Part 2 – Core Components

Towards seamless document interoperability

Research Studio Inter-Organisational Systems
Project Public Private Interoperability

Research Studios Austria
Forschungsgesellschaft mbH

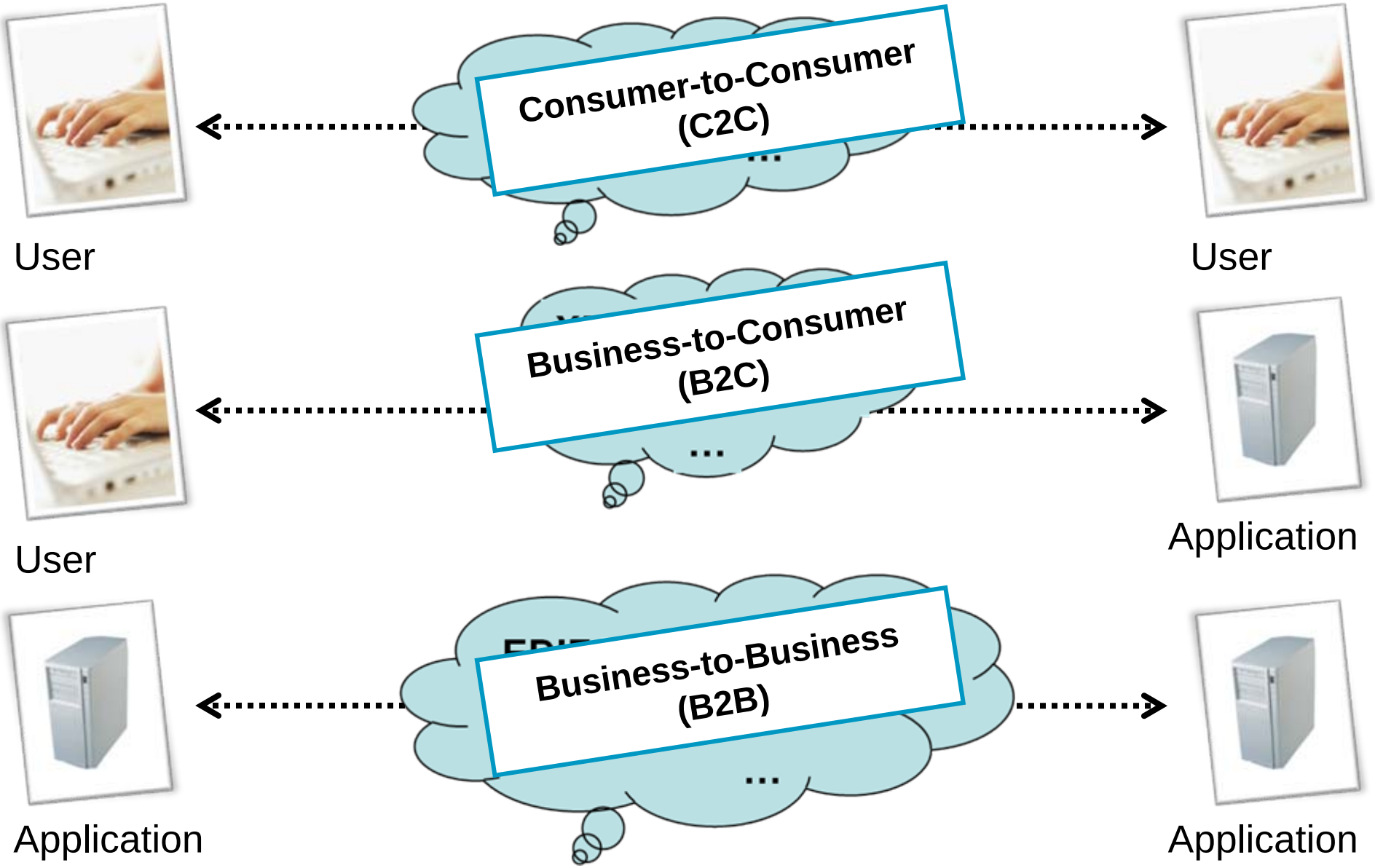
Agenda

- Motivation for a common business document exchange format
- UN/CEFACT's Core Components Technical Specification
  - Core Components
  - Business Information Entities

Different forms of data exchange

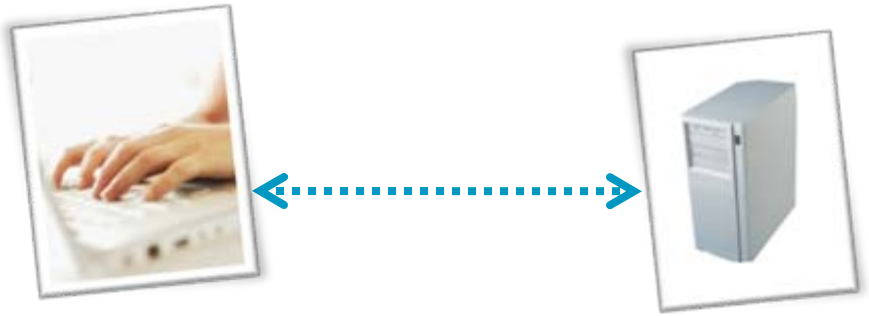
- Direct and vocal
  - Usually during a face-to-face communication
  - Mimic and gestural expression underpin the communication procedure
  - Common context
- Vocal using a transport channel
  - e.g. via radio or mobile phones
  - focus on the spoken word
- Using scripture
  - letters, books etc.
- » EDI in this context?

The goal of Electronic Data Interchange (EDI) is the seamless communication between enterprises – independent of software, hardware, or communication protocols

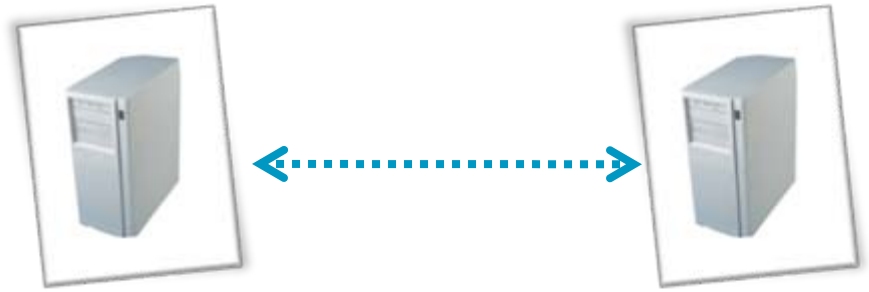


B2C vs. B2B

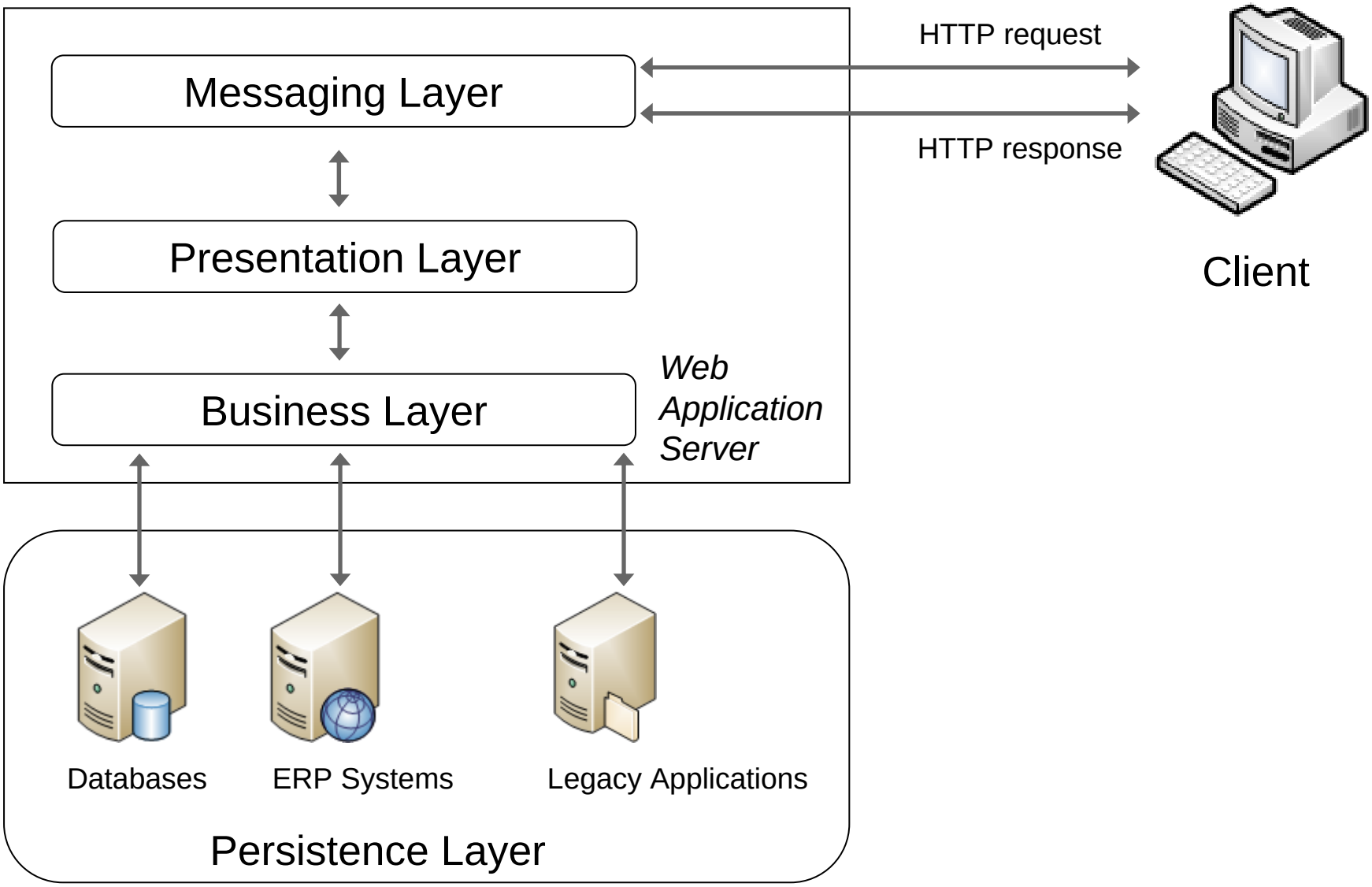
- B2C
  - Server dominates the business process
  - Consumer reacts on the fly



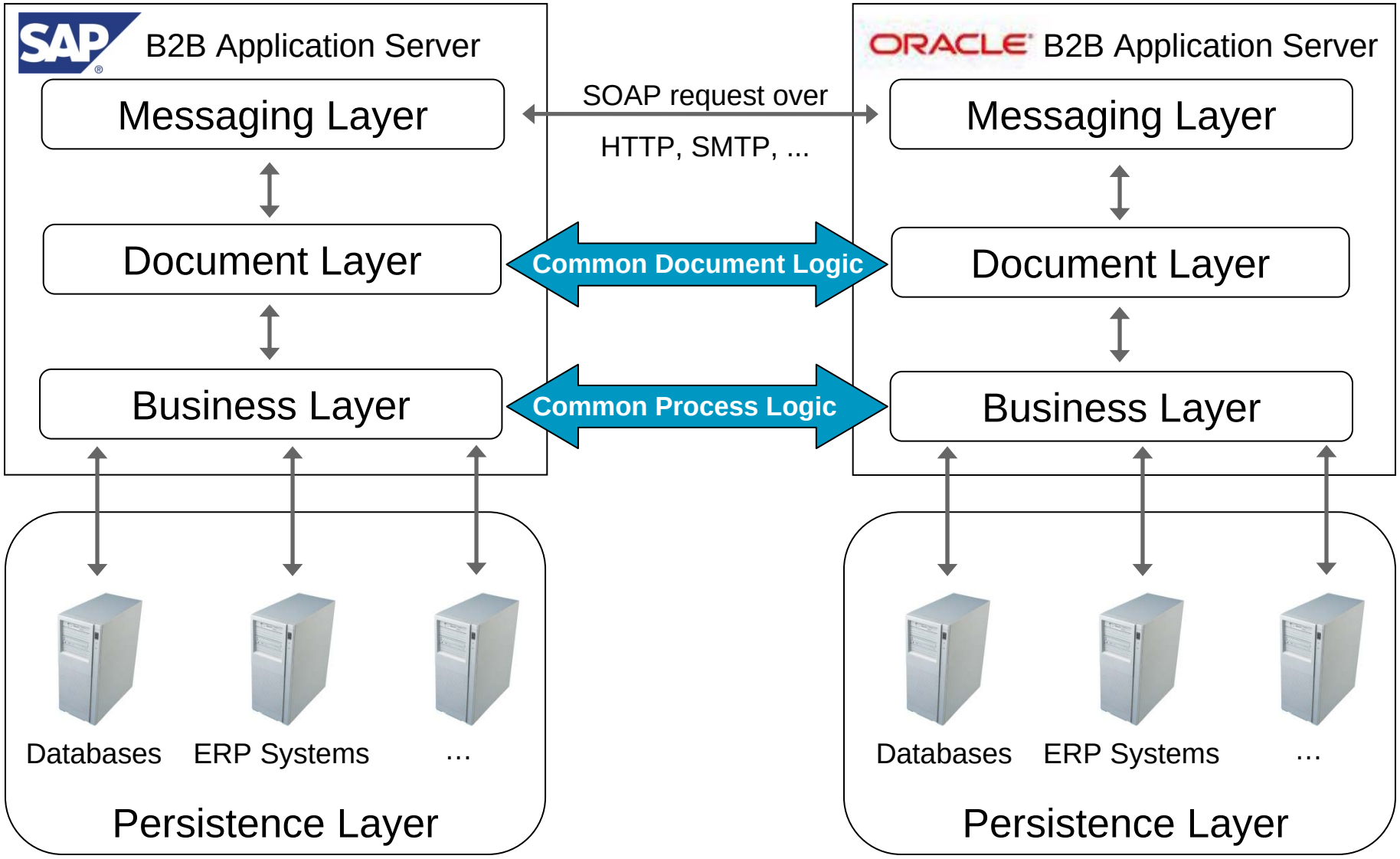
- B2B
  - Applications must interact with each other
  - Applications must follow an agreed
    - business process (UMM)
    - business document structure (CCTS)



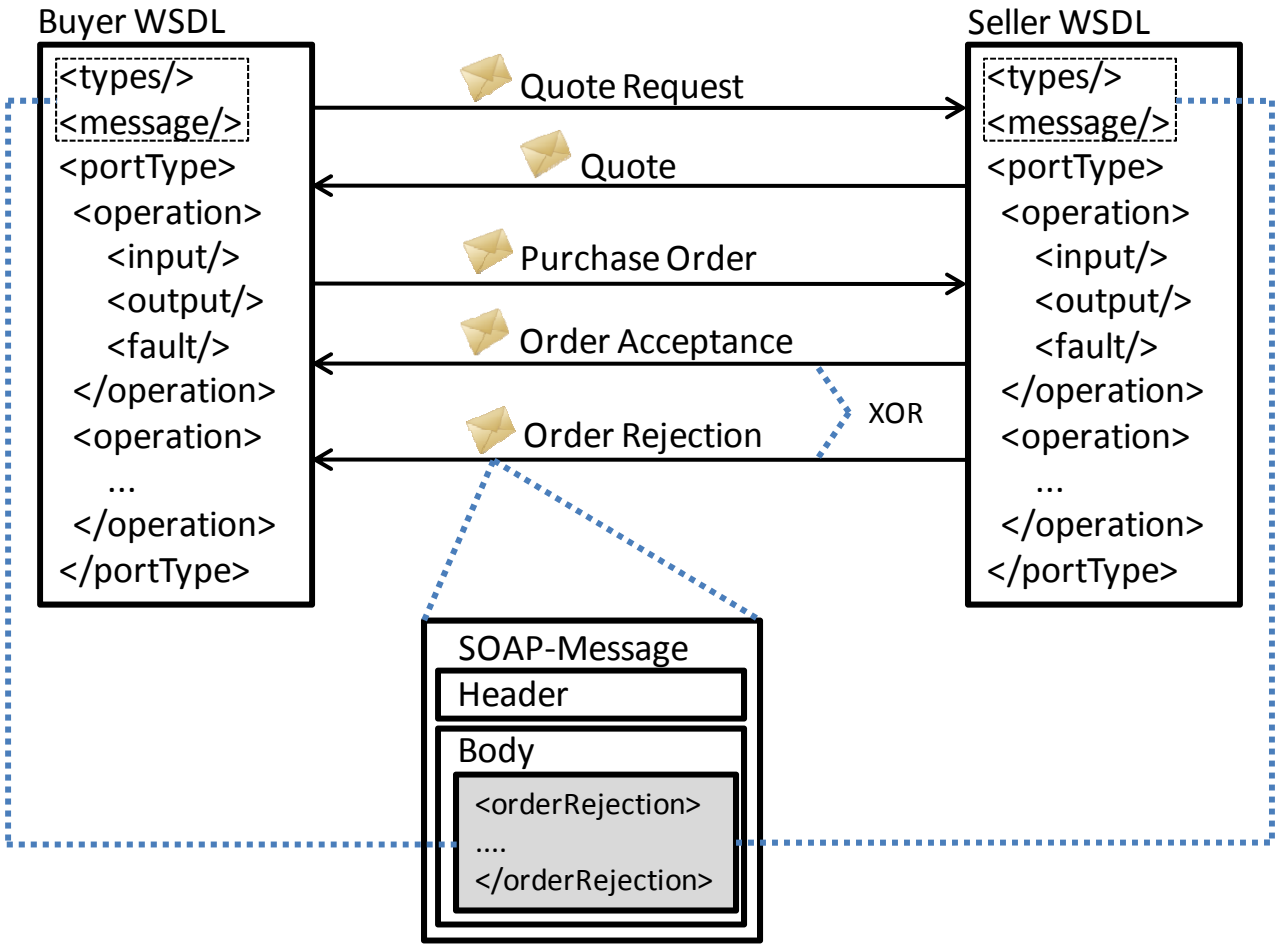
B2C – Client-Server Computing



B2B Application Computing



From processes to data – business documents as the common interface between enterprises



Is every standard an EDI standard?

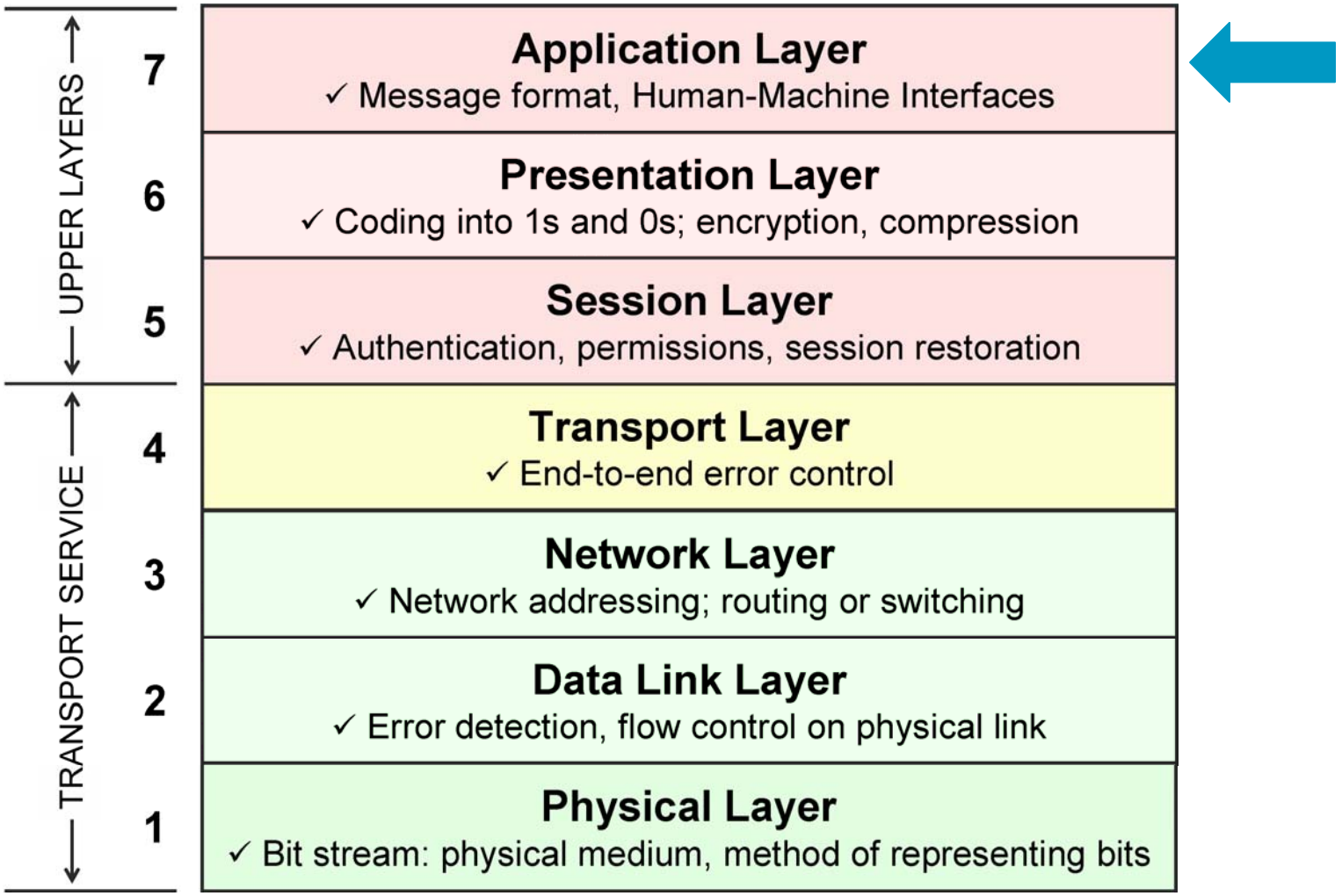
- 6d803ef64568e0191a85500f103ec39 **base16**
- <items><item>Book</item></items> **XML**
- 10101111010110000101001111100111101010 **binary**
- \BPR*C*77.77*C*ACH*CTX*01*234056789*DA*0099109999* **ANSI X.12**

EAN

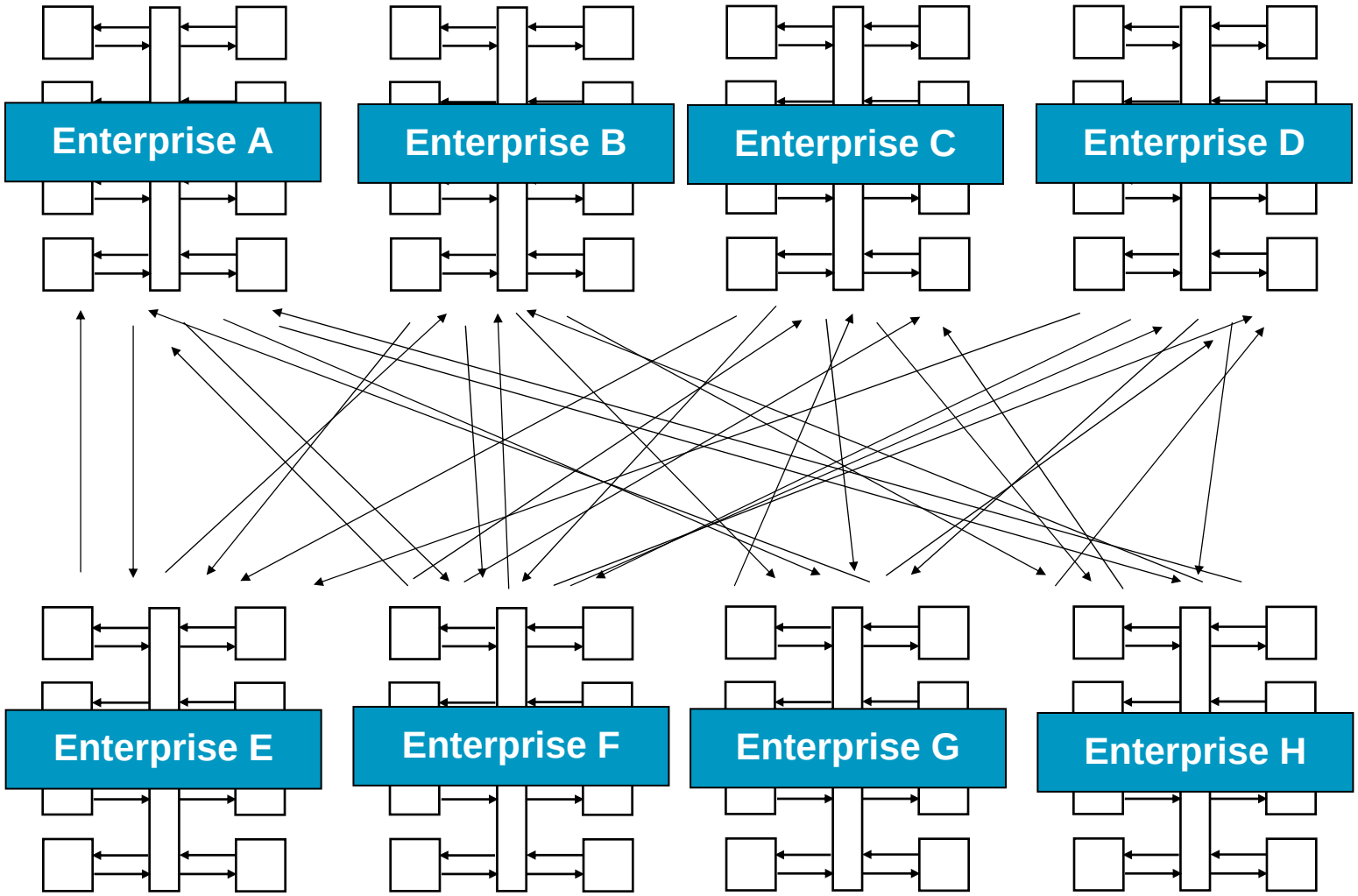
- MSH|^~\&||GA0000||VAERS PROCESSOR|20010331605||ORU^RO1|20010422GA03|T|2.3.1|||AL| **HL7**

Standards are defined on many different levels and in many different domains, however not every standard is an EDI standard.

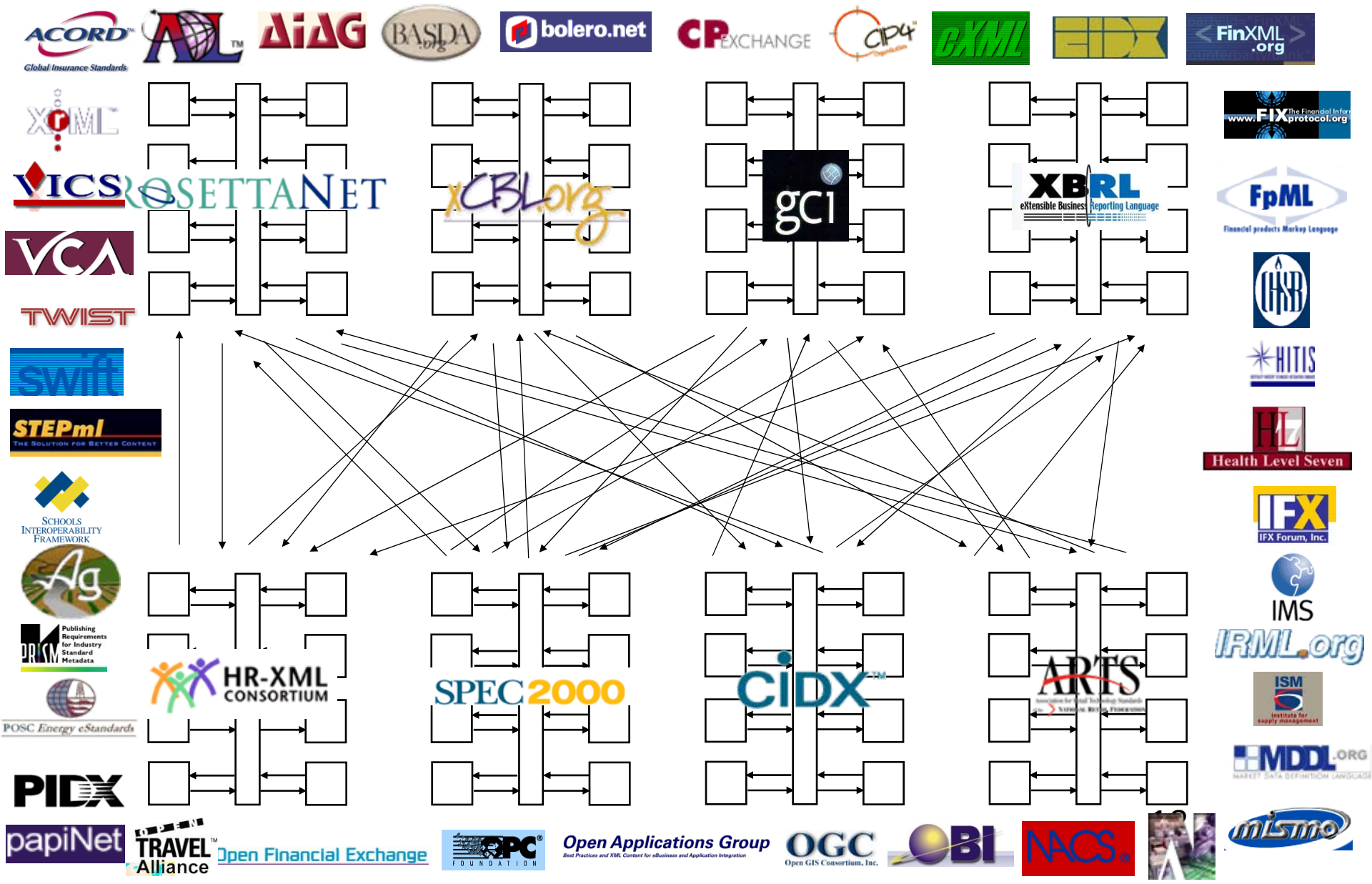
Is every standard an EDI standard?



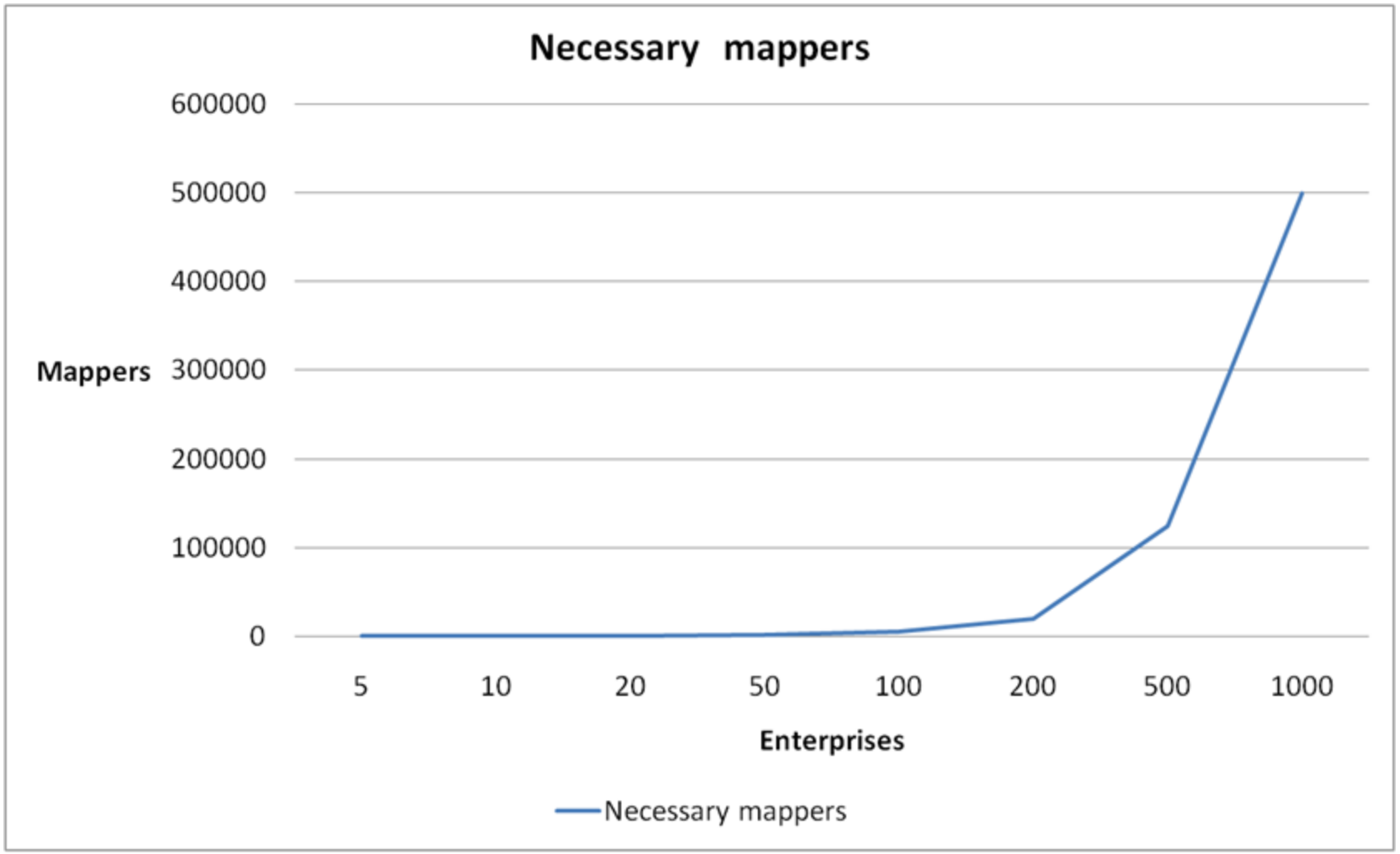
Status quo: Multitude of different interactions between enterprises



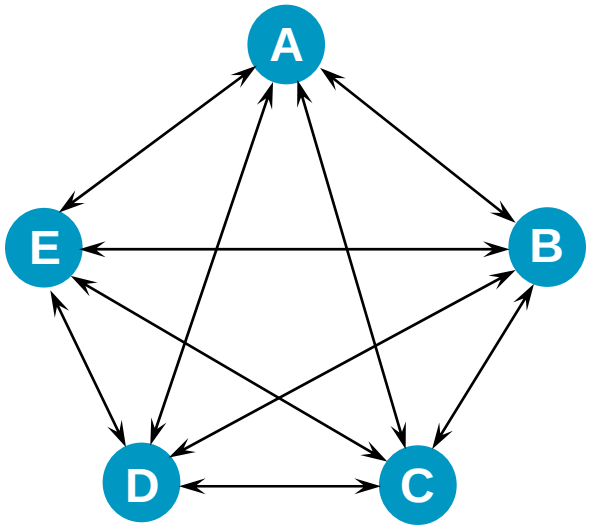
Business document standards to the rescue?

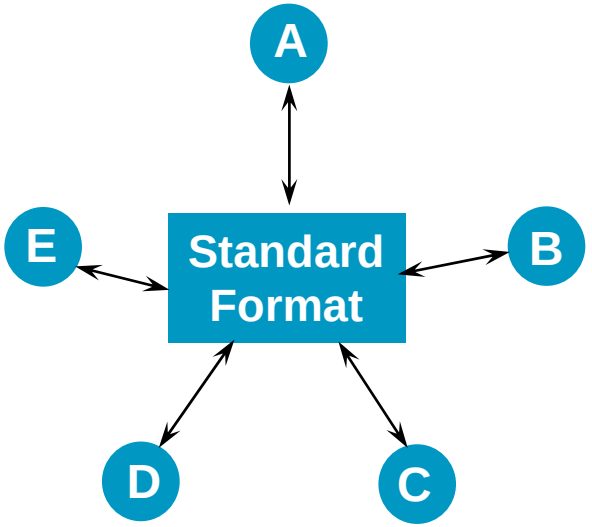


Necessary mappers

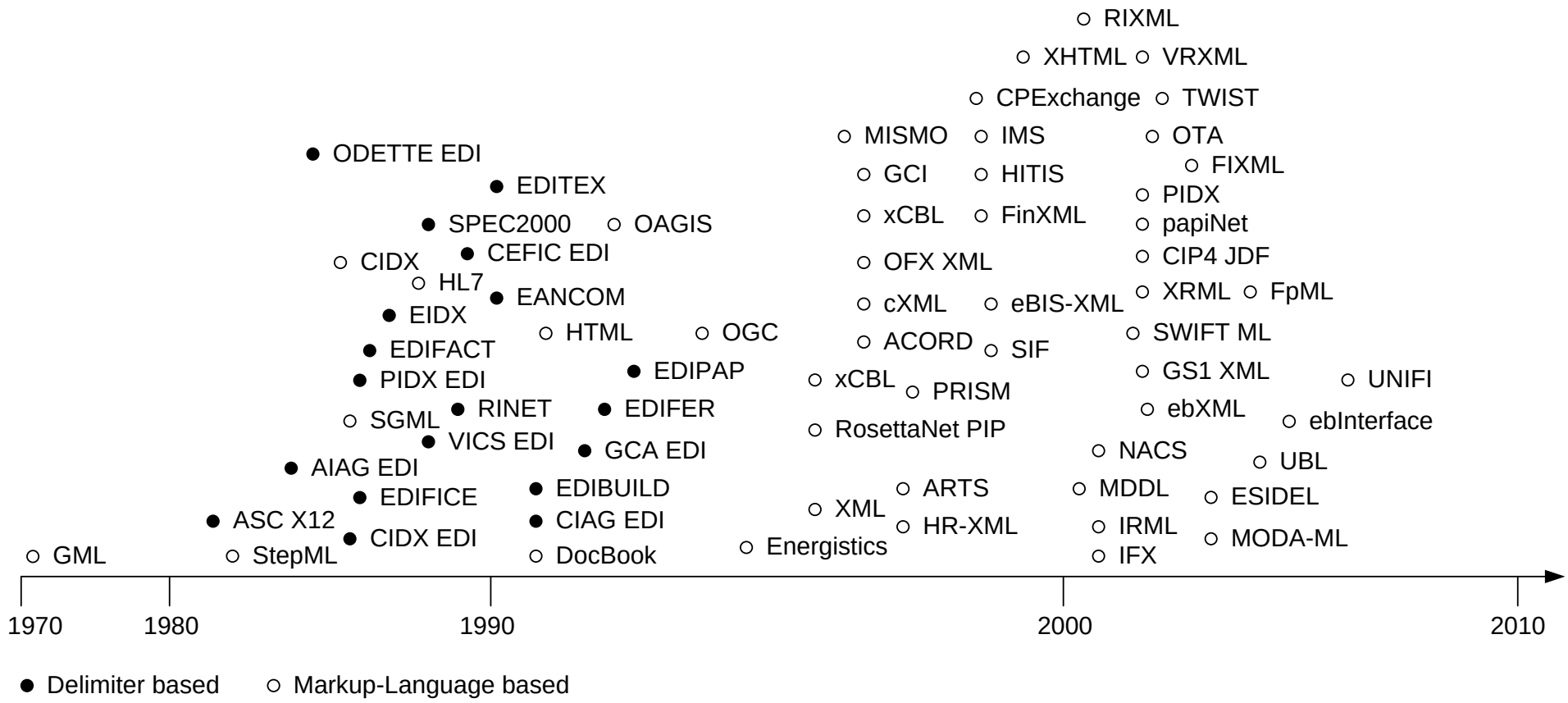


Standards: Provide a single approach for the definition of business documents





Standards over time

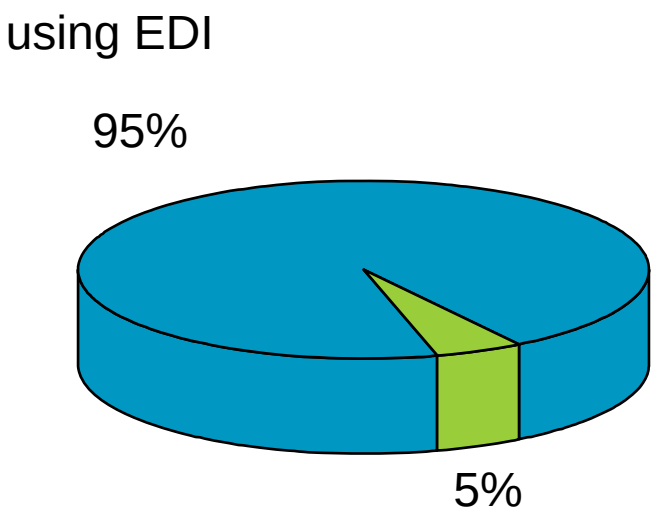


The situation in Electronic Data Interchange today

UNH+ME0000001+ORDERS:D:93A:
UN'**BGM**+220+123321'**DTM**+137:990
312:101'**DTM**+2:990503:101'**NAD**+B
Y+++Institute of Software
Technology:and Interactive
Systems:Vienna University of
Technology+Favoritenstraße 9-
11/188-3+ Vienna++1010+AT'
CTA+PE:HH:Hugo
Heuschreck'**NAD**+SE+++Hard &
Software GmbH+Wiedner
Hauptstrasse 12/8+Vienna++
1040+AT'
…
UNT+18+ME0000001'

Fortune 1000

using EDI



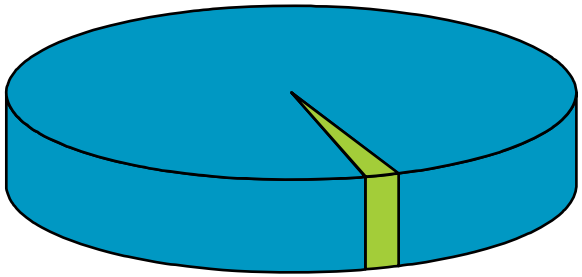
UN/EDIFACT is in use since 1987

… will not change this situation

EDI today cont'd

Small-and-medium sized enterprises

98% able to participate in e-Commerce



use EDI
2%

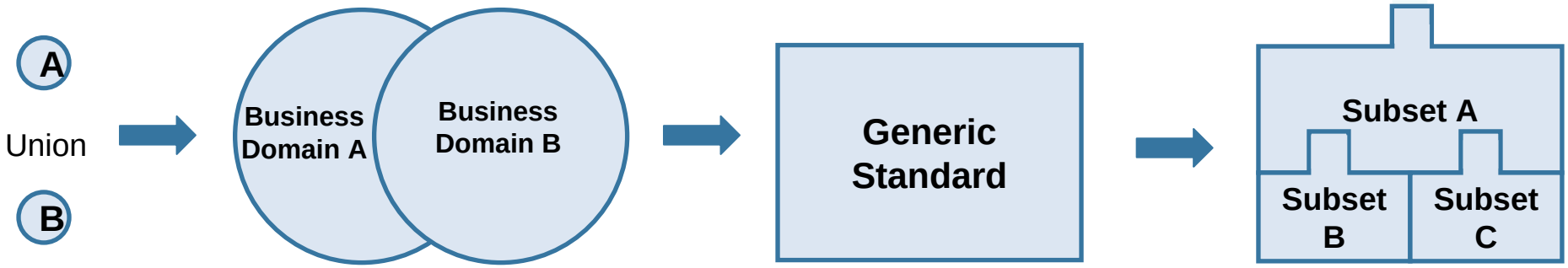
```
<?xml version="1.0" encoding="UTF-8"?>
<Invoice xmlns="urn:Invoice" xmlns:xsi="http://XMLSchema-
instance" xsi:schemaLocation="urn:Invoice.xsd">
    <ID>IN 2003/00645</ID>
    <IssueDate>2003-02-25</IssueDate>
    <TaxPointDate>2003-02-25</TaxPointDate>
    <ReferencedOrder>
        <BuyersOrderID>S03-034257</BuyersOrderID>
    <SellersOrderID>SW/F1/50156</SellersOrderID>
        <IssueDate>2003-02-03</IssueDate>
    </ReferencedOrder>
    <ReferencedDespatchAdvice>
        <ID>DEL-03/55-712</ID>
        <IssueDate>2003-02-24</IssueDate>
    </ReferencedDespatchAdvice>
    <BuyerParty>
        <ID/>AB123712</ID>
        <PartyName>
            <Name>Jerry Builder plc</Name>
        </PartyName>
        ...
```

EDI is too expensive to implement

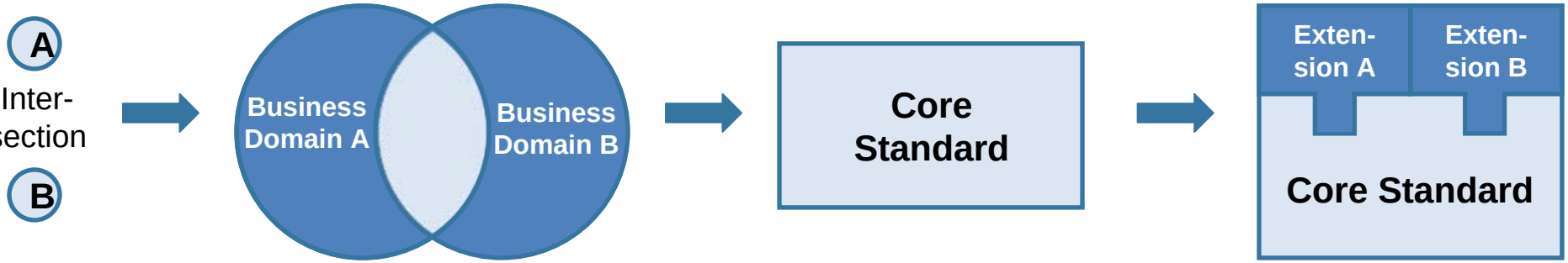
…lean XML based standard would help

The two main business document standard paradigms

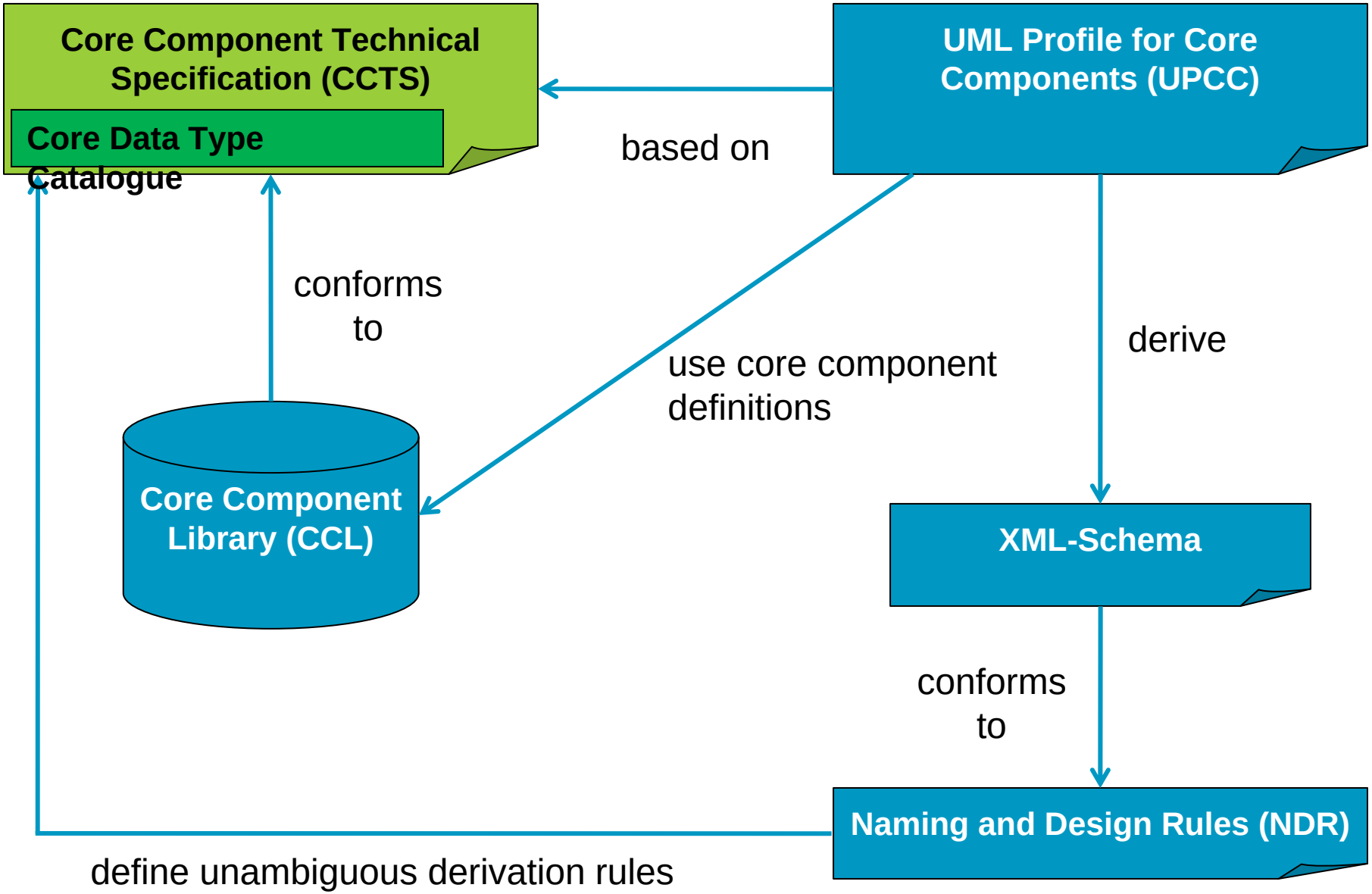
Top-down business document standards



Bottom-up business document standards



Overview of Core Component related UN/CEFACT specifications



Core Components Technical Specification (CCTS)

- Official UN/CEFACT version 2.01
- Current specification under development: **CCTS 3.0**

- Defines the meta-level concepts of Core Components
- Written in implementation neutral English prose
- Visualization of Core Component Concepts using Unified Modeling Language (UML)

- Editor: Mark Crawford, SAP Labs LLC, (USA)

UN/CEFACT's Core Components – Definition of reusable building blocks

- Semantic building blocks for the definition of business document data
- Context free – reuse in multiple business sectors
- Customization of generic core components to specific business sectors and application domains

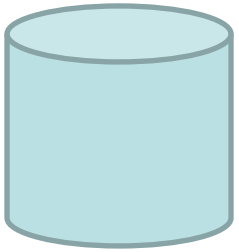
Definition of resuable building blocks

Core Components

Business document

Resuse of document building blocks

UN/CEFACT



Library

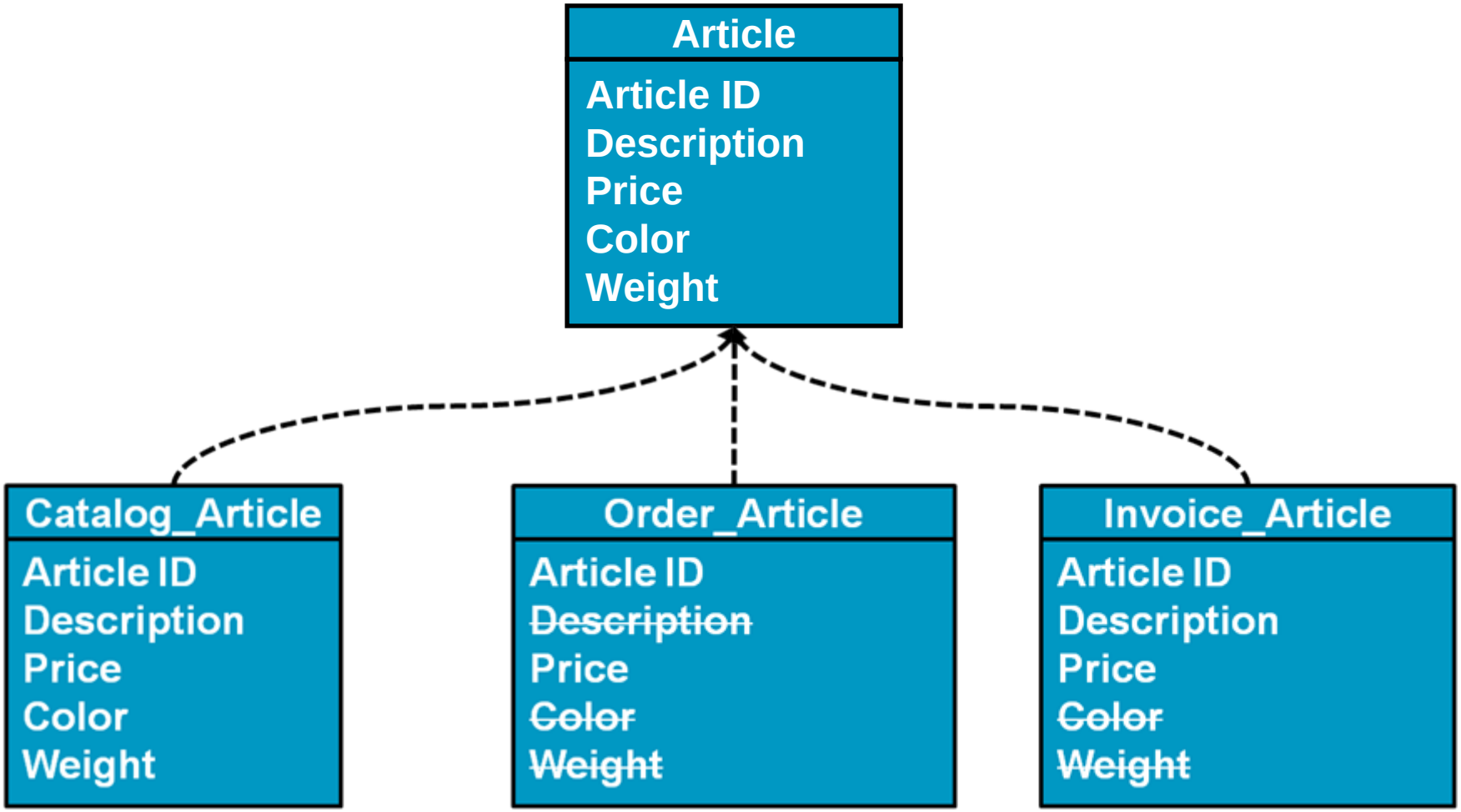
Enterprise C

Enterprise A

Enterprise B

Enterprise X

Reduction of complexity through resuse of components



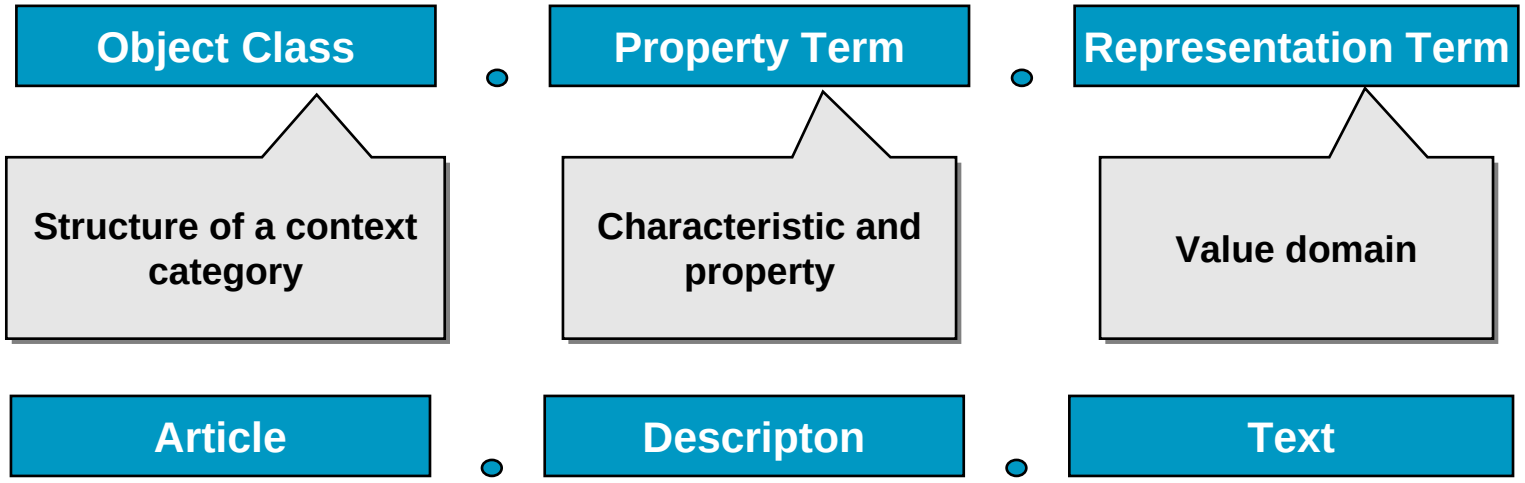
Contextualization by omitting non-used elements

Core Components at a glance

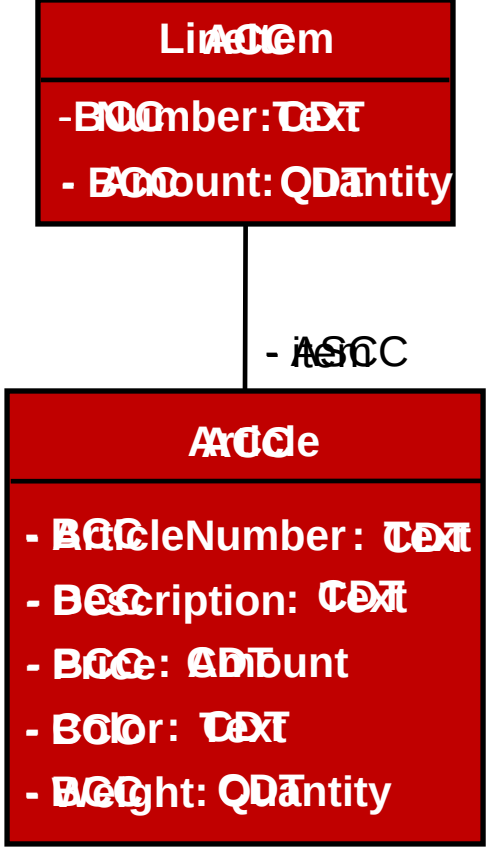
- Semantic building blocks
  - Reference data models
  - Business messages
- Based on a common semantic basis
  - Core Component Library (CCL)
- Implementation neutral definition
- Used to be part of the ebXML standard framework
- Today a dedicated UN/CEFACT project
  - (Core Component Technical Specification)

Core Component structure and semantics

- Core Component Names are referred to as **Dictionary Entry Names (DEN)**
- DENs are based on ISO 11179 (formerly known as ISO/IEC 11179 Metadata Registry (MDR) standard)
  - Standardized naming
  - Facilitates storage and retrieval of information
  - Facilitates common understanding of information



Core components in one slide



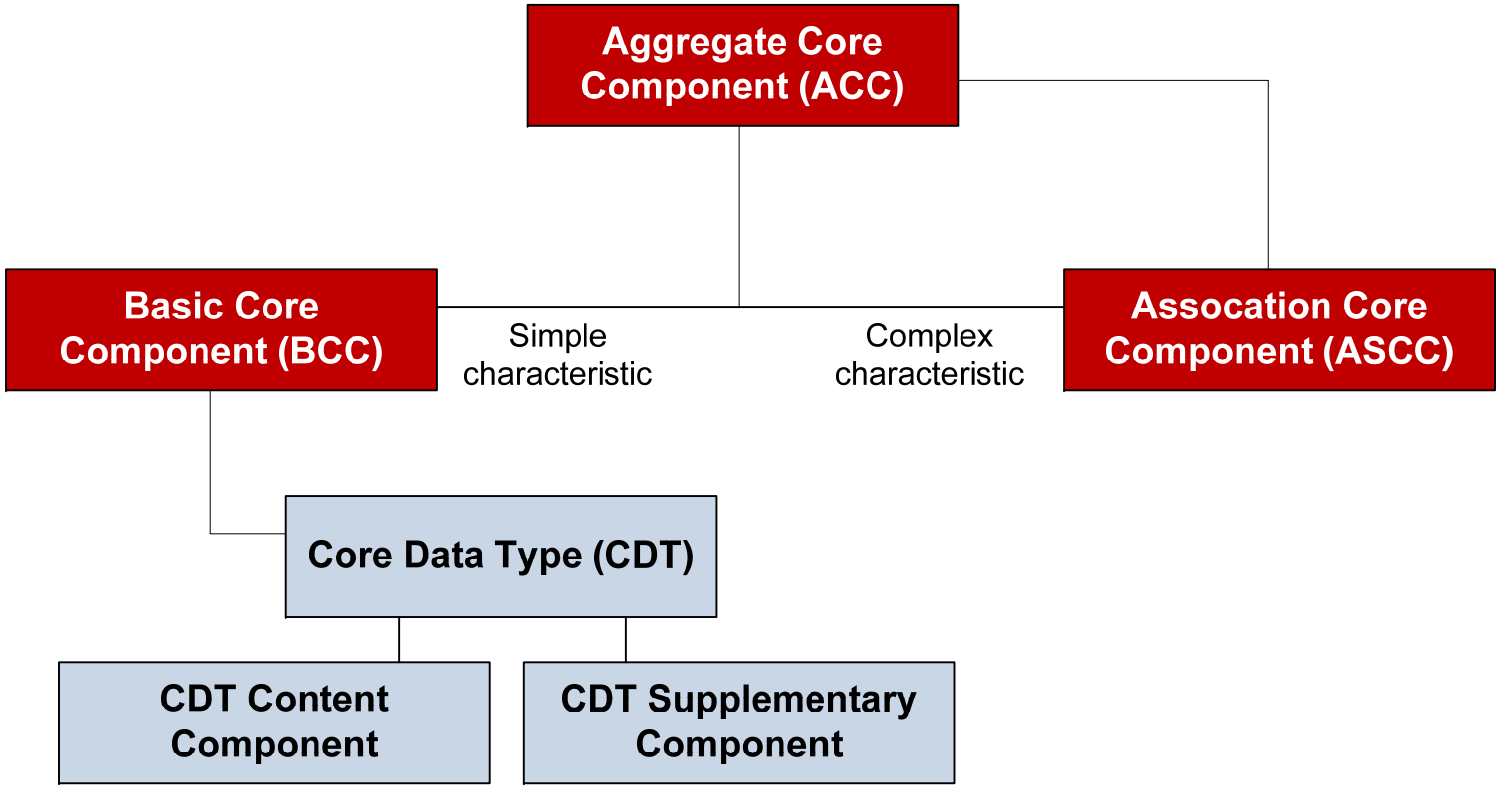
- Identification of objects
- Identification of object properties

- Two types of properties
  - Simple properties (text, number, date)
  - Complex properties (other objects)

- Object type = **A**ggregate **C**ore **C**omponent
- Simple Property = **B**asic **C**ore **C**omponent
- Simple Property Data Type = **C**ore **D**ata **T**ype
- Complex Property = **AS**sociation **C**ore **C**omponent

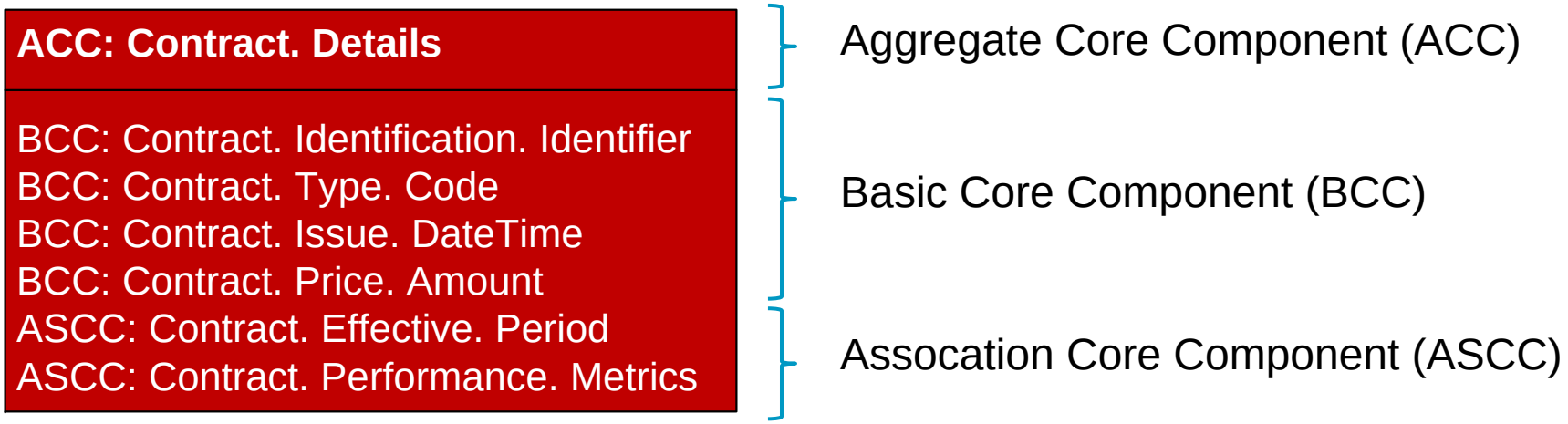
Core components

- Foundational concept of the Core Component Technical Specification
- Used for all aspects of data and information modeling
- Distinction between three different categories of core components



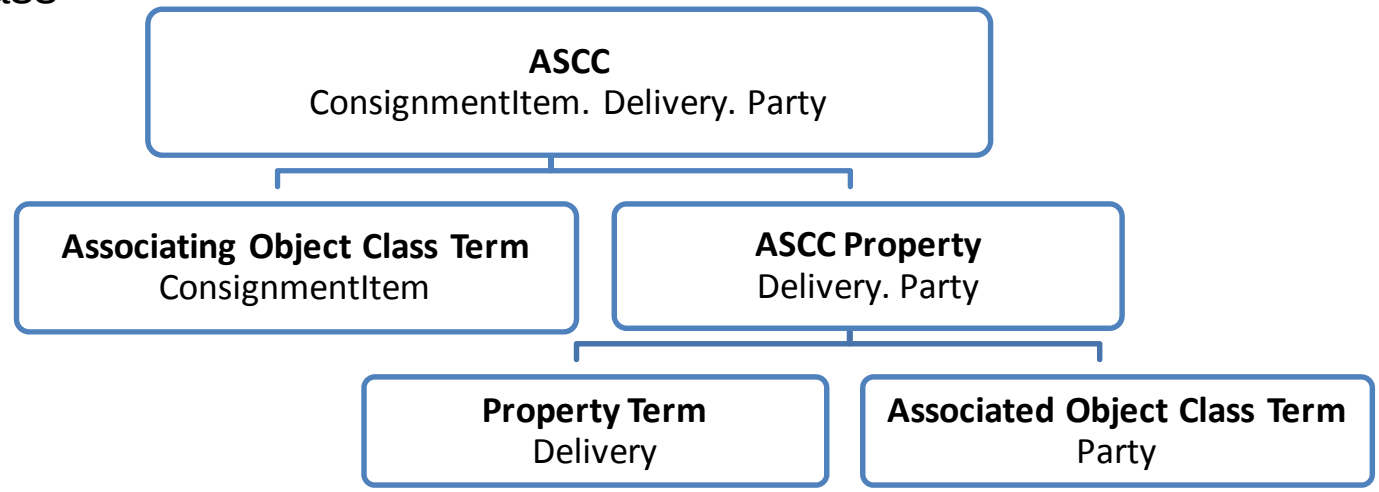
Aggregate Core Component (ACC)

- Collection of related pieces of information that together convey a distinct meaning
- Independent of any business context
- In data modeling terms an ACC is the representation of an entity or class with attributes and associations

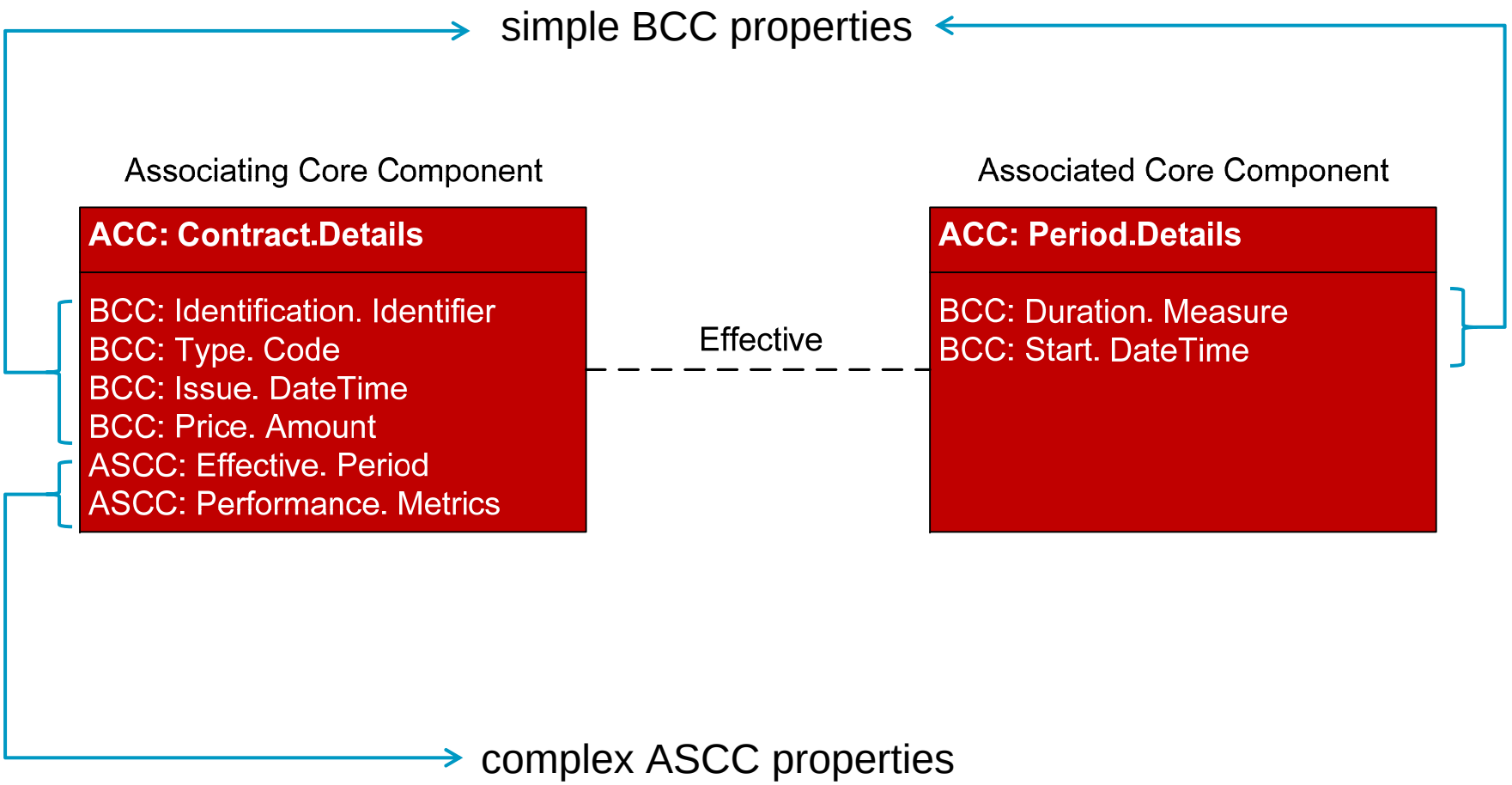


Association Core Component (ASCC)

- Defines a role in an association between on ACC (associating ACC) and another ACC (associated ACC)
- ASCC consists of an **ASCC property** plus the **object class** of the parent ACC
- ASCC property consists of a **property term** that expresses the nature of the association and the name of the **object class** being associated
- An ASCC property is reusable across object classes. Once it has been given the object class of the parent ACC, it becomes an ASCC that is unique for a given object class

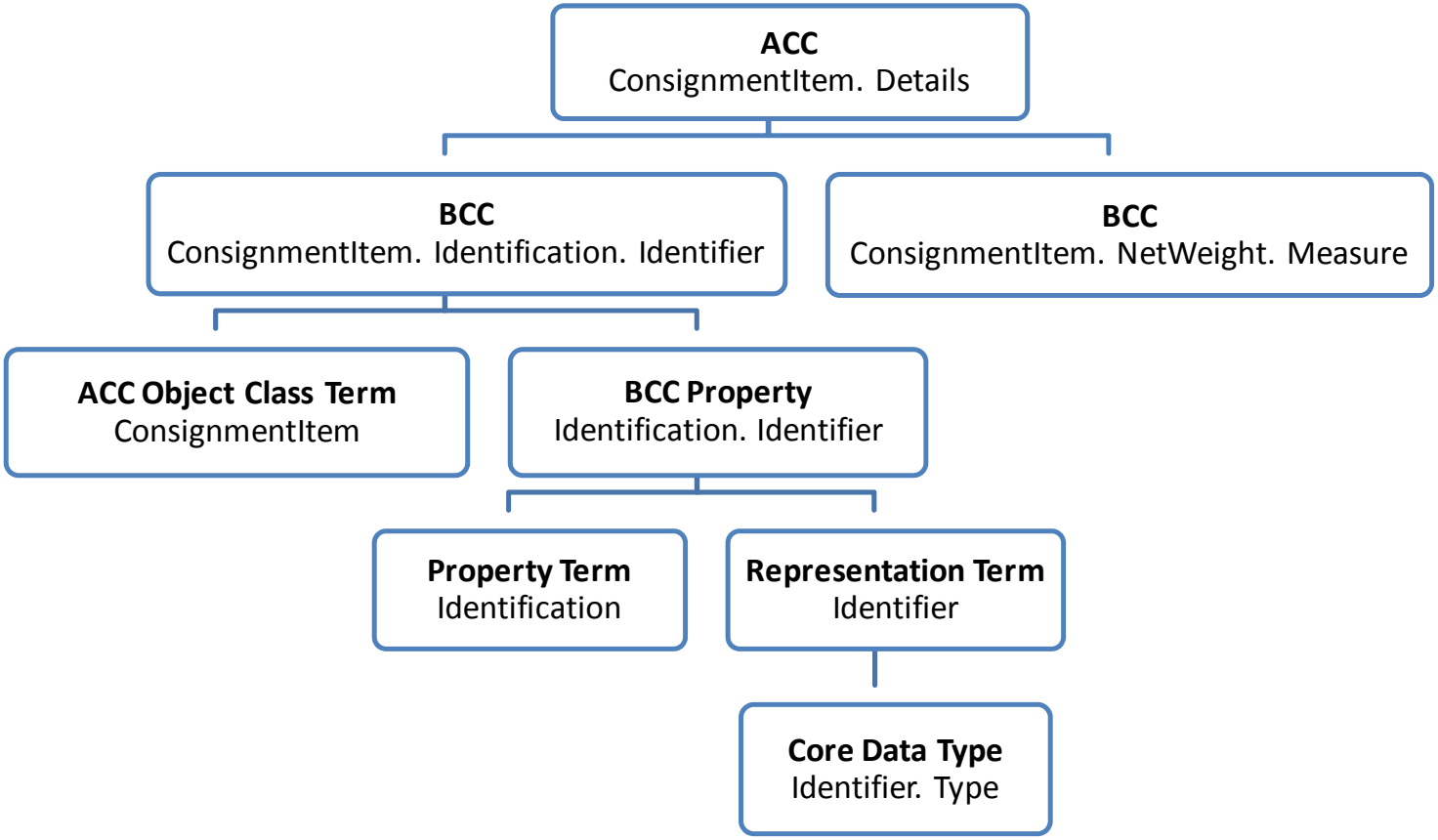


Association Core Component (ASCC)



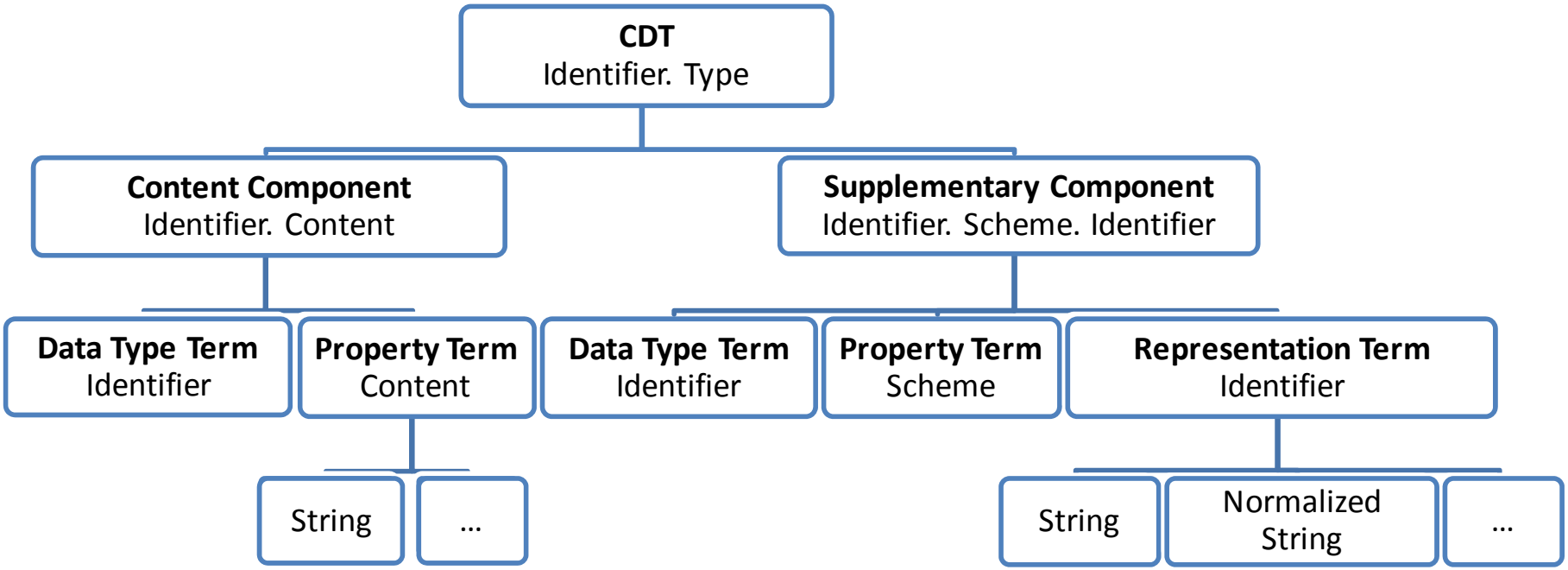
Basic Core Component (BCC)

- Represents a property of an ACC
- A BCC consists of **a BCC property** plus the **object class** of the parent ACC
- A BCC is reusable across object classes
- Once it has been given the object class of a parent ACC, it becomes a BCC that is unique to the object class to which it is assigned

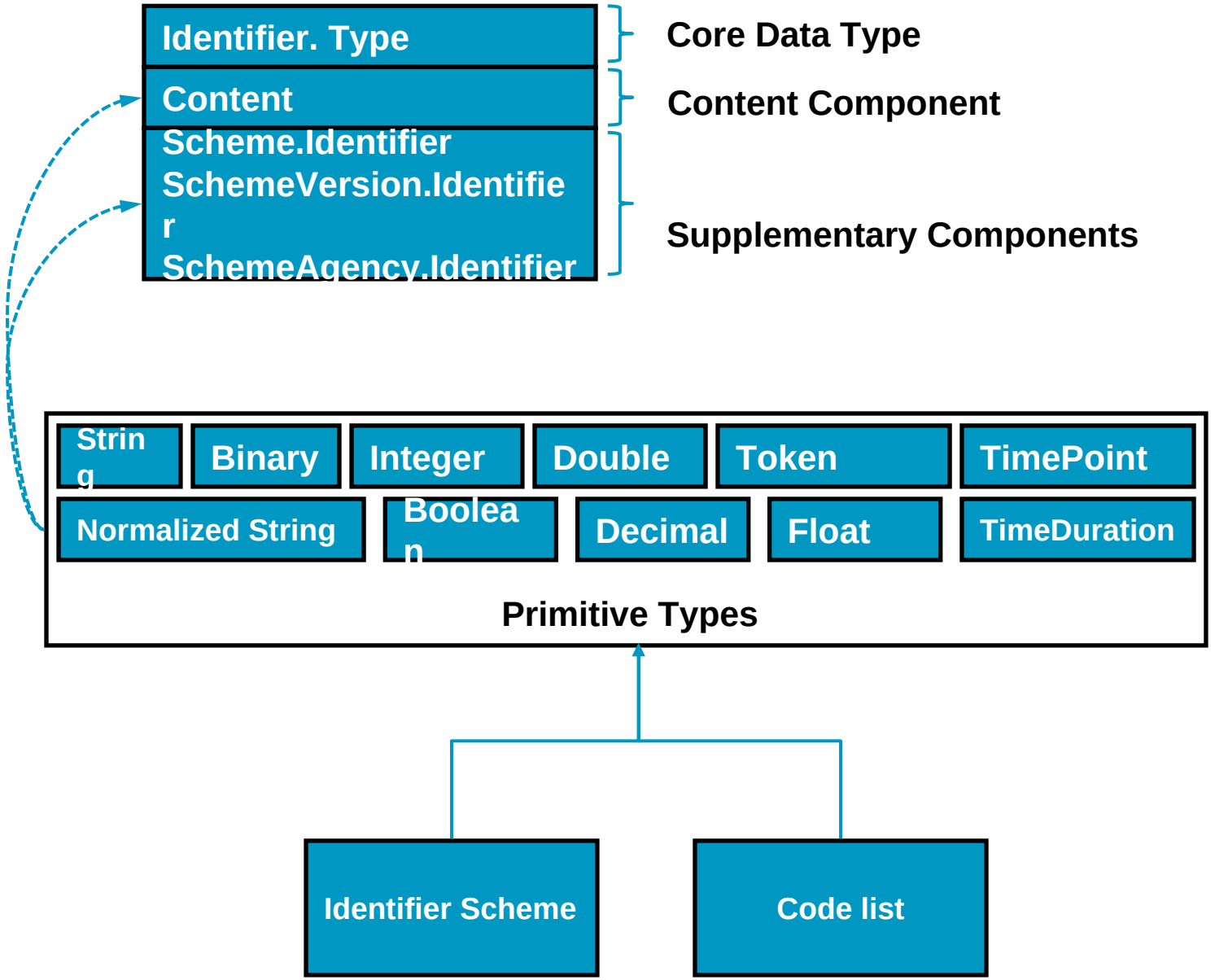


Core Data Types (CDT)

- A CDT represents the full range of values that can be used for representing an instance of a **Basic Core Component**
- Every CDT has a **content component** and zero or more **supplementary components**



Core data type (CDT) – example



Core Data Types (CDT)
UN/CEFACT Data Type Catalogue 3.0

- Released with every Core Component Technical Specification
- Defines allowed Core Data Types (CDT) and Primitive Types

Binary
Boolean
Decimal
Double
Float
Integer
Normalized String
String
TimeDuration
TimePoint
Token

11 Primitive Types

Amount
Binary Object
Code
Date
Date Time
Duration
Graphic
Identifier
Indicator
Measure
Name

Ordinal
Percent
Picture
Quantity
Rate
Ratio
Sound
Text
Time
Value
Video

22 Core Data Types (CDT)

UN/CEFACT Data Type Catalogue – primitive types

- For each primitive type a set of allowed **facets** is defined
- Facets may further restrict a primitive type

Primitive Type	Name	Description	Allowed Facets	Remarks
Binary	Binary	Binary is a finite sequence of binary digits (bits)	Enumeration Length Minimum Length Maximum Length Pattern	
Boolean	Boolean	Boolean denotes a logical condition through a predefined enumeration of the literals true (The Boolean condition is satisfied) and false (The Boolean condition is not satisfied).	None	Allowed literals = [true/false]
Decimal	Decimal	Decimal is a subset of the real numbers, which can be represented by decimal numerals	Enumeration Fractional Digits Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern Total Digits	Look for non-XSD reference for expanding the definition.
Double	Double	Double is the IEEE double precision 64 bits floating point type	Enumeration Fractional Digits Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern Total Digits	

Allowed Facets according to UN/CEFACT Data Type Catalogue 3.0

Facet Type	Facet Name	Description	Value
Enumeration	Enumeration	Defines a specified set of values	A set of values from the value domain of the data type.
FractionalDigits	Fractional Digits	Defines the maximum number of fractional digits to be used.	Non Negative Integer
Length	Length	Defines the number of units of length of the data type.	Non Negative Integer
MaximumExclusive	Maximum Exclusive	Defines the upper limit of the range of allowed values. The upper limit is no allowed value. [Note] This format restriction shall not be used in combination with the Maximum Inclusive format restriction.	Value from the value domain of the data type
MaximumInclusive	Maximum Inclusive	Defines the upper limit of the range of allowed values. The upper limit is also an allowed value.	Value from the value domain of the data type
MaximumLength	Maximum Length	Defines the maximum number of units of length. [Note] This format restriction shall not be used in combination with the Length format restriction	Non Negative Integer
MinimumLength	Minimum Length	Defines the minimum number of units of length. [Note] This format restriction shall not be used in combination with the Length format restriction.	Non Negative Integer
MinimumExclusive	Minimum Exclusive	Defines the lower limit of the range of allowed values. The lower limit is no allowed value. [Note] This format restriction shall not be used in combination with the Minimum Inclusive format restriction.	Value from the value domain of the data type
MinimumInclusive	Minimum Inclusive	Defines the lower limit of the range of allowed values. The lower limit is also an allowed value.	Value from the value domain of the data type
Pattern	Pattern	Defines a constraint on the lexical space of a datatype to literals in a specific pattern.	Regular Expression
TotalDigits	Total Digits	Defines a maximum number of digits to be used.	Positive Integer

UN/CEFACT Data Type Catalogue CDT example

1.1.1 Usage Guidance

Identifier. Type is used to represent objects to enable a common identification of objects. The common identification should be based on the common identification scheme concept used to create the individual identifiers. Typical examples are "*Product_ Identifier. Type*", "*Order_ Identifier. Type*". The *"Identifier. Type"* should be used in case of an infinite set of objects, and *Code. Type* should be used in case of a finite case of allowed values.

1.1.2 Identifier. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Identifier. Content	Identifier	Content	Normalized String String Token	1..1	A character string used to uniquely identify one instance of an object within an identification scheme that is managed by an agency.	UNDTRTB546

1.1.3 Identifier. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Identifier. Scheme. Identifier	Identifier	Scheme	Identifier	Normalized String String Token	0..1	The identification of the identifier scheme.	UNDT230W43 UNDTRTB546	It is required to have common concepts for the definition of identifier scheme patterns. The primitive is specified by the Identification Scheme.
Identifier. Scheme Version. Identifier	Identifier	Scheme Version	Identifier	Normalized String String Token	0..1	The identification of the version of the identifier scheme	UNDT230W43 UNDTRTB546	The primitive is specified by the Identification Scheme.
Identifier. Scheme Agency. Identifier	Identifier	Scheme Agency	Identifier	Normalized String String Token	0..1	The identification of the agency that manages the identifier scheme	UNDT230W43 UNDTRTB546	The primitive is specified by the Identification Scheme.

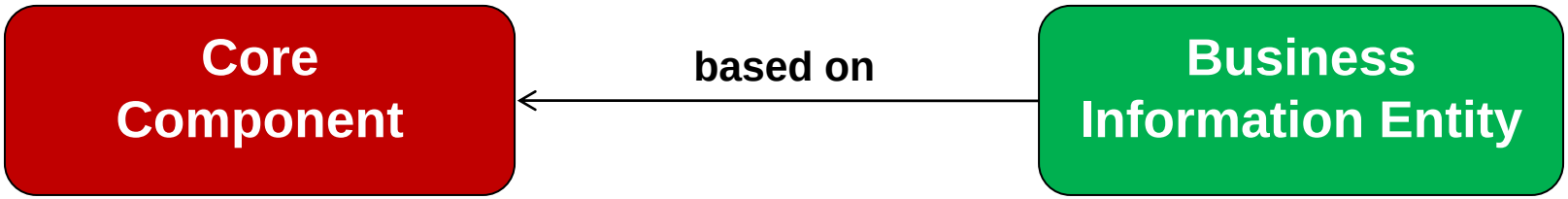
Restriction of a primitive type code list and identifier schemes

- **Code lists** represent a set of pre-defined values, based on a primitive type
  - typically enumerated
  - Example: ISO Country Code Lists

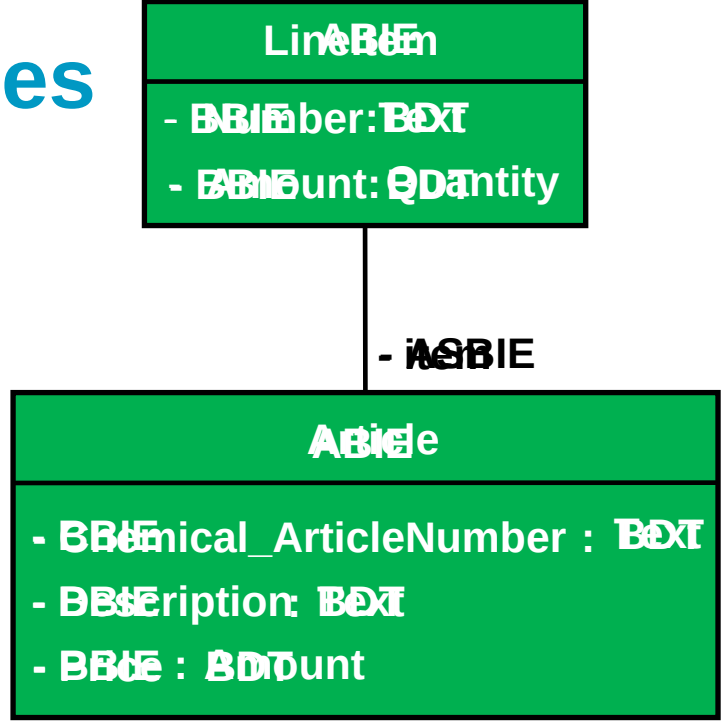
- Identifier schemes provide production rules for a certain primitive type (in most cases String)
  - typically not enumerated
  - provided e.g. as regular expression
  - Example: Bank account numbers

From Core Components to Business Information Entities

- Core Components act as conceptual models that are used to define **business information entities** (BIE)
- Business Information entities are created
  - by the application of context
  - by restricting a generic core component
- Business Information Entities and Core Components are complementary in many respects
- A Business Information Entity must always be based on a Core Component



Business Information Entities in one slide

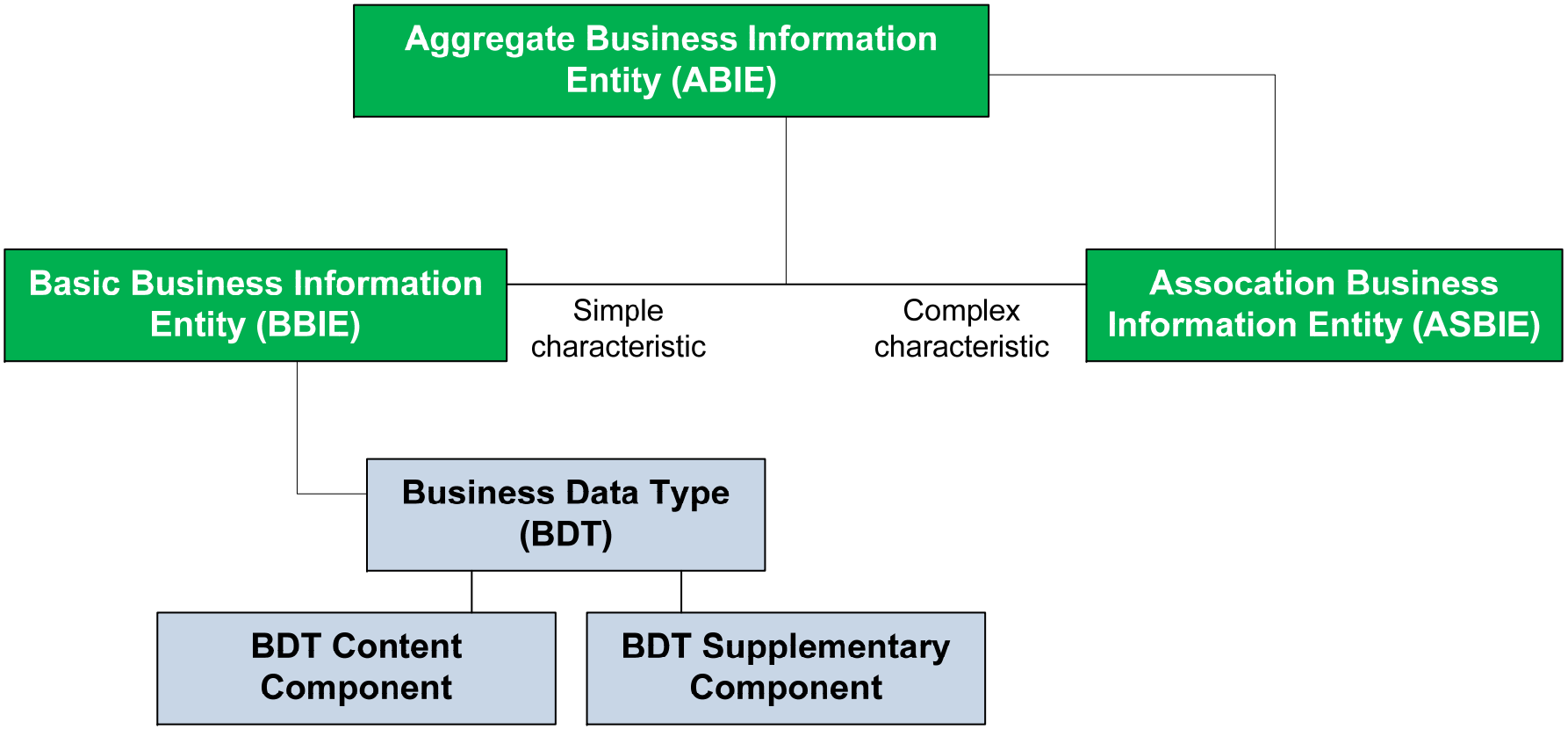


- Core Components in a specific context
- Qualifiers help to distinguish BIEs

- Two different type of properties
  - Simple properties (text, number, date)
  - Complex properties (other objects)

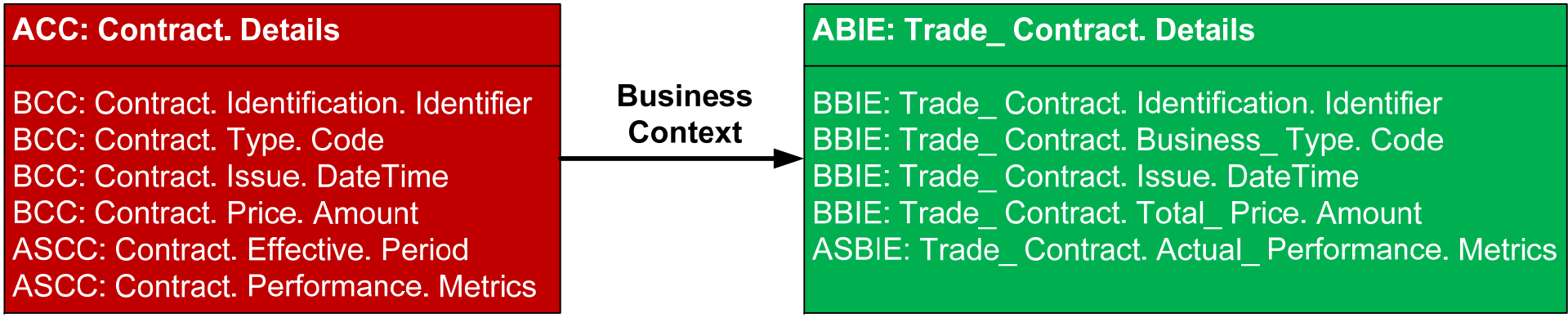
- Object type = **A**ggregate **B**usiness **I**nformation **E**ntity
- Simple Property = **B**asic **B**usiness **I**nformation **E**ntity
- Simple Property DT = **B**usiness **D**ata **T**ype
- Complex Property = **AS**sociation **B**usiness **I**nformation **E**ntity

Business Information Entities



Aggregate Business Information Entity (ABIE)

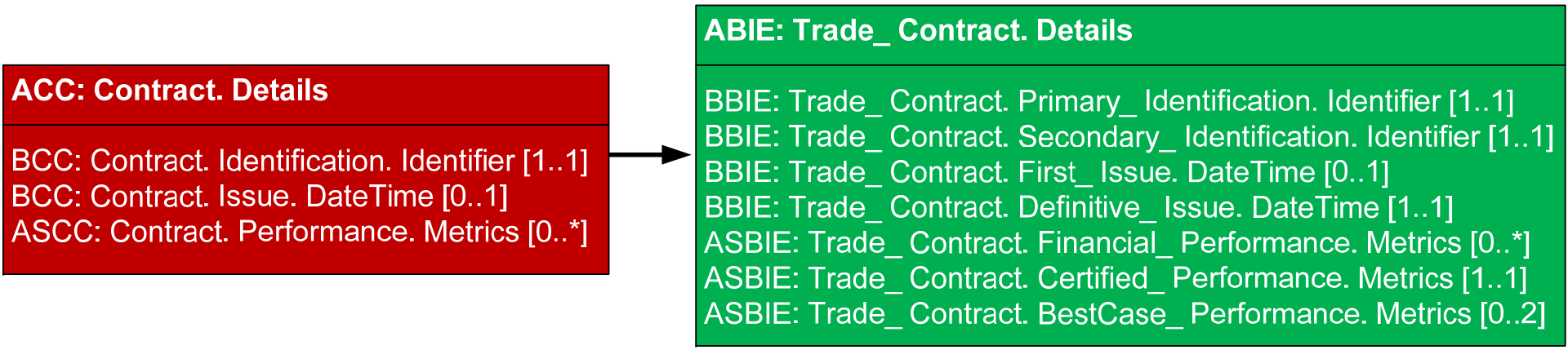
- An Aggregate Business Information Entity (ABIE) is derived from an ACC and refines the business semantic definition for a **specific business context**
- An ABIE may be **qualified** at the object class level, its properties may be qualified at the property term level



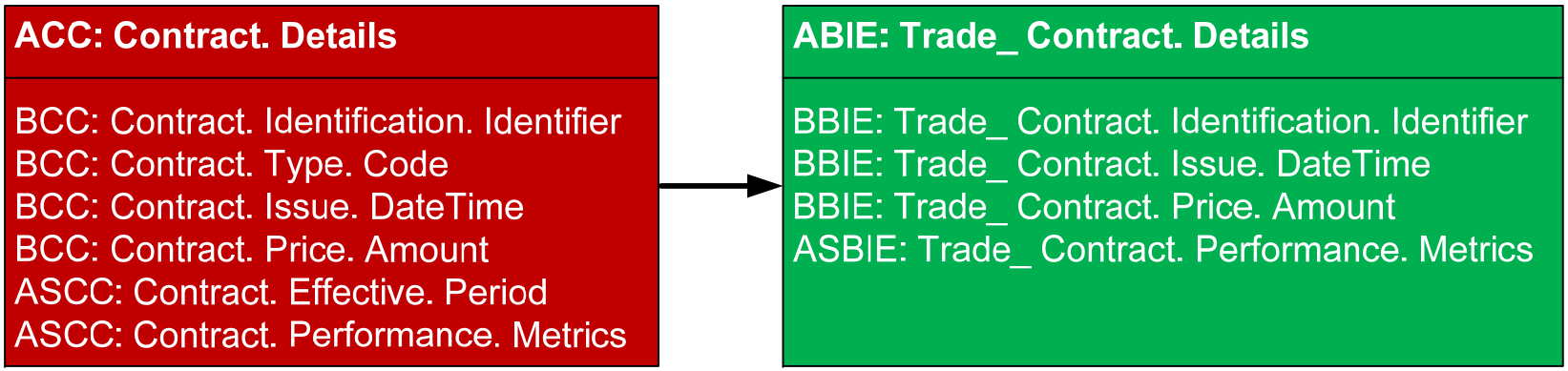
From Aggregate Core Components (ACC) to Aggreate Business Information Entities (ABIE)

- An ABIE can reflect restrictions on the content model of the underlying ACC through
  - Restrictions on the cardinality of the BCCs and ASCCs
  - Use and non-use of individual BCCs and ASCCs
  - Qualification of individual ASCC and BCC properties
  - Restrictions on the content model of an associated ACC for an ASCC
  - Restrictions on the data type of the BCC
  - Restrictions on the concept or conceptual domain of the ASCC property or BCC property as reflected in the definition of the usage rules

Restrictions on the cardinality of the BCCs and ASCCs

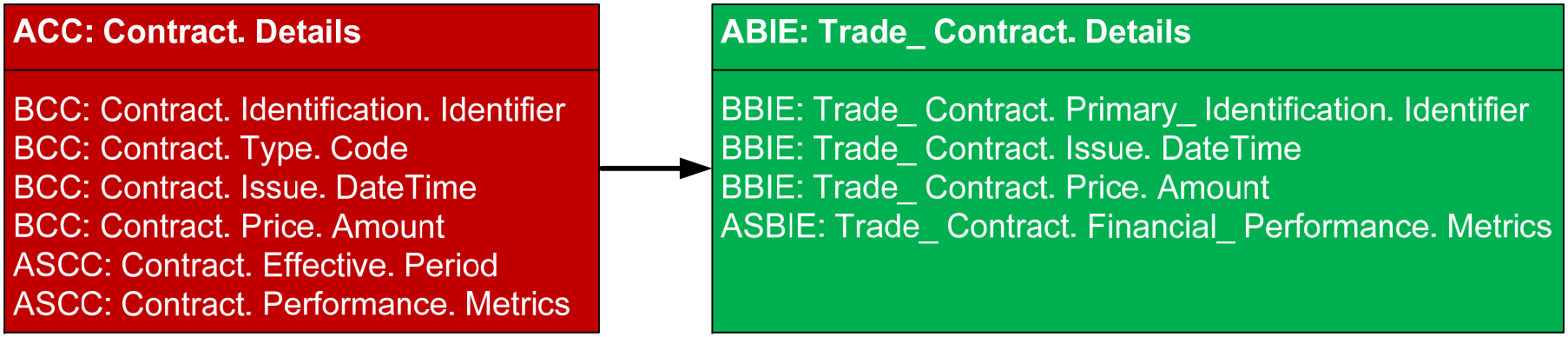


Use and non-use of individual BCCs and ASCCs



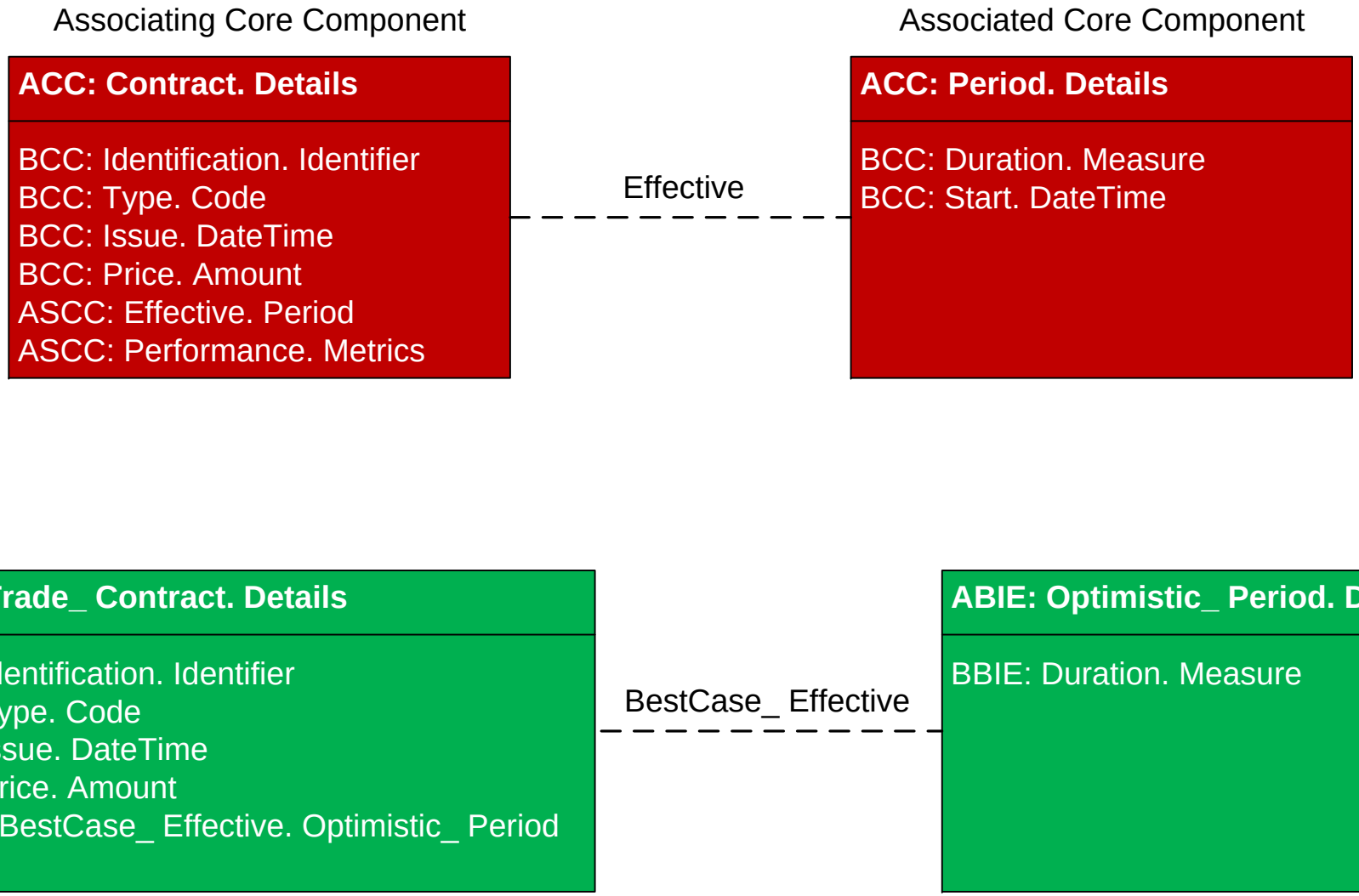
- Non-used ASCC
  - Contract. Effective. Period
- Non-used BCC
  - Contract. Type. Code

Qualification of individual ASCC and BCC properties

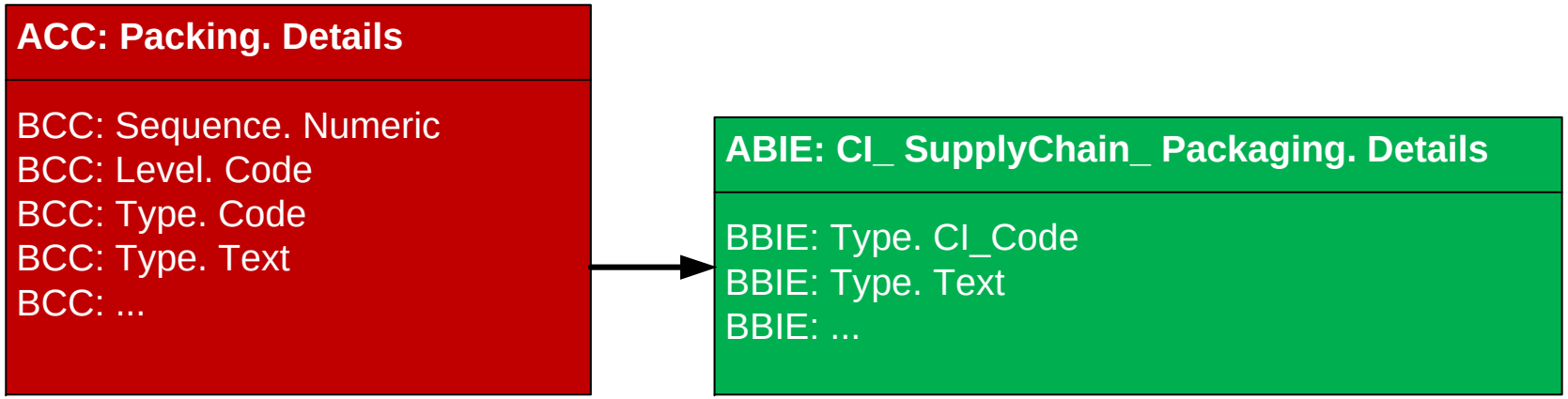


- Qualified BCC property
  - Trade_ Contract. **Primary**_ Identification. Identifier
- Qualified ASCC property
  - Trade_ Contract. **Financial**_ Performance. Metrics

Restrictions on the content model of an associated ACC for an ASCC

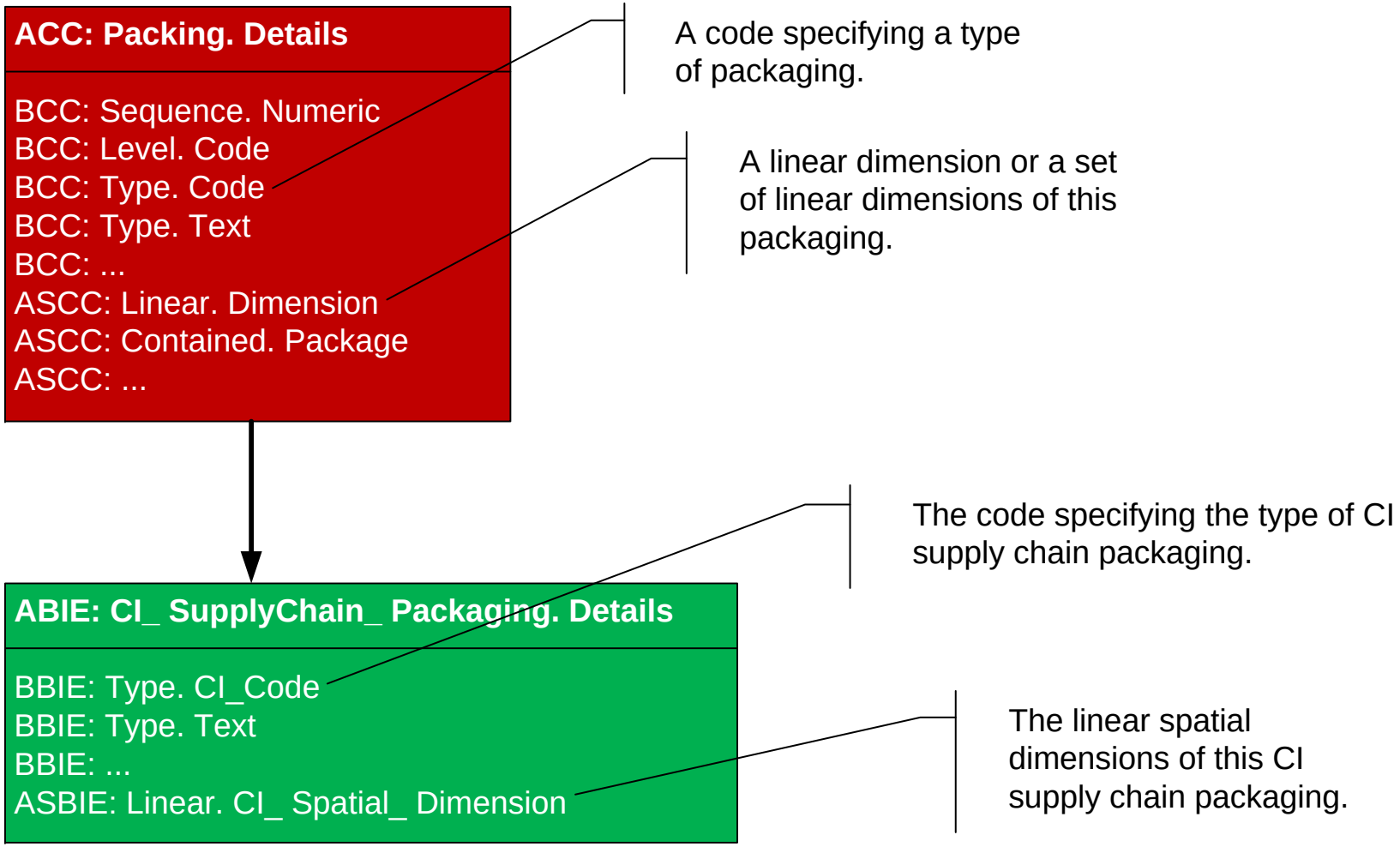


Restriction on the data type of the BCC



- New Business Data Type **CI_Code**, derived from the Core Data Type Code

Restrictions on the concept or conceptual domain of the ASCC property or BCC property as reflected in the definition and usage rules



Qualifier hiearchies

- ASCC properties and BCC properties may have different **qualifiers** applied
- This **may result in the ABIE, having a greater number of qualified properties**, than its corresponding ACCs unqualified properties
- Note that this is still considered a restriction, since each BIE property represents a restriction to its corresponding core component property
- Multiple qualifiers create a **qualifier hierarchy**, with each additional qualifier reflecting a further restriction of its less qualified BIE property

The multi-qualified ABIE

Electronic_ Trade_ Contract. Details

qualifies the qualified ABIE

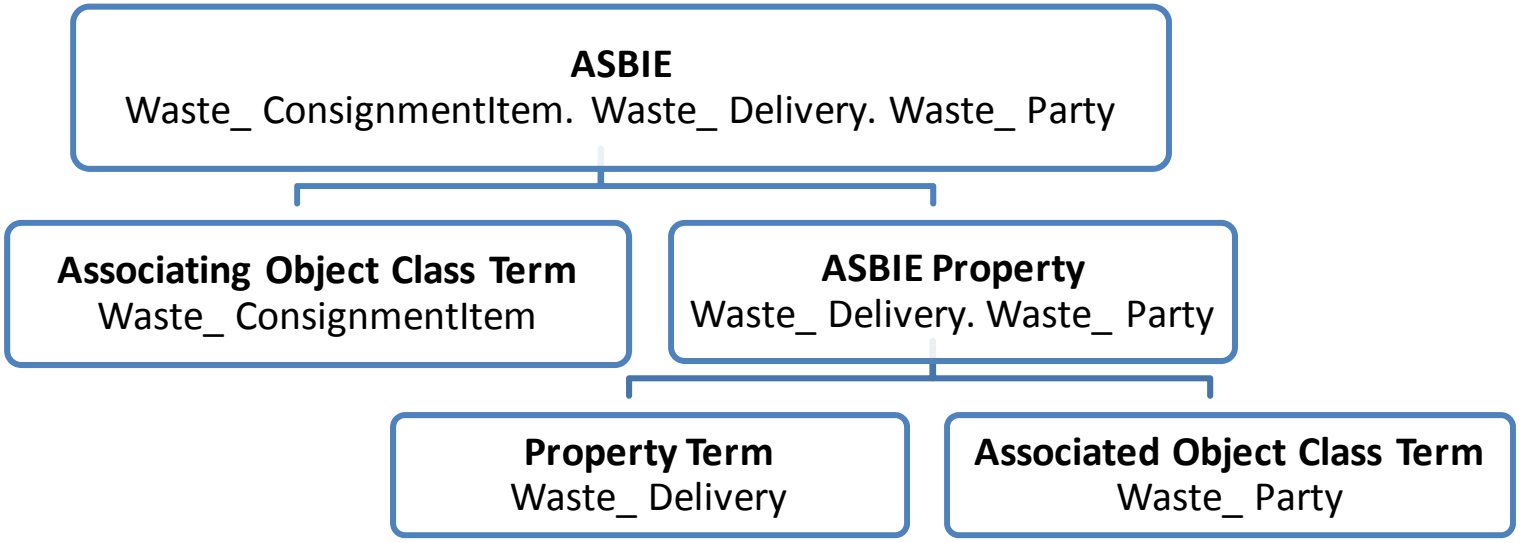
Trade_ Contract. Details

which qualifies the ACC

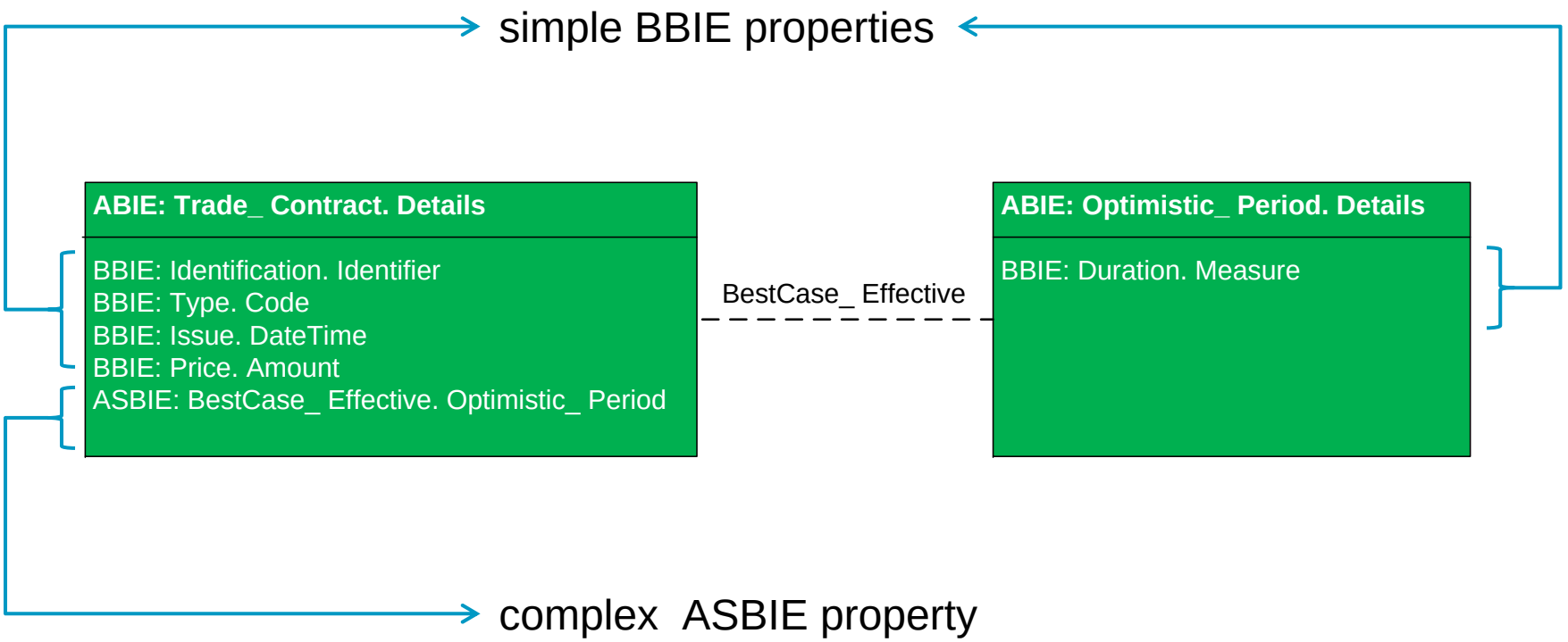
Contract. Details

Association Business Information Entity (ASBIE)

- An ASBIE has the structure of, and represents another ABIE
- An ASBIE is **based on an ASCC**, but exists in a business context
- An ASBIE consists of an **ASBIE property** plus the **object class** of the **parent ABIE**
- An ASBIE property is **reusable across object classes**, but once it has been given the object class of the parent ABIE, it becomes an ASBIE that is unique for the given ABIE it is assigned to

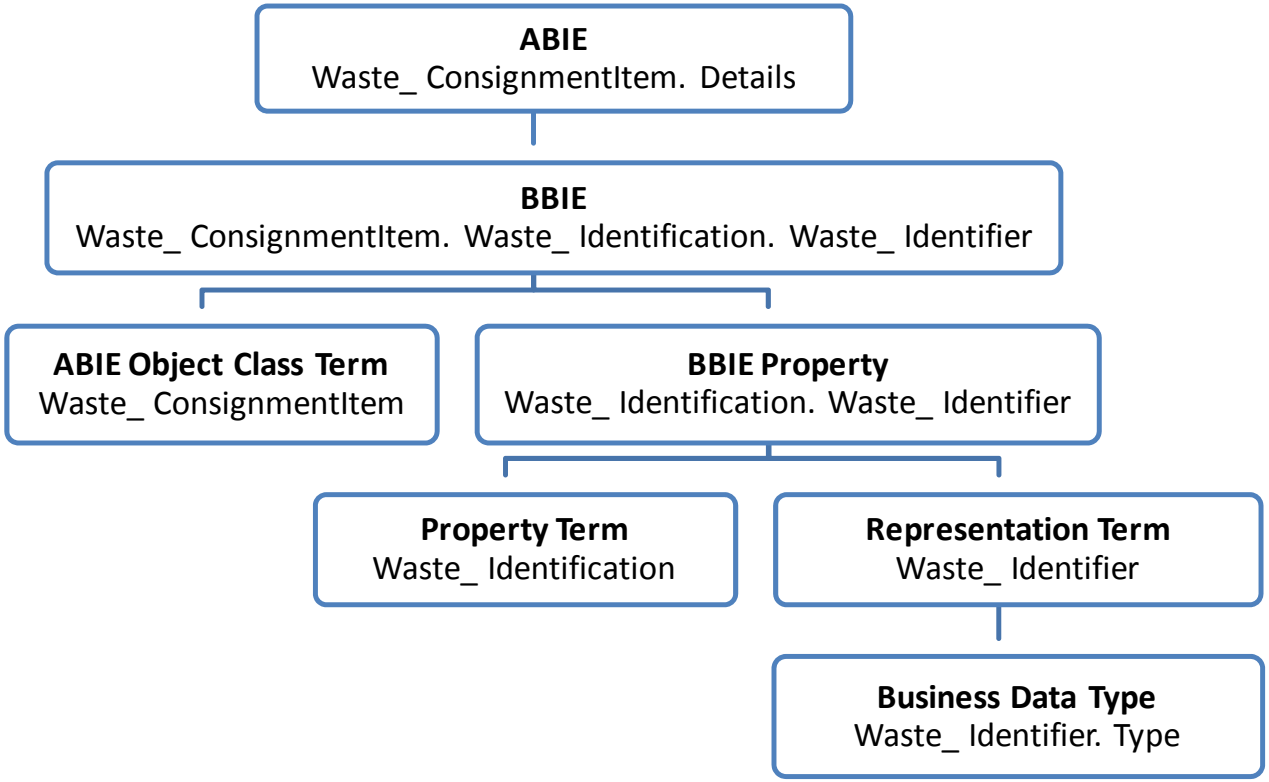


Association Business Information Entity (ASBIE) – example



Basic Business Information Entity (BBIE)

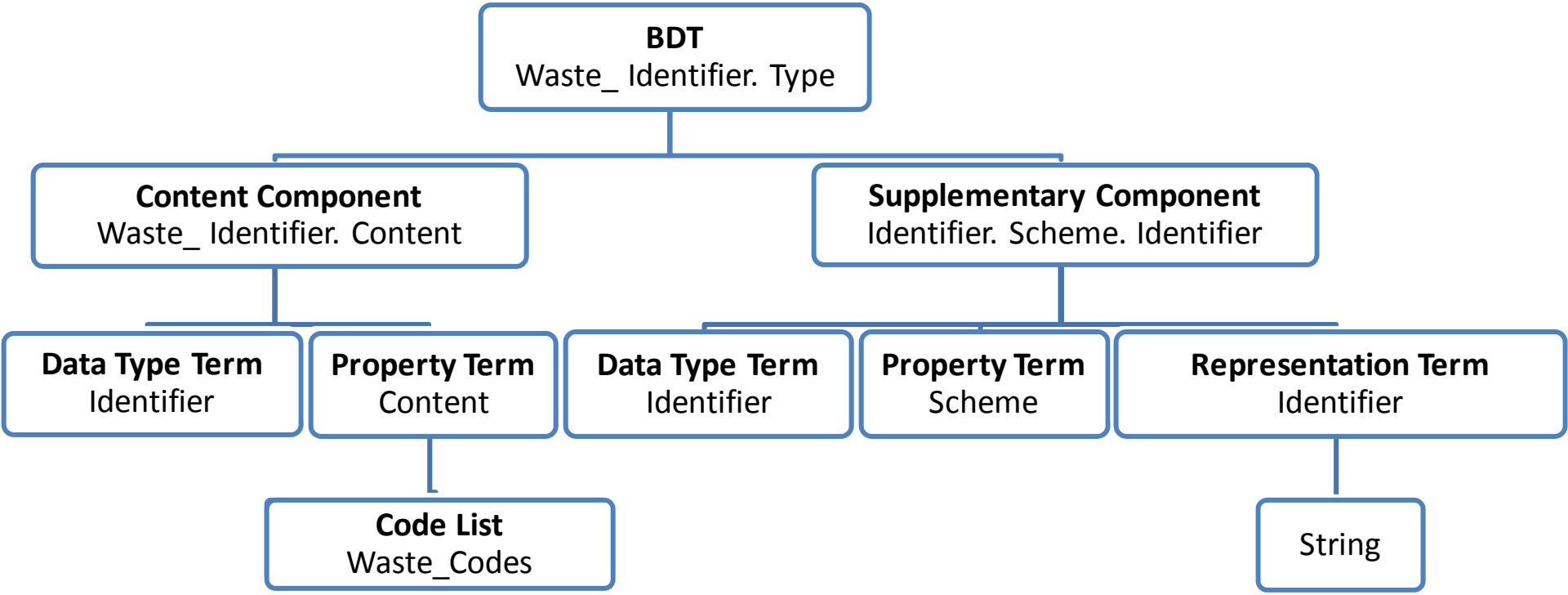
- A Basic Business Information Entity is a Basic Core Component (BCC), used in a specific business context
- Multiple BBIEs can be derived from a single BCC
- A BBIE consists of an **BBIE property** plus the **object class** of the **parent ABIE**
- A BBIE property is **reusable across object classes**, but once it has been given the object class of the parent ABIE, it becomes a BBIE that is unique for the given ABIE it is assigned to



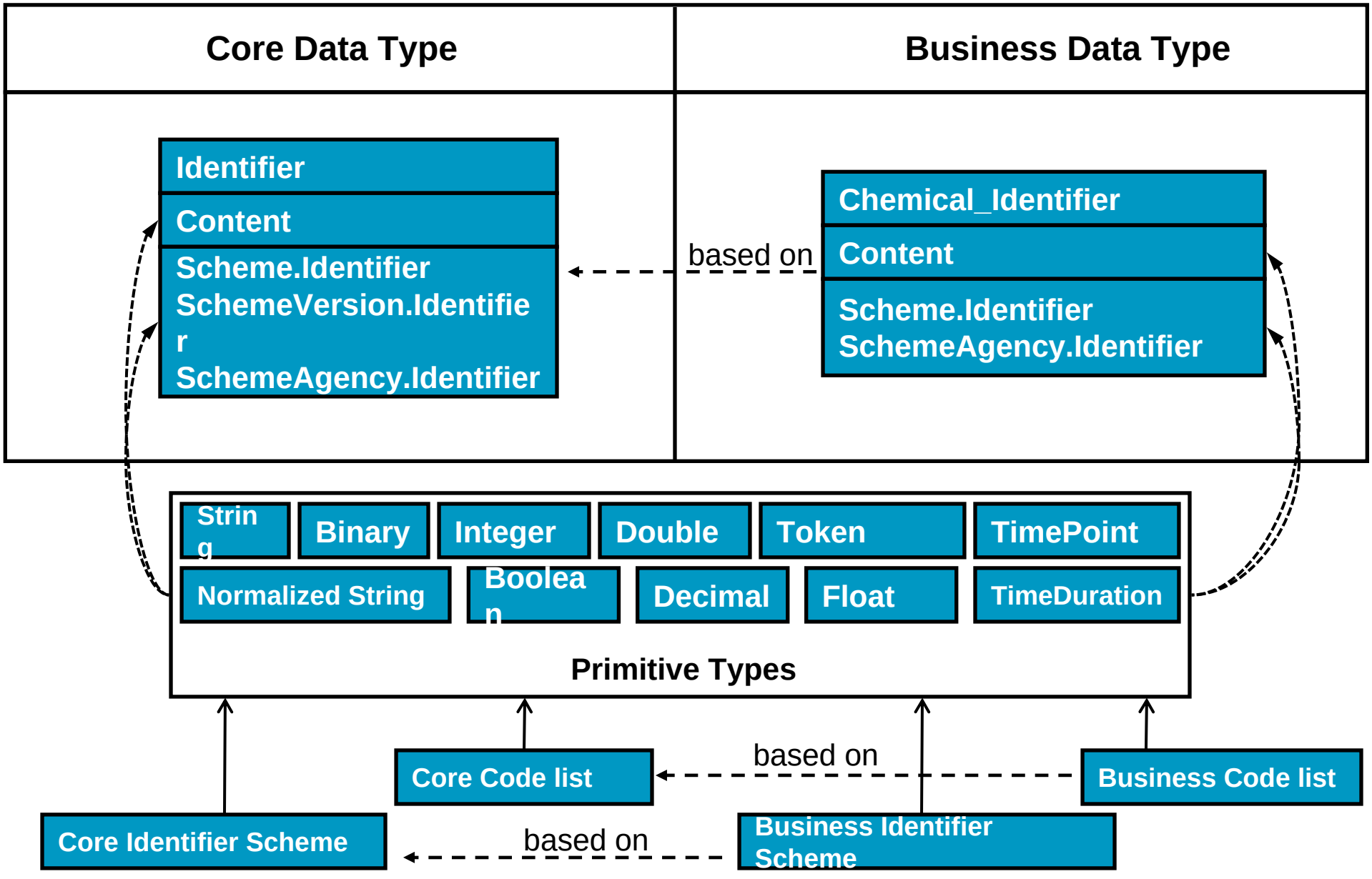
Business Data Types (BDT)

- A Business Data Type is used to **set the value domain of a Basic Business Information Entity**
- For every approved CDT, a corresponding unrestricted BDT will be created
  - This BDT will have no restrictions of the set of values of its source CDT's content and supplementary components
- Additional BDTs may be created that restrict the set of values of its source CDT's content and supplementary components
- The restrictions represent a qualification of the BDT similar to the qualification of BIEs

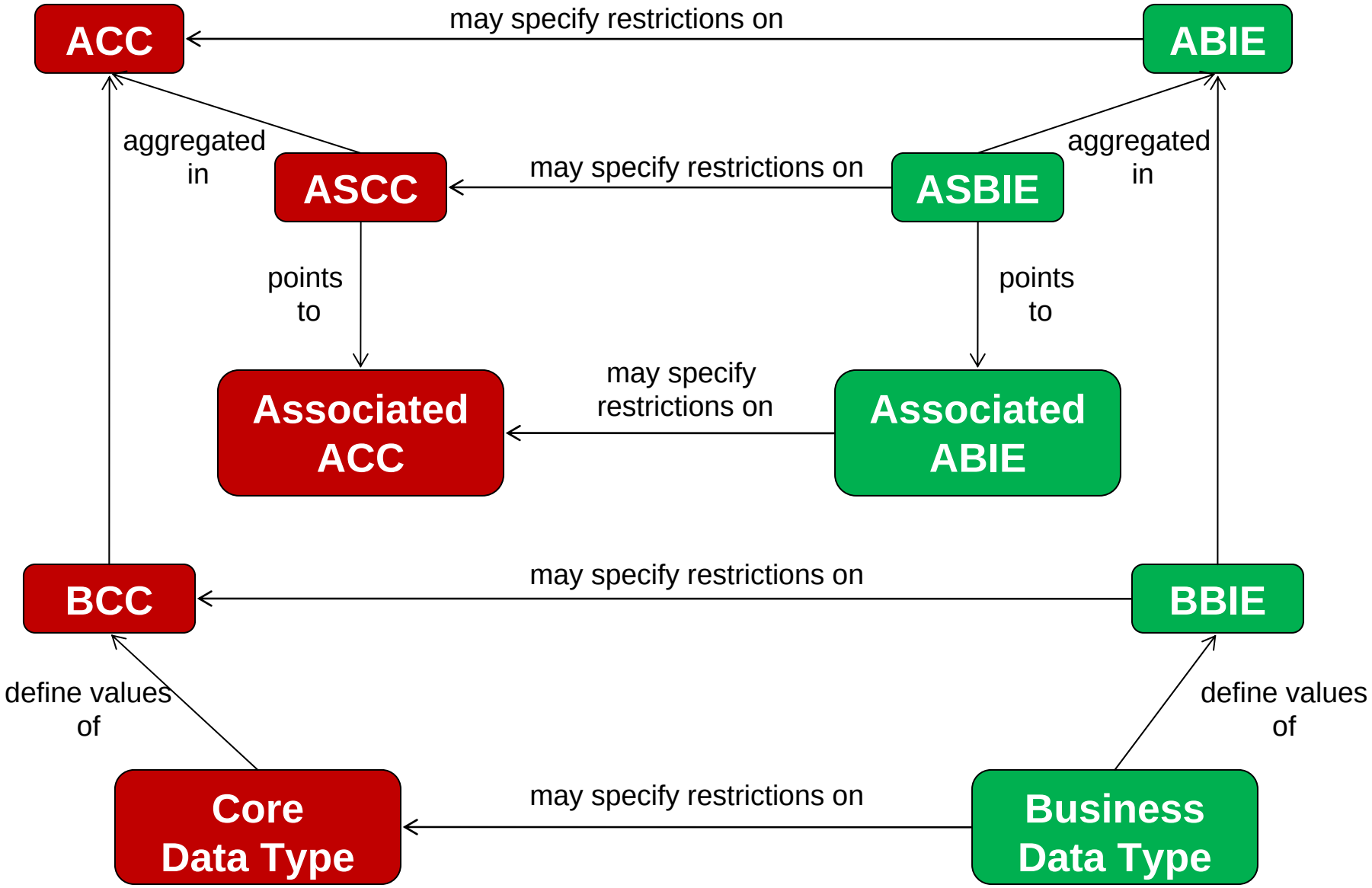
Business Data Types cont‘d



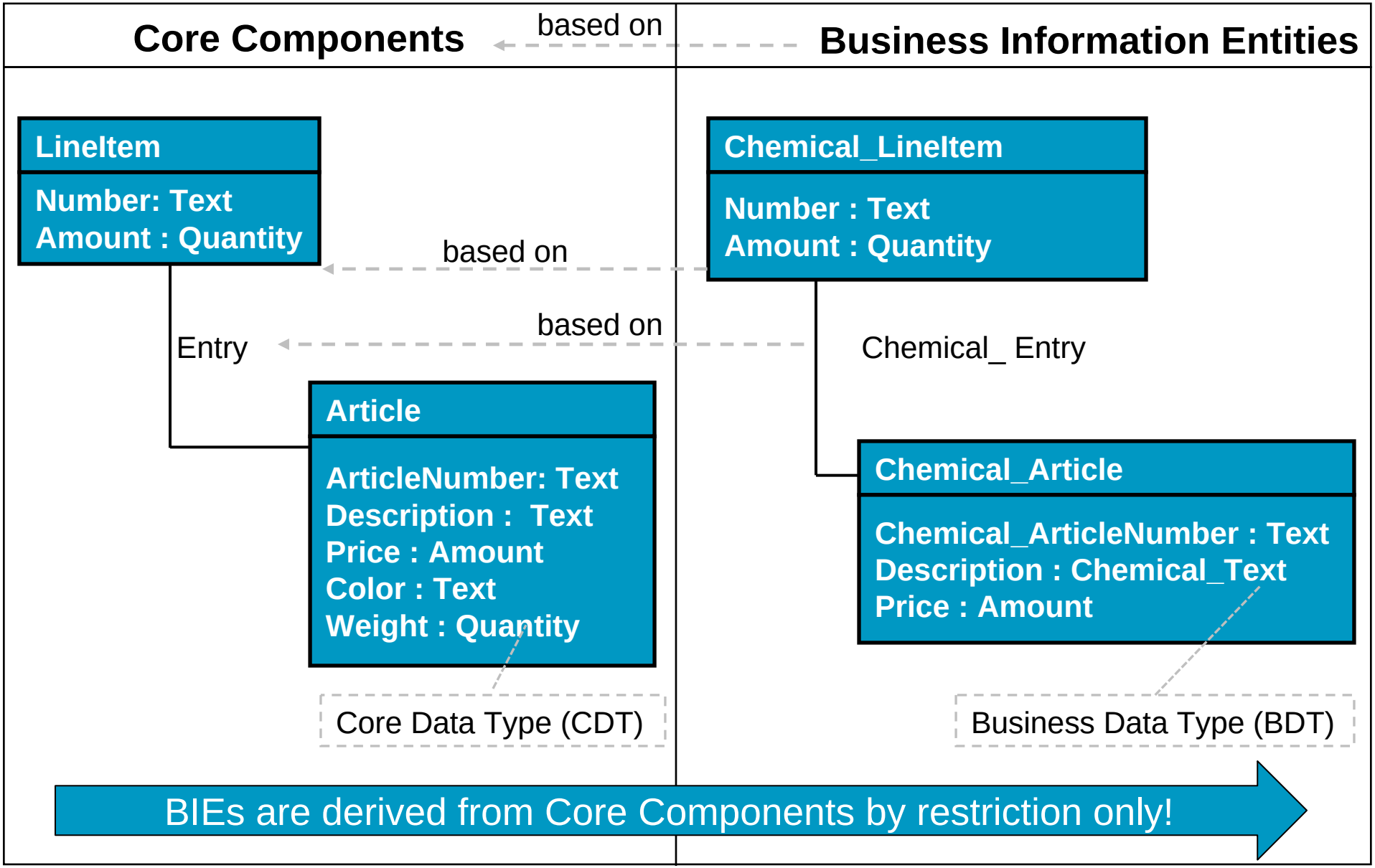
From Core Data Types to Business Data Types



Core Components and Business Information Entities

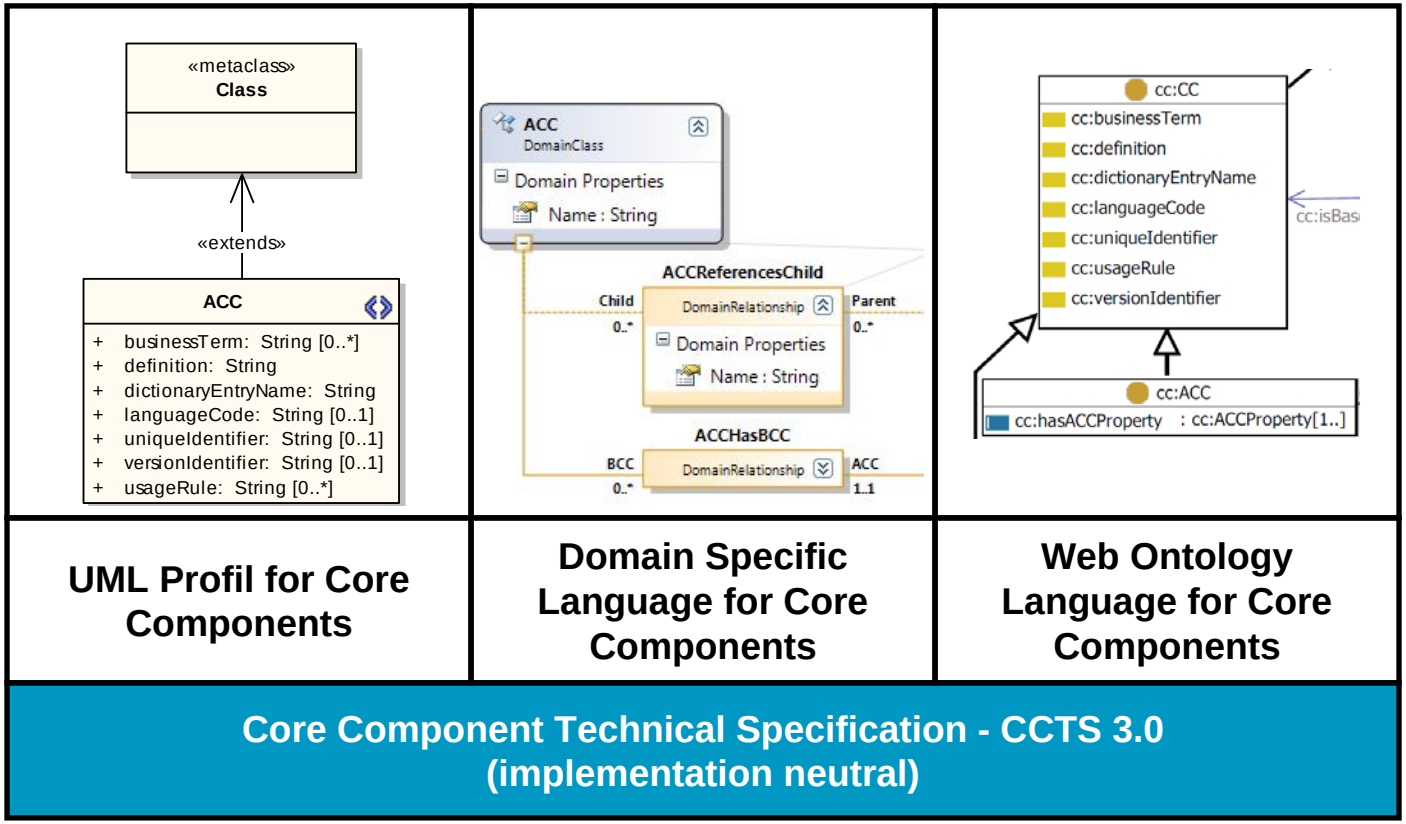


Dependency between core components and business information entities



Shortcomings of core components

- No formal representation mechanism available for core components
- No direct integration in modeling tools available
- Core components are defined in an implementation neutral manner



Thank you for your attention

<Lecturer>
<Name>**Philipp Liegl**</Name>
<Company>**Research Studios Austria**</Company>
<Department>**Research Studio Inter-Organisational Systems**</Department>
<Address>
<Street>**Thurngasse 8/3/20**</Street>
<ZIP>**1090**</ZIP><City>**Vienna**</City>
<Country>**Austria**</Country>
</Address>
<Contact>
<Email>**office.ios@researchstudio.at**</Email>
<Http>**http://www.researchstudio.at**</Http>
</Contact>
<? Presentation status="**questions**" ?>
</Lecturer>