(IN) COMPATIBLE LANGUAGE

Versiecontrole in de keten

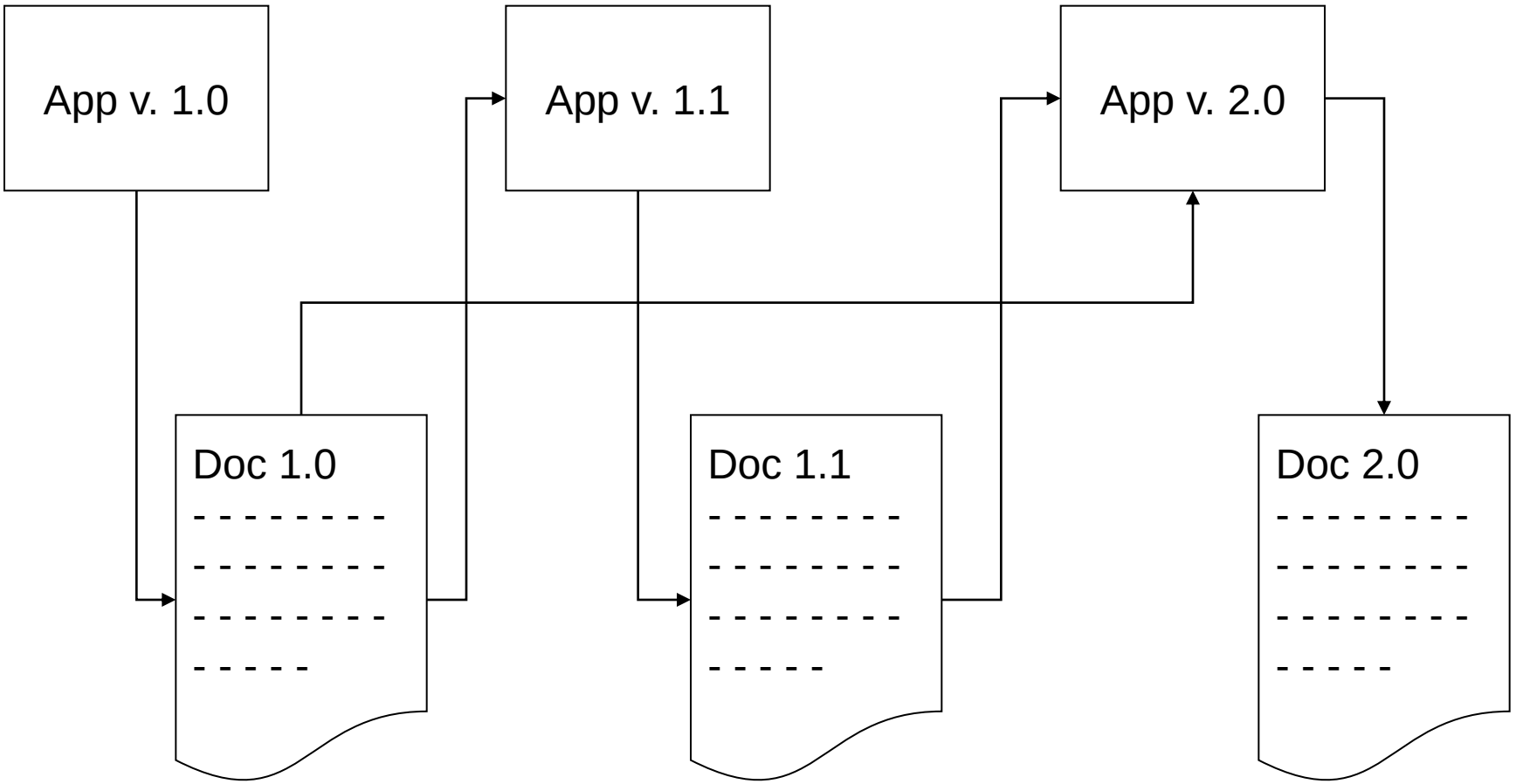
Het probleem

- we hebben een keten,
- en een uitwisseling, met berichten, documenten et cetera
- iedereen is gelukkig...
- en nu komt versie 2.0
  - moet iedereen tegelijk over naar 2.0?
  - en als dat niet kan?
  - hoe regel je een makkelijke overgang

Referenties

- **David Orchard**
  - **Extending and Versioning Languages Part 1, W3C**
    - **Draft TAG Finding**
  - **A Theory of Compatible Versions , XML.COM**
    - **Forward compatibility and extensibility**
- **James Clark**
  - **Validation not necessarily harmful, blog**
- **Marc de Graauw**
  - **On Compatibility - Back and Forth, blog**
  - **Syntactical and Semantical Compatibility , blog**
  - **More Compatibility Flavours , blog**
  - **A Smoother Change to Version 2.0, XML.COM**

Klassieke Backward Compatibiliteit



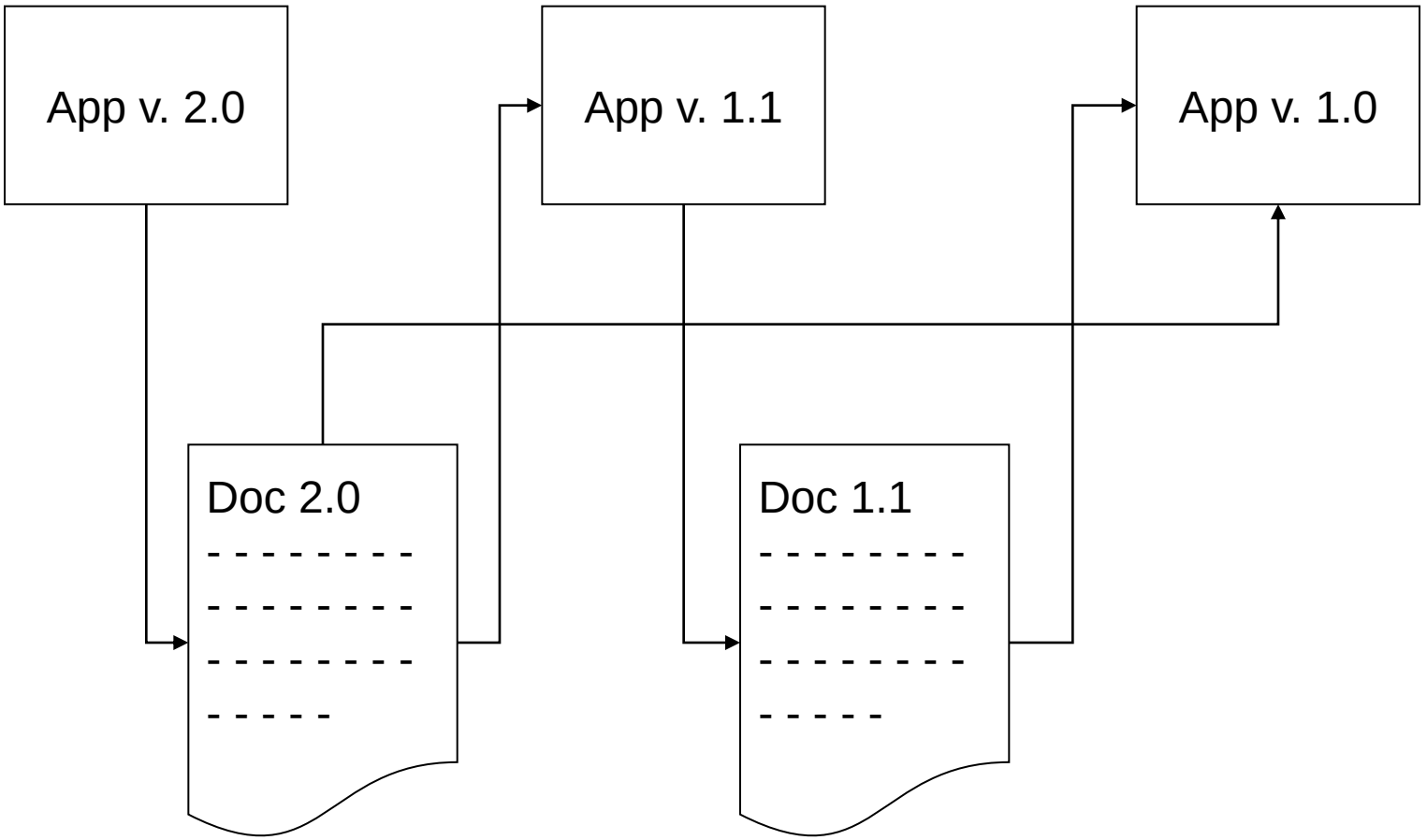
Klassieke Backward Compatibiliteit

- Nieuwe applicatie kan oude documenten lezen
- Eventueel: eenmalige upgrade van documenten
  - “Wilt u dit document opslaan als Word 97?”

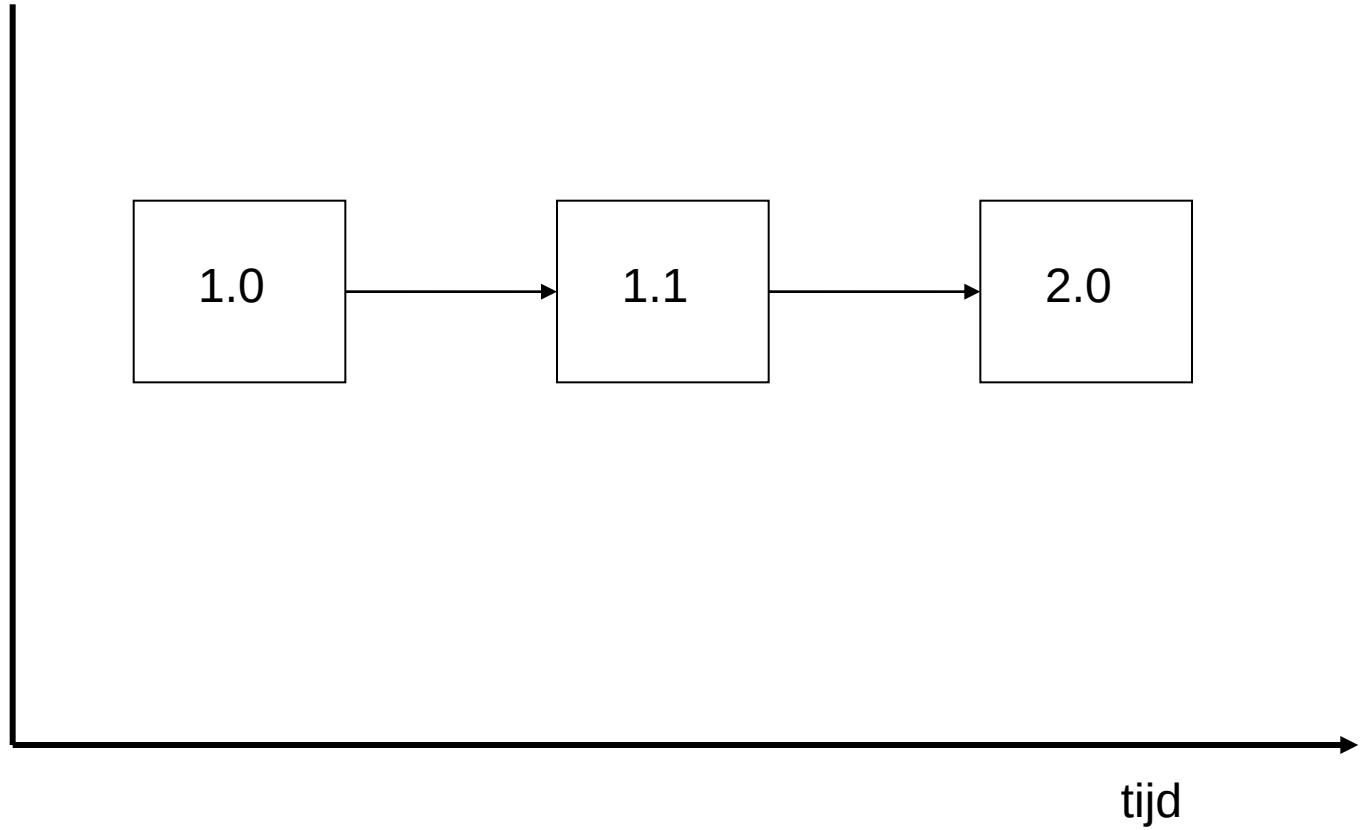
Forward Compatibiliteit

- Oude applicatie kan nieuwe documenten lezen
- Voorbeeld
- HTML – “Ignore Unkown”
  - ”If a user agent encounters an element it does not recognize, it should try to render the element’s content.” - HTML 4.01

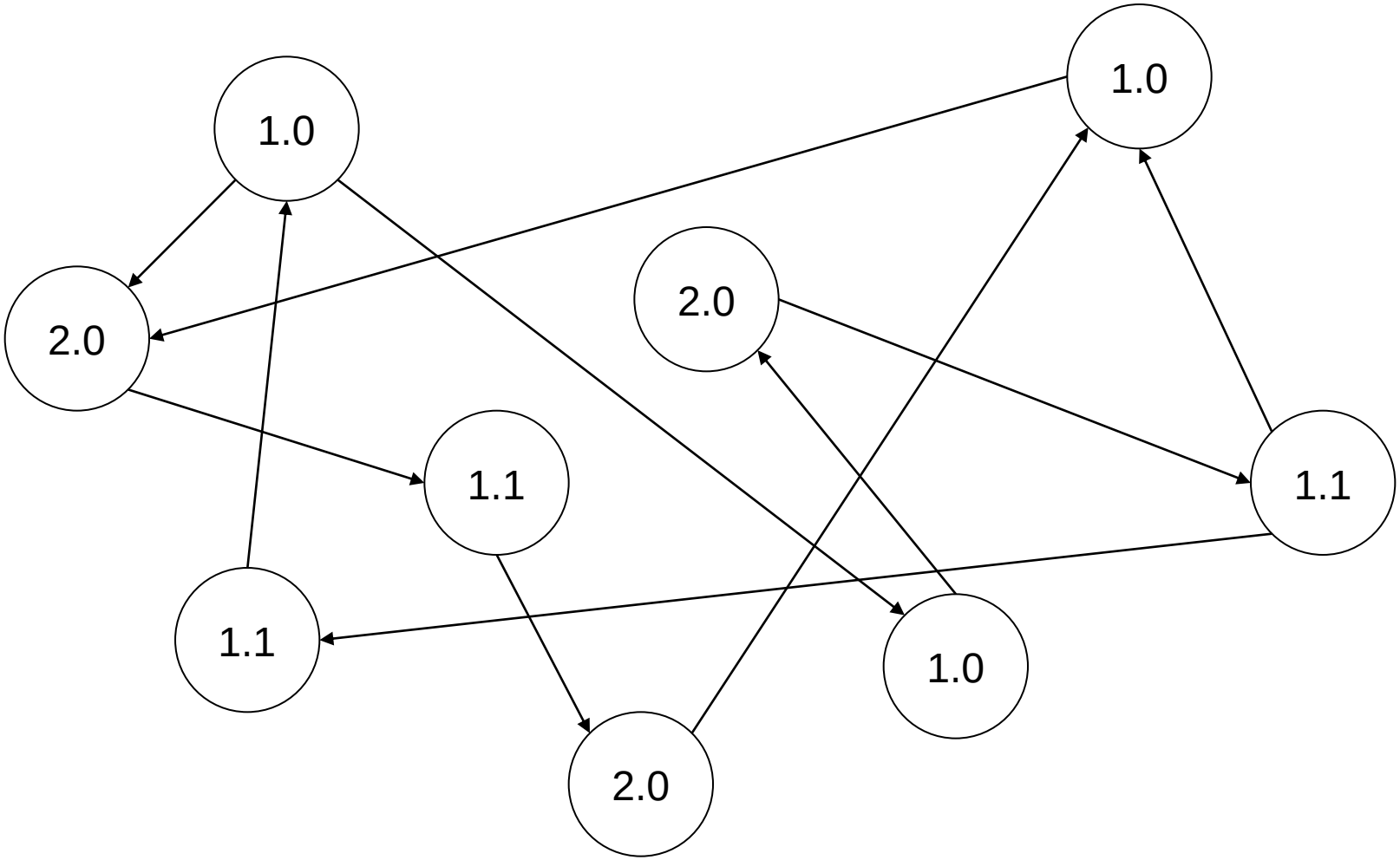
Forward Compatibiliteit



Klassiek...



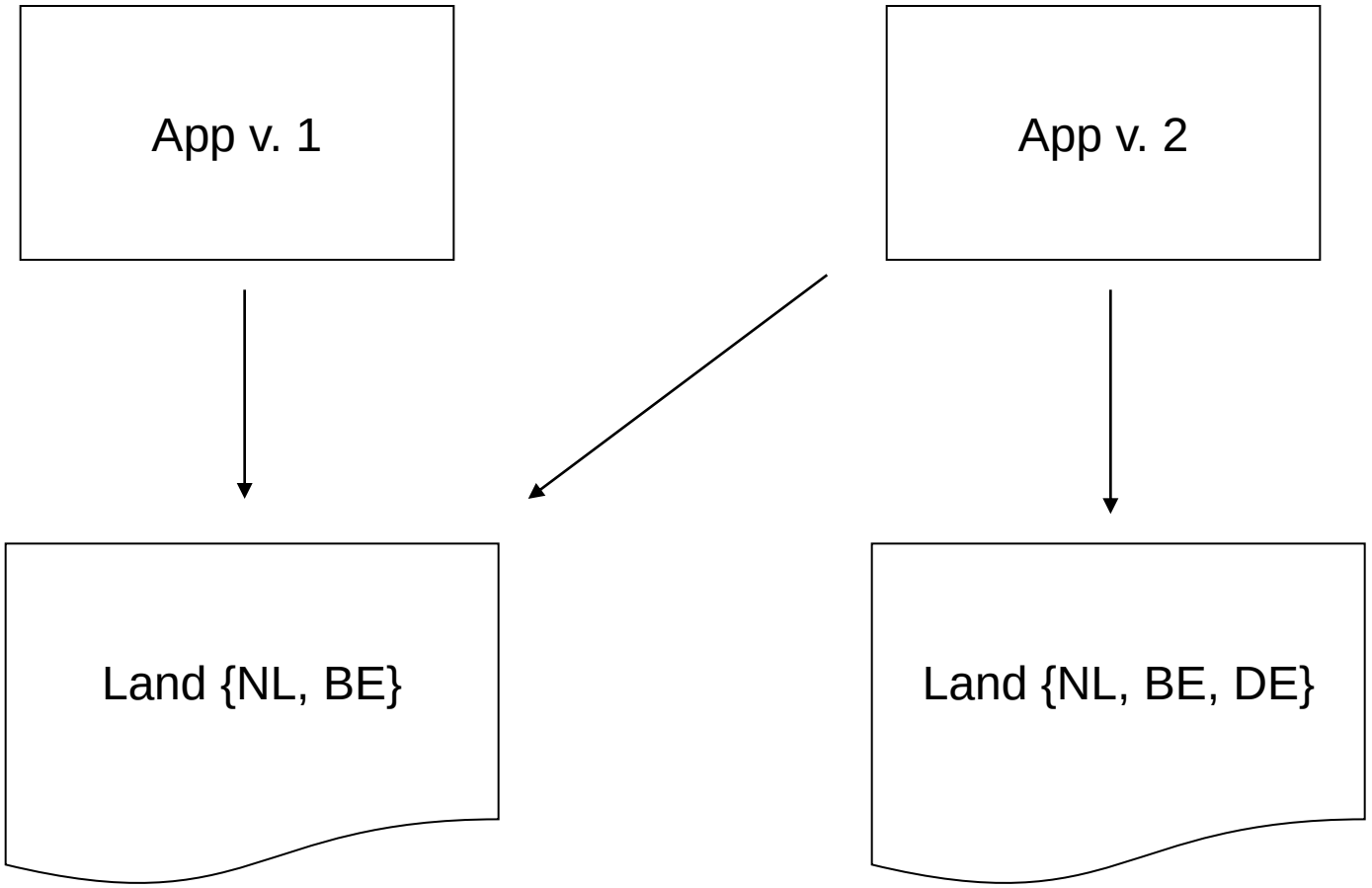
In de keten...



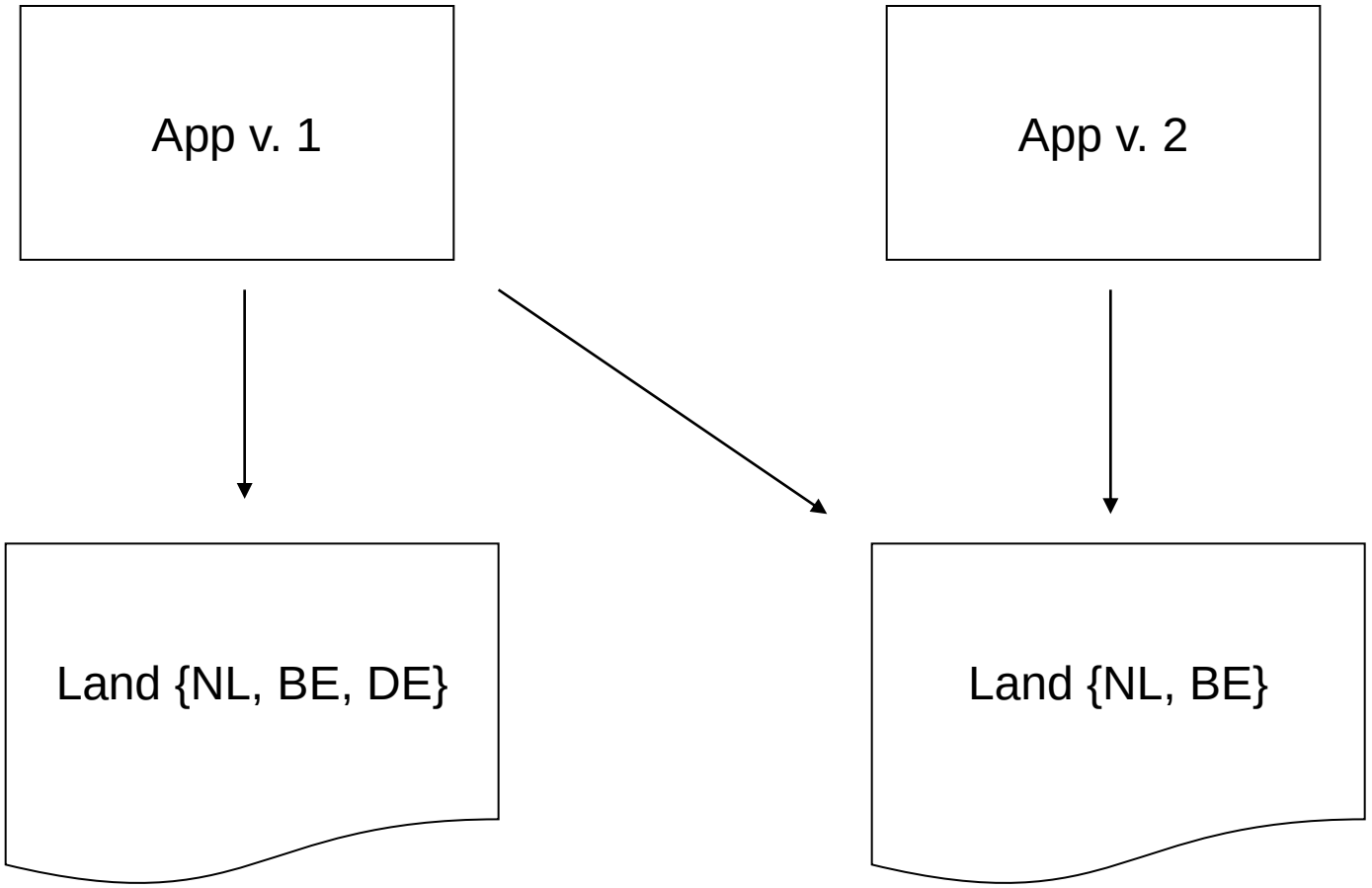
Compabiliteit in de keten...

- iedereen communiceert met iedereen...
- geen of beperkte centrale controle
  - in een (groot) bedrijf: allemaal upgraden
  - in een keten: vaak onmogelijk, altijd onwenselijk
- zenders en ontvangers

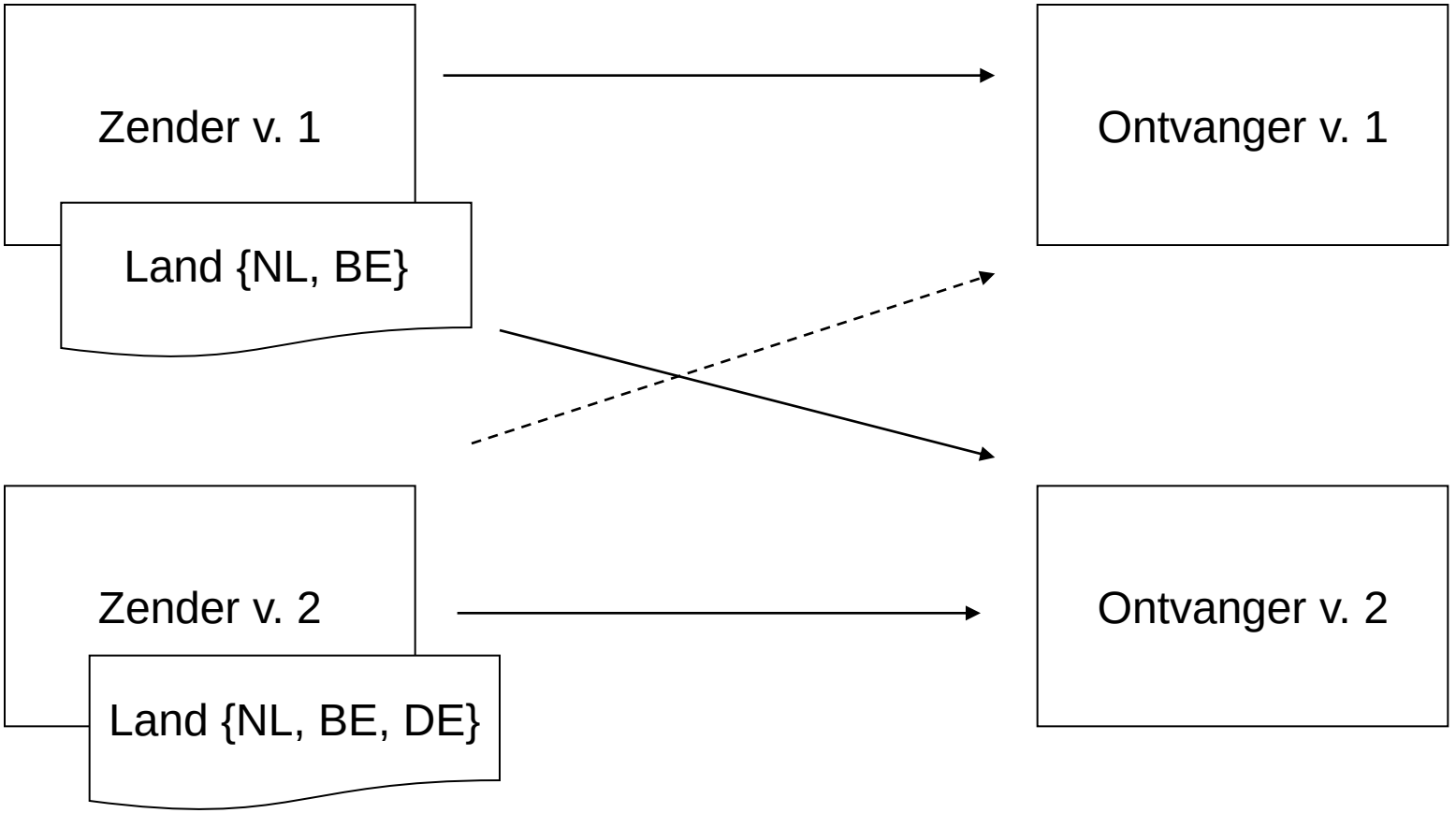
Recap: Backward, niet forward



Recap: Forward, niet backward

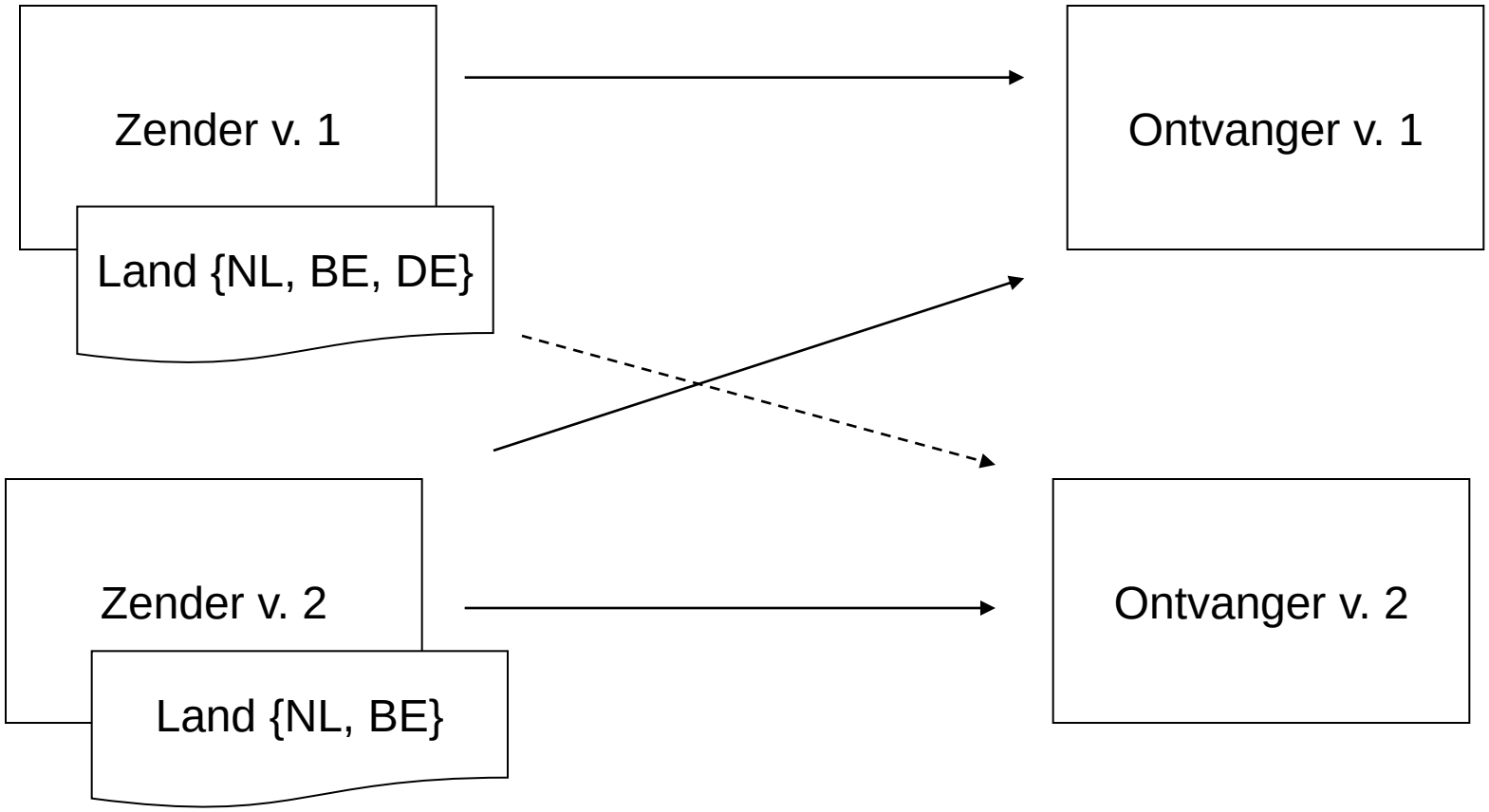


Geen forward compatibiliteit



Problemen voor nieuwe zender en oude ontvanger

Geen backward compatibiliteit



Problemen voor oude zender en nieuwe ontvanger

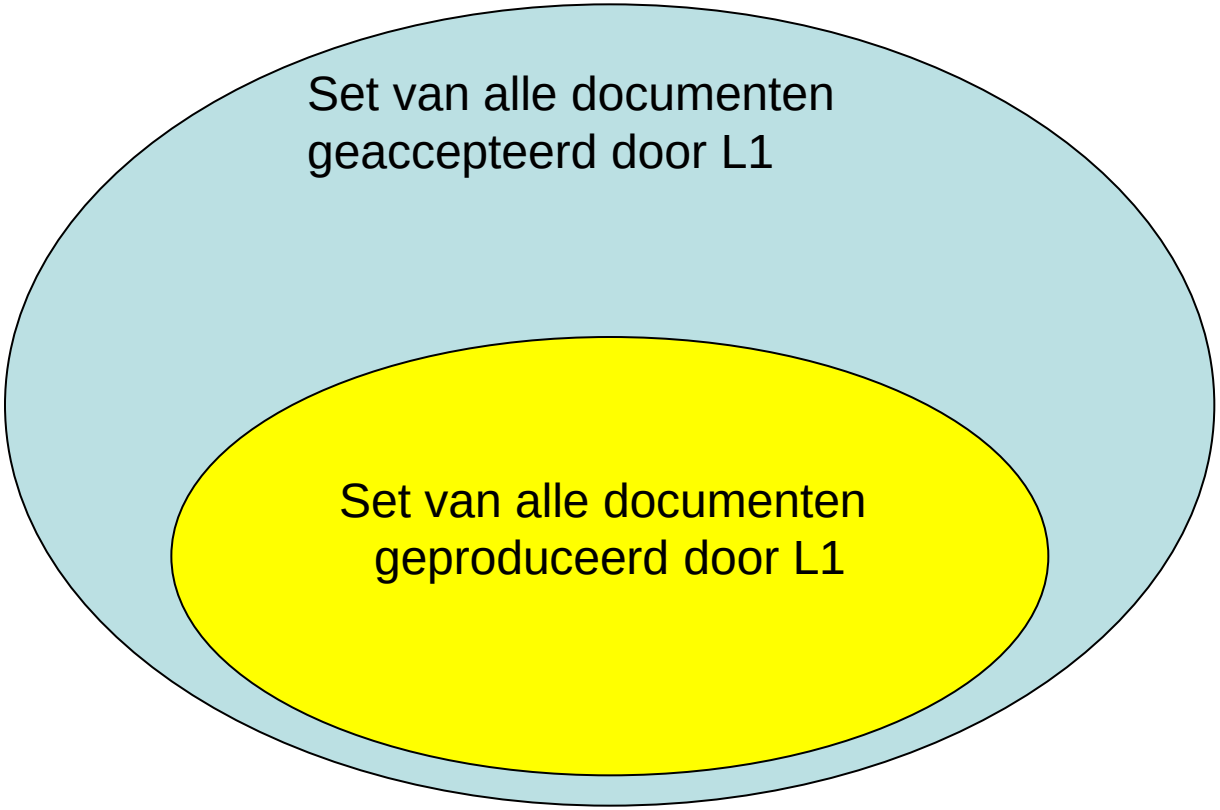
Compatibiliteit in de keten

- In de keten is er altijd een probleem, bij ontbreken FC en BC
- BC is niet zo moeilijk: oude dataformaten moeten geldig blijven
- mechanismen voor FC

Hoe krijg je forward comp.?

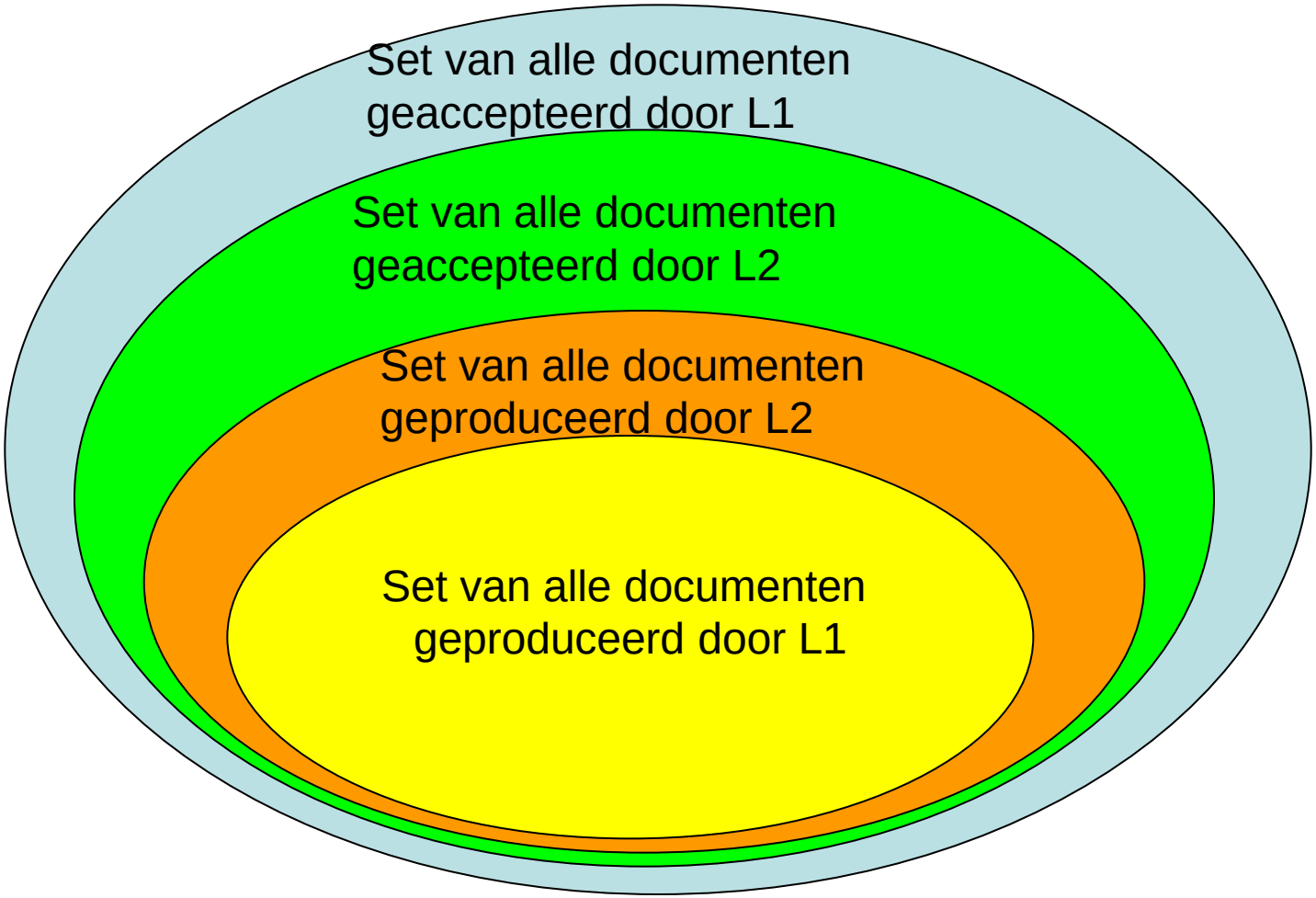
- In XML Schema met wildcard
  - voorbeeld
- “Ignore unknown”
  - in code bakken
  - met XSLT (voorbeeld)
- NVDL
  - Namespace-based Validation Dispatching Language

Wat houdt FC in?



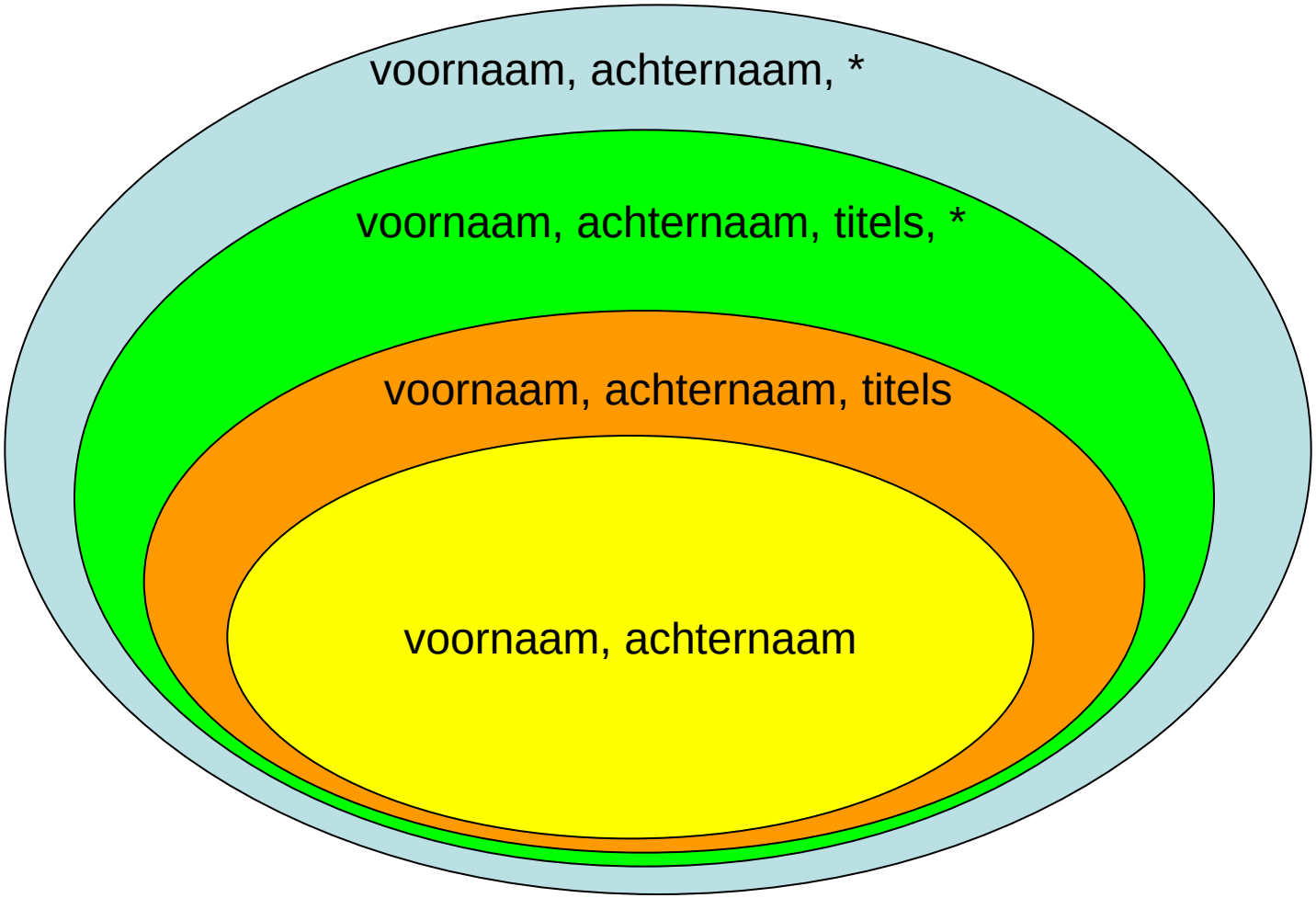
L1: voornaam, achternaam, *

Wat houdt FC in?



L1: voornaam, achternaam, * L2: voornaam, achternaam, titels, *

Wat houdt FC in?



L1: voornaam, achternaam, * L2: voornaam, achternaam, titels, *

Teveel Forward Compatibility?

Nadelen mustUnderstand

- <my:security-header soap:mustUnderstand = "1">
- alleen voor SOAP Headers
- verder toepasbaar, maar:
- extra attribuut per element
- werkt alleen voor elementen

Capability Compatibility Design Pattern

- zender: noem alle versies, ook oudere, waarvan je weet dat die je bericht mogen accepteren
- ontvanger: weet alle versies, ook oudere, die je kunt verwerken
- voorbeeld

mustUnderstand

- IgnoreUnknown is soms niet gewenst
  - medicatieberichten
  - financiële transacties
  - security
  - betrouwbaarheid
- SOAP Headers hebben mechanisme:
  - <my:security-header soap:mustUnderstand = "1">
  - overruled IgnoreUnknown principe

Vragen?