

# Het belang van open platforms voor massa adoptie

Kennisplatform Administratieve Software – September 2016



simpler invoicing

# E-facturatie.

Niet omdat het moet, maar omdat het kan



# Jaap Jan Nienhuis



- **Manager Simplerinvoicing**
- Geboren in 1979
- Passie voor innovatie, co-creatie en productontwikkeling

E: [jaapjan.nienhuis@sidn.nl](mailto:jaapjan.nienhuis@sidn.nl)

M: +31 6 15 06 32 10

T: @jj\_nienhuis



Niet omdat het **moet**. Maar omdat het **kan**!

e-factureren zo **simpel**  
maken dat het **makkelijker**  
is dan **papier en pdf**



Close

Manual Entry

Import from file

# Create new invoice...



Browse...



Save draft...



PRINT...

Business Name Invoice #1

Invoice 1  
 Date of Invoice 2011-01-01  
 Due Date 2011-01-07

From: Hospital Demo

Status: Unpaid

Taxed	Unit cost	Qty:	Price
No	€ 50.00 EUR	1	€ 50.00 EUR
No	€ 25.00 EUR	1	€ 25.00 EUR
No	€ 25.00 EUR	1	€ 25.00 EUR
Subtotal		1	€ 25.00 EUR

Close

Import from file

Manual Entry

# Create new invoice...



Browse...



Save draft...



Inboeken bij  
klant...

Select invoice draft

Business Name Invoice #1

Invoice 1  
Date of Invoice 2011-01-01  
Due Date 2011-01-07

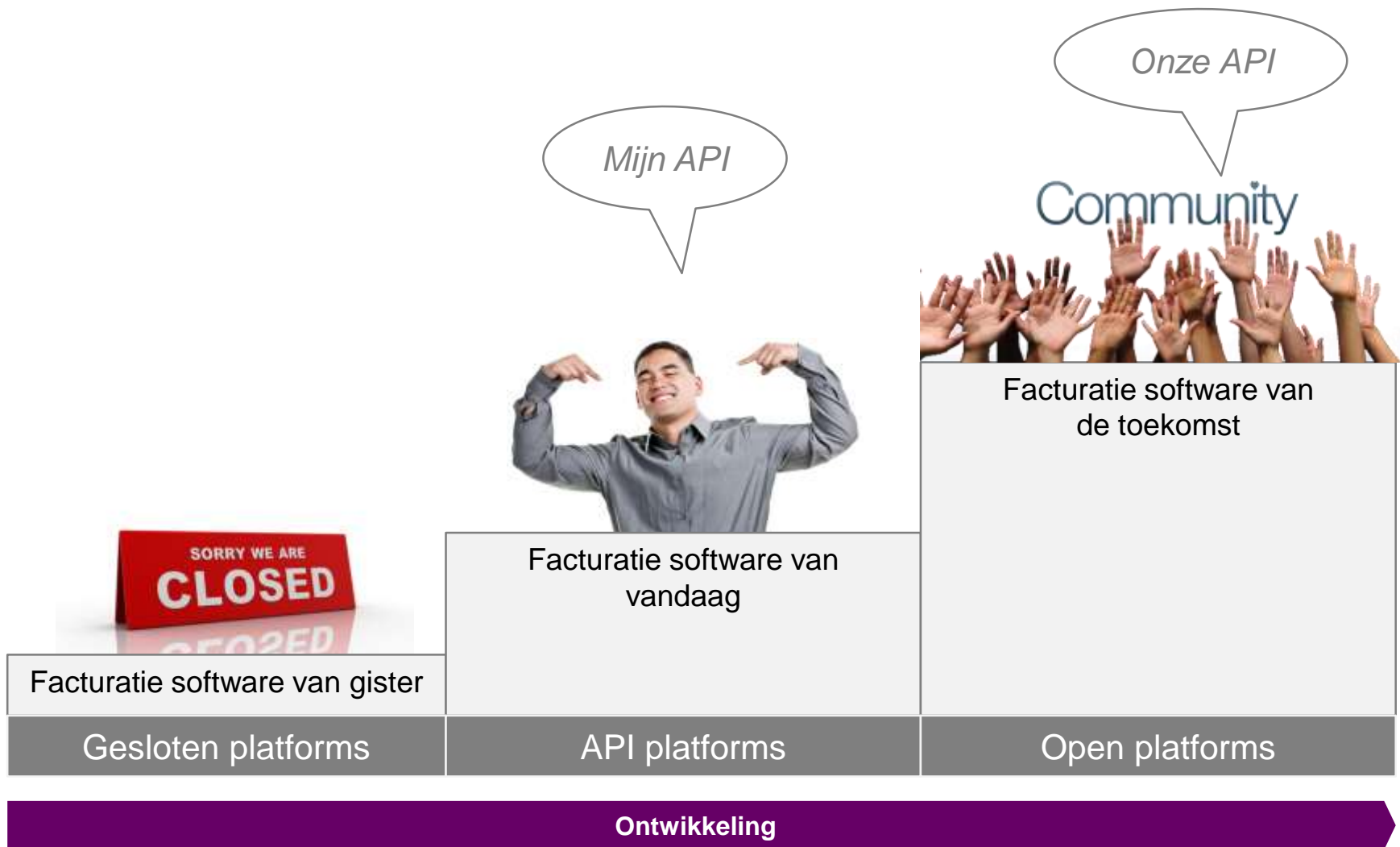
From:  
Huisbil Demo

Status: Unpaid

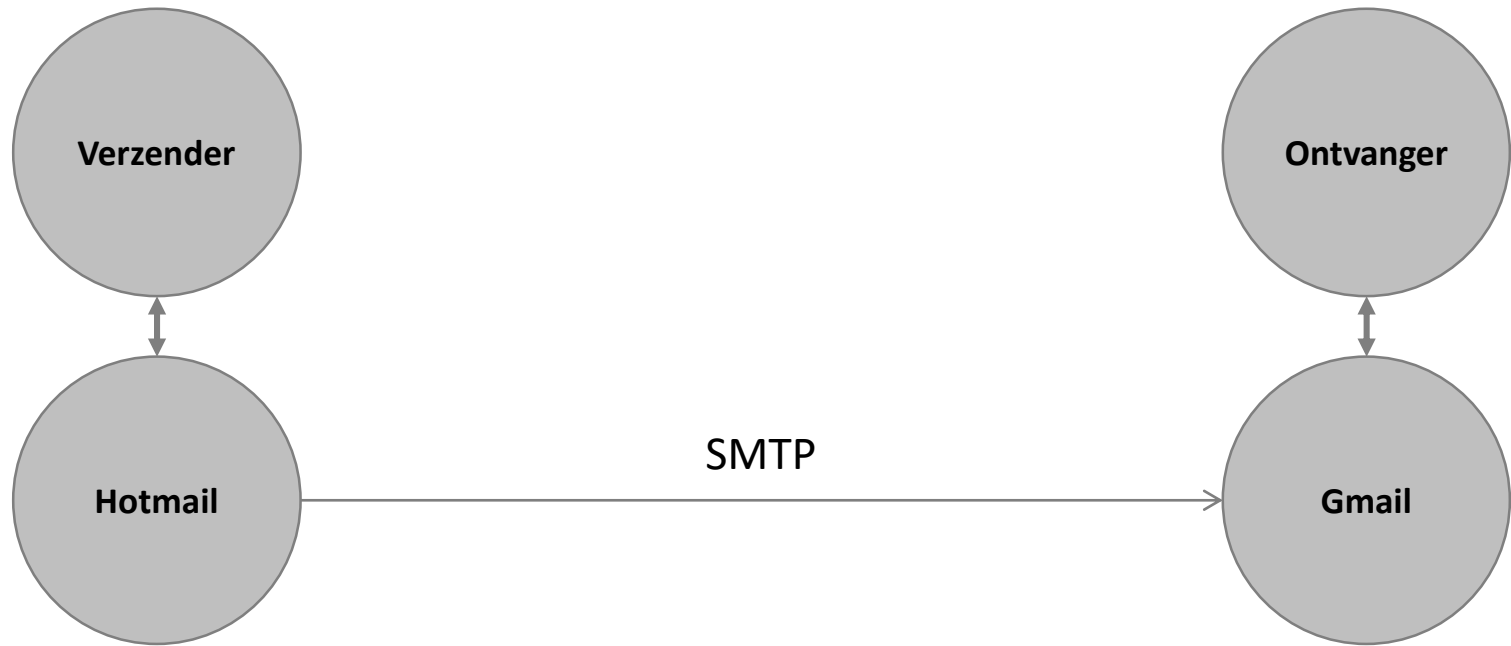
	Taxed	Unit cost	Qty:	Price
mydomain.com	No	€ 50.00 EUR	1	€ 50.00 EUR
et	No	€ 25.00 EUR	1	€ 25.00 EUR
stren	No	€ 25.00 EUR	1	€ 25.00 EUR
		Subtotal	1	€ 25.00 EUR

# De sleutel tot een compelling propositie is open platforms

# De evolutie van samenwerking...







**En net als het SMTP protocol**

**Dat email providers aan elkaar verbindt**



**Is Simplerinvoicing het protocol dat ERP en E-facturatie dienstverleners aan elkaar verbindt**

# Als we het opnieuw zouden bedenken...



1. **Standaard voor een e-factuur (in UBL)**
2. **Protocol voor transport (Vrij of PEPPOL)**
3. **Afspraken over veiligheid en betrouwbaarheid**

# SI-UBL: bouwsteen voor e-factureren

## doc:Invoice

[collapse all](#)

- [1..1] **doc:Invoice**
  - [1..1] **cbc:UBLVersionID**
  - [1..1] **cbc:CustomizationID**
  - [1..1] **cbc:ProfileID**
  - [1..1] **cbc:ID**
  - [1..1] **cbc:IssueDate**
  - [0..1] **cbc:DueDate**
  - [1..1] **cbc:InvoiceTypeCode**
  - [0..1] **cbc:Note**
  - [0..1] **cbc:TaxPointDate**
  - [1..1] **cbc:DocumentCurrencyCode**
  - [0..1] **cbc:TaxCurrencyCode**
  - [0..1] **cbc:AccountingCost**
  - [0..1] **cbc:BuyerReference**
  - [0..1] **cac:InvoicePeriod**
  - [0..1] **cac:OrderReference**
  - [0..1] **cac:BillingReference**
  - [0..1] **cac:ContractDocumentReference**
  - [0..n] **cac:AdditionalDocumentReference**
  - [0..1] **cac:ProjectReference**
  - [1..1] **cac:AccountingSupplierParty**
  - [1..1] **cac:AccountingCustomerParty**
  - [0..1] **cac:PayeeParty**
  - [0..1] **cac:BuyerCustomerParty**
  - [0..1] **cac:TaxRepresentativeParty**
  - [0..1] **cac:Delivery**
  - [0..1] **cac:DeliveryTerms**
  - [0..n] **cac:PaymentMeans**
  - [0..1] **cac:PaymentTerms**
  - [0..n] **cac:AllowanceCharge**
  - [0..1] **cac:TaxExchangeRate**
  - [1..1] **cac:TaxTotal**
  - [1..1] **cac:LegalMonetaryTotal**
  - [1..n] **cac:InvoiceLine**

## Invoice



Element [XML Examples](#) [Change requests](#) 

[Next version >>](#)

### Invoice type: doc:InvoiceType

#### Definition

Root element of the invoice document

#### Path

Invoice

#### Usage notes

Note: For document sizes up to 10MB the processing should be guaranteed. For other sizes processing guarantees should be discussed bilaterally.

Note: It is advised for the issuer/sender to check the validity of all data.

Note: Use the GS1 specification for the calculation of the amounts.

Note: Apart from the cardinality in this message format, the document must comply with applicable national and international law.

Note: The resulting XML document may not contain any empty elements.

#### Sub elements

#	Element	Definition	Multiplicity
1	<a href="#">UBLVersionID</a>	Identifies the earliest version of the UBL 2 schema for this document type that defines all of the elements that might be encountered in the current instance.	1..1
2	<a href="#">CustomizationID</a>	Identifies a user-defined customization of UBL for a specific	1..1

## 8 - RFC019 - OPTIONAL support for 'g-rekening' use-case

### Change request

Submitter	Date submitted	Status
Jaap Jan Nienhuis	22-07-2016	New

### References

[cac:PaymentMeans](#)
[cac:PaymentTerms](#)

### Request

In construction and HR industries, invoices often split an invoice into two payments:

1. Payment to a beneficiary account
2. Payment to a blocked 'g-account'.

The g-account is blocked by the bank and can only be used for payment to the tax authority

Unfortunately within UBL one is unable to include different amounts per PaymentMeans clause.

### Proposed Solution

The functionality is optional for both Sender and Receiver.

The policy is:

- Sending Service MAY send an invoice with a specified G-Account
- Such an invoice SHALL NOT contain other settlement instructions such as a discount or penalty charge
- The Receiving Service MAY be able to process the G-account information in the Invoice
- The Receiving Service MUST be able to process an invoice with G-account information in the Invoice

The proposed solution is:

Include 2 PaymentMeans clauses where each PaymentMeans is identified using the <ID> subelement.

1. The first PaymentMeans clause specifies the normal payment to beneficiary, and has an ID 'BENEFICIARY'
2. The second PaymentMeans clause specifies the g-account payment, and has an ID 'GACCOUNT'

Then the PaymentTerms are used to specify different amounts per PaymentMeans:

1. The first PaymentTerms refers to PaymentMeans 'BENEFICIARY' and contains the amount of payment to the beneficiary.
2. The second PaymentTerms refers to PaymentMeans 'GACCOUNT' and contains the amount of payment to the g-account number.

EXAMPLE (for an invoice where <PayableAmount> is EUR 210, to be split in EUR 120 and EUR 90:

```

<cac:PaymentMeans>
  <cbc:ID>BENEFICIARY</cbc:ID>
  <cbc:PaymentMeansCode listID="UNCL4461">30</cbc:PaymentMeansCode> <!-- credit transfer -->

```

si.t4simm.nl

# Samengevat

- Er is maar 1 manier om e-factureren adoptie te krijgen: propositie
- Een compelling propositie vraagt open samenwerking
- Simplerinvoicing is een open samenwerking met een 'standaard API': een protocol en UBL standaard
- Dus e-factureren kan al laagdrempelig, en zal de komende periode nog laagdrempeliger worden

[www.simplerinvoicing.org](http://www.simplerinvoicing.org)

# Vragen?

