# Comment transformer tous nos gestionnaires de parc en DeVops

Rémy Esberard

# Comment transformer tous nos gestionnaires de parc en DeVops

**Rémy ESBERARD**
Direction du Numérique de l'Université de Strasbourg
14 rue René Descartes
67000 STRASBOURG

## Résumé

*Le parc informatique de l'Université de Strasbourg géré avec la solution de la direction du numérique, c'est aujourd'hui 8000 postes. Le logiciel utilisé depuis 12 ans (Altiris), essentiellement pour le parc Windows, a subi ces dernières années une augmentation de tarif et une réduction des fonctionnalités nous obligeant à étudier un nouvel outil moins onéreux et répondant à l'évolution de nos besoins.*

*En 2022, le département Ingénierie et Informatique de Proximité de la Direction du Numérique de l'Université de Strasbourg a étudié, en collaboration avec des informaticiens de composantes, un outil capable de déployer des systèmes Windows, et d'installer des applications à distance tout en réduisant les coûts. Après une phase d'étude, notre choix s'est porté sur l'application WAPT.*

*Le projet WAPT a démarré en septembre 2023. Le système de déploiement historiquement basé sur des images a évolué vers du déploiement d'iso Microsoft avec WADS. Nous gérons les droits d'accès à la console WAPT par l'intermédiaire de comptes AD spécifiques. Pour disposer de restrictions d'accès par périmètres, nous avons mis en place une arborescence de certificats spécifique. Un système de renouvellement des certificats via des scripts Ansible est disponible sur une interface web AWX.*

*Concernant le packaging logiciel, nous avons utilisé la chaîne d'intégration continue basée sur GITLAB. Ces dépôts permettent de suivre nos versions applicatives et nos personnalisations de paquets dans WAPT.*

*La spécificité du projet était de faire évoluer les méthodes de travail, d'éviter au maximum de reproduire les mauvaises pratiques de l'ancien outil, tout en réunissant un maximum de gestionnaires de parc informatique de l'Université de Strasbourg dans l'optique de les transformer en DeVops.*

## Mots-clefs

*WAPT, WADS, GitLab, AWX, Ansible*

# 1 Étude pour le remplacement de l’outil de gestion de parc (Altiris)

L’outil de gestion de parc (Altiris) installé à l’université depuis plus de 12 ans, nécessitait d’être remplacé pour les raisons suivantes :

- il perdait des fonctionnalités au fur et à mesure des années ;
- il devenait difficile à maintenir (de plus en plus de bugs ou d’indisponibilité de la solution après application des mises à jour) ;
- le coût des licences devenait de moins en moins soutenable suite au rachat de la solution par Broadcom.

Il devenait nécessaire de réfléchir à une nouvelle solution avec l’objectif de migrer avant le dernier renouvellement de contrat.

La Direction du Numérique (DNum) a démarré une étude pour choisir le nouveau logiciel de gestion de parc au début de l’été 2022. Cette étude a intégré différents gestionnaires de parc du département Ingénierie et Informatique de Proximité (2IP) de la DNum, ainsi que certains informaticiens volontaires de composantes.

Il faut noter que la DNum gère environ 50 % du parc informatique de l’Université de Strasbourg, le reste étant géré par des informaticiens directement rattachés à des composantes ou laboratoires sans aucun lien fonctionnel ou hiérarchique avec la DNum.

L’objectif de cette étude était de recenser les solutions existantes, d'évaluer l’avenir de chacune d’entre elles, de définir des critères de choix et d’établir le planning ainsi que la mise en place du nouvel outil si un changement devait avoir lieu.

**Concernant les usages existants à l’université** :

Nous disposons de la solution Altiris qui est en fonction depuis environ 10 ans au moment de l’étude, nous avons également une plateforme VDI (Omnissa ex VMware) qui est en test depuis bientôt 2 ans, mais celle-ci n’a convaincu qu’une seule composante de l’Université par son fonctionnement et les coûts étaient trop importants par rapport au budget prévu initialement. Pour la gestion des périphériques Apple, nous utilisons JAMF qui en production depuis 2019. Enfin, nous mettons à disposition des personnels un serveur RDS qui leur permet d’accéder à un ensemble d’applications et à leur environnement en bureau à distance, celui-ci est en service depuis 2011.

**Les questions que devait se poser le groupe de travail** :

Devions-nous abandonner l’ensemble des solutions dans le but de trouver un outil capable de gérer tous les systèmes ? Accepter de devenir dépendant d’un seul outil ? Ou alors, continuer de disposer d’un outil par système ?

**Les critères de choix que nous avons retenu** :

- cartographier le parc et faire un classement par type de machine et d’usage ;
- proposer un ou des outils en fonction des résultats du premier critère ;
- rationaliser nos outils et nos services ;
- gagner du temps dans l’exploitation (afin de réaliser plus de missions d’ingénierie) ;
- réduire nos coûts de licences et d’infrastructure ;

- interconnecter nos outils de gestion au travers d'API ;
- déployer et installer des images et des logiciels ;
- gérer les logiciels et les licences ;
- gérer des politiques de mises à jour des logiciels et des systèmes d'exploitation pour homogénéiser notre parc informatique.

**Planning initial de l'étude** :


Nous avons ainsi mis en place durant cette étude, des maquettes des différentes solutions que nous avons retenu qui ont permis de réaliser différents tests en fonction des critères établis et ainsi de noter chaque solution.

Concernant la sélection des outils à tester, notre direction imposait de ne pas dépasser 60K€/an, cela a permis de réduire le choix sur ces 3 outils que nous avons comparés à l'actuel (Altiris) :

Synthèse technique

| | | | | SCCM | | Matrix42 | | WAPT | | Altiris | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fonctionnalités | Coefficient | Note Max | Pourcentage du critère | Note du critère | Note sur 20 | Note du critère | Note sur 20 | Note du critère | Note sur 20 | Note du critère | Note sur 20 |
| Télédistribution et mises à jour de logiciels et mise à jour | 54 | 540 | 28 % | 392 | 14,52 | 389 | 14,41 | 404 | 14,96 | 407 | 15,07 |
| Déploiement d'OS, d'imaging et outils de migration d'OS | 29 | 290 | 15 % | 250 | 17,24 | 251 | 17,31 | 241 | 16,62 | 245 | 16,90 |
| Fonctionnalités complémentaires | 20 | 200 | 10 % | 140 | 14,00 | 195 | 19,50 | 192 | 19,20 | 162 | 16,20 |
| Infrastructure (serveurs) | 36 | 360 | 19 % | 268 | 14,89 | 268 | 14,89 | 340 | 18,89 | 155 | 8,61 |
| Console (accès/paramétrage) | 28 | 280 | 15 % | 200 | 14,29 | 150 | 10,71 | 205 | 14,64 | 200 | 14,29 |
| Documentation/Aide | 9 | 90 | 5 % | 66 | 14,67 | 63 | 14,00 | 90 | 20,00 | 54 | 12,00 |
| Portail d'applications et virtualisation | 8 | 80 | 4 % | 40 | 10,00 | 40 | 10,00 | 40 | 10,00 | 40 | 10,00 |
| Tableaux de bord / Suivi / Reporting / Alertes | 9 | 90 | 5 % | 75 | 16,67 | 75 | 16,67 | 52 | 11,56 | 60 | 13,33 |
| Total valeur technique | 193 | 1930 | 100 % | 1431 | 14,83 | 1431 | 14,83 | 1564 | 16,21 | 1323 | 13,71 |

Suite à cette évaluation, notre choix s'est tourné vers la solution WAPT de l'éditeur Tranquil-IT.

# 2 Déploiement de l'infrastructure WAPT à l'Unistra

## 2.1 Architecture retenue

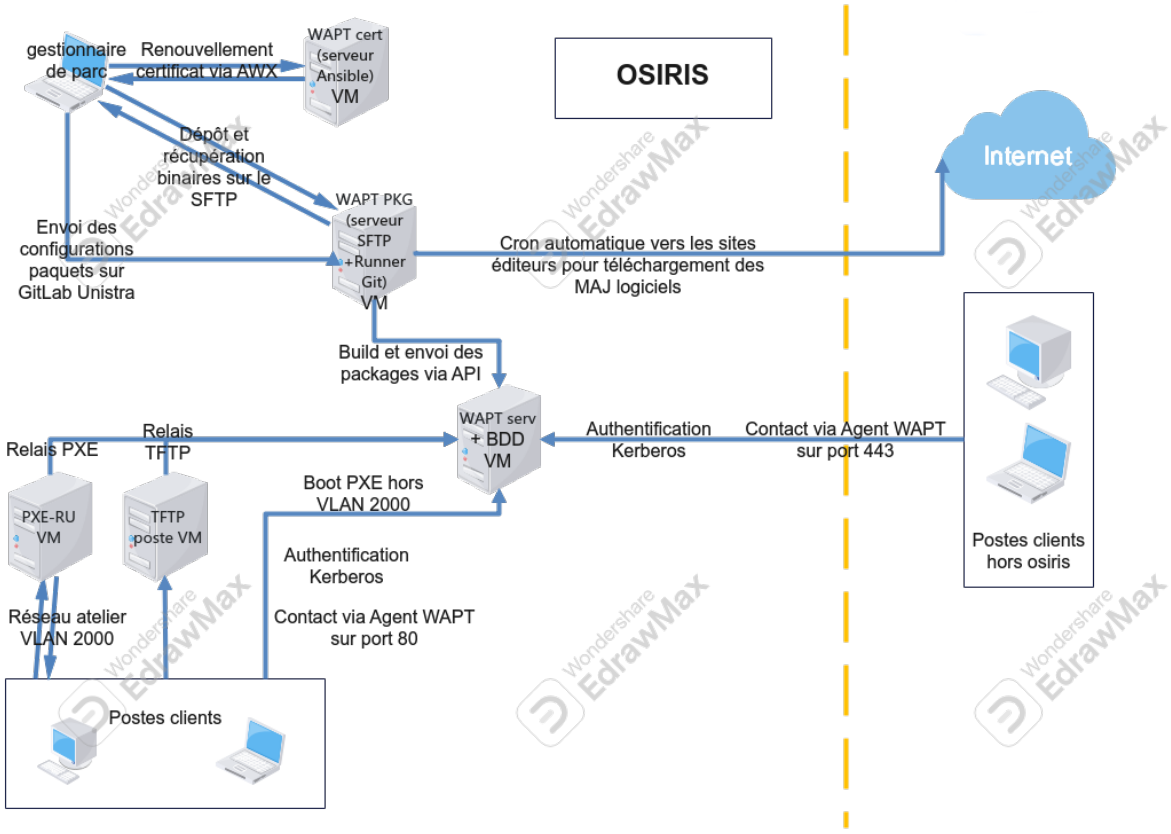


*Figure 1: Schéma architecture WAPT à l'UNISTRA*

Nous avons mené une analyse des différentes architectures possibles avec WAPT, en collaboration avec l'éditeur Tranquil-IT[1] et notre département Infrastructure qui s'occupe de tous les aspects serveurs et réseau. Nous avons finalement choisi de déployer une plateforme avec un seul serveur WAPT et d'étudier si besoin, le système de dépôts secondaire. Les serveurs nommés « WAPT cert » et « WAPT PKG » ne sont pas des serveurs WAPT mais des VM hébergeant des services liés à notre utilisation de la solution.

## 2.2 Gestion des accès et droits dans WAPT

Le parc informatique de l'université de Strasbourg est géré par plus de 50 gestionnaires de parc, répartis entre la Direction du Numérique et les différents informaticiens rattachés à certaines composantes.

Cette gestion est cadrée par la mise en place de différents périmètres d'intervention. Nous disposions dans notre précédent outil d'un système de droits qui permettait à chaque gestionnaire d'une zone, de voir et d'administrer uniquement les machines de son périmètre.

Le cloisonnement était un impératif et nous devions l'implémenter dans WAPT.

1 : https://www.tranquil.it/

Nous avons utilisé les groupes présents dans notre Active Directory pour gérer l'accès à la console d'administration de WAPT. Chaque technicien dispose de la console installée sur son poste, et une autre console a été installée sur un bastion en cas de besoin. Les droits d'accès sont associés aux comptes utilisateurs, qui sont récupérés directement depuis l'AD, et permettent de se connecter à la console.

Les comptes gestionnaires disposent de restrictions horaires d'utilisation mis en place au niveau de l'Active Directory, cela permet de sécuriser l'accès à WAPT en dehors des horaires de bureau.

Nous avons ensuite défini les droits d'utilisation que nous avons accordés aux différents gestionnaires de parc par l'intermédiaire du système d'ACL propre à WAPT.

La mise en place des certificats de zones par le biais de « paquets certificats » sur les différents sous-domaines et OU de l'Active Directory a permis de cloisonner la gestion des différents parcs informatiques de l'Université. Les « certificats de zone » sont générés par les administrateurs via la console d'administration de WAPT, mais ne sont pas attribués directement aux gestionnaires. Ils sont stockés sur un serveur et utilisés par des scripts Ansible pour générer des certificats enfants destinés aux gestionnaires. Chaque certificat de zone agit comme une Autorité de Certification (CA), lui permettant de signer de nouveaux certificats qui hériteront de ses droits. Le certificat de zone est ensuite appliqué aux machines de la zone correspondante. Les gestionnaires, dotés d'un certificat enfant issu de celui de la zone, pourront ainsi visualiser et déployer des paquets uniquement sur les machines de cette zone, grâce à l'héritage entre le certificat parent et les certificats enfants.

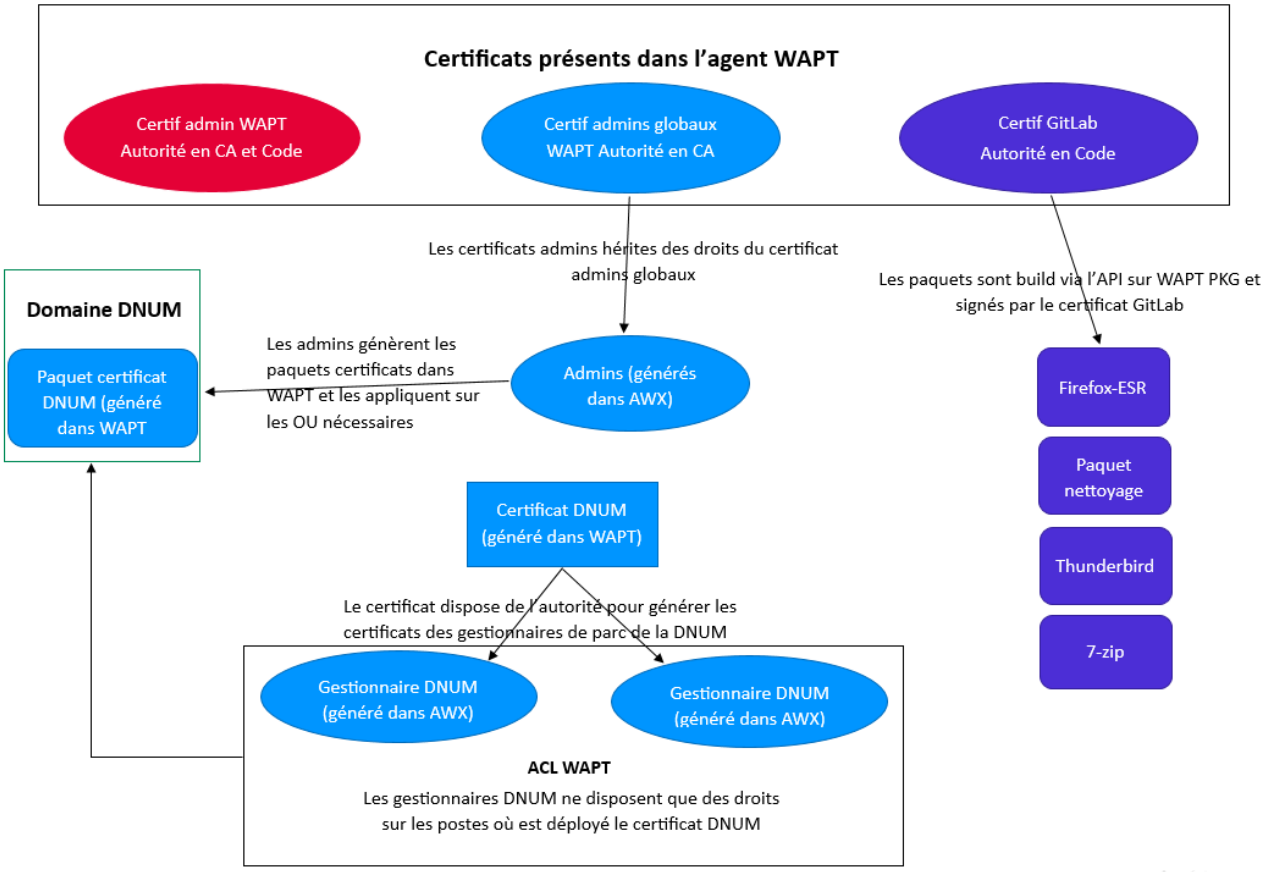


*Figure 2: Schéma simplifié de l'arborescence de certificats*

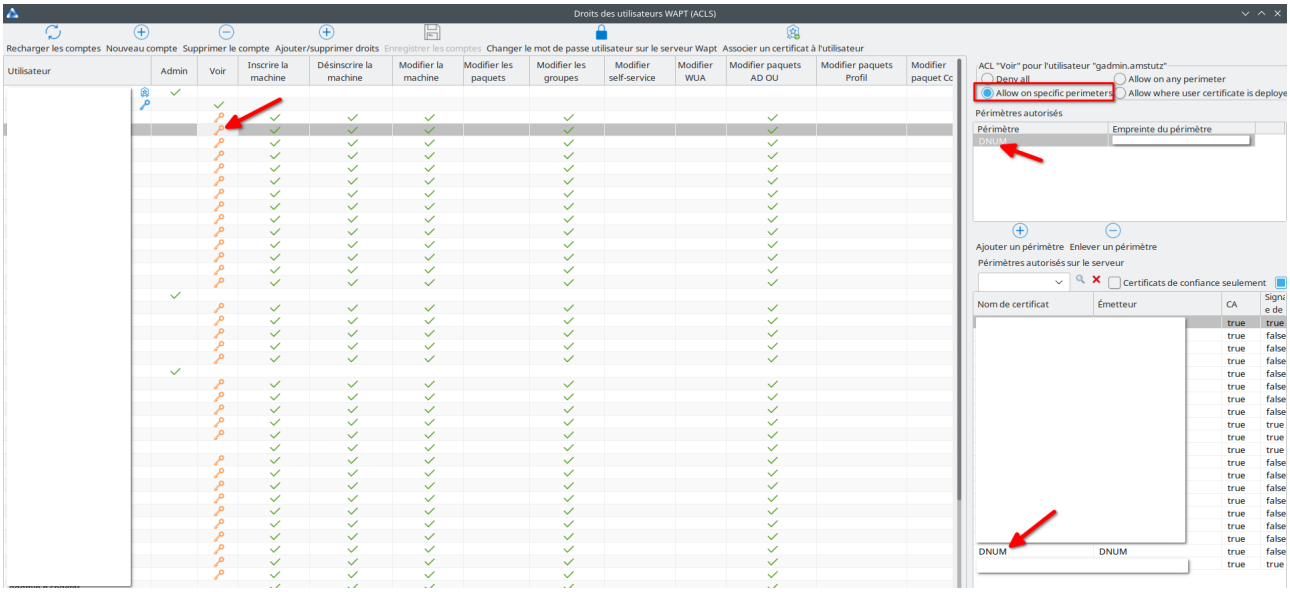


*Figure 3: Fenêtre des ACL WAPT*

Les certificats de zones sont générés au travers de WAPT directement avec une durée de vie de 10 ans (comme le certificat admin qui est généré à l'installation de la solution).

Nous avons renforcé la sécurité au niveau des certificats attribués aux gestionnaires de parc en limitant la durée de validité à six mois. Le renouvellement des certificats des gestionnaires, se fait au travers de la plateforme AWX[2], qui est un système open source fournit par Red Hat, qui met à disposition une interface web permettant d'exécuter des playbooks (scripts Ansible).

Cette interface est accessible aux gestionnaires de parc informatique depuis un réseau sécurisé et leur permet de renouveler leur certificat en totale autonomie.

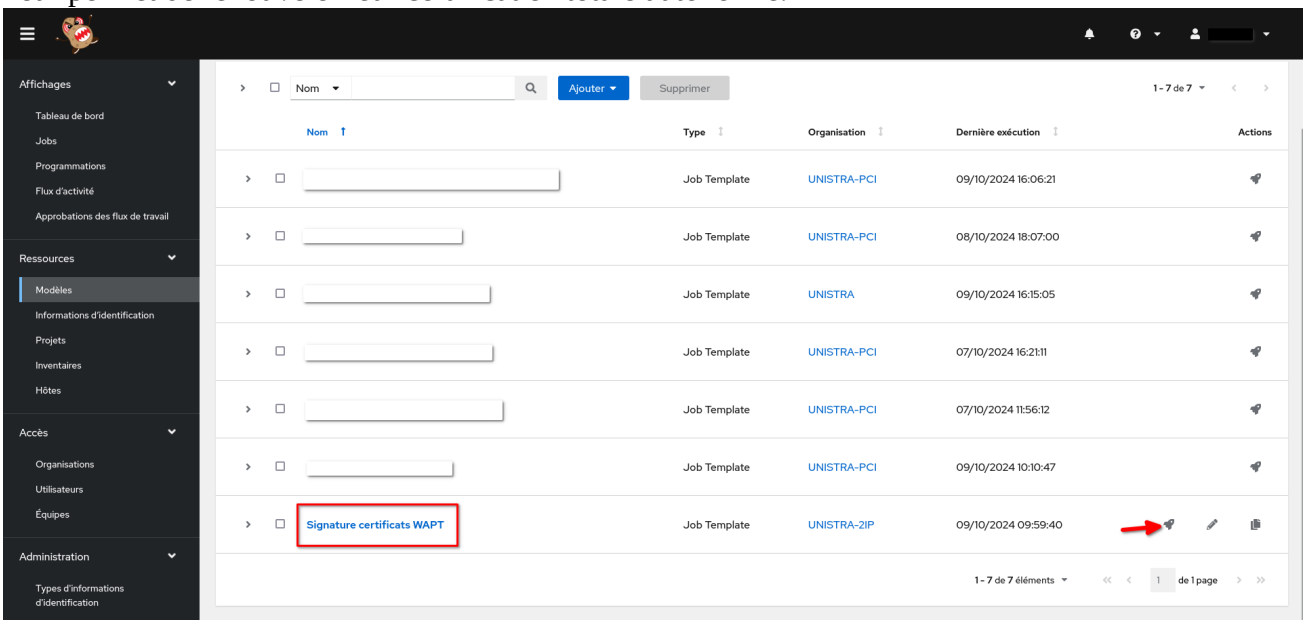


*Figure 4: Interface WEB de AWX*

Le système AWX utilise des certificats de zones (certificats racines) qui ont été préalablement générés dans WAPT et déployés sur les différents domaines ou unités organisationnelles (OU).

2 : https://github.com/ansible/awx

Ensuite, un script Ansible est utilisé pour se baser sur une liste de gestionnaires, que nous avons renseignée manuellement, afin de définir un périmètre spécifique pour chaque gestionnaire. Lorsqu'un gestionnaire se connecte à AWX avec son compte, cela permet de l'identifier et ainsi, lors de l'exécution du playbook, le périmètre auquel il appartient est déterminé, et un certificat lui est généré. Celui-ci est signé par le certificat de la zone qui lui est attribuée.

### 2.3 Déploiement de Windows

Historiquement, nous maintenions une image Ghost qui était mise à jour tous les mois. Cette méthode monopolisait une personne pendant au moins une journée tous les mois pour procéder aux mises à jour de l'image, appliquer des correctifs et téléverser la nouvelle image sur le serveur. À chaque nouvelle image, nous procédions à un ensemble de tests. Cette opération était chronophage et source d'erreurs (image mise à jour par l'intermédiaire de machine virtuelle).

Avec WAPT, nous utilisons le système de déploiement d'OS inclus dans l'application qui se nomme WADS (*WAPT Automated Deployment Services*). Cette nouvelle méthode, entièrement automatisée, nous évite les étapes précédentes listées ci-dessous et nous fait gagner un temps certain.

Celui-ci charge l'image ISO de Windows 10/11 que nous récupérons au préalable sur les dépôts officiels de Microsoft, un fichier de configuration au format xml et un fichier de configuration de post-installation. Au moment de la préparation à l'installation de l'OS, il est possible de choisir une bibliothèque de pilotes, alimentée par les administrateurs de WAPT à partir des sources récupérées sur les sites des fabricants. Grâce à un système de filtres basés sur le nom du fabricant et le modèle du poste, WAPT peut automatiquement sélectionner la bibliothèque de pilotes appropriée.

Cette nouvelle méthode permet dans des temps très réduits, de déployer le système ainsi que les pilotes et simplifie fortement la maintenance.

## 3 Gestion du packaging logiciel dans WAPT

### 3.1 Gestion des droits liés au packaging dans WAPT

Pour reprendre l'historique du fonctionnement avec notre ancien outil, chaque gestionnaire de parc pouvait créer des paquets logiciels. Mais par manque de centralisation et de contrôle, cela a généré des doublons logiciels, mais également des disparités d'états entre les parcs informatiques des différentes zones.

Nous avons fait évoluer ce fonctionnement pour homogénéiser et standardiser nos périmètres d'exploitation. L'idée était de mutualiser autant que possible les paquets logiciels et les paquets configurations entre les différentes zones.

Cependant, nous étions confrontés à une contrainte en essayant de gérer ce processus uniquement avec le système des ACL de WAPT, car il ne permettait pas de gérer les permissions de manière aussi précise que nous le souhaitions. C'est pourquoi nous avons choisi d'utiliser le système d'intégration continue de GitLab[3], qui offre une gestion des droits plus avancée ainsi qu'un système de versionnement pour nos paquets au fil du temps.

3 : https://about.gitlab.com/fr-fr/

Le fonctionnement de WAPT pour l'installation de logiciels en volume consiste à associer des paquets logiciels directement aux unités organisationnelles (OU) de l'Active Directory (AD). Toutes les machines présentes dans ces OU installent automatiquement le ou les logiciels assignés.

Grâce à ce fonctionnement, tous les postes reçoivent en quasi temps réel, les mêmes logiciels dans les mêmes versions.

## 3.2 Mise en place d’une chaîne d’intégration continue via GitLab

C'est à ce moment que GitLab intervient pour résoudre le problème de mutualisation des paquets logiciels. Si tous les gestionnaires avaient la possibilité de modifier les paquets dans WAPT, ils pourraient accidentellement introduire une erreur dans le code et mettre à jour un logiciel, ce qui risquerait de provoquer des dysfonctionnements sur l'ensemble des machines où ce logiciel est déployé.

Nous avons donc détourné la partie création et modification de paquets logiciels dans WAPT en n’accordant pas les droits de « Modifier un paquet » dans les ACL. Nous passons par la chaîne d’intégration continue GitLab.

GitLab nous permet ainsi de versionner nos paquets en fonction des différentes évolutions, de disposer d’un suivi complet de chaque modification d’un paquet et d’autoriser seulement certaines personnes à modifier uniquement certains paquets. Tous les gestionnaires ont la possibilité de proposer des mises à jour ou des correctifs sur des paquets en créant de nouvelles branches dans le projet du paquet dans GitLab. Une fois les modifications terminées, ils soumettent une "Merge Request" (ou Requête de Fusion) qui informe les mainteneurs du paquet des changements proposés. Ces derniers examinent les modifications et peuvent les approuver, ce qui déclenche l'intégration des changements dans la branche principale et génère automatiquement le nouveau paquet dans WAPT.

Cependant, nous n'utilisons pas GitLab pour stocker directement les fichiers binaires, car ce n'est pas sa fonction principale. À la place, nous avons déployé un serveur SFTP pour héberger tous les binaires. GitLab y accède lors de la construction des paquets afin de récupérer les binaires nécessaires et de les intégrer dans les paquets.

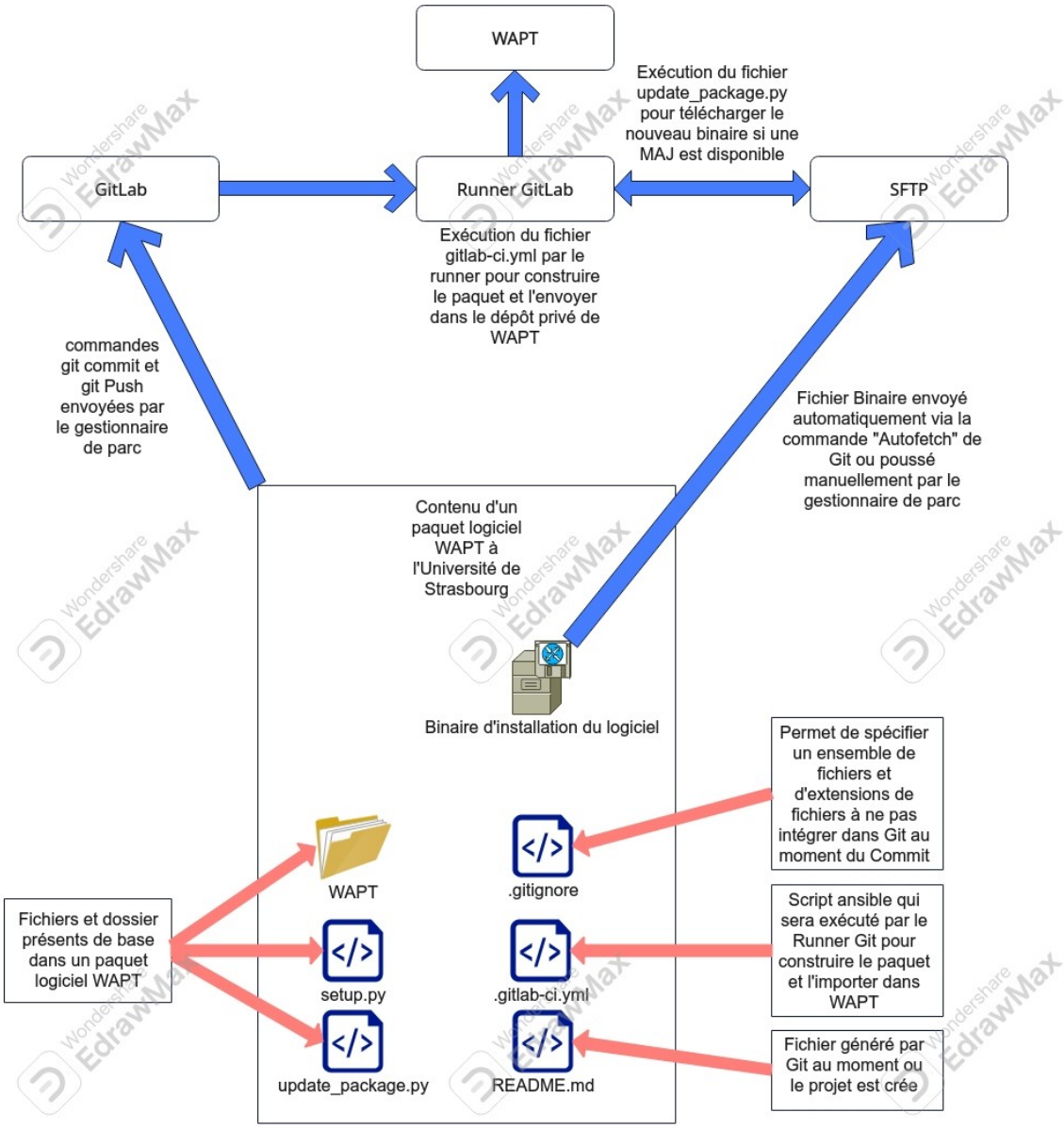


*Figure 5: Schéma du fonctionnement de la construction d'un paquet via GitLab*

Pour que GitLab puisse déposer les paquets logiciels dans le dépôt privé, nous avons généré un certificat spécifique à GitLab l'autorisant à signer le paquet et avons également créé un compte Gitlab dans WAPT avec des ACL permettant uniquement de modifier des paquets.

Nous avons également mis en place une restriction sur le parc par rapport aux logiciels sous licence par l'intermédiaire de GitLab. Tous les gestionnaires sont en mesure de voir les paquets logiciels dans WAPT et de les appliquer à un ou plusieurs postes. Nous voulions éviter qu'un gestionnaire puisse déployer un logiciel sous licence sur des postes pour lesquels celui-ci n'était pas destiné.

Pour cela, nous utilisons des certificats spécifiques de zone disposant de l'autorité pour signer du code qui sont différents des certificats de zone avec l'autorité en CA. Nous avons ensuite déployé ces certificats sur leurs zones respectives pour que ceux-ci soient installés sur les postes.

Puis nous avons intégré ces certificats dans GitLab afin qu'un gestionnaire d'une zone puisse créer un paquet logiciel sous licence spécifique à sa zone. Le logiciel sera automatiquement tagué avec le préfixe correspondant à l'acronyme de la zone du gestionnaire. Par exemple, un paquet signé avec le certificat GitLab principal portera le préfixe « unistra- », tandis qu'un paquet signé avec un certificat GitLab propre à une composante aura le préfixe « composante- ». Cela permet à GitLab de signer le paquet avec le certificat spécifique à la zone, garantissant que le logiciel ne pourra être installé que dans la zone du gestionnaire concerné.

Grâce à cette gestion au travers de GitLab, nous avons également mis en place un système de vérification automatique de mises à jour pour chaque logiciel. Le système s'appuie sur le fichier « update_package.py » présent dans chaque paquet, qui permet de vérifier directement sur le site de l'éditeur si une nouvelle version du logiciel est disponible et de la télécharger. Une tâche automatisée, exécutée via le Runner GitLab chaque nuit, parcourt tous les paquets et vérifie si des mises à jour sont disponibles. Ensuite, le runner envoie un email à une liste de gestionnaires pour chaque logiciel dont une mise à jour a été téléchargée sur le SFTP, permettant ainsi aux gestionnaires d'être informés des nouvelles versions à intégrer.

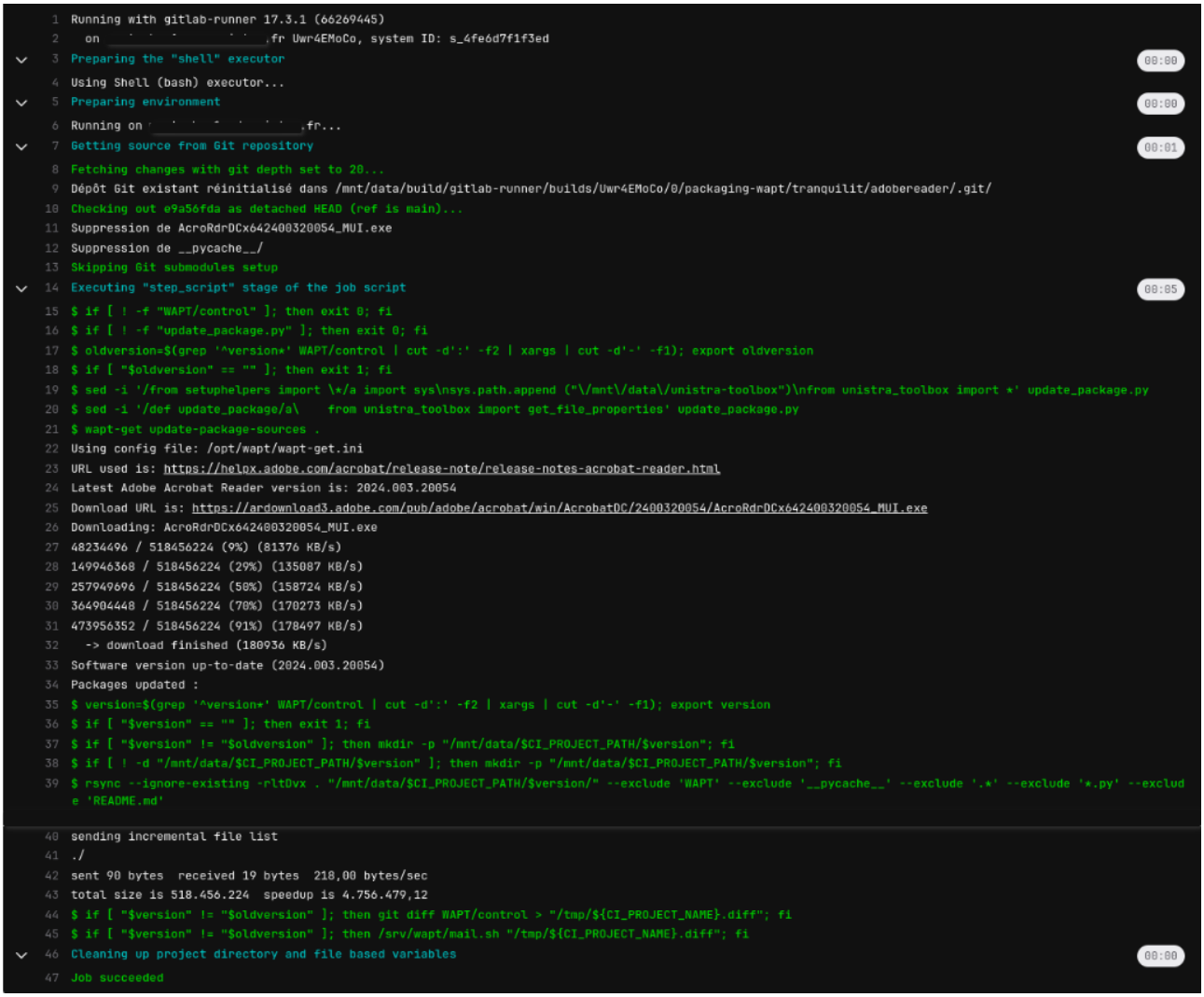


*Figure 6: Déroulement de l'update_package.py par l'autofetch de GitLab*

Quand un gestionnaire décide de mettre à jour le logiciel (en cas de faille de sécurité ou de montée en version majeure), il aura juste à se rendre dans le dépôt GitLab du logiciel, de modifier la version du logiciel avec le numéro de la nouvelle version et faire le commit qui aura pour effet de construire le paquet. Le paquet sera alors généré dans le dépôt privé de WAPT avec la nouvelle version en maturité « PREPROD ». Il est alors possible de déployer le logiciel sur un pool de machines de test pour vérifier que tout fonctionne normalement avant le passage en « PROD ». Le système de maturité dans WAPT permet de contrôler l'installation des paquets sur certaines machines et non sur d'autres, en ajustant les niveaux de maturité des paquets que différents agents WAPT sont autorisés à installer. Cela permet de définir quels agents peuvent accepter des paquets en fonction de leur stade de validation.

Avec ce nouveau mécanisme de déploiement logiciel, nous avons transformé nos gestionnaires de parc informatiques en DevOps. Ils ont été formés en interne sur le fonctionnement du processus, avec la mise à disposition de procédures détaillées étape par étape ainsi que des enregistrements vidéo illustrant le processus de création d'un paquet. Bien qu'ils n'aient pas suivi de formation en Python, ils ont accès à la documentation des setuphelpers de WAPT.

# 4 Migration de notre ancien outil vers WAPT

Une fois l’infrastructure WAPT mise en place, nous avons déployé l’agent WAPT sur l’ensemble des postes de notre périmètre par l’intermédiaire de notre ancien outil. Cela nous a permis de cibler la quasi-totalité des machines, même si celles-ci n’étaient pas intégrées au domaine AD de l’Université. Nous avons profité des grandes vacances d’été, qui est une période calme et propice à la mise à jour des salles de ressources (fixes et classes mobiles de pc portables) pour vérifier que celle-ci avait bien installé l’agent WAPT.

## 4.1 Déploiement des paquets logiciels et nettoyages

Cette migration a été l’occasion pour nous de remettre à plat la configuration standard d’un poste pour les personnels (administratif ou enseignant chercheur). On installe par défaut sur un poste standard un certain nombre de logiciels prédéfinis (Antivirus, Pack Microsoft Office, LibreOffice, 7-zip, etc.).

Une fois ce standard défini, nous avons créé les paquets pour l’ensemble des logiciels retenus. Après différentes étapes de tests, l’étape la plus critique consistait à déployer ces nouveaux standards sur l’ensemble du parc. Jusqu'à présent, nous n'avions pas de suivi strict, seules certaines applications ou configurations étaient régulièrement mises à jour, car ces opérations étaient très chronophages sur Altiris. Avec WAPT, qui uniformise la gestion de tout le parc, il a été nécessaire de migrer les postes (accusant pour certains plusieurs années de retard), tout en minimisant l'impact sur les utilisateurs.

Pour ce faire, nous avons déployé notre configuration par étapes, en définissant deux zones de test distinctes en amont afin de valider le déploiement. De plus, nous avons segmenté le déploiement en quatre phases afin de limiter l’impact sur les applications. Ce processus de déploiement s’est étalé sur une période de deux mois.

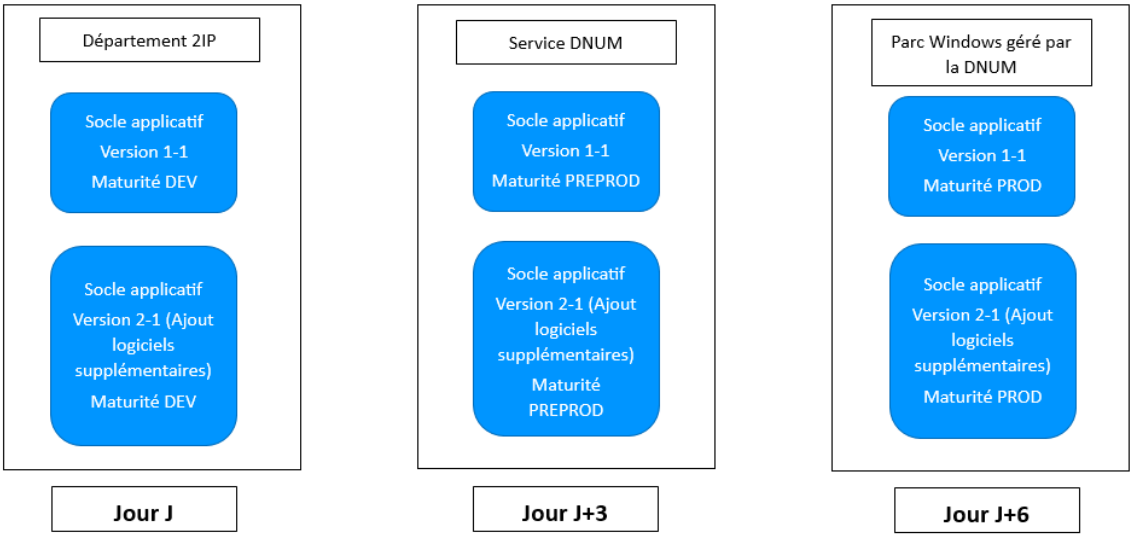


*Figure 7: Schéma de l'évolution d'un méta-paquet avec ces maturités*

Il a été nécessaire de créer des paquets de « configurations dynamiques de l'agent » permettant de modifier les valeurs dans le fichier de configuration de l'agent WAPT (wapt-get.ini). Dans notre cas, nous avons généré un paquet « agent-DEV » qui permet à l'agent WAPT d'accepter l'installation de paquets en maturités DEV, PREPROD et PROD. Nous avons également créé un paquet « agent-PREPROD » qui reconfigure l'agent pour qu'il n'accepte que les maturités PREPROD et PROD, ce qui permet de modifier les niveaux de maturités des paquets acceptés par les agents dans les différents périmètres où sont appliqués ces configurations.

Nous avons acté que le département 2IP (Ingénierie et Informatique de Proximité) serait le périmètre de « DEV ». L'ensemble de la Direction du Numérique ainsi qu'un service central de l'Université avec des utilisateurs avertis représentent le périmètre de « PREPROD ». Le reste des services et composantes de l'Université que nous gérons correspondent au périmètre de « PROD ».

Pour éviter durant ces phases de déploiement, de gérer le niveau de maturité de l'ensemble des paquets logiciels, nous avons décidé de passer par des « méta-paquets ». Un « méta-paquet » est un paquet dans lequel nous avons mis en dépendance un certain nombre d'autres paquets.

Nous avons juste eu à gérer les changements de maturité sur les quelques méta-paquets que nous avons sur nos unités organisationnelles pour les déployer au fur et à mesure sur le reste de notre parc.

Au début du déploiement, nous avons incorporé quelques logiciels qui ont suivi les évolutions de maturités de leur méta-paquet, puis nous avons ajouté au fur et à mesure des logiciels aux méta-paquets. Pour gérer ces ajouts de logiciels, nous avons utilisé le système de version de paquet que l'on incrémente entre les maturités. Pour cela, nous créons un paquet décliné dans les trois maturités possibles (DEV, PREPROD, PROD). Nous commençons par ajouter une liste de logiciels dans le paquet DEV. Ensuite, lorsque nous ajoutons un nouveau logiciel, nous incrémentons uniquement la version du paquet DEV, laissant celles de PREPROD et PROD inchangées. Ainsi, les postes en DEV installent un logiciel supplémentaire que les postes en PREPROD et PROD n'ont pas encore. Plus tard, nous soumettons une *merge request* dans GitLab pour copier le paquet DEV dans la branche « main », ce qui génère le paquet PREPROD et fait évoluer sa version pour correspondre à celle de DEV. À ce stade, DEV et PREPROD contiennent les mêmes logiciels, avec un de plus que la PROD. Ensuite, nous faisons une étiquette (ou tag) de la version de PREPROD dans GitLab, ce qui génère le paquet PROD incluant le logiciel ajouté. Ce processus permet de gérer l'ajout

progressif de logiciels tout en synchronisant les différents périmètres durant la période de déploiement de nos logiciels standards.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| base | OPOST | unistra-opost | 8-1 | window | x64 | | PROD | Meta paquet pour les logiciels OPOST | 06/09/2024 09:17:10 | gitlab-pkgbuild-20231218 | 8.1 KB |
| base | OPOST | unistra-opost | 8-1 | window | x64 | | PREPROD | Meta paquet pour les logiciels OPOST | 06/09/2024 09:16:20 | gitlab-pkgbuild-20231218 | 8.1 KB |
| base | OPOST | unistra-opost | 9-1 | window | x64 | | DEV | Meta paquet pour les logiciels OPOST | 06/09/2024 09:20:44 | gitlab-pkgbuild-20231218 | 8.1 KB |

*Figure 8: Gestion de l'ajout de logiciels dans un méta-paquet*

En complément de ces méta-paquets, nous avons également profité des capacités de WAPT pour faire un nettoyage des versions des logiciels obsolètes (postes disposant toujours de l'ancien antivirus, d'anciennes configurations logicielles qui coexistaient avec les nouvelles). Nous avons mis en place un « paquet conditionnel » dans lequel nous avons inscrit l'ensemble des nettoyages à réaliser sur le parc par l'intermédiaire du « setup.py ».

```
83  # Nettoyage Kaspersky
84  for kaspersky in installed_softwares(name="Agent d'administration de Kaspersky Security Center"):
85      print('suppression {}'.format(kaspersky['name']))
86      run('msiexec /x ' + kaspersky["key"].replace('InstallWIX_', '') + ' /qn KLLOGIN=    KLPASSWD=          ')
87
88  for kaspersky in installed_softwares(name="Kaspersky Endpoint Security for Windows"):
89      print('suppression {}'.format(kaspersky['name']))
90      run('msiexec /x ' + kaspersky["key"] + ' /qn KLLOGIN=    KLPASSWD=          ')
```

*Figure 9: Nettoyage de notre ancien antivirus "Kaspersky"*

Au final, pour limiter le nombre de paquets sur les OU, nous avons combiné dans certains cas, les fonctions de « méta-paquet » et « paquet conditionnel » en un seul et même paquet. Cela permet de gérer les installations/désinstallations de logiciels et les modifications de configurations en simultanés.

Comme indiqué en amont, nous avons également revu le standard des logiciels qui sont installés de base sur un poste administratif. Ce standard datait de la période COVID et nous avions choisi d'intégrer l'ensemble des logiciels de visio-conférence sur nos postes portables (Zoom,Skype,Teams,Webex).

Après discussion en interne, nous avons décidé que l'ensemble de ces logiciels ne serait plus installé de base sur les nouveaux postes, pour revenir à une base saine sur les nouveaux postes. Nous avons décidé de ne pas les désinstaller sur les anciens postes, tout en assurant un suivi dans les montées de versions. Nous avons donc mis en place un paquet conditionnel qui s'occupe de détecter la présence d'un de ces logiciels sur le poste, dans le but de déployer le paquet WAPT du logiciel, qui lui permettrait de se maintenir à jour dans l'avenir.

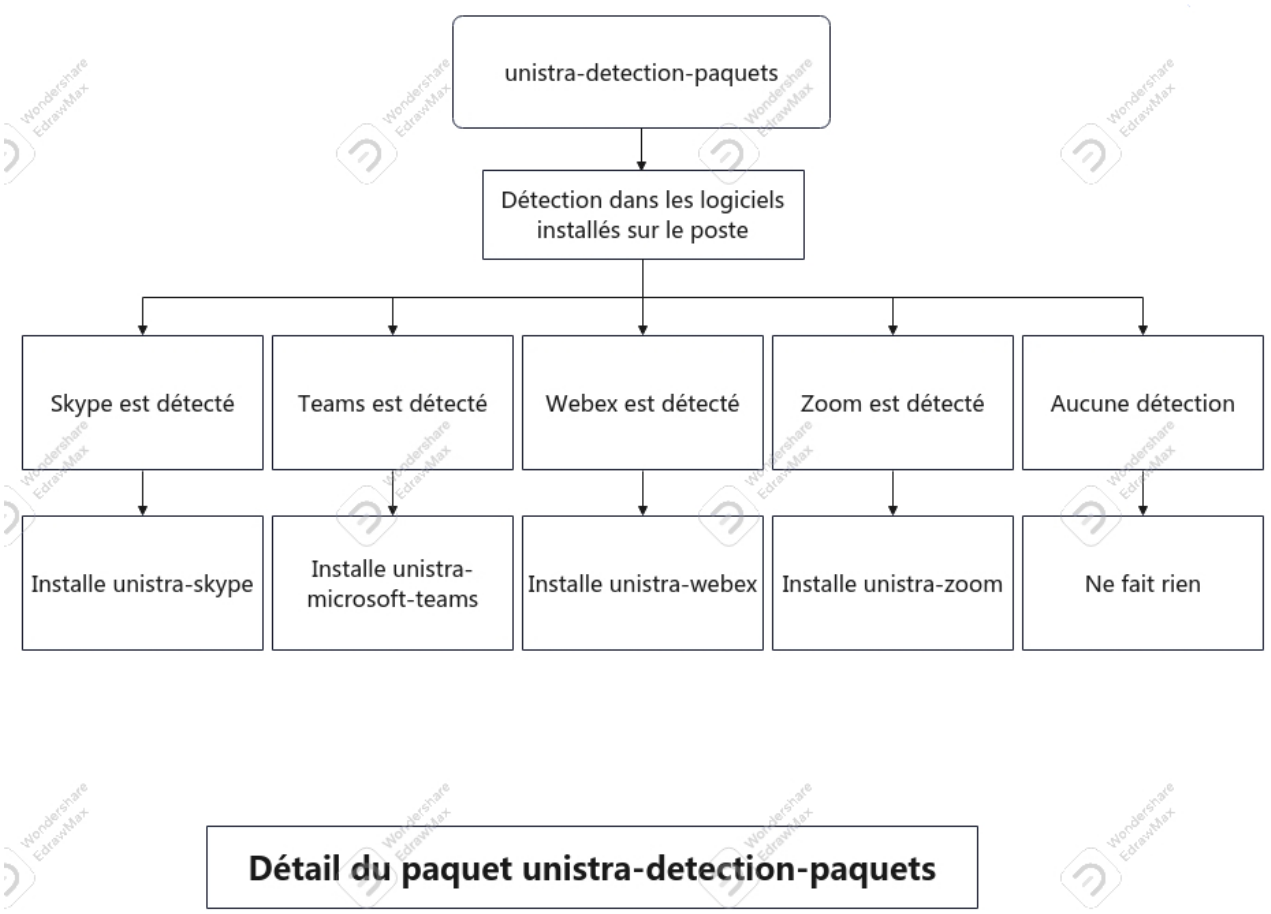


*Figure 10: Détail du paquet détection des outils de visio-conférence*

Grâce à ces paquets d'auto-détection, nous avons intégré dans WAPT la gestion des mises à jour des différents logiciels de visio-conférences sans devoir les déployer en masse sur l'ensemble du parc.

# 5 Les problèmes rencontrés et leur gestion

Au cours des premiers mois, nous avons rencontré plusieurs difficultés, notamment avec le système d'importation des machines depuis l'inventaire vers le déploiement d'OS ainsi que l'envoi et le déploiement de logiciels volumineux (exemple : AutoCAD).

En ce qui concerne le système d'importation de l'inventaire dans le déploiement d'OS, celui-ci provoquait des erreurs et faisait planter la console d'administration, car il n'était pas conçu pour gérer un nombre aussi élevé de machines. À notre demande, l'éditeur a corrigé ce problème en ajoutant un bouton dans l'onglet inventaire, permettant d'exporter une ou plusieurs machines vers WADS, évitant ainsi l'importation de l'inventaire complet, qui comprend plus de 7500 postes dans notre cas.

Pour les paquets volumineux, nous avons modifié la configuration du fichier « wapt.conf ». Les paramètres ajustés ont été communiqués à l'éditeur, qui les a intégrés dans une mise à jour de WAPT.

Suite à une réflexion avec l'éditeur au moment du choix de l'architecture, nous avons décidé de monter un seul serveur avec cette configuration :


*Figure 11: Configuration initiale du serveur WAPT*

À la suite d'un test de montée en charge effectué par un gestionnaire de parc au sein d'une composante, celui-ci avait déployé une salle entière avec plusieurs logiciels volumineux, dont certains atteignant environ 50 Go. Ce déploiement avait complètement paralysé notre serveur ainsi que les consoles d'administration des gestionnaires installées sur leur poste et qui y sont connectées. Nous avons travaillé étroitement avec l'éditeur pour trouver des solutions d'amélioration permettant de stabiliser la solution. Nous avons également procédé à la mise à niveau des caractéristiques de notre serveur, lui permettant à présent de supporter les déploiements à fortes charges :


*Figure 12: Configuration actuelle du serveur WAPT*

# 6 Nos développements en cours et à venir

Depuis fin juin, nous avons mis en place le « self-service » proposé par WAPT sur l'ensemble des postes des personnels gérés par la Direction du Numérique. Celui-ci met à disposition un catalogue d'environ 90 logiciels.

Cela permet de donner un peu plus d'autonomie à nos utilisateurs qui ne disposent pas de droits d'administrateur sur leur poste.

## 6.1 Gestion des mises à jour Windows au travers de WAPT

Nous avons également testé le système de gestion des mises à jour Windows via WAPT, en utilisant le service WUA (Windows Update Agent), sur quelques salles de ressources. Nous avons constaté que l'envoi et l'installation des mises à jour via WAPT sont nettement plus efficaces que le système WSUS (Windows Server Update Services) dans les salles de ressources de type « classe mobile » équipées de PC portables, souvent limités par un seul lien réseau partagé. Cela nous a permis de réduire de moitié le temps nécessaire pour que tous les postes téléchargent et installent leurs mises à jour. Nous envisageons donc de déployer ce système sur l'ensemble de notre parc prochainement.

## 6.2 Déploiement et utilisation de scripts d'exploitation

Notre ancien outil était capable d'exécuter des scripts de manière ponctuelle (à la demande) sur une ou plusieurs machines. Nous savons que ce n'est pas une fonction native de WAPT, mais nous avons tout de même trouvé une solution pour faire quelque chose de similaire sur celui-ci. Nous avons créé des paquets avec une nomenclature « préfixe_paquet-script-nom_du_paquet » et avons transposé nos scripts en langage Python que nous intégrons dans le « setup.py ». Les paquets script sont ensuite installés sur l'ensemble des postes et nous avons indiqué dans le «setup.py» du paquet, que celui-ci ne devait s'exécuter que par le biais de la commande « Installation forcée ». Voici un exemple de notre paquet script qui va désinstaller et réinstaller les imprimantes sur un poste :

```
# -*- coding: utf-8 -*-
from setuphelpers import *
from waptservice.enterprise import get_active_sessions,
start_interactive_process
import time

def install():
    if force:
        for session_id in get_active_sessions():
            print(f'Suppression des imprimantes réseau sur la session
{session_id}')
            start_interactive_process(r"wmic", "printer where \"Local='FALSE'\"
delete", session_id=session_id, hide=True)


        time.sleep(1)


        print(f'Suppression de toutes les clés restantes')
        registry_deletekey(root=HKEY_LOCAL_MACHINE, path=r'SOFTWARE\Microsoft\
Windows NT\CurrentVersion\Print\Providers', keyname='Client Side Rendering Print
Provider', force=True, recursive=True)


        print(f'Relance du spooler')
        service_restart("spooler")
        print(f'Spooler relancé')


        print(f'Pruge des pilotes non utilisés')
        run(r"cscript %WINDIR%\System32\Printing_Admin_Scripts\fr-FR\Prndrvr.vbs
-x")


        for session_id in get_active_sessions():
            print(f'Lancement forcée de la gpo sur la session {session_id}')
            start_interactive_process(r"cscript", r"\\nom.domaine.fr\SysVol\
nom.domaine.fr\Policies\{E1618AB6-4D98-49D5-A03C-6169471C4A57}\User\Scripts\
Logon\mount_printer.vbs", session_id=session_id, hide=True)


def uninstall():
    pass
```

# 7 Conclusion

La migration vers WAPT à l'Université de Strasbourg a permis de moderniser la gestion des postes informatiques en remplaçant Altiris, devenu onéreux et limité. WAPT a été sélectionné pour répondre aux besoins de déploiement de systèmes et d'applications tout en optimisant les coûts, passant de 60k€/an pour Altiris à un peu moins de 30k€/an pour WAPT. Le calcul des coûts en ressources humaines reste pour l'instant difficile, car l'adaptation au nouvel outil a exigé des ressources accrues en mode projet. Depuis son lancement en septembre 2023, le projet a introduit des innovations comme le déploiement d'ISO Microsoft via WADS, l'intégration continue avec GitLab pour gérer les paquets et automatiser au maximum leurs mises à jour tout en sécurisant le processus. L'adoption d'une approche DevOps a également permis d'améliorer les méthodes de travail des gestionnaires informatiques.

Par ailleurs, WAPT a permis de conserver les mêmes restrictions de vues et d'actions par périmètre qu'auparavant, tout en ajoutant des mesures de sécurité renforcées pour éviter les désordres précédemment rencontrés avec Altiris. Grâce à ce nouvel outil, certains gestionnaires ont pu acquérir de nouvelles compétences et endosser davantage de responsabilités, par exemple en devenant mainteneurs de paquets ou administrateurs de la solution. Bien que le changement suscite des réticences dans une structure de cette envergure, il s'avère globalement bénéfique pour l'avenir.