Mettre les technologies cloud au service de la production statistique

Romain Avouac
Insee
romain.avouac@insee.fr

Thomas Faria
Insee
thomas.faria@insee.fr

Frédéric Comte
Insee
frederic.comte@insee.fr

2025-01-27

Résumé French

1 Introduction

L'exploitation de sources de données non traditionnelles afin d'améliorer le processus de production statistique est une orientation majeure du Système Statistique Européen (SSE). Cette évolution vers un modèle de *Trusted Smart Statistics* [1] s'accompagne d'innovations dans les processus statistiques, permettant de tirer parti du potentiel de ces sources — plus grande disponibilité, résolution spatio-temporelle accrue, etc. — tout en faisant face à leur complexité et à leurs limites. Parmi ces innovations figurent les méthodes d'apprentissage automatique et leurs applications prometteuses dans les domaines du codage et de la classification, des redressements et de l'imputation [2]. Les multiples défis auxquels font face les instituts statistiques dans ce contexte d'évolution sont abordés dans le *Mémorandum de Bucarest sur les statistiques publiques dans une société numérisée*, qui anticipe que « la variété des nouvelles sources de données, paradigmes computationnels et outils nécessitera des adaptations de l'architecture métier statistique, des processus, des modèles de production, des infrastructures informatiques, des cadres méthodologiques et de qualité, ainsi que des structures de gouvernance correspondantes », et invite en conséquence le SSE à évaluer les adaptations requises et à les prioriser [3]. Cette évolution est également largement visible dans le cadre du service statistique public (SSP), dont elle constitue l'une des lignes directrices de la stratégie à horizon 2025 [4].

Dans l'optique de ces transformations, de nombreux travaux ont été menés dans le cadre de projets successifs à l'échelle européenne pour opérationnaliser l'utilisation de sources de données non-traditionnelles dans la production de statistiques officielles. Dans le cadre du projet ESSnet Big Data II (2018-2020), les instituts statistiques nationaux (INS) ont travaillé sur une large gamme de thématiques (offres d'emploi en ligne, transactions financières, traces GPS, etc.) afin de constituer les briques nécessaires pour intégrer ces sources dans les processus de production et identifier leurs limites [5]. En France, les travaux sur l'exploitation des données mobiles [6] ou des données

de caisse [7] ont permis d'illustrer le potentiel de ces sources pour construire de nouveaux indicateurs ou raffiner des indicateurs existants. Néanmoins, si un travail considérable a été consacré au développement de cadres méthodologiques [8], [9], de lignes directrices sur la qualité [10], ainsi qu'à la conception de processus sécurisant l'acquisition de données auprès de tiers [11], les infrastructures informatiques et les compétences nécessaires pour gérer ces nouveaux objets sont restées peu abordées dans la littérature.

Pourtant, ces nouvelles sources présentent des caractéristiques qui rendent leur traitement informatique complexe. On qualifie souvent de *big data* ces données qui se distinguent par leur volume (souvent de l'ordre de plusieurs centaines de Go voire du To), leur vélocité (vitesse de génération, souvent proche du temps réel) ou de leur variété (données structurées mais aussi non structurées, telles que les textes et les images). Or les « compétences pour automatiser, analyser et optimiser ces systèmes complexes ne font souvent pas partie des compétences traditionnelles de la plupart des instituts statistiques nationaux » [12]. Au cours des dernières années, on observe un nombre croissant de statisticiens publics formés aux méthodes de *data science*, permettant d'envisager l'intégration de ces sources dans des processus de production. Dans ses multiples acceptions, le terme « *data scientist* » reflète en effet l'implication croissante des statisticiens dans le développement informatique et l'orchestration de leurs opérations de traitement des données, au-delà des seules phases de conception ou de validation [13]. Toutefois, il est clair en pratique, à l'Insee et dans d'autres organisations, que la capacité de ces profils à tirer parti des sources *big data* et des méthodes d'apprentissage automatique se heurte à plusieurs défis.

Un premier défi réside dans l'absence d'infrastructures informatiques adaptées aux nouvelles sources de données auxquelles les INS ont désormais accès, ainsi qu'au besoin croissant de nouvelles méthodes statistiques. Par exemple, les sources *big data* nécessitent d'énormes capacités de stockage et s'appuient souvent sur des infrastructures et des méthodes de calcul distribué pour être traitées [14]. De même, l'adoption de nouvelles méthodes statistiques basées sur des algorithmes d'apprentissage automatique requiert des capacités informatiques — en particulier des GPU (processeurs graphiques) dans le cadre du traitement du texte ou de l'image — pour paralléliser massivement les calculs [15]. De telles ressources sont rarement disponibles dans les infrastructures informatiques traditionnelles. Lorsque des infrastructures de calcul adaptées sont disponibles, comme les supercalculateurs (HPC) utilisés dans certains domaines de recherche, elles nécessitent des compétences spécifiques — notamment pour leur mise en place et leur maintenance — qui sont rarement disponibles au sein des INS.

Un autre défi majeur pour les statisticiens est de disposer d'environnements de développement leur permettant d'expérimenter plus librement. L'essence de l'innovation dans les travaux statistiques réside dans la capacité à intégrer rapidement de nouveaux outils et méthodologies. Cette agilité est limitée lorsque les statisticiens dépendent excessivement des départements informatiques pour provisionner des ressources ou installer de nouveaux logiciels. Dans les configurations traditionnelles — ordinateurs personnels ou bureaux virtuels sur des architectures centralisées[1] —

[1] `AUSv3` est un exemple d'une telle infrastructure. Le statisticien utilise son poste de travail comme point d'accès à un bureau virtuel qui « reproduit » l'expérience habituelle du poste de travail. Néanmoins, les calculs qui sont

les départements informatiques privilégient généralement la sécurité et la stabilité du système au détriment de la fourniture de nouveaux services, ce qui limite le potentiel d'innovation. De plus, ces environnements rigides rendent difficile la mise en œuvre de bonnes pratiques de développement, telles que le travail collaboratif — nécessitant des environnements permettant de partager facilement des expérimentations avec ses pairs — et la reproductibilité.

Un troisième défi concerne la difficulté de passer des expérimentations innovantes à des solutions en production. Même lorsque les statisticiens ont accès à des environnements leur permettant d'expérimenter aisément, la transition vers le déploiement d'une application ou d'un modèle reste généralement difficile. Les environnements de production diffèrent souvent des environnements de développement, ce qui entraîne des coûts de développement supplémentaires importants pour passer d'une preuve de concept à une solution industrialisée qui rend du service dans la durée. Par exemple, dans le cas des projets d'apprentissage automatique, les modèles déployés nécessitent un suivi rigoureux pour s'assurer qu'ils conservent leur précision et leur utilité au fil du temps, et requièrent généralement des améliorations périodiques ou continues. Ces besoins plaident pour des environnements plus flexibles permettant aux statisticiens de gérer de manière autonome le cycle de vie complet de leurs projets de *data science*.

Ces différents défis ont un thème sous-jacent commun : le besoin d'une plus grande autonomie. La capacité des méthodes de *data science* à améliorer et potentiellement transformer la production des statistiques officielles dépend crucialement de la capacité des statisticiens à mener des expérimentations innovantes. Pour ce faire, ils doivent avoir accès à des ressources informatiques substantielles et diversifiées leur permettant de gérer le volume et la diversité des sources *big data* et d'exploiter les méthodes d'apprentissage automatique. Ces projets expérimentaux nécessitent des environnements de développement flexibles favorisant le travail collaboratif pour tirer parti de la diversité des profils et compétences des équipes projet. Enfin, pour tirer pleinement parti de ces expérimentations, les statisticiens ont besoin d'outils pour déployer des applications sous forme de preuves de concept et orchestrer leurs opérations statistiques en toute autonomie.

Dans ce contexte, l'Insee a développé `Onyxia` : un projet open source permettant aux organisations de déployer des plateformes de *data science* favorisant l'innovation en offrant aux statisticiens une plus grande autonomie[2]. Cet article vise à décrire le processus de réflexion ayant conduit à ce projet et à illustrer comment il autonomise les statisticiens à l'Insee, devenant ainsi un pilier de notre stratégie d'innovation. La section 2 offre une analyse approfondie des derniers développements de l'écosystème de la donnée, mettant en lumière les choix technologiques qui ont façonné le développement d'un environnement moderne de *data science*, adapté aux besoins spécifiques des statisticiens. En particulier, nous montrons comment les technologies *cloud* — en particulier la conteneurisation et le stockage objet — permettent de créer des environnements évolutifs et flexibles qui favorisent l'autonomie tout en promouvant la reproductibilité des projets statistiques. Malgré leurs atouts pour les applications modernes de *data science*, la complexité de configuration et d'utilisation des technologies *cloud* est souvent un obstacle à leur adoption. Dans la section

lancés — via `R` ou `Python` par exemple — sont effectués sur des machines virtuelles (VM) de calcul dédiées, et non sur le poste de travail de l'utilisateur.

[2]https://github.com/InseeFrLab/onyxia

3, nous détaillons en quoi le projet `Onyxia` permet précisément de rendre les technologies *cloud* accessibles aux statisticiens grâce à une interface dynamique et accessible et un catalogue étendu de services de *data science* prêts à l'emploi. Enfin, la section 4 illustre l'application pratique de ces technologies à un projet innovant de l'Insee : la classification des activités des entreprises françaises (APE) à l'aide de méthodes d'apprentissage automatique. Ce retour d'expérience vise à montrer comment l'utilisation de ces technologies permet de faciliter et fiabiliser la mise en production de modèles d'apprentissage en permettant d'appliquer les meilleures pratiques issues du *MLOps*.

2 Principes pour construire une architecture de données moderne et flexible pour les statistiques publiques

Avec l'émergence de sources de données massives et de nouvelles méthodologies prometteuses pour améliorer le processus de production des statistiques publiques, les statisticiens formés aux techniques de *datascience* sont désireux d'innover. Cependant, leur capacité à le faire est limitée par plusieurs défis. L'un des principaux défis réside dans le besoin d'une plus grande autonomie — qu'il s'agisse de dimensionner la puissance de calcul en fonction des chaînes de productions statistiques, de déployer des preuves de concept avec agilité et de manière collaborative, etc. Dans ce contexte, notre objectif était de concevoir une plateforme de *datascience* qui non seulement gère efficacement les données massives, mais qui renforce également l'autonomie des statisticiens. Pour y parvenir, nous avons étudié l'évolution de l'écosystème des données afin d'identifier les tendances significatives susceptibles de surmonter les limitations mentionnées précédemment[3]. Nos conclusions indiquent que l'adoption des technologies *cloud*, en particulier les conteneurs et le stockage objet, est essentielle pour construire des infrastructures capables de gérer des ensembles de données volumineux et variés de manière flexible et économique. De plus, ces technologies améliorent considérablement l'autonomie, facilitant ainsi l'innovation et favorisant la reproductibilité dans la production des statistiques publiques.

2.1 Limites des architectures traditionnelles pour les big data

Au cours de la dernière décennie, le paysage des *big data* s'est transformé de manière spectaculaire. Suite à la publication des articles fondateurs de Google introduisant le *MapReduce* [16]–[18], les systèmes basés sur Hadoop sont rapidement devenus l'architecture de référence dans l'écosystème des données massives, salués pour leur capacité à gérer d'importants ensembles de données grâce aux calculs distribués. L'introduction d'Hadoop a marqué une étape révolutionnaire, permettant aux organisations de traiter et d'analyser des données à une échelle sans précédent. En substance, Hadoop offrait aux entreprises des capacités complètes pour l'analyse de *big data* :

[3]En préambule à cette analyse, il convient de noter que, bien que nous ayons fait de notre mieux pour ancrer nos réflexions dans la littérature académique, une grande partie de nos observations provient de connaissances informelles acquises grâce à une veille technologique assidue et continue. Dans l'écosystème des données, qui est en constante évolution, les articles de recherche traditionnels cèdent de plus en plus la place aux billets de blog en tant que principales références pour les développements de pointe. Ce changement s'explique en grande partie par le rythme soutenu auquel les technologies et méthodologies liées aux données massives progressent, rendant souvent le processus de publication formelle trop long pour la diffusion de connaissances et d'innovations.

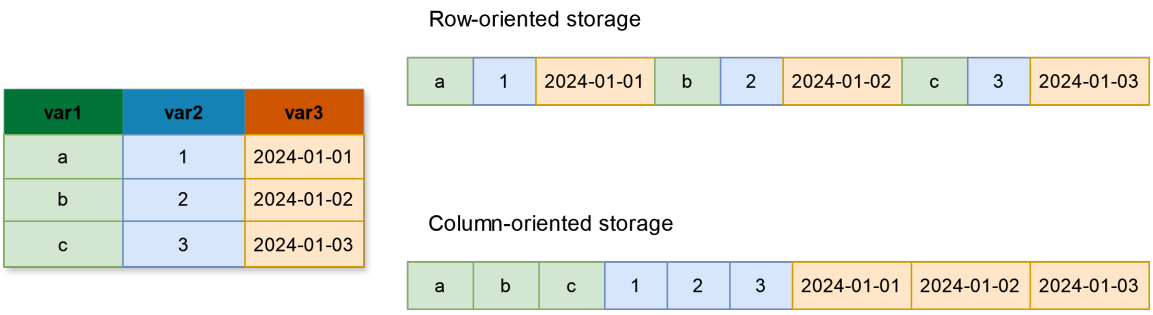
des outils pour la collecte, le stockage des données (HDFS) et des capacités de calcul (notamment Spark) [19], expliquant ainsi son adoption rapide dans les industries.

À la fin des années 2010, les architectures basées sur Hadoop ont connu un net déclin de popularité. Dans les environnements Hadoop traditionnels, le stockage et le calcul étaient co-localisés par construction : si les données sources étaient réparties sur plusieurs serveurs (scalabilité horizontale), chaque section des données était directement traitée sur la machine hébergeant cette section, afin d'éviter les transferts réseau entre serveurs. Dans ce paradigme, la mise à l'échelle de l'architecture impliquait souvent une augmentation linéaire à la fois des capacités de calcul et de stockage, indépendamment de la demande réelle. Dans un article volontairement provocateur et intitulé « Big Data is Dead » [20], Jordan Tigani, l'un des ingénieurs fondateurs de Google BigQuery, explique pourquoi ce modèle ne correspond plus à la réalité de la plupart des organisations centrées sur les données. Premièrement, parce que « dans la pratique, la taille des données augmente beaucoup plus rapidement que les besoins en calcul ». Alors que la quantité de données générées et nécessitant donc d'être stockées peut croître de manière linéaire au fil du temps, il est généralement vrai que nous n'aillons besoin d'interroger que les portions les plus récentes, ou seulement certaines colonnes et/ou groupes de lignes. Par ailleurs, Tigani souligne que « la frontière du *big data* ne cesse de reculer » : les avancées dans les capacités des serveurs et la baisse des coûts du matériel signifient que le nombre de charges de travail ne tenant pas sur une seule machine — une définition simple mais efficace du *big data* — a diminué de manière continue. En conséquence, en séparant correctement les fonctions de stockage et de calcul, même les traitements de données substantiels peuvent finir par utiliser « beaucoup moins de calcul que prévu […] et pourraient même ne pas avoir besoin d'un traitement distribué du tout ».

Ces observations concordent fortement avec nos propres constats à l'Insee au cours des dernières années. Par exemple, une équipe de l'Insee a mis en place un cluster Hadoop en tant qu'architecture alternative à celle déjà utilisée pour traiter les données des tickets de caisse dans le cadre du calcul de l'indice des prix à la consommation. Une accélération des opérations de traitement des données pouvant aller jusqu'à un facteur 10 a été obtenue, pour des opérations qui prenaient auparavant plusieurs heures [21]. Malgré cette amélioration des performances, ce type d'architectures n'a pas été réutilisé par la suite pour d'autres projets, principalement parce que l'architecture s'est révélée coûteuse et complexe à maintenir, nécessitant une expertise technique spécialisée rarement disponible au sein des Instituts Nationaux de Statistiques (INS) [22]. Bien que ces nouveaux projets puissent encore impliquer des volumes de données massifs, nous avons observé que des traitements efficaces pouvaient être réalisés à l'aide de logiciels conventionnels (R, Python) sur des systèmes à nœud unique, en tirant parti des récentes innovations importantes de l'écosystème des données. Tout d'abord, en utilisant des formats de stockage efficaces tels qu'Apache Parquet [23], dont les propriétés — stockage en colonnes [24] (voir Figure 1), optimisation pour les analyses « écrire une fois, lire plusieurs fois », possibilité de partitionner les données, etc. — le rendent particulièrement adapté aux tâches analytiques comme celles généralement effectuées dans les statistiques publiques [17]. Ensuite, en effectuant des calculs optimisés en mémoire tels qu'Apache Arrow [25] ou DuckDB [26] le proposent. Également basés sur une représentation en colonnes — travaillant ainsi en synergie avec les fichiers Parquet — ces deux logiciels améliorent considérablement les performances des requêtes de données grâce à l'utilisation de l'"éva-

luation paresseuse" (*lazy evaluation*) : au lieu d'exécuter de nombreuses opérations distinctes (par exemple, sélectionner des colonnes et/ou filtrer des lignes, puis calculer de nouvelles colonnes, puis effectuer des agrégations, etc.), ils les traitent toutes en une fois de manière plus optimisée. En conséquence, les calculs se limitent aux données effectivement nécessaires pour les requêtes, permettant le traitement de données beaucoup plus importantes que la mémoire disponible sur des machines classiques à nœud unique.

Figure 1. – Représentation orientée ligne et orientée colonne d'un même jeu de données.



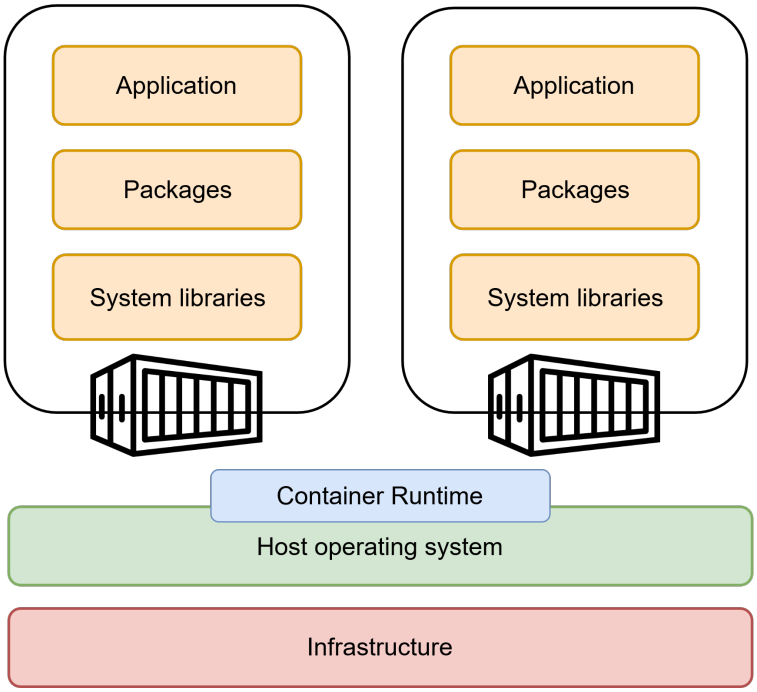
Note: De nombreuses opérations statistiques sont analytiques (OLAP) par nature : elles impliquent la sélection de colonnes spécifiques, le calcul de nouvelles variables, la réalisation d'agrégations basées sur des groupes, etc. Le stockage orienté ligne n'est pas bien adapté à ces opérations analytiques, car il nécessite de charger l'ensemble du jeu de données en mémoire afin d'effectuer une requête. À l'inverse, le stockage orienté colonne permet de ne lire que les colonnes de données pertinentes, ce qui réduit considérablement les temps de lecture et de traitement pour ces charges de travail analytiques. En pratique, les formats colonnes populaires tels que Parquet utilisent une représentation hybride : ils sont principalement orientés colonne, mais intègrent également un regroupement astucieux basé sur les lignes pour optimiser les requêtes de filtrage.

2.2 Adopter les technologies cloud

Suite à cette évolution de l'écosystème des big data, on observe un virage notable ces dernières années dans l'industrie vers des architectures plus flexibles et faiblement couplées. L'avènement des technologies *cloud* a joué un rôle déterminant dans cette transition. Contrairement à l'époque où Hadoop dominait, la latence réseau est devenue une préoccupation bien moindre, rendant le modèle traditionnel de solutions de stockage et de calcul sur site et co-localisées moins pertinent. Concernant la nature des données à traiter, on observe une évolution que certains ont qualifiée de passage « du big data aux données flexibles ». Les infrastructures modernes doivent non seulement être capables de traiter de grands volumes, mais aussi être adaptables sur de multiples dimensions. Elles doivent pouvoir prendre en charge diverses structures de données (allant des formats structurés et tabulaires aux formats non structurés comme le texte et les images), assurer la portabilité des données dans des environnements *multi-cloud* et *cloud* hybride, et prendre en charge une large gamme de calculs computationnels (des calculs parallèles aux modèles d'apprentissage profond nécessitant des GPU, ainsi que le déploiement et la gestion d'applications) [27]. Ces dernières années, deux technologies ont émergé comme des éléments fondamentaux pour atteindre cette flexibilité dans les environnements *cloud* : la conteneurisation et le stockage d'objets.

Dans un environnement *cloud*, l'ordinateur de l'utilisateur devient un simple point d'accès pour effectuer des calculs sur une infrastructure centrale. Cela permet à la fois un accès ubiquitaire et une scalabilité des services, car il est plus facile de mettre à l'échelle une infrastructure centrale — généralement de manière horizontale, c'est-à-dire en ajoutant davantage de serveurs. Cependant, ces infrastructures centralisées présentent deux limites bien identifiées qui doivent être prises en compte : la concurrence entre utilisateurs pour l'accès aux ressources physiques et la nécessité d'isoler correctement les applications déployées. Le choix de la conteneurisation est fondamental, car il répond à ces deux enjeux [28]. En créant des « bulles » spécifiques à chaque service, les conteneurs garantissent l'isolement des applications tout en restant légers, puisqu'ils partagent le système d'exploitation de support avec la machine hôte (voir Figure 2). Pour gérer plusieurs applications conteneurisées de manière systématique, les infrastructures conteneurisées s'appuient généralement sur un logiciel orchestrateur — le plus connu étant Kubernetes, un projet open source initialement développé par Google pour gérer ses nombreuses charges de travail conteneurisées en production [29]. Les orchestrateurs automatisent le processus de déploiement, de mise à l'échelle et de gestion des applications conteneurisées, coordonnant leur exécution sur différents serveurs. De manière intéressante, cette propriété permet de traiter de très grands volumes de données de manière distribuée : les conteneurs décomposent les opérations de traitement des données massives en une multitude de petites tâches, organisées par l'orchestrateur. Cela minimise les ressources requises tout en offrant une flexibilité supérieure aux architectures basées sur Hadoop [30].

Figure 2. – Architecture d'un environnement conteneurisé.



Note: Un conteneur est un regroupement logique de ressources permettant d'encapsuler une application (par exemple, du code R), les bibliothèques utilisées (par exemple, ggplot, dplyr) et les

bibliothèques système (l'interpréteur R, d'autres bibliothèques dépendantes du système d'exploitation, etc.) dans un seul package. Les applications conteneurisées sont isolées les unes des autres grâce à la virtualisation, ce qui permet d'attribuer des ressources physiques spécifiques à chaque application tout en garantissant leur indépendance totale. Contrairement aux machines virtuelles, qui virtualisent également le système d'exploitation (OS), les conteneurs s'appuient sur une forme de virtualisation légère : le conteneur partage l'OS de l'infrastructure hôte via le runtime de conteneur (par exemple, Docker). En conséquence, les conteneurs sont beaucoup plus portables et peuvent être déployés et redistribués facilement.

L'autre choix fondamental dans une architecture de données concerne la nature du stockage de ces données. Dans l'écosystème *cloud*, le « stockage d'objets » est devenu la référence *de facto* [31][4]. Dans ce paradigme, les fichiers sont stockés sous forme d'"objets" composés de données, d'un identifiant et de métadonnées. Ce type de stockage est optimisé pour la scalabilité, car les objets ne sont pas limités en taille et la technologie sous-jacente permet un stockage rentable de fichiers (potentiellement très) volumineux. Le stockage d'objets joue également un rôle clé dans la construction d'une infrastructure découplée comme celle évoquée précédemment : les dépôts de données — appelés *« buckets »* — sont directement interrogeables via des requêtes HTTP standards grâce à une API REST normalisée. Dans un contexte où la latence réseau n'est plus le principal goulot d'étranglement, cela signifie que le stockage et le calcul n'ont pas besoin d'être sur les mêmes machines, ni même dans le même lieu. Ils peuvent ainsi évoluer indépendamment en fonction des besoins spécifiques de l'organisation. Enfin, le stockage d'objets est un complément naturel aux architectures basées sur des environnements conteneurisés. Il fournit une couche de persistance — les conteneurs étant par construction sans état (*stateless*) — et une connectivité facile, sans compromettre la sécurité, voire en renforçant celle-ci par rapport à un système de stockage traditionnel [32].

2.3 Exploiter les technologies cloud pour accroître l'autonomie et favoriser la reproductibilité

Comprendre comment les choix technologiques décrits dans la discussion technique ci-dessus sont pertinents dans le contexte des statistiques publiques nécessite un examen approfondi des pratiques professionnelles des statisticiens dans leur utilisation des environnements informatiques. À la fin des années 2000, alors que la micro-informatique était à son apogée, une grande partie des ressources techniques utilisées par les statisticiens de l'Insee étaient locales : le code et les logiciels de traitement étaient situés sur des ordinateurs personnels, tandis que les données étaient accessibles via un système de partage de fichiers. En raison de la scalabilité limitée des ordinateurs personnels, cette configuration restreignait considérablement la capacité des statisticiens à expérimenter avec des sources *big data* ou des méthodes statistiques intensives en calculs, et cela impliquait des risques de sécurité liés à la diffusion étendue des données au sein de l'organisation. Pour surmonter ces limitations, une transition a été opérée vers des infrastructures informatiques centralisées, regroupant toutes les ressources — et donc globalement beaucoup plus — sur des serveurs centraux. Ces infrastructures, mises à disposition des statisticiens via un envi-

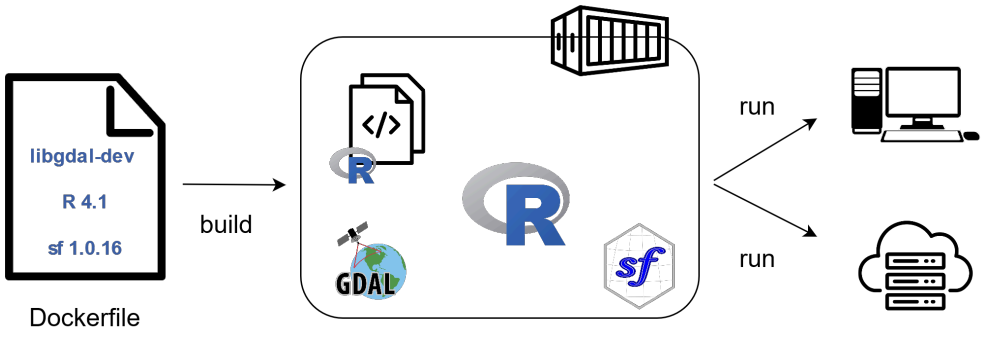
[4]Principalement grâce à l'implémentation « S3 » (Simple Storage Service) d'Amazon.

ronnement de bureau virtuel partagé pour faciliter leur utilisation, constituent encore la méthode dominante pour réaliser des calculs statistiques à l'Insee au moment de la rédaction de ces lignes.

À travers nos observations et nos discussions avec d'autres statisticiens, il est devenu évident que, bien que l'infrastructure informatique actuelle soutienne adéquatement les activités fondamentales de production statistique, elle restreint de manière notable la capacité des statisticiens à expérimenter librement et à innover. Le principal goulot d'étranglement dans cette organisation réside dans la dépendance des projets statistiques à la prise de décision centralisée en matière d'informatique, notamment en ce qui concerne l'allocation des ressources de calcul, l'accès au stockage partagé, l'utilisation de langages de programmation préconfigurés etc. En outre, ces dépendances conduisent souvent à un phénomène bien connu dans la communauté du développement logiciel, où les priorités des développeurs — itérer rapidement pour améliorer continuellement les fonctionnalités — entrent souvent en conflit avec l'objectif des équipes informatiques de garantir la sécurité et la stabilité des processus. À l'inverse, nous comprenons que les pratiques modernes en *datascience* reflètent une implication accrue des statisticiens dans le développement et l'orchestration informatique de leurs opérations de traitement de données, au-delà de la simple phase de conception ou de validation. Les nouvelles infrastructures de *datascience* doivent donc prendre en compte ce rôle élargi de leurs utilisateurs, en leur offrant plus d'autonomie que les infrastructures traditionnelles.

Nous soutenons que les technologies *cloud* sont une solution puissante pour offrir aux statisticiens une autonomie bien plus grande dans leur travail quotidien, favorisant ainsi une culture de l'innovation. Grâce au stockage d'objets, les utilisateurs obtiennent un contrôle direct sur la couche de stockage, leur permettant d'expérimenter avec des sources de données diverses sans être limités par les espaces de stockage souvent restreints et alloués par les départements informatiques. La conteneurisation permet aux utilisateurs de personnaliser leurs environnements de travail selon leurs besoins spécifiques — qu'il s'agisse de langages de programmation, de bibliothèques système ou de versions de packages — tout en leur offrant la flexibilité nécessaire pour adapter leurs applications à la puissance de calcul et aux capacités de stockage requises. Par construction, les conteneurs favorisent également le développement d'applications portables, permettant des transitions plus fluides entre les environnements (développement, test, pré-production, production), en garantissant que les applications peuvent être exécutées sans difficulté, évitant ainsi les problèmes liés aux incohérences d'environnement. Enfin, avec des outils d'orchestration tels que Kubernetes, les statisticiens peuvent déployer plus facilement des applications et des API, tout en automatisant l'ensemble du processus de construction. Cette capacité s'aligne avec l'approche DevOps, qui préconise la création de preuves de concept de manière itérative, plutôt que de chercher à développer la solution optimale (mais chronophage) pour un objectif préalablement défini [33].

Figure 3. – Par construction, les conteneurs favorisent la reproductibilité et la portabilité.



Note: Dans un environnement conteneurisé, les applications sont créées à partir de spécifications sous forme de scripts — un paradigme connu sous le nom d'*"infrastructure as code"*. Dans un fichier texte, conventionnellement nommé « Dockerfile », les *data scientists* peuvent spécifier l'environnement de travail de leur application : le code de l'application, les logiciels à inclure (par exemple, R), les packages utilisés pour leurs opérations de traitement (par exemple, le package R pour le calcul géospatial *sf*), ainsi que les bibliothèques système dépendant de l'OS appelées par ces packages (par exemple, GDAL, la bibliothèque qui permet de lire et de traiter les formats d'images géospatiales utilisée par la plupart des packages traitant des données géospatiales). Un point essentiel est que les versions des logiciels et des packages utilisés pour développer l'application peuvent être précisément spécifiées, ce qui garantit la reproductibilité des calculs effectués. Une étape de construction génère ensuite une image associée au Dockerfile, c'est-à-dire une forme empaquetée et compressée de l'environnement de travail de l'application. Les images créées de cette manière sont portables : elles peuvent être facilement distribuées — généralement via un registre de conteneurs — et exécutées de manière reproductible sur n'importe quelle infrastructure disposant d'un runtime de conteneur.

Outre la scalabilité et l'autonomie, ces choix architecturaux favorisent également la reproductibilité des calculs statistiques. Le concept de reproductibilité — à savoir la capacité de reproduire le résultat d'une expérience en appliquant la même méthodologie aux mêmes données — est un critère fondamental de validité scientifique [34]. Il est également très pertinent dans le domaine des statistiques publiques, car il constitue une base pour la transparence, essentielle pour établir et maintenir la confiance du public [35]. Favoriser la reproductibilité dans la production statistique implique de concevoir des solutions de traitement capables de produire des statistiques reproductibles, tout en étant partageables entre pairs [36]. Les infrastructures informatiques traditionnelles — qu'il s'agisse d'un ordinateur personnel ou d'une infrastructure partagée avec un accès à distance — sont insuffisantes à cet égard. Construire un projet ou calculer un simple indicateur statistique dans ces environnements implique généralement une série d'étapes manuelles (installation des bibliothèques système, des binaires du langage de programmation, des packages du projet, gestion des versions conflictuelles, etc.) qui ne peuvent pas être pleinement reproduites d'un projet à l'autre. En comparaison, les conteneurs sont reproductibles par définition, car leur processus de construction implique de définir précisément toutes les ressources nécessaires comme un ensemble d'opérations standardisées, allant de la « machine nue » à l'application en cours d'exécu-

tion [37]. De plus, ces environnements reproductibles peuvent être facilement partagés avec des pairs, car ils peuvent être publiés sur des registres ouverts (par exemple, un registre de conteneurs comme DockerHub) avec le code source de l'application (par exemple, sur une forge logicielle publique comme GitHub ou GitLab). Cette approche améliore considérablement la réutilisation des projets de code, favorisant un modèle de développement et d'innovation basé sur la collaboration communautaire.

3 Onyxia : un projet open source pour construire des plateformes de data science sur des technologies cloud

Cette section examine comment Onyxia, un projet open source initié par l'Insee, démocratise l'accès aux technologies *cloud* pour les statisticiens en fournissant des environnements modernes de *data science* favorisant l'autonomie. Nous analysons comment cette initiative s'inscrit dans l'objectif général de création des « connaissances communes » en promouvant et en développant des logiciels facilement réutilisables dans le domaine des statistiques publiques et ailleurs.

3.1 Rendre les technologies cloud accessibles aux statisticiens

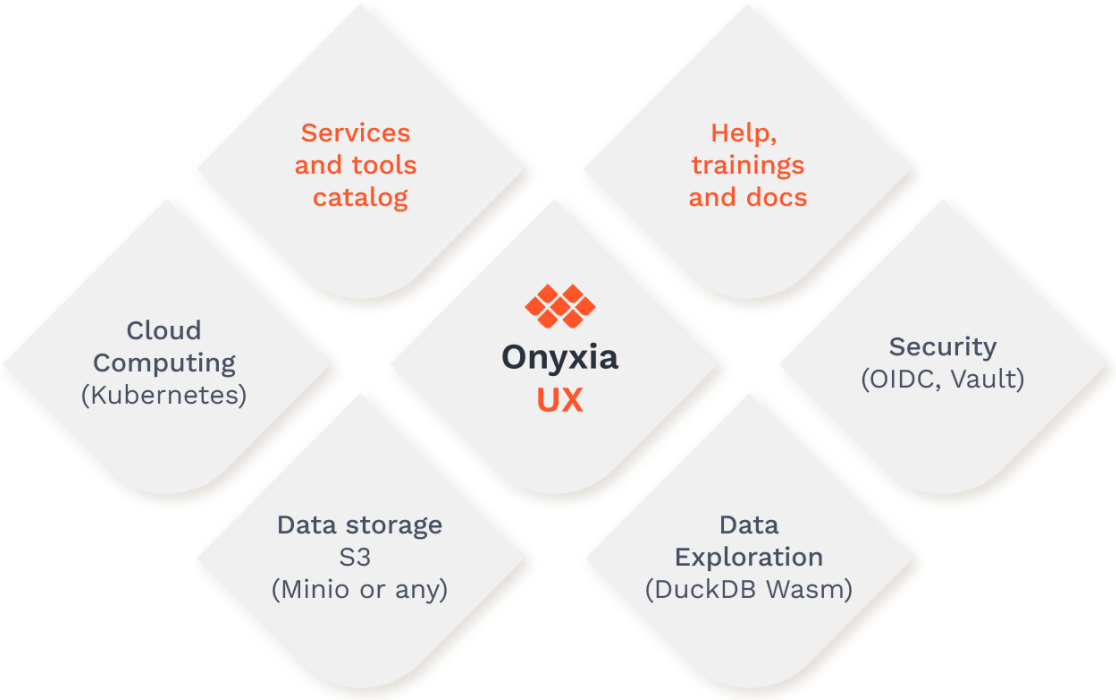
Notre veille technologique et notre revue de la littérature ont mis en évidence les technologies *cloud*, en particulier la conteneurisation et le stockage d'objets, comme des éléments clés pour construire une plateforme de *data science* à la fois scalable et flexible. En nous appuyant sur ces enseignements, nous avons mis en place notre premier cluster Kubernetes dans les locaux de l'Insee en 2020, en l'intégrant avec MinIO, un système de stockage d'objets *open source* conçu pour fonctionner de manière fluide avec Kubernetes. Cependant, nos premières expérimentations ont révélé un obstacle majeur à l'adoption généralisée des technologies *cloud* : la complexité de leur intégration. C'est une considération importante lorsqu'il s'agit de construire des architectures de données qui privilégient la modularité — une caractéristique essentielle pour atteindre la flexibilité que nous visons[5]. Toutefois, la modularité des composants architecturaux implique également que toute application de données lancée sur le cluster doit être configurée pour communiquer avec tous les autres composants. Par exemple, dans un environnement *big data*, la configuration de Spark pour fonctionner sur Kubernetes tout en interagissant avec des ensembles de données stockés dans MinIO nécessite de nombreuses et complexes configurations (définition des points d'entrée, des jetons d'accès, etc.), une compétence qui dépasse généralement l'expertise des statisticiens.

Par exemple, grâce à la compatibilité de MinIO avec l'API S3 d'Amazon, la source de stockage pourrait facilement être remplacée par une solution gérée par un autre fournisseur de *cloud* public, sans nécessiter de modifications substantielles.

[5]Un exemple révélateur de l'importance de construire une architecture modulaire est la capacité de basculer entre différentes sources de stockage (on-premise, fournisseur de *cloud* public, etc.). La solution de stockage que nous avons choisie, MinIO, est compatible avec l'API S3 d'Amazon, qui est devenue un standard *de facto* dans l'écosystème *cloud* grâce au succès de la solution de stockage S3 d'Amazon AWS. Ainsi, les organisations qui choisissent d'utiliser Onyxia ne sont pas liées à une solution de stockage spécifique : elles peuvent opter pour n'importe quelle solution conforme aux standards définis par l'API S3.

Cette idée est véritablement le fondement du projet Onyxia : choisir des technologies qui favorisent l'autonomie ne remplira pas cet objectif si leur complexité constitue une barrière à une adoption généralisée au sein de l'organisation. Ces dernières années, les statisticiens de l'Insee ont déjà dû s'adapter à un environnement en évolution en ce qui concerne leurs outils quotidiens : passer de logiciels propriétaires (SAS®) à des outils open source (R, Python), s'approprier des technologies qui améliorent la reproductibilité (contrôle de version avec Git), consommer et développer des API, etc. Ces changements, rendant leur travail de plus en plus semblable à celui de développeurs logiciels, impliquent déjà des efforts considérables en termes de formation et des modifications des pratiques de travail quotidiennes. Dans ce contexte, l'adoption des technologies *cloud* dépend totalement de leur accessibilité immédiate.

Figure 4. – Onyxia est le lien technique entre les composants modulaires du cloud.



Pour combler cet écart, nous avons développé Onyxia, une application qui agit essentiellement comme une interface entre les composants modulaires qui composent l'architecture (voir Figure 4). Le point d'entrée principal pour l'utilisateur est une application web ergonomique[6] qui lui permet de lancer des services à partir d'un catalogue de *data science* (voir Chapitre 3.3) sous forme de conteneurs exécutés sur le cluster Kubernetes sous-jacent. Le lien entre l'interface utilisateur (UI) et Kubernetes est assurée par une API[7], qui transforme essentiellement la demande d'application de l'utilisateur en un ensemble de manifestes nécessaires pour déployer des ressources Kubernetes. Pour une application donnée, ces ressources sont regroupées sous la forme de *charts* Helm, une méthode populaire pour empaqueter des applications potentiellement complexes sur

[6] https://github.com/InseeFrLab/onyxia-ui
[7] https://github.com/InseeFrLab/onyxia-api

Kubernetes [38]. Bien que les utilisateurs puissent configurer un service pour l'adapter à leurs besoins, la plupart du temps, ils se contentent de lancer un service prêt à l'emploi avec des paramètres par défaut et commencent à développer immédiatement. Ce point illustre parfaitement la valeur ajoutée d'Onyxia pour faciliter l'adoption des technologies cloud. En injectant automatiquement les informations d'authentification et de configuration dans les conteneurs lors de leur initialisation, nous veillons à ce que les utilisateurs puissent lancer et gérer des services de *data science* dans lesquels ils interagissent sans difficulté avec les données de leur *bucket* sur MinIO, leurs informations sensibles (jetons, mots de passe) dans un outil de gestion des secrets tel que Vault, etc. Cette injection automatique, associée à la pré-configuration des environnements de *data science* dans les catalogues d'images[8] et de *charts* Helm[9] d'Onyxia, permet aux utilisateurs d'exécuter des scripts potentiellement complexes — comme des calculs distribués avec Spark sur Kubernetes à l'aide de données stockées sur S3, ou l'entraînement de modèles d'apprentissage profond utilisant un GPU — sans se heurter aux difficultés techniques liées à la configuration.

3.2 Des choix architecturaux visant à favoriser l'autonomie des statisticiens

Le projet Onyxia repose sur quelques principes structurants, avec un thème central : favoriser l'autonomie, à la fois au niveau organisationnel et individuel. Tout d'abord, au niveau de l'organisation, en évitant l'enfermement propriétaire. Pour obtenir un avantage concurrentiel, de nombreux fournisseurs de *cloud* commerciaux développent des applications et protocoles spécifiques que les clients doivent utiliser pour accéder aux ressources *cloud*, mais qui ne sont pas interopérables, compliquant considérablement les migrations potentielles vers une autre plateforme *cloud* [39]. Sachant cela, une tendance émerge vers l'adoption de stratégies neutres vis-à-vis des *clouds* [40] afin de réduire la dépendance à des solutions spécifiques d'un seul fournisseur. En revanche, l'utilisation d'Onyxia n'est intrinsèquement pas restrictive : lorsqu'une organisation choisit de l'utiliser, elle choisit les technologies sous-jacentes — la conteneurisation et le stockage d'objets — mais pas la solution en elle-même. La plateforme peut être déployée sur n'importe quel cluster Kubernetes, qu'il soit *on-premise* ou sur des *clouds* commerciaux. De même, Onyxia a été conçue pour être utilisée avec MinIO, car il s'agit d'une solution de stockage d'objets *open source*, mais il est également possible de l'utiliser avec les solutions de stockage d'objets proposées par divers fournisseurs de *cloud* (AWS, GCP, etc.).

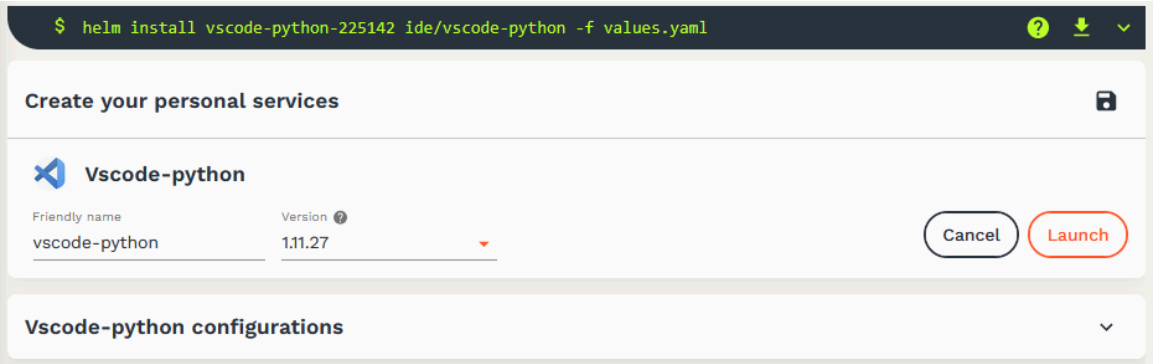
Onyxia favorise également l'autonomie au niveau des utilisateurs. Les logiciels propriétaires qui ont été intensivement utilisés dans les statistiques officielles — comme SAS ou STATA — induisent également un phénomène d'enfermement propriétaire. Les coûts des licences sont élevés et peuvent évoluer rapidement, et les utilisateurs se retrouvent dépendants de certaines méthodes de calcul, empêchant une montée en compétences progressive. Au contraire, Onyxia aspire à être amovible ; nous souhaitons améliorer la familiarité et le confort des utilisateurs avec les technologies *cloud* sous-jacentes plutôt que de devenir un élément permanent travail quotidien. Un exemple illustratif de cette philosophie est l'approche de la plateforme concernant les actions des utilisateurs : pour les tâches effectuées via l'interface utilisateur, comme le lancement d'un ser-

[8] https://github.com/InseeFrLab/images-datascience

[9] https://github.com/InseeFrLab/helm-charts-interactive-services

vice ou la gestion des données, nous fournissons aux utilisateurs les commandes terminal équivalentes, promouvant ainsi une compréhension plus approfondie de ce qui se passe réellement lors du déclenchement d'une action. De plus, tous les services proposés via le catalogue d'Onyxia sont *open source*.

Figure 5. – Lancer un service via l'interface web d'Onyxia.



Note: Les services du catalogue d'Onyxia peuvent être utilisés tels quels ou configurés par les utilisateurs pour répondre à leurs besoins spécifiques. Afin de limiter la dépendance des utilisateurs vis-à-vis d'Onyxia, chaque action effectuée par l'utilisateur via l'interface utilisateur est accompagnée de la commande exacte exécutée sur le cluster Kubernetes.

Naturellement, la manière dont Onyxia rend les statisticiens plus autonomes dans leur travail dépend de leurs besoins et de leur familiarité avec les compétences informatiques. Les statisticiens qui souhaitent simplement accéder à des ressources de calcul importantes pour expérimenter avec de nouvelles sources de données ou méthodes statistiques pourront, en quelques clics, accéder à des environnements de *data science* préconfigurés et faciles à utiliser, leur permettant de commencer à expérimenter immédiatement. Cependant, de nombreux utilisateurs souhaitent aller plus loin et développer de véritables prototypes d'applications de production pour leurs projets : configurer des scripts d'initialisation pour adapter les environnements à leurs besoins, déployer une application interactive offrant des visualisations de données aux utilisateurs de leur choix, ou encore déployer d'autres services que ceux disponibles dans nos catalogues. Pour permettre à ces utilisateurs avancés de continuer à repousser les limites de l'innovation, Onyxia leur donne accès au cluster Kubernetes sous-jacent. Cela signifie que les utilisateurs peuvent ouvrir librement un terminal sur un service interactif et interagir avec le cluster — dans les limites de leur *namespace* — afin d'appliquer des ressources personnalisées et de déployer des applications ou services personnalisés.

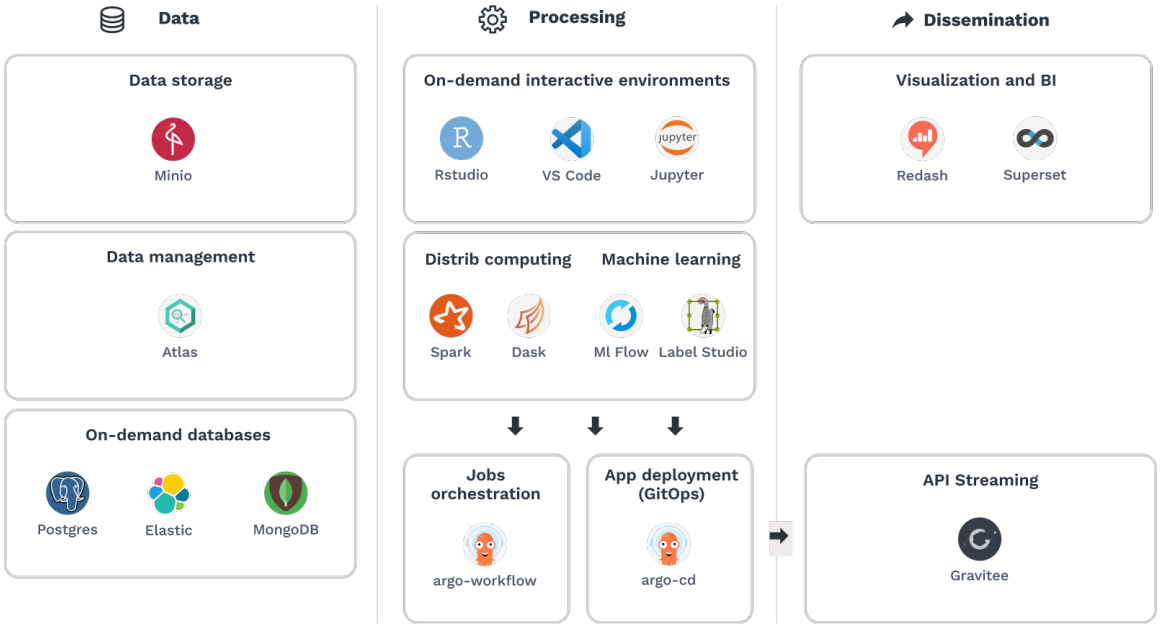
Au-delà de l'autonomie et de la scalabilité, les choix architecturaux d'Onyxia favorisent également la reproductibilité des calculs statistiques. Dans le paradigme des conteneurs, l'utilisateur doit apprendre à gérer des ressources qui sont par nature éphémères, puisqu'elles n'existent qu'au moment de leur mobilisation effective. Cela encourage l'adoption de bonnes pratiques de développement, notamment la séparation du code — hébergé sur une forge interne ou *open source* telle que GitLab ou GitHub —, des données — stockées sur une solution de stockage spécifique, comme MinIO —, et de l'environnement de calcul. Bien que cela impose un coût d'entré non né-

gligeable aux utilisateurs, cela les aide également à concevoir leurs projets sous forme de *pipelines*, c'est-à-dire une série d'étapes séquentielles avec des données en entrées et productions finales bien définies (semblables à un graphe orienté acyclique, ou DAG). Les projets développés de cette manière sont généralement plus reproductibles et transposables — ils peuvent fonctionner sans problème sur différents environnements de calcul — et sont ainsi plus facilement partageables avec leurs pairs.

3.3 Un catalogue exhaustif de services pour couvrir l'ensemble du cycle de vie des projets de data science

Lors du développement de la plateforme Onyxia, notre intention était de fournir aux statisticiens un environnement complet conçu pour accompagner le développement de bout en bout des projets de *data science*. Comme illustré dans Figure 6, la plateforme propose une vaste gamme de services couvrant l'ensemble du cycle de vie d'un projet de *data science*.

Figure 6. – Le catalogue d'Onyxia vise à couvrir l'ensemble du cycle de vie des projets de data science



L'utilisation principale de la plateforme est le déploiement d'environnements de développement interactifs (IDE), tels que RStudio, Jupyter ou VSCode. Ces IDE sont équipés des dernières versions des principaux langages de programmation *open source* couramment utilisés par les statisticiens publics (R, Python, Julia), ainsi que d'une vaste collection de librairies fréquemment employées en *data science* pour chaque langage. Afin de garantir que les services restent à jour et cohérents entre eux, nous maintenons nos propres images Docker sous-jacentes et les mettons à jour chaque semaine. Ces images sont entièrement *open source*[10] et peuvent donc être réutilisée en dehors d'Onyxia.

[10]https://github.com/InseeFrLab/images-datascience

Comme discuté dans les sections précédentes, la couche de persistance de ces environnements interactifs est principalement assurée par MinIO, la solution de stockage d'objets par défaut d'Onyxia. Étant basé sur une API REST standardisée, les fichiers peuvent être facilement interrogés depuis R ou Python à l'aide de librairies de haut niveau. Cela représente en soi une étape importante pour garantir la reproductibilité : les données ne sont pas sauvegardés localement, puis spécifiés via des chemins propres à une infrastructure ou un système de fichiers particulier. Au contraire, les fichiers sont spécifiés sous forme de requêtes HTTP, rendant la structure globale des projets bien plus extensible. D'après notre expérience, le paradigme du stockage d'objets répond très bien aux besoins de la plupart des projets statistiques que nous accompagnons. Cependant, des services de bases de données supplémentaires, tels que PostgreSQL et MongoDB, sont disponibles pour les applications ayant des besoins spécifiques, notamment celles nécessitant des capacités de traitement transactionnel en ligne (OLTP) ou un stockage orienté documents.

Comme Onyxia a été développée pour permettre l'expérimentation avec des sources de données volumineuses et des méthodes d'apprentissage automatique, nous proposons également des services optimisés pour passé à l'échelle facilement. Par exemple, des frameworks comme Spark et Trino, qui permettent d'effectuer des calculs distribués au sein d'un cluster Kubernetes. Ces services sont préconfigurés pour s'intégrer parfaitement avec le stockage S3, facilitant ainsi la création de *pipelines* de données intégrés et efficaces.

Au-delà de la simple expérimentation, notre objectif est de permettre aux statisticiens de passer des phases de test à des projets de qualité proche de celle requise en production afin de réduire le coût lors de la transmission d'un projet d'une équipe de production vers une équipe informatique. Conformément aux principes de l'approche DevOps, cela implique de faciliter le déploiement de prototypes et leur amélioration continue au fil du temps. À cette fin, nous proposons un ensemble de services *open source* visant à automatiser et industrialiser le processus de déploiement d'applications (ArgoCD, Argo-Workflows, MLflow). Pour les projets exploitant des modèles d'apprentissage automatique, les statisticiens peuvent exposer leurs modèles via des API, les déployer en utilisant les outils susmentionnés et gérer leur cycle de vie grâce à un gestionnaire d'API (par exemple, Gravitee). La Section 4 illustrera comment ces outils, en particulier MLflow, ont joué un rôle central dans la mise en production de modèles d'apprentissage automatique à l'Insee, en lien avec les principes de MLOps.

Dans la Chapitre 3.2, nous avons souligné qu'un des principes fondamentaux de conception d'Onyxia était d'éviter l'enfermement propriétaire. Dans cette optique, les organisations qui instancient Onyxia sont libres de personnaliser les catalogues pour répondre à leurs besoins spécifiques, ou même de créer leurs propres catalogues indépendamment des offres par défaut d'Onyxia. Cette flexibilité garantit aux organisations de ne pas être limité à une solution ou à un fournisseur unique, et qu'elles peuvent adapter la plateforme à l'évolution de leurs besoins.

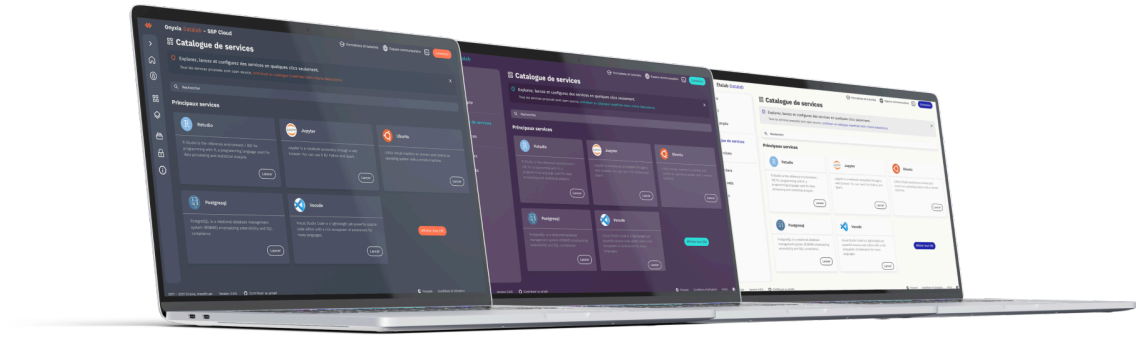
3.4 Construire des biens communs : un projet open source et une plateforme d'innovation ouverte

En tant qu'initiative entièrement *open source*, le projet Onyxia vise à construire des « connaissances communes » en promouvant et en développant des logiciels facilement réutilisables dans

les statistiques publiques et ailleurs [41]. Cela concerne, tout d'abord, les composants sur lesquels repose Onyxia : à la fois ses briques technologiques (Kubernetes, MinIO, Vault) et l'ensemble des services du catalogue, qui sont *open source*. Plus important encore, tout le code du projet est disponible publiquement sur GitHub[11]. Associée à une documentation détaillée[12], cette transparence facilite grandement la possibilité pour d'autres organisations de créer des instances de plateformes de *data science* basées sur le logiciel Onyxia et de les adapter à leurs besoins spécifiques (voir Figure 7). Cela a permis au projet d'attirer une communauté croissante de contributeurs issus des statistiques publiques (Statistique Norvège), des ONG (Mercator Ocean[13]), des centres de recherche et même de l'industrie, favorisant ainsi une transition progressive vers une gouvernance plus décentralisée du projet.

Dans les prochaines années, l'implication des INS (Instituts Nationaux de Statistique) du système statistique européen devrait augmenter, puisque le SSPCloud a été choisie comme plateforme *data science* de référence dans le cadre du projet AIML4OS[14].

Figure 7. – Un projet, de multiples instances : l'interface web est adaptable à l'identité graphique de chaque organisation



Une autre manière majeure de construire des communs est le développement et le maintien d'une instance de démonstration du projet Onyxia, le SSP Cloud [42]. Cette plateforme, équipée de ressources de calcul extensives et évolutives[15], est conçue comme un bac à sable pour expérimenter avec les technologies cloud et les nouvelles méthodes de science des données. Le catalogue complet des services d'Onyxia est disponible sur cette plateforme, permettant aux utilisateurs motivés d'aller au-delà de la simple expérimentation en produisant des « preuves de concept » avec une autonomie totale concernant la configuration et l'orchestration de leurs services.

Au-delà de ses capacités techniques, le SSP Cloud incarne les principes de l'innovation ouverte [43]. Déployé sur internet[16], il est accessible non seulement aux employés de l'Insee, mais égale-

[11]https://github.com/InseeFrLab/onyxia

[12]https://docs.onyxia.sh/

[13]Lien vers l'instance Onyxia de Mercator Ocean : https://datalab.dive.edito.eu/

[14]Plus d'informations à propos de ce projet disponibles à https://cros.ec.europa.eu/dashboard/aiml4os

[15]Sur le plan matériel, le SSP Cloud est constitué d'un cluster Kubernetes d'environ 20 serveurs, pour une capacité totale de 10 To de RAM, 1100 processeurs, 34 GPU et 150 To de stockage.

[16]https://datalab.sspcloud.fr/

ment, plus largement, aux agences gouvernementales françaises, aux universités françaises et aux autres INS européens. Il est dédié à l'expérimentation des méthodes de *data science* en utilisant des données ouvertes. Ainsi, les projets menés sur cette plateforme mettent en lumière l'abondance croissante des jeux de données publiés en libre accès par les organisations publiques ou privés, faisant écho à la loi pour une République numérique de 2016. La nature fondamentalement collaborative du SSP Cloud s'est avérée particulièrement bénéfique pour l'organisation d'événements innovants, tels que des hackathons — tant au niveau national qu'international — et dans le domaine académique. Il est devenu une ressource intégrale pour plusieurs universités et Grandes Écoles en France, favorisant l'utilisation d'environnements *cloud* et reproductibles, tout en évitant l'effet d'enfermement propriétaire dû à une dépendance excessive des institutions éducatives envers des solutions *cloud* propriétaires. En conséquence, la plateforme est désormais largement utilisée dans le service statistique publique français et ailleurs, avec environ 1000 utilisateurs uniques par mois début 2025. Ces utilisateurs forment une communauté dynamique grâce à un canal de discussion centralisé[17] ; ils contribuent à améliorer l'expérience utilisateur en signalant des bugs, en proposant de nouvelles fonctionnalités et en participant ainsi directement au projet.

4 Cas d'usage

5 Discussion

Le développement des méthodes de *data science* offre un potentiel considérable pour la statistique publique. Cependant, notre capacité à tirer profit de ces nouvelles méthodes dépend essentiellement de notre aptitude à produire des chaînes de production de qualité, robustes et adaptés à leurs objectifs. Cette évolution nécessite une réflexion approfondie sur ce qui constitue une infrastructure moderne et évolutive pour la *data science* dans le domaine des statistiques publiques. Cet article présente le projet Onyxia, une proposition pour une telle plateforme développée à l'Insee. En exploitant des technologies *cloud* devenues des standards dans l'écosystème de la donnée, le projet vise à accroître l'autonomie des statisticiens dans l'orchestration de leurs traitements statistiques, tout en favorisant la reproductibilité des statistiques produites. Les technologies *cloud* étant notoirement difficiles à configurer, la valeur principale d'Onyxia réside dans leur accessibilité pour les statisticiens grâce à une interface ergonomique, simple d'utilisation, et un catalogue de services préconfigurés couvrant les usages les plus courants d'un statisticien public. À travers un projet interne visant à refondre le processus de codification de l'Activité Principale de l'Entreprise (APE) en utilisant des méthodes d'apprentissage automatique, nous illustrons comment Onyxia permet de construire de manière itérative des projets de *machine learning* prêts à passer en production, favorisant l'amélioration continue, un principe fondamental de l'approche MLOps.

Initialement développé comme un projet interne, Onyxia a acquis une reconnaissance dépassant le cadre de l'Insee ou de l'administration française. Convaincues du potentiel des technologies *cloud* pour renforcer l'autonomie et exploiter pleinement le potentiel de la *data science*, plusieurs organisations disposent désormais d'une instance de production d'Onyxia (comme c'est le cas à l'Insee avec le déploiement récent de LS^3), et de nombreuses autres sont en phase de test ou

[17] Lien vers les canaux de discussion https://www.tchap.gouv.fr/#/room/#SSPCloudXDpAw6v:agent.finances.tchap.gouv.fr et https://join.slack.com/t/3innovation/shared_invite/zt-19tht9hvr-bZGMdW8AV_wvd5kz3wRSMw

d'implémentation. Par ailleurs, le choix d'Onyxia comme plateforme de *data science* de référence dans le cadre du projet AIML4OS devrait encore accroître son adoption au sein du SSE. Cette tendance est naturellement très bénéfique pour le projet Onyxia, qui passe d'un projet développé en *open source* — mais principalement à l'Insee — à un véritable projet *open source* avec une base croissante de contributeurs. Cela, en retour, facilite son adoption par d'autres organisations, en offrant davantage de garanties sur sa pérennité indépendamment de la stratégie de l'Insee. La gouvernance du projet évolue actuellement pour refléter cette tendance, notamment avec l'organisation de réunions communautaires mensuelles et la création d'un canal public et d'une feuille de route pour le projet[18].

Malgré ce succès, nous constatons plusieurs limites à l'adoption généralisée du projet au sein des organisations. Tout d'abord, il est essentiel de rappeler que le choix fondamental fait par les organisations qui adoptent Onyxia ne porte pas sur le logiciel en lui-même, mais sur les technologies sous-jacentes : la conteneurisation (via Kubernetes) et le stockage d'objets. Ces technologies peuvent représenter des coûts d'entrée important pour les organisations, puisqu'elles nécessitent un investissement dans le développement et le maintien de compétences qui ne sont pas toujours présentes dans les INS. Cependant, on constate que les organisations qui manipulent de la donnée tendent de plus en plus à s'orienter vers des solutions *cloud* qui pourrait atténuer ces défis à long terme.

De même, la transition vers les technologies *cloud* impose des coûts d'entrée pour les statisticiens. Tout d'abord, ils sont souvent confrontés à une perte de repères quant à l'endroit où les calculs sont réellement effectués : bien qu'ils soient habitués à effectuer des calculs sur des serveurs centralisés plutôt que sur un ordinateur personnel, le conteneur ajoute une couche d'abstraction qui rend cet emplacement difficile à appréhender au départ. Mais le changement le plus pertubant dans ce paradigme est la perte de persistance des données. Dans les configurations traditionnelles — qu'il s'agisse d'un ordinateur personnel ou d'un serveur accessible via un bureau virtuel — le code, les données et l'environnement de calcul sont souvent mélangés dans une sorte de boîte noire. À l'inverse, les conteneurs, par construction, n'ont pas de persistance. Si le stockage d'objets fournit cette persistance, une utilisation adéquate de ces infrastructures exige une variété d'outils et de compétences : utilisation d'un système de contrôle de version pour le code (e.g. Git), interaction avec l'API de stockage d'objets pour enregistrer les données, gestion de fichiers de configuration et/ou de secrets et variables d'environnement, etc. En un sens, ces coûts d'entrée peuvent être considérés comme le « prix » de l'autonomie : grâce aux technologies *cloud*, les statisticiens ont désormais accès à des environnements évolutifs et flexibles leur permettant d'expérimenter plus librement, mais cette autonomie exige une montée en compétences significative, qui peut être intimidante et, *in fine*, limiter cette adoption. Cependant, notre expérience à l'Insee montre que cet effet peut être largement atténué grâce à une combinaison de formation des statisticiens aux bonnes pratiques de développement et d'accompagnement des projets statistiques lors de leur transition vers des infrastructures *cloud*.

Bien qu'Onyxia ait significativement démocratisé l'accès aux technologies *cloud* pour les statisticiens, l'intégration effective des méthodes de *data science* dans la production statistique des INS

[18]Toutes les informations sont disponibles sur le dépôt GitHub du projet : https://github.com/InseeFrLab/onyxia

soulève des défis plus larges, d'ordre organisationnel. Une leçon majeure tirée du déploiement de notre premier modèle de *machine learning* en production est la nécessité de surmonter la segmentation des compétences entre les équipes informatiques, métier et innovation. Par nature, les projets de *machine learning*, pour pouvoir passer en production, impliquent un large éventail de compétences — connaissance du domaine métier, entraînement et amélioration des modèles ainsi que leur déploiement et supervision — et nécessitent donc une collaboration efficace entre des professionnels aux cultures de travail et langages de programmation variés. Notre expérience montre que les technologies *cloud*, en favorisant l'autonomie des *data scientists*, apportent plus de continuité aux projets d'apprentissage automatique et facilitent cette collaboration essentielle entre les différents profils. Toutefois, répondre pleinement à ces défis nécessite des choix qui dépassent le domaine technique. Par exemple, intégrer certaines compétence en *data science* directement au sein des équipes métier, en complément des équipes d'innovation centralisées, pourrait favoriser une meilleure collaboration. De même, recruter des profils qui ne sont pas traditionnellement présents dans les INS, comme les *data engineers* ou les *ML engineers*, pourrait apporter de nouvelles compétences à l'intersection des méthodologies statistiques et des techniques informatiques. Au final, la transition vers une approche axée sur la *data science* dans la production statistique doit s'appuyer sur une stratégie équilibrée qui lie des solutions techniques comme Onyxia avec des ajustements organisationnels et humains, favorisant une culture de collaboration, de formation continu et d'innovation.

Bibliographie

[1] F. Ricciato, A. Wirthmann, K. Giannakouris, M. Skaliotis, et others, « Trusted smart statistics: Motivations and principles », *Statistical Journal of the IAOS*, vol. 35, nº 4, p. 589-603, 2019.

[2] T. Gjaltema, « High-Level Group for the Modernisation of Official Statistics (HLG-MOS) of the United Nations Economic Commission for Europe », *Statistical Journal of the IAOS*, vol. 38, nº 3, p. 917-922, 2022.

[3] DGINS, « Bucharest Memorandum on Official Statistics in a Datafied Society ». 2018.

[4] INSEE, « Horizon 2025 ». 2016.

[5] EUROSTAT, « ESSnet Big Data 2 - Final Technical Report ». 2021.

[6] B. Sakarovitch, M.-P. d. Bellefon, P. Givord, et M. Vanhoof, « Estimating the residential population from mobile phone data, an initial exploration », *Economie et Statistique*, vol. 505, nº 1, p. 109-132, 2018.

[7] M. Leclair, I. Léonard, G. Rateau, P. Sillard, G. Varlet, et P. Vernédal, « Scanner data: advances in methodology and new challenges for computing consumer price indices », *Economie et Statistique*, vol. 509, nº 1, p. 13-29, 2019.

[8] P. Descy, V. Kvetan, A. Wirthmann, et F. Reis, « Towards a shared infrastructure for online job advertisement data », *Statistical Journal of the IAOS*, vol. 35, nº 4, p. 669-675, 2019.

[9] D. Salgado, L. Sanguiao-Sande, S. Barragán, B. Oancea, et M. Suarez-Castillo, « A proposed production framework with mobile network data », in *ESSnet Big Data II - Workpackage I - Mobile Network Data*, 2020.

[10] A. Kowarik et M. Six, « Quality Guidelines for the Acquisition and Usage of Big Data with additional Insights on Web Data », in *4th International Conference on Advanced Research Methods and Analytics (CARMA 2022)*, 2022, p. 269-270.

[11] F. Ricciato, F. De Meersman, A. Wirthmann, G. Seynaeve, et M. Skaliotis, « Processing of mobile network operator data for official statistics: the case for public-private partnerships », in *104th DGINS conference*, 2018.

[12] A. Ashofteh et J. M. Bravo, « Data science training for official statistics: A new scientific paradigm of information and knowledge development in national statistical systems », *Statistical Journal of the IAOS*, vol. 37, nº 3, p. 771-789, 2021.

[13] T. H. Davenport et D. Patil, « Data scientist », *Harvard business review*, vol. 90, nº 5, p. 70-76, 2012.

[14] L. Liu, « Computing infrastructure for big data processing », *Frontiers of Computer Science*, vol. 7, p. 165-170, 2013.

[15] A. Saiyeda et M. A. Mir, « Cloud computing for deep learning analytics: A survey of current trends and challenges. », *International Journal of Advanced Research in Computer Science*, vol. 8, nº 2, 2017.

[16] S. Ghemawat, H. Gobioff, et S.-T. Leung, « The Google file system », in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003, p. 29-43.

[17] A. I. Abdelaziz, K. A. Hanson, C. E. Gaber, et T. A. Lee, « Optimizing large real-world data analysis with parquet files in R: A step-by-step tutorial », *Pharmacoepidemiology and Drug Safety*, 2023.

[18] J. Dean et S. Ghemawat, « MapReduce: simplified data processing on large clusters », *Communications of the ACM*, vol. 51, nº 1, p. 107-113, 2008.

[19] B. Dhyani et A. Barthwal, « Big data analytics using Hadoop », *International Journal of Computer Applications*, vol. 108, nº 12, p. 975-8887, 2014.

[20] J. Tigani, « Big data is dead ». [En ligne]. Disponible sur: https://motherduck.com/blog/big-data-is-dead/

[21] M. Leclair et others, « Using Scanner Data to Calculate the Consumer Price Index », *Courrier des statistiques*, vol. 3, p. 61-75, 2019.

[22] S. Vale, « International collaboration to understand the relevance of Big Data for official statistics », *Statistical Journal of the IAOS*, vol. 31, nº 2, p. 159-163, 2015.

[23] A. S. Foundation, « Apache Parquet ». 2013.

[24] D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden, et others, « The design and implementation of modern column-oriented database systems », *Foundations and Trends® in Databases*, vol. 5, nº 3, p. 197-280, 2013.

[25] A. S. Foundation, « Apache Arrow ». 2016.

[26] M. Raasveldt et H. Mühleisen, « Duckdb: an embeddable analytical database », in *Proceedings of the 2019 International Conference on Management of Data*, 2019, p. 1981-1984.

[27] Y. Li *et al.*, « Big data and cloud computing », *Manual of digital earth*, p. 325-355, 2020.

[28] O. Bentaleb, A. S. Belloum, A. Sebaa, et A. El-Maouhab, « Containerization technologies: Taxonomies, applications and challenges », *The Journal of Supercomputing*, vol. 78, nº 1, p. 1144-1181, 2022.

[29] R. Vaño, I. Lacalle, P. Sowiński, R. S-Julián, et C. E. Palau, « Cloud-native workload orchestration at the edge: A deployment review and future directions », *Sensors*, vol. 23, nº 4, p. 2215-2216, 2023.

[30] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu, et W. Zhou, « A comparative study of containers and virtual machines in big data environment », in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018, p. 178-185.

[31] S. Samundiswary et N. M. Dongre, « Object storage architecture in cloud for unstructured data », in *2017 International Conference on Inventive Systems and Control (ICISC)*, 2017, p. 1-6.

[32] M. Mesnier, G. R. Ganger, et E. Riedel, « Object-based storage », *IEEE Communications Magazine*, vol. 41, nº 8, p. 84-90, 2003.

[33] L. Leite, C. Rocha, F. Kon, D. Milojicic, et P. Meirelles, « A survey of DevOps concepts and challenges », *ACM Computing Surveys (CSUR)*, vol. 52, nº 6, p. 1-35, 2019.

[34] M. McNutt, « Reproducibility », *Science*, vol. 343, nº 6168, p. 229-230, 2014.

[35] European Commission, « European Statistics Code of Practice — revised edition 2017 ».

[36] S. Luhmann, J. Grazzini, F. Ricciato, M. Mészáros, J.-M. Museux, et M. Hahn, « Promoting reproducibility-by-design in statistical offices », in *2019 New Techniques and Technologies for Statistics (NTTS) conference*, 2019, p. . doi: 10.5281/zenodo.3240198.

[37] D. Moreau, K. Wiebels, et C. Boettiger, « Containers for computational reproducibility », *Nature Reviews Methods Primers*, vol. 3, nº 1, p. 50-51, 2023.

[38] S. Gokhale *et al.*, « Creating helm charts to ease deployment of enterprise application and its related services in kubernetes », in *2021 international conference on computing, communication and green engineering (CCGE)*, 2021, p. 1-5.

[39] J. Opara-Martins, R. Sahandi, et F. Tian, « Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective », *Journal of Cloud Computing*, vol. 5, p. 1-18, 2016.

[40] J. Opara-Martins, M. Sahandi, et F. Tian, « A holistic decision framework to avoid vendor lock-in for cloud saas migration », *Computer and Information Science*, vol. 10, nº 3, 2017.

[41] C. M. Schweik, « Free/open-source software as a framework for establishing commons in science », 2006.

[42] F. Comte, A. Degorre, et R. Lesur, « SSPCloud: a creative factory to support experimentations in the field of official statistics », *Courrier des Statistiques, INSEE*, vol. 7, p. 68-85, 2022.

[43] H. W. Chesbrough, *Open innovation: The new imperative for creating and profiting from technology*. Harvard Business Press, 2003.