

# ODF AND ITS TOOLKIT

FOSDEM  
FEB. 1, 2025

[Svante.Schubert@gmail.com](mailto:Svante.Schubert@gmail.com)



# OASIS OpenDocument Technical Committee (TC)

## Voting Members

- Patrick Durusau [**OASIS TC chair & editor**] (freelancer)
- Svante Schubert [**OASIS TC chair & editor**] (freelancer)
- Francis Cave [**OASIS TC secretary & editor**] (freelancer)
- Michael Stahl [**OASIS TC editor**] (*allotropia*)
- Regina Henschel (*TDF*)
- Alfred Hellstern (*Microsoft*)
- Prof. Andreas Guelzow (worked on Gnumeric - left mid 2023)

# Basics

## What is a Standard?

- A standard is a **blueprint** like a **cooking recipe!**
- A standard creates interoperability (e.g. DIN A4)
- A standard improves reusability
- A standard prevents Lock-In-Effect
- A standard is lowering costs (e.g. share tests & validator)

# ODF Toolkit

## Status

- ODF Toolkit Release overdue (usually one end-of-year)
- ODF Reference Implementation by ODFDOM possible
- ODF Toolkit Code generation blocking 1.0 release

# OASIS and ISO ODF Standard

## Status

- ODF 1.3 was published (27. April 2021)
- ISO standard ODF 1.3 is “in the queue”
- OASIS ODF 1.4 standard is “in the queue”
  - Awaiting further reference implementations  
aside [Microsoft 365 apps ODF 1.4 support](#)
- Work on ODF 1.5 standard has started!

# Larger Innovation still possible?

Too much maintenance?



# Basics

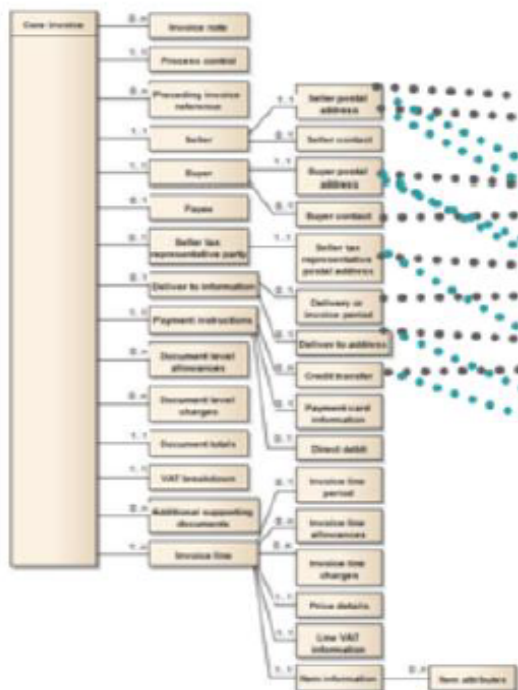
What is Semantik & Syntax?

- **Semantic** (*Wine*) described by **Syntax**/Language (*wine/vino*)



# Syntax Bindings

## Connecting Semantik & Syntax (EU eInvoice)



**Semantic data model of the core elements**  
(Business terms and rules)

Syntax bindings

```
<Invoice xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2" xsi:  
<cbc:UBLVersionID>2.1</cbc:UBLVersionID>  
<cbc:CustomizationID>urn:www.cenbii.eu:transaction:biitms010:ver2.0:ext:  
<cbc:ProfileID>urn:www.cenbii.eu:profile:bi05:ver2.0</cbc:ProfileID>  
<cbc:ID>998877</cbc:ID>  
<cbc:IssueDate>2013-05-31</cbc:IssueDate>  
<cbc:InvoiceTypeCode listID="UNCL1001">380</cbc:InvoiceTypeCode>  
<cbc:Note>PEPPOL BIS 5a: The "MAX DATA" test case. Test of certain s  
<cbc:TaxPointDate>2013-05-31</cbc:TaxPointDate>  
<cbc:DocumentCurrencyCode listID="ISO4217">SEK</cbc:DocumentCur  
<cbc:TaxCurrencyCode listID="ISO4217">EUR</cbc:TaxCurrencyCode>  
<cbc:AccountingCost>Project-123</cbc:AccountingCost>  
<cac:InvoicePeriod>  
<cbc:StartDate>2013-05-01</cbc:StartDate>  
<cbc:EndDate>2013-05-31</cbc:EndDate>  
</cac:InvoicePeriod>
```

OASIS UBL 2.1

```
<ram:CrossIndustryInvoice xmlns:xi="http://www.w3.org/2001/XMLSchema-instance" xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDataTypes-3" xmlns:ram="urn:oasis:names:specification:ubl:schema:xsd:ReusableAggregateBusinessInformationEntity-3" xsi:schemaLocation="http://www.oasis-open.org/ubl/schemas/ubl2-1/ubl2-1.xsd http://www.oasis-open.org/ubl/schemas/ubl2-1/ubl2-1.xsd" >  
<ram:ExchangedDocument>  
<ram:ID>123</ram:ID>  
<ram:TypeCode>380</ram:TypeCode>  
<ram:IssueDateTime>  
<udt:DateTimeString>20130601</udt:DateTimeString>  
</ram:IssueDateTime>  
<ram:EffectiveSpecifiedPeriod>  
<ram:StartDateTime>  
<udt:DateTimeString>20130501</udt:DateTimeString>  
</ram:StartDateTime>  
<ram:EndDateTime>  
<udt:DateTimeString>20130601</udt:DateTimeString>  
</ram:EndDateTime>  
<ram:EffectiveSpecifiedPeriod>  
<ram:IssuerTradeParty>  
<ram:ID>token</ram:ID>  
<ram:GlobalID>token</ram:GlobalID>  
<ram:Name>String</ram:Name>
```

Cross Industry Invoice D16B



# Connecting Semantic with Syntax

Syntax Binding (EU e-invoice) (~400 pages of buyable PDF tables)

ID	Level	Card.	BT	Desc.	DT	Path	Type	Card.	Match	Rules
BT-1	1	1..1	Invoice number	A unique identification of the Invoice.	I	/Invoice/cbc:ID	I	1..1		
BT-2	1	1..1	Invoice issue date	The date when the Invoice was issued.	D	/Invoice/cbc:IssueDate	D	1..1		
BT-3	1	1..1	Invoice type code	A code specifying the functional type of the Invoice.	C	/Invoice/cbc:InvoiceTypeCode	C	0..1	CAR-2	
BT-5	1	1..1	Invoice currency code	The currency in which all Invoice amounts are given, except for the Total VAT amount in accounting currency.	C	/Invoice/cbc:DocumentCurrencyCode	C	0..1	CAR-2	
BT-6	1	0..1	VAT accounting currency code	The currency used for VAT accounting and reporting purposes as accepted or required in the country of the Seller.	C	/Invoice/cbc:TaxCurrencyCode	C	0..1	SEM-2	

<https://www.evs.ee/en/cen-ts-16931-3-2-2020>

# Connecting Semantic with Syntax

Syntax Binding (EU e-invoice) - extracted as XML from ODT tables using ODFDOM

ID	Level	Card.	BT	Desc.	DT	Path	Type	Card.	Match	Rules
BT-1	1	1..1	Invoice number	A unique identification of the Invoice.	I	/Invoice/cbc:ID	I	1..1		
BT-2	1	1..1	Invoice issue date	The date when the Invoice was issued.	D	/Invoice/cbc:IssueDate	D	1..1		
BT-3	1	1..1	Invoice type code	A code specifying the functional type of the Invoice.	C	/Invoice/cbc:InvoiceTypeCode	C	0..1	CAR-2	

```
<semantic id="BT-1" level="1" card="1..1" bt="Invoice number" desc="A unique identification of the Invoice.">
  <xml path="/Invoice/cbc:ID" card="1..1"/>
</semantic>
<semantic id="BT-2" level="1" card="1..1" bt="Invoice issue date" desc="The date when the Invoice was issued.">
  <xml path="/Invoice/cbc:IssueDate" card="1..1"/>
</semantic>
<semantic id="BT-3" level="1" card="1..1" bt="Invoice type code" desc="A code specifying the functional type of the Invoice.">
  <xml path="/Invoice/cbc:InvoiceTypeCode" card="0..1"/>
</semantic>
```

<https://github.com/svanteschubert/en16931-data-extractor>

<https://github.com/svanteschubert/en16931-data-extractor>

# ODF / OOXML Interoperability

Often adding OOXML functionality to ODF

- How about creating a Syntax Binding for ODF/OOXML?
- Starting to add define Semantics in ODF Spec
- Syntax was never the first citizen – Semantic is!
- Large Language Models draft Syntax Mappings
- Already in LO Code, unclear what's missing!

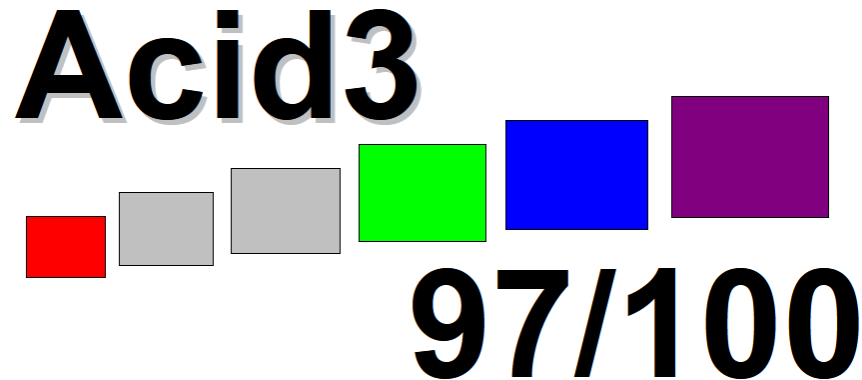
# ODF Runtime API

Syntax is filtered away, Semantic stays!

- All browsers have a Runtime DOM API! Most powerful!
- **Syntax is filtered away, Semantic stays!**
- Have a **Semantic DOM** API (SDOM)
- Required to interoperable  
communicate changes across ODF Apps
- One SDOM Change (one user change)  
mapped to list of LO API calls
- First Step:  
Make ODF Semantics/Features explicit in the ODF Spec
- BUT aim to embrace the complexity of OOXML

# Document Standards

## API on Semantic & Syntax



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this reference rendering](#).

- ODF XML only the final state of a document (load / save)
- ODF defines no state changes (e.g. column insertion in table)
- No state changes == no API == no regression test at standard  
Like [HTML/CSS ACID regression tests](#)

# ODF API based Feature Testing

## Guarantee Enterprise Functionality

- 1) Load Testdoc
- 2) Do ODF API (or equivalent GUI) change
- 3) Test with expected Layout (e.g. bullets correct positioned?)
- 4) Save Testdoc
- 5) Test with expected ODF XML

# ODF Toolkit

## Resources

- **Website:**

<https://odftoolkit.org/>

<https://tdf.github.io/odftoolkit/docs/> (latest)

- **Sources:**

<https://github.com/tdf/odftoolkit>

- **Online Validator (hosted by TDF)**

<https://odfvalidator.org/>

- **ODF Specification**

<http://docs.oasis-open.org/office/OpenDocument/v1.3/os/>

- **ODF Specification Tooling**

- <https://github.com/oasis-tcs/odf-tc/>