Xerces2:
The Sequel With No Equal

Andy Clark

Introduction

- Speaker
  - Worked for IBM
  - Currently unemployed ☺
- Parser
  - First developed in IBM's Tokyo research lab
  - Maintained and expanded in California
  - Donated to Apache
  - Work continues in Toronto

Agenda

- Xerces1 Overview
  - Design and problems
- Xerces2 Overview
  - Challenges and design
- Q & A

Xerces1 Overview:
Design and Problems

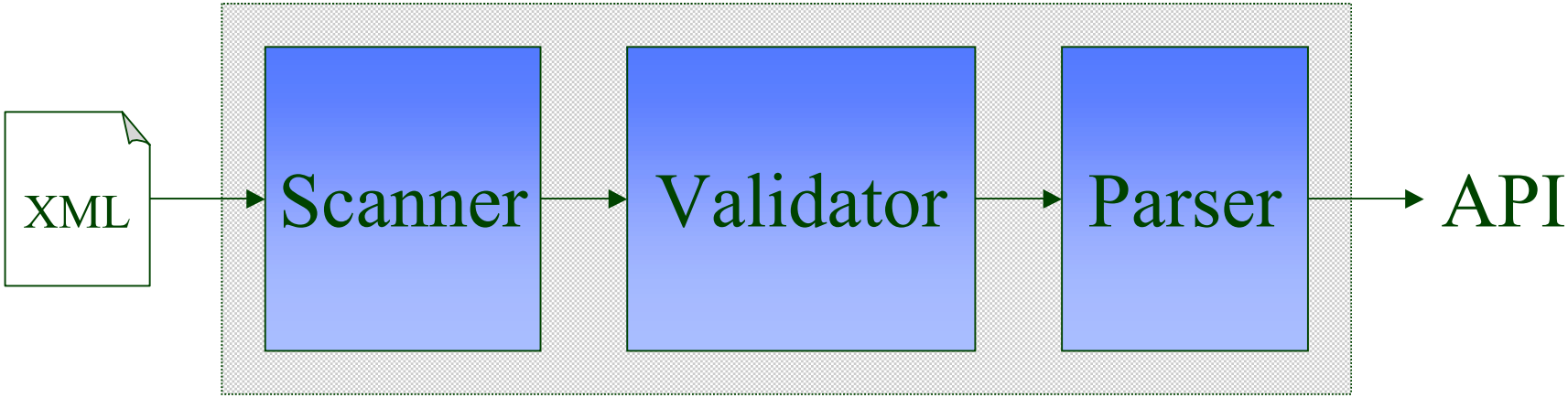
Andy Clark

Design

- XML4J/Xerces1 designed for performance
- Parser Implementation
  - Parsing pipeline
  - Custom reader implementations
  - StringPool
    - Defers transcoding of byte buffers until needed
    - Symbol table for common document strings

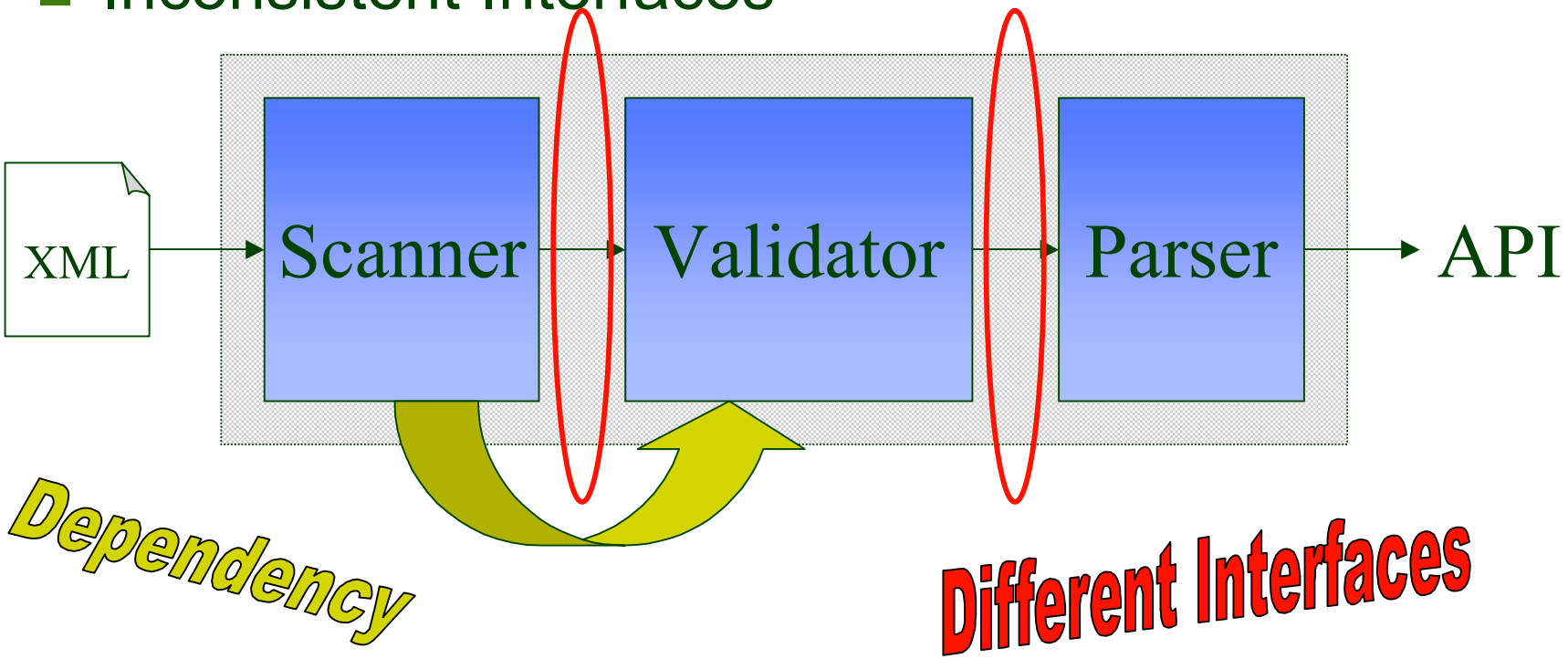
Pipeline Configuration

- Intended to be generic

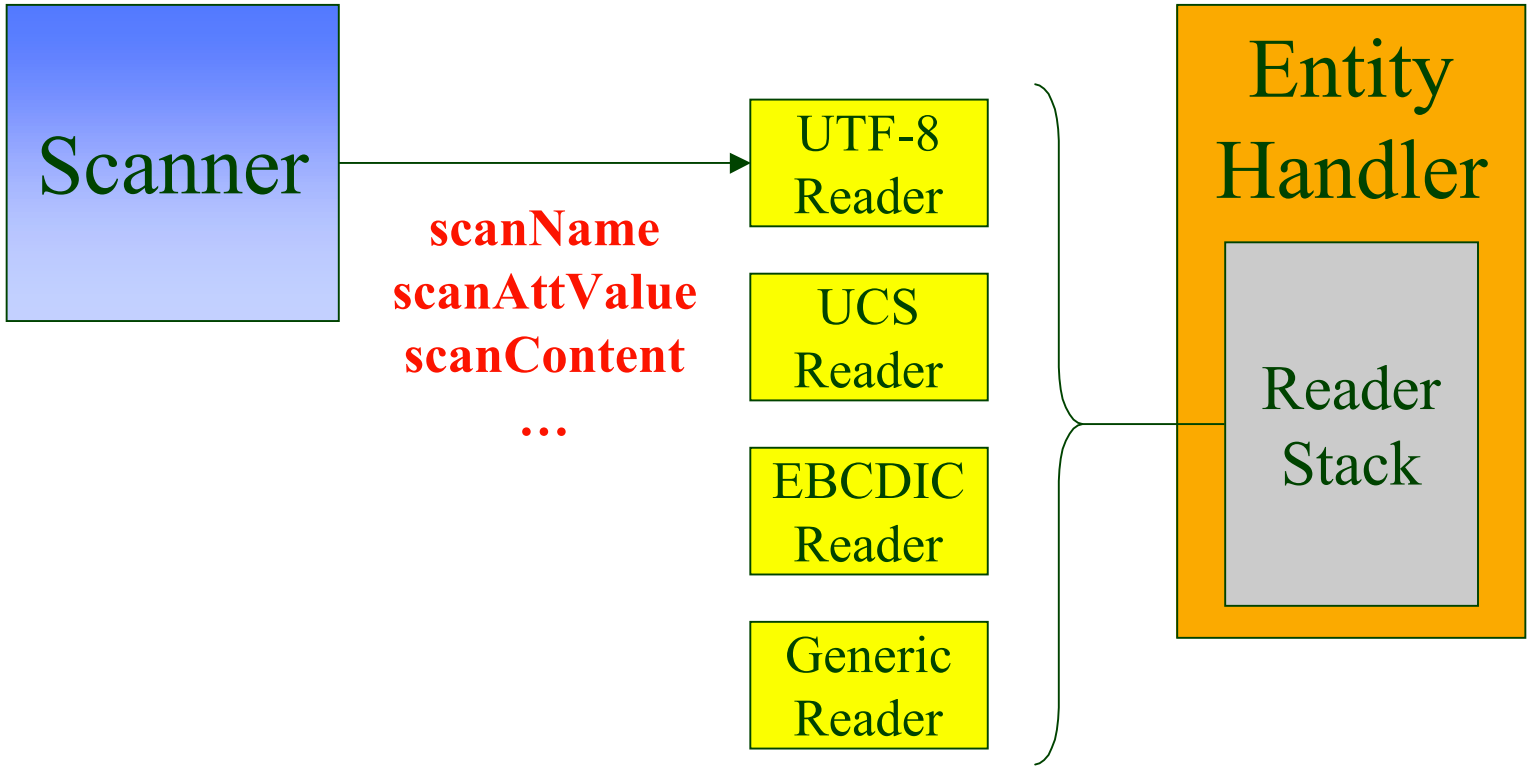


Pipeline Configuration Problems

- Hard-coded dependencies on implementation
- Inconsistent Interfaces



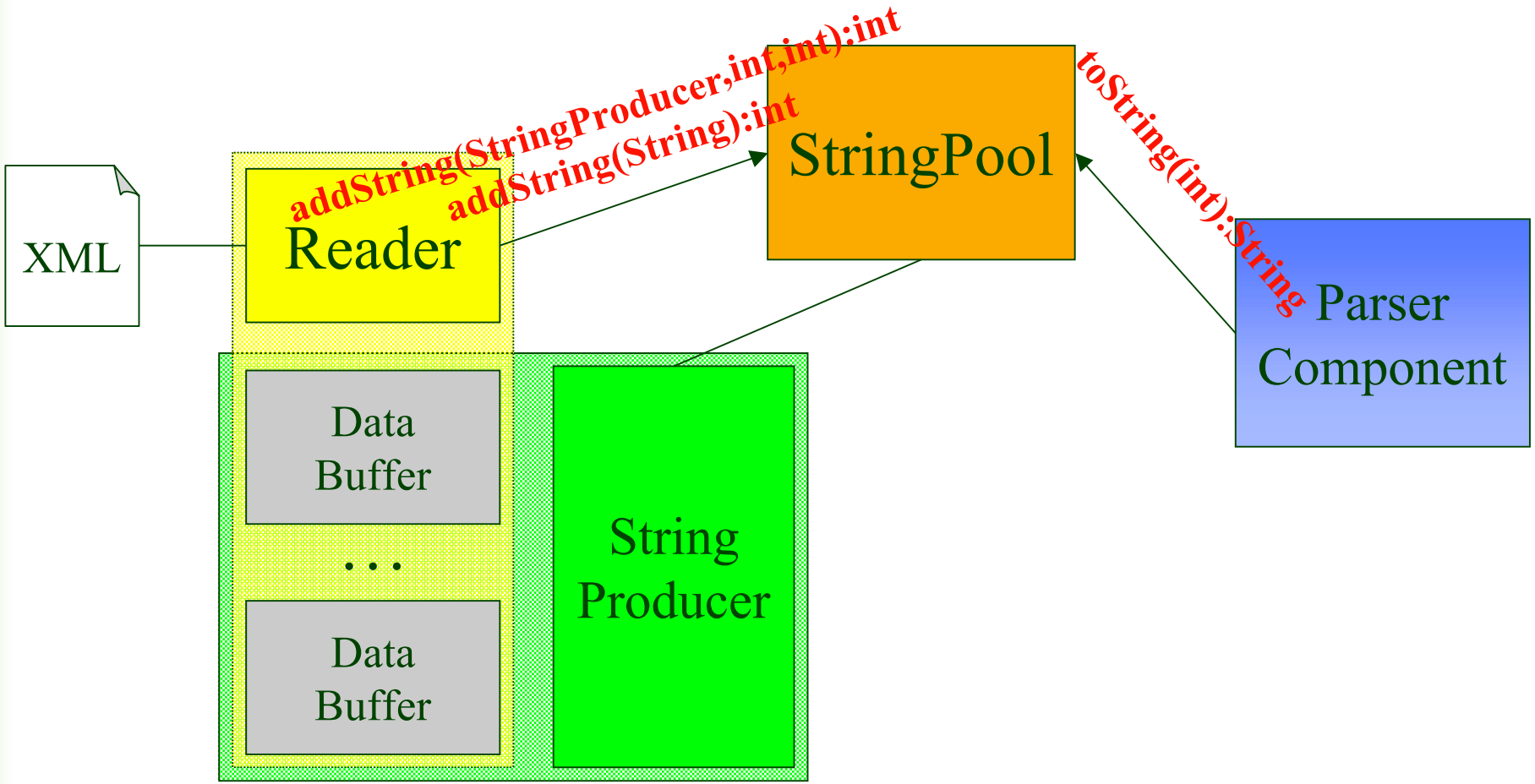
Custom Readers



Custom Readers Problems

- Duplicated code
  - Allows more bugs to appear
  - Bugs are different based on encoding because code is not shared
- More complicated

Deferred Transcoding



Deferred Transcoding Problems

- All components need reference to StringPool
  - Strings not immediately available to methods
  - Must make call to StringPool to query String
- Memory management is complicated
  - Responsibility of callee to free resources
  - Uses more memory

Xerces2 Overview:
Challenges and Design

Andy Clark

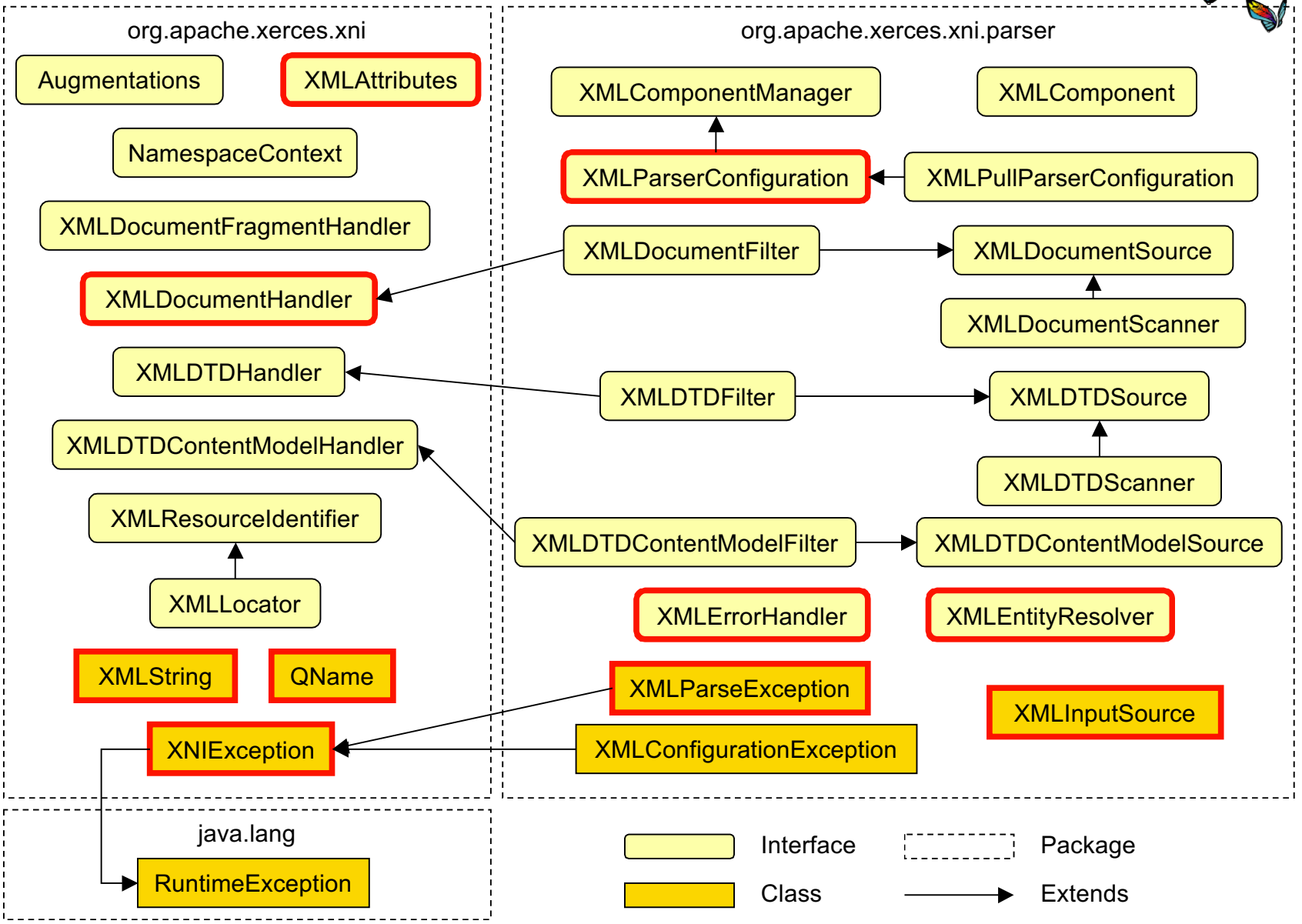
Challenges

- Requirements
  - Simple design and implementation
  - Easy to maintain
  - More modularity and configurability
  - Support current and future features
- Design Decisions
  - Always transcode bytes into Unicode characters
    - Removes StringPool and dependencies
  - Clean architecture

Xerces Native Interface (XNI)

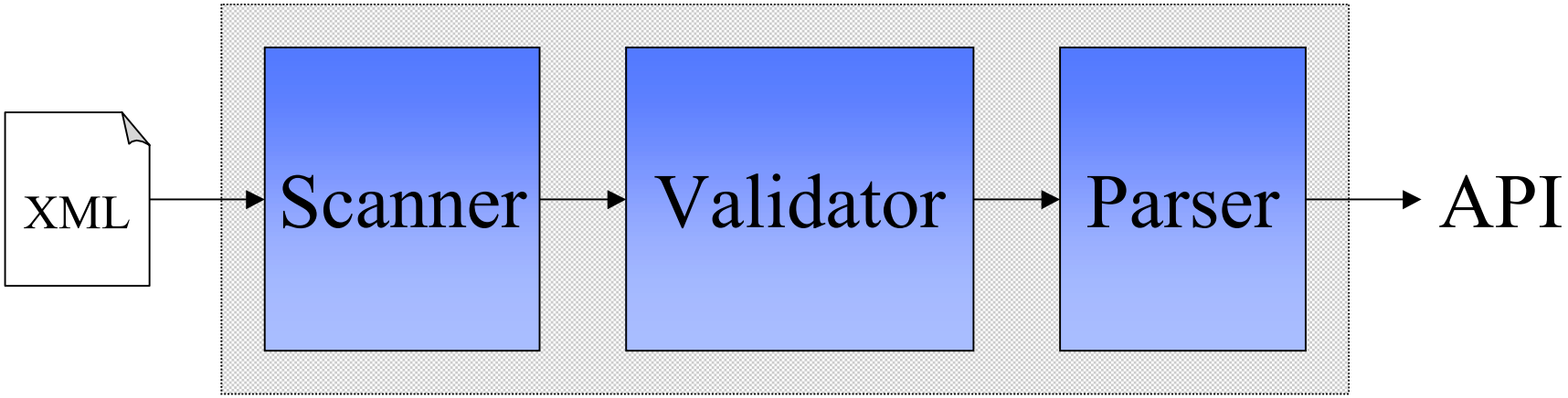
- "Streaming" Information Set
  - Similar to SAX
  - No loss of document information*
- Parser configuration and layering
- Future extensions
  - Native pull-parser, tree model, etc.

* Does not preserve *all* document information but communicates more information to the application than DOM or SAX.

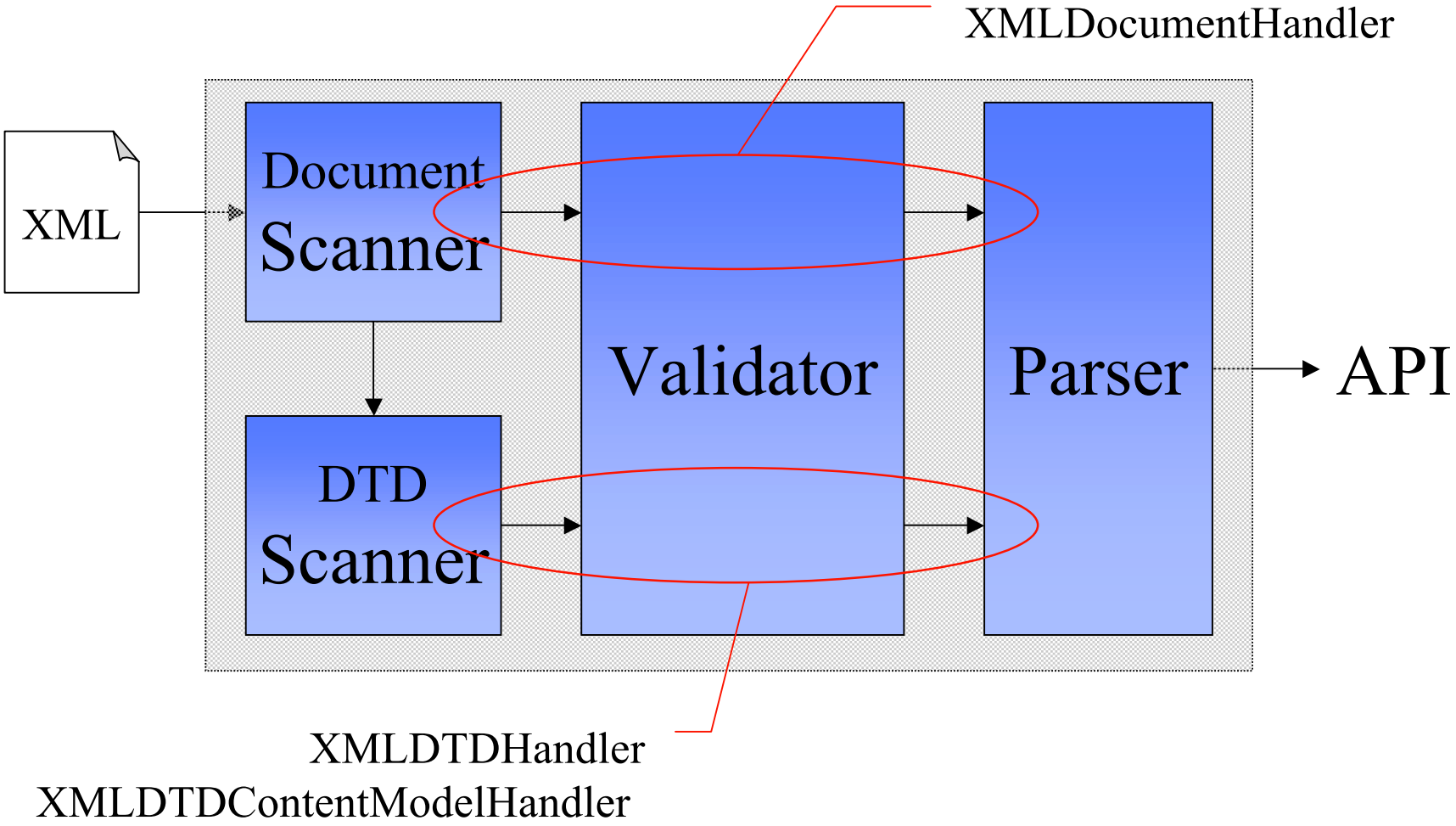

org.apache.xerces.xni
Augmentations
XMLAttributes
NamespaceContext
XMLDocumentFragmentHandler
XMLDocumentHandler
XMLDTDHandler
XMLDTDContentModelHandler
XMLResourceIdentifier
XMLLocator
XMLString
QName
XNIException
org.apache.xerces.xni.parser
XMLComponentManager
XMLComponent
XMLParserConfiguration
XMLPullParserConfiguration
XMLDocumentFilter
XMLDocumentSource
XMLDocumentScanner
XMLDTDFilter
XMLDTDSource
XMLDTDScanner
XMLDTDContentModelFilter
XMLDTDContentModelSource
XMLErrorHandler
XMLEntityResolver
XMLParseException
XMLInputSource
XMLConfigurationException
java.lang
RuntimeException
Interface
Package
Class
Extends

Parsing Pipeline

- Handlers communicate information between parser components

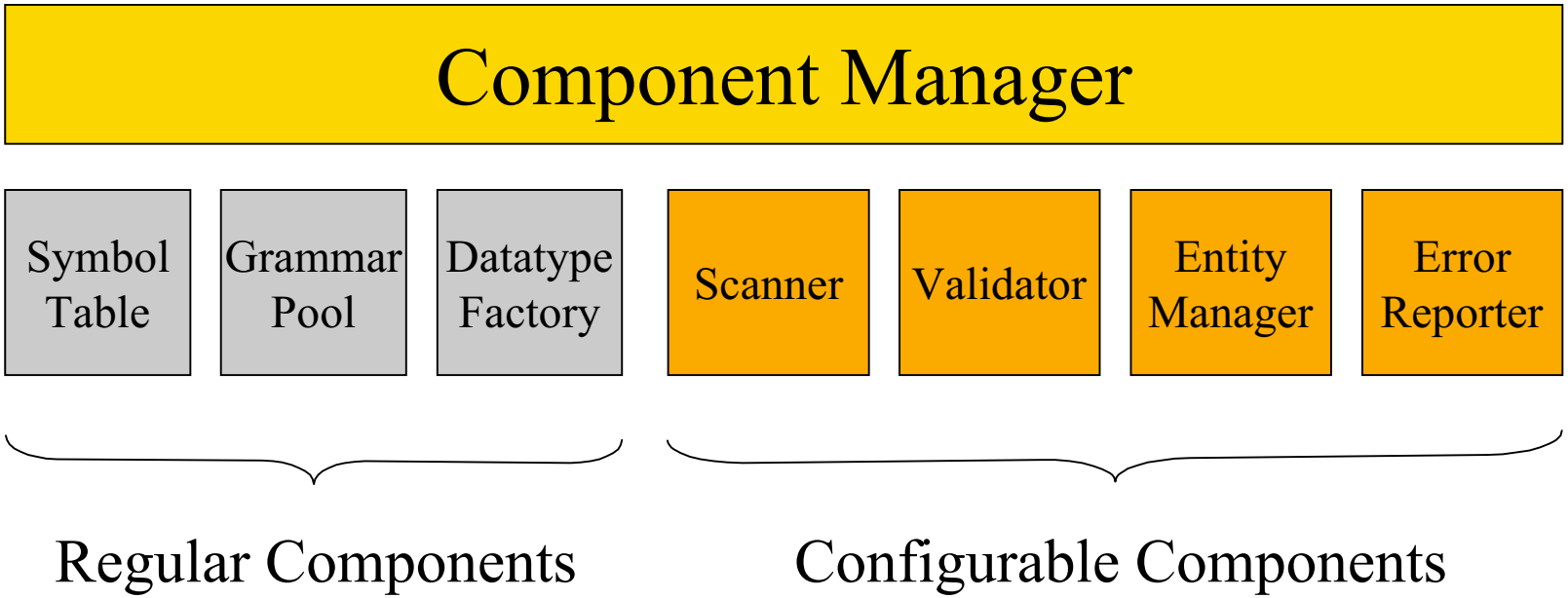


Handler Overview

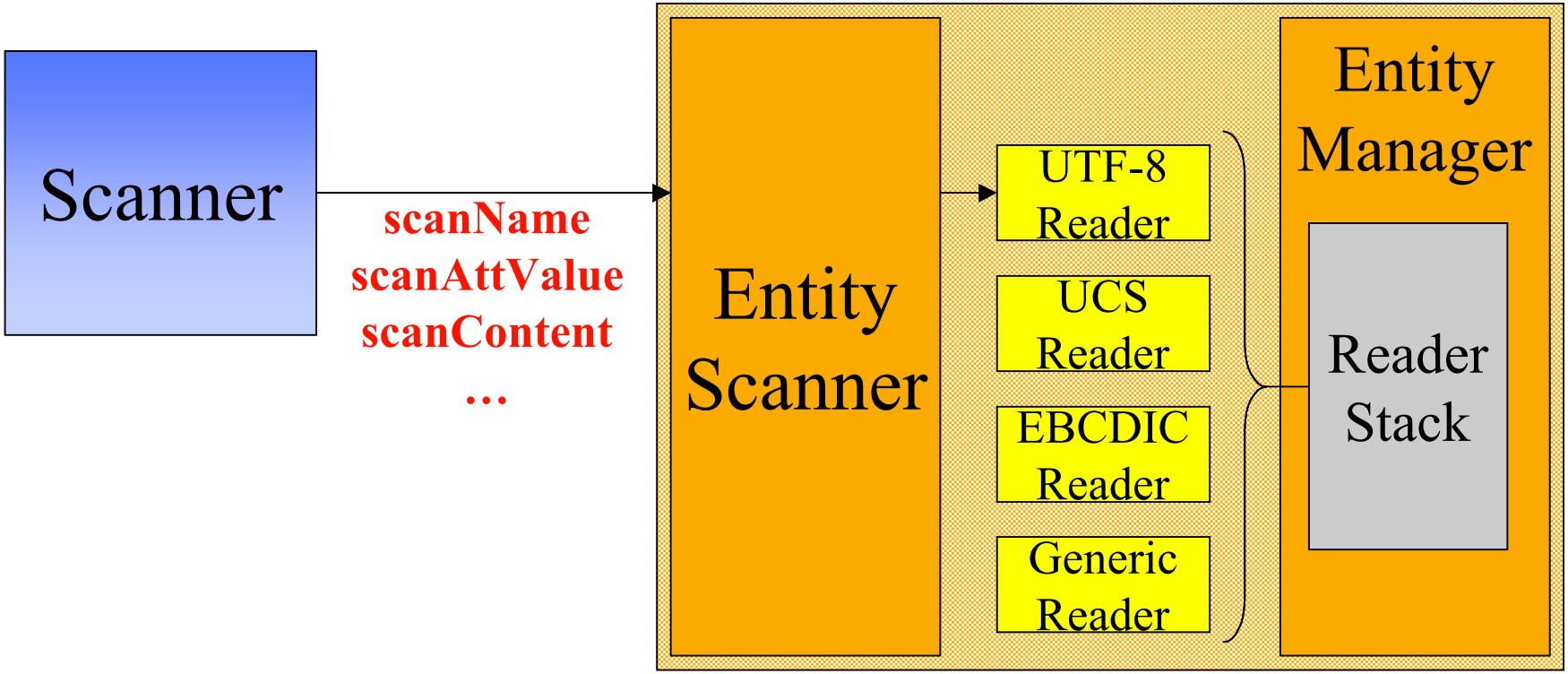


Parser Layout

- Components and Manager

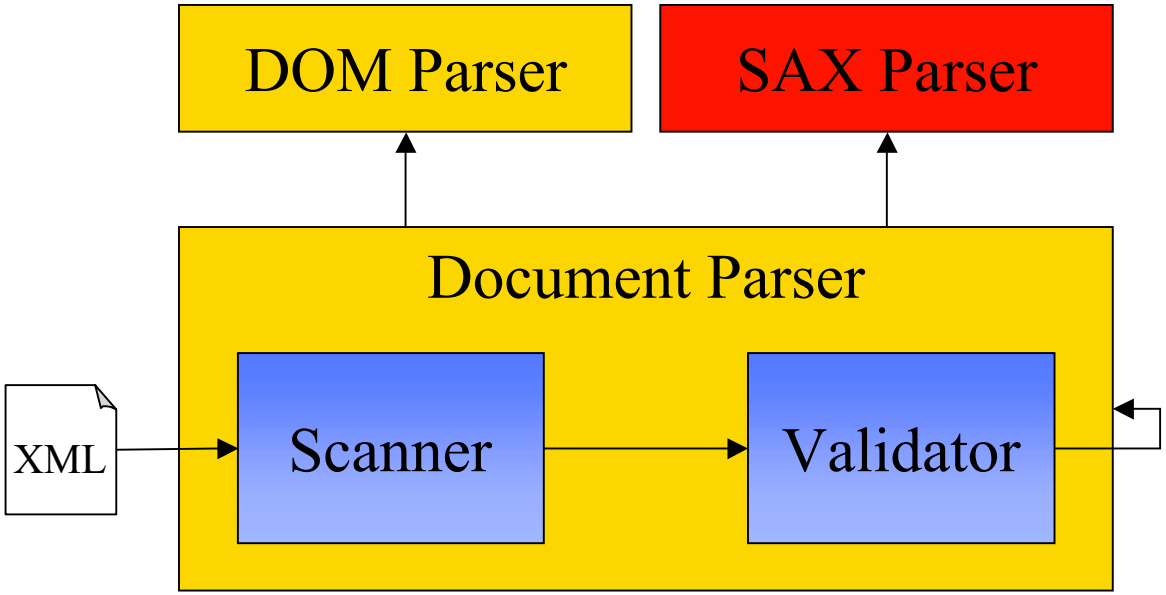


Reader Management



Parser Configuration

- Before

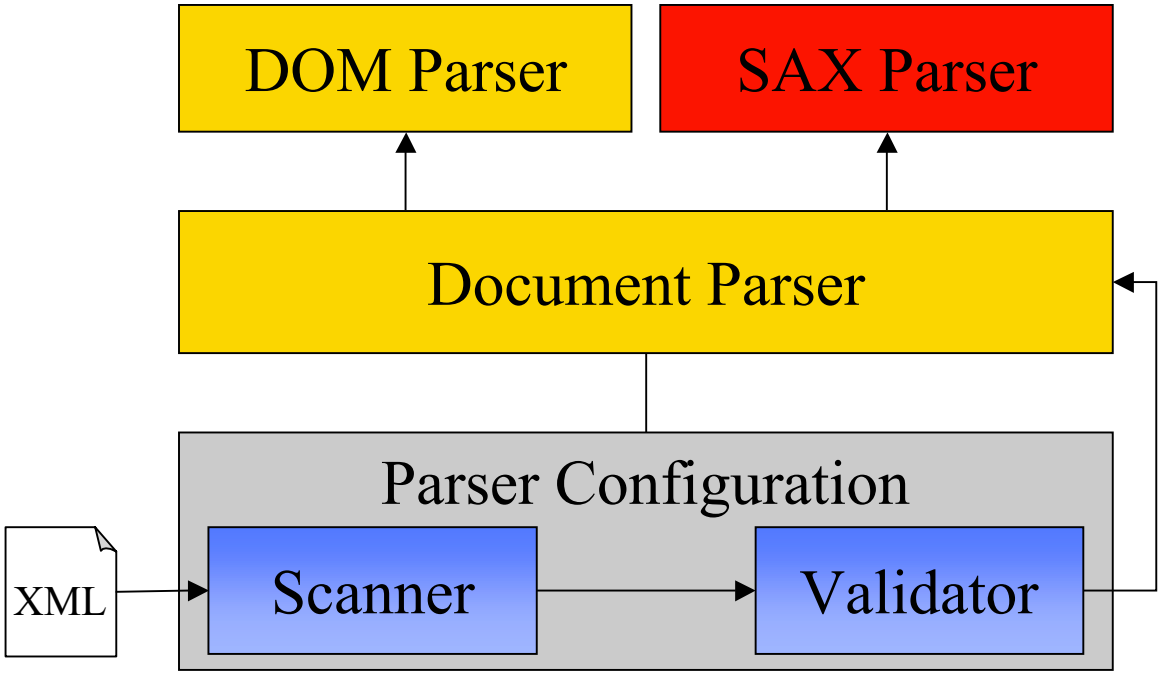


* Parser pipeline is part of the document parser base class.

* Required duplication to re-configure parser and still take advantage of API generator code.

Parser Configuration

- After

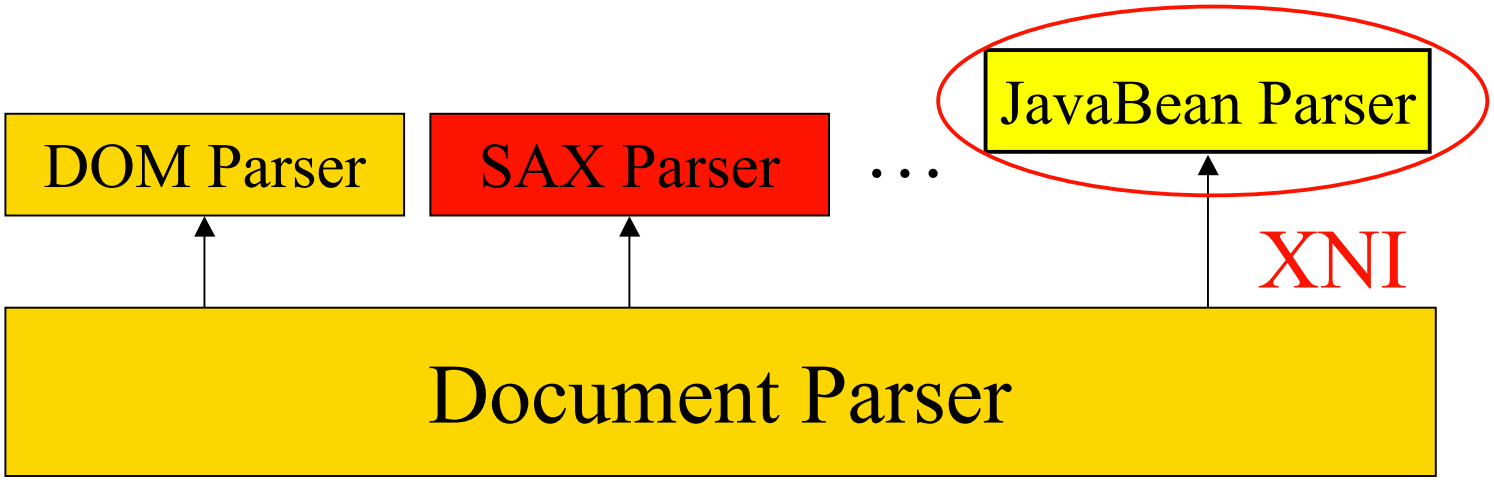


* Parser pipeline and settings are specified in a separate parser configuration object.

* Allows re-use of framework without rewriting existing code.

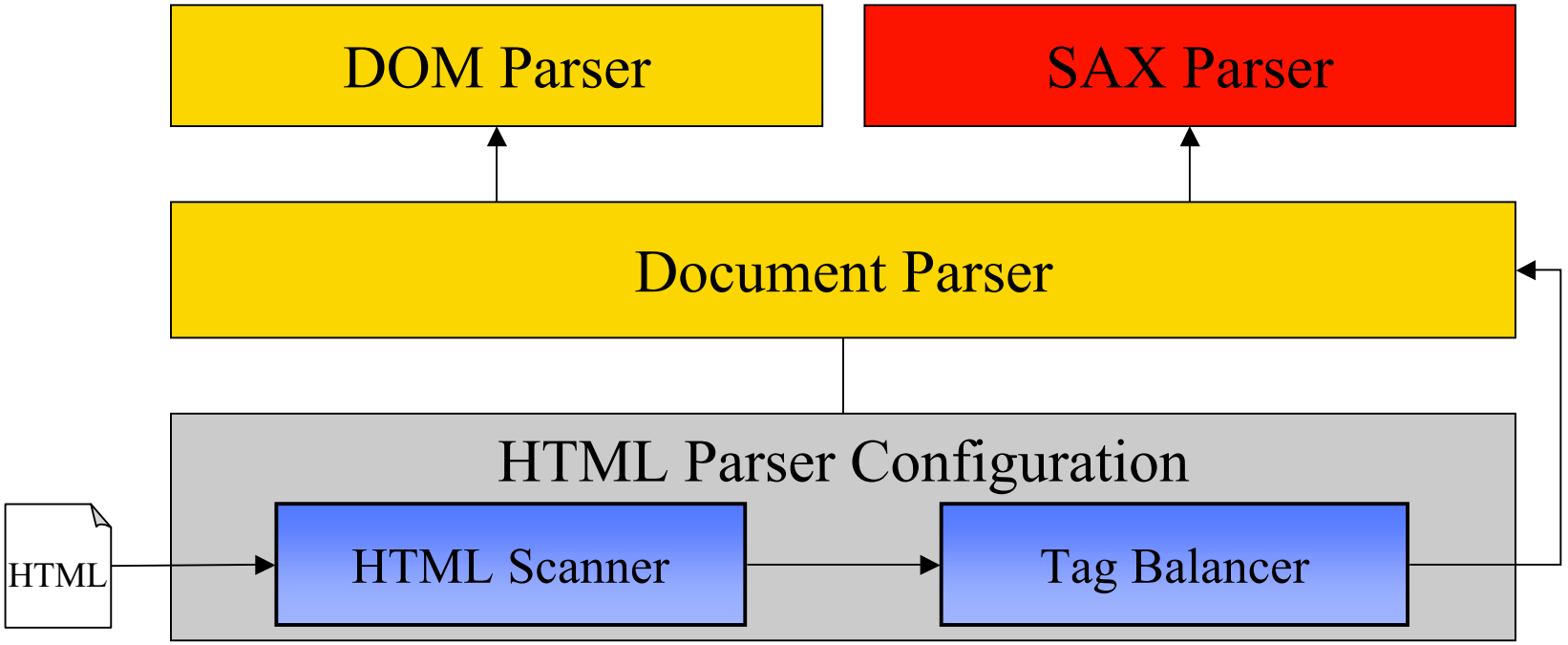
API Generators

- Different APIs can be generated from same document parser



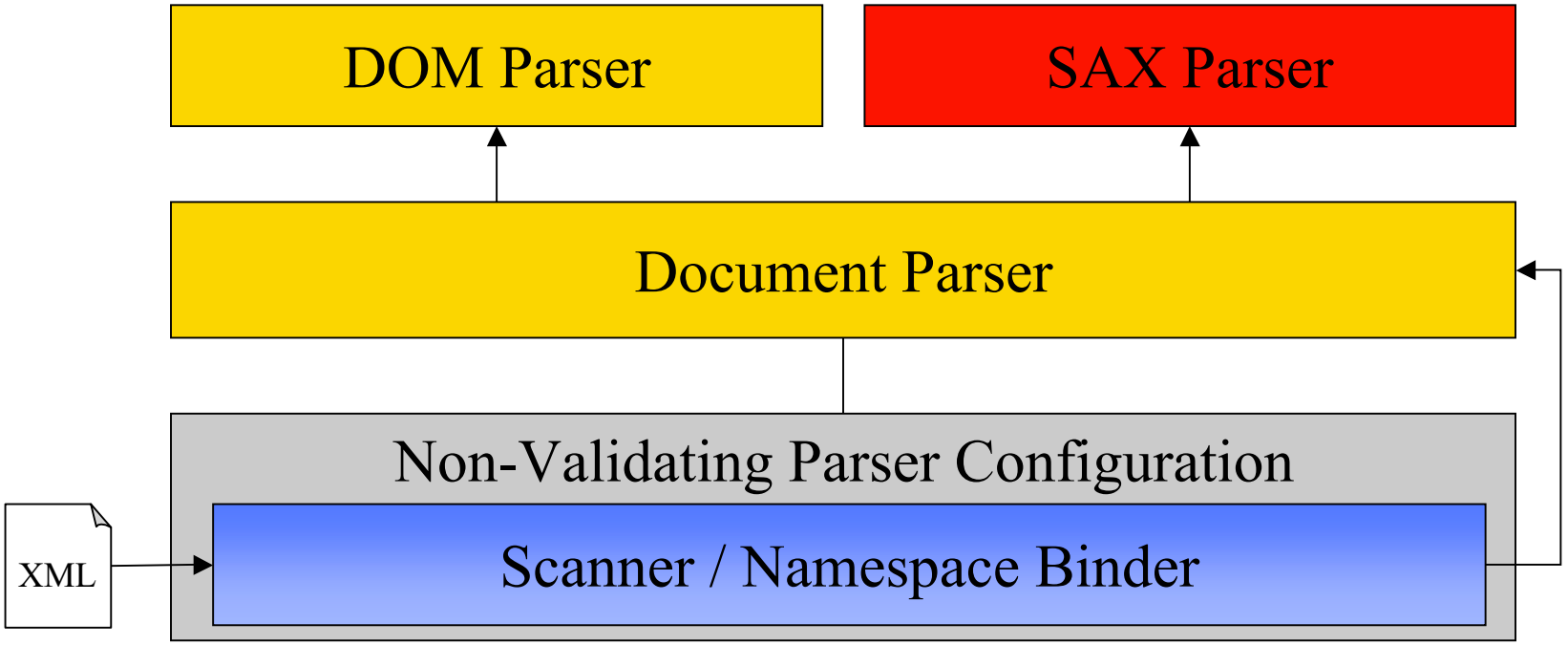
Sample Parser Configuration #1

- HTML parser
  - Available as NekoHTML download



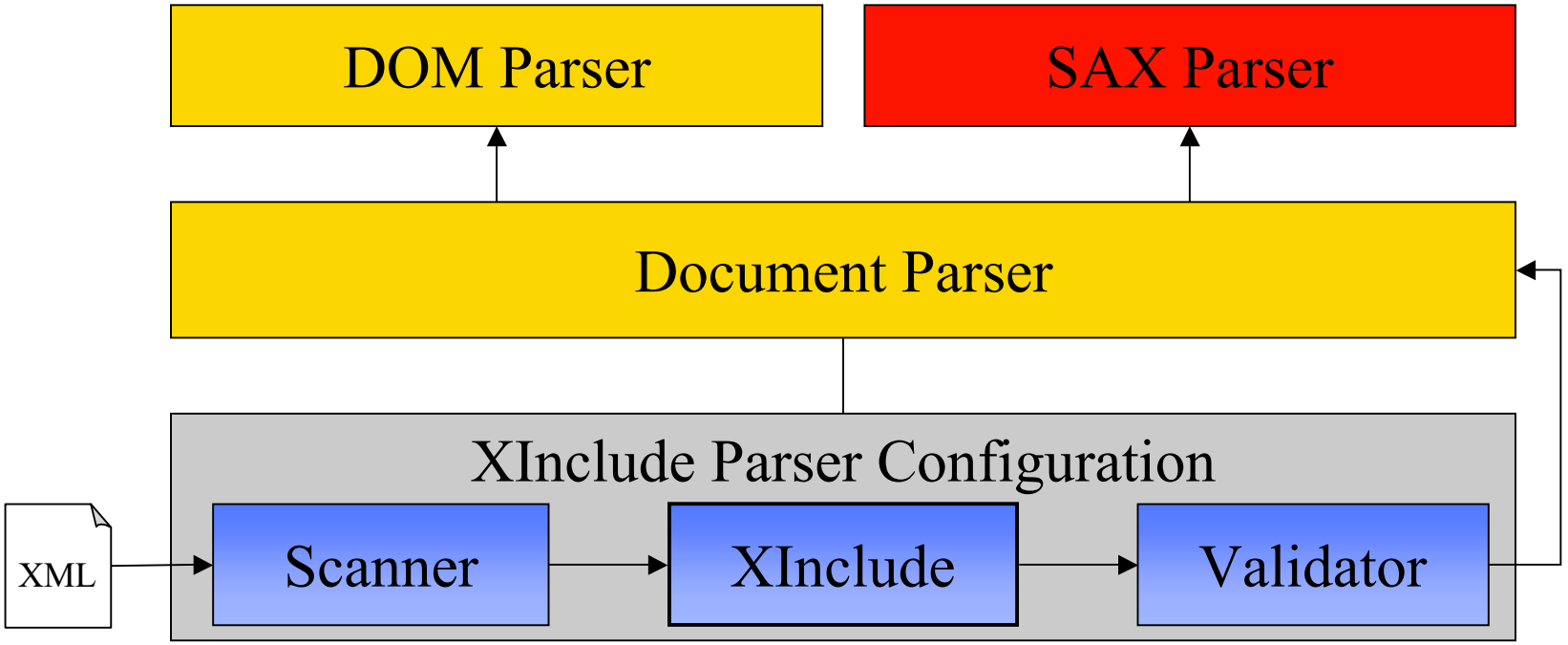
Sample Parser Configuration #2

- Non-validating parser (for performance)
  - Available with Xerces download



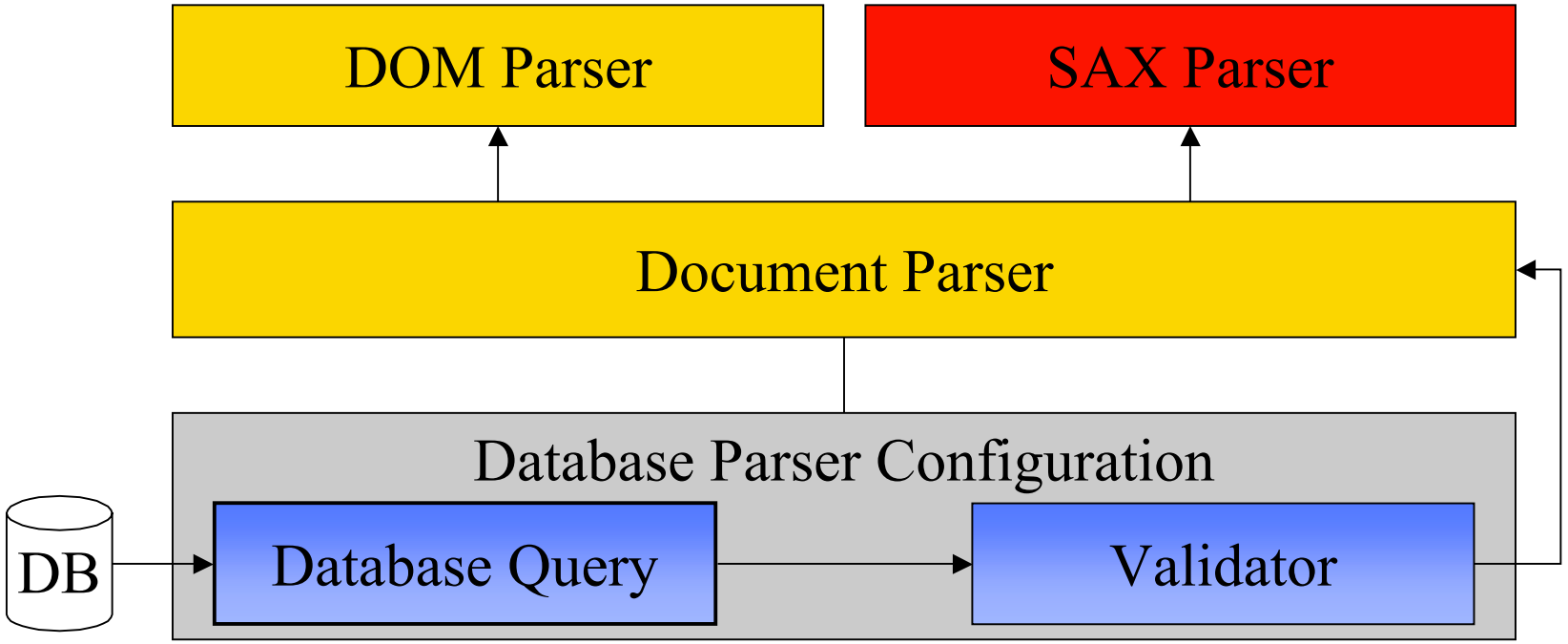
Sample Parser Configuration #3

- XInclude processing
  - Not yet implemented



Sample Parser Configuration #4

- Database result set converted to XML
  - Not yet implemented



That's All, Folks!

- Question and Answers
  - Any questions?
- Links
  - http://www.apache.org/~andyc/xml/present/
  - http://xml.apache.org/xerces2-j/
  - http://www.apache.org/~andyc/neko/