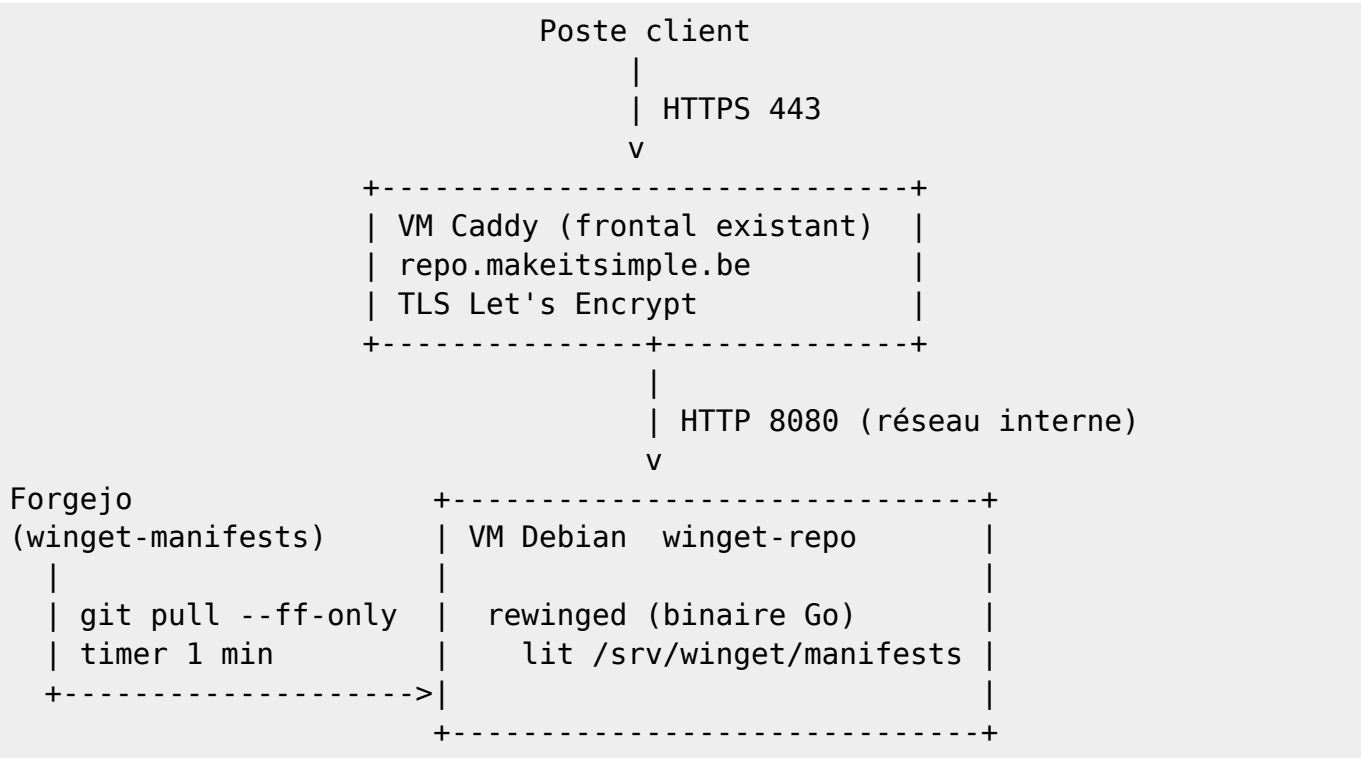


Serveur winget privé (rewinged)

Source de paquets winget auto-hébergée pour le parc géré Make IT Simple. Permet d'aligner les versions logicielles sur l'ensemble des parcs clients sans donner à l'agent de déploiement le droit de remplacer son propre binaire.

1. Architecture



Composant	Machine	Rôle
Forgejo	existante	Source de vérité des manifests. Revue et historique.
Caddy	frontal existant	Terminaison TLS uniquement. Aucun fichier local.
rewinged	VM dédiée	Sert l'API REST Microsoft.Rest et les installateurs.
Wazuh	existant	FIM sur manifests et binaire de la VM rewinged.

Deux machines distinctes. rewinged n'écoute donc pas sur la boucle locale mais sur l'interface interne, filtrée au pare-feu vers la seule IP du frontal.

Principe de sécurité : l'intégrité des paquets ne repose pas sur le TLS mais sur le SHA256 déclaré dans chaque manifest. Un installateur substitué échoue à la vérification et winget refuse l'installation.

2. Prérequis

- VM Debian 12 minimale, 1 vCPU / 1 Go RAM / 20 Go suffisent
- Enregistrement DNS public `repo.makeitsimple.be` vers l'IP de la VM
- Ports 80 et 443 joignables (ACME + service)

- Dépôt Forgejo mis/winget-manifests créé
- Clé de déploiement Forgejo **en lecture seule**

3. Installation de rewinged

3.1 Utilisateur et arborescence

```
useradd --system --home /srv/winget --shell /usr/sbin/nologin rewinged
mkdir -p /srv/winget/{manifests,installers}
chown -R rewinged:rewinged /srv/winget
```

3.2 Binaire

Télécharger la release depuis [GitHub](#).

Épingler une version précise. Le projet est en pré-1.0 : l'auteur indique que configuration, sortie et comportement peuvent changer à tout moment.

```
VERSION="0.15.0"
cd /tmp
# adapter le nom d'archive à celui de la release
tar xzf rewinged_${VERSION}_linux_amd64.tar.gz
install -m 0755 rewinged /usr/local/bin/rewinged
/usr/local/bin/rewinged --help
```

<WRAP center round important> Relever dans `--help` le nom exact du flag désignant le répertoire de manifests, et vérifier si rewinged recharge à chaud ou s'il faut un signal après un `git pull`. Adapter les sections 3.3 et 6 en conséquence. </WRAP>

3.3 Service systemd

[/etc/systemd/system/rewinged.service](#)

```
[Unit]
Description=rewinged winget package source
After=network.target

[Service]
User=rewinged
Group=rewinged
ExecStart=/usr/local/bin/rewinged -listen 10.0.0.20:8080
Restart=on-failure
RestartSec=5

# Durcissement
ProtectSystem=strict
```

```
ProtectHome=true
PrivateTmp=true
PrivateDevices=true
NoNewPrivileges=true
RestrictSUIDSGID=true
LockPersonality=true
MemoryDenyWriteExecute=true
ReadOnlyPaths=/srv/winget/manifests
ReadWritePaths=/srv/winget/installers

[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
systemctl enable --now rewinged
systemctl status rewinged
```

rewinged écoute en clair sur le réseau interne. Le TLS est assuré par le frontal Caddy, sur une autre machine.

3.4 Pare-feu de la VM rewinged

Le port 8080 ne doit être joignable que par le frontal.

```
CADDY_IP="10.0.0.10"
ufw default deny incoming
ufw allow from ${CADDY_IP} to any port 8080 proto tcp
ufw allow from 100.64.0.0/10 to any port 22 proto tcp
ufw enable
```

4. Configuration du frontal Caddy

À ajouter au Caddyfile **de la machine frontale**, pas de la VM rewinged.

</etc/caddy/Caddyfile>

```
repo.makeitsimple.be {
    tls vincent@makeitsimple.be

    reverse_proxy 10.0.0.20:8080

    header {
        Strict-Transport-Security "max-age=31536000"
        X-Content-Type-Options "nosniff"
        -Server
    }
```

```
log {
    output file /var/log/caddy/repo.log
    format json
}
```

Un seul `reverse_proxy`, sans bloc `file_server` : le frontal n'héberge aucun fichier de la solution. Les installateurs sont servis soit par Forgejo, soit par rewinged lui-même (section 6).

Winget exige HTTPS avec une chaîne validée par le magasin de confiance Windows. Let's Encrypt sur un nom public évite d'avoir à déployer une CA interne sur chaque poste.

5. Dépôt Git des manifests

Le dépôt ne contient **que du YAML**. Les binaires sont hébergés séparément (voir section 6).

5.1 Structure

```
winget-manifests/
├── manifests/
│   └── m/
│       ├── MakeItSimple/
│       │   ├── Agent/
│       │   │   └── 1.2.0/
│       │   │       └── MakeItSimple.Agent.yaml
│       │   └── RustDesk/
│       │       └── 1.4.3/
│       │           └── MakeItSimple.RustDesk.yaml
└── README.md
```

5.2 Clone initial

```
sudo -u rewinged git clone \
git@forgejo.makeitsimple.be:mis/winget-manifests.git \
/srv/winget/manifests
```

La clé de déploiement doit être en **lecture seule** : le serveur de paquets ne doit jamais pouvoir écrire dans le dépôt.

6. Stockage des installeurs

Trois options. Choix retenu : **package registry Forgejo**, qui évite le SFTP tout en gardant le dépôt Git léger et purement textuel.

Option	Avantage	Inconvénient
Package registry Forgejo	Dépôt Git léger, publication en une commande	Un système de plus à connaître
Git LFS	Manifest et binaire dans un même commit	Pas de delta, volume croissant
URL éditeur	Rien à héberger	Ne convient qu'aux logiciels tiers

6.1 Publication d'un installeur

```
curl --user vincent:$FORGEJO_TOKEN \
  --upload-file mis-agent-1.2.0.msi \
  https://forgejo.makeitsimple.be/api/packages/mis/generic/agent/1.2.0/mis-agent-1.2.0.msi
```

L'URL obtenue est celle à placer dans le champ `InstallerUrl` du manifest.

<WRAP center round important> **Point bloquant à vérifier.** Winget ne sait pas s'authentifier auprès d'une source auto-hébergée : il télécharge l'installeur en anonyme. Si l'organisation Forgejo mis est privée, le téléchargement échouera sur tous les postes.

Tester avant de bâtir dessus :

```
curl -I
https://forgejo.makeitsimple.be/api/packages/mis/generic/agent/1.2.0/mis-agent-1.2.0.msi
```

Un 401 ou 404 impose de rendre l'organisation lisible publiquement, ou de basculer sur l'internalisation rewinged (6.2). </WRAP>

6.2 Logiciels tiers

Pour Chrome, Acrobat et assimilés, deux voies :

- pointer directement l'URL éditeur dans le manifest
- utiliser l'internalisation de rewinged, qui télécharge et sert lui-même

l'installeur — utile pour figer une version que l'éditeur a retirée, et

pour les postes sans accès internet direct

7. Synchronisation automatique

Pas de webhook : un timer évite d'exposer un endpoint entrant sur la machine qui décide de ce que le parc installe. Le flux part toujours du serveur vers Forgejo.

[/etc/systemd/system/winget-sync.service](#)

[Unit]

```
Description=Sync winget manifests from Forgejo
```

```
[Service]
```

```
Type=oneshot
```

```
User=rewinged
```

```
WorkingDirectory=/srv/winget/manifests
```

```
ExecStart=/usr/bin/git pull --ff-only
```

[/etc/systemd/system/winget-sync.timer](#)

```
[Unit]
```

```
Description=Sync winget manifests every minute
```

```
[Timer]
```

```
OnBootSec=1min
```

```
OnUnitActiveSec=1min
```

```
[Install]
```

```
WantedBy=timers.target
```

```
systemctl daemon-reload
systemctl enable --now winget-sync.timer
systemctl list-timers winget-sync.timer
```

Le `--ff-only` est délibéré : si l'historique local diverge, le pull échoue au lieu de fusionner silencieusement. Une divergence signifie une écriture locale, ce qui ne doit jamais arriver.

8. Format d'un manifest

[MakeItSimple.Agent.yaml](#)

```
PackageIdentifier: MakeItSimple.Agent
```

```
PackageVersion: 1.2.0
```

```
PackageLocale: en-US
```

```
Publisher: Make IT Simple SRL
```

```
PackageName: MIS Agent
```

```
License: Proprietary
```

```
ShortDescription: Agent d'inventaire et de mise à jour
```

```
Installers:
```

```
- Architecture: x64
```

```
  InstallerType: msi
```

```
  InstallerUrl:
```

```
https://forgejo.makeitsimple.be/api/packages/mis/generic/agent/1.2.0/mis-agent-1.2.0.msi
```

```
  InstallerSha256: A1B2C3D4E5F6...
```

```
  Scope: machine
```

```
InstallerSwitches:
  Silent: /quiet /norestart
AppsAndFeaturesEntries:
  - ProductCode: '{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}'
    UpgradeCode: '{YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY}'
ManifestType: singleton
ManifestVersion: 1.6.0
```

Calcul du hash :

```
sha256sum mis-agent-1.2.0.msi | tr 'a-f' 'A-F'
```

Types d'installateurs supportés : msi, exe, inno, nullsoft, wix, burn, msix, appx, zip, portable. Ne pas utiliser portable pour un service Windows — il ne fait qu'un alias sur le PATH.

Le bloc AppsAndFeaturesEntries sert à l'adoption des logiciels déjà installés depuis d'anciens paquets custom : y lister leurs ProductCode pour que winget établisse le lien.

9. Workflow de publication

1. Construire l'installateur, relever son ProductCode (Orca ou Get -Package)
2. Pousser le binaire dans le package registry Forgejo (6.1)
3. Calculer le SHA256
4. Créer le manifest dans une branche, ouvrir une PR
5. Revue, merge sur main
6. Le timer applique dans la minute
7. Valider depuis un poste témoin avant diffusion

```
winget search -s mis-repo MakeItSimple
winget install --id MakeItSimple.Agent -s mis-repo --silent
```

10. Configuration des postes

10.1 Enregistrement de la source

```
winget source add -n mis-repo `
-a https://repo.makeitsimple.be/api `
-t "Microsoft.Rest"
```

À exécuter en **scope machine**, sinon la source n'existe que pour l'utilisateur courant et reste invisible à l'agent tournant en service.

10.2 Winget en contexte SYSTEM

winget n'est pas dans le PATH pour le compte SYSTEM. Résoudre le chemin versionné :

```
$winget = (Get-ChildItem `
    "C:\Program
Files\WindowsApps\Microsoft.DesktopAppInstaller_*_x64__8wekyb3d8bbwe\winget.
exe" |
    Select-Object -Last 1).FullName
```

Ajouter `-disable-interactivity` à tous les appels non interactifs, sinon un prompt bloque le processus indéfiniment.

10.3 Adoption du parc existant

Un paquet installé autrement que depuis la source privée n'est pas reconnu par winget upgrade tant qu'il n'a pas été installé une fois depuis cette source. Deux contournements :

- renseigner les AppsAndFeaturesEntries des anciens builds
- passer une réinstallation depuis mis - repo

À valider sur un poste témoin **avant** de dimensionner la campagne.

10.4 Compatibilité

Système	App Installer
Windows 10 1809+ / 11	Livré
Windows Server 2025	Livré
Windows Server 2019 / 2022	Absent — installation manuelle, non supportée
Éditions N et LTSC sans Store	Installation manuelle

Inventorier la part de Server 2019/2022 du parc : elle décide si winget est le canal unique ou s'il faut un second canal.

11. Surveillance Wazuh

11.1 FIM

[ossec.conf](#)

```
<syscheck>
  <directories check_all="yes" realtime="yes" report_changes="yes">
    /srv/winget/manifests
  </directories>
  <directories check_all="yes" realtime="yes">
    /srv/winget/installers
  </directories>
  <directories check_all="yes" realtime="yes">
```

```
/usr/local/bin
</directories>
</syscheck>
```

11.2 Réduction du bruit

La FIM se déclenche à chaque `git pull` légitime. Pour distinguer le légitime de l'anormal :

- signer les commits en GPG et vérifier la signature dans le service de

sync avant application — un manifest modifié sans commit signé devient

```
alors une alerte non ambiguë
* à défaut, corrélér avec les logs de ''winget-sync.service''
```

11.3 Alertes complémentaires

- arrêt du service `rewinged`
- échecs répétés de `winget-sync.timer` — un dépôt figé est un mode de

panne silencieux

- volumétrie anormale de téléchargements sur `/files/`

12. Points ouverts

Sujet	Action
Flags <code>rewinged</code>	Confirmer le nom du flag de répertoire et le rechargement à chaud via <code>—help</code>
Bug de matching	Tester sur un poste témoin avec un logiciel préinstallé
Server 2019/2022	Inventorier la proportion dans le parc
Upgrade RustDesk	Vérifier que la mise à jour préserve <code>RustDesk2.toml</code> et le mot de passe permanent
Rétention	Définir une politique de purge des anciennes versions d'installateurs
Forgejo anonyme	Vérifier que les paquets génériques sont téléchargeables sans token (6.1)
Internalisation	Relever le chemin d'URL servi par <code>rewinged</code> si Forgejo n'est pas lisible en anonyme

13. VersionID

```
function Get-MsiCodes {
    param([string]$Path)
    $installer = New-Object -ComObject WindowsInstaller.Installer
    $db =
    $installer.GetType().InvokeMember('OpenDatabase', 'InvokeMethod', $null, $installer, @( $Path, 0 ))
```

```
$props = @{}
foreach ($p in
'ProductCode','UpgradeCode','ProductVersion','ProductName') {
    $view =
$db.GetType().InvokeMember('OpenView','InvokeMethod',$null,$db,
        @("SELECT Value FROM Property WHERE Property='$p'"))
$view.GetType().InvokeMember('Execute','InvokeMethod',$null,$view,$null)
    $rec =
$view.GetType().InvokeMember('Fetch','InvokeMethod',$null,$view,$null)
    if ($rec) {
        $props[$p] =
$rec.GetType().InvokeMember('StringData','GetProperty',$null,$rec,1)
    }
}
[PSCustomObject]$props
}
```

Get-MsiCodes 'C:\build\mis-agent-1.2.0.msi'

14. Références

- [rewinged — dépôt GitHub](#)
- [Documentation Windows Package Manager](#)
- [RustDesk Server Pro — console et tokens API](#)

From:

<https://wiki.makeitsimple.be/> - **makeITsimple** wiki

Permanent link:

<https://wiki.makeitsimple.be/doku.php?id=linux:rewinged&rev=1789196740>

Last update: **2026/09/12 07:05**

