



HAL
open science

MyBookings – Application de réservation de ressources en ligne

Lylian Blaud, Pierre Raynaud

► **To cite this version:**

Lylian Blaud, Pierre Raynaud. MyBookings – Application de réservation de ressources en ligne. JRES (Journées réseaux de l'enseignement et de la recherche) 2024, Renater, Dec 2024, Rennes, France. hal-04893981

HAL Id: hal-04893981

<https://hal.science/hal-04893981v1>

Submitted on 17 Jan 2025

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.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

MyBookings – Application de réservation de ressources en ligne

Lylian Blaud

DSI, UCA
7 avenue Blaise Pascal
63170 Aubière

Pierre Raynaud

DSI, UCA
7 avenue Blaise Pascal
63170 Aubière

Résumé

Née lors des discussions autour du futur Learning Center de l'Université Clermont Auvergne, la plateforme MyBookings doit répondre à une problématique essentielle : permettre aux étudiants de réserver eux-mêmes les espaces de travail disponibles.

Suite à un état des lieux en interne et une étude des solutions existantes, il a été décidé de développer une application en interne répondant à ce besoin et couvrant également le périmètre fonctionnel du prêt de matériel.

Cette application offre une solution flexible pour réserver diverses ressources telles que des salles de réunion, des véhicules, de l'équipement informatique, audiovisuel ou autres équipements spécifiques en lien avec différents services connexes (gestion de parc, contrôle d'accès ...).

Elle a été conçue dès le début comme une application pouvant s'adapter à de très nombreux cas d'usage et des environnements techniques et organisationnels divers grâce à une grande possibilité de configuration et un socle technique simple (PHP Symfony, VueJS, MariaDB, Docker).

MyBookings a pour vocation d'être un service pouvant être proposé à d'autres établissements souhaitant répondre à cette problématique.

Plus d'informations sur le projet : <https://mybookings.uca.fr/about>

Mots-clefs

Catalogue, ressources, API, CI/CD, back-end, front-end

1 Contexte

1.1 L'établissement

L'Université Clermont Auvergne (UCA) est née en 2017 de la fusion de deux établissements : l'Université Blaise Pascal (Clermont-Ferrand II) et l'Université d'Auvergne (Clermont-Ferrand I).

L'UCA regroupe plusieurs facultés, instituts et écoles, couvrant une large gamme de disciplines pour ses 36000 étudiants. Elle est organisée en 20 composantes : facultés, écoles et instituts dans des domaines comme les sciences, la médecine, le droit, les sciences humaines et sociales, les lettres, l'ingénierie, l'économie et la gestion. L'UCA est également très active dans la recherche, avec de 47 structures de recherche, souvent en partenariat avec d'autres institutions publiques et

privées. Ses domaines de recherche incluent les sciences de la vie, l'environnement, la physique, les sciences humaines, et les technologies.

1.2 La gestion des ressources

Depuis toujours, l'université met à disposition de ses étudiants, enseignants et personnels une grande variété de ressources matérielles et de services. Ces équipements incluent du matériel informatique, des salles de réunion, des box de travail pour les étudiants, des véhicules de service, ainsi que du matériel scientifique spécialisé. Cette offre diversifiée et variée a pour objectif de faciliter le travail quotidien et de répondre aux besoins spécifiques de chaque usager, qu'il s'agisse de projets académiques, de recherches ou d'activités administratives.

L'université utilise un système de gestion des ressources (GRR) qui, bien que central, est aujourd'hui considéré comme vieillissant et mal adapté aux besoins actuels. Ce système n'est plus suffisamment flexible ni ergonomique pour répondre aux demandes croissantes et diversifiées des utilisateurs. De plus, il souffre d'une intégration insuffisante au système d'information (SI) global de l'université, ce qui le rend moins efficace et plus difficile à utiliser pour certains services. Les limites du GRR ralentissent ainsi les processus de gestion et peuvent générer des frustrations, tant pour les gestionnaires que pour les utilisateurs finaux.

En parallèle, chaque service de l'université gère de manière autonome ses propres ressources et déploie des procédures de réservation et de mise à disposition qui peuvent varier considérablement. Selon le type de service ou de matériel, les utilisateurs doivent passer par différents canaux pour effectuer des demandes : cela peut se faire via une présence physique, un appel téléphonique, un courriel, ou encore par le biais de calendriers partagés. Dans certains cas, un système de tickets est également utilisé pour gérer les demandes et suivre leur traitement. Cette hétérogénéité des pratiques rend parfois la réservation de ressources complexes, surtout pour les nouveaux usagers ou ceux qui doivent interagir avec plusieurs services.

Face à ces défis, une modernisation des outils de gestion des ressources semble nécessaire pour améliorer l'efficacité et l'accessibilité des services. Une solution plus intégrée et adaptée aux besoins actuels pourrait non seulement faciliter la gestion des réservations, mais aussi offrir une expérience utilisateur harmonisée et fluide.

1.3 Le projet Learning Centre

Dans le cadre de son développement stratégique et après avoir été lauréate du projet iSite CAP 20-25 en 2017, l'Université Clermont Auvergne a planifié la création d'un Learning Centre innovant pour la rentrée 2024. Ce nouvel espace, destiné à mêler bibliothèque universitaire (BU), salles de travail collaboratif et espaces dédiés à l'innovation pédagogique, a pour objectif de répondre aux nouvelles exigences des méthodes d'apprentissage modernes, en offrant un environnement flexible adapté aux besoins des étudiants.

Lors de la conception de ce projet, un des points clés du cahier des charges concernait la mise en place d'une application permettant aux étudiants de réserver en toute autonomie des box de travail sur des horaires étendus. Les exigences fonctionnelles définies étaient claires : l'application devait

permettre aux utilisateurs de consulter les plannings disponibles, de formuler une demande de réservation, et de recevoir l'attribution d'un box de manière automatique.

En parallèle, il avait été spécifié que l'application devait interagir avec le système de contrôle d'accès du Learning Centre. Ainsi, lors de sa réservation, l'étudiant doit pouvoir accéder à son box réservé en utilisant simplement sa carte d'étudiant pour ouvrir la porte, garantissant à la fois un accès sécurisé et un usage fluide des espaces de travail.

2 Phase préalable du projet

Dans la phase préalable à la création de l'application, plusieurs grands principes ont été définis pour orienter le projet. L'une des exigences majeures était de concevoir un outil qui réponde non seulement à la problématique spécifique du Learning Centre, mais aussi de proposer une solution globale capable de couvrir l'ensemble des besoins en matière de réservation et gestion des ressources à l'Université. L'objectif était d'éviter la fragmentation des outils et de centraliser la gestion des différents espaces et équipements de l'UCA.

Une étude comparative de plusieurs outils disponibles sur le marché a été menée. Bien que chacun d'entre eux présente des avantages spécifiques, aucun ne s'est avéré exhaustif ni suffisamment malléable pour s'adapter pleinement aux exigences de l'université. Ces solutions étaient souvent rigides, offrant peu de flexibilité dans la personnalisation des fonctionnalités ou l'intégration aux systèmes existants.

Face à ce constat, la décision a été prise de valider un développement local, en suivant des principes clairs. L'outil devait être « API centric, » c'est-à-dire conçu autour d'une structure d'API pour faciliter les intégrations avec d'autres systèmes, tout en restant très ouvert et hyper paramétrable. Cette flexibilité devrait permettre d'adapter la solution aux différents besoins des services, tout en garantissant une évolutivité à long terme.

La stratégie projet adoptée consistait à développer d'abord un outil minimal viable (MVP), fonctionnel mais concentré sur les besoins essentiels. Ce prototype a ensuite été présenté aux différents services de l'université pour qu'ils puissent ajouter leurs contraintes spécifiques. Grâce à la nature hautement paramétrable de l'outil, ces nouvelles exigences ont pu être facilement intégrées dans les phases suivantes du projet, assurant ainsi une réponse complète aux besoins de l'UCA.

3 Architecture technique

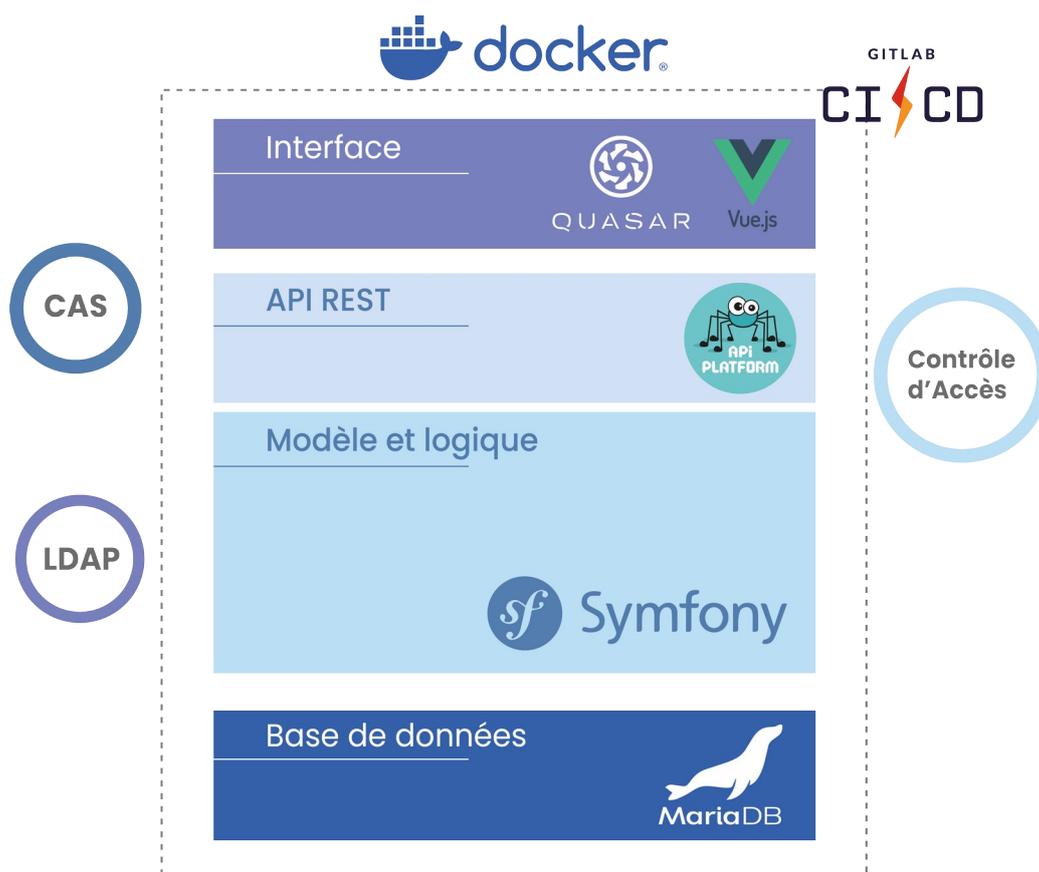


Figure 1 – Architecture technique

L'architecture technique de l'application repose sur une approche moderne utilisant la conteneurisation Docker. Elle permet une gestion optimale des dépendances et une portabilité facilitée. Chaque composant de l'application, qu'il s'agisse de l'interface utilisateur, de l'API ou de la base de données, est encapsulé dans un conteneur distinct, garantissant ainsi une isolation et une évolutivité accrues.

L'interface utilisateur de l'application est développée en Vue.js, un framework JavaScript reconnu pour sa capacité à créer des interfaces dynamiques et réactives. Cette interface est renforcée par l'intégration de Quasar, un framework complémentaire qui offre une vaste collection de composants et outils permettant de concevoir des applications web à la fois esthétiques et performantes, tout en étant totalement *responsive*.

Le cœur de l'application repose sur une architecture API REST, qui assure la communication entre les différentes couches du système. Cette API centralise l'ensemble des échanges de données entre le *front-end* et le *back-end*, tout en garantissant une flexibilité qui permet à l'application de s'adapter à différents types de clients.

Du côté logique et modèle métier, l'application utilise le framework Symfony, reconnu pour sa robustesse et sa modularité. Symfony gère l'ensemble des processus métiers et les interactions avec

les services *back-end*, tout en offrant une architecture évolutive qui facilite la maintenance du code sur le long terme.

Pour la gestion des données, l'application s'appuie sur MariaDB, une base de données relationnelle open-source dérivée de MySQL. MariaDB garantit une persistance des données tout en assurant de bonnes performances, même avec des volumes de données importants.

L'application ne fonctionne pas de manière isolée, elle s'intègre avec plusieurs systèmes externes. En matière d'authentification, elle se connecte à des services tels que CAS (Central Authentication Service) et LDAP (Lightweight Directory Access Protocol), permettant ainsi une gestion centralisée et sécurisée des accès utilisateurs. Cette approche simplifie le processus d'authentification, en particulier dans les environnements où plusieurs systèmes doivent cohabiter.

En parallèle, l'application dialogue avec des *webservices*. Appelés actionneurs, ils permettent d'enrichir les capacités de l'application afin d'accéder à des services métiers sans avoir à tout développer en interne. Dans notre cas, CASTEL, le service d'accréditation de l'université, nous permet d'accréditer les badges des utilisateurs lors de certaines réservations.

Enfin, l'application est conçue pour être continuellement améliorée et déployée de manière automatisée grâce à une intégration continue et déploiement continu (CI/CD). Cela signifie que chaque modification du code passe par une série de tests automatisés avant d'être intégrée dans l'environnement de production.

4 Concepts de l'application

MyBookings doit offrir des fonctionnalités intuitives tout en maintenant un haut niveau de flexibilité. L'application que nous vous présentons intègre plusieurs concepts clés pour répondre à ces exigences.

4.1 Service

Le service joue un rôle central dans l'organisation de l'application. Il permet de segmenter et de simplifier la gestion des ressources et des catalogues, en organisant ces éléments autour de différents espaces d'administration spécifiques. Chaque service agit comme une entité indépendante, où l'on peut créer et gérer des ressources, définir des *workflows* et organiser des catalogues, offrant ainsi une approche modulaire et claire dans l'administration globale de l'application.

Au sein de chaque service, il existe deux niveaux de gestion avec des rôles bien distincts : administrateurs et modérateurs. L'administrateur détient l'ensemble des droits de gestion au sein du service. Il peut ainsi créer et modifier les catalogues, ajouter des ressources, définir les *workflows* et ajuster les règles associées à la gestion des réservations. En somme, l'administrateur a un contrôle total sur l'organisation et le fonctionnement du service.

Le modérateur, quant à lui, dispose d'un rôle plus limité. Son intervention se concentre exclusivement sur la modération des réservations. Cela signifie qu'il peut approuver, refuser ou ajuster les demandes de réservation, mais n'a pas accès à la modification des ressources, des

catalogues ou des *workflows*. Ce rôle est conçu pour assurer un suivi des réservations sans donner accès aux fonctionnalités de configuration plus sensibles.

Le service constitue donc le point d'entrée pour toute la gestion opérationnelle : c'est à partir de cette entité que l'on crée les catalogues, les ressources et que l'on met en place les *workflows*. En segmentant les droits et responsabilités, il devient plus facile de gérer l'ensemble des activités tout en garantissant un contrôle adapté selon les rôles, assurant ainsi une gestion sécurisée et organisée des différents éléments.

4.2 Catalogues et ressources

Les catalogues sont des éléments clés dans la gestion de l'application. Un catalogue regroupe un ensemble de ressources au sein d'un service, facilitant ainsi leur organisation et leur gestion. Les catalogues sont conçus pour structurer les ressources de manière claire et logique, afin de permettre aux utilisateurs de les retrouver facilement. Pour ce faire, les catalogues sont organisés en catégories et sous-catégories, créant une hiérarchie qui aide à la navigation et à la recherche. Cette classification permet aux utilisateurs de filtrer rapidement les catalogues en fonction de leurs besoins spécifiques, améliorant ainsi leur expérience.



Figure 2 – Catalogue en vue 'Collection'

En plus de cette organisation, les catalogues peuvent être configurés avec deux types de vues distinctes : la vue Collection et la vue Lot. La vue Collection donne à l'utilisateur la possibilité de choisir une ressource spécifique au moment de la réservation. Cela est utile lorsque les ressources dans le catalogue sont variées et que l'utilisateur doit sélectionner une ressource en fonction de caractéristiques précises (par exemple, choisir une salle particulière ou un équipement spécifique).

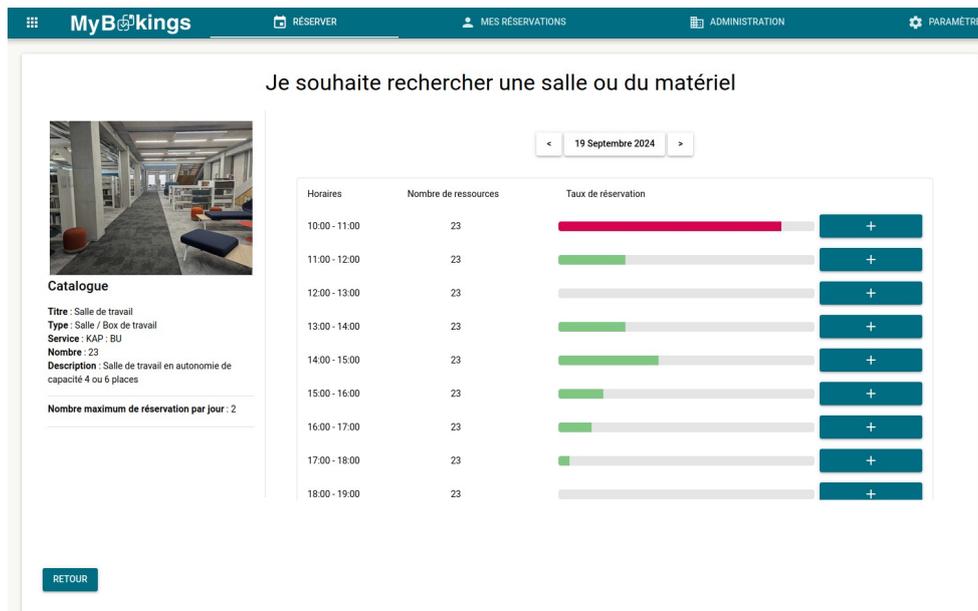


Figure 3 – Catalogue en vue ‘Lot’

En revanche, la vue Lot suppose que toutes les ressources du catalogue sont équivalentes. L'utilisateur ne choisit pas directement une ressource particulière, car elles sont considérées comme étant interchangeables. Dans ce cas, la ressource affectée sera déterminée en fonction des critères de la réservation (par exemple, la première ressource disponible ou celle correspondant à une certaine configuration). Cette approche est particulièrement utile lorsque l'on gère des ressources homogènes où le choix individuel n'est pas nécessaire, comme des lots de matériel identique.

4.3 Workflows : processus de validation

Les *workflows* permettent de contrôler le déroulement complet du processus de réservation, de la confirmation à la restitution, en offrant la possibilité de choisir entre des validations automatiques ou manuelles pour chaque étape. Cette personnalisation permet de s'adapter aux exigences spécifiques de chaque service ou organisation, tout en assurant une gestion rigoureuse et efficace des ressources.

Le *workflow* intervient ainsi sur trois moments critiques : la confirmation de la réservation, l'accès à la ressource, et le rendu de la ressource.

La confirmation de la réservation concerne l'approbation initiale de la demande de réservation. Le *workflow* peut être configuré pour que cette validation soit automatique (sans intervention humaine) ou manuelle, nécessitant alors la modération d'un responsable. Cela permet de filtrer les réservations en amont et d'assurer que seules les demandes légitimes ou conformes aux règles passent à l'étape suivante.

Une fois la réservation confirmée, le *workflow* peut également définir comment l'utilisateur accède à la ressource. Ici encore, il est possible d'automatiser cette étape, rendant immédiatement accessible la ressource après confirmation, ou de la soumettre à une nouvelle validation manuelle. Cela est particulièrement utile pour garantir que des conditions spécifiques soient respectées avant l'accès.

Enfin, la dernière étape concerne le retour de la ressource. Le *workflow* peut automatiser cette étape en considérant que la ressource a été correctement rendue une fois la période de réservation terminée. Cependant, il peut également être configuré pour exiger une validation manuelle, où un modérateur doit vérifier l'état de la ressource avant de finaliser le processus.

En plus de ces trois étapes, les *workflows* permettent de personnaliser davantage le processus en intégrant un système de notifications par e-mail. Pour chaque étape, un *mailing* spécifique peut être paramétré afin qu'un modérateur particulier soit alerté et affecté à la modération de la réservation. Cela garantit que les bonnes personnes sont informées et impliquées au bon moment, facilitant ainsi la gestion des réservations tout en maintenant un niveau de contrôle adéquat.

4.4 Mise à disposition

La mise à disposition d'un catalogue est une fonctionnalité essentielle permettant de contrôler sa visibilité et son accessibilité pour les utilisateurs. Elle repose sur plusieurs paramètres qui définissent non seulement quand et comment un catalogue devient visible, mais aussi les conditions d'accès aux ressources qu'il contient.

Tout d'abord, la mise à disposition permet de définir une date d'ouverture et une date de clôture du catalogue. Cela détermine la période durant laquelle le catalogue sera visible et disponible pour les réservations. À l'intérieur de cette période globale, il est possible d'affiner davantage en spécifiant des périodes de mise à disposition plus précises, telles que certains jours de la semaine (par exemple, uniquement le lundi, mardi, et mercredi). Ces périodes peuvent également être définies avec une heure de début et une heure de fin, permettant ainsi de contrôler les créneaux horaires pendant lesquels les utilisateurs peuvent effectuer leurs réservations.

Un autre paramètre clé est la durée maximale de réservation. Ce paramètre fixe la limite de temps pendant laquelle une ressource peut être réservée en une seule fois. En fonction de cette durée, il est possible de configurer des règles supplémentaires, comme le nombre maximum de réservations qu'un utilisateur peut effectuer sur une période donnée (par exemple, un maximum de X réservations par semaine) ou encore le nombre de réservations par jour.

La mise à disposition permet également de gérer les groupes de visibilité du catalogue. Il est ainsi possible de restreindre l'accès aux catalogues selon des critères spécifiques, par exemple en rendant certains catalogues visibles uniquement pour les étudiants, les personnels ou encore pour des formations particulières. Cette fonctionnalité garantit que les bonnes ressources sont accessibles aux bons utilisateurs, selon les besoins de chacun.

Enfin, un processus de validation (ou *workflow*) peut être défini pour chaque catalogue, déterminant si une réservation nécessite une validation manuelle avant d'être confirmée. En fonction du *workflow* mis en place, certaines réservations peuvent être automatiquement validées, tandis que d'autres devront passer par un processus d'approbation spécifique, garantissant ainsi un contrôle plus poussé sur l'utilisation des ressources.

4.5 Règles personnalisées

Les règles personnalisées ajoutent un niveau d'intelligence au processus de mise à disposition des ressources en permettant aux administrateurs de définir des conditions spécifiques pour la gestion et la réservation des catalogues. Grâce à ces règles, l'application peut automatiser des décisions complexes concernant l'attribution des ressources, en fonction de critères prédéfinis, garantissant une meilleure gestion des disponibilités et des conditions d'utilisation.

Ces règles permettent à une administration d'appliquer des restrictions ou des conditions particulières sur un catalogue de ressources. Par exemple, il peut s'agir de limiter l'accès à certaines ressources en fonction du nombre de participants ou d'autres critères pertinents. Un cas concret pourrait être la gestion du box de travail de différentes tailles. Dans ce scénario, une règle personnalisée peut être mise en place stipulant que ces box ne peuvent être réservés que si le nombre de participants est supérieur à 50 % de la capacité totale du box. Cela garantit une utilisation optimale des espaces, évitant qu'une grande salle soit réservée par un groupe trop petit, et assurant ainsi une répartition plus efficace des ressources.

4.6 Actionneurs

Les actionneurs sont des éléments automatisés qui s'activent une fois qu'une réservation a été confirmée dans l'application. Leur rôle est de déclencher des actions spécifiques vers des systèmes externes ou des processus complémentaires, facilitant ainsi l'intégration et la coordination avec d'autres services ou outils utilisés par l'organisation.

Un actionneur peut être configuré pour réaliser diverses tâches automatisées après la confirmation d'une réservation. Par exemple, il peut effectuer un appel au système de contrôle d'accès afin de permettre à l'utilisateur d'accéder physiquement à une salle ou à un équipement réservé. De la même manière, un actionneur peut envoyer une notification automatique à l'utilisateur ou à un groupe de personnes concernées, leur confirmant la réservation ou leur fournissant des instructions supplémentaires.

5 Bilan et perspectives

Depuis son déploiement, MyBookings rencontre un franc succès. Actuellement, neuf services, dont des laboratoires, des services informatiques et la bibliothèque universitaire, utilisent cette solution au quotidien. Un des succès majeurs concerne la réservation des box de travail avec accès automatique : plus de 3 500 réservations ont déjà été effectuées sur différents sites, permettant aux étudiants de réserver et d'accéder aux espaces de manière autonome et fluide.

Face à ce succès, de nombreuses demandes émanent d'autres services et composantes pour rejoindre le système, soulignant l'intérêt croissant pour cet outil centralisé et adaptable.

Roadmap : vers de nouvelles fonctionnalités

Pour répondre à cette demande et améliorer encore l'expérience utilisateur, plusieurs axes d'évolution ont été identifiés dans la feuille de route. Parmi les priorités :

- La possibilité d'étendre la réservation à des périodes plus longues, avec des options de réservation à la journée ou à la semaine.
- L'ajout d'une fonctionnalité pour l'authentification des utilisateurs extérieurs via un lien direct, facilitant l'accès aux ressources pour les partenaires.
- Interaction avec Sterennes, un outil de business intelligence (BI) mis en place par l'Université de Rennes permettant de produire des statistiques détaillées sur l'utilisation des ressources, afin de mieux piloter la gestion et l'optimisation des espaces.
- L'exploration d'une solution basée sur l'intelligence artificielle, qui pourrait servir d'assistant intelligent pour simplifier encore davantage les processus de réservation.

Mutualisation et déploiement élargi

Cet outil a été conçu dès le départ pour permettre la **mutualisation** de la solution à d'autres établissements. Il est ainsi envisagé de proposer l'outil en mode SaaS (Software as a Service), hébergé sur le data center de l'UCA ; les modalités d'accès, les coûts et les engagements nécessaires pour ce type d'hébergement sont actuellement à l'étude pour disposer d'une solution performante, maintenable et économique pour chacun.

Pour accompagner ces perspectives, plusieurs aménagements techniques sont à prévoir :

- La mise en place d'une instance de démonstration, avec authentification Shibboleth, pour permettre aux nouveaux utilisateurs de tester la solution.
- Le développement d'un proxy logiciel pour faciliter l'accès aux ressources internes des établissements partenaires, en faisant transiter les requêtes de l'application centrale vers les systèmes de contrôle d'accès locaux notamment.
- Une personnalisation accrue de l'outil, avec des options pour adapter la charte graphique et les libellés aux besoins spécifiques de chaque établissement.

Ces évolutions permettront non seulement de répondre aux demandes croissantes des services internes, mais aussi d'étendre l'utilisation de l'outil à une échelle nationale, tout en assurant flexibilité, sécurité et performance.