**Loading...**



[**VZVZ-dagen 2016**](http://www.bartje.org/wiki/)

**VZVZ-leveranciersdag**

Aanmelden

Bovenkant formulier

Gebruikernaam: 

Wachtwoord: 

CapsLock is on.

Aanmelden

* [Ik ben mijn wachtwoord vergeten](http://www.bartje.org/wiki/tiki-remind_password.php)



Onderkant formulier

[**Versiebeheer met bouwstenen**](http://www.bartje.org/wiki/tiki-index.php?page=Versiebeheer+met+bouwstenen)

**versie beheer**

**Hoe gaan we om met verschillende versies van bouwstenen**.
Dit is niet alleen bij het opvragen, maar ook met het verzenden van 'push'berichten.
Ook is de vraag **hoeveel versies er nodig zijn van bouwstenen**.
Bijvoorbeeld :
t - 1 de oude versie
t - de huidige versie
t + 1 een pilot versie
Of willen we een **baseline** versie (die door iedereen wordt ondersteund)

Ook hebben we te maken met **oude berichten en nieuwe bouwstenen**.
Is de verstrekkingslijst de voorloper van toedienafspraak bouwsteen?
Of is de PS de voorloper van de journaal bouwsteen?
Of naast elkaar (dus oude querie naast nieuwe query)
Of pas generieke query als alles op bouwstenen zit

**Pushverkeer**:
Bijv. In waarneem verslag zit een nieuwe bouwsteen.

* Moet de verzender opvragen welke versie de ontvanger kan ontvangen. (Dat kan met de ZAB)
* Of moeten alle versies van de bouwstenen worden ondersteund? (Dus negegeren wat je niet begrijpt)
* Wat zijn de technische gevolgen hiervan?
* Je zou ook een notificatie kunnen sturen en de informatie laten querien

**Wanneer is een nieuwe versie nodig**

* Moet de ontvanger tegen nieuwe versie kunnen (dus negeren wat hij niet kent)
* Wat als 1 toepassing een nieuwe versie nodig heeft en een andere toepassing dit niet nodig heeft

(bijvoorbeeld e-lab tussen ziekenhuizen en lab voor apotheken, of allergien in ICA, HWG en ziekenhuizen).

* Moeten we de template ID als kenmerk gebruiken en niet de interactie ID.
* Dus nieuwe interactie id als het schema wijzigt en alleen wanneer deze niet backwards compatible is.
* Elke wijziging van het bericht betekent een nieuwe kwalificatie.
* In hoeverre moeten *niet zorg gerelateerde* berichten op een nieuwe versie over (bijv nieuwe wrappers)

**Discussie ronde 1**

**Vragen**

1. Wat zijn bouwstenen en hoe worden ze gebruikt?
2. Wat beschrijft het template en wanneer wordt het template veranderd?

Er is een overkoepeld templateid
master-template met een hierarchy

1. Vraag3: Hoe werkt het versiebeheer
2. Vraag4: Kan er niet een prototype worden gemaakt waarbij er twee versies van dezelfde bouwsteen wordt verstuurd in hetzelfde bericht?
3. Vraag5: Waarom niet altijd backwards compatible versienummer?
4. Vraag 6: Waarom niet ook forward compatibiliteit?

Dit wordt al toegepast bij JGZ, daar worden de onbekende elementen genegeerd.

Bouwsteen per concept? Medicatieafspraak en voorschriften.

In de overgangsfase hoe gaan we daar mee om. oude variant ondersteunen en de vraag 2 keer stellen?
Dit werkt alleen als er met generieke query wordt gewerkt, maar stel dat

Waarom niet de baseline laten afhangen van de langzaamst ontwikkelende partij. Die bepaalt wat de baselineversie wordt.

Versiebeheer navragen bij de bankiers. Hoe doen zij dat?

Vraag
Als we overgaan naar een nieuwe baseline hoe gaat dat?
2 versies ondersteunen door bronsystemen in het migratietraject.

1. Conclusie

baseline pas definieren als iedereen er opzit
de onbekende elementen genegeerd. dit zorgt voor forward compatibiliteit
Overzicht van de technische en functionele verschillen zodat er druk kan worden uitgevoerd bij het overgaan naar een nieuwe versie en zodat het wel ondersteund gaat worden door de leverancier

**Conclusies**

1. Conclusie1
2. Conclusie2
3. Conclusie3

**Argumenten**

1. Argument1
2. Argument2

**Discussie ronde 2**

**Discussiethema Bouwstenen**
Wouter Tesink (inhoudelijk trekker)
Erwin Oord en Joris Smits (vz)
Jacobine Wieggers (notulist)
Sven Berkvens (Zorgdoc), Wim Oudakker (Vanad Enovation), Annemarieke Homan (nictiz), Edwin van Gelder (Promedico), Richard van Adirchem (Centric), Martijn Dijke (CGV/Labelsoft), Andre van Kastelen (KPN), Alex Knoop (Pharmapartners), Erwin Oord (VZVZ)

**Vragen**

1. Vraag1

Hoe omgaan met oude berichten en nieuwe bouwstenen
Wat bedoelen we met bouwstenen? Moet eerst duidelijk zijn wat de definitie van bouwstenen is voordat we over versiebeheer kunnen praten. Dan over meerdere versies van de bouwstenen en hoe daar mee omgaan
Ondersteuning van meerdere versies
Definitie wat is een bouwsteen en definitie wat is een versie, discussie major release, minor release. Implementatiehandleiding moet daarin begeleiden.
Richten op de implementatie van de bouwstenen?
2 definities nodig: Bouwsteen - Versie

1. Vraag2

Wat is een bouwsteen
Nictiz definieert zorginformatiebouwstenen (medisch concept), Registratie aan de bron traject
Vergelijking met atomen en moleculen als bouwstenen
Maar bij ketenzorg gaat het breder, bijv. bij meetwaarden, zijn informatiestructuren
Informatiemodellen worden ook bouwstenen genoemd
Als architect kijk je ernaar als een legoblokje
Ketenzorg: niet berichten definiëren specifiek voor ketenzorg maar inhoud beschrijven als bouwstenen die in andere situaties ook toegepast kunnen worden. Herbruikbaarheid van stukjes informatie. Hoe het er van binnen uitziet maakt niet veel uit.
Bouwstenen hebben 4 niveaus:
Over alle niveaus moet je wat zeggen over versiebeheer.
1. Atomen
2. Moleculen
3. Structuur
4. Container
Creëert ruimt om in het midden transformaties te doen.
In het begin discussie over transformatie
Vervoersmiddelen kan je transformeren zonder aan de inhoud te komen.

FHIR
besluit VZVZ transformatie van HL7 v3 naar FHIR
Op container niveau zou dit moeten kunnen op bouwstenen niveau is dit lastiger.
Als het niet nodig is moet je transformatie voorkomen

1. Vraag3

Definitie van versie?
AORTA releasestructuur: releaseversie, minor en major
Infrastructuur versie
Zorgtoepassing versie
VZVZ bepaald welke versies van zorgtoepassing op welke infrastructuur past
LSP deel v8, Zorgtoepassingen v6
Moeten 2 versies van een major release ondersteunen
Moet backward compatible zijn (bericht dat in 6.12 verstuurd wordt moet ook door een partij op 6.11 ontvangen kunnen worden)
In de praktijk wordt meestal gebruikt dat een nieuwe partij de oude berichten moet versturen.
Bij versies hebben we het over de niveaus: Structuur en container
Nu probleem met versie 6 en versie 8, daar zit alles door elkaar
Niveaus uit elkaar halen
Loskomen van het feit dat het aan elkaar vast zit zodat het makkelijke onafhankelijk van elkaar gewijzigd kunnen worden. Het interactie ID gaat du over de container en de doos.
Container blijft lang hetzelfde (vb zeecontainer) aan de hand van de bouwstenen er in definieer je het bericht.
Transformatie is een organisatorisch probleem.
Versie 6 container is nu heel complex
Deze wordt versimpelt in versie 8, wordt backwards compatible gedaan.
Vraag van de leveranciers: de container moet agnostisch kunnen zijn van wat er in zit.

**Argumenten**

1. Argument1

Straks bij bouwstenen, een systeem kan bijvoorbeeld doos 1-3 uitpakken, 4 en 5 niet, dit is afhankelijk van waar je je voor kwalificeert en wat de leverancier zijn klanten wil bieden.
De container heeft een vrachtbrief, deze staat los van de inhoud (bouwstenen), de vrachtbrief kan ook een bouwsteen zijn. Zodat dit ook makkelijker gewijzigd kan worden.
Het versiebeheer moet een laag dieper.
Als je een vraag stelt kan je aangeven welke versie je wilt.
Push verkeer kan uitgaan van de registers die weet welk systeem welke versies ondersteunen.

1. Argument2

Container met doosjes, elk doosje heeft een eigen versienummer. Waar komt het versienummer te staan, op het doosje en op de vrachtbrief van de container? Kan lastig zijn, creëert complexiteit. Container gaat over hoe iets vervoert moet worden.
Doosjes in de container kunnen ook wat anders bevatten dan XML
1e doosje wat je tegenkomt is de vrachtbrief die vertelt wat de rest is.
De inhoudelijke bouwstenen kan je dan ook encrypt versturen, ook de bouwsteen vrachtbrief.
Op versiebeheer krijg je dan interessante discussies.
Bouwsteen kan van type A, naar B, naar C, hoe omvormen van C naar B naar A.

**Conclusies**

1. Conclusie1

Versiebeheer op 4 niveaus:
1. Atomen
2. Moleculen
3. Structuur
4. Container

1. Conclusie2

Bij versies hebben we het over de niveaus: Structuur en container
Manier van transport loslaten van de inhoud
Vrachtbrief bouwsteen ontwikkelen met wat er verstuurd wordt, losgekoppeld van de container.

1. Conclusie3

Transformatiedienst daar waar mogelijk is.

* [Versiebeheer met bouwstenen](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Versiebeheer_met_bouwstenen)
* [versie beheer](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#versie_beheer)
	+ [Discussie ronde 1](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Discussie_ronde_1)
		- [Vragen](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Vragen)
		- [Conclusies](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Conclusies)
		- [Argumenten](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Argumenten)
	+ [Discussie ronde 2](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Discussie_ronde_2)
		- [Vragen](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Vragen_2)
		- [Argumenten](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Argumenten_2)
		- [Conclusies](http://www.bartje.org/wiki/Versiebeheer%20met%20bouwstenen#Conclusies_2)

Aangedreven door [Tiki Wiki CMS Groupware](http://tiki.org/)  | Thema: Readable