.	Un nouvel événement SemWeb pro	text/markdown	07/02/2023 16 17 26	07/02/2023 16 21 35

FIGURE 4 – La nouvelle interface d’administration React-Admin

de créer automatiquement la configuration OnTop à partir d’un schéma YAMS (<https://forge.extranet.logilab.fr/cubicweb/cubes/ontop>). Il nous faudrait continuer ces expérimentations pour valider l’hypothèse qu’il est possible d’interroger une instance de CubicWeb en SPARQL.

Enfin, nous avons l’intuition que notre approche concernant CubicWeb, qui permet notamment de faire une séparation claire entre le serveur fournissant les données et les interfaces permettant de les afficher, est très proche de ce qui est proposé dans SOLID<sup>8</sup>. Nous souhaitons approfondir ce lien pour déterminer comment une instance CubicWeb peut interagir avec un pod SOLID, voir comment une instance de CubicWeb pourrait devenir un pod SOLID.

## Références

- [1] Fabien Amarger, Simon Chabot, Nicolas Chauvat, and Elodie Thiéblin. Cubicweb : vers un outil pour des applications clé en main dans le web sémantique. In *31es Journées francophones d’Ingénierie des Connaissances*, 2020.
- [2] Fabien Amarger, Nicolas Chauvat, and Elodie Thiéblin. Owl2yams : créer une application cubicweb à partir d’une ontologie owl. In *Journées francophones d’Ingénierie des Connaissances 2022*, 2022.
- [3] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop : Answering sparql queries over relational databases. *Semantic Web*, 8(3) :471–487, 2017.

8. <https://solidproject.org/>