ecosio
a VERTEX company

E-Rechnung: Technische Details beim Empfangen, Validieren, Erstellen und Versenden

Dr. Philipp Liegl

E-Mail: philipp.liegl@ecosio.com
https://ecosio.com

About me

Philipp Liegl

Wirtschaftsinformatiker mit Schwerpunkt B2B-Prozesse

Mitarbeit in der UN/CEFACT-Standardisierung und der e-Rechnungs-Standardisierung

Über 20 Jahre Projekterfahrung in den Bereichen FMCG, Einzelhandel, Fertigung, Industrie und Automotive

Technischer Projektleiter bei verschiedenen internationalen e-Rechnungsprojekten

Tüftelt immer noch mit Begeisterung an Integrations-lösungen - vor allem mit SAP

Agenda

Rechnungen empfangen

Rechnungen validieren

Rechnungen versenden

Zielarchitektur

Rechnungen empfangen

Eine elektronische Rechnung muss im XML-Format gemäß EN 16931 vorliegen

E**x**tensible **M**arkup **L**anguage

Unterscheidung zwischen Markup (Struktur) und dem eigentlichen Inhalt

- Markup: Rot und Orange
- Inhalt: Weiß und Grün

Verwendete Konzepte

- Definition der Struktur: **XML Schema**
- Integritätsbedingungen: **Schematron**
- Selektion von Elementen/Attributen: **XPath**
- Einzelnes Dokument nennt man auch **XML-Instanz**

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoice  xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
 xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
 xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDataTypes-2"
 xmlns:udt="urn:oasis:names:specification:ubl:schema:xsd:UnqualifiedDataTypes-2"
 xmlns:ccts="urn:un:unece:uncefact:documentation:2"
 xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2">
    <cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:xoev-de:kosit:standard:xrechnung_1.2</
    cbc:CustomizationID>
    <cbc:ID>12115118</cbc:ID>
    <cbc:IssueDate>2015-01-09</cbc:IssueDate>
    <cbc:DueDate>2015-01-09</cbc:DueDate>
    <cbc:InvoiceTypeCode>380</cbc:InvoiceTypeCode>
    <cbc:DocumentCurrencyCode>EUR</cbc:DocumentCurrencyCode>
    <cbc:BuyerReference>A test buyer reference</cbc:BuyerReference>
    <cac:AccountingSupplierParty>
        <cac:Party>
            <cac:PostalAddress>
                <cbc:StreetName>Postbus 71</cbc:StreetName>
                <cbc:CityName>Velsen-Noord</cbc:CityName>
                <cbc:PostalZone>1950 AB</cbc:PostalZone>
                <cac:Country>
                    <cbc:IdentificationCode>NL</cbc:IdentificationCode>
                </cac:Country>
            </cac:PostalAddress>
            <cac:PartyTaxScheme>
                <cbc:CompanyID>NL8200.98.395.B.01</cbc:CompanyID>
                <cac:TaxScheme>
                    <cbc:ID>VAT</cbc:ID>
                </cac:TaxScheme>
            </cac:PartyTaxScheme>
            <cac:PartyLegalEntity>
                <cbc:RegistrationName>De Koksmaat</cbc:RegistrationName>
                <cbc:CompanyID>57151520</cbc:CompanyID>
            </cac:PartyLegalEntity>
            <cac:Contact>
```

EU-Richtlinie 2014/55/EU und die EN 16931

Schreibt die **Entwicklung eines einheitlichen EU-weiten E-Rechnungsstandards** vor

Verpflichtet den Einsatz der elektronischen Rechnung im Bereich der öffentlichen Beschaffung in Europa

Öffentliche Auftraggeber müssen elektronische Rechnungen von Lieferanten akzeptieren

Die **EU-Richtlinie resultierte in der EN 16931** (syntax-neutral)

Zwei offiziell unterstützte Syntaxen

- Universal Business Language (UBL)
- Cross Industry Invoice (CII)

ISSN 1977-0677

Official Journal **L 133**

of the European Union

English edition | Legislation | **Volume 57**, **6 May 2014**

Contents | *I Legislative acts* | page

DIRECTIVES

* **Directive 2014/55/EU of the European Parliament and of the Council of 16 April 2014 on electronic invoicing in public procurement** ([1]) 1

Einsatz der EN 16931: XRechnung in Deutschland

https://xeinkauf.de/xrechnung/

Die XRechnung ist eine Core Invoice Usage Specification (CIUS) der Europäischen Norm EN 16931

Sie enthält Einschränkungen/Erweiterungen der EN 16931 speziell für Deutschland (Verwendung von Leitweg-ID, Abbildung von Skonto, Subline-Items, etc.)

Aktuelle Version: 3.0.1 (gültig seit 01.02.2024)

Gültige Syntaxen:

- **Cross Industry Invoice (CII)**
  - XML-Standard, der bei ZUGFeRD/Factur-X verwendet wird
  - Eine ZUGFeRD 2.2 Rechnung ist auch eine gültige XRechnung (nur wenn das Profil EN 16931 verwendet wird)
- **Universal Business Language (UBL)**
  - XML-basierter Standard für elektronische Dokumente
  - Wird bei Peppol standardmäßig verwendet
- **ZUGFeRD/Factur-X**
  - PDF/A-3 mit eingebettetem CII XML

Beispiel für EN16931/XRechnung

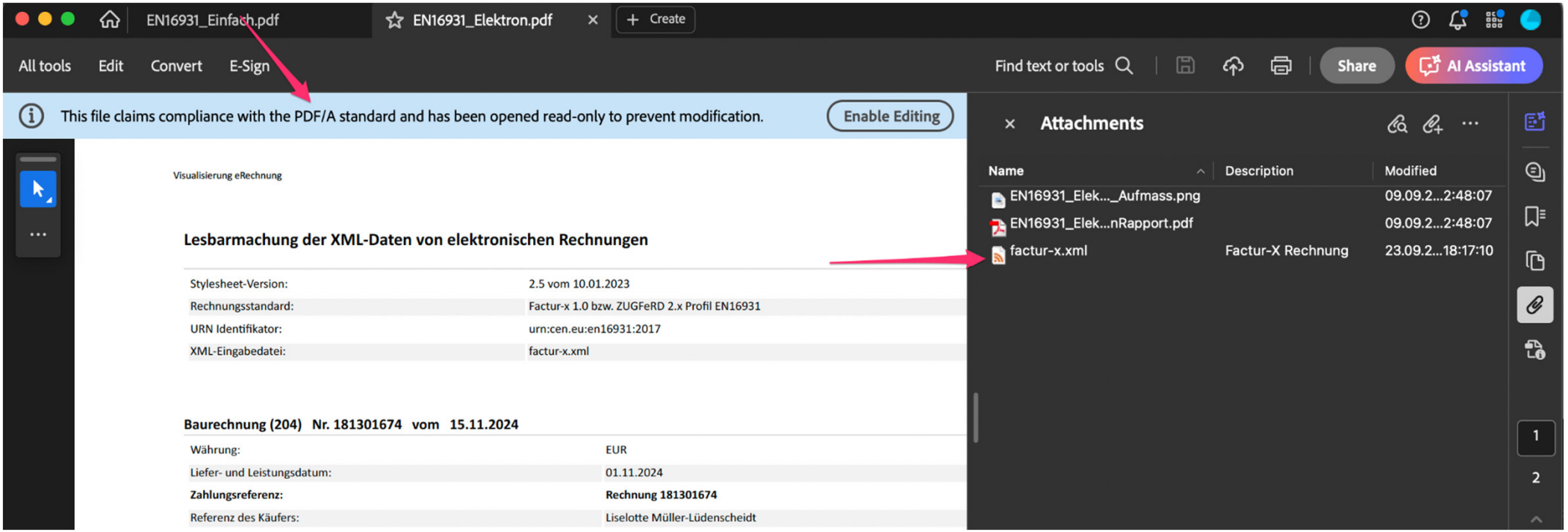
UBL – Universal Business Language

```
<?xml version="1.0" encoding="UTF-8"?>
<ubl:Invoice xmlns:ubl="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2"
             xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
             xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2 http://docs.oasis-open.org/ubl/os-
    <cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:xoev-de:kosit:standard:xrechnung_1.2</cbc:CustomizationID>
    <cbc:ID>PRG1502112</cbc:ID>
    <cbc:IssueDate>2015-04-24+01:00</cbc:IssueDate>
    <cbc:InvoiceTypeCode>380</cbc:InvoiceTypeCode>
    <cbc:Note>#ADU#Trainer: Herr […]</cbc:Note>
    <cbc:DocumentCurrencyCode>EUR</cbc:DocumentCurrencyCode>
    <cbc:TaxCurrencyCode>EUR</cbc:TaxCurrencyCode>
    <cbc:BuyerReference>99000000-18188-16</cbc:BuyerReference>
    <cac:InvoicePeriod>
        <cbc:StartDate>2015-04-20+01:00</cbc:StartDate>
        <cbc:EndDate>2015-04-24+01:00</cbc:EndDate>
    </cac:InvoicePeriod>
    <cac:ProjectReference>
        <cbc:ID>PR456789</cbc:ID>
    </cac:ProjectReference>
```

CII – Cross Industry Invoice

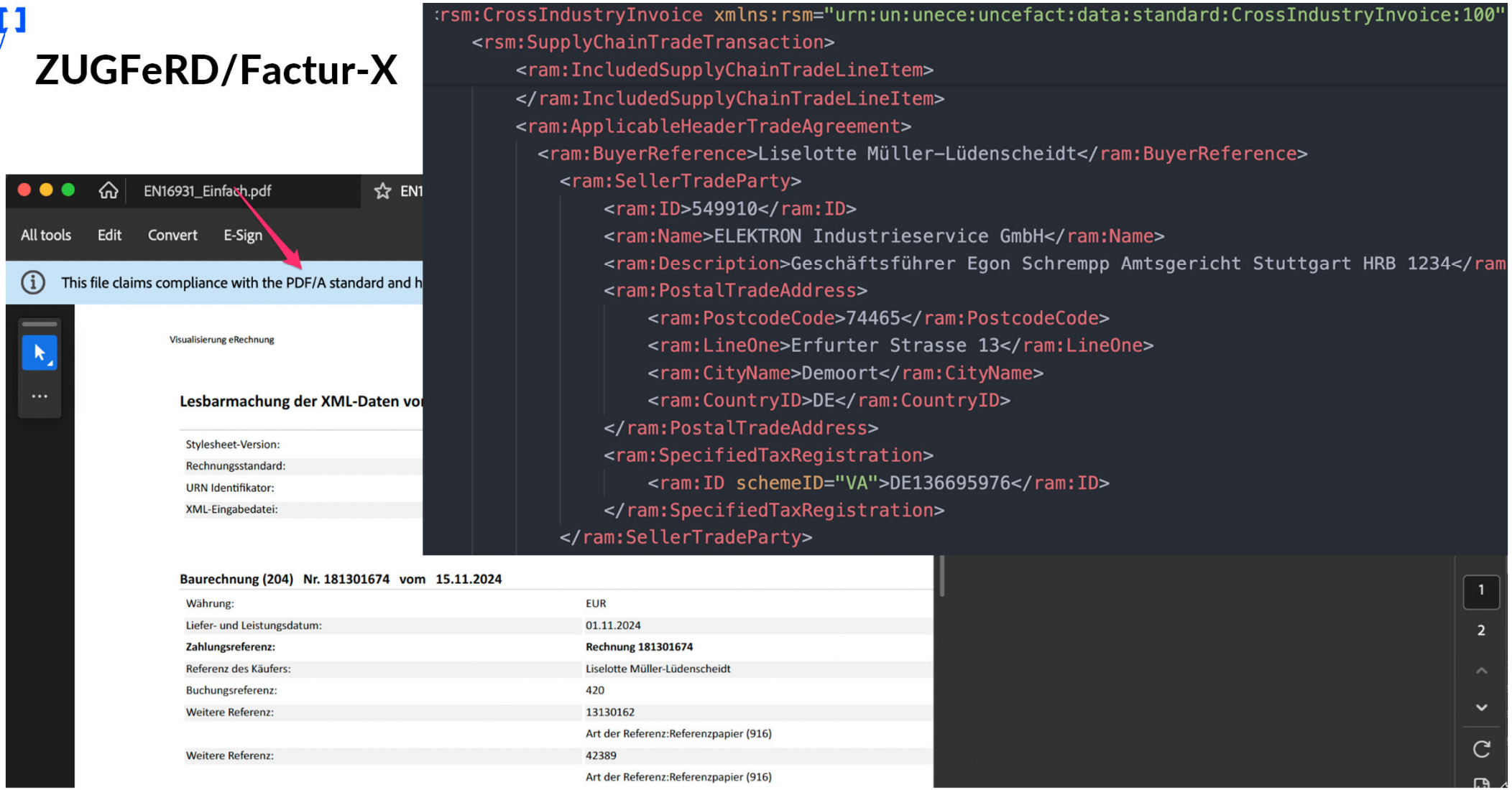
```
<?xml version="1.0" encoding="UTF-8"?>
<rsm:CrossIndustryInvoice xmlns:rsm="urn:un:unece:uncefact:data:standard:CrossIndustryInvoice:100"
                          xmlns:ram="urn:un:unece:uncefact:data:standard:ReusableAggregateBusinessInforma
                          xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:100"
                          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                          xsi:schemaLocation="urn:un:unece:uncefact:data:standard:CrossIndustryInvoice:10
    <rsm:ExchangedDocumentContext>
        <ram:GuidelineSpecifiedDocumentContextParameter>
            <ram:ID>urn:cen.eu:en16931:2017#compliant#urn:xoev-de:kosit:standard:xrechnung_1.2</ram:ID>
        </ram:GuidelineSpecifiedDocumentContextParameter>
    </rsm:ExchangedDocumentContext>
    <rsm:ExchangedDocument>
        <ram:ID>PRG1502112</ram:ID>
        <ram:TypeCode>380</ram:TypeCode>
        <ram:IssueDateTime>
            <udt:DateTimeString format="102">20150424</udt:DateTimeString>
        </ram:IssueDateTime>
        <ram:IncludedNote>
            <ram:Content>Trainer: Herr […]</ram:Content>
            <ram:SubjectCode>ADU</ram:SubjectCode>
        </ram:IncludedNote>
```

ZUGFeRD/Factur-X



- Das PDF muss dem PDF/A-3-Standard genügen
- Es muss genau ein Attachment mit dem Namen **factur-x.xml** oder **xrechnung.xml** in der PDF/A-3-Datei vorhanden sein
- Weitere Attachments (z.B. rechnungsbegleitende Unterlagen) sind zulässig
- XMP-Metadaten gemäß ZUGFeRD-Standard müssen eingebettet sein

ZUGFeRD/Factur-X



Rechnungen validieren

Wann ist eine Rechnung eine gültige Rechnung?

Validierung einer XML-Instanz

Schritt 1: **Sicherstellung der Wohlgeformtheit**

- z.B. *end tag* für alle *start tags*
- Siehe https://www.w3resource.com/xml/well-formed.php

Schritt 2: **Sicherstellung der XML-Schemakonformität**

- Erfüllt die XML-Instanz die Regeln des XML-Schemas?
- Entspricht das Encoding der XML-Instanz den Vorgaben des XML-Schemas? (zumeist UTF-8)

Schritt 3: **Sicherstellung der Schematron-Konformität**

- Entspricht die XML-Instanz den Schematron-Regeln der EN16931 (wenn/dann-Bedingungen)
- Siehe https://github.com/ConnectingEurope/eInvoicing-EN16931/tree/master/ubl/schematron

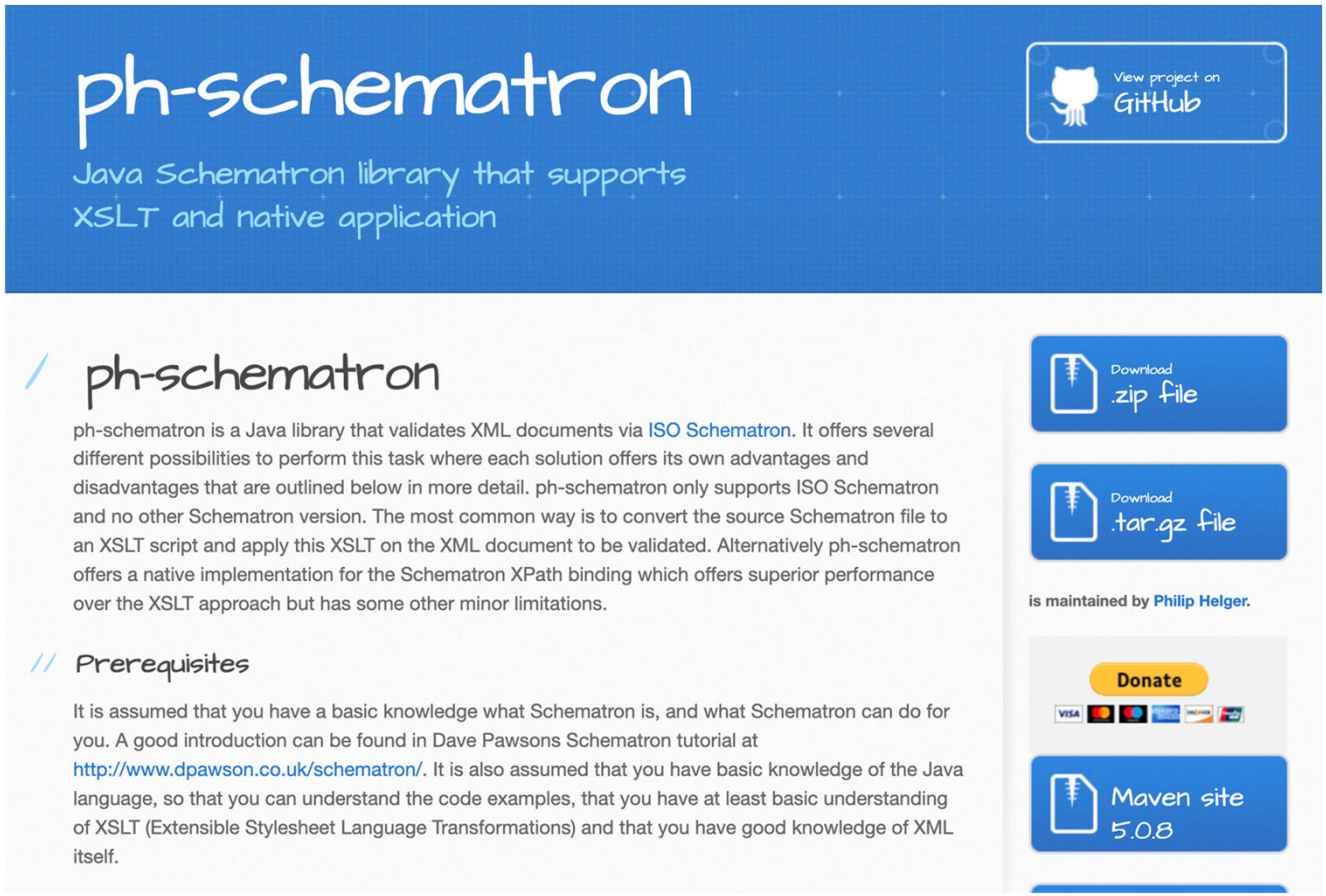
Beispiel für eine XML-Schemavalidierung

```java
package com.ecosio.ihkdemo;

import java.io.File;
import java.io.IOException;
import javax.xml.XMLConstants;
import javax.xml.transform.stream.StreamSource;
import javax.xml.validation.Schema;
import javax.xml.validation.SchemaFactory;
import javax.xml.validation.Validator;
import org.xml.sax.SAXException;

public class XMLValidation {

    public static void main(String[] args) {
      System.out.println("invoice.xml validates against EN16931.xsd? "+validateXMLSchema("Invoice-EN16931.xsd", "invoice.xml"));
    }

    public static boolean validateXMLSchema(String xsdPath, String xmlPath) {
        try {
            SchemaFactory factory =
                    SchemaFactory.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI);
            Schema schema = factory.newSchema(new File(xsdPath));
            Validator validator = schema.newValidator();
            validator.validate(new StreamSource(new File(xmlPath)));
        } catch (IOException | SAXException e) {
            System.out.println("Exception: "+e.getMessage());
            return false;
        }
        return true;
    }
}
```

Schematron-Regeln der EN16931

```
<rule context="/ubl:Invoice | /cn:CreditNote">
  <assert id="BR-01" flag="fatal" test="normalize-space(cbc:CustomizationID) != ''">[BR-01]-An Invoice shall have a Specification identifier (BT-24).
  <assert id="BR-02" flag="fatal" test="normalize-space(cbc:ID) != ''">[BR-02]-An Invoice shall have an Invoice number (BT-1).</assert>
  <assert id="BR-03" flag="fatal" test="normalize-space(cbc:IssueDate) != ''">[BR-03]-An Invoice shall have an Invoice issue date (BT-2).</assert>
  <assert id="BR-04" flag="fatal" test="normalize-space(cbc:InvoiceTypeCode) != '' or normalize-space(cbc:CreditNoteTypeCode) !=''">[BR-04]-An Invoice sh
  <assert id="BR-05" flag="fatal" test="normalize-space(cbc:DocumentCurrencyCode) != ''">[BR-05]-An Invoice shall have an Invoice currency code (BT-5).</
  <assert id="BR-06" flag="fatal" test="normalize-space(cac:AccountingSupplierParty/cac:Party/cac:PartyLegalEntity/cbc:RegistrationName) != ''">[BR-06]-A
  <assert id="BR-07" flag="fatal" test="normalize-space(cac:AccountingCustomerParty/cac:Party/cac:PartyLegalEntity/cbc:RegistrationName) != ''">[BR-07]-A
  <assert id="BR-08" flag="fatal" test="exists(cac:AccountingSupplierParty/cac:Party/cac:PostalAddress)">[BR-08]-An Invoice shall contain the Seller post
  <assert id="BR-10" flag="fatal" test="exists(cac:AccountingCustomerParty/cac:Party/cac:PostalAddress)">[BR-10]-An Invoice shall contain the Buyer posta
  <assert id="BR-16" flag="fatal" test="exists(cac:InvoiceLine) or exists(cac:CreditNoteLine)">[BR-16]-An Invoice shall have at least one Invoice line (B
  <assert id="BR-53" flag="fatal" test="every $taxcurrency in cbc:TaxCurrencyCode satisfies exists(//cac:TaxTotal/cbc:TaxAmount[@currencyID=$taxcurrency]
  <assert id="BR-66" flag="fatal" test="count(cac:PaymentMeans/cac:CardAccount) &lt;= 1">[BR-66]-An Invoice shall contain maximum one Payment Card accou
  <assert id="BR-67" flag="fatal" test="count(cac:PaymentMeans/cac:PaymentMandate) &lt;= 1">[BR-67]-An Invoice shall contain maximum one Payment Mandate
```

XML-Instanz gegen Schematron validieren

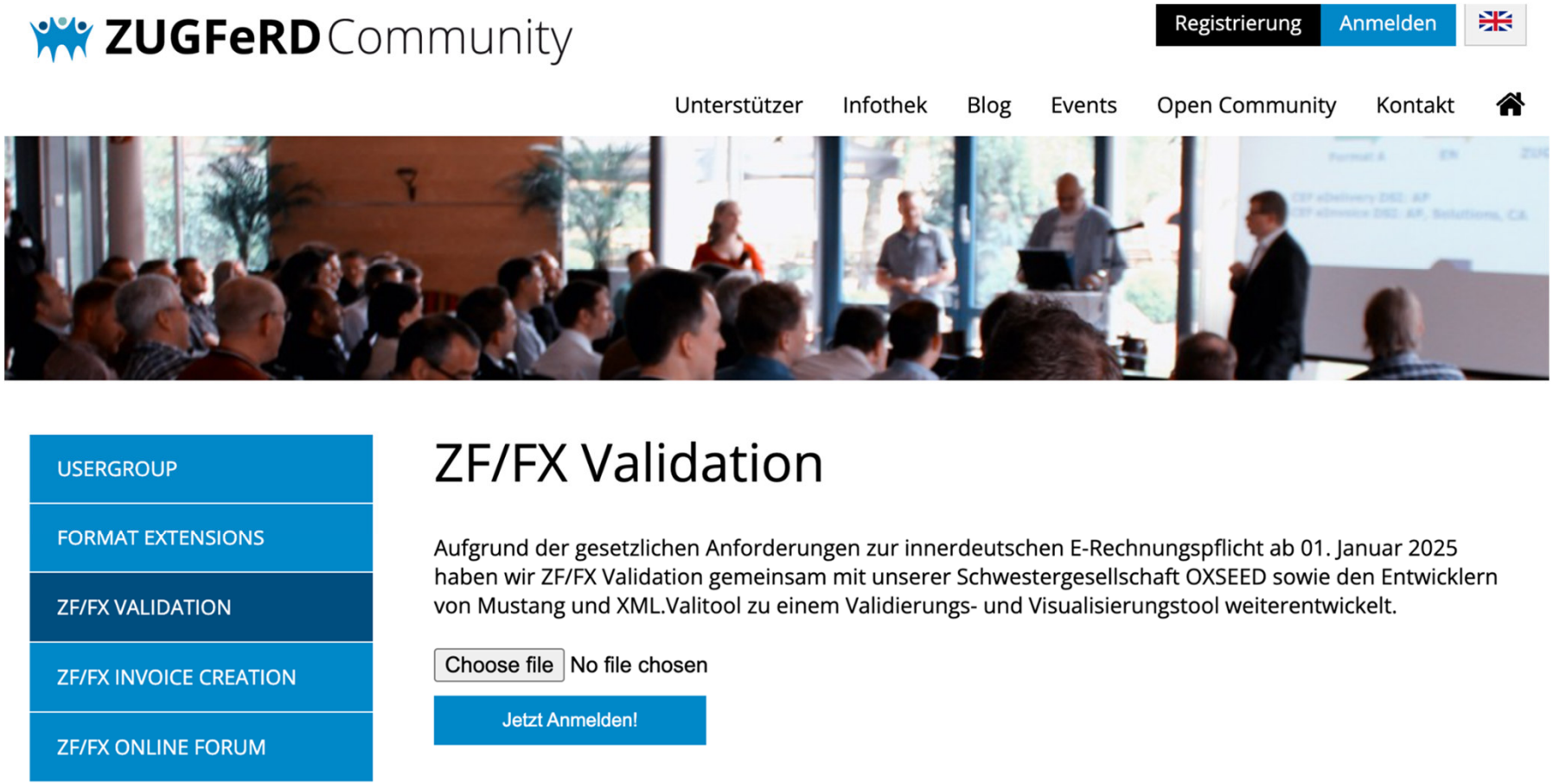


https://github.com/phax/ph-schematron

Schema + Schematron Validierung als kostenfreies Online-Service

https://ecosio.com/en/peppol-and-xml-document-validator/

ZUGFeRD validieren



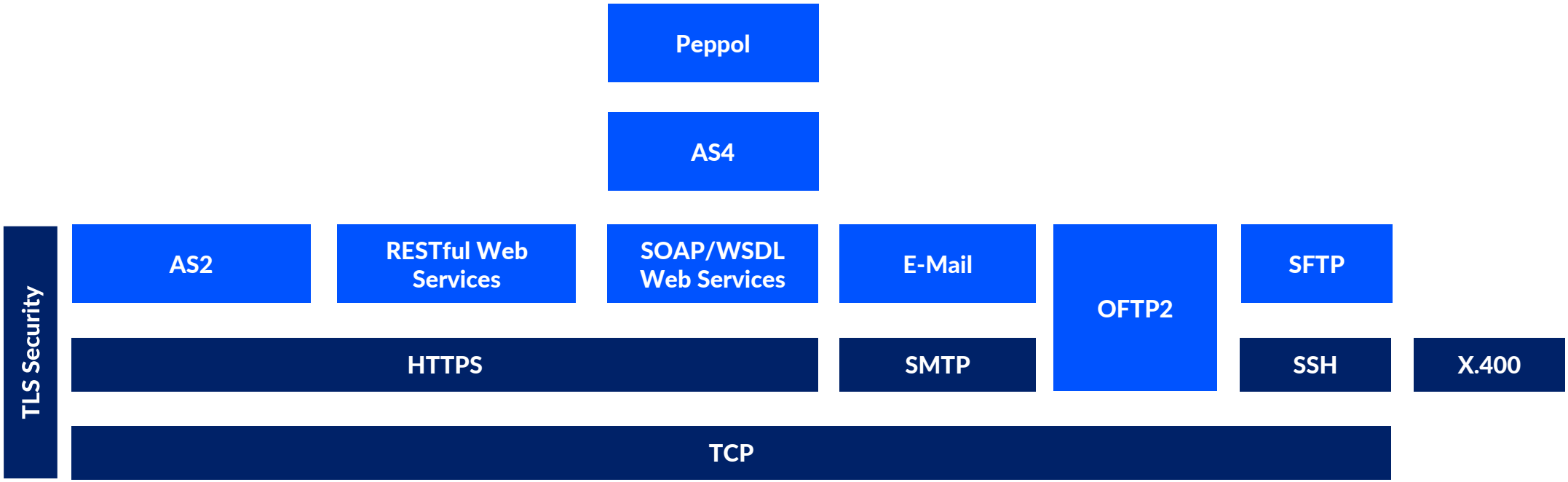
https://www.zugferd-community.net/de/open_community/validation

Rechnungen versenden

Die neue gesetzliche Regelung enthält **keine Vorgaben zum Übermittlungsweg** von elektronischen Rechnungen.

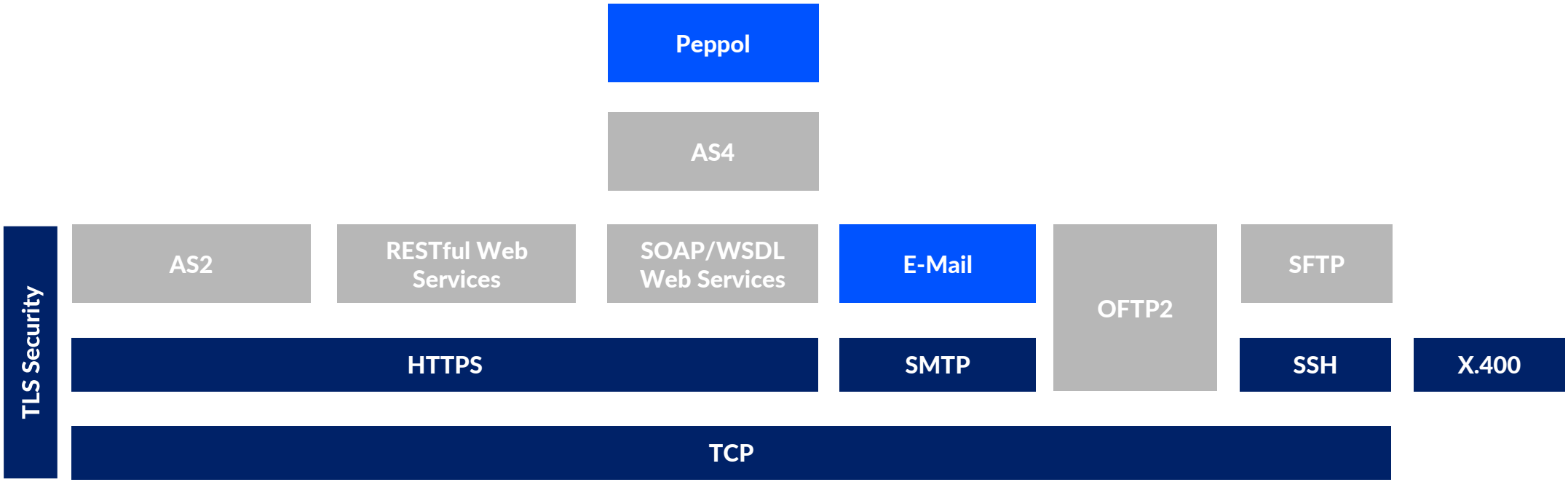
In Frage kommen als prinzipiell “viele” potenzielle Übertragungswege

Beispiel-Protokolle aus dem B2B-Bereich



Realistisch sind aber nur jene, die keine vorherige Abstimmung erfordern

Beispiel-Protokolle aus dem B2B-Bereich



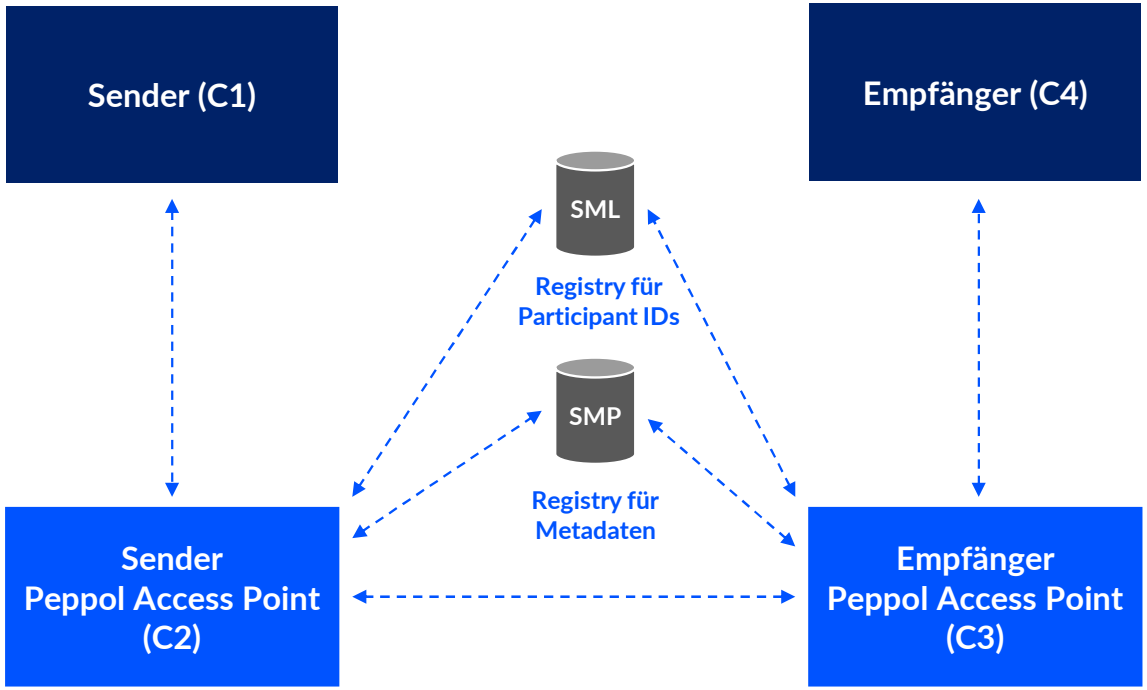
Herausforderungen bei E-Mail

- Fehlende Nichtabstreitbarkeit des Erhalts (kein Acknowledgement)
- Keine Zustellgarantie
  - Mailbox voll
  - Absender-Adresse blockiert
  - ...
- Vollkommen offen - jeder kann an jeden irgendetwas senden
- Keine Validierung von Rechnungen vor dem Versand
- Spam

Was wäre wenn?

Source: https://commons.wikimedia.org/wiki/File:Crowd_of_people_with_phones.jpg

Peppol-Übersicht



Four-corner Modell für Cross-Border E-Procurement

Sender und Empfänger können einen beliebigen Access Point wählen

“Connect once – connect to all“

Keine Roaming-Gebühren zwischen den APs

Was braucht ein Sender, um Ihnen eine Nachricht über Peppol zu senden?

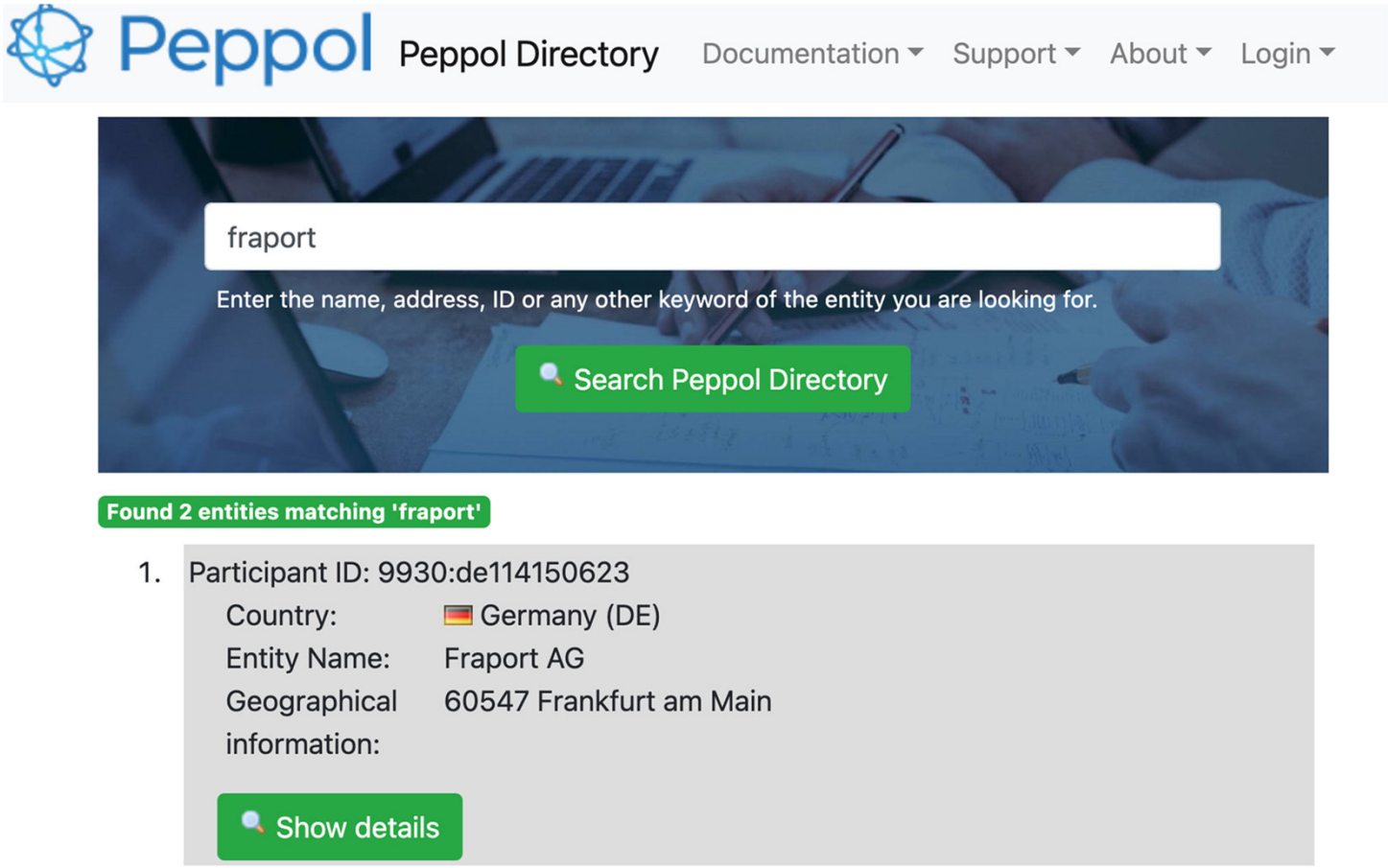
Zwei Dinge:

A) Ihre Peppol ID - z.B. 9930:DE265038025

A) Die Rechnung oder Gutschrift im UBL-Format

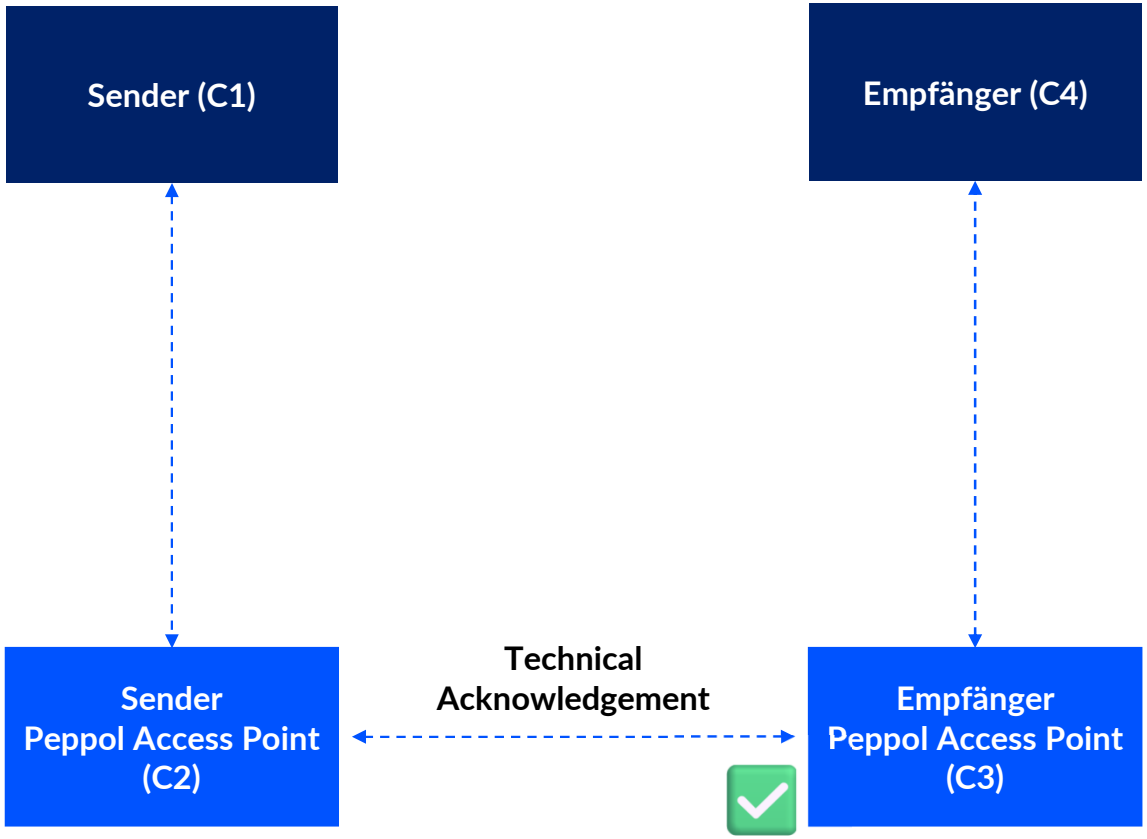
Peppol
Access Point
CERTIFIED PROVIDER

Voraussetzung für den Empfang über Peppol: registrierte Peppol ID



Aufwand für das Eintragen: 10 Minuten

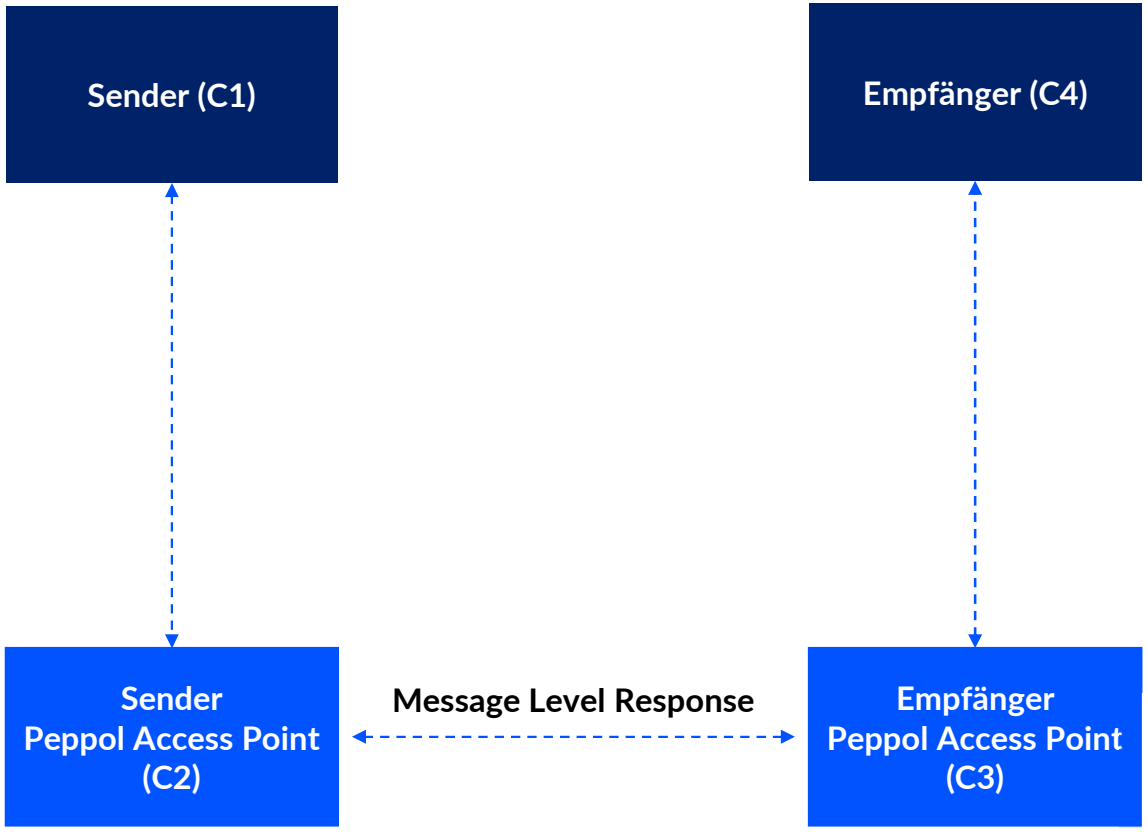
Bestätigungsarten bei Peppol



Technical Acknowledgement (Signal Message)

- Bestätigung, dass der technische Empfang am Access Point funktioniert hat
- Garantiert die Nichtabstreitbarkeit des Erhalts von C3 an C2

Bestätigungsarten bei Peppol

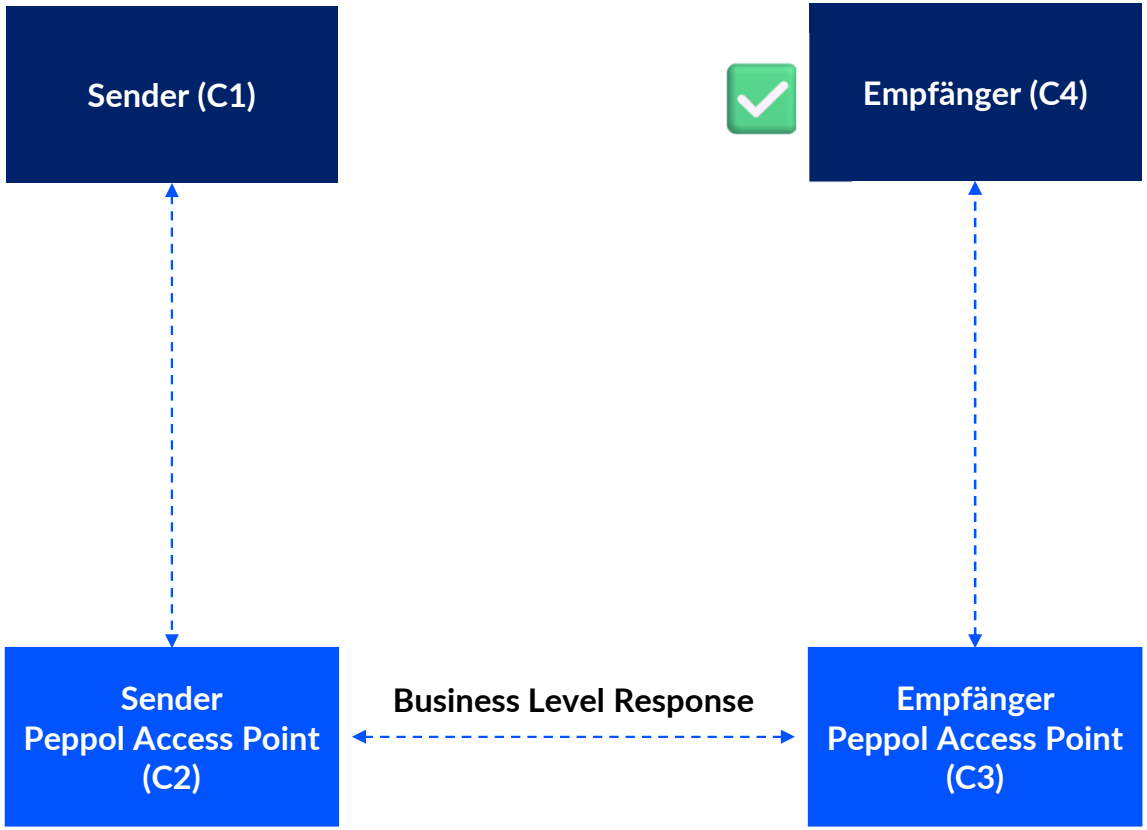


Message Level Response

- Bestätigung, dass die Verarbeitung des gesendeten XMLs am C3 funktioniert hat
- Einer korrekten Weiterverarbeitung am C4 steht daher nichts im Wege

✅ **XML Validation passed OK**

Bestätigungsarten bei Peppol



Business Level Response

- Bestätigung, dass die XML-Nachricht korrekt im Zielsystem C4 verarbeitet werden konnte
- Informiert den Sender über den Verarbeitungszustand

AB: basic message acknowledgement

AP: invoice accepted

RE: invoice rejected

IP: in progress

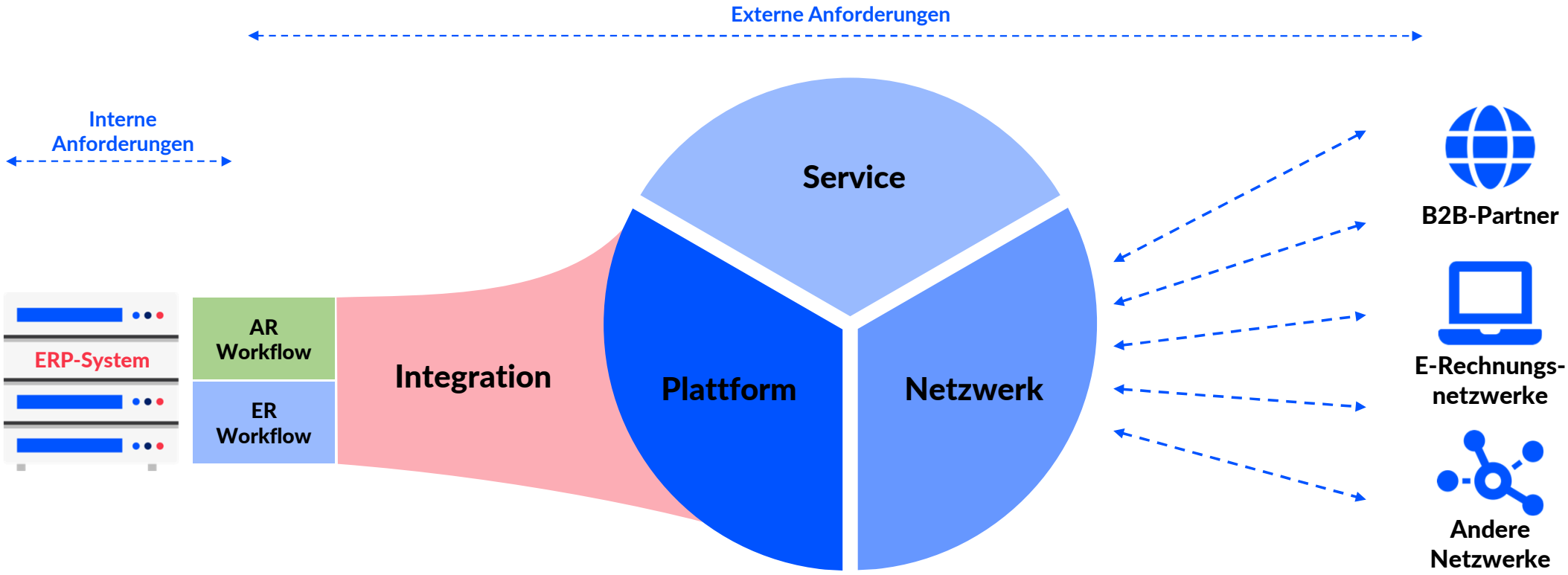
UQ: under query

CA: conditionally accepted

PD: paid/completed

Zielarchitektur

Zielarchitektur einer e-Rechnungslösung



Technische Ansätze für E-Rechnungslösungen

Lokale
Lösung
(z.B. Lokaler Konverter
oder ERP Add-On)

Integration
Platform as a Service
Lösung (iPaaS)

E-Rechnung as a Service
Lösung

Externe Verantwortung

Ihre Verantwortung

CONNECTIONS THAT WORK

Close to our customers

GERMANY
Leoprechtingstraße 32
81739 München
Email: edi@ecosio.com

AUSTRIA
Lange Gasse 30
1080 Wien
Email: edi@ecosio.com

UNITED KINGDOM
77 Farringdon Road
London EC1M 3JU
Email: edi@ecosio.com