



# Markup and the Canadian National Model Building Codes

Balisage 2013, Montréal

Brent Nordin

# Introductory Comments

- Canadian National Model Building Codes:
  - Building
  - Fire
  - Plumbing
  - Energy Code for Buildings
  - Farm Buildings
- Developed by a secretariat at Canada's National Research Council
- Talk will cover markup aspects of the Codes documents:
  - XML content lifecycle
  - Publishing chains

# The XML Library

- Codes content converted from proprietary format to SGML for the 1995 printing
- Data model expressed as a DTD
- Converted to XML for 2005 printing
- One library for all Codes documents
- Content stored as XML fragments
  - A single sentence or table of the normative material
  - A larger self-contained chunk for non-normative material

# XML Library Structure

- A single tree of XML fragments
- Leaves contain the bulk of the Codes text (sentences, tables, appendix notes, intent analysis)
- Higher levels in the tree contain structural information fragments
- Tables, appendix notes, and intent analysis fragments are referenced from sentence fragments

# Leaf node

- Sentence, table, appendix note, objective analysis
- Rich structure
- Leaf nodes related to a single building Code sentence (provision).
- Tied together through REFIDs or Xlink:href



# Non-leaf nodes (structural fragments)

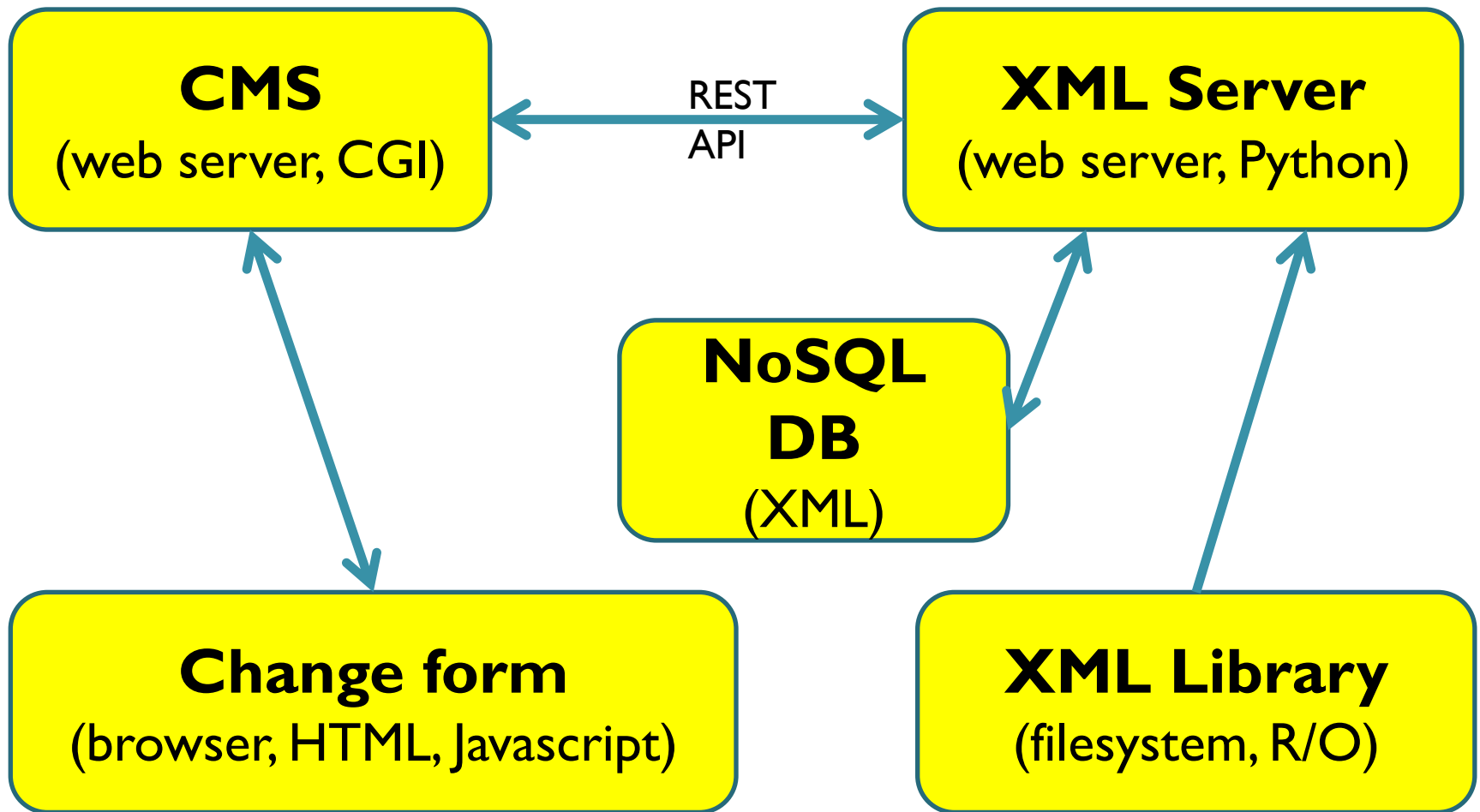
- Articles
  - Include a title
  - References to sentences and tables
- Parts
  - Include part/section/subsection
  - Include titles for each level
- Book
  - Include references to parts



# XML Fragment Maintenance

- Formal workflow
  - Acceptance for review
  - Technical committee work
  - Governance review
  - Stakeholder review
  - Public review
  - Editing
  - Translation
  - Publication
- Workflow now mediated by a CMS
  - Electronic form for each proposed change
  - XML fragments attached to form as separate documents

# System Architecture





# NoSQL Database

- One per Codes document
- Structural representation of each document
- Includes some publishing artifacts like sequence numbers
- Queries are parameterized XSLT transforms



# XML Server

- XML web server adds Codes-specific XML functionality to the CMS
- RESTful API
- Implemented in Python (on top of CherryPy framework)
- Interactions with the NoSQL DB and any generated XML documents based on XSLT or Xpath
  - No horrible DOM coding
  - No data model mismatches between XML and programming language

# Composite Fragments

- Attached to the Codes change form
- Built up in stages
  - 1st stage from NoSQL DB
  - 2<sup>nd</sup> stage adds referenced files
  - 3<sup>rd</sup> stage looks for appendix note references, builds link templates
  - 4<sup>th</sup> stage adds appendix notes, resolves link templates
- Creates a complete work package for the technical committees



# Fragment Bursting

- Edited composite fragments must be returned to XML library
- Editing governed by DTD (in Arbortext) so content is structurally sound
- Bursting process does limited semantic checking
  - ID, IDREF syntax
  - ID, IDREF links
- Recreates structural and leaf node XML fragments

# Publishing Chains

- XML to HTML
  - for preview and online viewing
- HTML to PDF
  - for print
- HTML to HTML
  - For side by side output
- XML to XML to HTML
  - for consolidated print

# XML to HTML

- Main publishing chain
- Multi-stage rendering chain
  - CMS server-side job setup
  - 2 preprocessing steps
  - Main rendering step
  - Post-filtering
  - HTML Tidy



# Side by Side Rendering

- Merge HTML versions of rendered French and English Codes change forms
- Relies on class information in HTML for synchronization



# Consolidated Print

- Merges all open proposed changes with reference content
- Designed to show what a Codes document would look like:
  - Relative to the previously published version
  - Including overlapping proposed changes
  - With new sequencing to allow for discussions





# Summary

- A semantically rich data model enables unforeseen capabilities
- A clear distinction between content and metadata has been critical
- The Canadian Building Codes will outlive both me and any markup technologies
  - SGML was good
  - XML is good
  - A markup technology yet to be invented will be good



Thank you

Brent Nordin

[anglebrackets@outlook.com](mailto:anglebrackets@outlook.com)