WG: Multiple vendor

Status: historical. Notes and proposals from 2023, based on Nuts spec v1. Kept for reference and no longer maintained.

In English. This workgroup will resolve challenges that are related to multiple vendors providing services to a single care organisation.

Nuts Slack: #werkgroep-multiple-vendor

- Problem definition
- Meeting notes
  - 2023-01-16 Scope
  - 2023-02-08 Trust & functional brainstorm
  - 2023-02-14 Technical team discussion
  - 2023-03-16 Working group periodic update
- Design
  - First proposal
- eOverdracht data flow management issue
  - Multiple systems on the sender's side
  - Multiple systems on the receiver's side
- Appendix A. Overdracht bij gebruik meerdere systemen (by Michiel Bruins, Dutch version)
  - Probleem
  - Oplossingsrichtingen
  - 1) Cliëntadministratie als centrale spil
  - 2) Stuur de Verpleegkundige overdracht naar alle Vendors
  - 3) Stuur de verpleegkundige Overdracht naar één partij
  - Combinaties
  - Aanvullend: verzenden van de Verpleegkundige overdracht

    - Kwalificatie

- Appendix B. Overdracht for multiple vendors (by Michiel Bruins, English version)
    - Problem
    - Solution directions
    - 1) Client administration as the central pivot
    - 2) Send the Overdracht to all Vendors
    - 3) Send the Overdracht to one party
    - Combinations
    - Additionally: sending the Overdracht
    - Qualification

Problem definition

A healthcare organisation usually has a multitude of vendors providing digital services. In some cases a use-case requires multiple systems. It could also be the case that certain data could be usefull to display in different systems. Without support for handling this, a healthcare organisation would probably need an integrator for certain use-cases.

Legislation dictates that the healthcare organisation is responsible for controlling access to patient data. In use-cases, that organisation grants access to other organisations and not to individual systems. For example: when organisation "A" grants access to "B" for a patient regarding medication data, than any system from "B" should be able to fetch the medication data. It should also be possible for "B" to fetch this data from any system of "A", not only the system that granted the access.

Within the scope of the Nuts specifications (V1) this is a challenge because the Verifiable Credentials (VC) use the Decentralized Identifier (DID) as technical ID for *issuer* and *subject*. Because DIDs are bound to a private key, these credentials can't be shared accross systems.

Meeting notes

Meeting notes

2023-01-16 Scope

This meeting was scheduled to determine the scope of the challenge and to make sure it was covered from multiple angles. 4 different vendors attended the meeting.

Resources

Video available @ youtube

Presentation available

Conclusion

The following points were concluded from the meeting:

- An authorization must be usable by any node/app from the organization which is authorized.
- Any node/app of an organization must accept this authorization
- Authorizations must only be issued to nodes that support the use-case
- Nodes/apps that issue authorizations must be authorized by the organization to do so for a use-case
- Authorizations must be automatically issued/revoked/sent when the topology changes (new nodes, nodes removed)
- The trust between the authorization issuer and authorization verifier within an organization requires extra care

Addtional notes

Currently the Nictiz qualification process can only qualify a pair of vendors/applications where each party covers the entire side of a use-case. Eg 1 side covers sending and the other receiving. It was concluded that to support multiple vendors properly, a combination of vendors at one side must be able to qualify as well.

Meeting notes

2023-02-08 Trust & functional brainstorm

This live meeting was scheduled to get a first sense of direction. 2 subjects were on the agenda: how can one vendor trust the authorizations of another vendor within the same organisation context? And, what feature would allow authorizations to be used by multiple vendors.

Trust

The general direction that was decided on is to have a default level of trust between vendors. Vendors qualify their application and the care organisation only uses qualified applications. If a care organisation uses two vendors for a use case and both are qualified then those vendors automatically trust eachother within the context of that use case. Vendors will not require additional configuration or trust regarding that use case. This does mean that the care organisation has to authorize a vendor for a certain use case.

Functional solution

The main direction of the discussion is towards adding a functional identifier to an organisation credential and then use that identifier in the credential subject of a verifiable credential.

This changes the flow for requesting an access token. The requester will have to send both the authorization credentials and the organisation credential in the request. The combination of both credentials would satisfy the security constraints.

The issuer of the authorization credential will also have to monitor the network for any new organisation identifiers that match. When noticed, the DID that has received that functional identifier as subject will also have to receive an authorization credential.

Using a combination of organisation credential and authorization credential would allow the credential holder to use the authorization credential at a different verifier than the issuer. The verifier can validate the given credentials. How the verifier can check that one of its customers has the same functional identifier as the issuer is to be researched.

Meeting notes

2023-02-14 Technical team discussion

With the Nuts technical team, consisting of the highest nuts-node code contributors, we discussed on how this feature could be supported. The following points have become clear from the discussion:

- AuthorizationCredentials will still be issued from and to a DID.
- `iss` and `sub` in the OAuth access token (request) will also remain a DID. This might be subject to change when the final design is being made.
- The result from the above two points is that the access token request will not need to contain any additional VCs/VPs.
- An issuer will have to respond to new DIDs that have the same logical identifier as DIDs that have already received an AuthorizatoionCredentials.
- A verifier will have to do a lookup on the issuer of the AuthorizationCredential and match it to a DID it controls via the logical identifier. The final design might also include the logical identifier in the access token request to prevent finding dupicates.

The points above require the following specification changes:

- Introduce the concept of a logical identifier.
- Implementations will need to be able to extract a logical identifier from a specific VC/VP. This probably needs to be configurable.
- Implementations will need to have a authorization registry where logical identifiers are mapped to AuthorizationCredential templates.
- Implementations will need to respond to new logical identifiers detected on the network and issue new AuthorizationCredentials to a DID
- Implementations will need to respond to revocations of identifiers revoke any AuthorizationCredential that has been issued using logical identifiers.
- Introduce the concept of scoping/purpose of use. Possibly enforced by a credential.

Additional implications for the network as a whole:

The working group already agreed on intra-vendor trust if the care organization approves both vendors for a particular use-case. This also means that both vendors are qualified to service that use-case. When looking at the Trust over IP model, this means that a use-case authority could perform the role of issueing a *scopeCredential* (needs better name). The combination of logical identifier and scope would allow nodes to automatically issue and trust AuthorizationCredentials.

Possible roadmap:

1. Place an IRMA signature on a contract using kvk credentials that link the vendor and scope to a logical identifier (kvk number)

2. Create and host *authority* software that issues a credential containing the scope. Care organizations can login on that software using the IRMA kvk credential. Administrators on the software can approve specific vendors for a use-case.
3. LRZA also issues VCs. From the vendor, a care organization official jumps to the *authority* software (with a VP) and logs in using the kvk IRMA credential. It approves the scope and the *authority* software issues a VC to the care organization wallet. This slightly more complicated step is needed so the LRZA can revoke credentials for care organizations that are no longer a care organization.

Different time paths can be assigned to each step.

Meeting notes

2023-03-16 Working group periodic update

Agenda

- Access token request
- templating
- filtering

Access token request

Currently RFC014 had a line in §4.2:

The credential issuer equals the sub field of the JWT in the access token request.

This will probably have to change to *the issuer must be trusted for the use-case*?

Templating

Because VCs ar not directly constructed but are to be constructed in response to *events* we'll need some sort of templating to create the right VCs. This has to be coordinated with efforts on implementing OIDC4VCI.

Filtering

To select the right *events* for VC generation, some sort of filtering language would be required. DIF presentation exchange seems the right candidate.

Design

Examples, proposals and the final design to solve the challenge

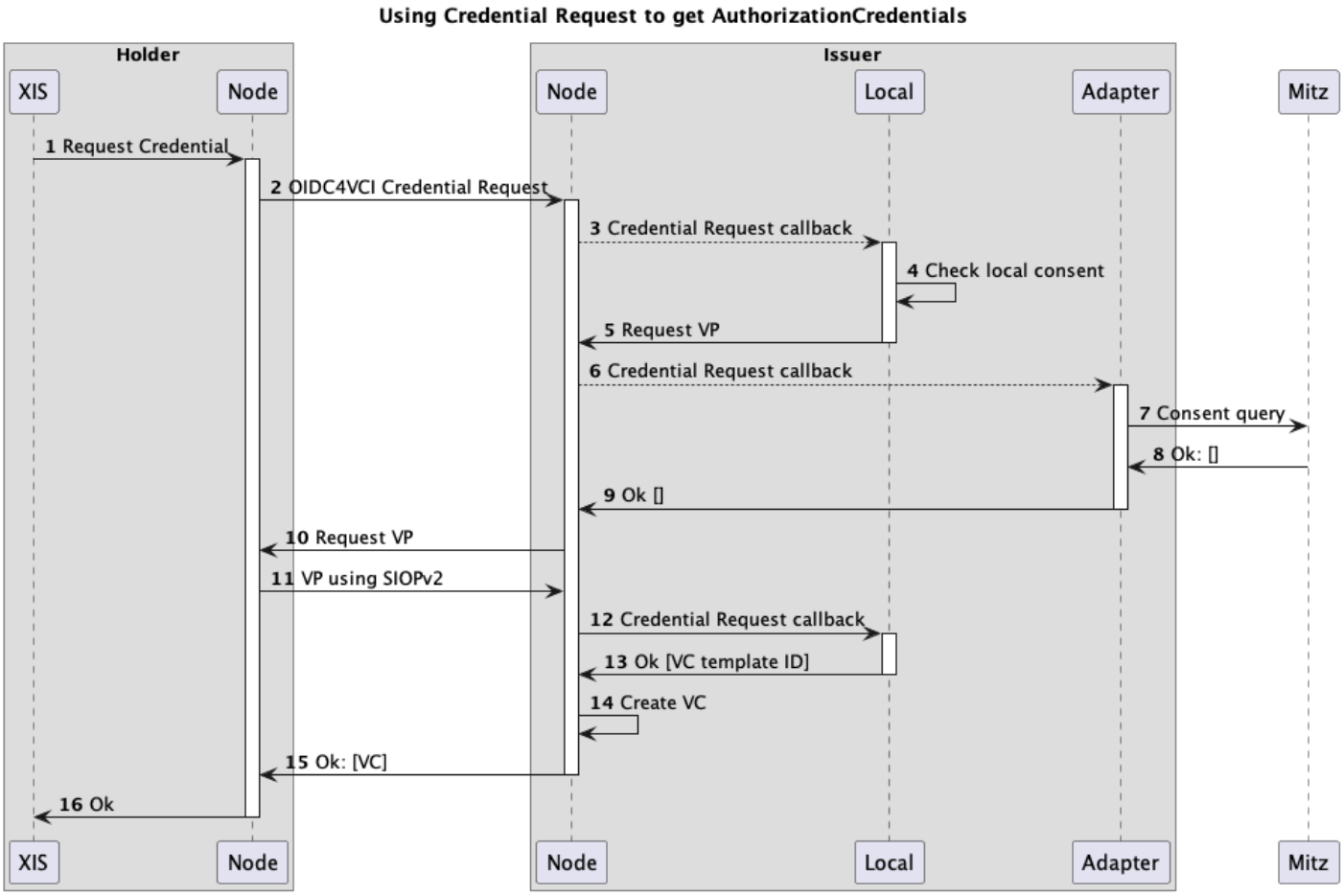
First proposal

This page contains a proposal that could solve the problem. It does not have all the details worked out yet.

Using credential request

The credential request flow is part of the OIDC4VCI specification and will be implemented by the Nuts node. It's also used in some of the other working groups.

The main idea is that when a care organization already knows care is given by another care organization, it can ask that care organization for an authorization. The *other* care organization acts as an credential issuer and will need to respond to the request. The answer to the request depends on any available legal basis and if the requester submitted the correct data.



The flow below shows a basic example.

1. A XIS/ECD sends a credential requests to its Nuts node. The request could look like this for eOverdracht:

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://nuts.nl/credentials/v1"
  ],
  "types": [
    "VerifiableCredential",
    "NutsAuthorizationCredential"
  ],
  "issuer": "did:nuts:123",
  "credentialSubject": {
    "id": "did:nuts:456",
    "purposeOfUse": "eOverdracht",
    "case#": "10475098459138475"
  }
}
```

or this for same organization access:

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://nuts.nl/credentials/v1"
  ],
  "types": [
    "VerifiableCredential",
    "NutsAuthorizationCredential"
  ],
  "issuer": "did:nuts:123",
  "credentialSubject": {
    "id": "did:nuts:456",
    "purposeOfUse": "same-org",
    "patient_identifier": "123456782"
  }
}
```

2. The node will find the correct endpoint and forwards the request to another node. Nodes can act as both a wallet and issuer. The node will embed the request in the `authorization_details` field:

```
[{
   "type":"openid_credential",
   "format": "ldp_vc",
   "credential_definition": {
      "@context": [
         "https://www.w3.org/2018/credentials/v1",
         "https://nuts.nl/credentials/v1"
      ],
      "types": [
         "VerifiableCredential",
         "NutsAuthorizationCredential"
      ],
      "issuer": "did:nuts:123",
      "credentialSubject": {
         "id": "did:nuts:456",
         "purposeOfUse": "eOverdracht",
         "case#": "10475098459138475"
      }
   }
}]
```

3. The received request will be forwarded to any configured *adapter* using the same format as seen in step 2. Multiple adapters can be contacted in this way.
4. The adapter will apply business logic to determine the correct answer. In this example the local consent registry only stores consent based on logical organization identifiers.
5. Due to missing context data, the local adapter will respond with a *400 - not enough data* and a `presentation_definition`:

```
{
  "presentation_definition": {
    "id": "first simple example",
    "input_descriptors": [
      {
        "id": "A specific type of VC",
        "name": "A specific type of VC",
        "purpose": "We want a VC of this type",
        "constraints": {
          "fields": [
            {
              "path": [
```

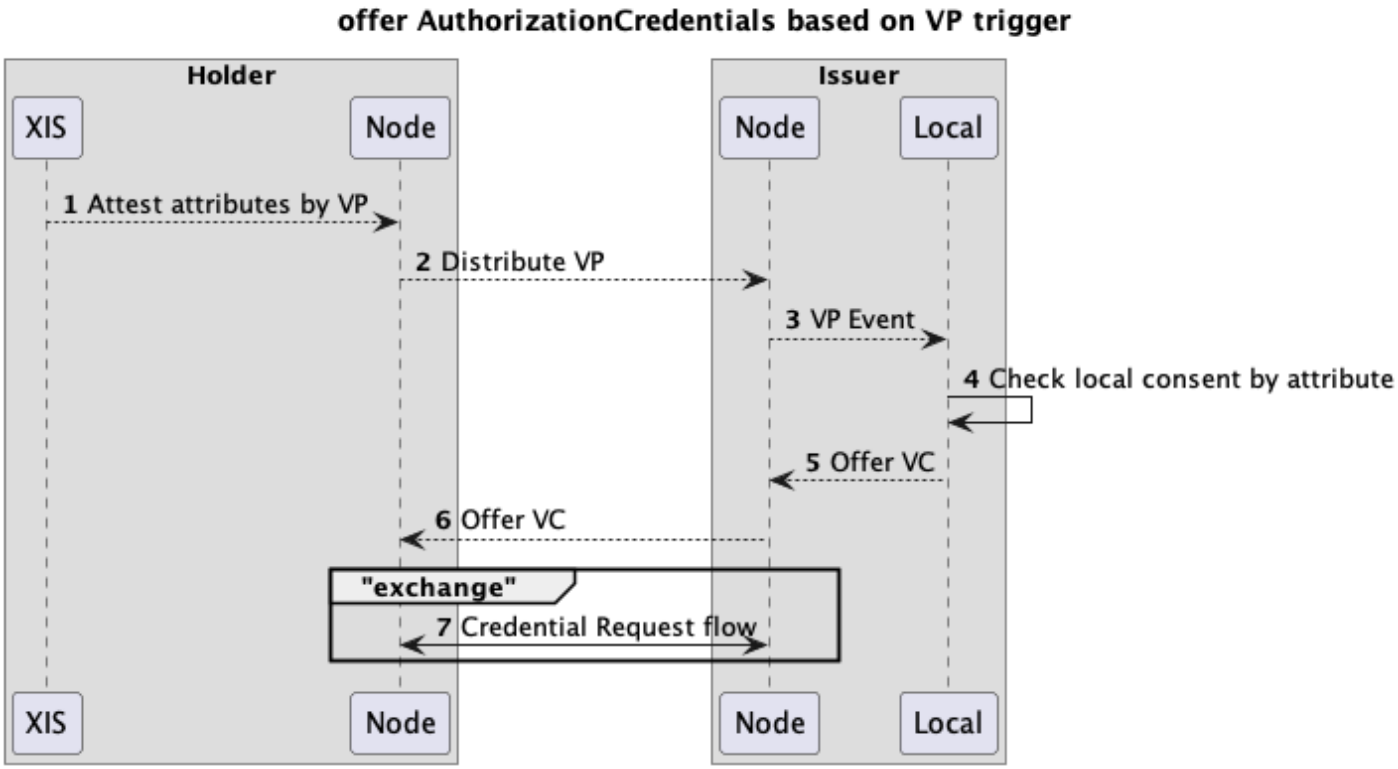
```
                "$.type"
              ],
              "filter": {
                "type": "string",
                "pattern": "NutsOrganizationCredential"
              }
            },
            {
□□□  "path": [
□□□□"$.organization.identifier"
□□□  ],
□□□  "filter": {
□□□□"type": "object",
□□□□"minProperties": 1
□□□  }
□□□},
          ]
        }
      }
    ]
  }
}
```

The presentation definition tells the issuer node it should acquire a verifiable presentation containing a *NutsOrganizationCredential* 6. same as step 3 7. The Mitz adapter contacts Mitz using a FHIR consent query. 8. Nothing registered, empty response. 9. This adapter returns an empty result. No rules to issue a credential on. 10. The presentation definition from the local adapter is used to perform acquire a `Verifiable Presentation` from the requestor using `SIOPc2`. 11. If the requestor holds a credential that matches the presentation definition, it can respond with a verifiable presentation. 12. The issuer node submits the request again, but now with a verifiable presentation in the `vp_token` field. The request now fulfils the requirements of the local adapter. 13. The adapter responds with a template of a credential that may be issued by the issuer node. 14. A credential is created, it's identifier stored. 15. The credential is returned to the wallet of the holder. 16. complete

Using credential/presentation trigger

In the case the holder does not know where to get an authorization, the issuer could send a *credential offer* to the holder based on information the holder publishes. For example: a newly published *NutsOrganizationCredential* could trigger an issuer to offer a credential based on logical identifier.

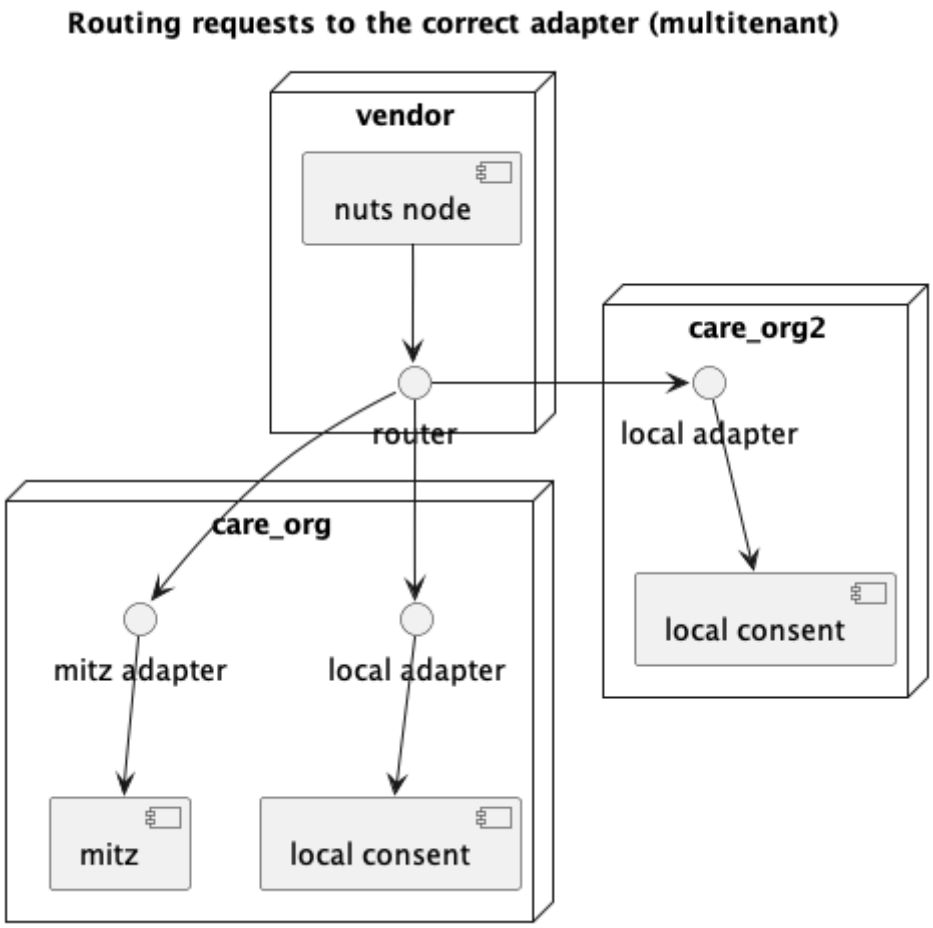
The flow could look like this:



1. A XIS/ECD creates a new wallet in a node and loads a credential. This could be a *NutsOrganizationCredential* or any other credential containing identifying information.
2. The credential is synchronized over the network.
3. The Nuts node of the issuer emits an event for the newly perceived credential.
4. The local consent store listens to these type of events and receives the published credential. The logical identifier is used to query its local DB on *active* consent records.
5. For all records that match an offer is sent through the Nuts node (batching is possible).
6. The Nuts node gets the correct endpoint and forwards the offer to the Nuts node of the holder.
7. The holder node/wallet will initiate a flow similar to the previous paragraph.

Routing requests to the correct adapter

There's an additional challenge that surfaced: the Nuts node is hosted per vendor, but local and/or other adapters might be specific per care organization. This might mean that some sort of routing is needed from the Nuts node to the actual adapter. The adapter might be from a different vendor than the Nuts node. Any security configuration between node and adapter has to be arranged by the vendors.



The router should be able to map DIDs to adapter endpoints. The only reason for not having a router is when all adapters, the wallet, local DBs and node are managed by the same vendor. Since this would restrict the freedom of choice, we'll have to assume this is not the case.

eOverdracht data flow management issue

As mentioned on the Problem definition page, healthcare organizations usually have multiple software providers for different purposes. The data flow management issue arises when trying to transfer data through the Nuts network between organizations whilst at least one of them uses multiple systems (multiple Nuts nodes).

Multiple systems on the sender's side

The following challenges rise when a customer makes use of multiple software systems that own different pieces of patient's data and wants to send data through eOverdracht:

1. The customer usually wants to send all the patient's data in one eOverdracht session instead of starting multiple eOverdrachts in all systems.
2. The customer usually wants to send different combinations of data from different systems depending on what services the patient gets from the customer (e.g. inpatients and outpatients may receive different services, or some of the care patients receive treatment and some of them don't).

Multiple systems on the receiver's side

The Nuts Network is vendor-based. Each vendor's system has its own Nuts Node connected to others within the Network. But customers are used to work in terms of organizations, subdivisions, and locations. So when a customer wants to send patient's data through eOverdracht to another care organization that use software systems of multiple vendors, they face some difficulties.

For example, for the customer it is uncertain to which of the multiple receiver's systems they should send patient's data. Because for them it is not clear what data each of the systems owns. It is also easy in this case to send the data to a wrong system.

Another example is when the customer wants to send data to a specific location or subdivision within the receiver organization. Each location/subdivision may use one or more systems that makes the sending process more complicated.

Appendix A. Overdracht bij gebruik meerdere systemen (by Michiel Bruins, Dutch version)

Bronnen:
* Nictiz Architectuur https://informatiestandaarden.nictiz.nl/wiki/vpk:V4.0_FHIR_eOverdracht)
* Nuts Bolt: https://nuts-foundation.gitbook.io/bolts/eoverdracht/leveranciersspecificatie

Probleem

In de praktijk worden door de meeste zorginstellingen in een aantal sectoren meer dan één leverancier of systeem actief bij de behandeling. De architectuur voor de Verpleegkundige Overdracht gaat echter wel uit van maar één zender en één ontvanger.

Een voorbeeld hiervan is de ouderenzorg waar vaak de cliëntadministratie, zorgdossier, behandeldossier, Apotheek dossier e.a. naast elkaar worden gebruikt bij de behandeling van een cliënt. Deze systemen wisselen onderling ook gegevens uit, maar het landschap is heel divers en verschilt per zorginstelling.

Een opmerking over terminologie: er worden meerdere begrippen gebruikt afhankelijk van de bron: waar Nictiz het heeft over een XIS, wordt in Nuts gesproken over een Vendor, en in overleg vaak benoemd als Leverancier. In dit stuk volg ik de Nuts terminologie: dus een 'Vendor' voor leverancier/XIS, een 'Customer' voor Organisatie/zorgaanbieder, etc. De Verpleegkundige Overdracht - ook wel bekend als eOverdracht - kort ik af tot Overdracht.

Oplossingsrichtingen

In de breakout sessie zijn twee mogelijkheden kort besproken maar niet in detail uitgewerkt. Ik beschrijf hier enkele richtingen met voor en nadelen, maar dit is zeker geen compleet beslissingsdocument. Het is bedoeld als startpunt voor verdere discussie over oplossingen.

1) Cliëntadministratie als centrale spil

Het idee is dat de Overdracht altijd gestuurd naar de Vendor waar de administratie wordt gedaan: de cliëntadministratie (of transferverpleegkundige e.a.). Deze Vendor maakt vervolgens een nieuwe Overdracht Task aan voor de overige systemen waarmee de hele ‘nursingHandoff’ set van resources 1-op-1 wordt doorgestuurd. Voor dit ‘intern’ doorsturen kan de shortcut in het proces worden gebruikt waarbij de Task gelijk op status ‘in-progress’ wordt gezet en de negotiate-phase (‘requested’, ‘accepted’ etc.) kan worden overgeslagen.

Voordelen:

- De verzendende partij ziet in het Nuts Adresboek maar 1 ‘eOverdracht-receiver’ Service bij één Vendor per Customer.
- Het verspreiden van de gegevens kan via dezelfde architectuur en software plaatsvinden (gewoon een nuts Overdracht).

Nadelen/vereisten:

- De cliëntadministratie moet alle resources zoals ontvangen als resource weer aanbieden - ook de gegevens die niet zelf ondersteund/gebruikt worden.
- De Vendor die de Overdracht ontvangt moet kennis hebben van de gebruikte systemen binnen de zorgorganisatie en de Overdracht naar de juiste leveranciers doorsturen. Een uitdaging hierbij is hoe deze te vinden in het adresboek zonder dat deze door de gebruikers van de verzender getoond worden: zijn dit een ander type services? ‘eOverdracht-receiver-intern’ bijvoorbeeld?
- Alle cliënten in de zorginstelling moeten dan bekend zijn in één cliëntadministratie. Dit compliceert situaties waar bijvoorbeeld voor revalidatiezorg of wijkverpleging een ander administratief systeem wordt gebruikt: mogelijk moeten er dan toch meerdere services voor Customers worden geconfigureerd. Bijvoorbeeld: ‘<zorgorganizatie>-intramuraal, ‘<zorgorganizatie>-wijkverpleging’, ‘<zorgorganizatie>-revalidatie’. Dit maakt het natuurlijk wel weer lastiger voor de verzender om de juiste Customer te kiezen.
- De aanvullende Vendors - naast de cliëntadministratie - zullen nog wel een gebruiker moeten hebben die inlogt en zich autoriseert via Irma om de gegevens op te kunnen halen. Dus een (minimale) manuele handeling is wel vereist.
- Het gebruik en de werkafspraken binnen de organisatie wordt complexer.
- Data moet meer dan eens worden verwerkt.

2) Stuur de Verpleegkundige overdracht naar alle Vendors

Een aanvulling op de Nuts Bolt proces waarbij als dezelfde Customer bij meerdere Vendors is geregistreerd, er voor elke ‘eOverdracht-Receiver’ service bij elke Vendor een Task wordt gemaakt, dus meerdere taken per customer.

Voordelen:

- De ontvangende Vendors hebben geen kennis nodig van welke andere systemen gebruikt worden: elk van de Vendors neemt de resources over die voor hun rol in de behandeling relevant zijn: geen Customer specifieke configuratie nodig.
- Er is geen centrale cliëntenadministratie nodig waar alle patiënten bekend zijn, hoewel dan de gebruikers van de verschillende Vendors/systemen dan wel moeten weten of ze de cliënt wel of niet moeten accepteren. Meerdere Customers configureren zoals bij 1) blijft verder ook een optie.
- Het vereist voor zover ik kan zien geen aanpassing aan de Nictiz architectuur, alleen het proces / afspraken binnen Nuts.

Nadelen:

- Dit werkt alleen als alle verzendende partijen dit ondersteunen. Dus het principe dat er meer dan 1 ‘eOverdracht-Receiver’ Service kan zijn voor een Customer in het adresboek, en als dit het geval is dat er voor elke Service een eigen Overdracht gedaan moet worden moet dan wel onderdeel worden van het afsprakenstelsel.
- Voor de verzender voegt het de complexiteit toe als de Task status bij verschillende Vendors niet dezelfde status heeft, bijvoorbeeld dat 1 Vendor het accepteert, en de andere niet.
- Elke ‘Vendor’ in het Nuts adresboek definieert de eigen lijst van ‘Customers’. Er moet dan een label/identifier zijn in het adresboek waardoor de verzender kan zien dat de Customer bij vendor A dezelfde is als de Customer bij Vendor B, en deze Customer in de zoekresultaten voor de gebruiker maar één keer tonen.

Opmerking: een variatie hierop is het idee om één Task te maken en een notificatie te sturen naar de verschillende Vendors. Maar ik denk dat dit problematisch is wat betreft het accepteren en verwerken van het dossier: welk van de Vendors accepteert de Task dan? De Nuts architectuur en de Nuts autorisatie tokens staan dit verder wel toe. Als dit met werkprocessen binnen de Customer kan worden opgelost is dit een richting die ook verder onderzocht kan worden.

3) Stuur de verpleegkundige Overdracht naar één partij

Deze optie is in de breakout sessie niet meer besproken. Los het probleem niet op met Nuts, maar via het huidige werkproces. Het is een optie om het aan de werkprocessen binnen de zorginstelling over te laten. Omdat de meeste leveranciers van plan zijn de resources zo-ie-zo op te slaan, ook als de data niet past bij het eigen systeem is de data dan wel beschikbaar voor de medewerkers van die zorginstelling. Ook wordt een deel van de gegevens nu al via reguliere datakoppelingen uitgewisseld. Alleen de gegevens die niet door het ontvangende systeem worden verwerkt en doorgestuurd via de reguliere koppelingen, maar wel relevant zijn voor andere systemen zouden dan met de hand moeten worden overgenomen.

Voordeel:

- Huidige uitgangssituatie.
- Eén leidend systeem en simpele adressering.

Nadelen:

- Leidende systeem krijgt een regierol in verwerking van data door derden. Dit is niet in lijn met het gedachtengoed van Actiz wat betreft professionele omgeving.
- Zibs die door het leidende systeem niet worden ondersteund kunnen ook niet worden doorgegeven.

Dit is niet een echte oplossing, maar zou wel als een eerste pragmatische stap gebruikt: dit is nog altijd een grote besparing op het administratieve proces ten opzichte van het huidige situatie.

Combinaties

Overigens sluiten oplossingsrichting 1, 2 en 3 elkaar niet uit: ze zouden naast elkaar kunnen bestaan zonder elkaar in de weg te zitten: de service configuratie in het adresboek, en configuratie bij de Customer zelf bepaalt welke route wordt gevolgd.

Aanvullend: verzenden van de Verpleegkundige overdracht

Bij de besproken richtingen in de breakout sessie lag de focus bij het ontvangen van Overdrachten. Maar bij het gebruik van meerdere leveranciers is natuurlijk ook het versturen van een Overdracht een uitdaging.

Als een Overdracht centraal vanuit de Cliëntadministratie wordt verstuurd wil je ook resources beschikbaar maken van gegevens die in andere systemen staan geregistreerd. Bijvoorbeeld medicatie uit de apotheek, behandel aanwijzingen uit het behandeldossier of hulpmiddelen uit Infozorg. Hier moet dan ook een oplossing voor worden gezocht.

Overigens is het binnen de huidige architectuur en afspraken stelsel ook mogelijk om vanuit de verschillende Vendors elk een eigen Overdracht te doen. Als de ontvangende partij om kan gaan met het ontvangen van een Overdracht van een dossier dat al bestaat, en dan de ontvangen resources verwerkt heb je uiteindelijk weer een compleet dossier in de verschillende systemen. Dit is mogelijk gedachtengoed voor toekomstig overleg.

Kwalificatie

Optie 1 en 2 zijn niet te realiseren binnen de huidige tijdlijnen van de inzicht regeling en is ook geen onderdeel van de kwalificatie door Nictiz.

Appendix B. Overdracht for multiple vendors (by Michiel Bruins, English version)

Sources
* Nictiz Architectuur https://informatiestandaarden.nictiz.nl/wiki/vpk:V4.0_FHIR_eOverdracht)
* Nuts Bolt: https://nuts-foundation.gitbook.io/bolts/eoverdracht/leveranciersspecificatie

Problem

In practice, most healthcare institutions in a number of sectors have more than one supplier or system active in the treatment. However, the architecture for the Verpleegkundige Overdracht is based on only one sender and one receiver.

An example of this is elderly care, where the client administration, care dossier, treatment dossier, pharmacy dossier, etc. are often used side by side when treating a client. These systems also exchange data with each other, but the landscape is very diverse and differs per healthcare institution.

A note about terminology: several terms are used depending on the source: where Nictiz talks about an XIS, Nuts talks about a Vendor, and is often referred to as a Supplier in consultation. In this piece I follow the Nuts terminology: so a 'Vendor' for supplier/XIS, a 'Customer' for Organisation/care provider, etc. I abbreviate the Verpleegkundige Overdracht - also known as eOverdracht - to Overdracht (Transfer).

Solution directions

In the breakout session, two options were briefly discussed but not elaborated in detail. I describe here some directions with pros and cons, but this is by no means a complete decision document. It is intended as a starting point for further discussion of solutions.

1) Client administration as the central pivot

The idea is that the Overdracht is always sent to the Vendor where the administration is done: the client administration (or transfer nurse and others). This Vendor then creates a new Overdracht Task for the other systems with which the entire 'nursingHandoff' set of resources is forwarded 1-to-1. The shortcut in the process can be used for this 'internal' forwarding, whereby the Task is immediately set to status 'in-progress' and the negotiate phase ('requested', 'accepted', etc.) can be skipped.

Advantages:

- The sending party only sees one 'eOverdracht-receiver' Service at one Vendor per Customer in the Nuts Address Book.
- The distribution of the data can be done through the same architecture and software (Nuts eOverdracht).

Disadvantages/Requirements:

- The client administration Vendor must again provide all resources to other systems as received - also the data that is not supported/used itself by the client administration.
- The client administration Vendor must have knowledge of the systems used within the healthcare organization and route the Overdracht to the appropriate suppliers. A challenge here is how to find them in the address book without being shown to the sender's users: are these a different type of service? ‘eOverdracht-receiver-internal’ for example?
- All clients in the healthcare institution must be known in one client administration. This complicates situations where, for example, a different administrative system is used for rehabilitation care or district nursing: multiple services may still have to be configured for Customers. For example: \‘-intramuraal’, \‘-wijkverpleging’, \‘-revalidatie’. This of course makes it more difficult for the sender to choose the right Customer.
- The additional Vendors - in addition to the client administration one - will still need a user who logs in and authorizes himself via Irma in order to retrieve the data. So a (minimal) manual action is required.
- The use and working arrangements within the organization are becoming more complex.
- Data needs to be processed more than once.

2) Send the Overdracht to all Vendors

An addition to the Nuts Bolt process where if the same Customer is registered with multiple Vendors, a Task is created for each 'eOverdracht-Receiver' service at each Vendor, i.e. multiple tasks per customer.

Advantages:

- The receiving Vendors do not need knowledge of which other systems are used: each of the Vendors takes over the resources relevant to their role in the treatment: no Customer specific configuration required.
- There is no need for a central client administration where all patients are known, although the users of the various Vendors/systems then have to know whether or not to accept the client. Configuring multiple Customers as in 1) also remains an option.
- It doesn't require any modification to the Nictiz architecture as far as I can tell, just the process/agreements within Nuts.

Disadvantages:

- This only works if all sending parties support it. So the principle that there can be more than one 'eOverdracht-Receiver' Service for a Customer in the address book, and if this is the case that a separate Overdracht must be made for each Service, must then become part of the agreement system.
- For the sender it adds complexity if the Task status does not have the same status at different Vendors, for example one Vendor accepts it and the other does not.
- Each 'Vendor' in the Nuts address book defines its own list of 'Customers'. There must then be a label/identifier in the address book through which the sender can see that the Customer at Vendor A is the same as the Customer at Vendor B, and only show this Customer once in the search results for the user.

Note: A variation on this is the idea of creating one Task and sending a notification to the different Vendors. But I think this is problematic in terms of accepting and processing the file: which of the Vendors accepts the Task? The Nuts architecture and the Nuts authorization tokens allow this. If this can be solved with work processes within the Customer, this is a direction that can also be further investigated.

3) Send the Overdracht to one party

This option was not discussed again in the breakout session. Do not solve the problem with Nuts, but through the current work process. It is an option to leave it to the work processes within the healthcare institution. Because most suppliers intend to store the resources one way or another, even if the data does not fit their own system, the data will still be available to the employees of that healthcare institution. Also, some of the data is already being exchanged via regular data links. Only the data that is not processed by the receiving system and forwarded via the regular links, but is relevant to other systems, would then have to be transferred manually.

Advantages:

- Current baseline.
- One leading system and simple addressing.

Disadvantages:

- Leading system is given a coordinating role in the processing of data by third parties. This is not in line with Actiz's ideas regarding a professional environment.
- Zibs that are not supported by the leading system cannot be passed on either.

This is not a real solution, but should be used as a first pragmatic step: this is still a big saving on the administrative process compared to the current situation.

Combinations

Incidentally, solution directions 1), 2) and 3) are not mutually exclusive: they could co-exist without getting in each other's way: the service configuration in the address book and the configuration at the Customer itself determine which route is followed.

Additionally: sending the Overdracht

The directions discussed in the breakout session focused on receiving Transfers. But when using multiple suppliers, sending an Overdracht is of course also a challenge.

If an Overdracht is sent centrally from the Client Administration, you also want to make resources available from data registered in other systems. For example, medication from the pharmacy, treatment instructions from the treatment dossier or aids from Infozorg. A solution must therefore be found for this.

Incidentally, within the current architecture and system of agreements, it is also possible for each of the various Vendors to make their own Overdracht. If the receiving party can handle receiving an Overdracht of a dossier that already exists, and then process the received resources, you will eventually have a complete file in the different systems. These may be ideas for future consultation.

Qualification

Options 1 and 2 cannot be realized within the current timelines of the Inzicht scheme and are also not part of the qualification by Nictiz.