OpenWEC

Un serveur de collecte de journaux d'événements Windows basé sur le protocole WEF

William BRUNEAU & Vincent RUELLO - SSTIC 2023

1. Collecte de logs sous Windows

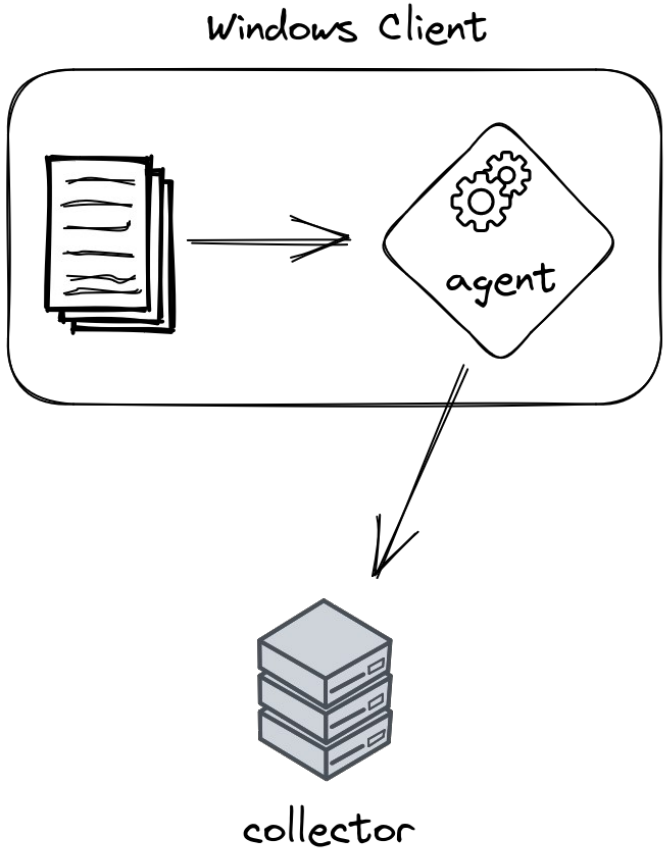
En version courte

Collecte de journaux d'événements

Avec un agent local tiers

Un agent récupère le contenu des journaux **localement** et les transmet via un protocole quelconque.

- Augmentation de la **surface d'attaque**
- Agent potentiellement privilégié

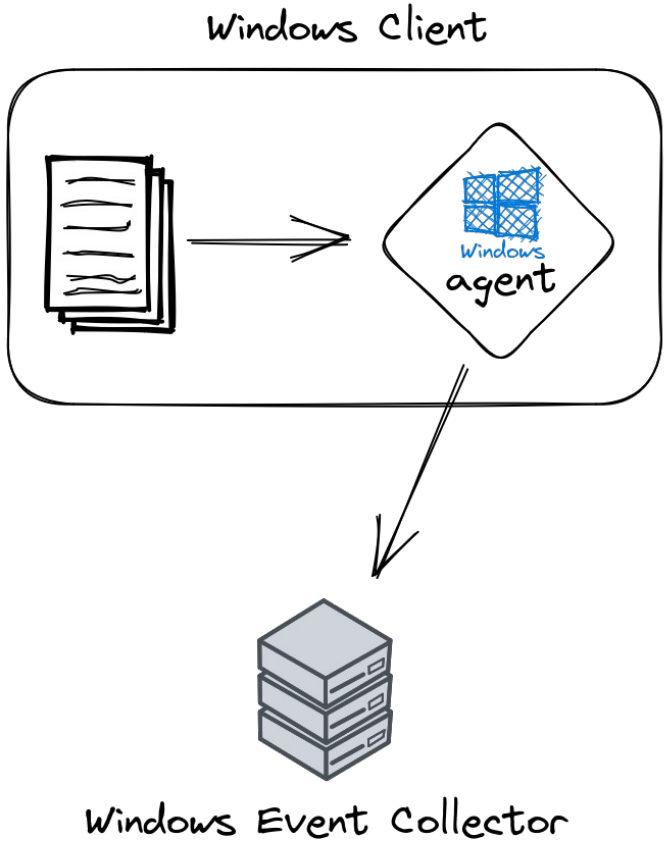


Collecte de journaux d'événements

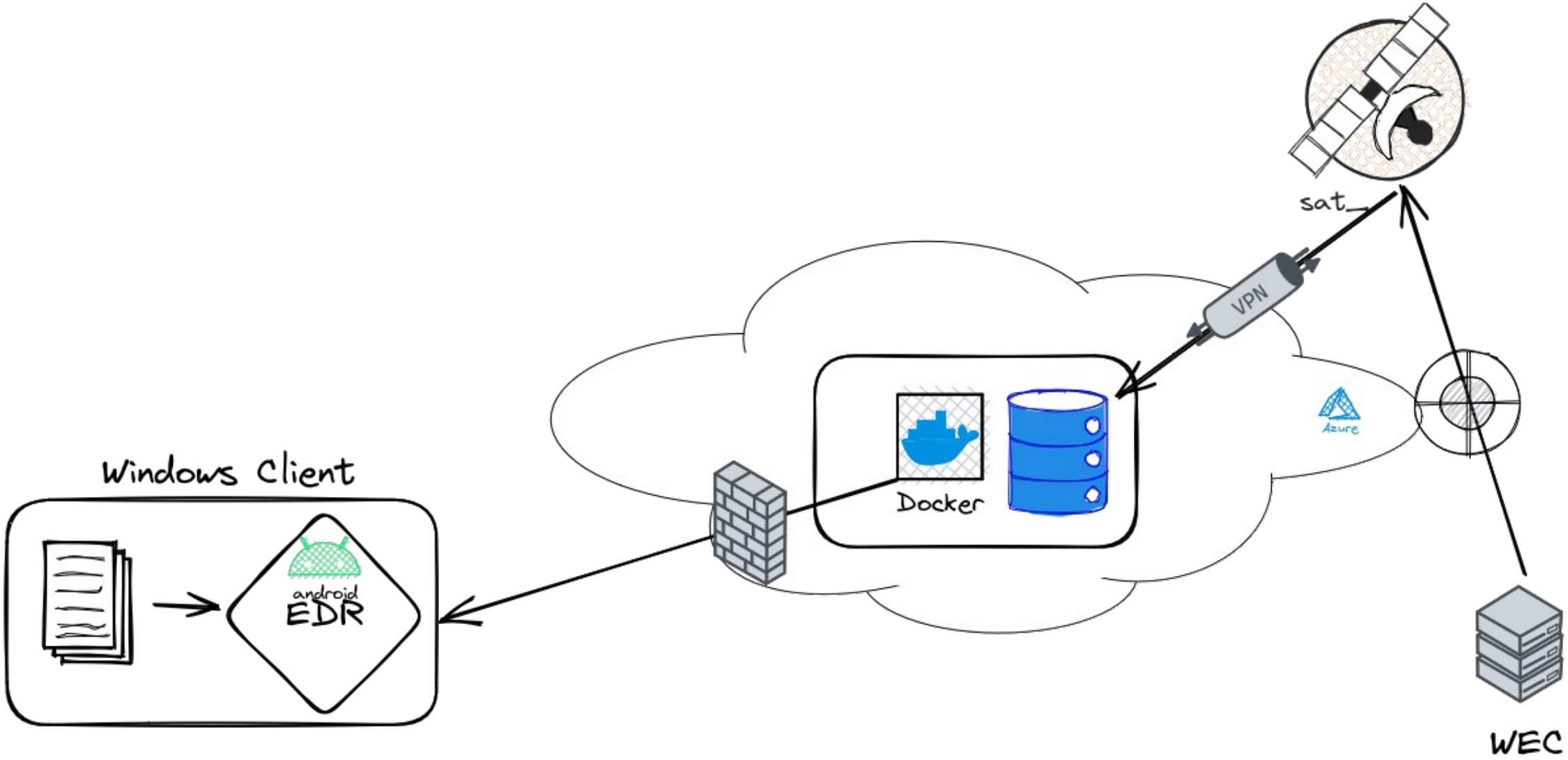
Avec l'agent local natif

On utilise l'EventLog-Forwarder builtin de Windows.

- **N'augmente pas (ou peu) la surface d'attaque**
- Protocole **Windows Event Forwarding**
- S'exécute avec les privilèges « Network Service »



Collecte de journaux d'événements



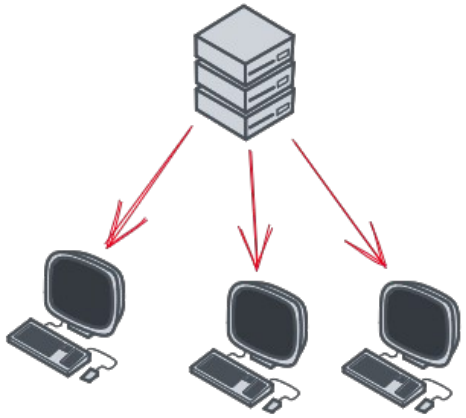
Windows Event Forwarding

- Basé sur Web Services for Management (*WS-Management*)
- Surcouche Microsoft documentée (MS-WSMV)
- HTTP/SOAP (**XML** 💔)
- **Authentification** et **chiffrement** (via Kerberos ou TLS)
- Les événements sont **compressés** (SLDC)
- Plusieurs configurations possibles : mode de transfert, délai, ...

Modes de transfert

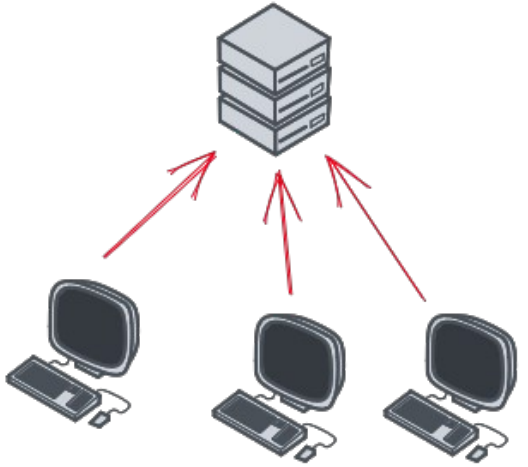
« Pull » ou « Collector Initiated »

Le collecteur se connecte aux clients pour récupérer leurs événements.



« Push » ou « Source Initiated »

Chaque client se connecte au collecteur pour lui envoyer des événements.



Choix du collecteur WEF

Utiliser le collecteur builtin de Windows Server ?

- Windows Event Collector (WEC)

- Pas de **redondance**
- Informations « manquantes »
  - Adresse IP des clients
  - Identification des clients (principal Kerberos)
- Intégration au SIEM ?
- **Maîtrise ?**

Choix du collecteur WEF

Utiliser une solution commerciale sous Linux ?

- Fonctionnalités limitées
- Informations toujours « manquantes »
- Maîtrise ?

- Exemples : NXLog, Cribl Edge

Choix du collecteur WEF

Et si on réinventait la roue ?

- On **maîtrise** vraiment l'outil
- On peut **l'adapter à nos besoins**
- On peut rendre disponible pour la communauté une implémentation de ce protocole
  - PoC existant sur Github : owinec

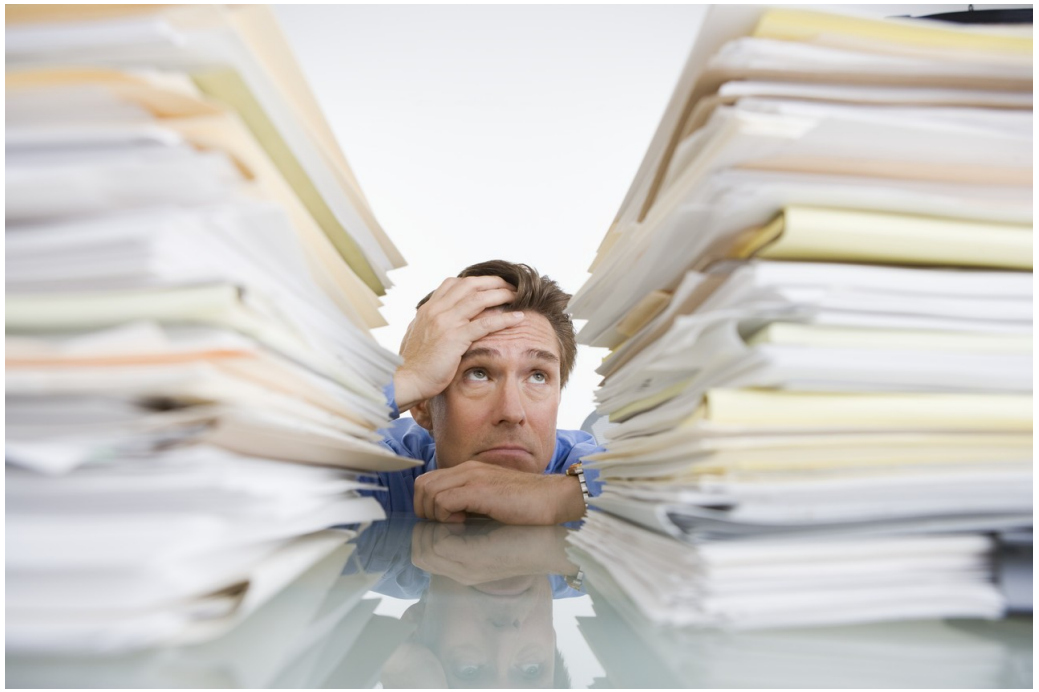
- mais… **il faut comprendre comment ça marche** !

2. Le protocole Windows Event Forwarding

Enfin… ce qu'on en a compris !

Méthodologie

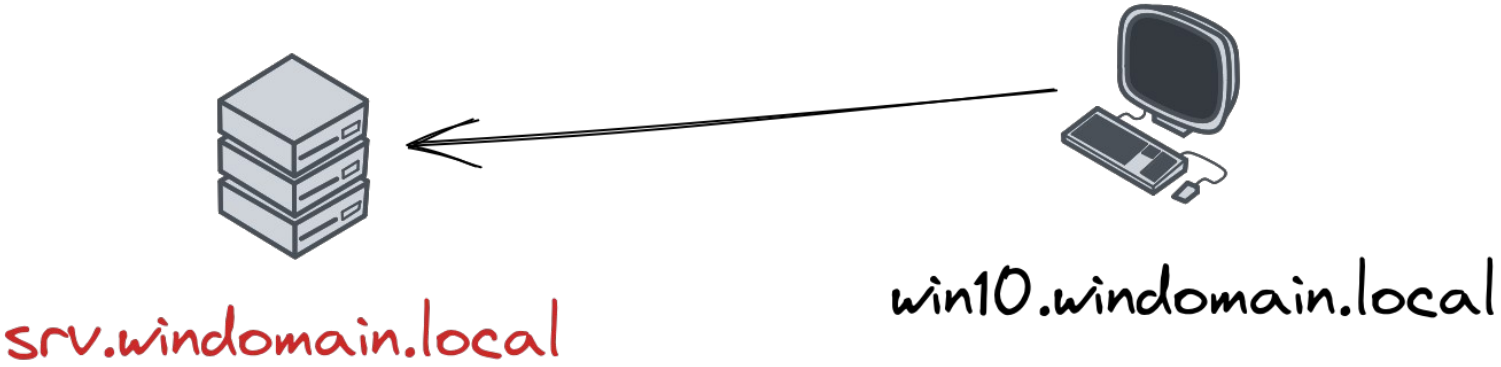
- Un peu (beaucoup) de **documentation** :
  - WS-Management : DSP0226_1.0.0
  - Surcouche de Microsoft : MS-WSMV
  - SLDC : ECMA-321
- **Analyse de captures réseaux**
- ~~Rétro-ingénierie~~



Lab' élite

- Configuration du client Windows :

```
Server=http://srv.windomain.local:5985/wsman/SubscriptionManager/WEC,Refresh=30
```



Le client s'authentifie auprès du collecteur

- Le client s'authentifie via **Kerberos** auprès du collecteur :

```
POST /wsman/SubscriptionManager/WEC HTTP/1.1
Connection: Keep-Alive
Content-Type: application/soap+xml;charset=UTF-16
Authorization: Kerberos YIIH9AYJKoZIhvcSAQI...
User-Agent: Microsoft WinRM Client
Content-Length: 0
Host: srv.windomain.local:5985
```

Le client envoie des données chiffrées

- Le client envoie une requête **multipart** contenant des données **chiffrées** par une **clé de session Kerberos**.

```
    POST /wsman/SubscriptionManager/WEC HTTP/1.1
    Content-Type: multipart/encrypted;protocol="application/HTTP-Kerberos-session-encrypted";boundary="Encrypted Boundary"
    [...]

    --Encrypted Boundary
    Content-Type: application/HTTP-Kerberos-session-encrypted
    OriginalContent: type=application/soap+xml;charset=UTF-16;Length=3240
    --Encrypted Boundary
    Content-Type: application/octet-stream
    [blob de données chiffrées]
    --Encrypted Boundary--
```

Déchiffrement des données échangées

- Wireshark peut :
  - déchiffrer les parties chiffrées des tickets Kerberos
  - déchiffrer les **données** chiffrées avec une clé de session Kerberos
  - à condition de **lui fournir une keytab** !

- On génère une keytab avec les secrets de `srv.windomain.local` :
  - Récupération des clés avec `secretsdump.py` (impacket 💚)
  - Génération d'une keytab avec le module `keytab` de gmsad

Contenu des messages

```
<s:Envelope>
    <s:Header>
        <a:To>http://srv.windomain.local:5985/wsman/SubscriptionManager/WEC</a:To>
        <m:MachineID>win10.windomain.local</m:MachineID>
        <a:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</a:Action>
        [...]
    </s:Header>
    <s:Body>
        <n:Enumerate>
            <w:OptimizeEnumeration/>
            <w:MaxElements>32000</w:MaxElements>
        </n:Enumerate>
    </s:Body>
</s:Envelope>
```

D'autres messages


TOO MUCH XML
imgflip.com

Abonnement

Permet de collecter des événements.

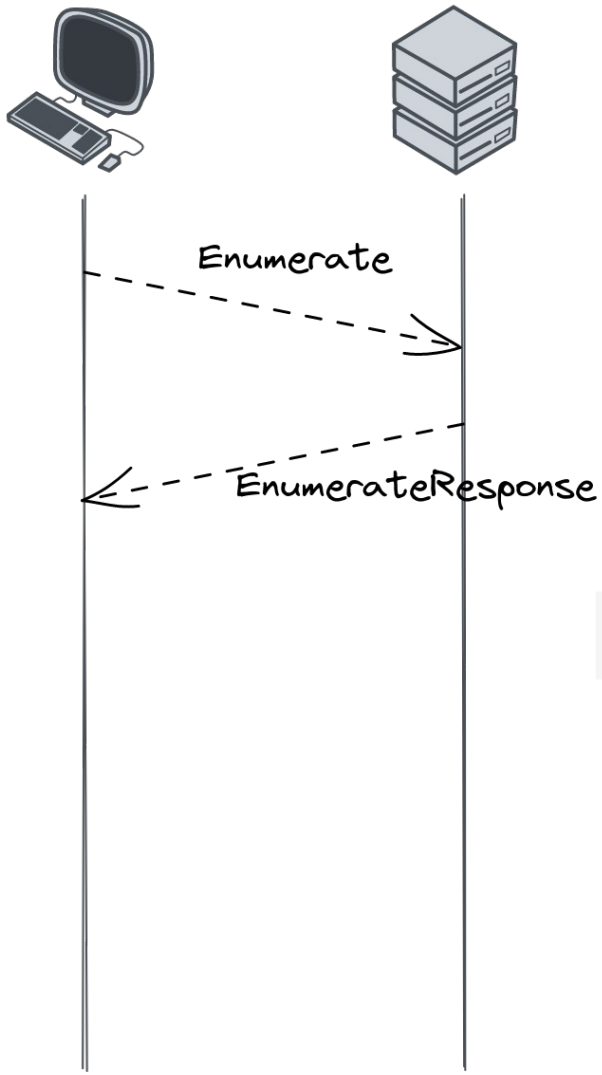
- Un **nom**
- Un **filtre** d'événements
- Une **configuration de transfert** d'événements
- Une **version**

Bookmark

- Représente un **pointeur** dans le **flux d'événements** d'un client
- **Envoyé par le client** avec chaque batch d'événements
- Le collecteur **sauvegarde** le dernier *bookmark* envoyé par **chacun des clients pour chaque abonnement**
- Permet au collecteur d'indiquer au client où il en est dans son flux d'événements

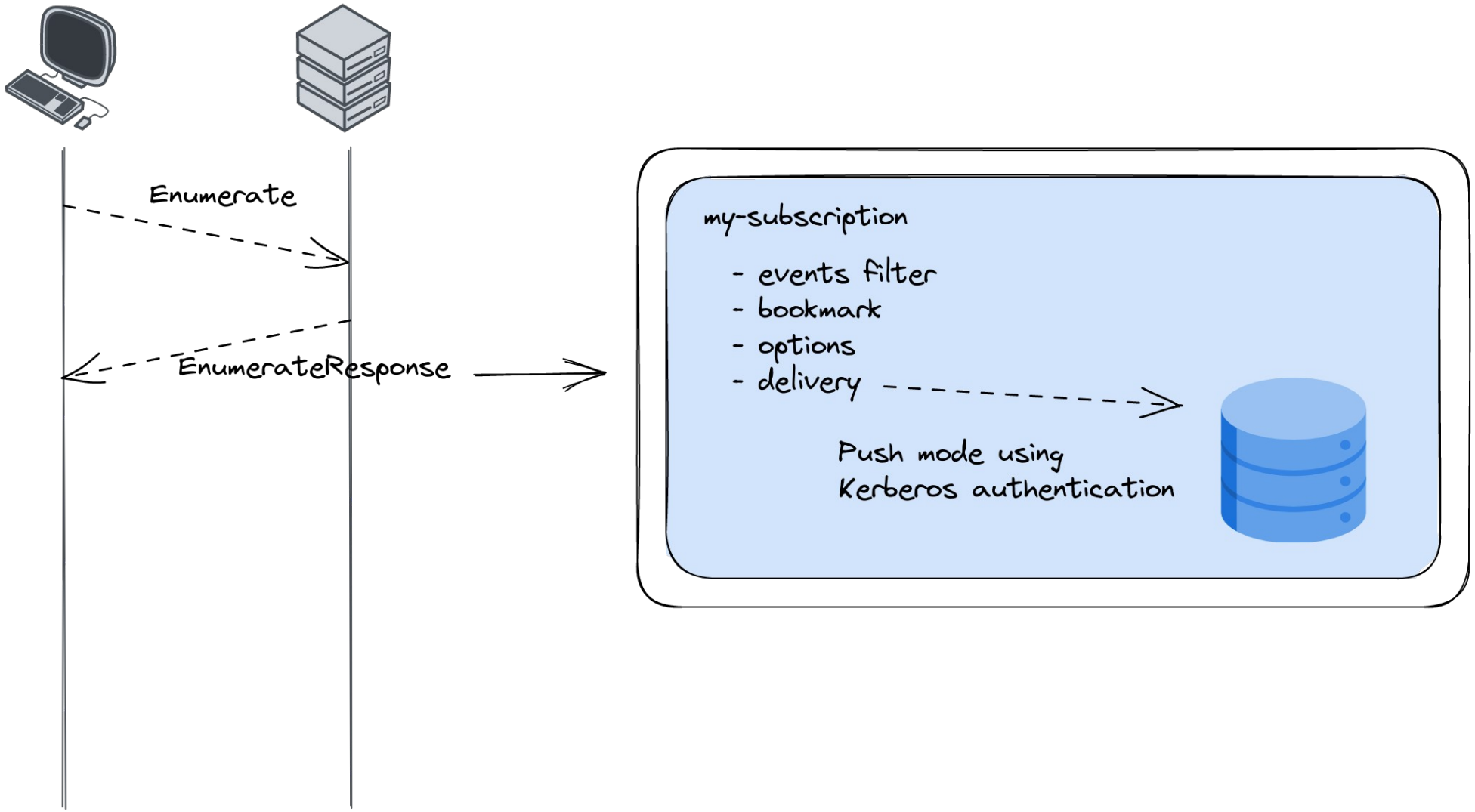
```
<BookmarkList>
    <Bookmark Channel="Microsoft-Windows-WinRM/Operational" RecordId="149161"
IsCurrent="true"/>
</BookmarkList>
```

Protocole

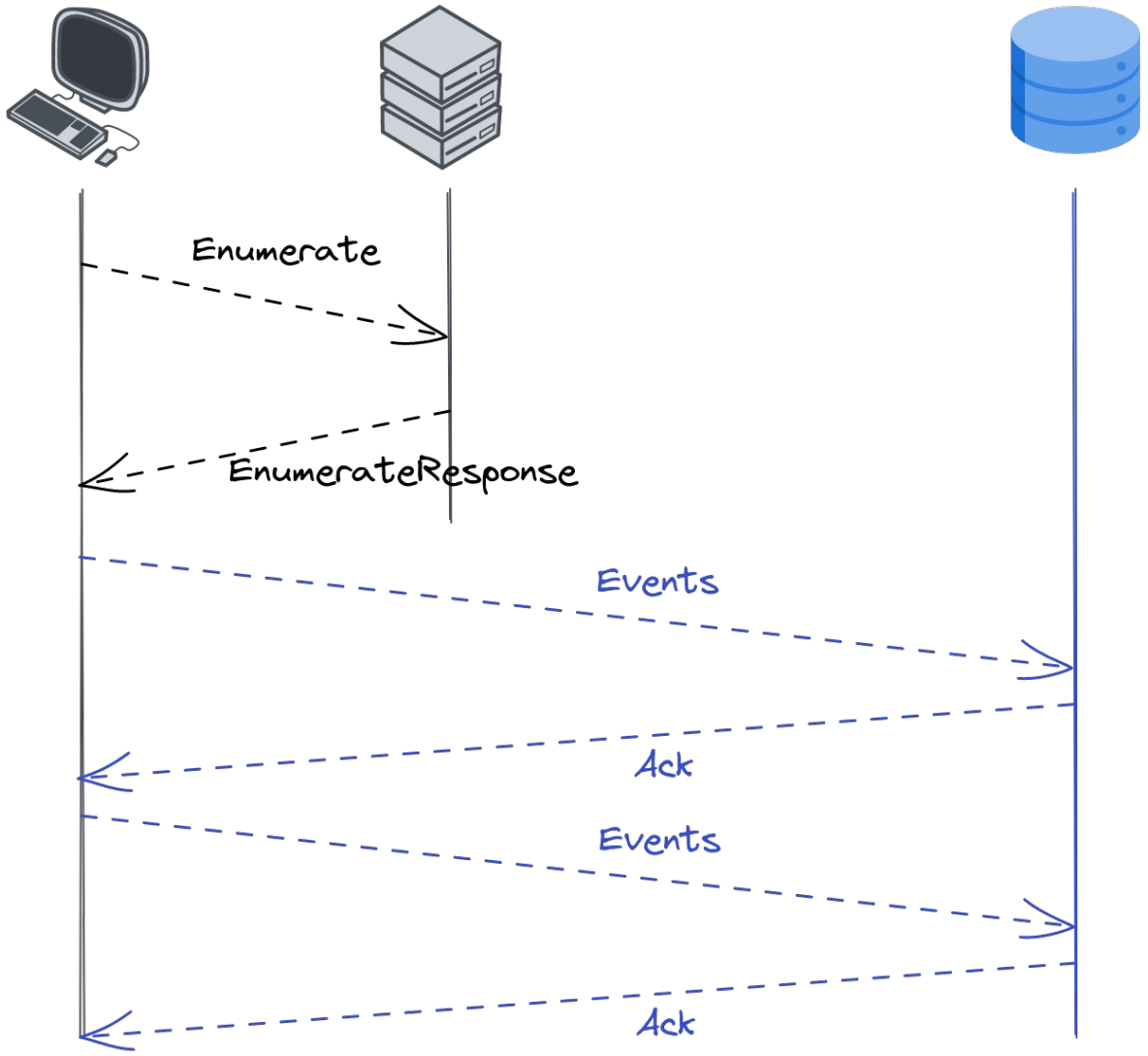



```
Server=http://srv.windomain.local:5985/...,Refresh=30
```

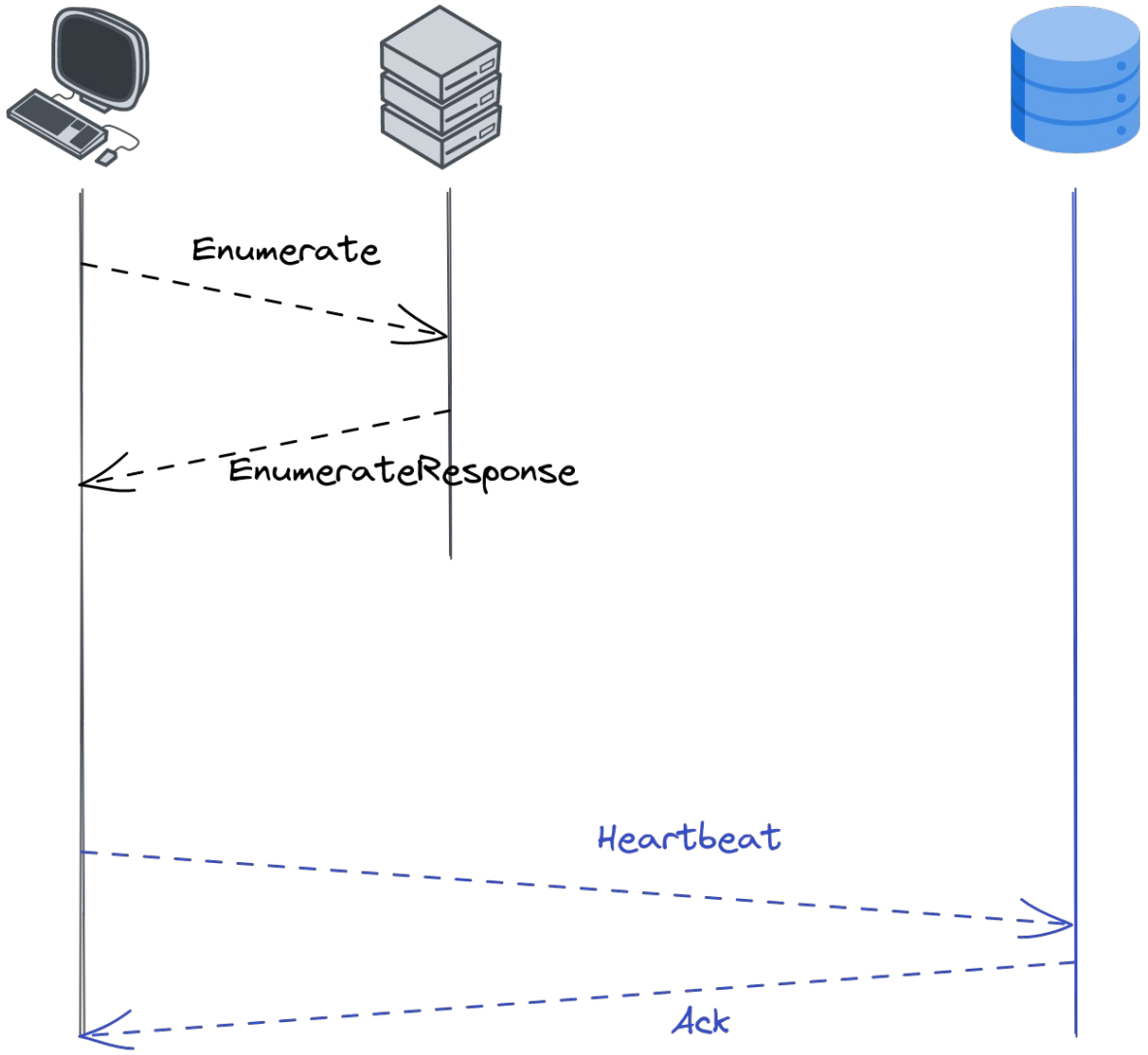
Protocole


Enumerate
EnumerateResponse
my-subscription
- events filter
- bookmark
- options
- delivery
Push mode using
Kerberos authentication

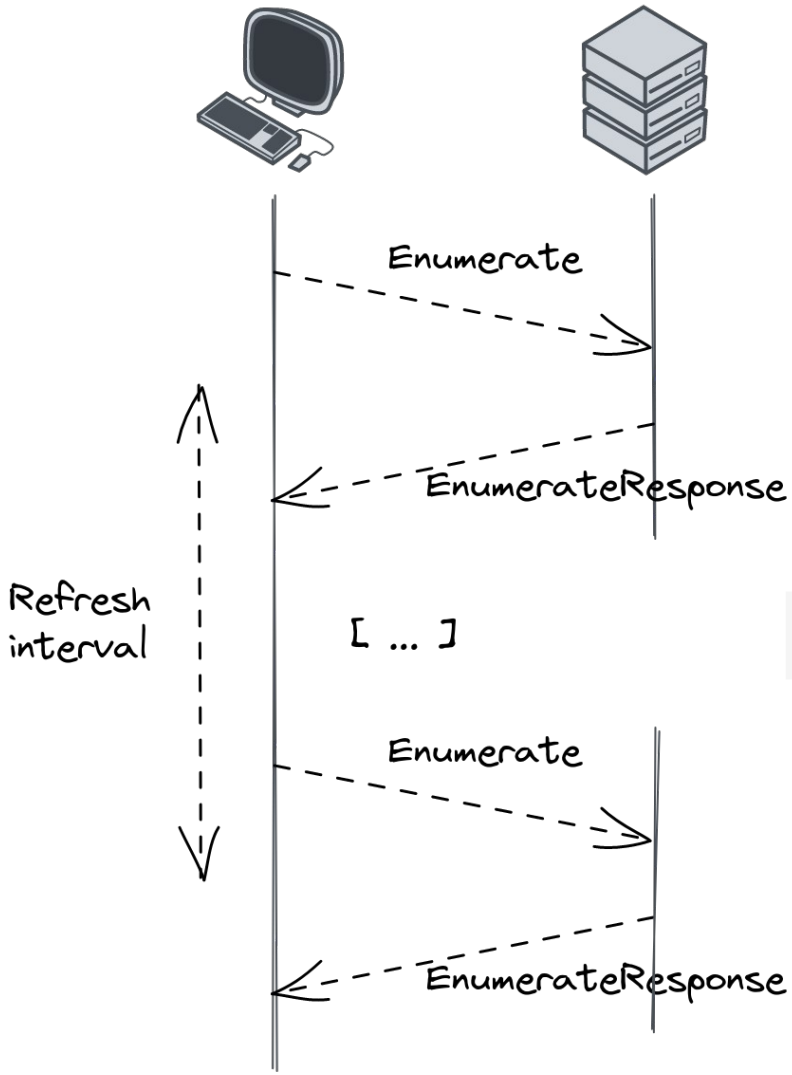
Protocole



Protocole

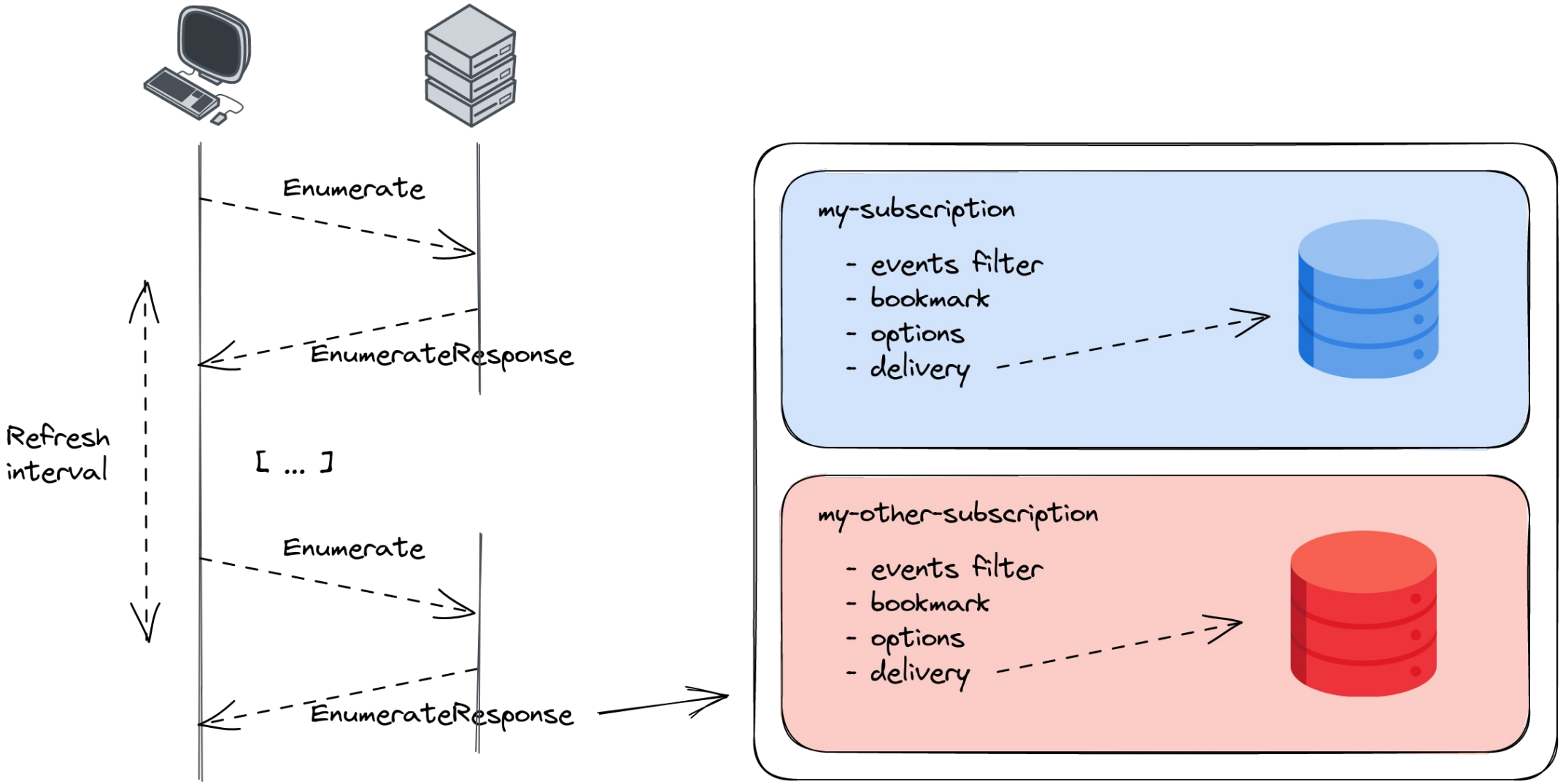


Protocole

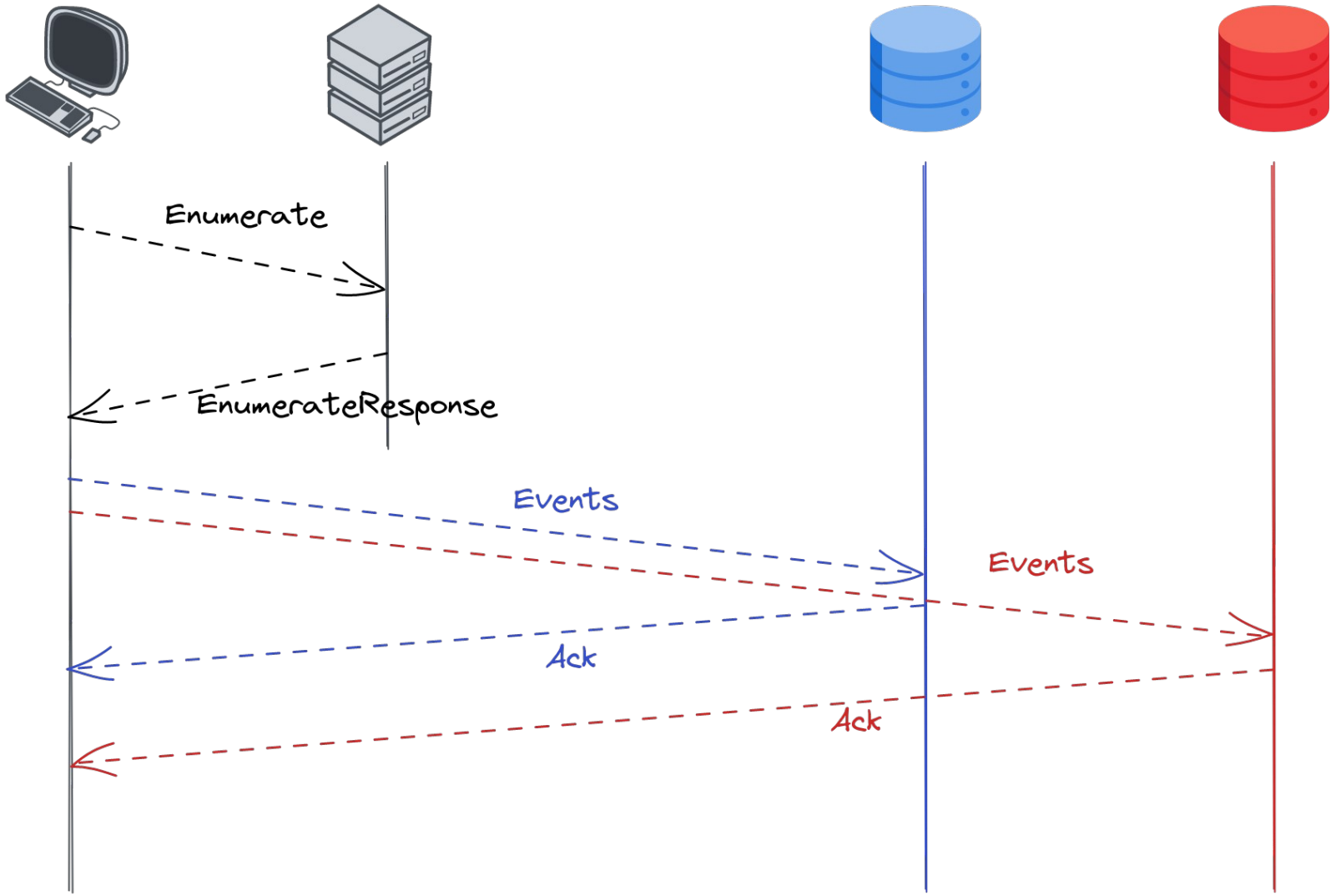



```
Server=http://srv.windomain.local:5985/...,Refresh=30
```

Protocole


Enumerate
EnumerateResponse
Refresh interval
[...]
Enumerate
EnumerateResponse
my-subscription
- events filter
- bookmark
- options
- delivery
my-other-subscription
- events filter
- bookmark
- options
- delivery

Protocole



3. Ça a l'air faisable, let's go !

OpenWEC

Première tentative

- Été 2021 :
  - Premier POC en Python en 2021
  - Support de TLS
  - Merci Romain ❤️

- Été 2022 : nouveau départ !

Cahier des charges

- Implémentation du protocole WEF
  - Uniquement en mode « **Push** »
  - Support de **Kerberos** (authentification et chiffrement)
  - Support de la **compression**
- En **Rust**
- Pour **Linux**
- Plusieurs sorties possibles (fichiers, Kafka, ...)
- Redondance et répartition de charge

OpenWEC

https://github.com/cea-sec/openwec

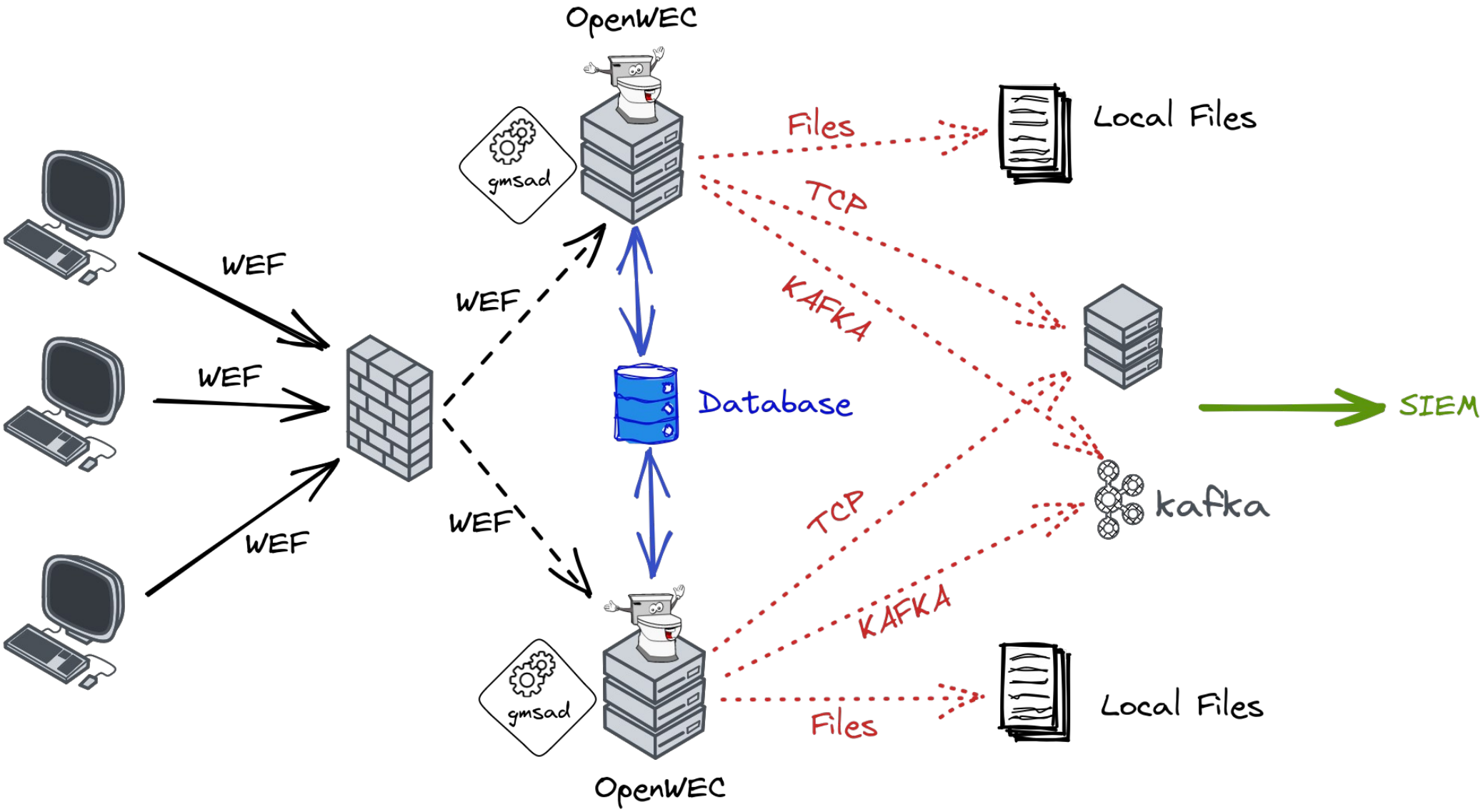
OpenWEC

- Les **abonnements** et leurs **métadonnées** (bookmarks, ...) sont stockées dans une **base de données** :
  - SQLite
    - Un seul noeud
    - Donc pas de redondance/répartition de charge
  - Postgres
    - Pensé pour CockroachDB
    - Plusieurs noeuds possibles

OpenWEC

- Une fois récupérés, les événements sont écrits dans une ou plusieurs **sorties** propres à **chaque abonnement**.
- Chaque sortie est associée à un format d'évémenents :
  - Raw (XML de base)
  - Json
- Plusieurs types de sorties sont disponibles :
  - Fichiers (local au collecteur)
  - TCP
  - Kafka

Architecture




“ Une démo vaut mille mots. ”
Jean-Michel Jedemaux

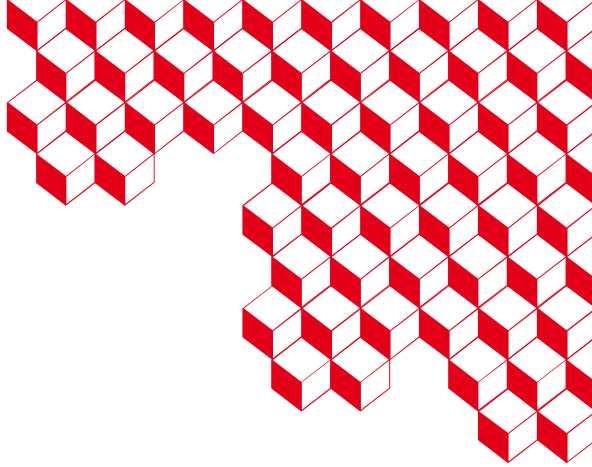
Retour d'expérience

- Déployé sur **quelques milliers de machines** sur un de nos réseaux
- Problèmes liés au déploiement de WEF
  - Filtre d'événements : nombre de sources limité, volume associé à certaines sources
  - **Permissions** sur les sources

  - Ressource utile : guide de l'ANSSI « Recommandations de sécurité pour la journalisation des systèmes Microsoft Windows en environnement Active Directory »

Évolutions

- Ajouter le **support de TLS**
- Intégration de l'outil dans des distributions
- Implémentation d'autres **sorties et formats**



On recrute !

https://github.com/cea-sec/openwec

CEA DAM

Centre de Bruyères-le-Châtel

91297 Arpajon Cedex

Établissement public à caractère industriel et commercial

William BRUNEAU & Vincent RUELLO - SSTIC 2023

Sunny Qi
10,731 • Microsoft Vendor

May 18, 2021, 11:19 AM

Hi,

Thanks for posting in Q&A platform.

Event forwarding depends on WSMan/WinRM (windows remote management service). For domain joined scenarios, this uses Kerberos as a default for authentication and encryption, which requires a service principal name (SPN). SPNs are meant to be unique. Therefore, no two domain joined computers should be permitted to register the same SPN for their computer accounts/identity, which makes load balancing with default setup unworkable. I'm afraid your goal cannot achieved by Windows Failover Clustering since the service provided by cluster was failover service such as when server 1 down, then server 2 will work continually.

https://learn.microsoft.com/en-us/answers/questions/398630/how-to-load-balance-multiple-windows-event-collect

Le client énumère les abonnements du collecteur

```
<s:Envelope>
    <s:Header>
        <a:To>http://srv.windomain.local:5985/wsman/SubscriptionManager/WEC</a:To>
        <m:MachineID>win10.windomain.local</m:MachineID>
        <a:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</a:Action>
        [...]
    </s:Header>
    <s:Body>
        <n:Enumerate>
            <w:OptimizeEnumeration/>
            <w:MaxElements>32000</w:MaxElements>
        </n:Enumerate>
    </s:Body>
</s:Envelope>
```

Le collecteur renvoit une liste d'abonnements

```
<s:Envelope>
    <s:Header>
        <a:Action>
            http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse
        </a:Action>
        [...]
    </s:Header>
    <s:Body>
        <wsen:EnumerateResponse>
            <wsen:EnumerationContext></wsen:EnumerationContext>
            <wsman:Items>
                [abonnements]
            </wsman:Items>
            <wsman:EndOfSequence/>
        </wsen:EnumerateResponse>
    </s:Body>
</s:Envelope>
```

Abonnement

```
<m:Subscription>
    <m:Version>uuid:219C5353-5F3D-4CD7-A644-F6B69E57C1C1</m:Version>
    <s:Envelope>
        <s:Header>
            <a:Action>http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe</a:Action>
            <w:OptionSet>
                <w:Option Name="SubscriptionName">Toto</w:Option>
                <w:Option Name="Compression">SLDC</w:Option>
                <w:Option Name="CDATA" xsi:nil="true"/>
                <w:Option Name="ContentFormat">RenderedText</w:Option>
            </w:OptionSet>
            [...]
        </s:Header>
        <s:Body>
            [Body]
        </s:Body>
    </s:Envelope>
</m:Subscription>
```

Abonnement

```
<e:Subscribe>
    <w:Filter Dialect="http://schemas.microsoft.com/win/2004/08/events/eventquery">
        <QueryList>
            <Query Id="0">
                <Select Path="Microsoft-Windows-WinRM/Operational">*</Select>
            </Query>
        </QueryList>
    </w:Filter>
    <w:Bookmark>
        <BookmarkList>
            <Bookmark Channel="Microsoft-Windows-WinRM/Operational" RecordId="149140"
IsCurrent="true"/>
        </BookmarkList>
    </w:Bookmark>
    <w:SendBookmarks/>
    <e:Delivery Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/Events">
        [delivery]
    </e:Delivery>
    [...]
</e:Subscribe>
```

Abonnement

```
<e:Delivery Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/Events">
    <e:NotifyTo>
        <a:Address>HTTP://srv.windomain.local:5985/wsman/subscriptions/B6BDBB59-FB07-4EE5-
841F-EBEC9D67CDD4/1</a:Address>
        <c:Policy>
            <c:ExactlyOne>
                <c:All>
                    <auth:Authentication
Profile="http://schemas.dmtf.org/wbem/wsman/1/wsman/secprofile/http/spnego-kerberos"></
auth:Authentication>
                </c:All>
            </c:ExactlyOne>
        </c:Policy>
        [...]
    </e:NotifyTo>
    <w:Heartbeats>PT3600.000S</w:Heartbeats>
    <w:ConnectionRetry Total="5">PT60.0S</w:ConnectionRetry>
    <w:MaxTime>PT30.000S</w:MaxTime>
    <w:MaxEnvelopeSize Policy="Notify">512000</w:MaxEnvelopeSize>
    <w:ContentEncoding>UTF-16</w:ContentEncoding>
</e:Delivery>
```

Le client envoie des événements

- Le client ouvre une **nouvelle connexion TCP** vers l'adresse de destination de l'abonnement.
- Après une ré-authentification, le client envoie des événements **compressés via SLDC** :

```
<s:Envelope>
    <s:Header>
        <a:Action>http://schemas.dmtf.org/wbem/wsman/1/wsman/Events</a:Action>
        <w:Bookmark>
            <BookmarkList>
                <Bookmark Channel="Microsoft-Windows-WinRM/Operational" RecordId="149161"
IsCurrent="true"/>
            </BookmarkList>
        </w:Bookmark>
        <w:AckRequested/>
        [...]
    </s:Header>
    <s:Body>
        [events]
    </s:Body>
</s:Envelope>
```

Événements Windows

- Chaque message « Events » contient un batch d'événements Windows :

```
<w:Events>
    <w:Event Action="http://schemas.dmtf.org/wbem/wsman/1/wsman/Event">
        <![CDATA[<Event>[...]</Event>]]>
    </w:Event>
    <w:Event Action="http://schemas.dmtf.org/wbem/wsman/1/wsman/Event">
        <![CDATA[<Event>[...]</Event>]]>
    </w:Event>
    [...]
</w:Event>
```