

Reliable Messaging

Marc de Graauw



Betrouwbaar transport

- Netwerk is niet betrouwbaar
- Het is niet te garanderen dat twee partijen beide 100% zeker weten dat communicatie geslaagd is
- Het is wel te garanderen dat als de communicatie niet geslaagd is, tenminste een partij dat weet
- Aanname: vroeg of laat is het netwerk weer beschikbaar

Eisen betrouwbaar transport

- Ieder bericht wordt minimaal een keer afgeleverd
- Ieder bericht wordt maximaal een keer afgeleverd
- Berichten worden afgeleverd in de volgorde waarin ze verzonden zijn
- Afgeleverde berichten worden bewaard

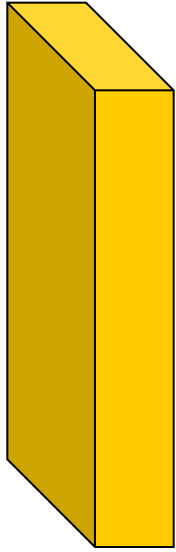
Het Issue

- Transport over HTTP is niet betrouwbaar
- Geen antwoord op een bericht:
 - door timeout, netwerk down, server down
 - niet zeker of het bericht al dan niet aangekomen en/of verwerkt is

HTTP Methods (RFC 2616)

- safe
 - heeft geen neveneffecten op de server
 - HTTP: GET
 - b.v.: opvragen informatie
- idempotent
 - laat server in zelfde staat na herhaling
 - HTTP DELETE, PUT
 - b.v. verwijderen record
- niet-idempotent
 - wijzigt server
 - HTTP POST
 - b.v. 100 aandelen kopen

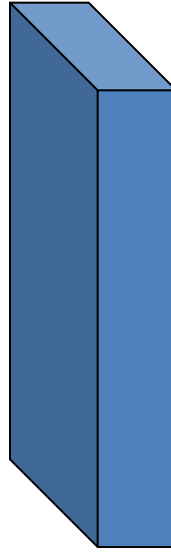
Zender



Bericht

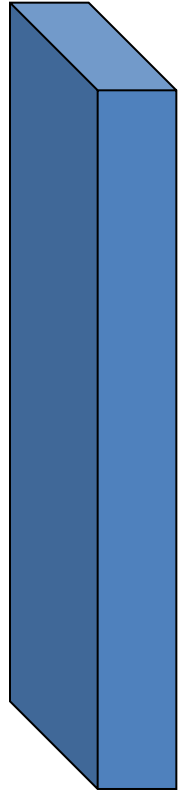
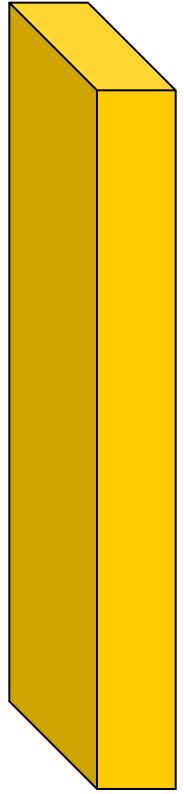


Ontvanger

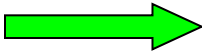


Zender

Ontvanger



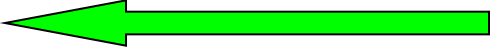
Bericht



Kopie van bericht



Ontvangstbevestiging

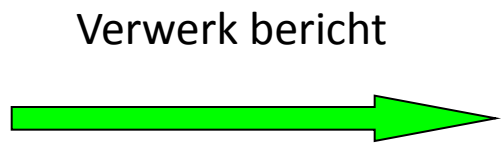
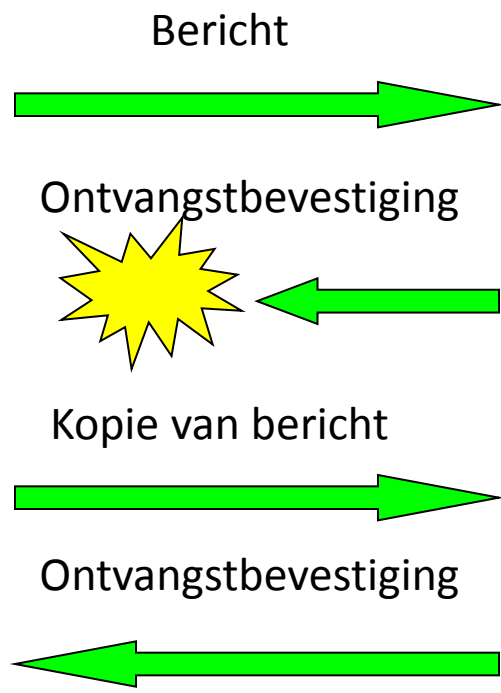
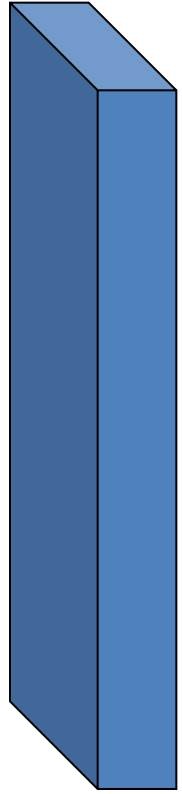
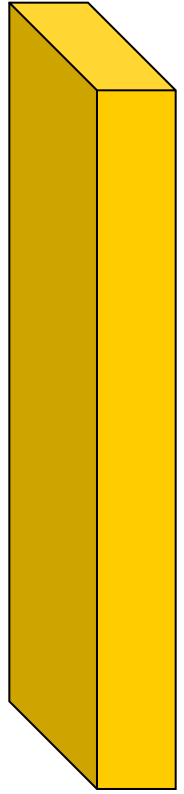


Verwerk bericht



Zender

Ontvanger



Betrouwbare aflevering

- Gegarandeerde aflevering
 - berichtnummers
 - ontvangstbevestiging
 - eventueel duplicaat zenden
- Eenmalige aflevering
 - duplicaatdetectie
 - duplicaatverwijdering
- Volgordelijke aflevering
 - oplopende berichtnummering

Eisen aan betrouwbaar transport

- Eis 1: duplicaatdetectie is nodig voor niet-idempotente acties
- Eis 2: ontvangstbevestigingen zijn nodig bij niet-directe acties
- Ontwerprichtlijn: Geef bij iedere interactie aan of deze veilig en/of idempotent is
- Ontwerprichtlijn: Ieder bericht waarop niet direct een (ander) antwoord komt, krijgt een ontvangstbevestiging
- Ontwerprichtlijn: Idempotente berichten waarop het verwachte antwoord niet komt, worden herhaald met een nieuw bericht-id. Geen duplicaat dus, maar een nieuw bericht
- Ontwerprichtlijn: De ontvanger hoeft voor idempotente berichten geen status (bericht-id en verzonden antwoord) te bewaren
- Ontwerprichtlijn: Niet-idempotente berichten waarop het verwachte antwoord niet komt, worden herhaald met een duplicaat.

Transport Level Reliability

WS-ReliableMessaging

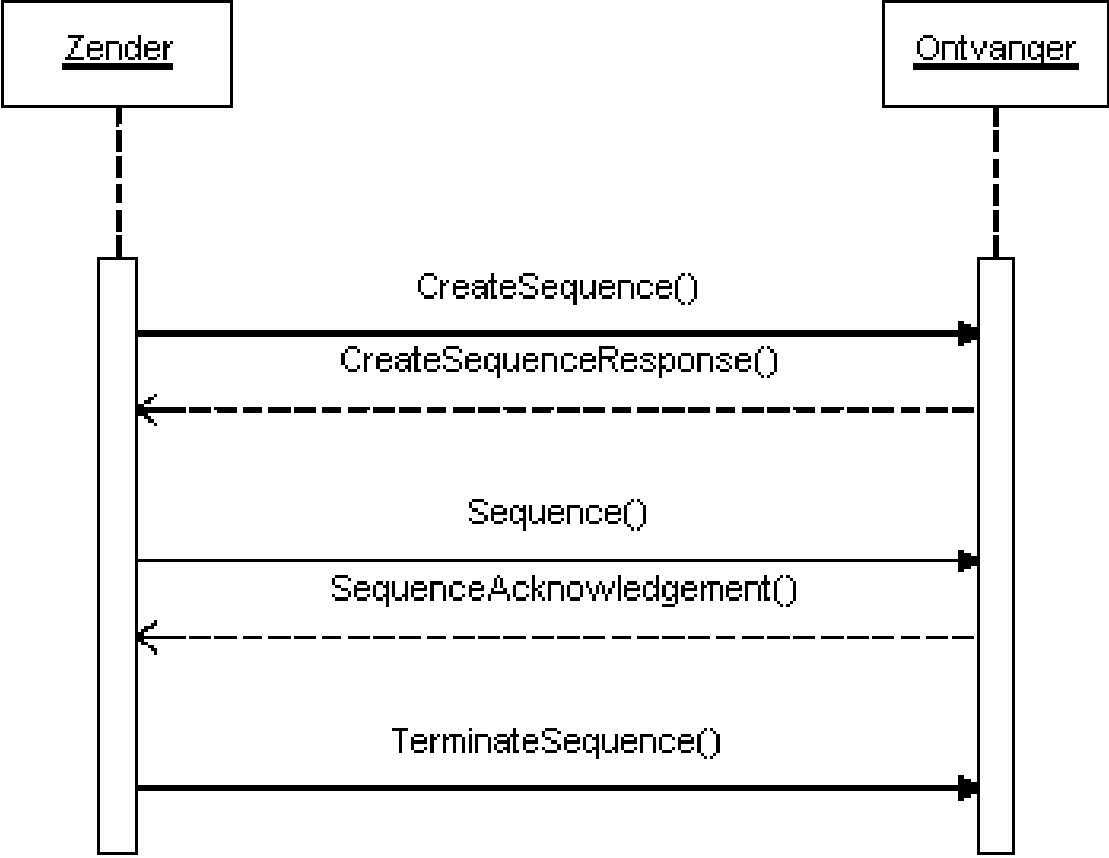
- Apache Sandesha 2 ondersteund WS-RM 1.1 voor Axis2 Java en C++
- <http://ws.apache.org/sandesha/sandesha2/index.html>
- WSO2 bevat Sandesha2 en ondersteund daarmee ook WS-RM 1.1.

- Metro 3.0.1 (onderdeel van J2EE reference implementatie "glassfish") ondersteund in ieder geval WS-RM 1.1.
- <https://metro.dev.java.net/>

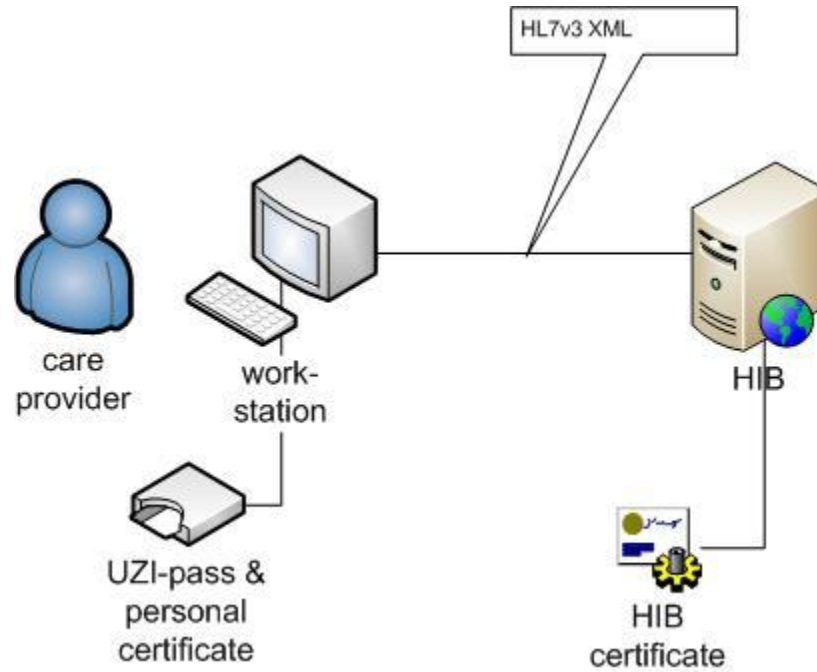
- Apache CXF ondersteund gezien de source alleen WS-RM 1.0.

- JBossWS vanaf versie 3.0.1: ondersteund 1.0 en 1.1.
- <https://jira.jboss.org/jira/browse/JBWS-515>
- http://jbossws.jboss.org/mediawiki/index.php?title=WS_Reliable_Messaging

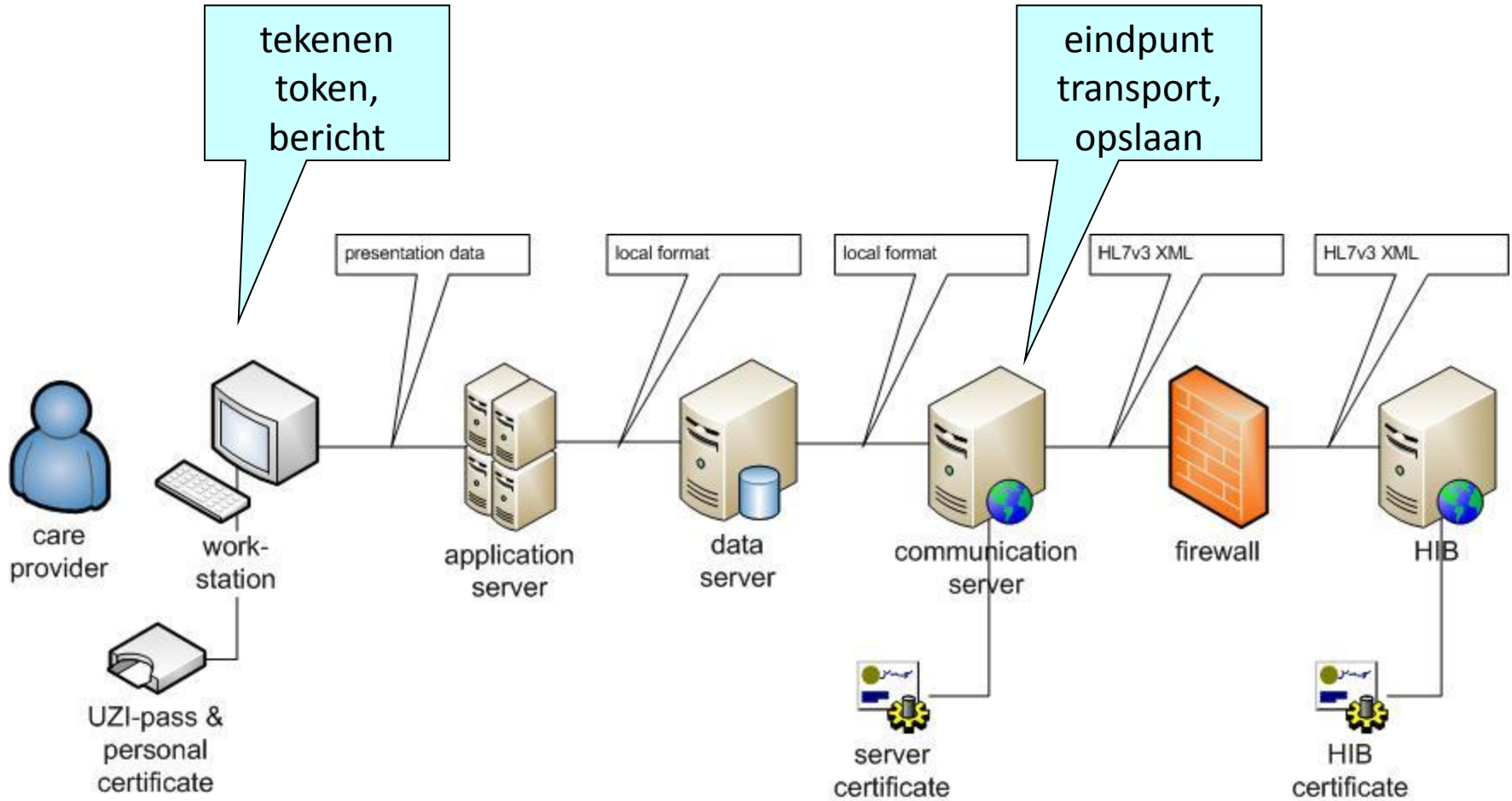
- Microsoft .Net ondersteund WS-RM 1.1 vanaf versie 3.5, Windows Communication Framework

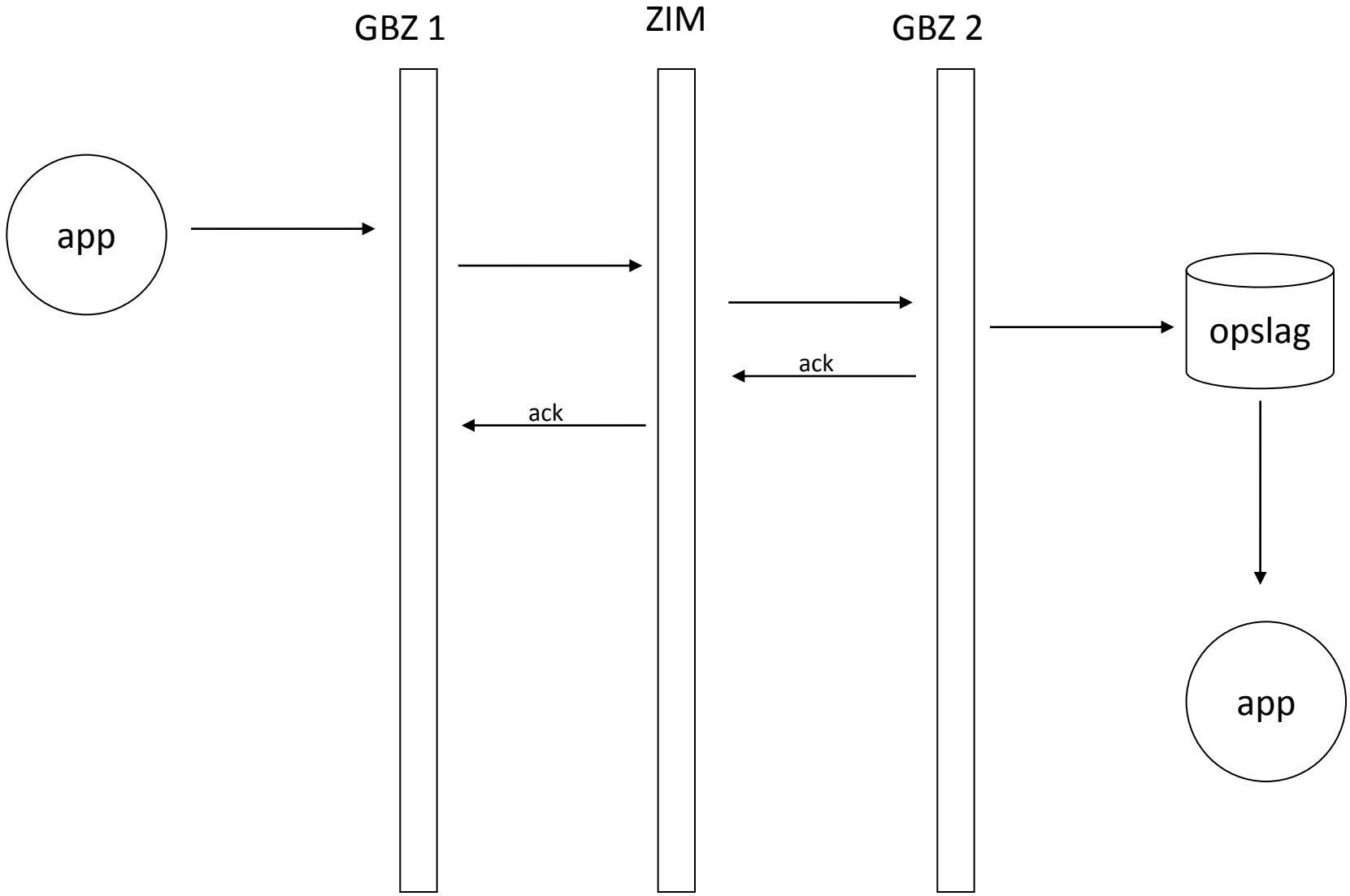


Infra 1



Infra 2





WSRM patronen

- 1-way
- bidirectional
- anonymous
- veel berichten (atom. aanmelden)

- plaatjes

Zender

Ontvanger

1. Queries

HL7v3 Query

HL7v3 Antwoord

**2. Bericht
zonder
antwoord**

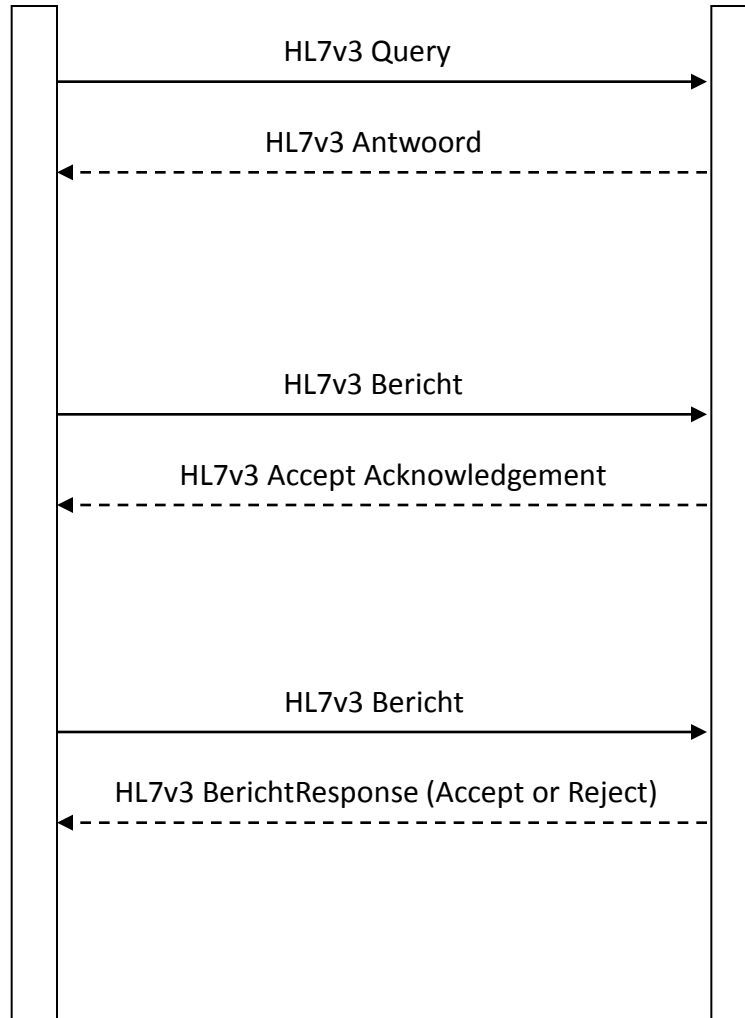
HL7v3 Bericht

HL7v3 Accept Acknowledgement

**3. Bericht
met
antwoord**

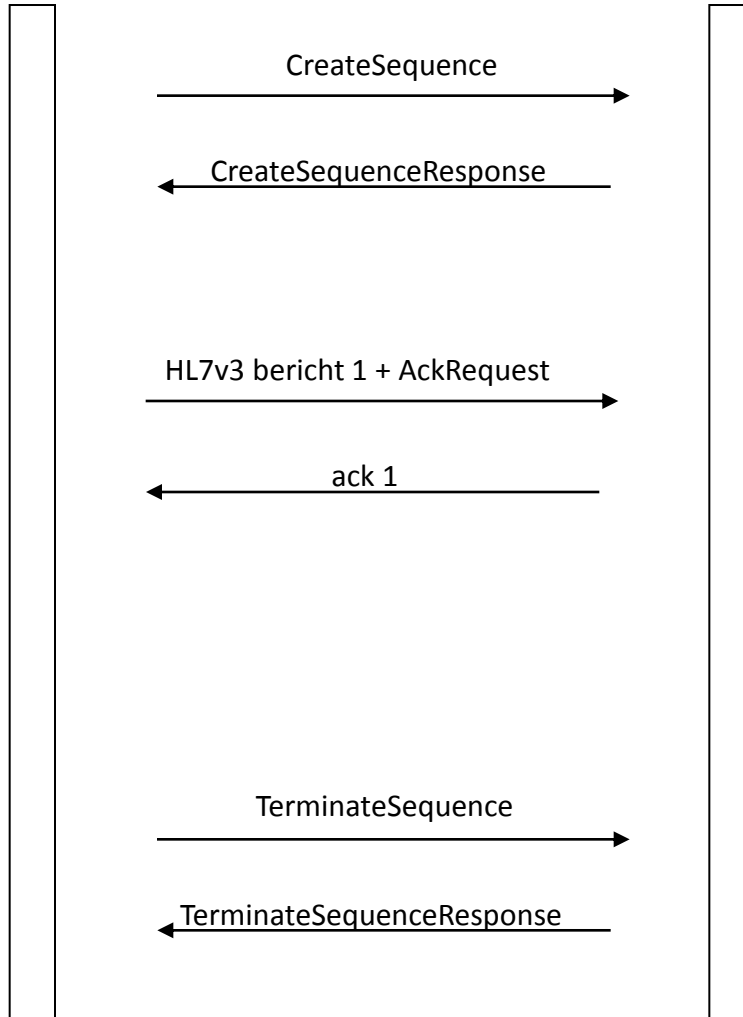
HL7v3 Bericht

HL7v3 BerichtResponse (Accept or Reject)



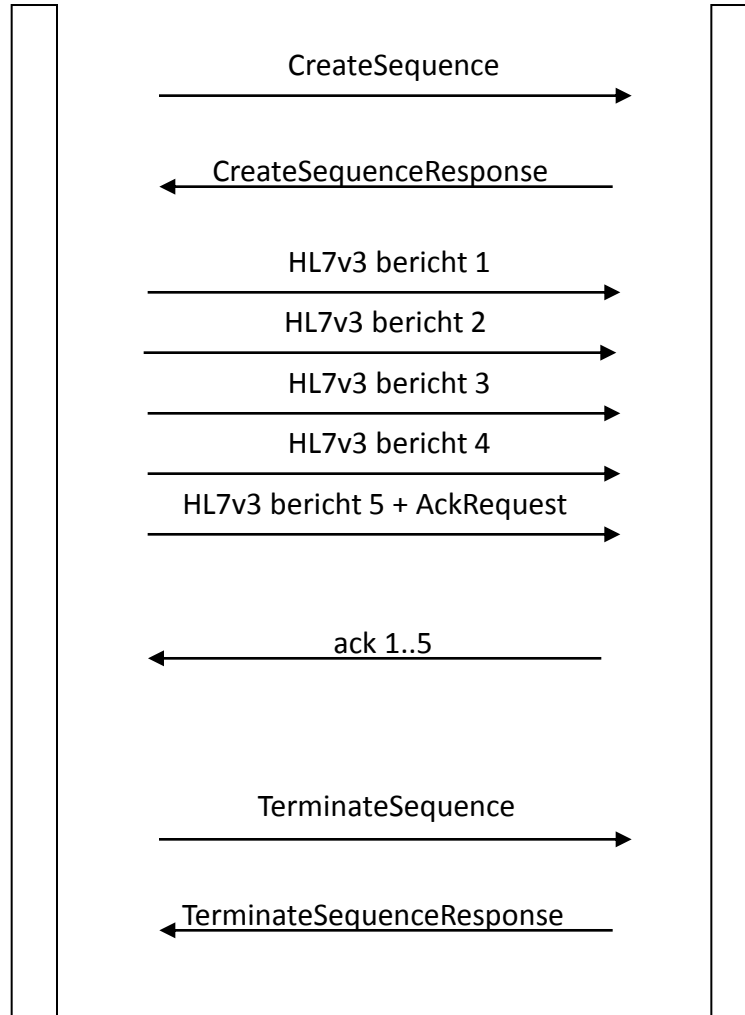
GBZ 1

ZIM



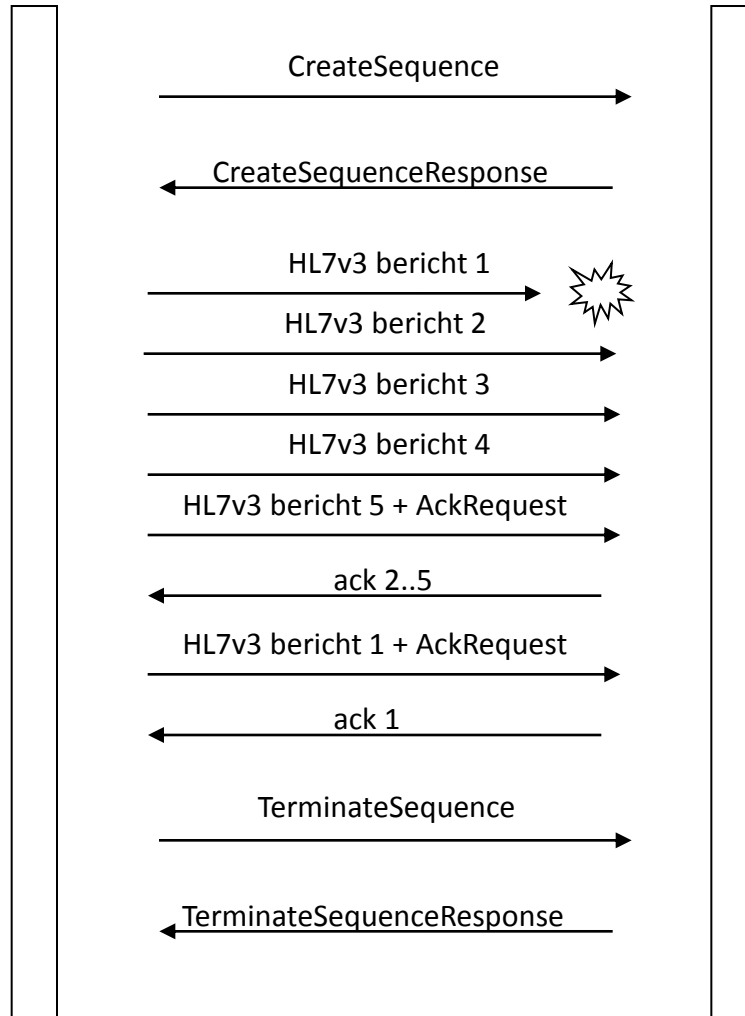
GBZ 1

ZIM



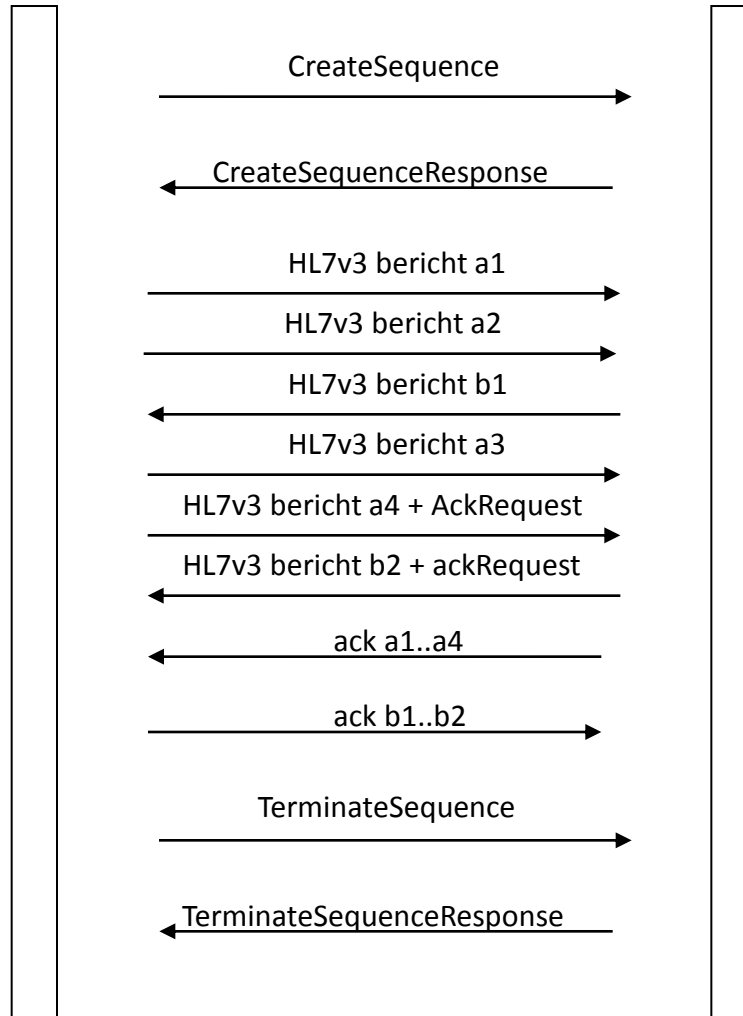
GBZ 1

ZIM



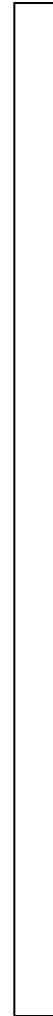
GBZ 1

ZIM



GBZ 1

ZIM



HTTP Request + HL7v3 bericht 1

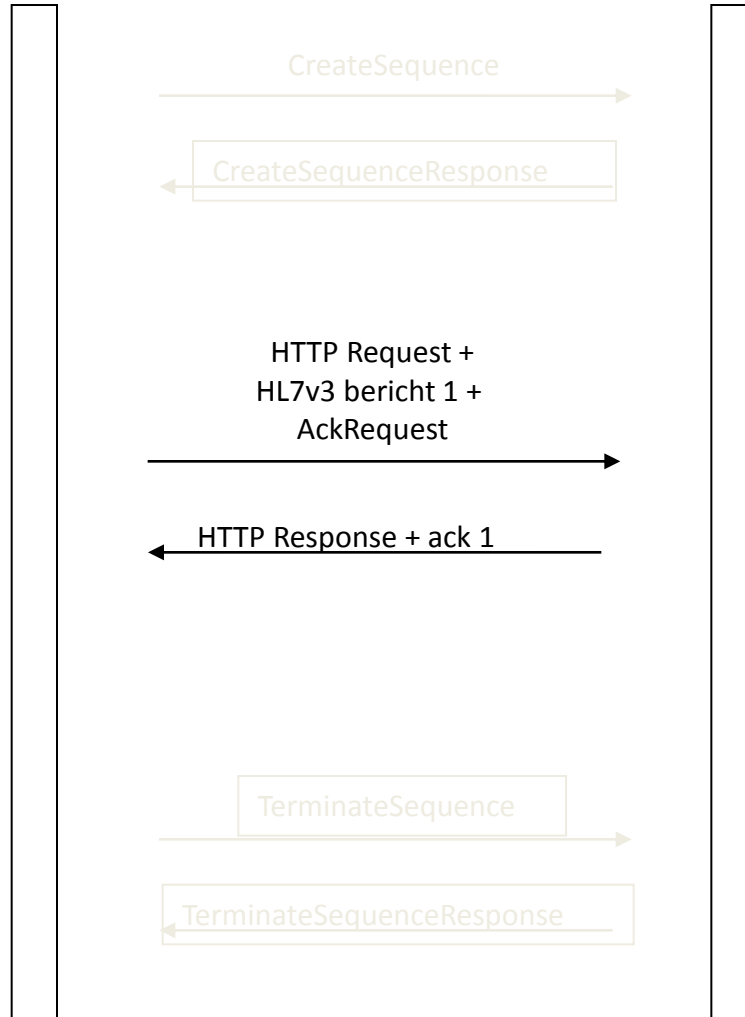


HTTP Response + HL7v3 Accept Ack



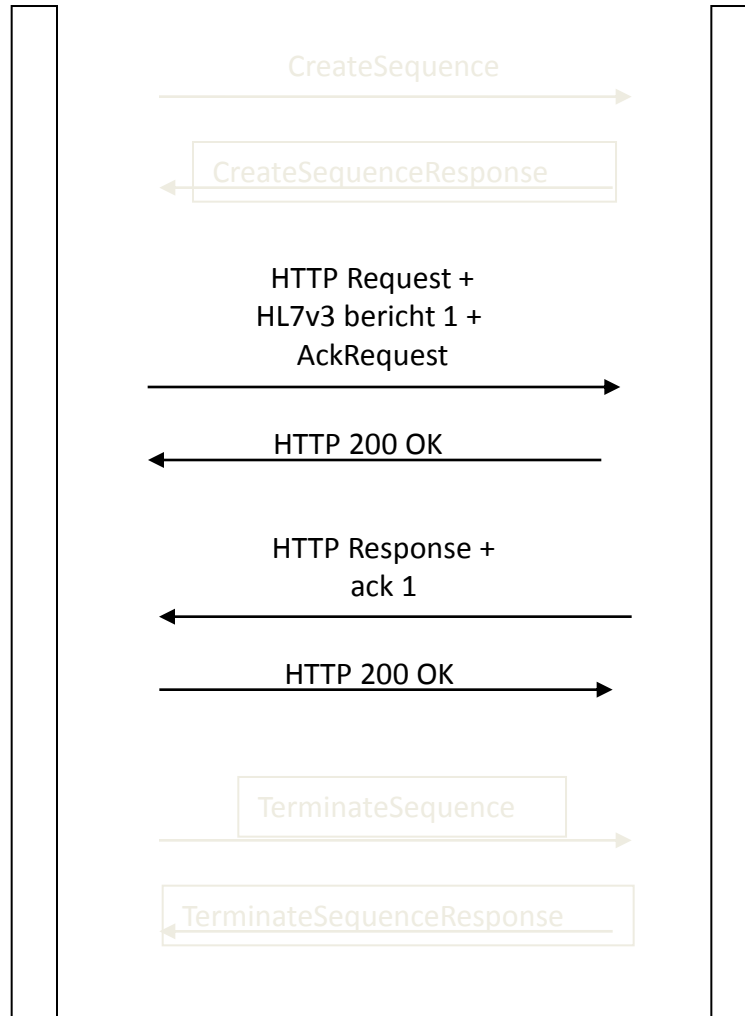
GBZ 1

ZIM



GBZ 1

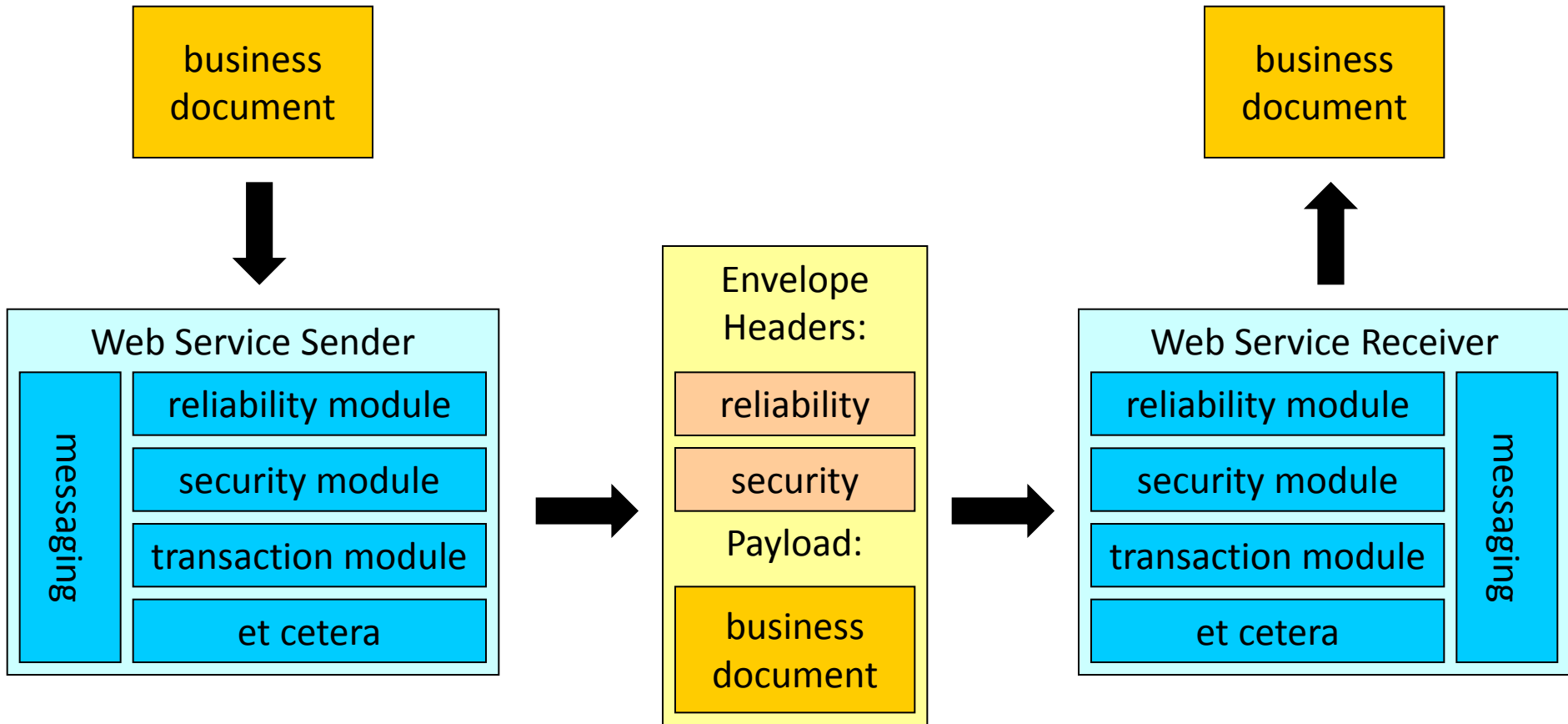
ZIM



Welke communicatiepatronen?

- anonymous one-way, one message, one ack
- anonymous one-way, many messages, many acks
- addressed one-way, many messages, one ack?
- bidirectional – (nog) niet?
- alles anonymous?

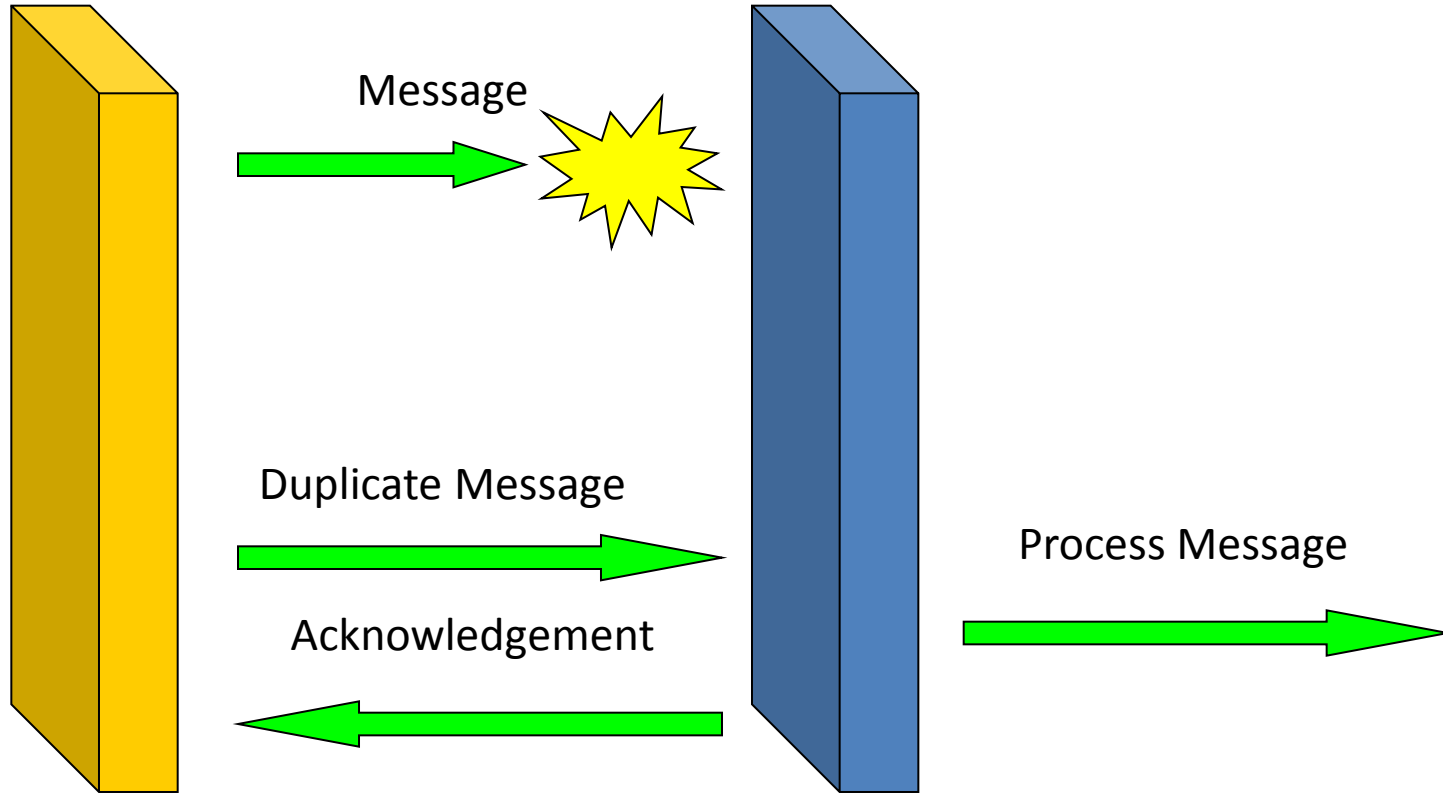
Business level Reliability



Scenario 1: Resending works

Sender

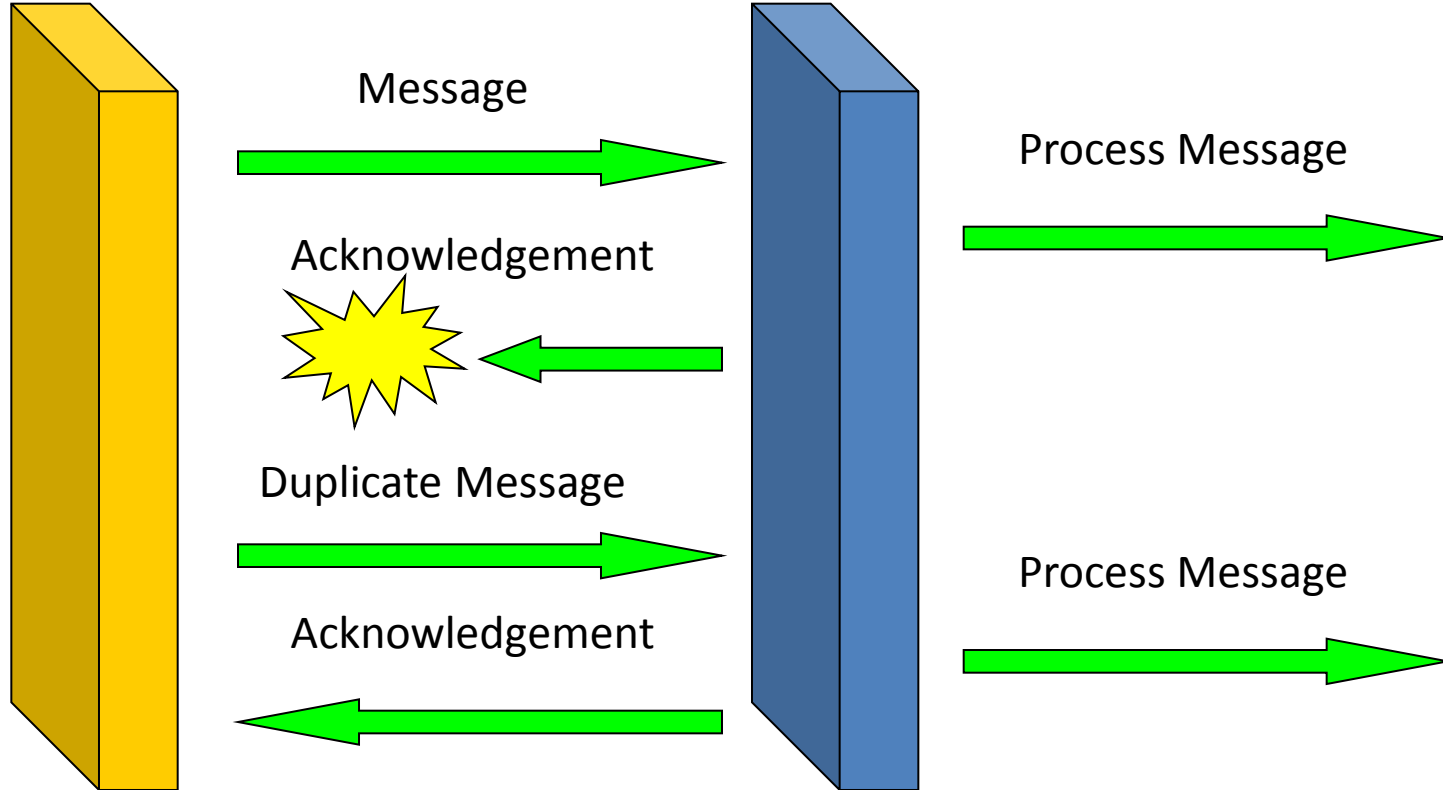
Receiver



Scenario 2: Resending fails

Sender

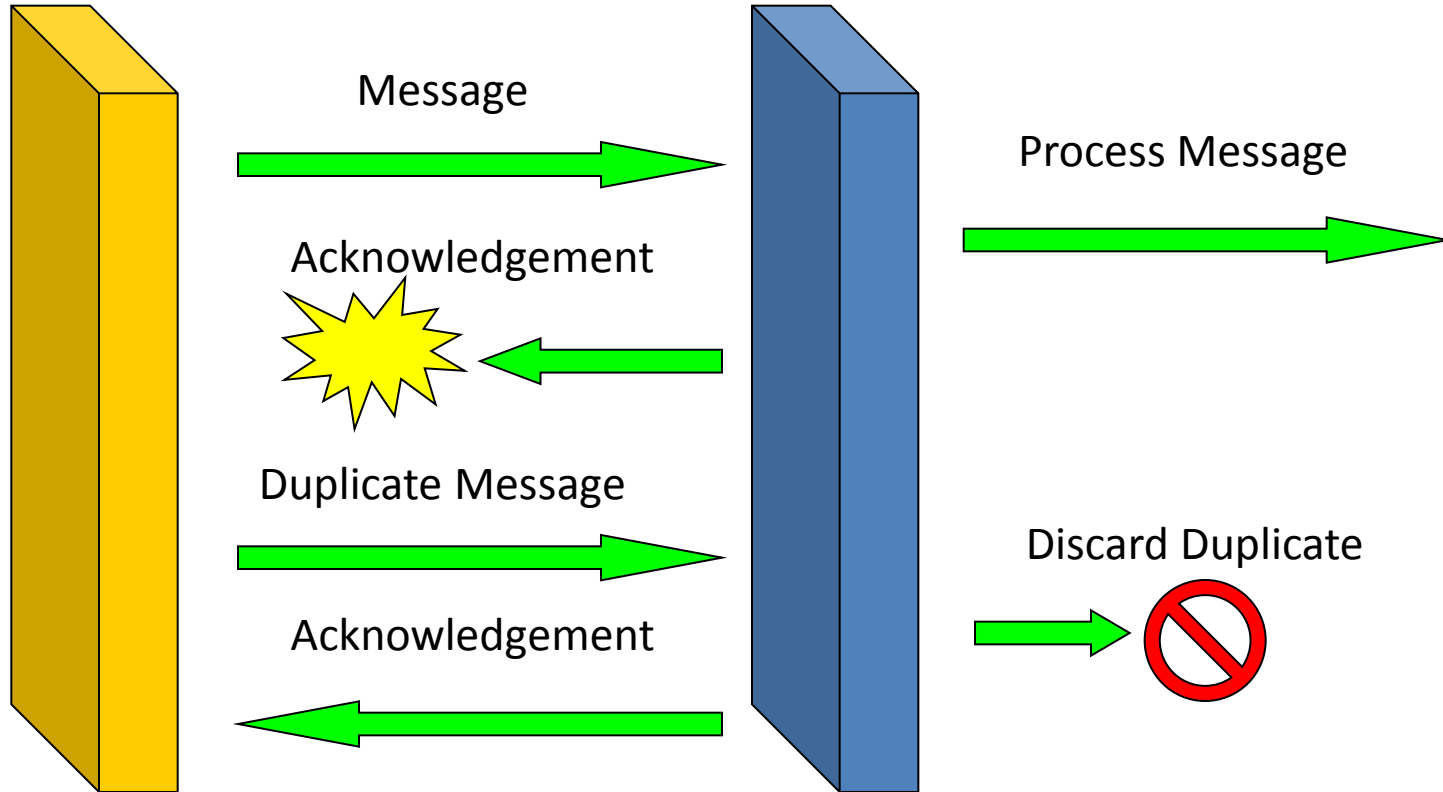
Receiver



Scenario 3: Reliable messaging

Sender

Receiver

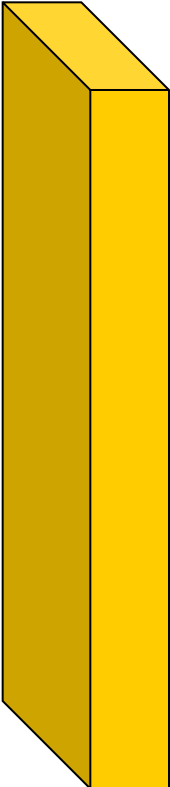


Sender's
Business
Module

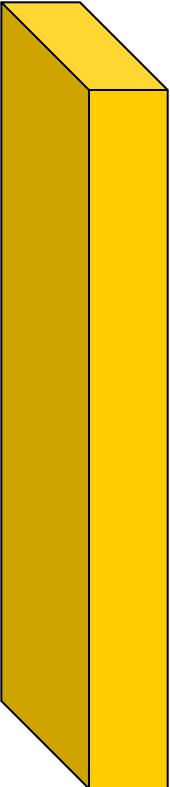
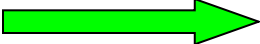
Sender's
Transport
Module

Receiver's
Transport
Module

Receiver's
Business
Module



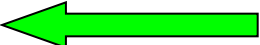
Place Order



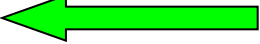
Order
Message



Transport Ack



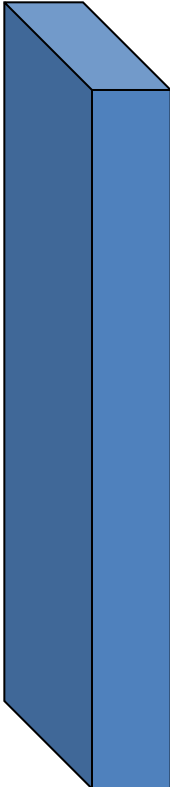
Accept
Message



Transport Ack



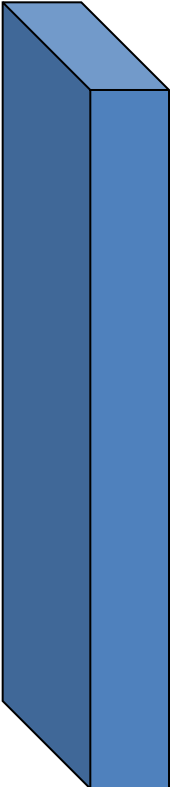
Submit
Accept



Submit
Order



Accept Order



GP's
Consultation
Application

GP's
Transport
Module

Pharmacists
Transport
Module

Pharmacists
Prescription
Application

