Nederlandse Notificatie Strategie

4e bijeenkomst berichtenwerkgroep

23 september 2021

Pleio website

Unsplash © Carl Nenzen Loven

Agenda

1. Basis-Berichtformaat
2. Abonneren en filteren
3. Abonneren en verstrekken (push, pull, poll)
4. CloudEvents architectuur
5. CloudEvents JSON format specification
6. CloudEvents HTTP binding specification
7. *CloudEvents Webhook specification en WebSub (in volgend overleg)*
8. Vervolg

Basis-berichtformaat

Laatste openstaande punten om te komen tot eerste werkversie (v0.1)

Overzicht attributen en extensions

Core

REQUIRED attributes
- id
- source
- specversion
- type

OPTIONAL attributes
- datacontenttype
- dataschema
- (subject)
- time

Event Data
- data

Extensions

Extensions
- Dataref (Claim Check Pattern)
- ~~Distributed Tracing~~
- ~~Partitioning~~
- ~~Sampling~~
- Sequence

NL Extensions

- …

ID

	Cloud events specification v1.0.1	Changes
Part of	Cloud events REQUIRED Attributes	
Name	id	
Type	String	• Bijvoorkeur UUID
Description	• Identifies the event. Producers MUST ensure that source + id is unique for each distinct event. • If a duplicate event is re-sent (e.g. due to a network error) it MAY have the same id. • Consumers MAY assume that Events with identical source and id are duplicates.	~~2e en 3e bullet worden nagevraagd.~~ => Geen respons. Voorstel: We nemen onderstaande interpretatie als uitgangspunt: • 2e bullet negeren we, inhoud is te ambigu. • Interpretatie 3e bullet: Consumers MUST assume that Events with identical source and id are duplicates.
Constraints	• REQUIRED • MUST be a non-empty string • MUST be unique within the scope of the producer	• We houden vast aan het voorschrift van de standaard: Het ID is verplicht. • Verplicht betekent dat er ook daadwerkelijk een ID ingevuld moet worden en geen workaround zoals “onbekend”. • Doel van het ID is dat dit daadwerkelijk een event bij de bron identificeert. (Zo zou het ID wellicht gebruikt kunnen worden om bij de bron informatie over het event op te vragen). • Het wordt aanbevolen om ook ‘surrogaat events’ (CRUD of CRUD + eventtype) te persisteren en daarbij van een ID te voorzien. • Indien er geen ID voorhanden is dat het event duurzaam kan identificeren (duurzaam = er kan later nog bij de bron aan gerefereerd worden) dan mag een random ID gebruikt worden. De beperkingen van dit ID (oa. dat het geen zin heeft om er met de bron over te communiceren) moeten duidelijk in de domeinstandaard/documentatie vermeld worden.
Examples	• An event counter maintained by the producer • A UUID	

Source – Format/Inhoud van attribuut

- Keuze voor URN:
  - Format = urn:[nid]:[nns]
  - NID = Namespace Identifier
  - NNS = Namespace-specific string

- Hoe komen we aan een NID?
  1. Misschien kan een van de 60 bestaande gebruikt worden? Een enkele lijkt namelijk generiek.
     - Zijn nagelopen. Geen van de 60 is inzetbaar voor ons doel.
     - Er zijn twee landen met een eigen NID. Deze gebruiken de ISO 3166 landencode (3 karakters).
  2. Registreren
     - Zou eigenlijk elders belegd moeten worden want overheidsbreed inzetbaar. Maar waar?
  3. Niet registreren
     - Veel NID's zijn niet geregistreerd.

- **Voorstel:** Vastlegging is geen prioriteit. Werken met 'NLD'. Community/beheerder kan later besluiten tot registratie.

- Voorbeelden:
  - urn:nld:gemeente-amersfoort
  - urn:nld:rvig - of - urn:nld:rijksdienst-voor-identiteitsgegevens
  - urn:nld:belastingdienst

Type – Format/Inhoud van attribuut

- Reverse-DNS
- Domein hoeft niet echt te bestaan. (Net zoals bijvoorbeeld java.io.File)
- Naam wordt bepaald door het domein, niet door deze generieke standaard.
- Aanbevelingen voor eerste deel van de naam (het domein):
  - Voorkeur
    - Indien voorhanden de naam van een register of bron (BRP, HR, BRK …)
      - nl.brp.verhuizing
  - Eventuele alternatieven
    - Indien geen register voorhanden dan benaming van het domein. Bijvoorbeeld natuurlijke-personen.
      Kiezen we voor enkelvoud of voor meervoud? => Voorstel: We volgen de naamgevingsconventies voor resources vanuit de NL API strategie en kiezen voor meervoud. Let op dit gaat alleen om het domein (dus personen ipv persoon) NIET om het daadwerkelijke type dat is enkelvoud (dus verhuizing ipv verhuizingen).
      - nl.natuurlijke-personen.verhuizing
      - ~~nl.persoon.verhuizing~~ : ambigu, kan natuurlijk of niet-natuurlijk persoon zijn
    - Indien gegevens goed toe te bedelen zijn aan wet- of regelgeving: De naam van de wet- of regelgeving
      - ~~nl.burgerlijkestand.huwelijk~~ : niet handig, de gegevens binnen de brp vallen onder allerlei verschillende stukken regelgeving (burgerlijkstand, paspoortwet, kieswet, emigratie en immigratie regelgeving etc)
      - nl.amsterdam.erfpacht.overdracht zou wellicht wel handig kunnen zijn (aangenomen dat Amsterdam daarvoor eigen regelgeving heeft. Merk op dat Amsterdam hier niet gebruikt wordt als organisatie maar als juridische afbakening – ‘de regelgeving van Amsterdam’)
  - Vermijden
    - Namen van organisaties
      - nl.rvig.verhuizing of nl.bzk.verhuizing

Abonneren en filteren

Eigen filterdialect introduceren?

- Waarom wel:
  - Maakt het mogelijk om te filteren op attributen in de data. Bij opendata is dit een begrijpelijke wens.

- Waarom niet:
  - We willen kunnen hosten bij allerlei cloud providers. Deze providers ondersteunen logischerwijze geen alternatieve filterdialecten (maatwerk). Dat maakt ons afhankelijk van maatwerk infrastructuur. (Uitgangspunt was onafhankelijkheid)
  - Bij informatiearm notificeren hebben we geen data.
  - Bij notificaties over opendata hebben we een alternatief: de te filteren data verplaatsen naar contextattributen.
    - Uitgangspunt was te voorkomen om af te wijken van de CE standaard. Afwijking is in dit geval niet te rechtvaardigen omdat er een alternatief voorhanden is.

- Voorstel: Geen eigen filterdialect introduceren

Filtercriteria in de context

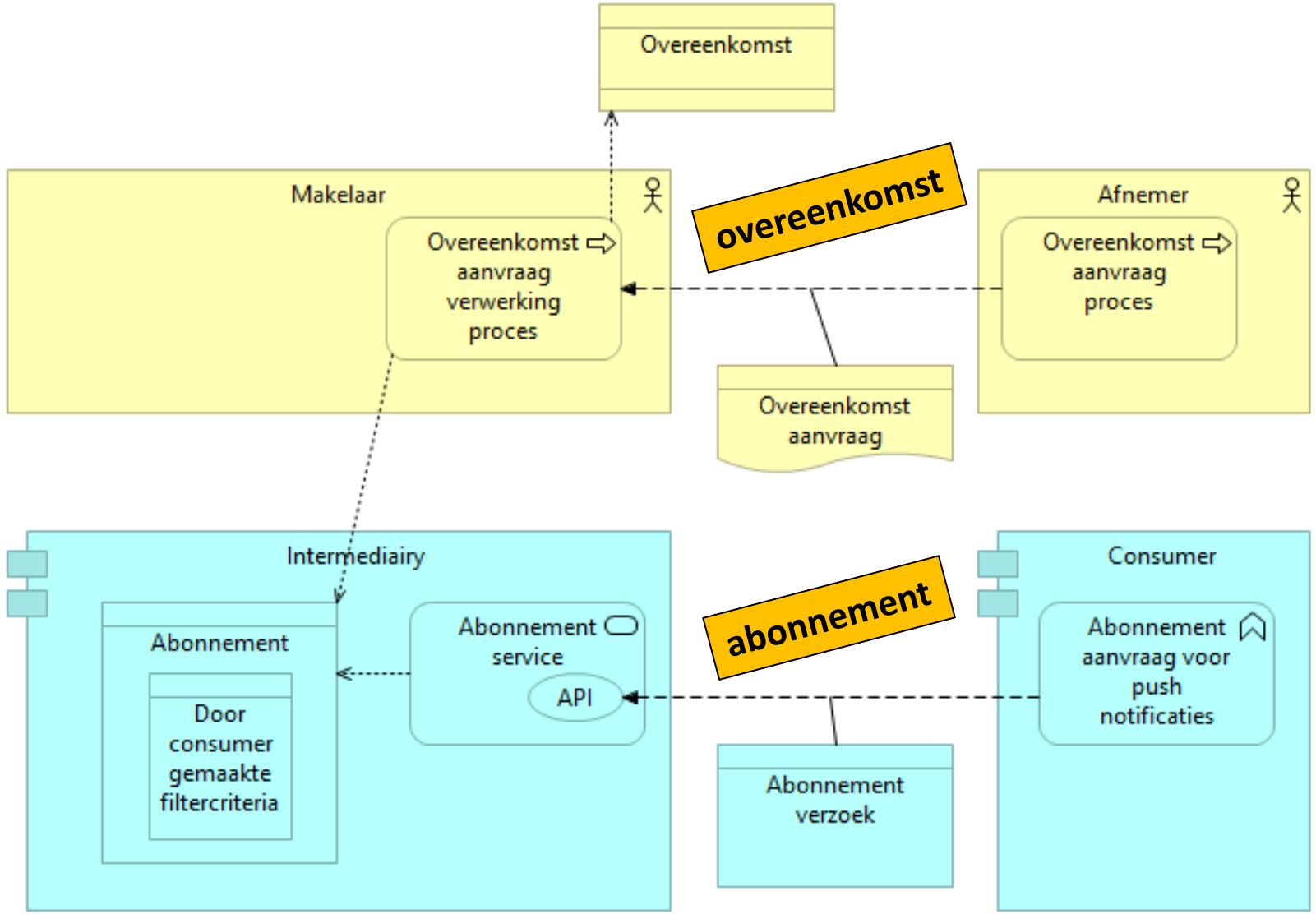
- Opties
  1. Vanuit de NL standaard een x aantal attributen definieren
     - NL-Filterattribuut1, NL-Filterattribuut2 …
     - Vragen om problemen als gevolg van onduidelijke semantiek.
     - Maakt het ‘te makkelijk’ om eventuele privacy gevoelige informatie zonder privacy impact analyse in de context te plaatsen. (Privacy-by-design)
  2. Vanuit de NL-standaard een aantal generieke attributen definieren
     - NL-BSN, NL-Adres, NL-Postcode
     - Semantiek lijkt wellicht duidelijker maar is dat is schijn. Wat als er meerdere betrokkenen met BSN’s? Bedoelen we over alle standaarden heen hetzelfde met ‘adres’? Etc
     - Kan leiden tot een zeer groot aantal attributen.
  3. De domeinspecifieke standaarden definieren zelf attributen indien nodig
     - NL-BSN maar dan vanuit bijvoorbeeld de BRK
     - Domeinen definieren zelf de semantiek op basis van concrete behoeften aan filtering.
  4. Poging om ‘name-value’ attributen binnen de CE standaard te krijgen
     - Gezien de beoordeling van andere voorstellen in de CE community lijkt een voorstel hiertoe weinig kans van slagen te hebben.

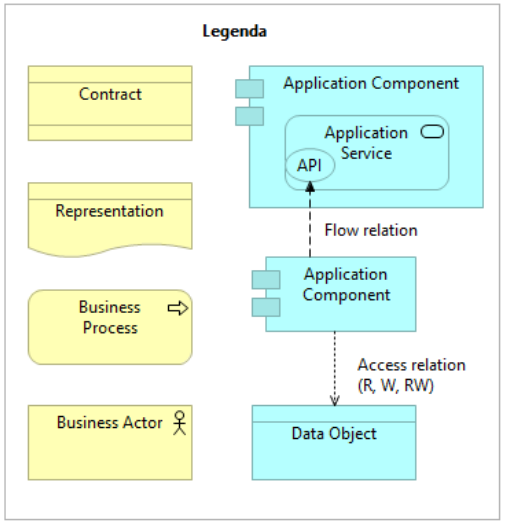
- Voorstel: Optie 3: Domeinen definieren zelf, indien nodig, context attributen.

Neutrale extension attributen

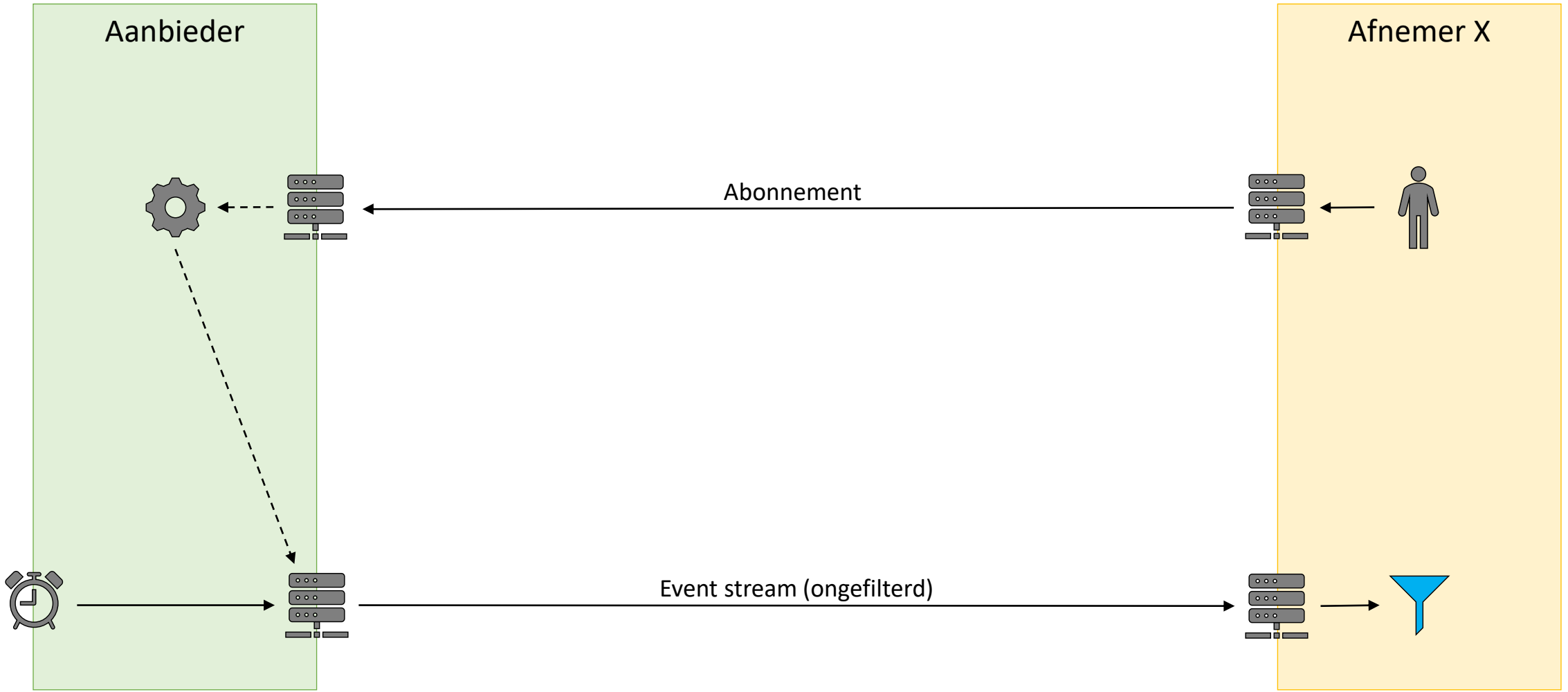
Extension attribuut met JSON name-value pairs

Terminologie: Overeenkomst/Abonneren

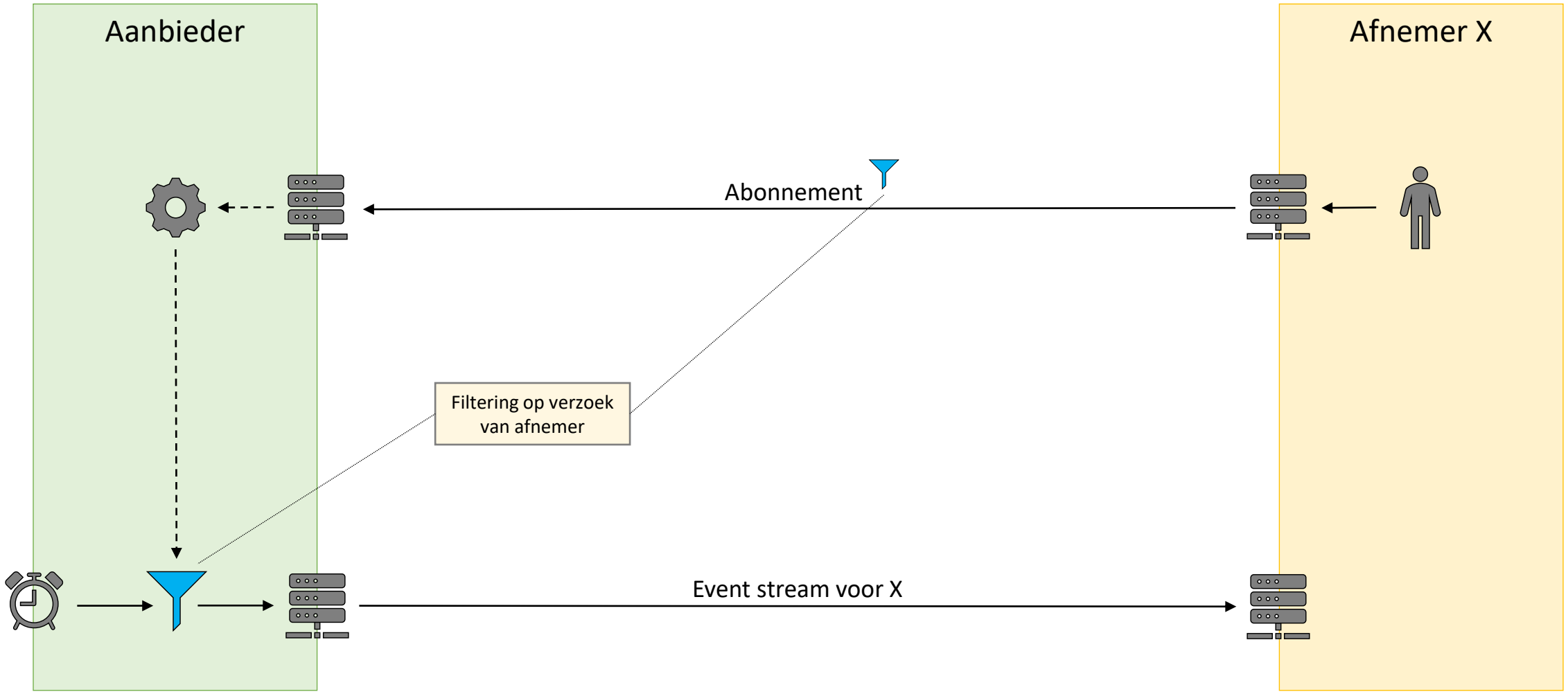




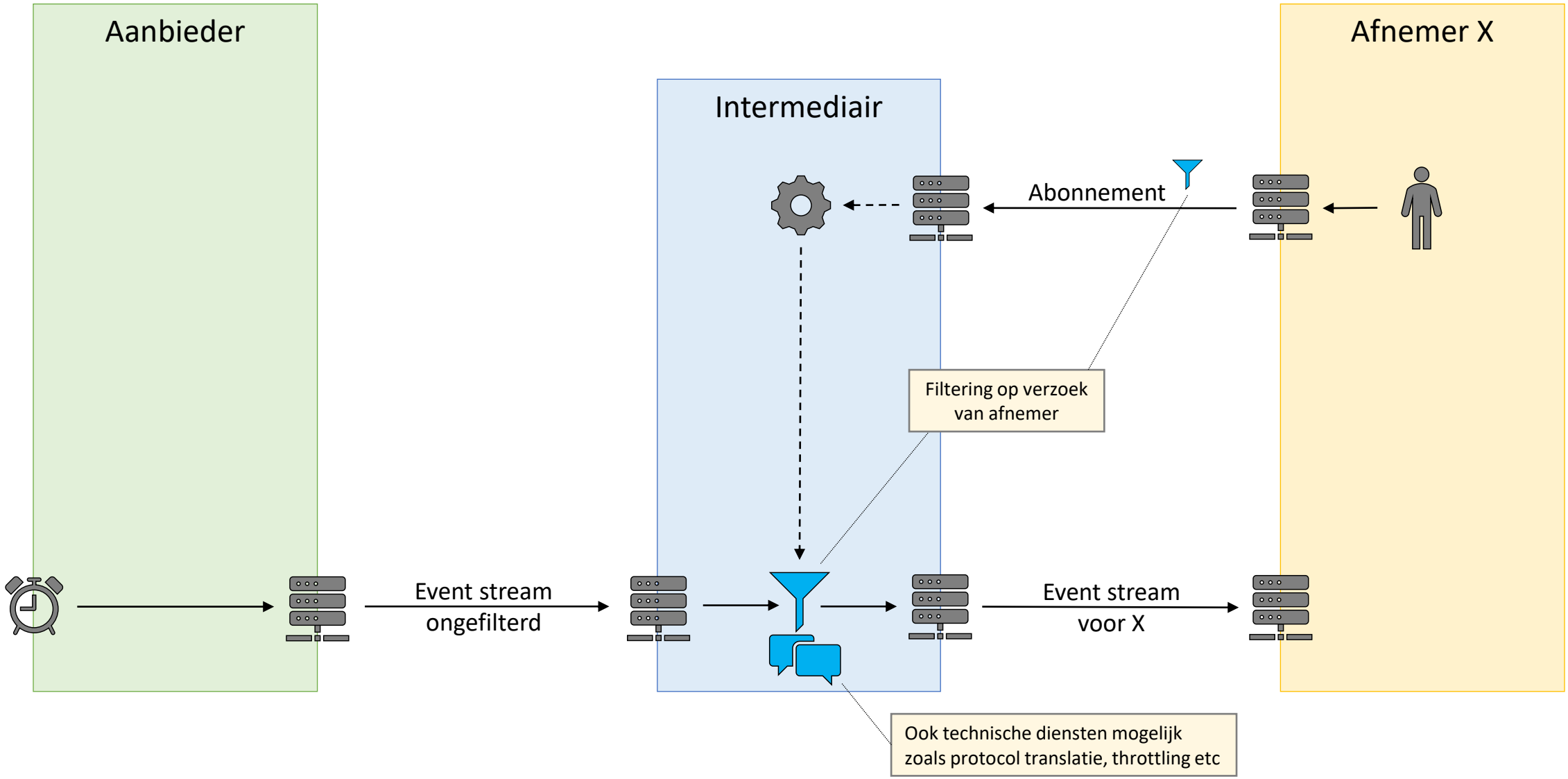
Case: Openbare informatie - filtering door afnemer - geen intermediair



Case: Openbare informatie - filtering door aanbieder voor afnemer - geen intermediair

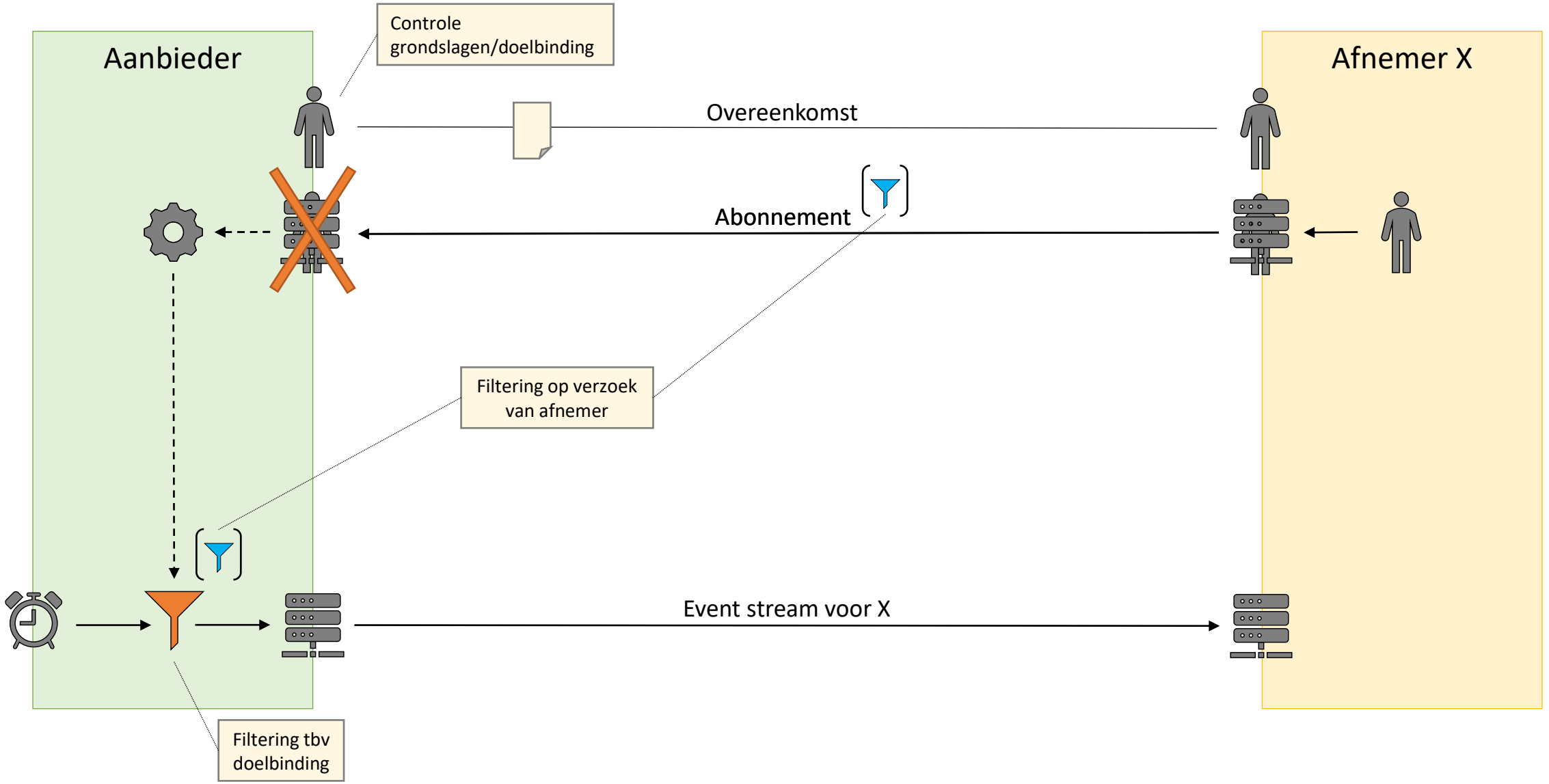


Case: Openbare informatie - intermediair - filtering door intermediair voor afnemer

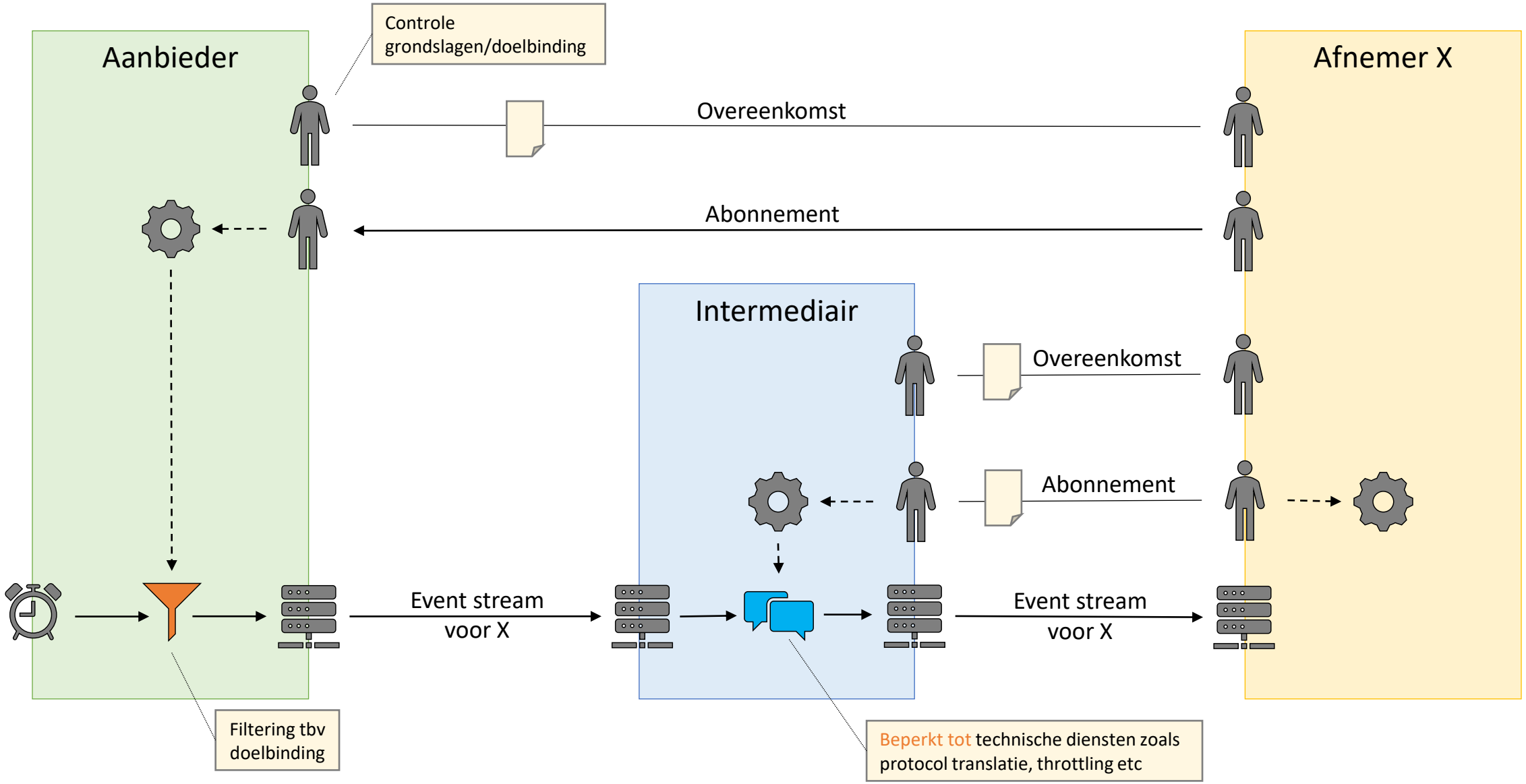


Case: Vertrouwelijke informatie - filtering door aanbieder tbv doelbinding - geen intermediair

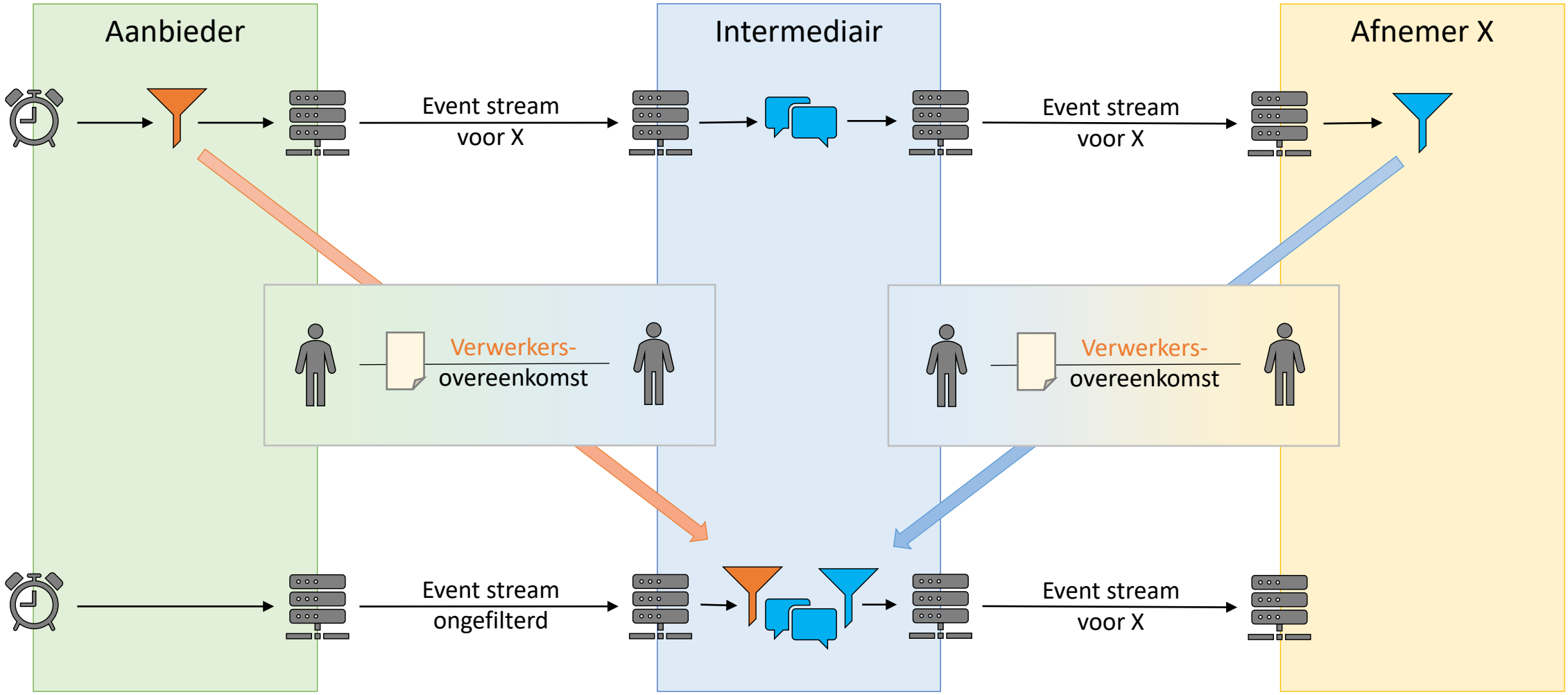
Optioneel: Filtering op verzoek afnemer



Case: Vertrouwelijke informatie - filtering door aanbieder tbv doelbinding - intermediair



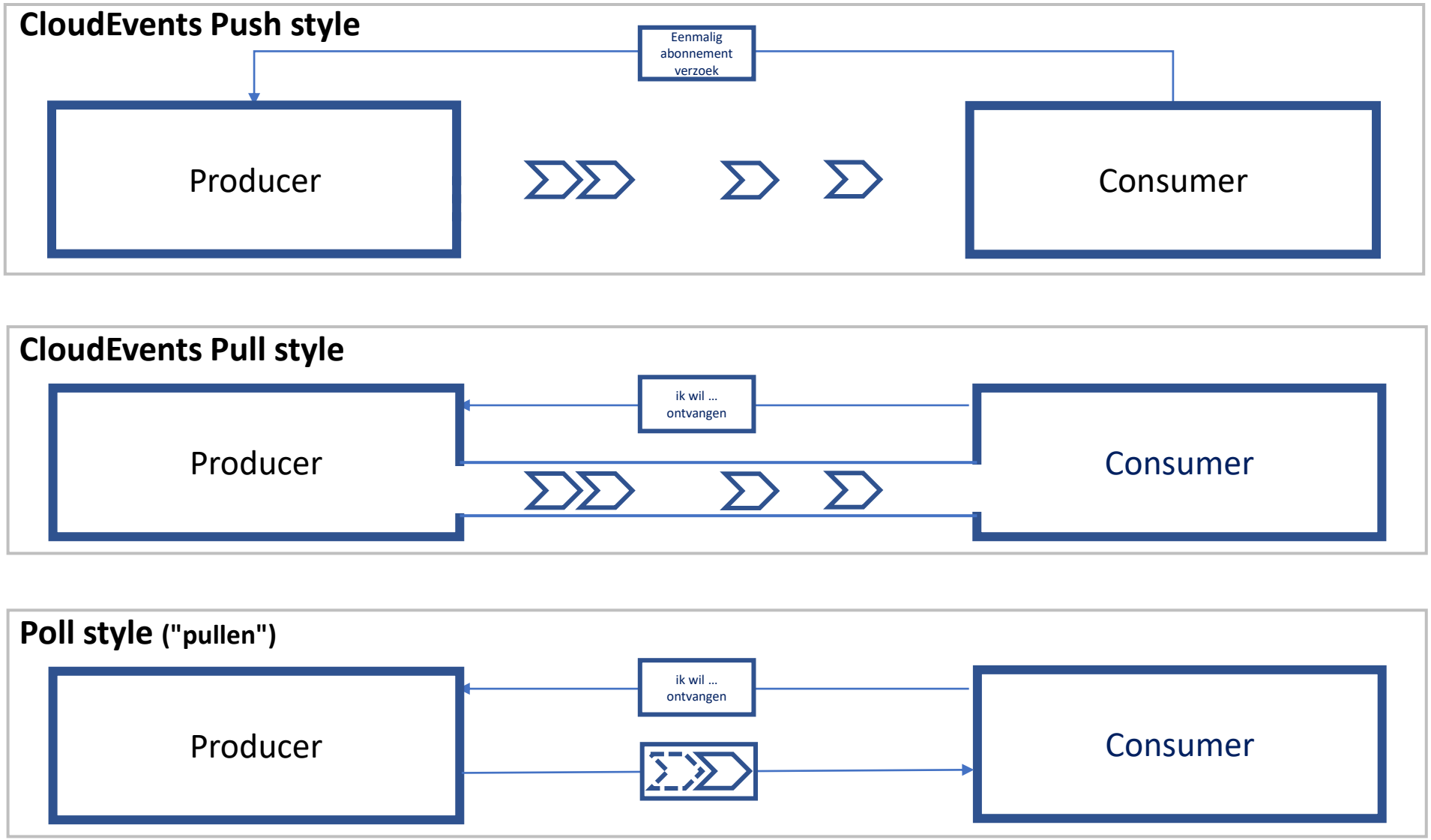
Case: Vertrouwelijke informatie - invloed van verwerkersovereenkomsten



Abonneren en verstrekken (push, pull, poll)

A. CloudEvents Pull Style en Poll Style

Subscription types



As with the core CloudEvents specification, the goal of this specification is to reuse mechanisms based on existing standards and conventions where such exist and only introduce new mechanisms where needed.

Therefore, this specification not only defines

a new subscription management API for certain use-cases, **<= CE Push style**

but also refers to existing mechanisms available in the specifications of transport protocols for which CloudEvents bindings exist. **<= CE Pull style**

= MQTT, AMQP, NATS, Kafka (nu)

"Using any of these subscription management mechanisms ought to allow an application to claim conformance with this specification"

Native mechanismes voor pull-style

MQTT

- "SUBSCRIBE" en "UNSUBSCRIBE" operation

AMQP

- receive-role link met "distribution-mode" "copy" of "move",
- Supporting "copy" mode is OPTIONAL
- Filters via AMQP Filter Expressions 1.0 extension specification of Apache Qpid project's filter definitions

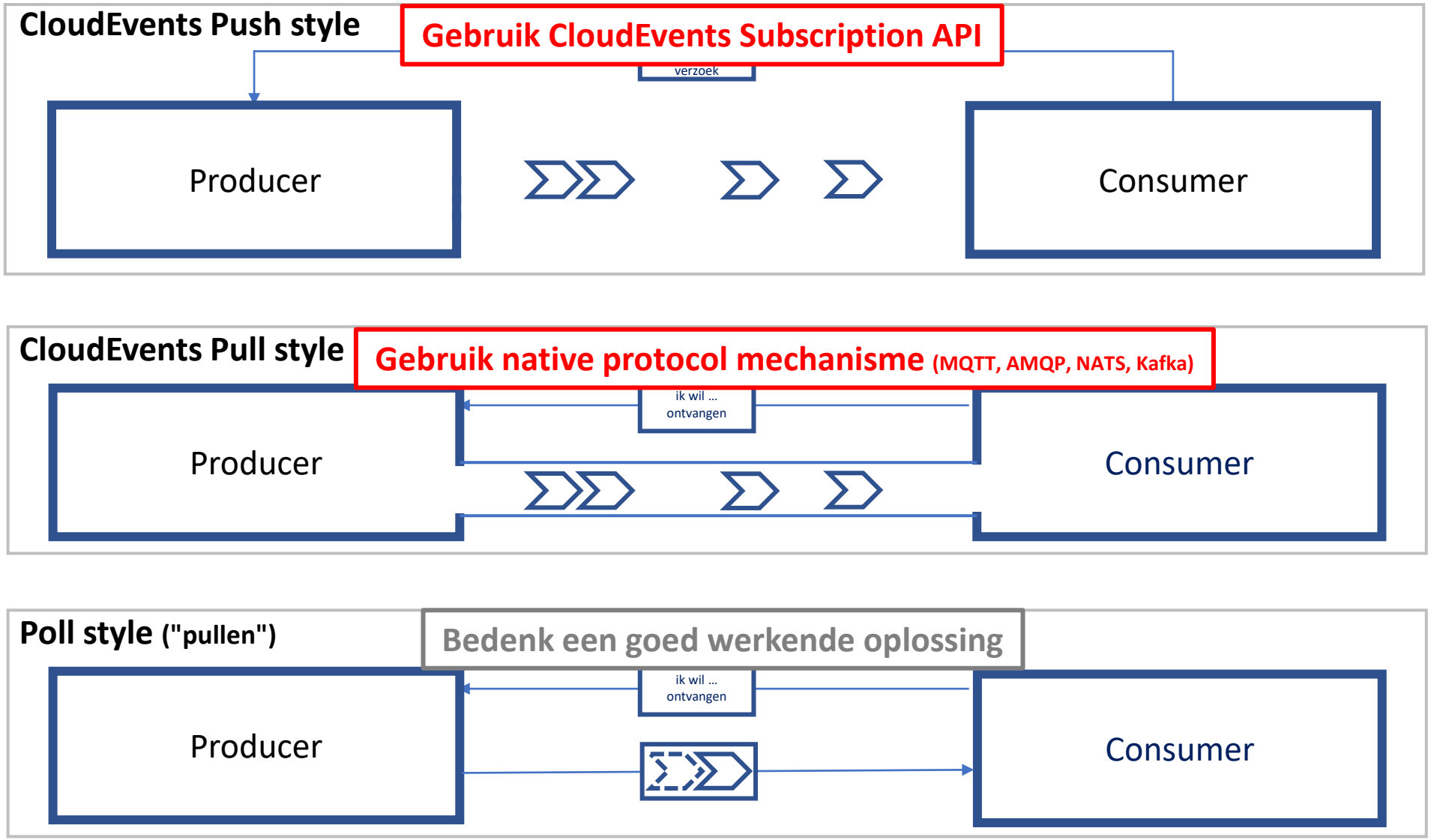
NATS

- subscribe operation "SUB" en "UNSUB"
- publish via "PUB" operation

Apache Kafka

- client controls from which offset in the partition's log store events are being read from
- management of subscription and selecting which events are being fetched and dispatched into the application lies completely with the consumer

CloudEvents adviseert:



HTTP ?

HTTP does not have a built-in subscription mechanism that allows for establishing a flow of events similar to the aforementioned protocols.

HTTP allows for modeling pull-style retrieval of events from a store and, with HTTP/2 "server push", even for continuous delivery an event stream triggered by a initial request,

but for event delivery scenarios, these techniques are applications of HTTP rather than inherent features of HTTP per-se.

Primair request/response maar ook…

HTTP Long-Polling: The connection is held open until the server has new information.

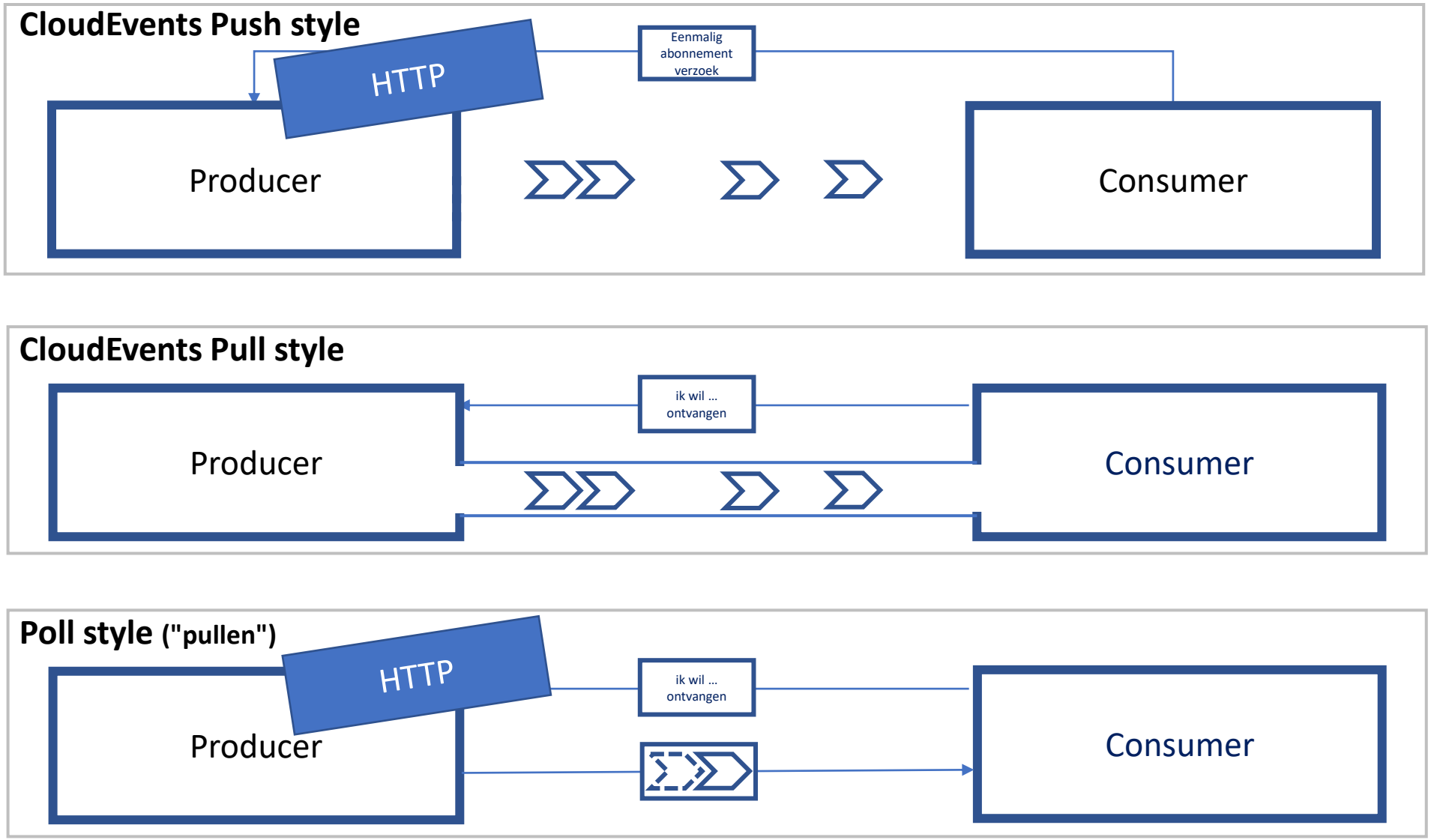
HTTP Streaming: The connection is held open and new pieces of information can be pushed over that existing connection, from server to client.

Server-Sent Events: a standardisation of of HTTP Long-Polling and HTTP-Streaming

HTTP/2 Server Push: Mechanism for pushing from server to client (Chrome to Drop Support for HTTP/2 Server Push, dec-2020 'high complexity, low adoption')

[WebSockets (HTTP++): Full bi-directional and full duplex communication over a single TCP connection within a web browser (or any web client)]

Subscription types



Voorstel

1. CloudEvents berichtstandaard overheidsbreed toepassen
   A. Bij notificaties naar consumers: MUST
   B. Bij gebeurtenisberichten van producer naar intermediar: SHOULD
   C. Geldend voor zowel push-style, pull-style als poll-style*

2. CloudEvents patronen voor abonneren en filteren toepassen
   A. Bij CE push-style: MUST
   B. Bij CE pull-style: SHOULD

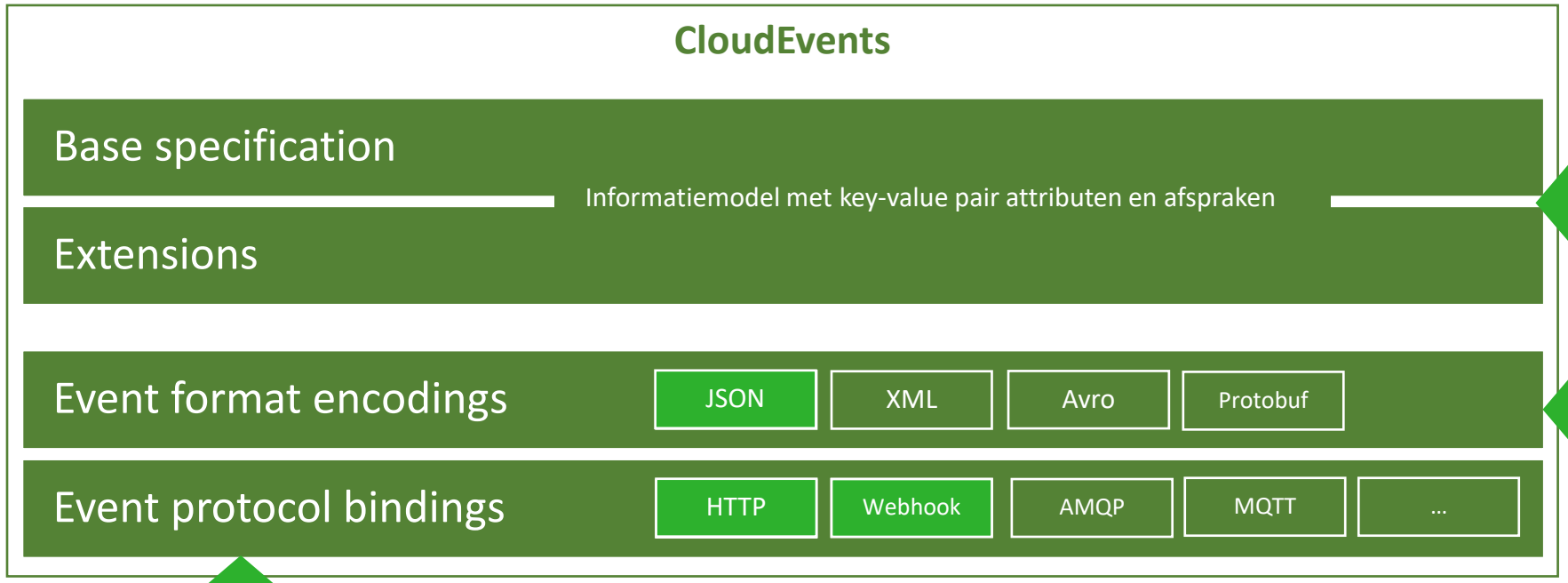
ad *) Voorbeelden van gebruik van het CloudEvents berichtformaat bij poll-style:

- CSV bestanden die via FTP worden opgehaald bevatten headers conform CloudEvents-attributen
- Een resource 'events' die via een REST-API wordt geraadpleegd levert conform CloudEvent berichtformaat

Cloud events architectuur

Gelaagde architectuur met de lagen 'formaat' en 'protocol'

CloudEvents gelaagde architectuur



Formaat- en protocol- neutraal !

The event format encodings, e.g. JSON, define how the information model of the base specification together with the chosen extensions is encoded for mapping it to header and payload elements of an application protocol.

The protocol bindings, e.g. HTTP, defines how the CloudEvent is bound to an application protocol's transport frame, in the case of HTTP to the HTTP message. The protocol binding does not constrain how the transport frame is used, meaning that the HTTP binding can be used with any HTTP method and with request and response messages.

CloudEvents JSON format specification

How events are expressed in JavaScript Object Notation (JSON) Data Interchange Format (RFC8259)

JSON Type mapping

CloudEvents	JSON
Boolean	boolean
Integer	number, only the int component is permitted
String	string
Binary	string, Base64-encoded binary
URI	string following RFC 3986
URI-reference	string following RFC 3986
Timestamp	string following RFC 3339 (ISO 8601)

- Unset attributes MAY be encoded to the JSON value of null.
- CloudEvents JSONSchema for the spec is located here

Voorbeeld:

CloudEvents	Type	Exemplary JSON Value
type	String	"com.example.someevent"
specversion	String	"1.0"
source	URI-reference	"/mycontext"
subject	String	"larger-context"
subject	String (null)	null
id	String	"1234-1234-1234"
time	Timestamp	"2018-04-05T17:31:00Z"
time	Timestamp (null)	null
datacontenttype	String	"application/json"

JSON Envelope

Each CloudEvents event can be wholly represented as a JSON object and MUST use the media type **application/cloudevents+json**

Event attribute:

- 'data' => decoded using the default JSON type mapping for the used runtime of
- 'data_base64' => MUST decode into a binary runtime data type.

Voorbeelden:

```
…
"datacontenttype" : "text/xml",
"data" : "<much wow=\"xml\"/>"
…
```

```
…
"datacontenttype" : "application/vnd.apache.thrift.binary",
"data_base64" : "... base64 encoded string ..."
…
```

```
…
"datacontenttype" : "application/json",
"data" : { "appinfoA" : "abc", "appinfoB" : 123, "appinfoC" : true }
…
```

JSON Batch format (optioneel te ondersteunen)

```
[
  {
      "specversion" : "1.0",
      "type" : "com.example.someevent",
      "source" : "/mycontext/4",
      "id" : "B234-1234-1234",
      "time" : "2018-04-05T17:31:00Z",
      "comexampleextension1" : "value",
      "comexampleothervalue" : 5,
      "datacontenttype" : "application/vnd.apache.thrift.binary",
      "data_base64" : "... base64 encoded string ..."
  },
  {
      "specversion" : "1.0",
      "type" : "com.example.someotherevent",
      "source" : "/mycontext/9",
      "id" : "C234-1234-1234",
      "time" : "2018-04-05T17:31:05Z",
      "comexampleextension1" : "value",
      "comexampleothervalue" : 5,
      "datacontenttype" : "application/json",
      "data" : {
          "appinfoA" : "abc",
          "appinfoB" : 123,
          "appinfoC" : true
      }
  }
]
```

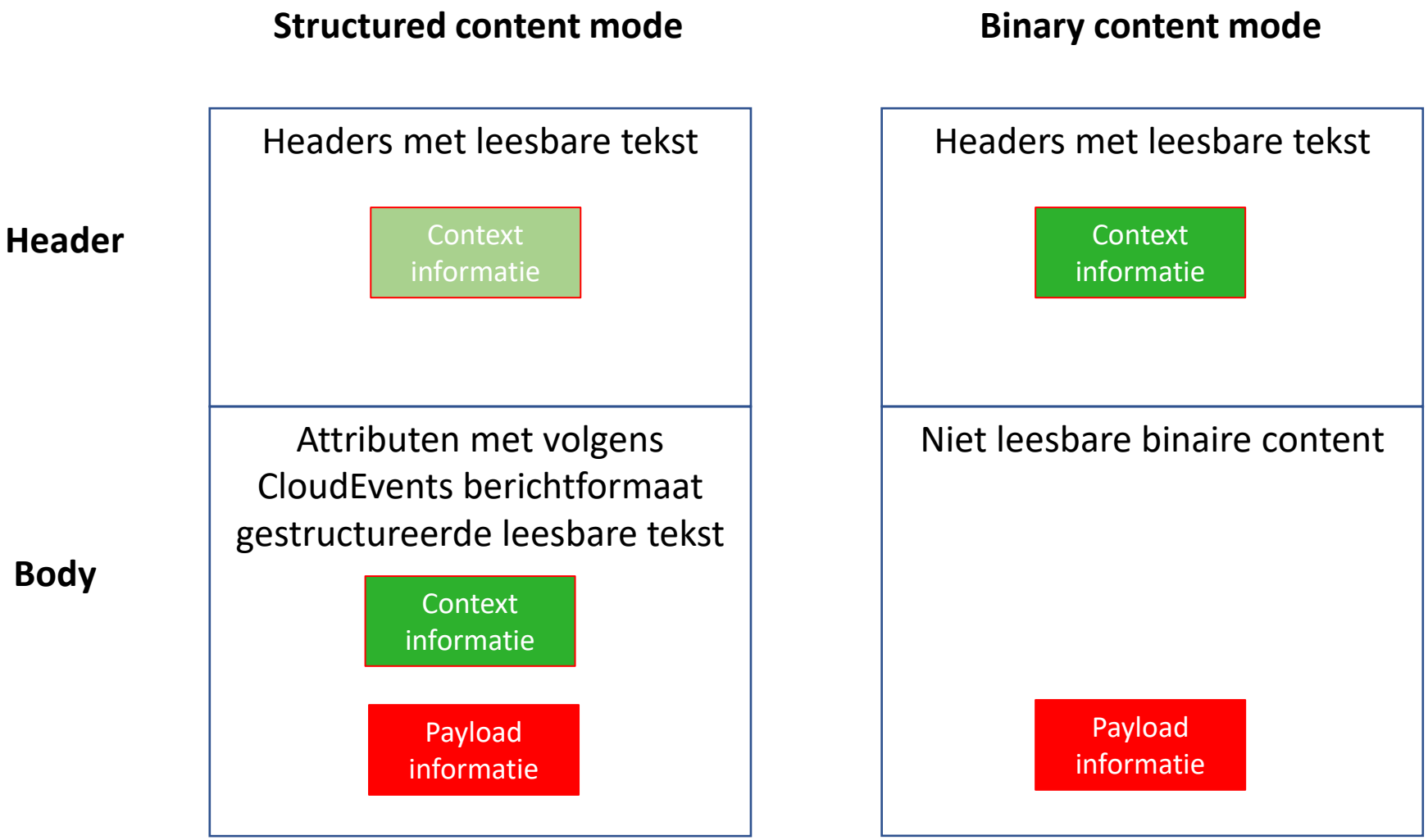
CloudEvents HTTP binding specification

How events are mapped to HTTP request and response messages

Request & response

- Events can be transferred with all standard or application-defined HTTP request methods that support payload body transfers.
- Events can be also be transferred in HTTP responses and with all HTTP status codes that permit payload body transfers
- CE attribuut 'datatype' moet in HTTP-Header 'Content-Type' .

HTTP Content Modes



+ **Batched content mode** =
"array van structured content mode events"

Bepalen content mode door ontvanger

```
let content-type = request.headers('content-type')

if (content-type = "application/cloudevents-batch") {
        contentmode = "batched"
} else if (content-type. startsWith("application/cloudevents") ) {
        contentmode = "structured"
} else {
        contentmode = "binary"
}
```

Bijv. "application/cloudevents+json"

Structured content mode

- MUST support JSON event format (non batching)
- MAY support any additional, including proprietary, formats
- Alle CE-contextattributen MOETEN in het bericht
- Het CE-attribuut 'datatype' MOET in HTTP Header 'Content-Type'
- Het gebruikte event format bepaalt formaat en betekenis van attributen

```
PUT /myresource HTTP/1.1
Host: webhook.example.com
Content-Type: application/cloudevents+json; charset=utf-8
Content-Length: nnnn

{
    "specversion" : "1.0",
    "type" : "com.example.someevent",

    ... further attributes omitted ...

    "data" : {
        ... application data ...
    }
}
```

Binary content mode

- Alle CE-contextattributen MOETEN in HTTP Headers (en krijgen dan het voorvoegsel 'ce-')
- Het mediatype van het event format MOET in HTTP Header 'Content-Type'
- Het gebruikte event format bepaalt formaat en betekenis van attributen

Request

```
POST /someresource HTTP/1.1
Host: webhook.example.com
ce-specversion: 1.0
ce-type: com.example.someevent
ce-time: 2018-04-05T03:56:24Z
ce-id: 1234-1234-1234
ce-source: /mycontext/subcontext
    .... further attributes ...
Content-Type: application/json; charset=utf-8
Content-Length: nnnn

{
    ... application data ...
}
```

Response

```
HTTP/1.1 200 OK
ce-specversion: 1.0
ce-type: com.example.someevent
ce-time: 2018-04-05T03:56:24Z
ce-id: 1234-1234-1234
ce-source: /mycontext/subcontext
    .... further attributes ...
Content-Type: application/json; charset=utf-8
Content-Length: nnnn

{
    ... application data ...
}
```

Vervolg

Vervolg

7 oktober - volgende bijeenkomst

Backlog

• Verwerken van resultaten uit deze sessie	Project
• Werkversie van basis-bericht specificatie opleveren	Project
• Toetsen van toepasbaarheid basis-bericht in de studieomgeving	Project
• Vergelijking gemaakte keuzes met ZGW Notificatie-standaard	Project