

ONERA



**xAltarica**

**Christophe Mareschal**

[mareschal@cert.fr](mailto:mareschal@cert.fr)

**ENSAE**

**10 Avenue Edouard Belin , BP 4032**

**31055 Toulouse cedex 4**

# De l'extension d'xADL à l'apport d'XML à Altarica

## Extension d'xADL2.0 à la SdF

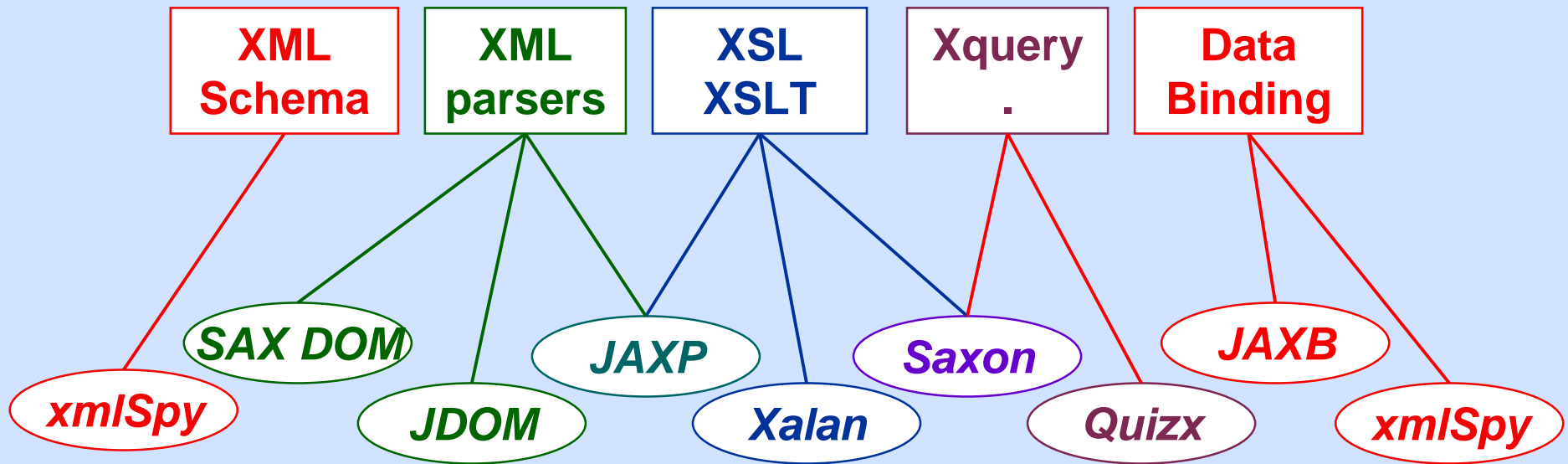
- Export de la sémantique d'altarica vers xADL2.0
- **Spécification du langage altarica en XML = xAltarica**
- Mapping xAltarica - xADL2.0

## L'apport d'XML pour le langage altarica et pour ses outils support

- Process XML mis en œuvre
- Valeur ajoutée
  - Extensibilité
  - Interopérabilité

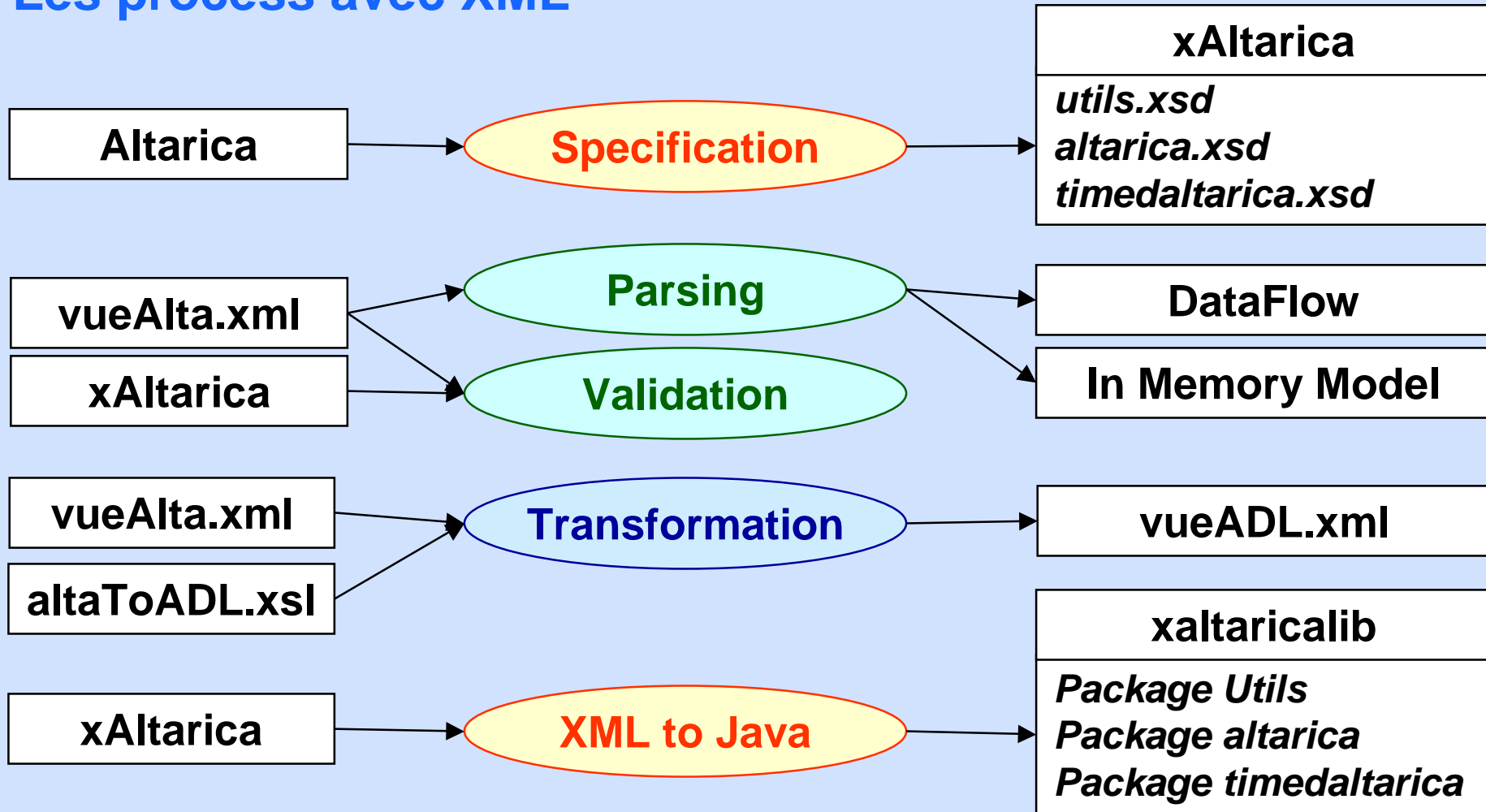
# Le monde XML

Des techniques normalisées et répandues ...



... et des outils accessibles : Applications, Bibliothèques, API

# Les process avec XML



# Specification : XML Schema

**Specification**

Parsing

Validation

Transformation

XML to Java

ONERA

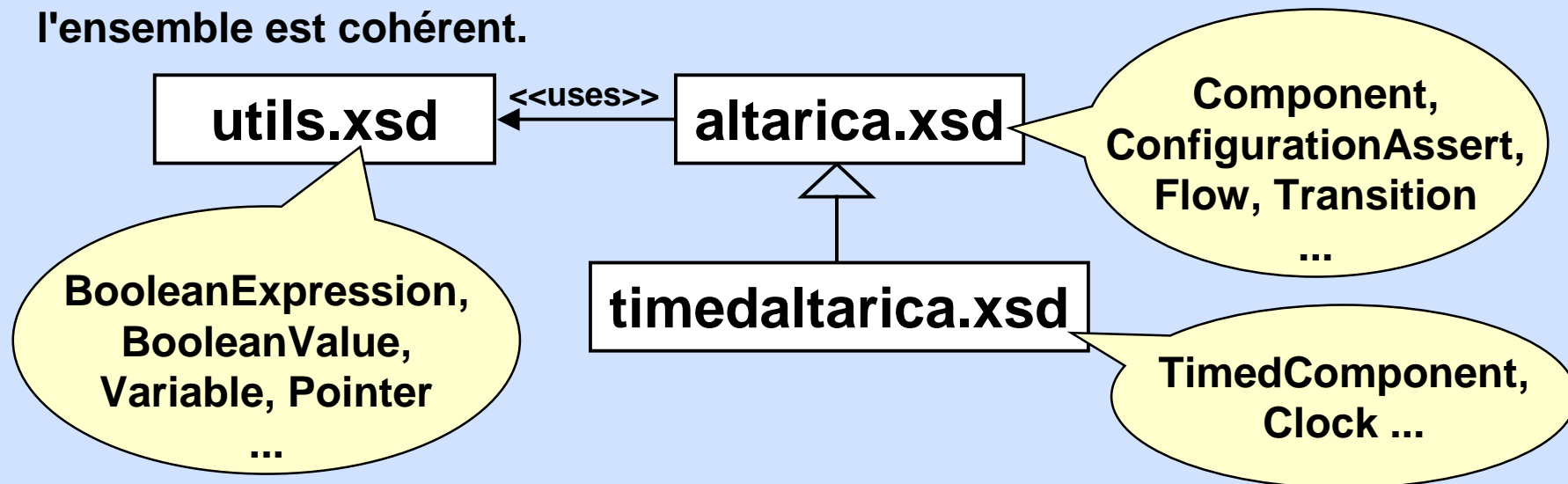
# Spécification xAltarica : XML Schema



**DTD, XML Schema : Méta langages XML**

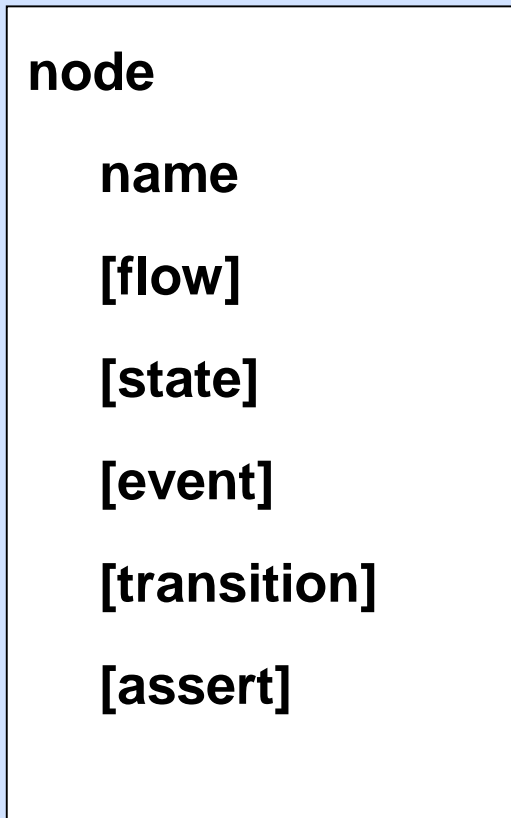
## Spécification xAltarica : XML Schema

- Le langage est spécifié par une collection de "schémas".
- Un schéma est un document xml conforme à la norme XML schema
- Chaque schéma définit une partie du langage = "namespace".
- Un schéma peut utiliser les éléments d'un autre schéma (import et include), l'ensemble est cohérent.

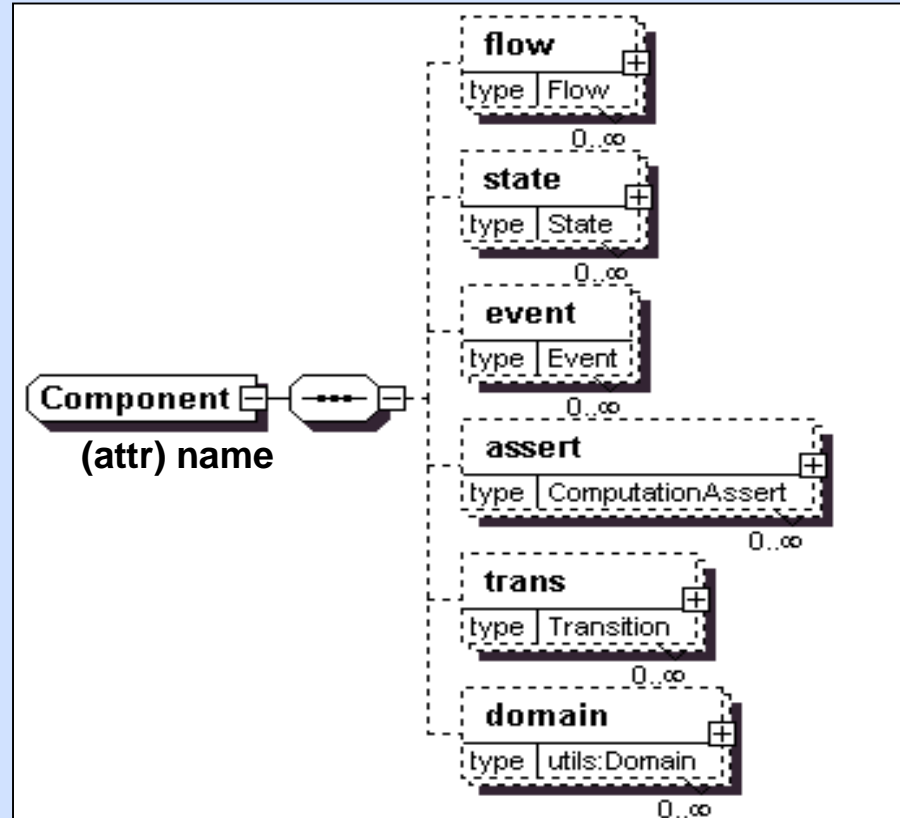


# <xsd:complexType name="Component">

*Altarica : node*



*xAltarica/altarica.xsd : Component*

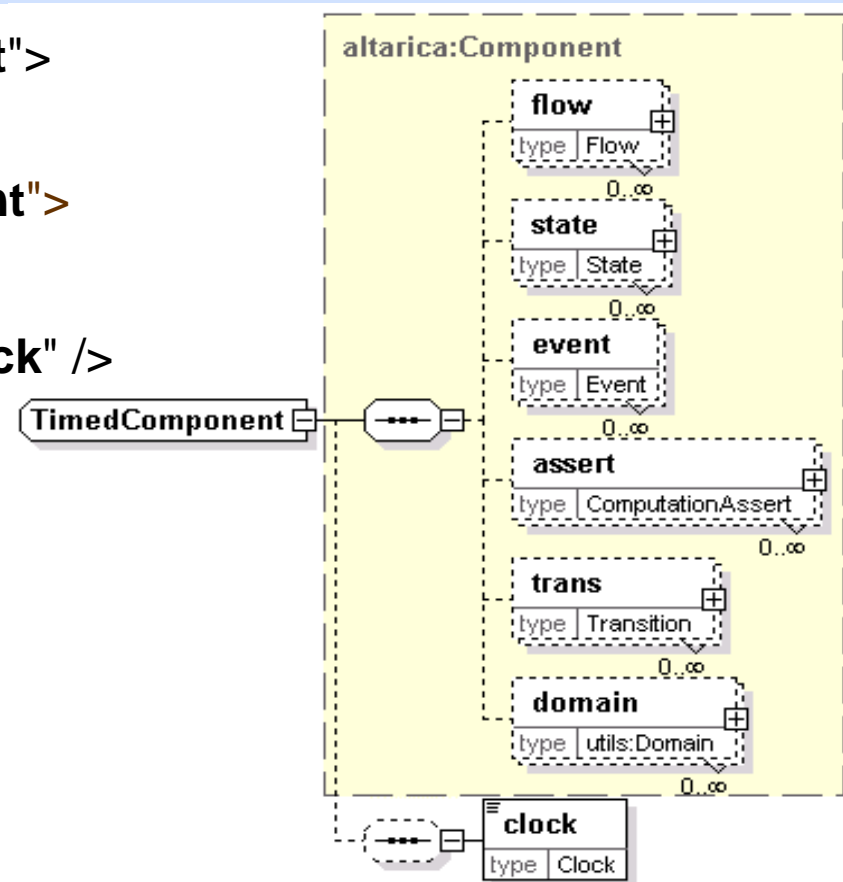




## Typage et Héritage :

`<xsd:complexType name="TimedComponent">`

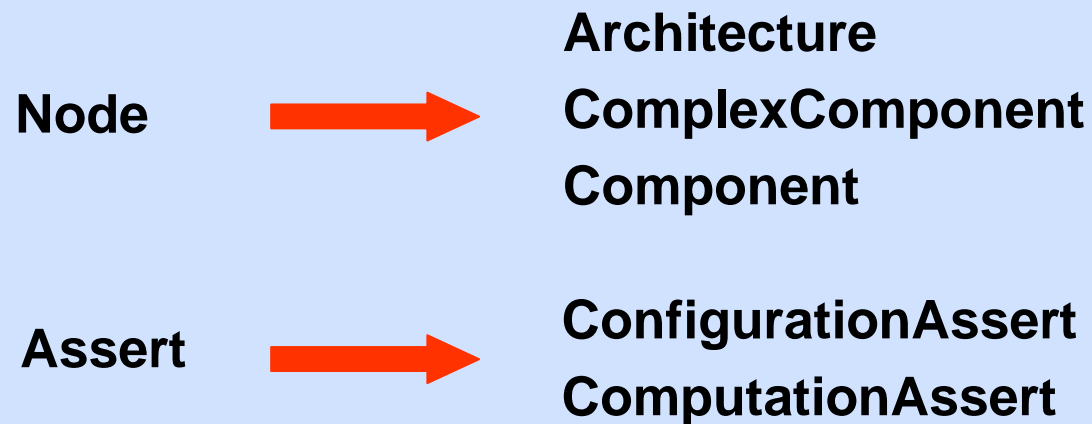
```
<xsd:complexType name="TimedComponent">
  <xsd:complexContent>
    <xsd:extension base="altarica:Component">
      <xsd:sequence minOccurs="0">
        <xsd:element name="clock" type="Clock" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



## De Altarica à xAltarica

**Ce qui ne change pas : La sémantique**

**Ce qui change peu : Le lexique et la syntaxe**



# Parsing et validation

Specification

**Parsing**

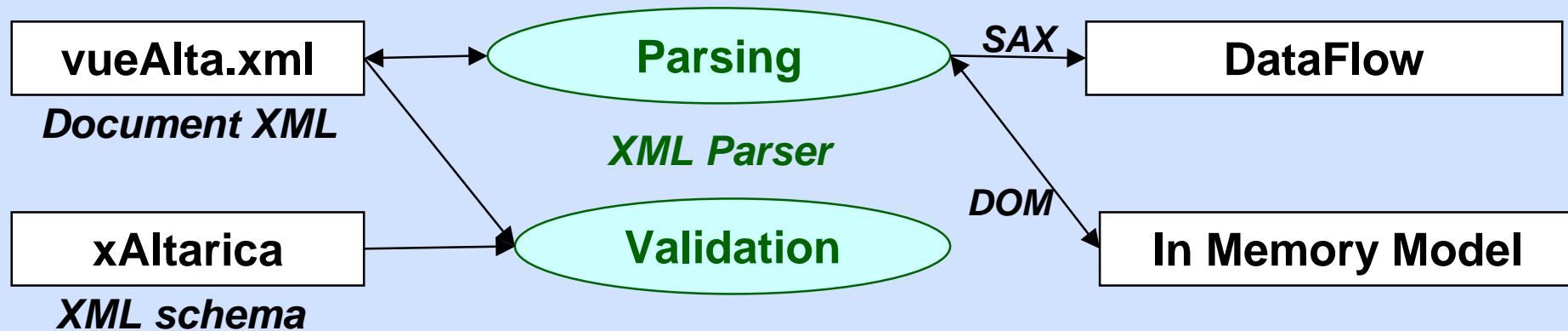
**Validation**

Transformation

XML to Java

ONERA

## Parsing et validation : SAX, DOM, JAXP, JDOM



**SAX** = "Simple API for XML" : "Read Only", lecture séquentielle

**DOM** = "Document Object Model" : "Read-Write", arbre syntaxique

## Parsing et validation : SAX, DOM, JAXP, JDOM

### Les parseurs communs sont génériques

- *Lecture et écriture de documents xml*
- *Interface entre le document xml et l'application*
- *Contrôle de conformité lexicale et syntaxique*

**Quel que soit le langage, quelles que soient ses évolutions**

**Avec XML Schema, la validation gagne en puissance**

**JAXP = "Java API for XML Parsing" (Sun) : SAX & DOM**

# Transformation : XSL

Specification

Parsing

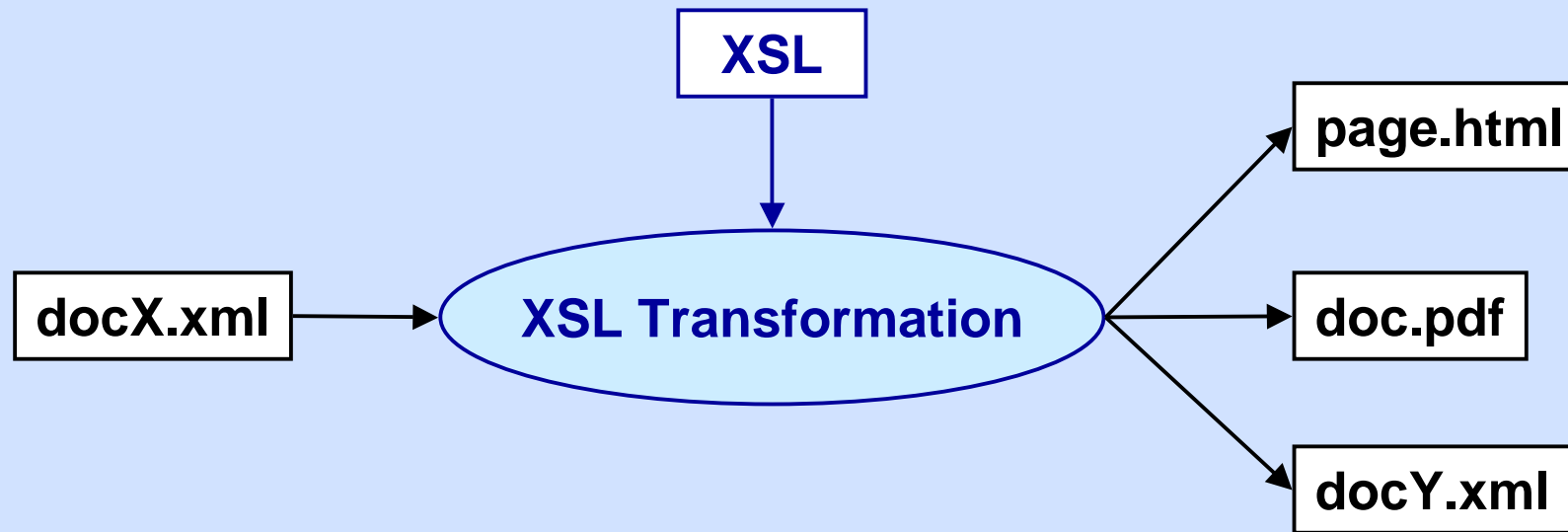
Validation

**Transformation**

XML to Java

ONERA

## Transformation : XSL, XSLT

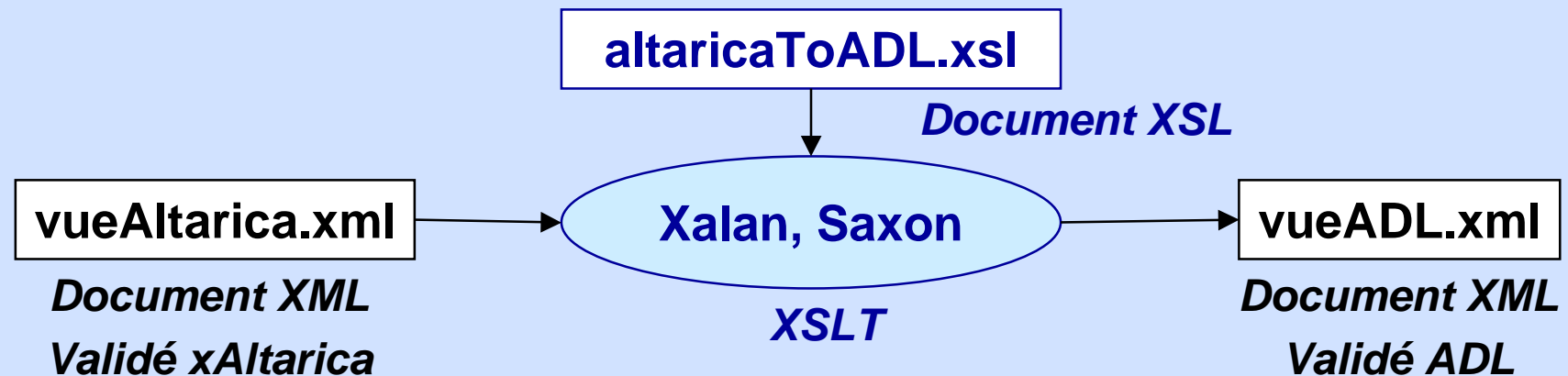


**XSL = eXtensible Stylesheet Language**

**XSLT = XSL Transformation**

## Transformation : XSL, XSLT

Le document XSL contient les règles de transformation

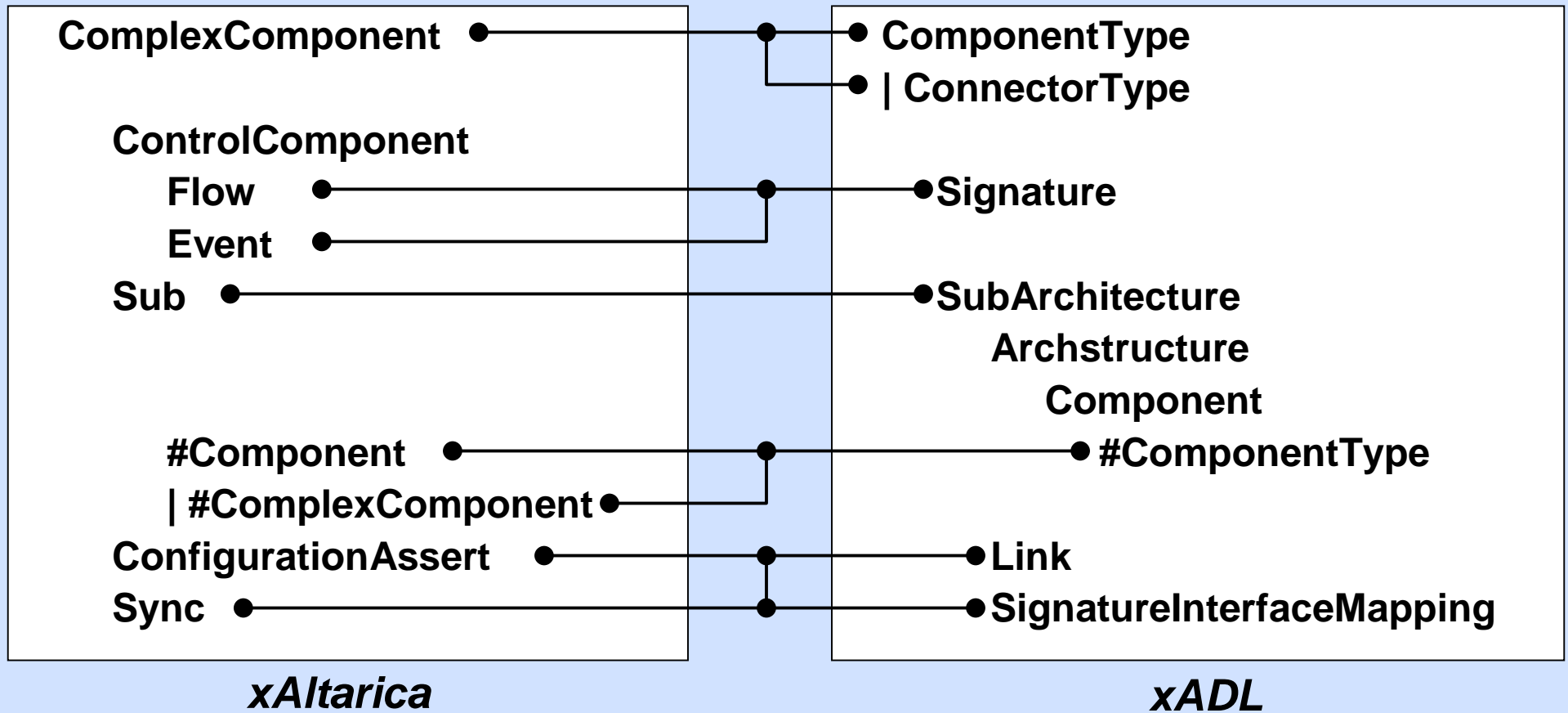


**Xalan : Processeur XSLT java (inclus dans JAXP)**

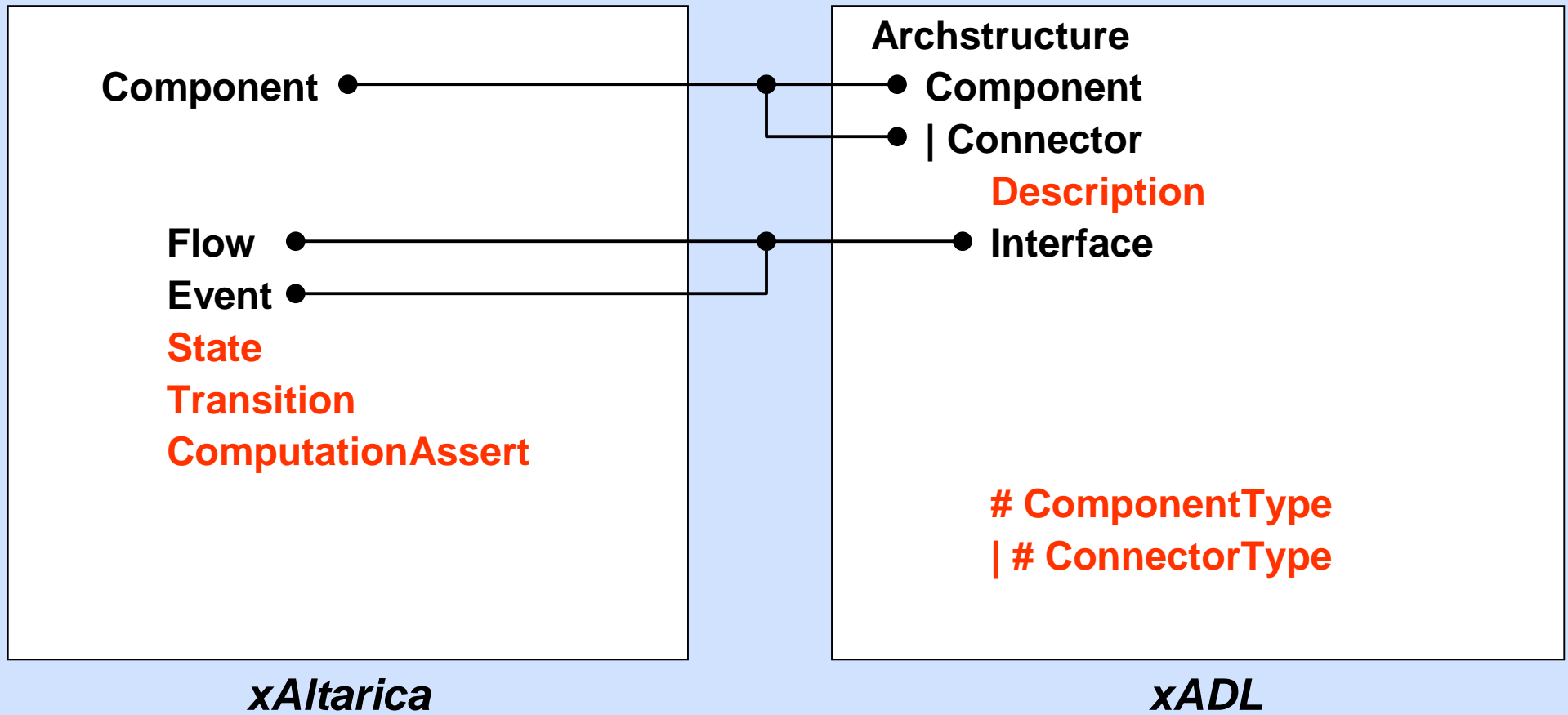
**Saxon : un autre processeur XSLT**



# Mapping : xAltaRica $\Leftrightarrow$ xADL



# Mapping : xAltaRica $\Leftrightarrow$ xADL



# XML Java Data Binding

Specification

Parsing

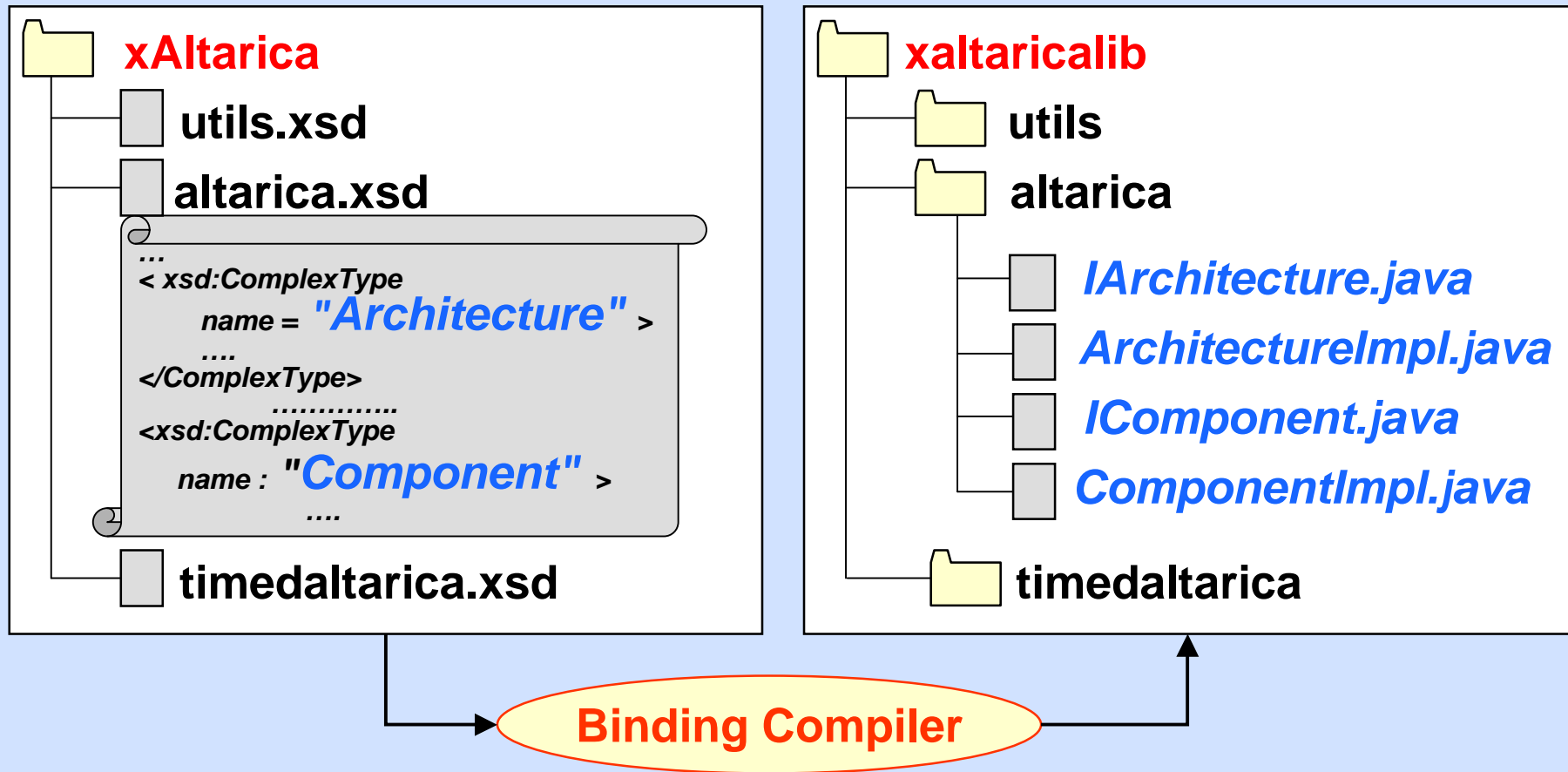
Validation

Transformation

**XML to Java**

ONERA

# XML JAVA Data Binding



## XML JAVA Data Binding : JAXB

getParentNode()  
getChilidsNode()  
createElement()  
getAttribute(String name)  
setAttributeNode(String name, String Value)  
getElementByTagName(String name)  
....

} *DOM*

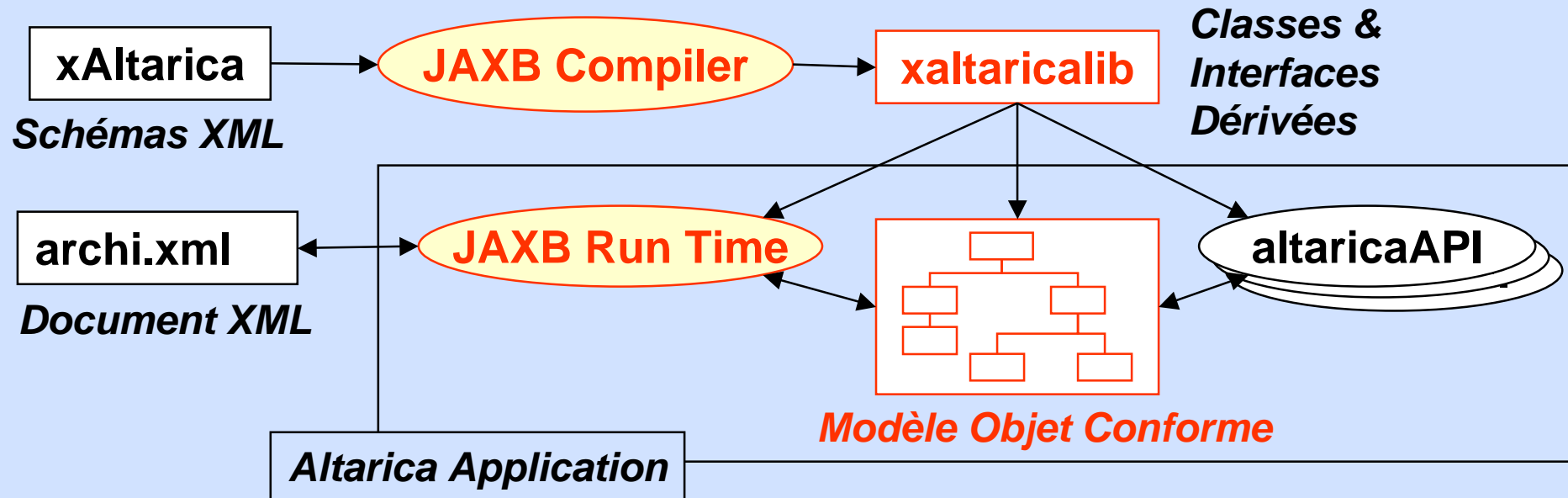
*xaltarica + JAXB*

Component.getFlows()  
Component.getStates()  
Flow.setName(String name)  
new Component  
...

**JAXB = "Java API for XML Data Binding " (Sun)**



# XML JAVA Data Binding : JAXB



**JAXB Compiler = Automatisation du Data Binding**  
**JAXB Run Time = Parsing, validation**

# Valeur Ajoutée

## Evolutivité & Interopérabilité

Specification

Parsing

Validation

Transformation

XML to Java

ONERA

## Evolutivité

- Extensions XML schema modulaires
- Non régressives sur le reste du langage
- Sans effet sur les outils xml génériques
- Adaptation DataBindingLibrary par simple recompilation
- Adaptation API "*Syntax directed*" : de facto

=> un langage extensible

=> des outils adaptables et pérennes



# Interopérabilité

- Utilisation modulaire de plusieurs schémas
- Transformation XSL : xml vers xml ou autre
- DataBindingLibrary + API spécifiques
- xQuery : A voir

=> des usages multiples et adaptés

=> des échanges facilités

# Synthèse

- **Spécification d'altairica au format xml (XML Schema)**
- **Traitements XML**
  - Parsing, Validation Syntaxique, Transformation, Accès aux données ...
- **Intégration XML**
  - API génériques (DOM, XSLT, JAXP)
  - Couple XML-Java (JAXP) Data Binding (JAXB)
- **Bénéfice**
  - Extensibilité
  - Interopérabilité

## Perspectives

- Développer la spécification : versions et variantes d' xAltarica
- Adapter et développer des outils compatibles xml/xAltarica
- Etudier le mapping et l'interopérabilité avec xADL (et +)

*A Suivre ...*



<http://www.cert.fr/francais/deri/mareschal/altarica/xaltarica.html>

