

CoP SA

Informatiedomeinen

Definitie en toepassing

Vastgesteld in CoP SA 17, d.d. 11-12-2024

Versiegeschiedenis

Versie	Datum	Inhoud	Wijziging	Opmerking
0.1	01-05-2024	Eerste concept	Afgestemd met betrokkenen bij de eerste werksessies	
0.2	01-07-2024	Tweede concept	Aanpassingen n.a.v. feedback	
0.3	01-09-2024	Derde concept	Aanvullingen o.b.v. MIM/SBB	
0.4	01-11-2024	Review versie CoP SA	Afbakening en formulering 'MVP'	De toevoeging van 1 implicatie en 1 richtlijn zoals beschreven in dit document zijn vastgesteld in CoP SA 17
0.41	12-12-2024	Uitbreiding n.a.v. CoP SA	Toelichting domein aangescherpt	Vaststelling van versie 0.4 wijzigt hierdoor niet
1.0	22-01-2024	Publicatie	Geen wijzigingen, versie zoals vastgesteld in CoP SA 17	Voor publicatie

Werkgroep Informatiedomeinen

Doel van deze werkgroep: vaststellen van de informatiedomeinen die het vertrekpunt vormen voor de definitie van API's

- De vast te stellen informatiedomeinen dienen als begrenzing voor domein gebaseerde API's.
- Een domein gebaseerde API stelt alleen services binnen één informatiedomein beschikbaar t.b.v. bijv. wendbaarheid, schaalbaarheid en beschikbaarheid.
- De scope van deze informatiedomeinen betreft de gegevensuitwisseling binnen de systeemprocessen en het delen van energiedata confoem het afsprakenstelsel
- De uitgangspunten voor de vast te stellen informatiedomeinen zijn o.a.:
 - NBility
 - Business en Platforms Solutions zoals gedefinieerd door EDSN

[NBility – EDSN.nl](#)

Voorstel nieuwe implicatie en richtlijn t.a.v. de afbakening van API's en de toepassing van informatiedomeinen daarbij

1. Nieuwe Implicatie: Voor het ontsluiten van registers (zowel de sectorale als lokale registers) in de transactionele marktprocessen (systeemprocessen en datadelen) gebruiken we Domein API's (zie definitie).

- Rationale: Het is aan de consumer ('hij die de informatiebehoefte kent') om de informatie bij elkaar te halen. Gebaseerd op het concept van 'late integration' en 'loosely coupled' waarbij de producer zo min mogelijk kennis wil hebben van de use case van de consumer.
- Experience API's zijn daarom in beginsel niet toegestaan in bovenstaande situatie, alleen na een uitspraak binnen de architectuur governance kan een experience API worden toegestaan.
- In het geval van niet-transactionele API's, bijv. API's met een analytisch karakter (die bijvoorbeeld een algoritme ontsluiten of waarvoor een complexe combinatie en verrijking van gegevens noodzakelijk is) is domeinafbakening minder relevant en geldt dit principe niet.

Categorie	Definitie	Scope
System	Deze services worden aangeboden vanuit een applicatie (wel/niet standaard aanwezig) en worden gebruikt voor integraties tussen applicaties en front-ends binnen het domein.	Intern
Domain	Deze services worden standaard aangeboden vanuit een domein (bounded context).	Extern
Experience	Dit type service wordt optioneel aangeboden op basis van behoeften van afnemers. Ze zijn gebaseerd op domein services van één of meer domeinen.	Extern



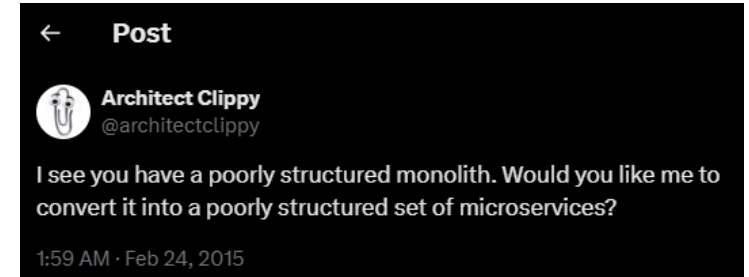
NAP07. Bouw diensten modulair op

IMP030	Zorg voor logische scheiding van datasets
IMP031	Ontwerp op modulaire wijze
IMP083	Wissel gegevens uit met goed gedefinieerde interfaces
IMP087	Verwerk bericht-interactie persistent
NEDU PP3	Enkelvoudige processen: processen kennen enkelvoudige operaties
NEDU PP4	Ontkoppelde processen: processen zijn zelfstandig en worden niet aan andere processen gekoppeld
NEDU JP1	Communicatie is ontkoppeld: de communicatie in marktprocessen is niet afhankelijk van andere processen binnen marktpartijen of de secto
NEDU JP3	Het uitwisselen van gegevens tussen componenten vindt plaats via gedefinieerde interfaces op basis van gegevensdefinities gebaseerd op standaarden
IMP093	Voor het ontsluiten van registers in transactionele marktprocessen maken we gebruik van Domein API's.

Voorstel nieuw principe en richtlijn t.a.v. de afbakening van API's en de toepassing van informatiedomeinen daarbij

2. Nieuwe Richtlijn: De afbakening van domein API's baseren we op de afbakening van de bedrijfsobjecten in NBility c.q. de registers waarlangs deze bedrijfsobjecten worden ontsloten.

- Wanneer in een gegevensuitwisseling, gegevens betreffende meerdere bedrijfsobjecten c.q. registers worden uitgewisseld, worden daar meerdere API's (per bedrijfsobject / register) voor gedefinieerd.
- Refereren naar andere bedrijfsobjecten (in andere registers) is aan de orde en legitiem. Een 'meting' refereert vast naar een aantal zaken. Stel dat een meting refereert naar een installatie (ander domein), dan worden (minimaal) twee API's aangeroepen om één samenhangende set aan gegevens uit deze verschillende registers op te vragen.
- Let op, dit is geen pleidooi voor zgn. entity-based services (zie definitie rechts), aangezien dat als anti-patroon wordt gezien. Goed API of service design vraagt om een goed doordachte afbakening van domeinen.
- *Wanneer een bepaald object niet in NBility is opgenomen, zoeken we een afbakening c.q. granulariteit die in lijn is met die van NBility (zie slide 5 en 6).*
- *Een verdere uitwerking van een sector-informatiemodel is buiten scope van deze richtlijn en pakken we in later instantie op (backlog)*



[Entity Services is an Antipattern - InfoQ](#)

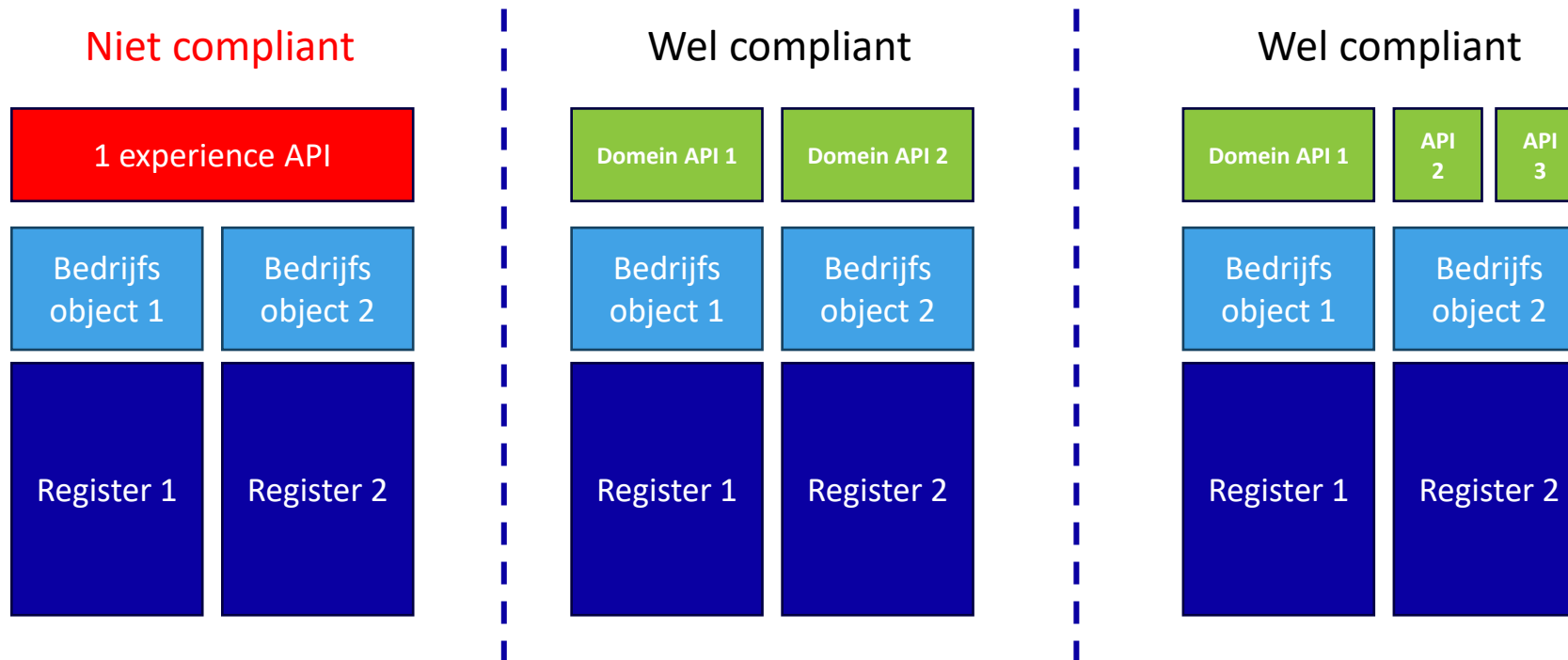
[Entity services: when microservices are worse than monoliths](#)

NAP07. Bouw diensten modulair op

07.12. Processen worden onafhankelijk van andere processen gedefinieerd;
 07.13. Afhankelijkheden worden gestuurd via de data die in de diverse processen wordt gebruikt;
 07.14. Een afhankelijkheid van een ander proces kan worden opgelost door een (data) trigger / event, die het resultaat is van dat proces, zodat de processen onafhankelijk van elkaar kunnen worden uitgevoerd

07.15 De afbakening van domein API's baseren we op de afbakening van de bedrijfsobjecten in NBility c.q. de registers waarlangs deze bedrijfsobjecten worden ontsloten.

Voorbeeld toepassing van deze implicatie en richtlijn (TS055)



Toelichting bij de terminologie

Verschillende raamwerken gebruiken (ongeveer) dezelfde termen. In deze toelichting lichten we toe welke termen we in dit document en in de resulterende principes, implicaties en richtlijnen gebruiken

- EBIX spreekt over business domains, dit zijn feitelijk (hoofd-)processen in het EBIX referentiemodel; in de finale versie van dit document hebben we geen gebruik gemaakt van het EBIX referentiemodel, deze term is dus ook niet van toepassing
- In de definitie van verschillende type API's wordt de term Domein gebruikt als een synoniem voor een 'bounded context'; vaak komt een bounded context overeen met een organisatie-onderdeel. In dit document hebben we wel de verschillende type API's omarmd maar maken we verder geen gebruik van domain driven of micro service architectuur termen zoals bounded context.
- In de hieronder beschreven architectuur richtlijn die hoort bij de architectuur implicatie koppelen we de afbakening (granulariteit) van een Domein in de definitie van een Domein API gelijk aan de (afbakening van) Bedrijfsobject in het NBility referentiemodel.
- NBility maakt gebruik van business-domeinen (niveau 1) en business-subdomeinen (niveau 2) om bedrijfsobjecten (niveau 3) te ordenen. In dit geval ligt de afbakening van Domein API's dus op een lager niveau dan de busines-(sub)domeinen in NBility.
- In de volgende slide is dit aangegeven: business-domein "klant", business-subdomein "klantovereenkomst", bedrijfsobject "klantproduct"

NAP07. Bouw diensten modulair op

07.12. Processen worden onafhankelijk van andere processen gedefinieerd;
07.13. Afhankelijkheden worden gestuurd via de data die in de diverse processen wordt gebruikt;
07.14. Een afhankelijkheid van een ander proces kan worden opgelost door een (data) trigger / event, die het resultaat is van dat proces, zodat de processen onafhankelijk van elkaar kunnen worden uitgevoerd

07.15 De afbakening van domein API's baseren we op de afbakening van de bedrijfsobjecten in NBility c.q. de registers waarlangs deze bedrijfsobjecten worden ontsloten.

mffbas