RUHR-UNIVERSITÄT BOCHUM

Bridging the Gap: Secure and lossless conversion of XML data structures into the JSON format

Bachelor thesis ■ March 30, 2017 – June 29, 2017

April 11, 2017

Advisors: Dennis Felsch & Paul Rösler

Jan Holthuis

hgi Lehrstuhl für Netz- und Datensicherheit

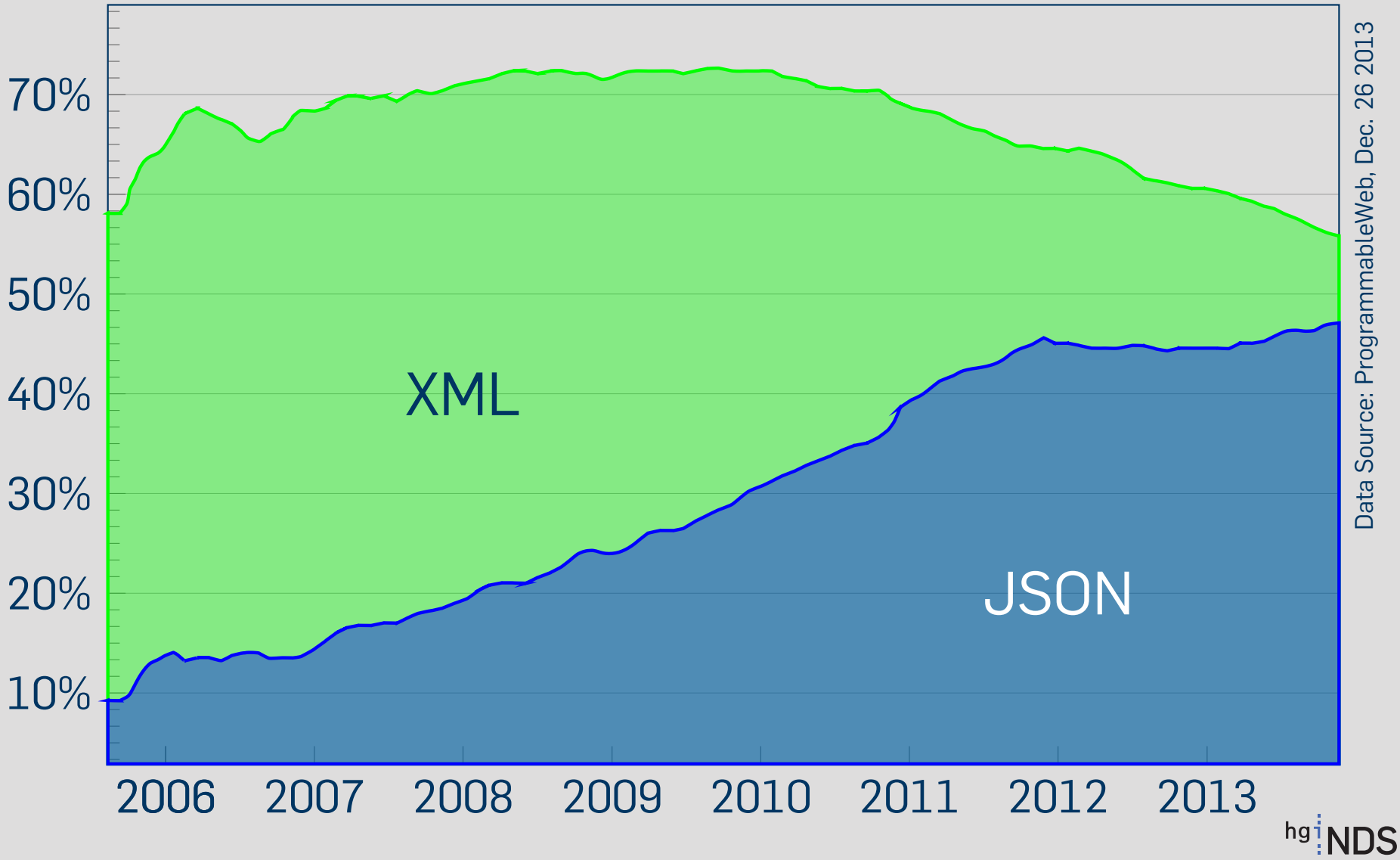
Overview

Usage of JSON and XML

- Web APIs are booming since *Web 2.0* and *IoT* hype
  - most of them use XML, JSON or both as data format
- Some “normal” websites are now based on these formats (e.g. *AngularJS*)
- Lots of file formats are XML-based (e.g. RSF/ASF, MathML, SVG, XHTML, ODT, OOXML, . . .)
- There are even JSON-based databases like *CouchDB* and *MongoDB*
- Countless industry standards in all sectors use XML

Support by Web APIs

From mid 2005 until end of 2013



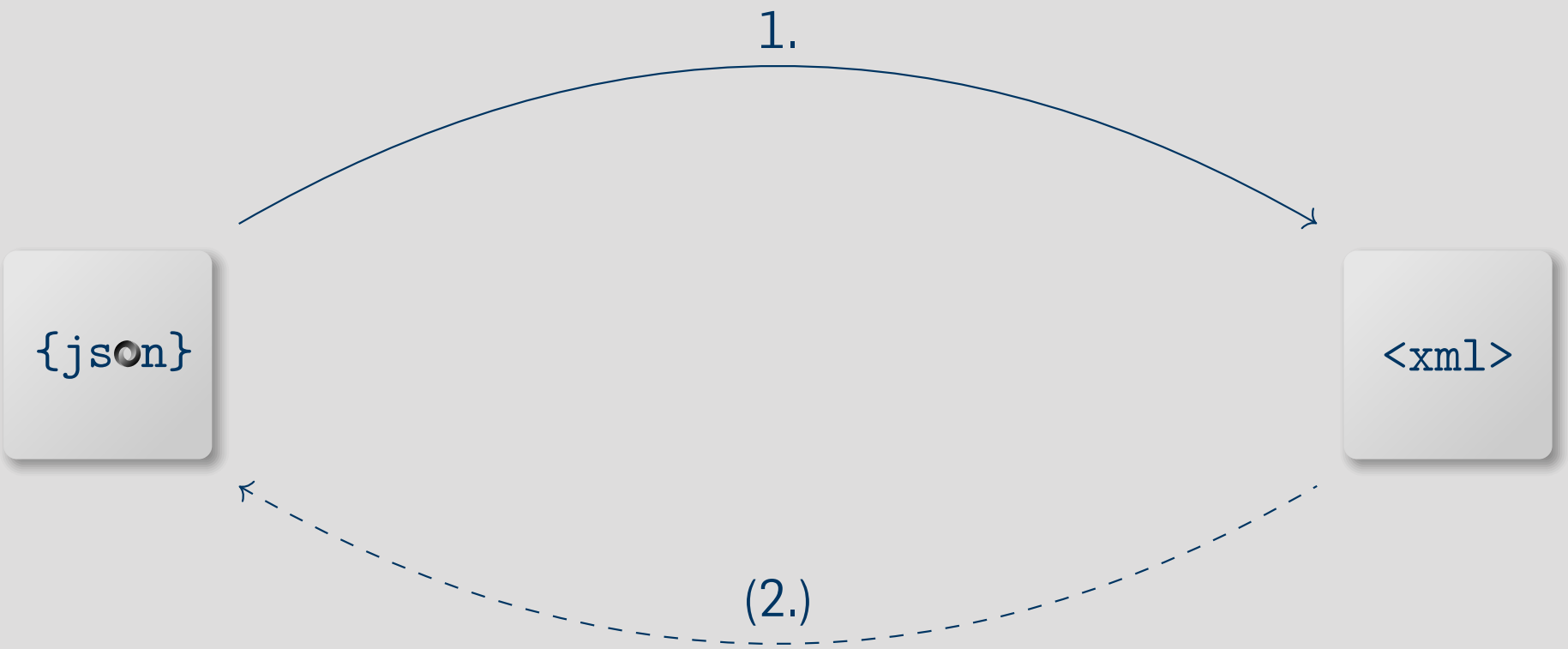
Why convert between XML and JSON?

- Parsing JSON is usually faster and less resource heavy than XML
- XML has more features and is widely used by the industry
- . . . but the complexity makes it harder for humans to read and adds more overhead
- Support by programming languages, frameworks and libraries is inconsistent

⇒ **plenty of reasons for converting between XML and JSON!**

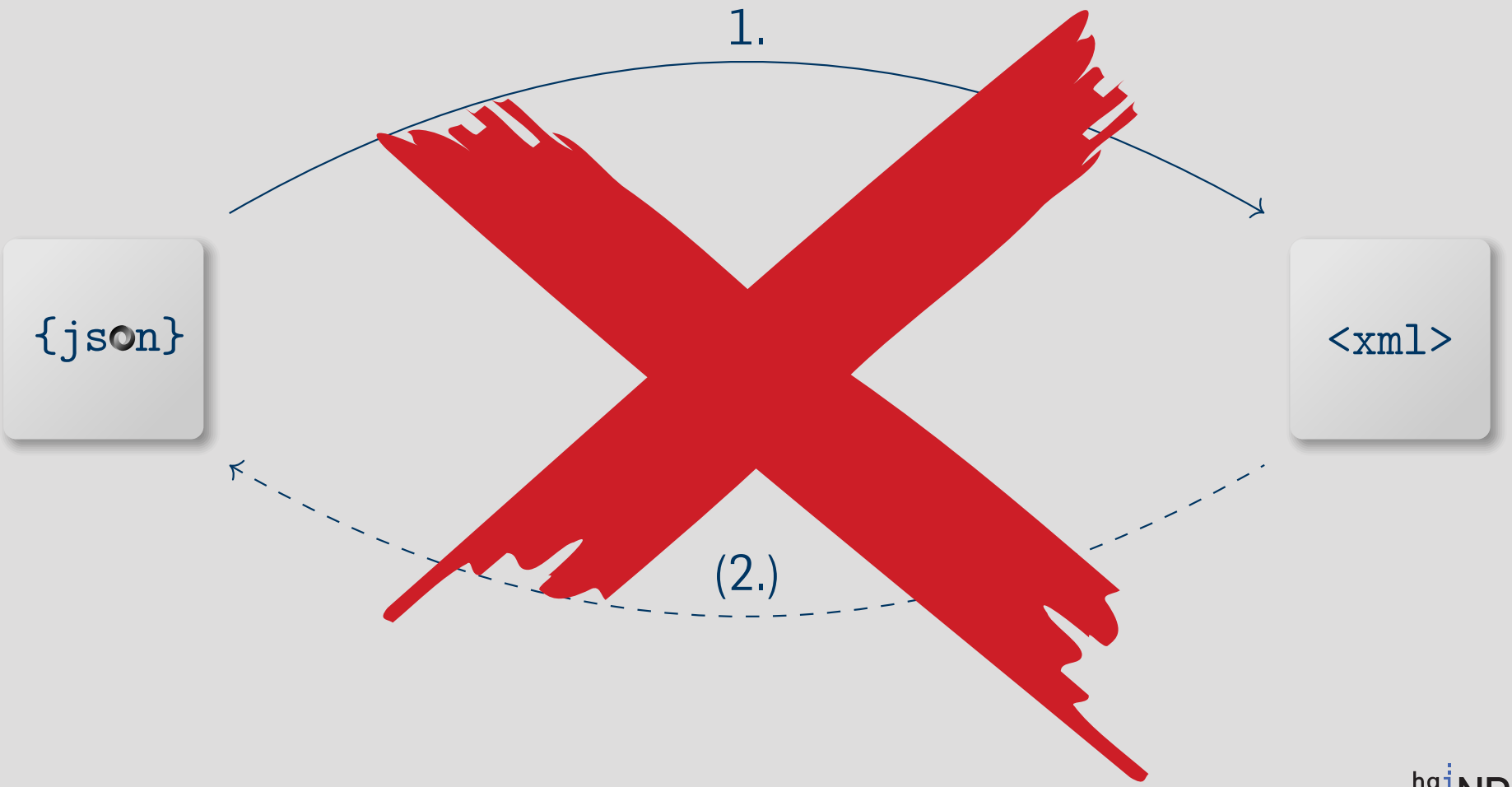
Conversion != Conversion

Converting arbitrary JSON into XML-based format



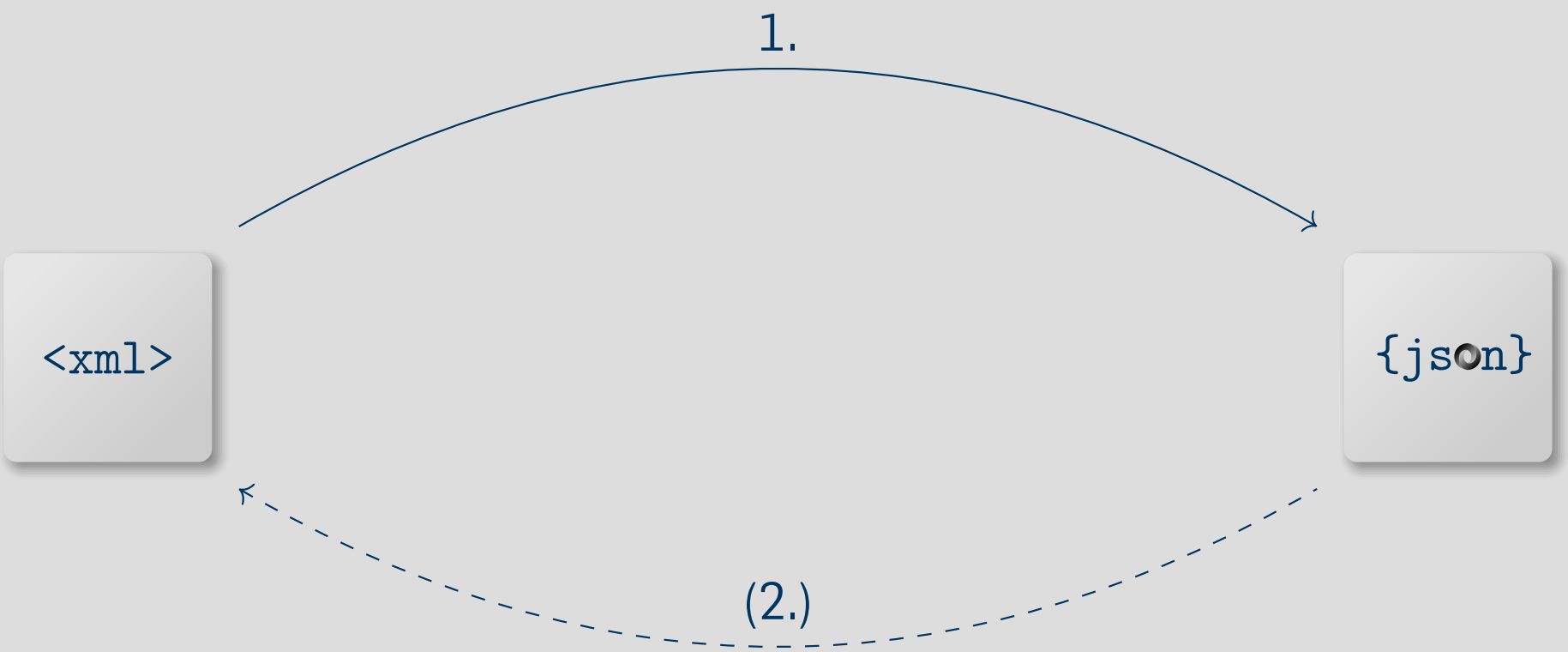
Conversion != Conversion

Converting arbitrary JSON into XML-based format



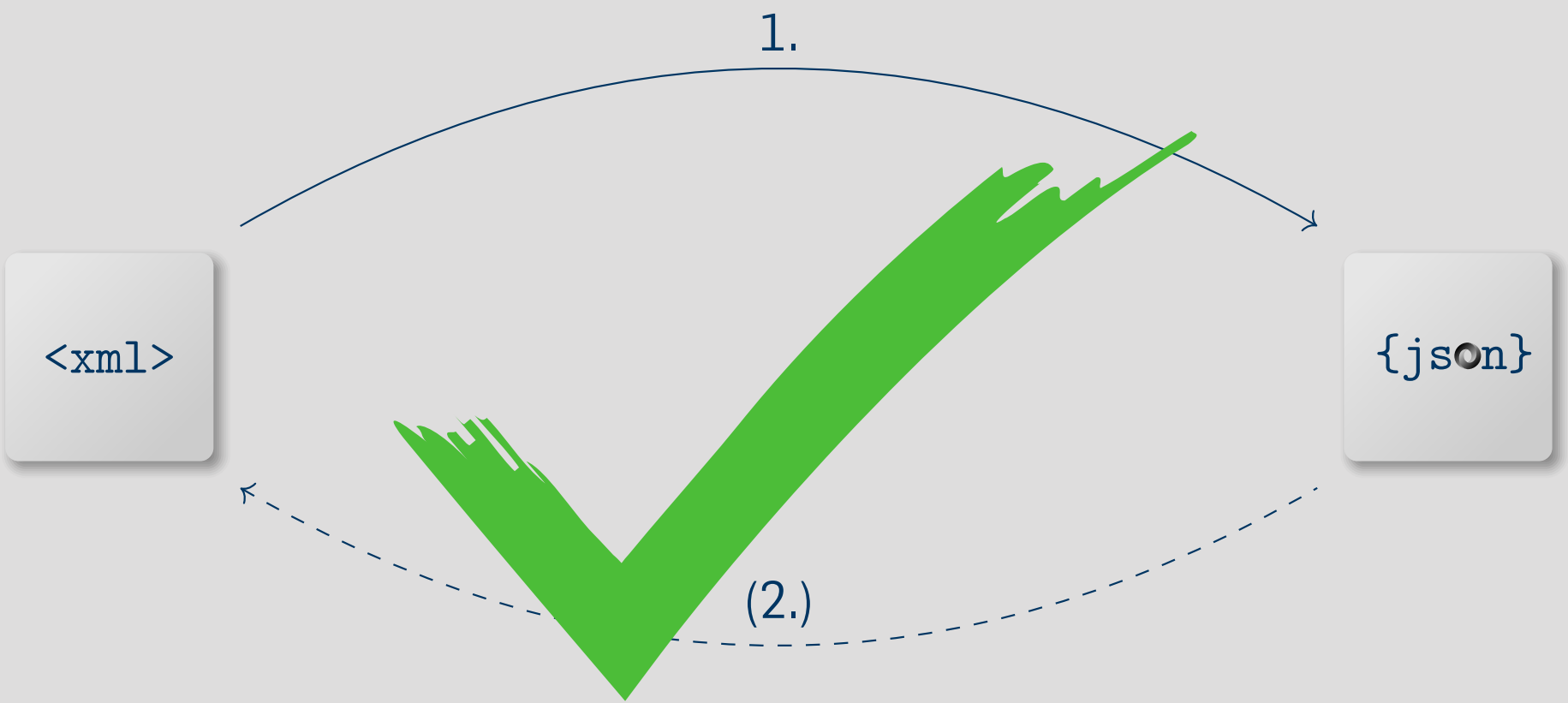
Conversion != Conversion

Converting XML into JSON-based format



Conversion != Conversion

Converting XML into JSON-based format



Scope of thesis

- Find a way to convert arbitrary XML documents into JSON
- Be able to convert the JSON documents back to XML
- The conversion should …
  - result in **well-formed JSON/XML**,
  - **require no additional metadata** (type hints, etc.),
  - be **lossless** and
    - XML documents before and after XML → JSON → XML round-trip should be (logically) equivalent
  - be **secure**.
    - Not vulnerable to known attacks against parsers

Overview

XML

Overview

- *“eXtensible Markup Language”*
- Derived from SGML (ISO 8879)
- First published by W3C in 1998
- Currently two flavors:
  - XML 1.0 (Fifth Edition), published November 26, 2008
  - XML 1.1 (Second Edition), published August 15, 2006

XML

Related specifications

- XML Schema Definition (XSD)
- XML Path Language (XPath)
- XML Pointer Language (XPointer)
- XML Query (XQuery)
- XML Signature
- XML Encryption
- XML Remote Procedure Call Protocol (XML-RPC)
- Simple Object Access Protocol (SOAP)
- and many more...

JSON

Overview

- *"JavaScript Object Notation"*
- Popularized by Douglas Crockford in the early 2000s
- First specified officially as RFC 4627 (2006)
- Currently defined by:
  - RFC 7159, published in March 2014
  - ECMA-404, published in October 2013

JSON

Related specifications

- JSON Schema
- XPath for JSON (JSONPath)
- JSON Pointer
- JSON Query Language (JSONiq)
- JSON Web Signature (JWS)
- JSON Web Encryption (JWE)
- JSON Remote Procedure Call Protocol (JSON-RPC)
- SOAP using JSON-RPC (SOAPjr)
- and many more...

XML and JSON

Related specifications

RUB

XML

- Schema Definition (XSD)
- XPath
- XPointer
- XQuery
- XML Signature
- XML Encryption
- XML-RPC
- SOAP

JSON

- JSON Schema
- JSONPath
- JSON Pointer
- JSONiq Query Language
- JSON Web Signature (JWS)
- JSON Web Encryption (JWE)
- JSON-RPC
- SOAPjr

I SEE WHAT YOU DID THERE...

hgi NDS

Overview

Converting XML to JSON

It's complicated...

XML

```xml
<albums>
  <album id="1">
    <title>Courts the Count</title>
    <artist>Shorty Rogers</artist>
    <year>1954</year>
  </album>
  <album id="2">
    <title>Birth of the Cool</title>
    <artist>Miles Davis</artist>
    <year>1949</year>
  </album>
</albums>
```

JSON

```json
{ "albums": [
  { "id": 1,
    "title": "Courts the Count",
    "artist": "Shorty Rogers",
    "year": 1954
  },
  { "id": 2,
    "title": "Birth of the Cool",
    "artist": "Miles Davis",
    "year": 1949
  }
]}
```

Converting XML to JSON

It's complicated...

RUB

XML

```
<albums>
  <album id="1">
    <title>Courts the Count</
    <artist>Shorty Rogers</arti
    <year>1954</year>
  </album>
  <album id="2">
    <title>Birth of the Coo
    <artist>Miles Davis<
    <year>1949</year>
  </album>
</albums>
```

JSON

```
{ "alb
  {
      ": "Courts the Count",
     tist": "Shorty Rogers",
    'year": 1954

          2,
          "Birth of the Cool",
          Miles Davis",
    "yea
  }
]}
```

hgNDS

XML vs. JSON

Features

RUB

XML

- Document based format
- It's extensible!
- Supports attributes, comments, namespaces, CDATA, etc.

JSON

- Data based format
- Not extensible
- Does not support those features.

hgi NDS

XML vs. JSON

Datatypes

RUB

XML

No datatype support:

- Everything is a string
- If you want to specify types, you need a *schema*

JSON

Syntactic datatype support for:

- Strings
- Numbers
  - Integers
  - Fractions
  - Exponents
- Arrays
- Booleans (`true`, `false`)
- `null`

hgi NDS

XML vs. JSON

Security Vulnerabilities

XML

Various generic attacks on XML Parsers:

- Denial of Service Attacks (“*Billion Laughs*”, *Quadratic Blowup Entity Expansion*)
- Local/Remote File Inclusion (LFI/RFI) using *External Entity Expansion*
- Server-Side-Request Forgery (SSRF) via DTD Retrieval

JSON

Several attacks target JavaScript:

- Cross-Site-Scripting (XSS) if JavaScript’s `eval()` is used instead of`JSON.parse()`
- In-browser XSS/CSRF attacks against JSON-P
- XSS if a JSON web service does not set the `Content-Type`
- . . . but no attacks on JSON parsing itself!

Overview

Working Method

1. Look at existing solutions
2. Establish criteria for conversion quality
3. Check which converters satisfy those criteria (if any)
4. If no converters tick all the boxes $\rightarrow$ develop custom algorithm that does this
   - If possible, make necessary changes to existing solution instead of developing an algorithm from scratch

Current Status

Existing solutions

Already found some ways to convert between XML and JSON, e.g.

- *BadgerFish* convention (+ Ruby implementation, MIT License),
- *Parker* convention,
- *JSON Markup Language (JsonML)* (+ JavaScript implementation, MIT License),
- *x2js* (JavaScript, Apache 2.0 License),
- *json-lib* (Java, Apache 2.0 License),
- `org.json` package (Java, MIT License),
- *jxon* library (JavaScript, GNU Public License 3.0),
- *pesterfish* library (Python, MIT License).
- ...

Current Status

XML/JSON Conversion Checker (xjcc)

- Started development of Python tool to test conversion quality
- Currently ≈ 1100 LOC and 72% unittest coverage
- Converts XML documents to JSON and back, then compares XML output using *XML Canonicalization*
- Runs inside a docker container for easy setup
- Already created plugins to invoke most of the converters
- Can output results as text, CSV or JSON

Current Status

XJCC Tool demo

```
$ xjcc test-conversion

Converter 'x2js':
  [ FAILED ] complex-document
  [   OK   ] element-order
  [ FAILED ] whitespace
Converter 'jsonml':
  [   OK   ] complex-document
  [   OK   ] element-order
  [   OK   ] whitespace

```

Next Steps

- Add a way to check for security vulnerabilies
- Tool does not check if output is well-formed XML/JSON yet
- Research and establish conversion criteria
- Create test documents
- Evaluate current solutions using the test documents
- Possibly develop custom algorithm

Thanks!

RUB

Questions?

Reach out via email:

- **Jan Holthuis**
  jan.holthuis@rub.de