

Het gebruik van ebMS contracten (CPA's) in complexe infrastructuren

Whitepaper

Ernst Jan van Nigtevecht

Oktober 2009

Inleiding

Contracten die gepubliceerd worden voor een ebMS service hebben tot doel om de ebMS services te beschrijven zoals die in het publieke domein zichtbaar zijn. Echter, de organisaties die de services implementeren hebben een bepaalde *infrastructuur* met daarin onder andere een ebMS adapter¹. Die infrastructuur kan eenvoudig zijn, maar ook complex. Denk hierbij aan het gebruik van een SSL-offloading device (proxy of load-balancer met ssl-offloading), het gebruik van verschillende zones (zoals een DMZ), XML Firewalls, etcetera. Hoe complexer de infrastructuur, hoe groter de kans is dat contracten zoals die gepubliceerd zijn voor het publieke domein niet 1-op-1 bruikbaar is voor de ebMS adapter.

De vraag moet daarom gesteld worden hoe een contract, dat gepubliceerd is voor het publieke domein, gebruikt moet worden voor de adapter in de infrastructuur van zo'n organisatie. Om deze vraag te concretiseren worden een aantal infrastructuren besproken. Aan de hand daarvan zal bekeken worden hoe een ebMS contract (CPA) gebruikt moet worden voor de configuratie van de ebMS adapters.

ebMS Contract

Een ebMS contract is een overeenkomst tussen twee organisaties en waarmee het ebMS koppelvlak geconfigureerd wordt. In dat contract worden naast de functionele zaken (de 'services' en de 'operaties') ook logistieke gegevens gepubliceerd. In dit verhaal zullen we ons concentreren op een beperkt aantal logistieke gegevens in ebMS contract. (De technische weergave van het contract heet een 'Collaboration Protocol Agreement', een CPA.) Het gaat dan om:

- De **identificatie** van de organisatie (of afdeling van een organisatie) op basis van een partyId.
- Het **transport url** (het 'endpoint') zoals die vanuit het publieke domein zichtbaar is. Deze url is zichtbaar in het publieke domein: het is geen url die alleen bereikbaar is 'binnen' de eigen overheidsorganisatie.
- De **publieke certificaten** voor de client en server authenticatie zoals die zichtbaar zijn op transport nivo (HTTPS) of voor de beveiliging van de 'payload' (door versleuteling en/of ondertekening).

Het contract zoals dat wordt gepubliceerd zal de interne infrastructuur als een 'black box' beschouwen. De vraag is dan in hoeverre het contract bruikbaar is voor de configuratie² van de ebMS adapter.

1) De ebMS adapter verzorgt de verwerking van de ebMS berichten.

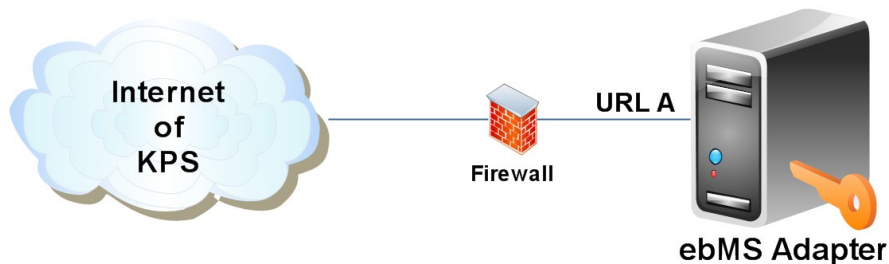
2) Er wordt vanuit gegaan dat het contract, een CPA, als xml-document ingelezen kan worden in de ebMS adapter waardoor de configuratie geheel automatisch doorgevoerd wordt.

Infrastructuren

De implementatie van de ebMS adapter in de infrastructuur is afhankelijk van verschillende factoren. Die factoren kunnen zijn het beveiligingsbeleid, outsourcing, performance eisen, beschikbaarheids eisen, etcetera. Omdat niet op voorhand te zeggen is hoe deze zaken door een organisatie zijn ingevuld, zullen we drie typen infrastructuren beschouwen.

INFRA 1

Als eerste beschouwen we een eenvoudige infrastructuur. Hierin wordt de ebMS adapter direct aangesloten op het publieke domein en verzorgt de ebMS adapter de HTTP TLS/SSL beveiliging en authenticatie (met de client/server certificaten). Een eventuele firewall zal transparant zijn voor de HTTP TLS/SSL; de firewall zal ingaand verkeer naar de ebMS adapter toestaan. Het onderstaande figuur schetst deze infrastructuur.



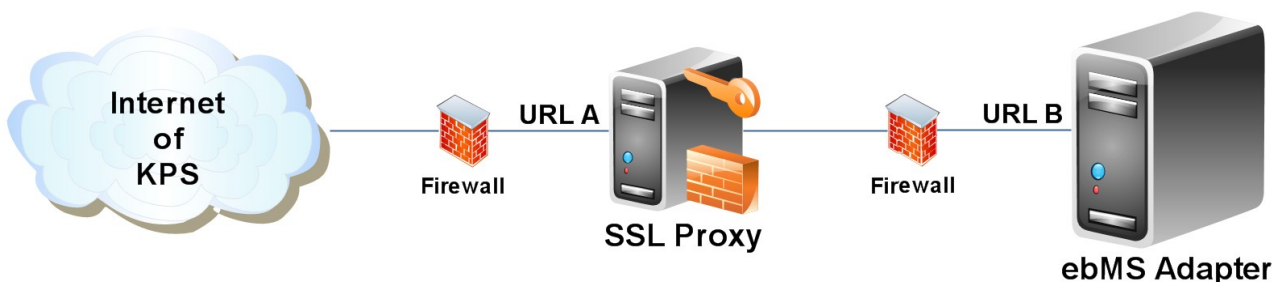
Figuur 1: Eenvoudige infrastructuur.

De ebMS adapter is voor de buitenwereld bereikbaar met URL A. De ebMS adapter heeft de private keys van alle certificaten (het client certificaat, het server certificaat en de payload beveiligings certificaat).

Het ebMS contract zoals dat wordt gepubliceerd bevat URL A en de public keys van de certificaten. Dit contract kan 1-op-1 worden ingelezen door de ebMS adapter voor de configuratie.

INFRA 2

Het tweede type infrastructuur voegt enige complexiteit toe door het gebruik van een niet transparante SSL-offloader, zowel voor ingaand als voor uitgaand verkeer. De ebMS adapter heeft een interne koppeling met de SSL-offloader, via HTTP (zonder TLS). Merk op dat de SSL-offloader ook een onderdeel kan zijn van een load-balancer om de performance en/of de beschikbaarheid te verhogen (waardoor de ebMS adapter in een cluster zal komen te staan). Het onderstaande figuur schetst deze infrastructuur waarbij de SSL-offloader als “SSL Proxy” getekend wordt.



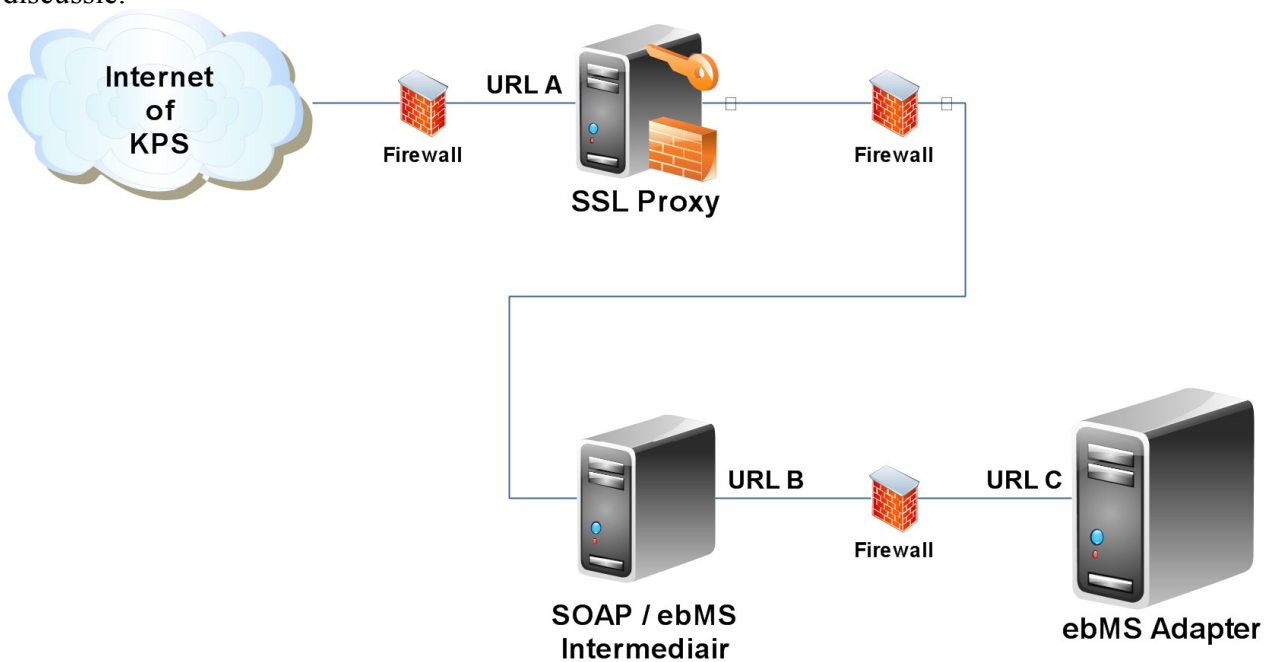
Figuur 2: Infrastructuur met SSL offloading.

De ebMS adapter is in deze infrastructuur niet direct bereikbaar voor de buitenwereld. De aangegeven url (URL A) adresseert alleen de SSL proxy. Deze zal de TLS/SSL termineren (en opzetten naar buiten voor uitgaand verkeer) en zal berichten naar de ebMS adapter doorsturen naar de ebMS adapter op basis van een andere url, te weten: URL B. Deze url maakt geen gebruik van TLS/SSL. De ebMS adapter heeft geen private keys van de client en server certificaten, maar wel van de payload beveiligings certificaten.

INFRA 3

Als derde type infrastructuur wordt op basis van het tweede type een intermediair geïntroduceerd. De intermediair is een soort gateway ('broker') met dien verstande dat het ebMS protocol 'doorgezet' wordt: de ebMS berichten worden gerouteerd naar achterliggende systemen met het ebMS protocol.

De routing kan plaatsvinden op basis van de functionele aspecten (de service, de actie of operatie, de rol, ...) maar ook op basis van de **identificatie** (partyId). Het partyId is per slot van rekening uniek voor die (deel-) organisatie of afdeling en kan dus gebruikt worden voor de routing. Voor de eenvoud gaan we uit van routing op basis van de partyId, maar is feitelijk ondergeschikt aan de discussie.



Figuur 3: Infrastructuur met SSL-offloading en een SOAP/ebMS Intermediair.

De ebMS adapter is in deze infrastructuur niet direct bereikbaar voor de buitenwereld. De aangegeven url (URL A) adresseert alleen de SSL proxy. Deze zal de TLS/SSL termineren (en opzetten naar buiten voor uitgaand verkeer) en zal berichten naar de intermediair doorsturen. De intermediair zal op basis van de partyId (van de ontvanger) het bericht doorsturen naar de juiste ebMS adapter; in het voorbeeld is dit het adres met URL C (deze url maakt geen gebruik van TLS/SSL).

Als er berichten verstuurd worden vanuit de ebMS adapter, dan wordt hiervoor de (interne) url gebruikt van de intermediair: URL B. De intermediair zal het bericht op basis van de partyId van de ontvanger doorsturen, via de SSL Proxy, naar de ontvanger. De SSL proxy handelt de TLS/SSL af. De ebMS adapter heeft geen private keys van de client en server certificaten, maar wel van de payload beveiligings certificaten.

Transformatie stappen

Het ebMS contract zoals dat voor het publieke domein gepubliceerd is, beschouwd de interne infrastructuur van een organisatie als een 'black box'. Om het contract te kunnen gebruiken voor de configuratie van de ebMS adapter in de eigen infrastructuur zullen er een aantal transformatie stappen doorgevoerd moeten worden. Hieronder wordt beschreven welke dit zijn, voor de beschreven infrastructuren.

INFRA 1

Voor deze infrastructuur hoeft het contract niet aangepast te worden. Het contract kan 1-op-1 ingelezen worden door de ebMS adapter.

INFRA 2

Het ebMS contract zoals dat op voor het publieke domein gepubliceerd wordt bevat URL A en de public keys van de client en server certificaten. (Eventueel ook het certificaat voor de beveiliging van de payload.)

Dit contract *kan niet 1-op-1 worden ingelezen* door de ebMS adapter voor de configuratie om de volgende redenen:

- De ebMS adapter kan alleen berichten sturen naar de SSL proxy. De URL van de SSL proxy is niet bekend in het contract (een 'interne aangelegenheid'), terwijl die wel nodig is om de ebMS adapter mee te configureren op basis van het contract. (NB: als de SSL proxy transparant zou zijn voor het berichtenverkeer, dan hoeft de URL *niet* aangepast te worden; het volgende punt blijft evenwel van kracht.)
- de ebMS adapter heeft geen private keys van de certificaten die in het contract genoemd worden: de ebMS adapter kan het contract zoals dat voor het publieke domein gepubliceerd is niet accepteren.

De volgende acties zullen moeten worden doorgevoerd:

1. Omdat de verbinding van de ebMS adapter naar de SSL proxy alleen HTTP verkeer is, kunnen de certificaten van de partner niet in het contract blijven staan. Verwijder daarom alle certificaat gegevens (en gerelateerde xml structuren) uit het contract. (Met uitzondering van de certificaten voor de payload beveiliging, indien deze functionaliteit gebruikt wordt.)
2. Vervang in het contract de transport URL van de *partner* door de URL van de SSL proxy: de ebMS adapter zal de berichten willen versturen naar de andere partner en zal door de contract wijziging het adres van de SSL proxy gebruiken.
3. Configureer de SSL proxy zodat berichten die bestemd zijn voor de partner, gerouteerd worden naar de URL zoals die in het oorspronkelijke contract stond. De SSL proxy zal op basis van de interne URL moeten 'weten' welke 'externe' URL gebruikt moet worden. Meestal vindt hier een vorm van pattern matching en substitutie plaats.

De ebMS adapter kan nu wel geconfigureerd worden met het gewijzigde contract. Als de ebMS adapter een bericht gaat versturen, zal deze kijken naar de URL van de partner: dus de URL van de SSL proxy! Voor de oplettende lezer: dat is **niet** URL A (!).

Voorbeeld

De situatie voor de organisatie is als volgt:

- De organisatie heeft als endpoint de URL: <https://www.urlA.nl/29426453>
- De SSL proxy heeft de interne URL: <http://sslproxy.org-x.nl/ingaand>

De nieuwe URL voor het contract wordt:

- <http://sslproxy.org-x.nl/ingaand/29426453>

Als de ebMS adapter een bericht naar de partner stuurt, zal deze de nieuwe URL gebruiken: het

bericht gaat naar de SSL proxy. De SSL proxy zal de ingaande URL mappen op de oorspronkelijke URL van de partner:

- Mapping: <http://sslproxy.org-x.nl/ingaand/29426453> naar <https://www.urlA.nl/29426453>

Tevens zal de SSL proxy de HTTP verbinding opzetten met TLS/SSL en client authenticatie.

Einde voorbeeld

INFRA 3

Het ebMS contract zoals dat wordt gepubliceerd bevat URL A en de public keys van de client en server certificaten. Dit contract *kan niet 1-op-1 worden ingelezen* door de ebMS adapter voor de configuratie, om de volgende redenen:

- de ebMS adapter alleen berichten kan sturen naar de intermediair. De URL van de intermediair is niet bekend in het contract (een 'interne aangelegenheid'), terwijl die wel nodig is om de ebMS adapter mee te configureren op basis van dat contract.
- de ebMS adapter geen private keys heeft van de certificaten die in het contract genoemd worden: de ebMS adapter kan het contract dus niet accepteren.

De volgende acties zullen moeten worden doorgevoerd:

1. Omdat de verbinding van de ebMS adapter naar de intermediair alleen HTTP verkeer is, kunnen de certificaten van de partner niet in het contract blijven staan. Verwijder daarom alle certificaat gegevens uit het contract en verander de HTTPS in de endpoint URL door HTTP. (Met uitzondering van de certificaten voor de payload beveiliging, indien deze functionaliteit gebruikt wordt.)
2. Vervang in het contract de transport URL van de *partner* door de URL van de intermediair: de ebMS adapter zal de berichten willen sturen naar de andere partner en gebruikt daarvoor de URL van de *andere partner* zoals die in het contract staat.
3. Configureer de intermediair zodat op basis van de partyId de url van de partner gebruikt wordt voor het doorsturen van het bericht. (Merk op dat de typering, te weten het PartyId type attribuut, ook gebruikt zal moeten worden als de partyId zelf niet onderscheidend is.)

De ebMS adapter kan nu wel geconfigureerd worden met het gewijzigde contract. Als de ebMS adapter een bericht wil gaan versturen, zal deze kijken naar de URL van de partner: feitelijk de url van de intermediair! Voor de oplettende lezer: dat is **niet** URL A (!) maar een interne url, te weten URL B.