Rules Fest 2010

International Conference on Reasoning Technologies

October 11-14 • San Jose, CA USA

Grindwork Corporation
Intelligent Automation

IBM

JBoss by Red Hat

morris technical solutions

VisionArts communications
strategic thinking • creative action

3 Good Reasons to Adopt the W3C Rule Interchange Format (RIF)

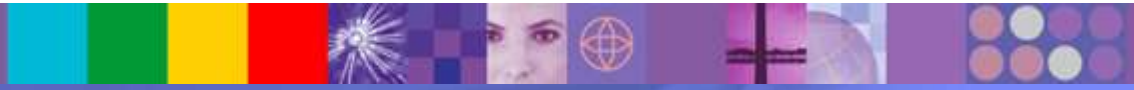
Christian de Sainte Marie

IBM (ILOG)

What is the W3C rule interchange format (RIF)?

- A standard XML serialization for rules
  - with a standard semantics
  - allowing rules written for one application to be published, shared, and re-used in other applications and other rule engines (between rule languages with compatible semantics)
  - Encouraging interoperability
  - Compatible with relevant standards (PRR, XSD, RDF, OWL, …)
- A rule is (just another) data item
  - that is produced by (or for) one application and that can be re-used in other applications
  - that can be interchanged, published, shared and executed across multiple platforms and technologies
- A W3C recommendation
  - http://www.w3.org/standards/techs/rif#w3c_all
  - An extensible framework

W3C RIF 101

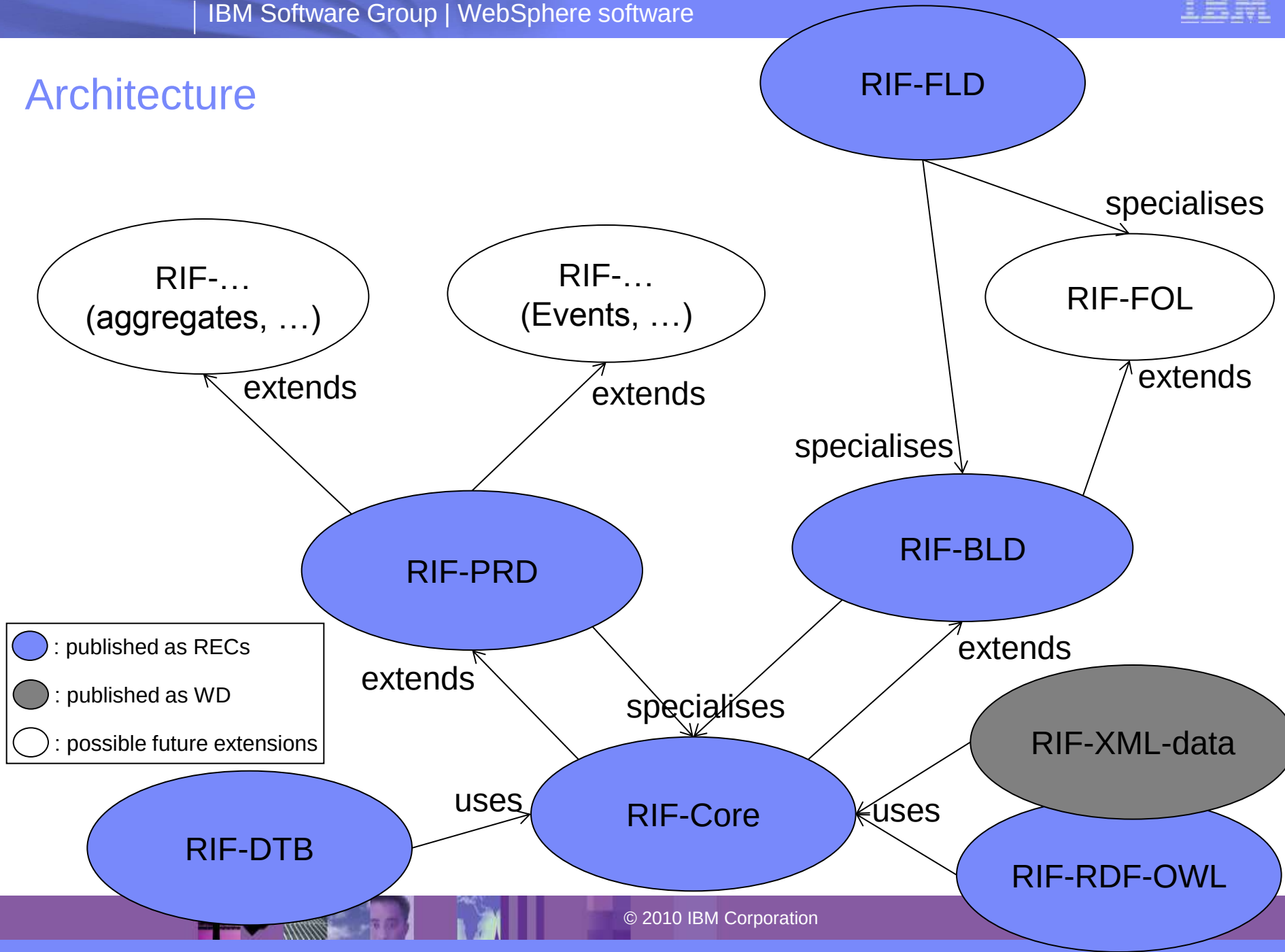


WebSphere software

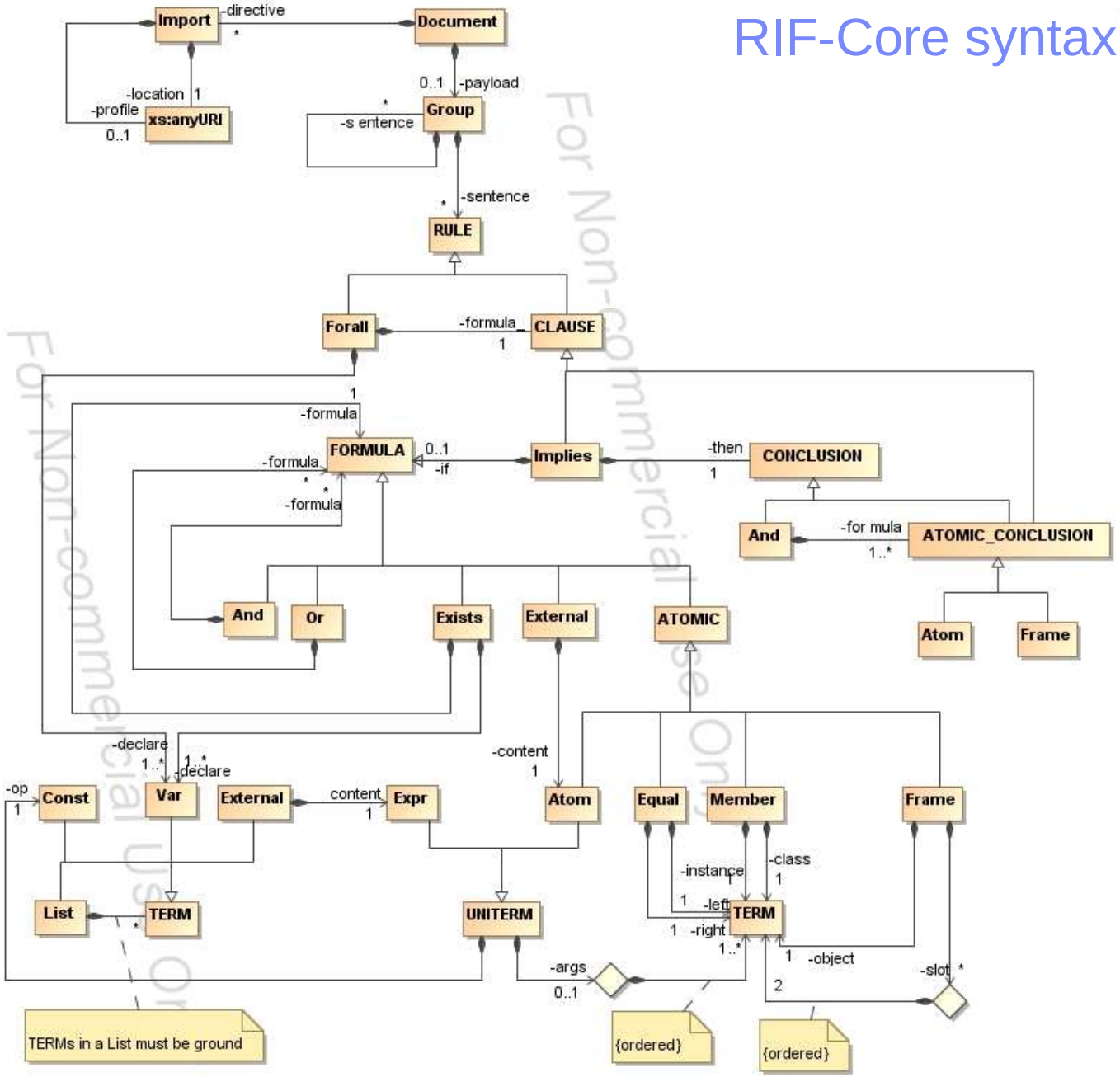
W3C RIF design principles

- Translation paradigm
  - No intrusion in covered rule languages and rule sets
- Same semantics ⇔ same syntax
  - Share constructs accross dialects wherever they agree on the semantics
  - Different constructs where semantics do not agree
- Alternating normal form XML
  - alternating <Class> and <role> tags
  - Metadata can be attached to any class element
  - Except…
- Only XML schema is normative
  - Presentation syntax for specification's readability (examples, semantics etc)

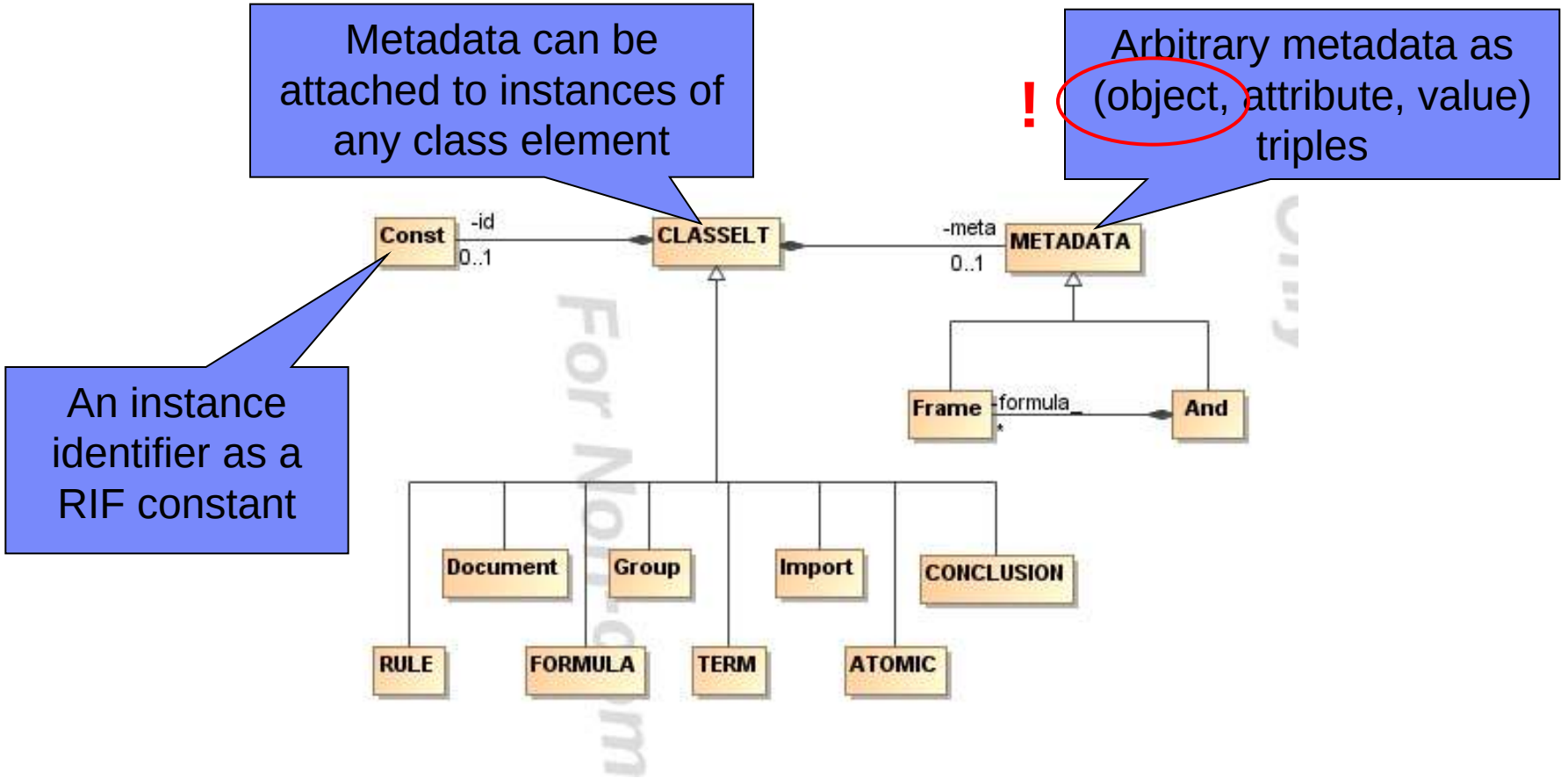
Architecture



RIF-Core syntax



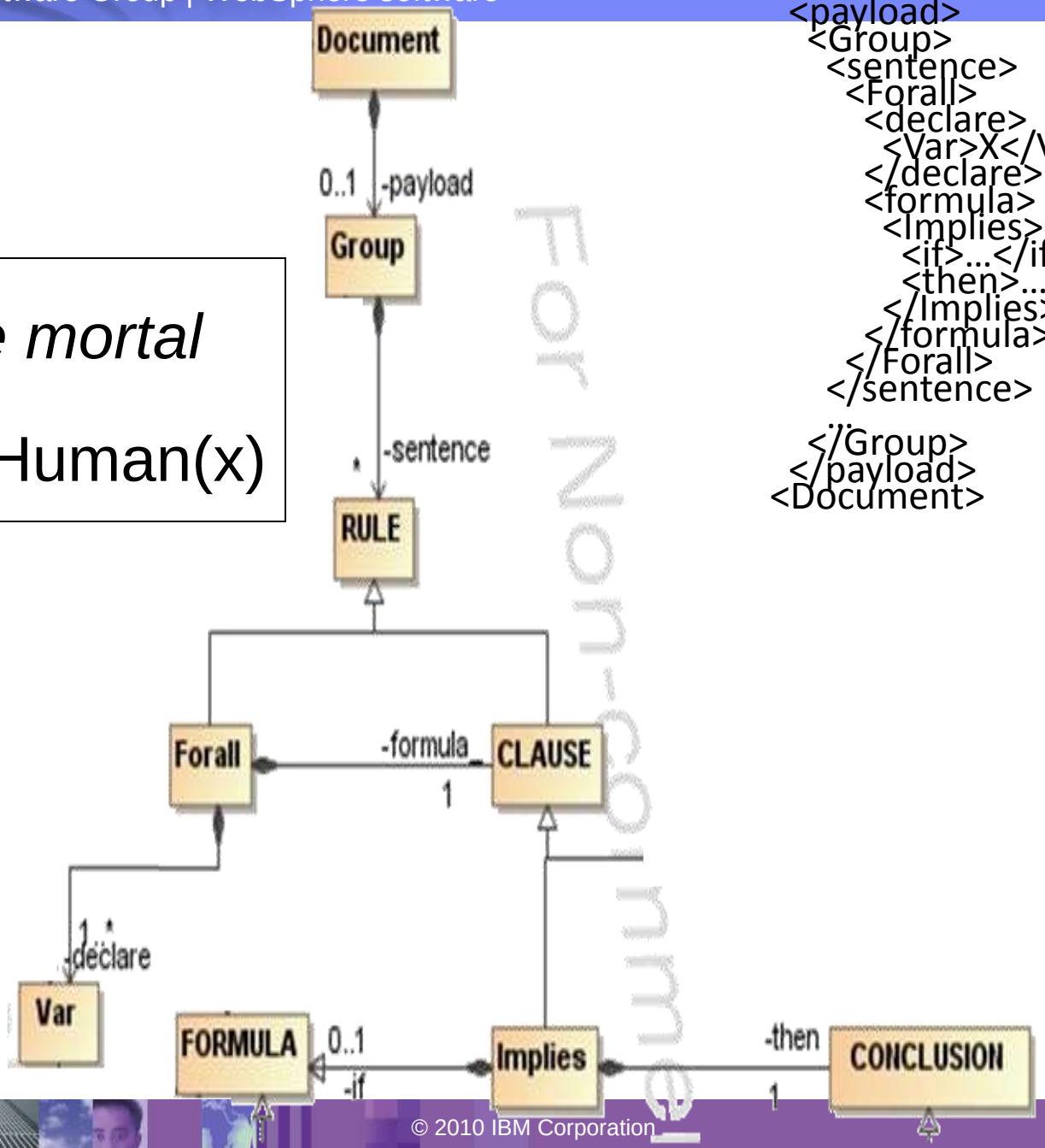
Metadata



XML syntax

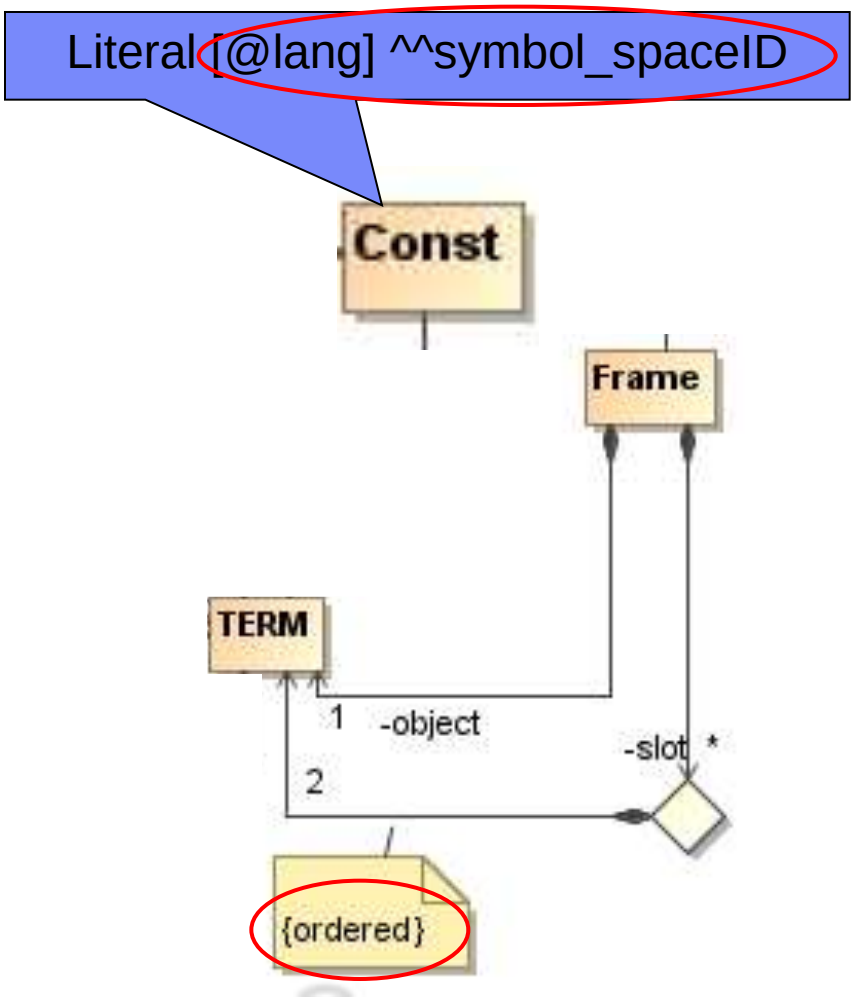
Humans are mortal

Mortal(x) :- Human(x)




```
<Document xmlns=…>
 <payload>
  <Group>
   <sentence>
    <Forall>
     <declare>
      <Var>X</Var>
     </declare>
     <formula>
      <Implies>
       <if>…</if>
       <then>…</then>
      </Implies>
     </formula>
    </Forall>
   </sentence>
   …
  </Group>
 </payload>
<Document>
```

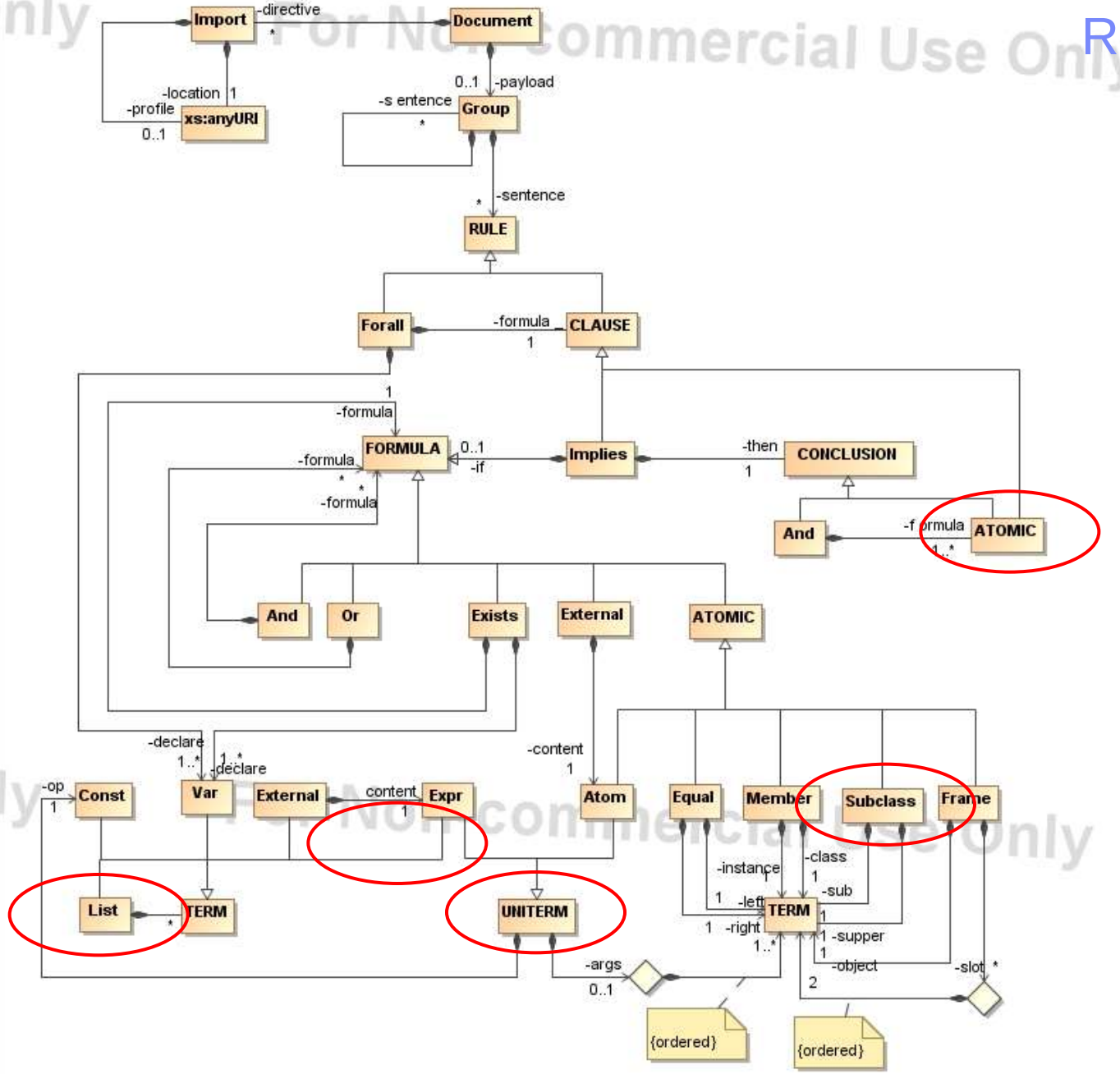
XML syntax (attributes)

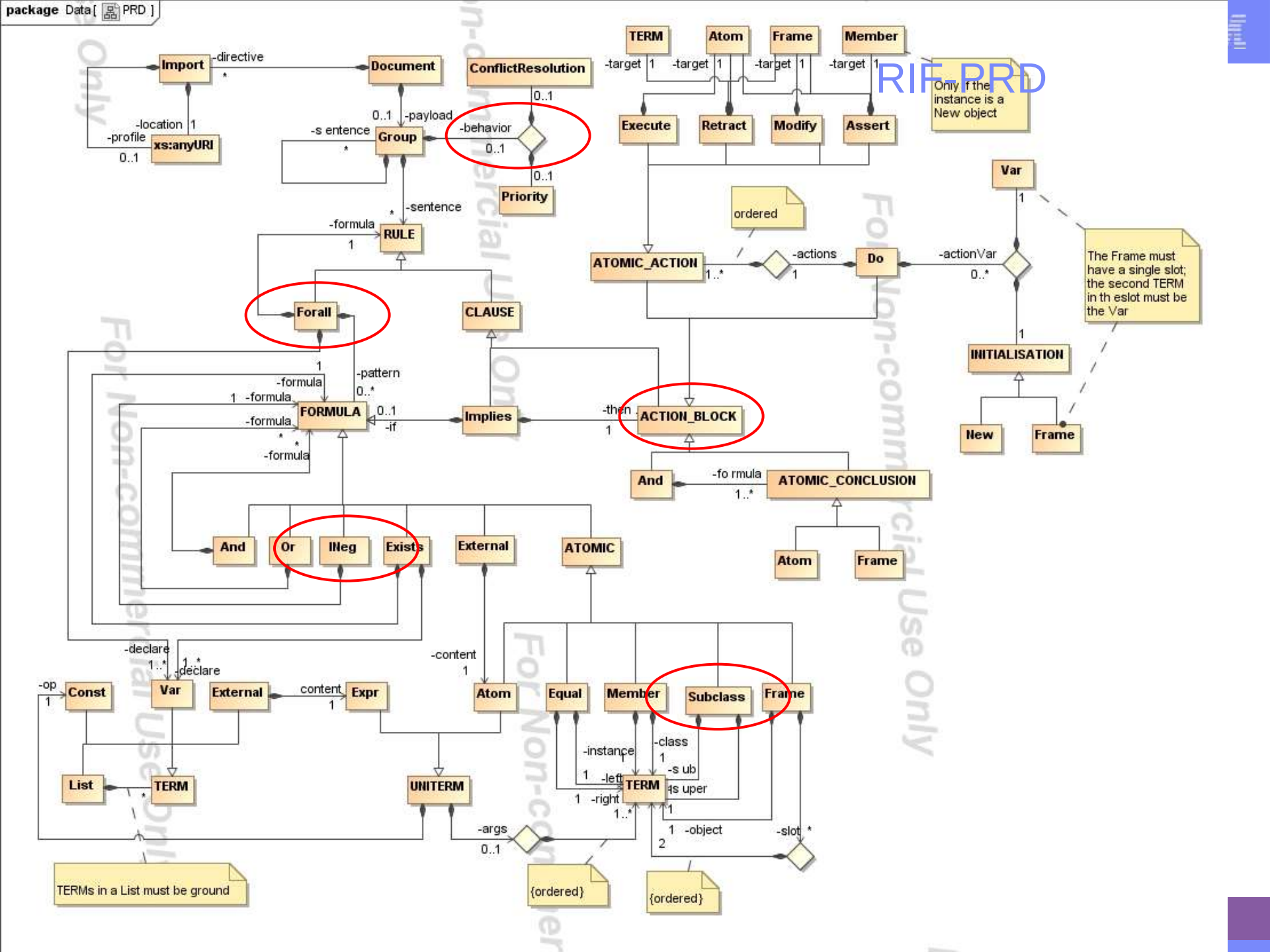


?X["human" -> ?Y]

```
<Frame>
   <object>
    <Var>?X</Var>
   </object>
   <slot ordered="yes">
    <Const type="&xs;string">
     human
    </Const>
    <Var>?Y</Var>
   </slot>
</Frame>
```

RIF-BLD




package Data [PRD]
RIF-PRD
Import
-directive
Document
ConflictResolution
-location
-profile
xs:anyURI
-s entence
Group
-payload
-behavior
Priority
-sentence
-formula
RULE
Forall
CLAUSE
-pattern
FORMULA
Implies
-then
ACTION_BLOCK
-if
And
Or
INeg
Exists
External
ATOMIC
-declare
-content
-op
Const
Var
content
Expr
Atom
Equal
Member
Subclass
Frame
List
TERM
UNITERM
-instance
-class
-s ub
-s uper
-left
-right
-args
-object
-slot
TERMs in a List must be ground
{ordered}
TERM
Atom
Frame
Member
-target
Execute
Retract
Modify
Assert
Only f the instance is a New object
ordered
ATOMIC_ACTION
-actions
Do
-actionVar
Var
The Frame must have a single slot; the second TERM in th eslot must be the Var
INITIALISATION
New
Frame
-fo rmula
ATOMIC_CONCLUSION
Atom
Frame
For Non-commercial Use Only

Syntax examples

- Look at
  - http://www.w3.org/2005/rules/wiki/Category:Test_Case
- Select test case
- Click on « view XML »

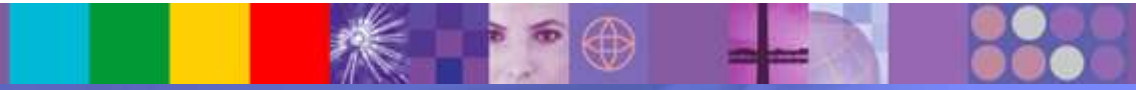
Built-in data types and externals

- All XML Schema 1.1 data types
  - Except `anyType`, `anySimpleType`, `anyAtomicType`, `normalizedString` and derived types, `NOTATION`, `precisionDecimal`, `QName`, `ENTITIES`, `IDREFS`, `NMTOKENS`
  - Plus `rdf:XMLLiteral` and `rdf:PlainLiteral`
- Symbol spaces without value spaces
  - `rif:local` and `rif:iri`
- RIF built-in functions and predicates are *external* in the sense that their semantics is defined outside of – and independently from – the rules where they are used
  - As opposed to logic functions and predicates
  - Include:
    - Data type comparison, guards, convertion and casting
    - Numeric, boolean, string, date, time and duration operators and comparison predicates
    - rdf:XMLLiteral and rdf:plainLiteral specific predicates and operators
    - Lists predicates and operators
    - Dialect-specific built-in functions
      - act:print (RIF-PRD)
- Many RIF built-in functions and predicates are adapted from *XQuery 1.0 and XPath 2.0 Functions and Operators*
  - The differences from the original XF&O include the handling of errors, the differentiation between predicates and functions, and a few other specific differences
  - If an argument value is outside of its domain, the value of the function or predicate is left unspecified

RIF semantics

- RIF-BLD: standard first-order model-theoretic semantics with functions and equality
- RIF-PRD: standard RETE-style operational semantics
  - Default standard conflict resolution strategy `rif:forwardChaining`
    - Refaction
    - Priority
    - Recency
    - Implementation dependent selection
  - Caveat: `Modify` is not an atomic action
    - It is a shortcut for `Retract+Assert`
- RIF-Core: essentially Datalog
  - Model-theoretic semantics of RIF-BLD, restricted to RIF-Core syntax
    - No subclass, no logic functions, no named argument UNITERMS
    - Only atoms and frames in the head
  - Operational semantics of RIF-PRD, restricted to RIF-Core syntax
    - No negation
    - Pattern2Condition equivalence
    - Only Asserts in the RHS: Do2And equivalence
    - With those restriction, the operational semantics of RIF-PRD reduces to Datalog inflationary semantics, equivalent to minimal Herbrand model semantics
- RIF Core and RIF-PRD require **safeness** (RIF-BLD does not)
  - All variables must be safely bound in the condition before being used in the conclusion

Implementation



WebSphere software

Disclaimer

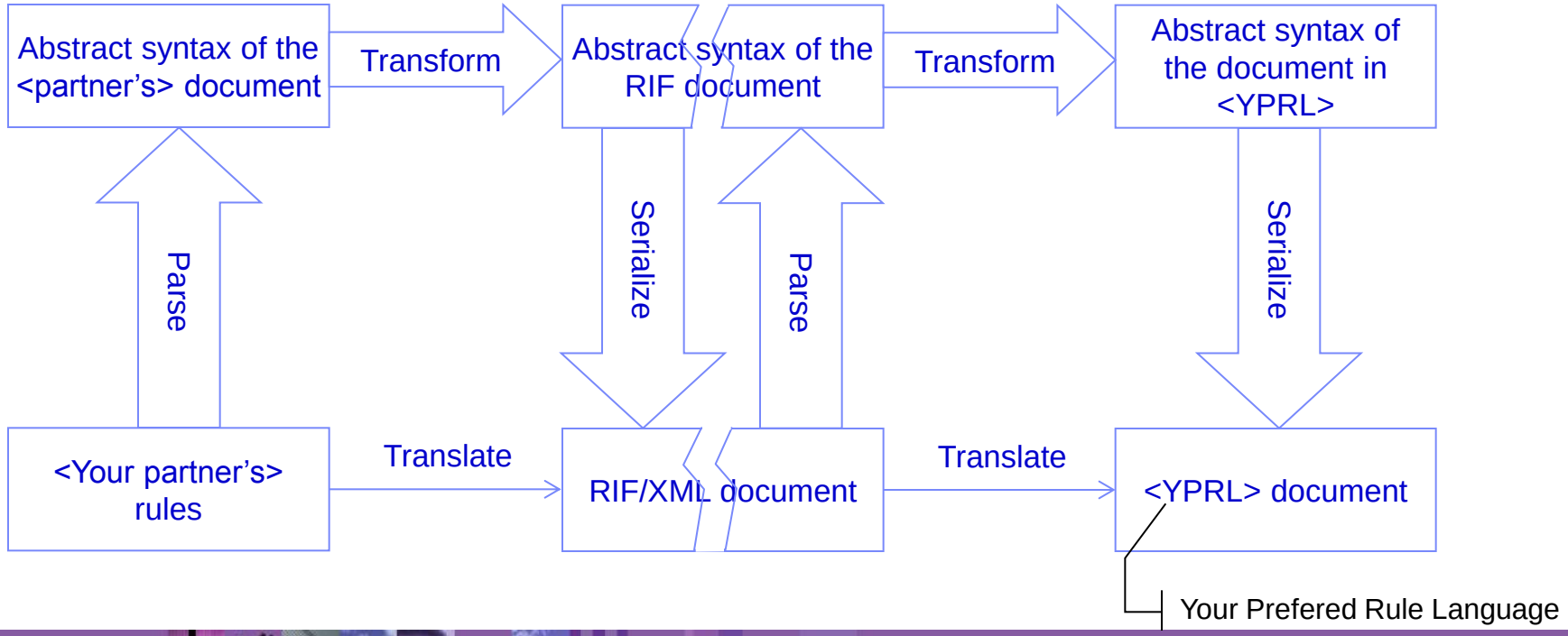
- ***Disclaimer:*** *Information regarding potential future products is intended to outline IBM's general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for IBM's products remains at IBM's sole discretion.*

Implementations

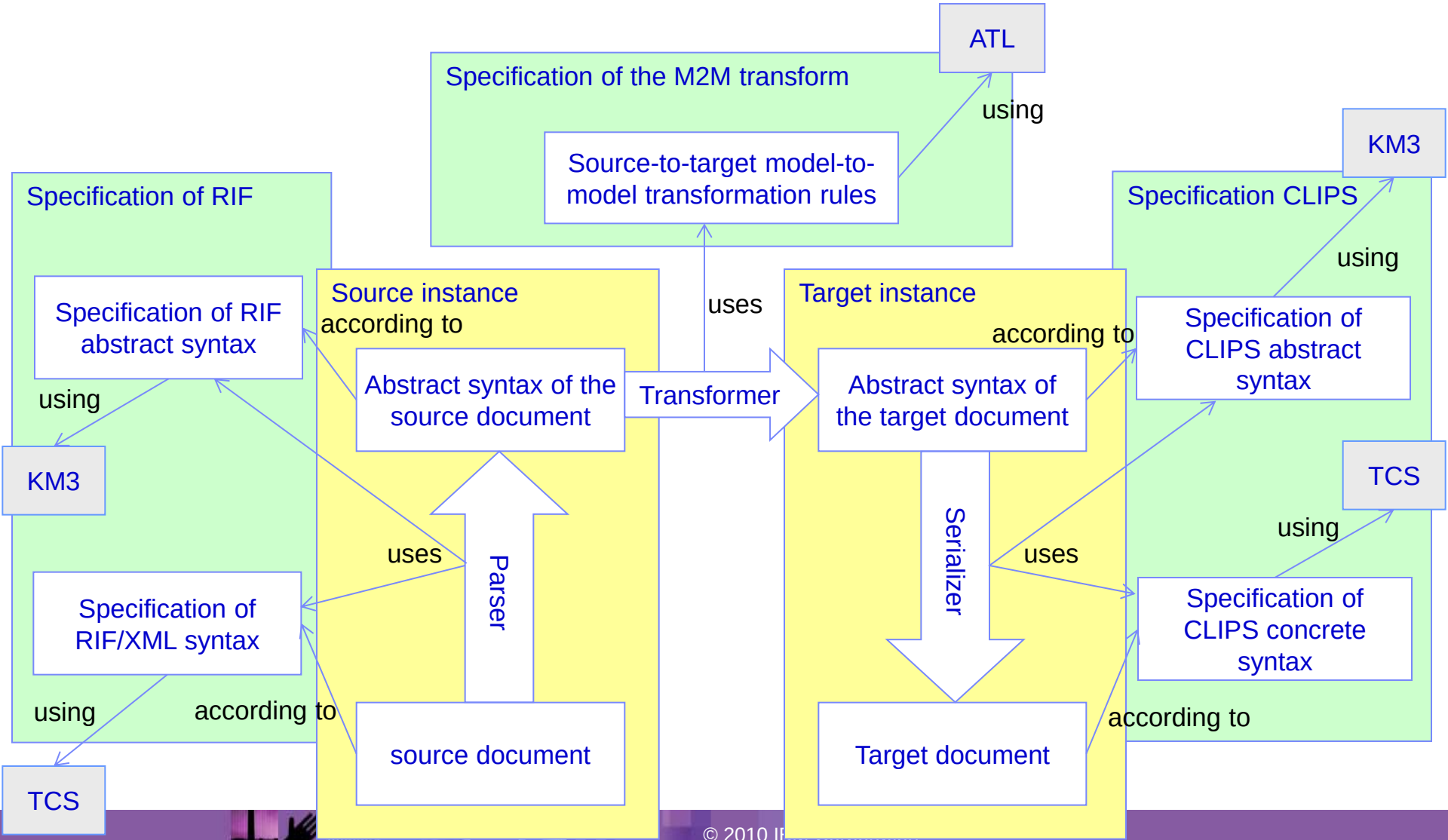
- 13 reference implementations
  - listed at http://www.w3.org/2005/rules/wiki/Implementations
  - Including open source tools such as
    - Java library of (most) RIF built-ins functions and predicates
      - http://sourceforge.net/apps/mediawiki/rifle/index.php?title=RIF_DTB
    - RIF-BLD converter to TPTP syntax
      - http://www.freewebs.com/riazanov/software.htm
- More commercial and open source implementations to come (including for IBM Websphere ILOG JRules)

Example: implementing RIF based on model to model transform

- Assumption: rule languages the same family, as covered by a RIF dialect, as likely to have very similar abstract structures, even if they have very different concrete syntaxes
- Conclusion: it should be relatively easy to do the translation at the level of the abstract, rather than the concrete, syntaxes



Example: RIF to CLIPS using KM3, TCS and ATL

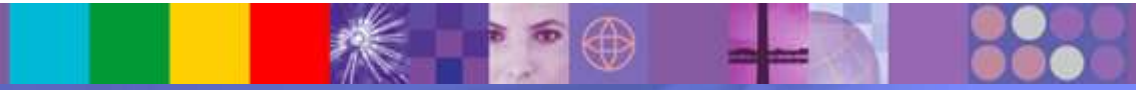


KM3, TCS and ATL

- TCS, KM3 and ATL are Eclipse component
  - TCS (Textual Concrete Syntax) enables the specification of textual concrete syntaxes for Domain-Specific Languages (DSLs)
    - generates parsers (text-to-model) and serializers (model-to-text) for DSLs
    - provides an Eclipse editor with syntax highlighting, outline, etc
    - http://www.eclipse.org/gmt/tcs/
  - KM3 (Kernel Meta Meta Model) is an implementation-independent language to define abstract syntaxes of DSLs
    - provides an Eclipse editor with syntax highlighting, outline, etc
    - http://wiki.eclipse.org/KM3
  - ATL (ATLAS Transformation Language) is a model transformation language and toolkit
    - the ATL Integrated Environnement (IDE) provides a number of standard development tools (syntax highlighting, debugger, etc.)
    - http://www.eclipse.org/m2m/atl/
- The TCS specifications of RIF-PRD/XML and CLIPS textual concrete syntaxes, the KM3 definitions of RIF-PRD and CLIPS abstract syntaxes, and the ATL specifications of the RIF-PRD to CLIPS and CLIPS to RIF-PRD transforms are freely available
  - @@@URL
  - Notice that the RIF-PRD TCS and KM3 specifications can be re-used to specify transforms to and from other rule languages

Demonstration

Use cases and usage issues



WebSphere software

Rules standards are good for you

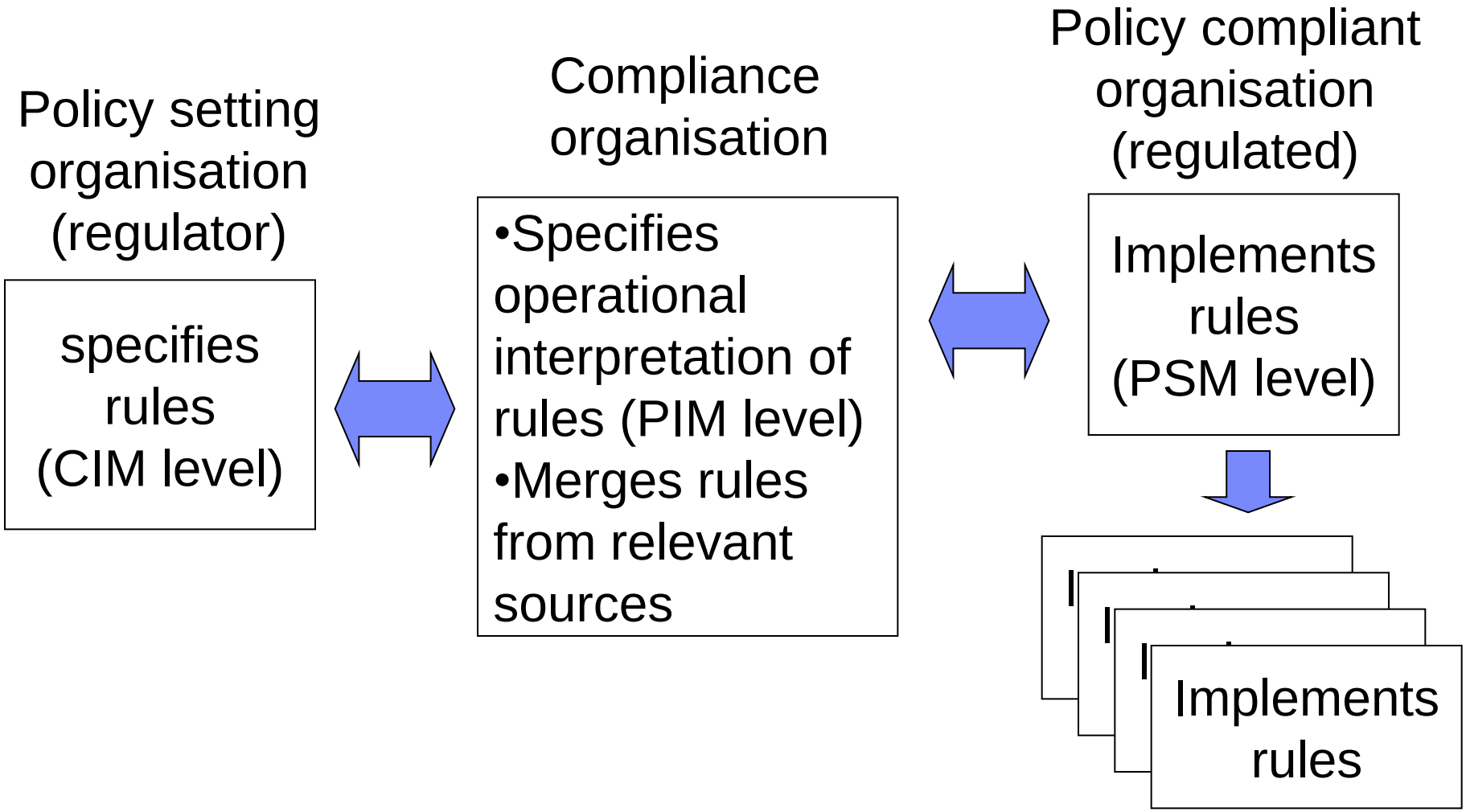
- Standardisation: an agreement to support a common interface or representation (with cooperation of the vendors)
  - Standard semantics
  - Platform neutral persistence
  - Use across different platforms
- Users benefits
  - Avoid vendor's lock-in
    - Commoditization
  - Allows cooperative development/deployment
    - Share rules/policies/documentation
  - Improves communication
    - Improves quality, traceability, maintainability
  - Preserve investment, increases ROI
    - re-use rules across applications and tools
    - skill sets transferable
- Vendors point of view
  - Ease integration with applications, process servers, events servers, business process modeling environments, and business process management systems
  - Extend reach
  - Increase market confidence

Use cases

- Publication
  - Enforcement down the line
  - Mismo, regulatory compliance
- Design once, deploy everywhere
  - ArcelorMittal
- Collaborative design
  - UC1
- Migration
- Negociation
- Persistence

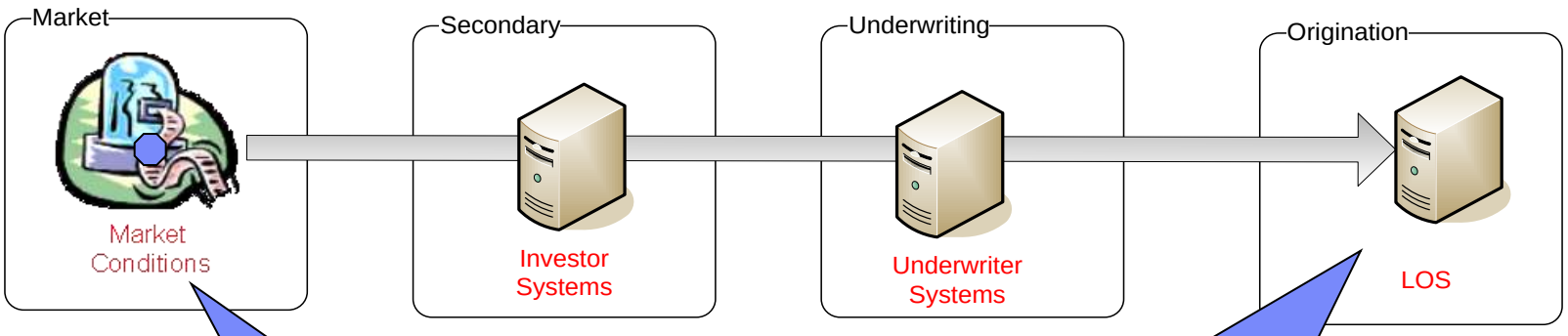
Business Rules interchange

Compliance Use Case



Rule interchange use case

- The expanding effect of changing rules throughout the mortgage process
  - As rules change early in the process their affect can be more significant as it move through the mortgage process.



A relatively small change in interest rates…

…can cause the amount of risk to balloon as it moves through the service chain.

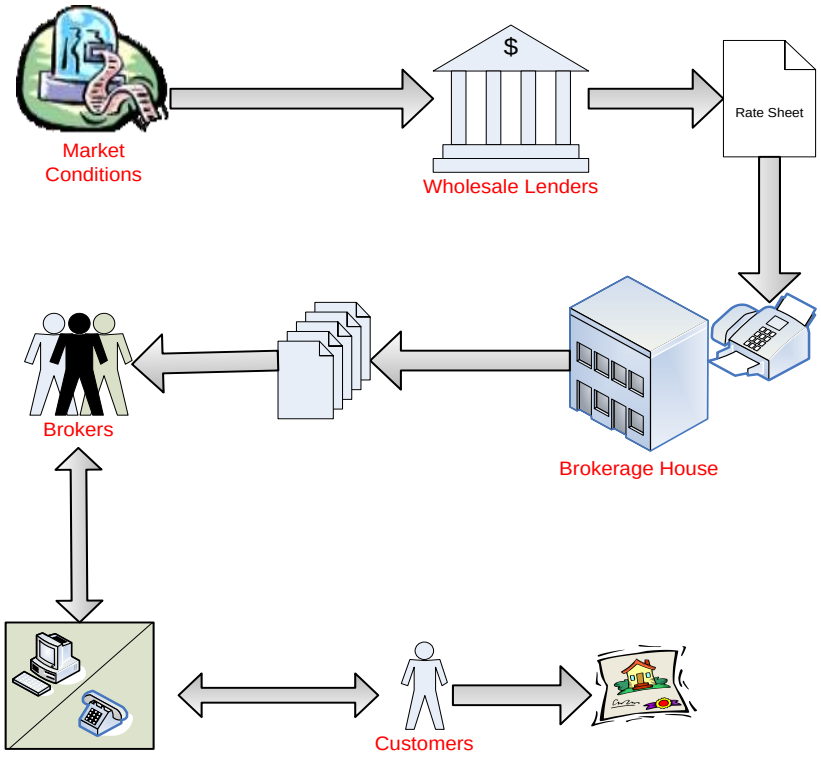
(borrowed from Doug Doedens, FirstAm and MISMO)

Use case: Mortgage industry

- Rate Sheet Distribution Today
  - Wholesale Lenders fax rate sheets to brokers two times per day.
- Processes created by brokers to limit risk
  - All loans must be submitted to a single “approval manager” who looks at current rates and makes sure loans submitted by brokers are based off of the current rates (faxed documents).
  - No loans can be locked after 3pm.

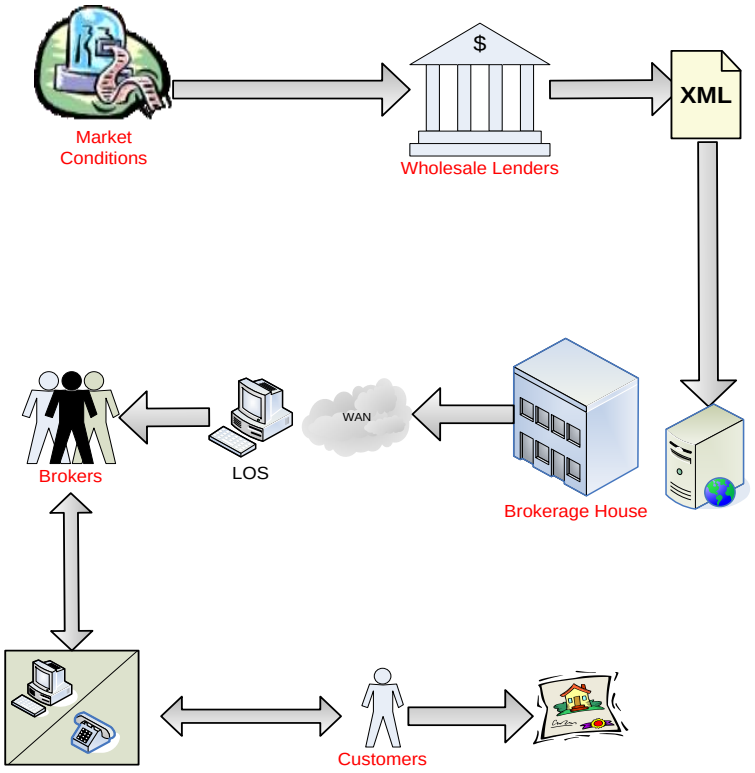
(borrowed from Doug Doedens, FirstAm and MISMO)

Current Rate sheet distribution process



(borrowed from Doug Doedens, FirstAm and MISMO)

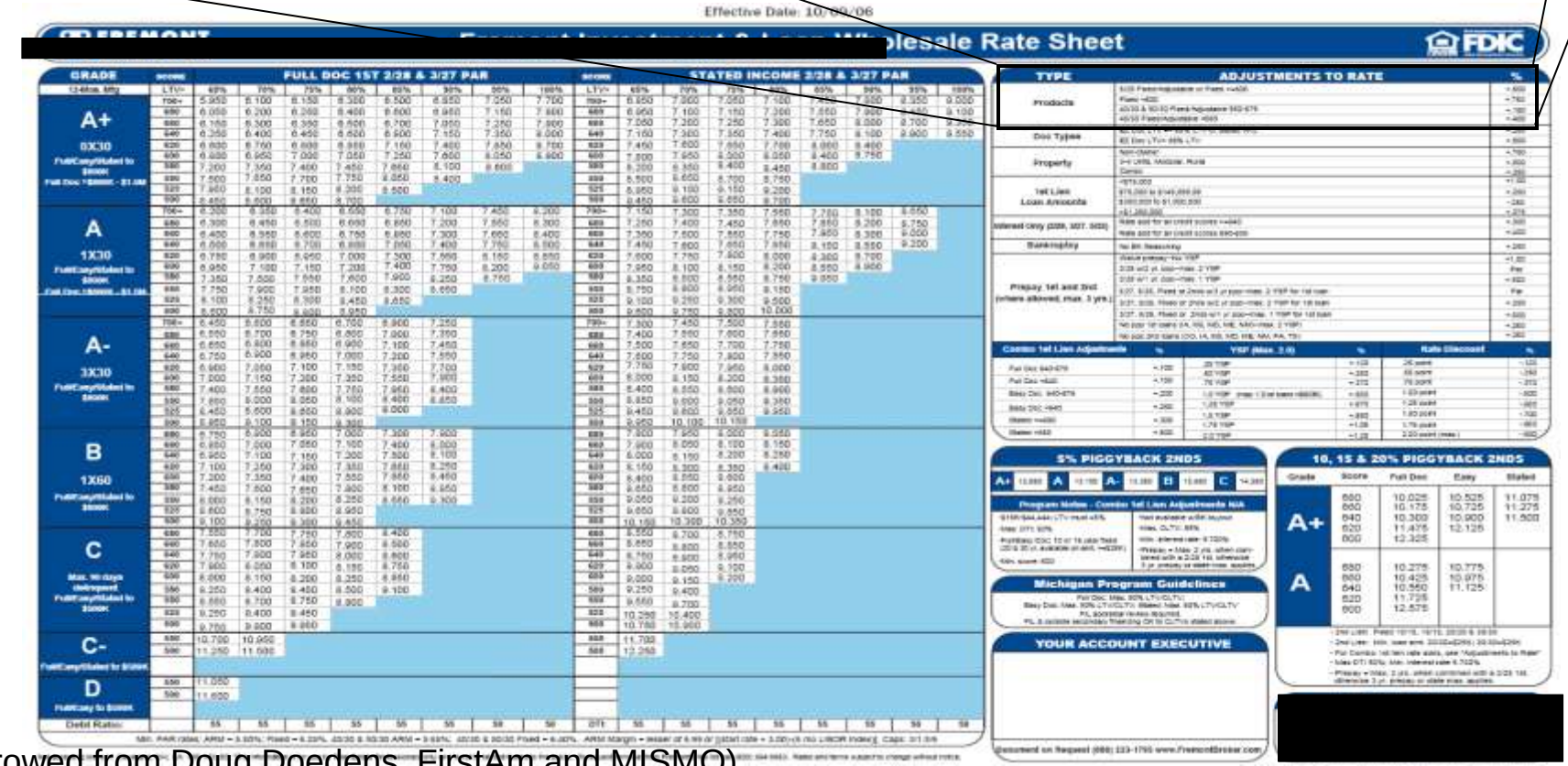
Future distribution process with BR interchange



(borrowed from Doug Doedens, FirstAm and MISMO)

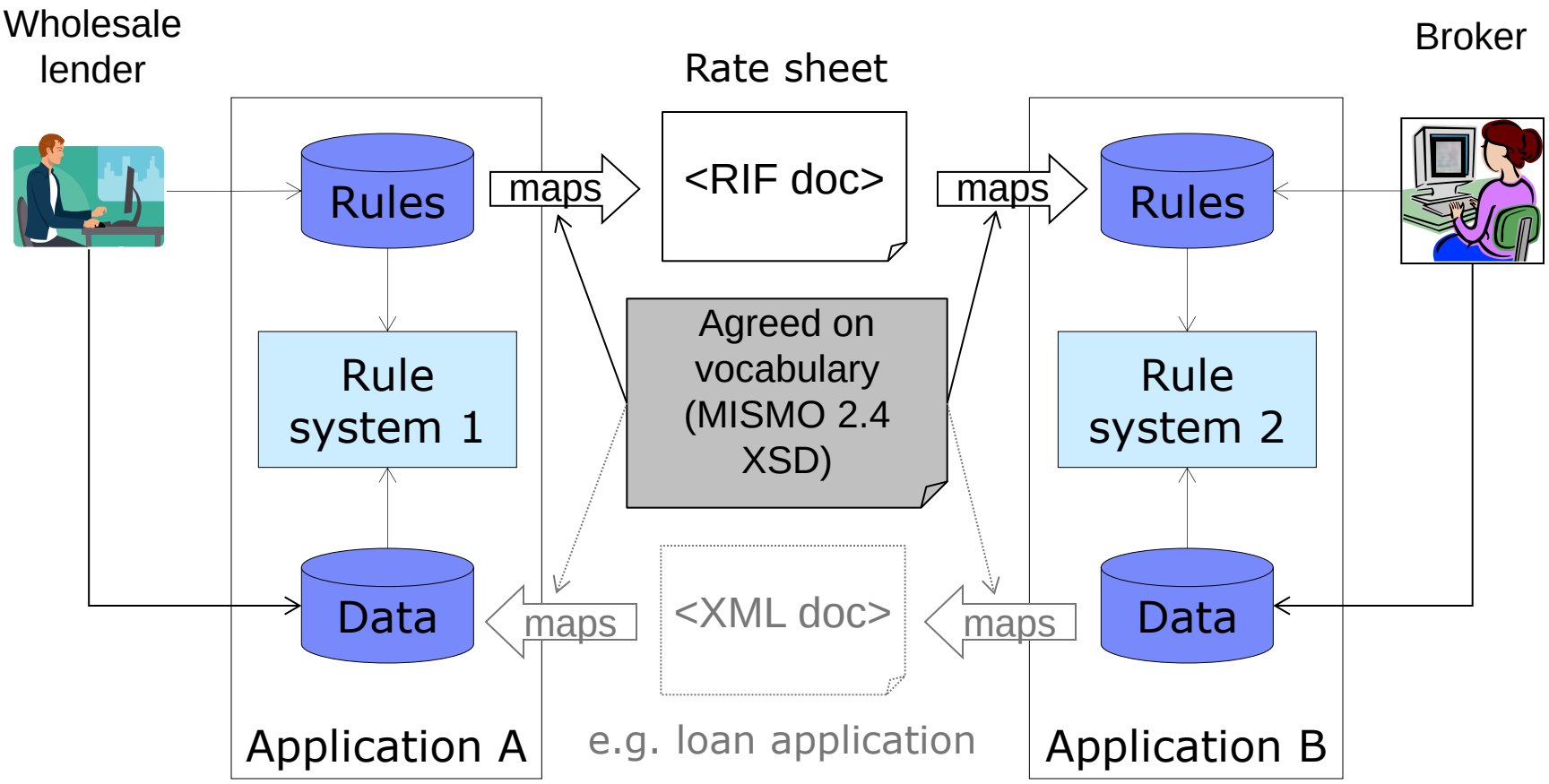
Sample Rate Sheet filled with RULES

TYPE	ADJUSTMENTS TO RATE	%
Products	5/25 Fixed/Adjustable or Fixed >=600	+.500
	Fixed <600	+.750
	40/30 & 50/30 Fixed/Adjustable 550-679	+.150
	40/30 Fixed/Adjustable <550	+.400



(borrowed from Doug Doedens, FirstAm and MISMO)

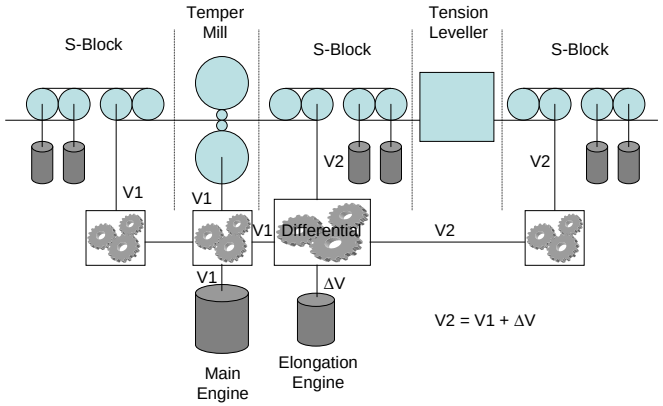
Business rules interchange using RIF



ArcelorMittal

Customer

Line

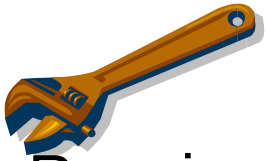


Alternate Customer

Product

Expert Review

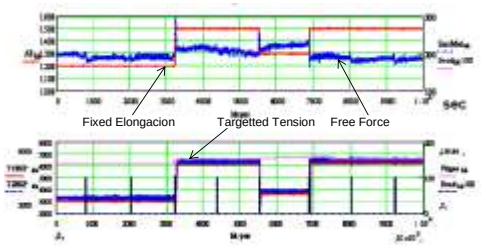
Scrap



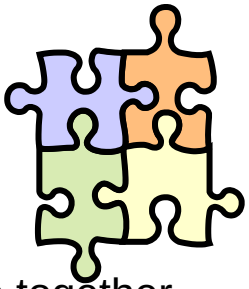
Repair

Cut

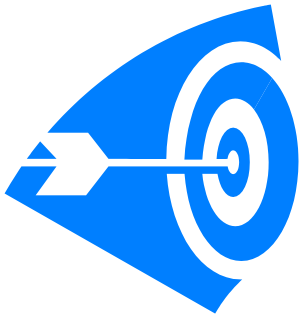
ArcelorMittal



Evaluate process data -> extract events and individual defects

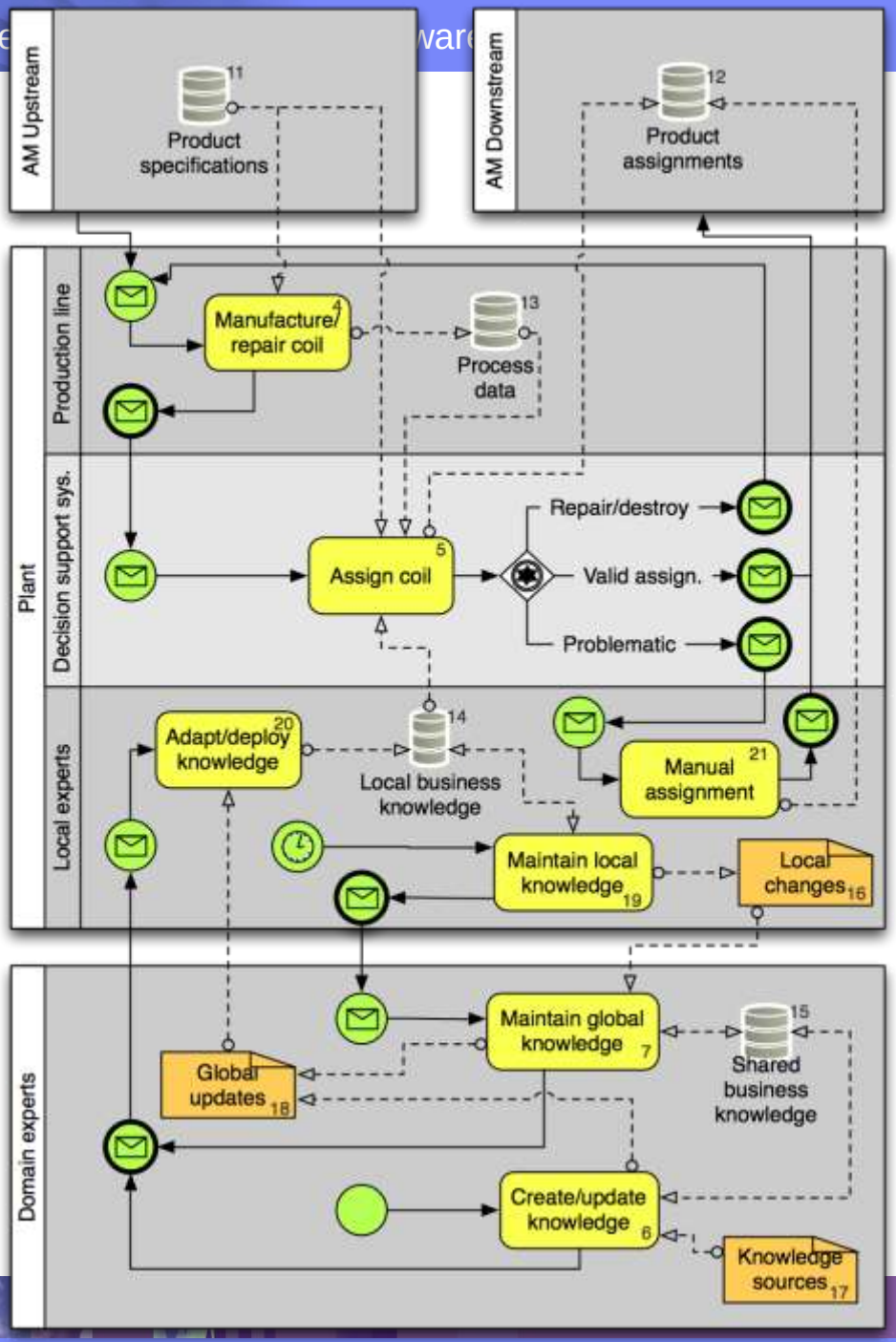


Piece together individual events and defects to characterise the product (types, severity, extension of the defects)



Decide the appropriate path for the product: send to client, downgrade, repair, cut, scrap, to be examined by experts

ArcelorMittal



W3C RIF is not...

- ...a data format (except for rules)
  - Not designed for facts interchange
- ...a data modeling language
  - Not designed for data models interchange
- But rules are about data and data obeys a data model...
- RIF specifies the combination with data and data modeling standards
  - RIF combination with RDF and OWL
  - RIF combination with XML data and XML schemas

The Import directive

```
<Import>
    <location> xs:anyURI </location>
    <profile> xs:anyURI </profile>?
</Import>
```

- Import RIF documents
  - Semantically equivalent to merging the importing and imported document, with a caveat:
    - `rif:local` constants are local to a document
    - Two `rif:local` constants with the same literal that occur in two different documents are not equal
- Import RDF and OWL graphs (schema and/or data)
  - RDF triple *s p o* mapped to frame *s'[p'->o']*
    - Conditions on data types alignment
  - OWL 2 Full compatibility is straightforward extension of RDF compatibility
  - OWL 2 DL requires syntactic restrictions and semantic extension of RIF frames
- Import XML Schema and/or data
  - Element names and schema types mapped to class names
  - Sub-element and attribute names mapped to slot names

```
Forall ?c such ?c #  "/Customer"
    Forall ?a such that And( ?a # "/Customer/Address"
                             ?c["Address" ->?a]
      If Or( ?c [@xml:lang -> "en" ^^xs:language]
             ?a["City" -> "London"] )
      Then Do( Assert( speakEnglish(?n) ) )
```

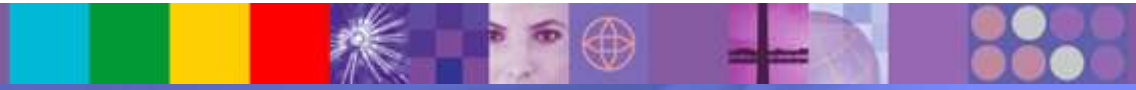
```
<Customer xml:lang="en">
   <Name> John </Name>
   <Age> 39 </Age>
   <Address>
     <City> London </City>
     <Street> ... </Street>
   </Address>
 </Customer>
```

Conclusion

- What we have now, with W3C RIF
  - Consistent foundations for standard XML-based rule interchange
  - Covers the two main families of rule languages
  - Combines with XML, XSD, RDF, OWL
  - Ready for deployement
- Why you should implement, use and/or deploy it now
  - « Because it's there! » (George Mallory)
  - XYZ
  - @@@
- What we need next
  - Atomic Modify for RIF-PRD
  - A standard Java RIF API?
    - An obvious complement to the Java Rule Engine API (JSR-94)
  - Events? Aggregates?

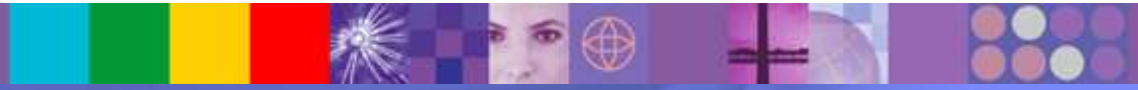
Thank you!

Questions?



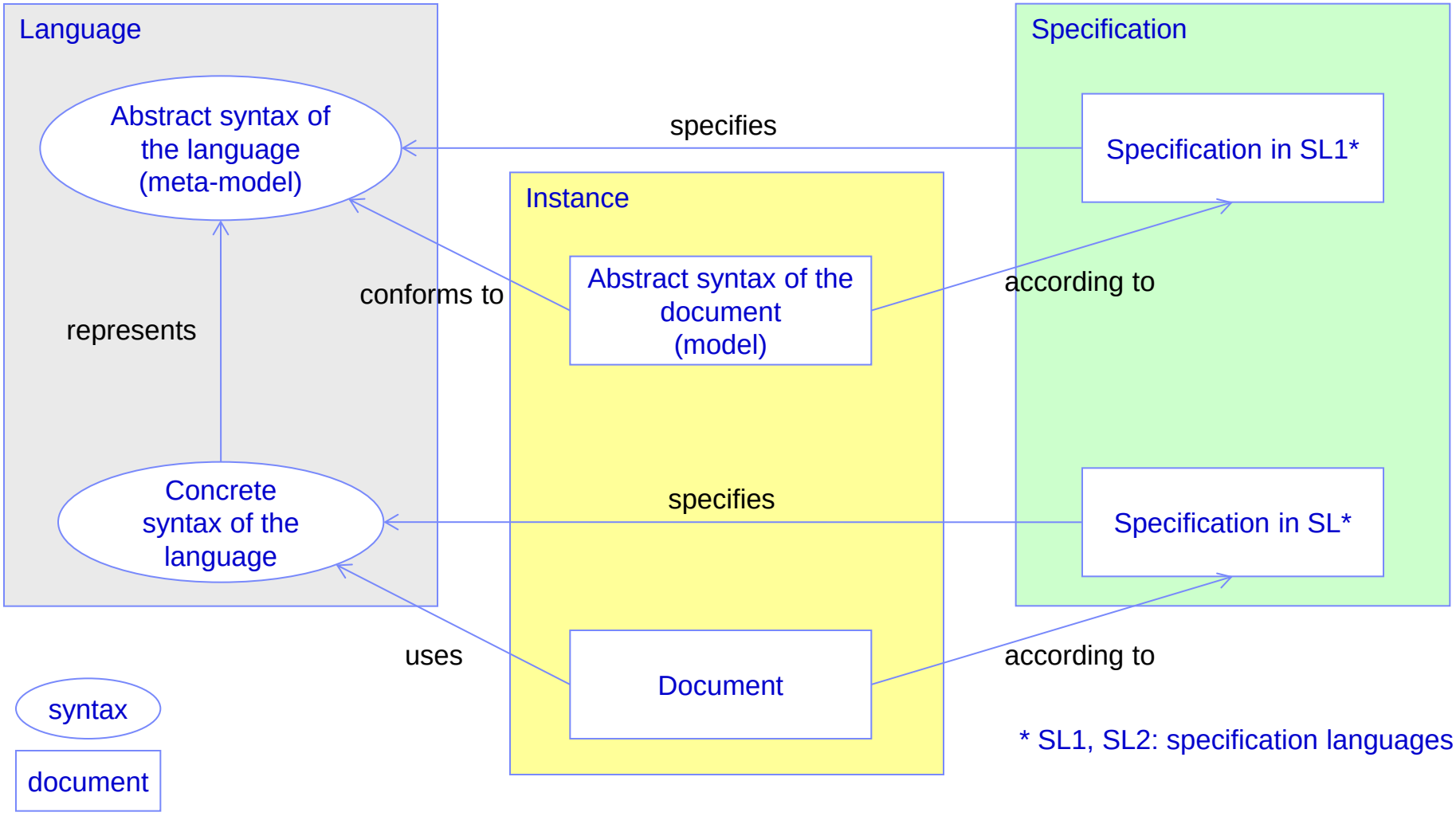
WebSphere software

Back-up slides

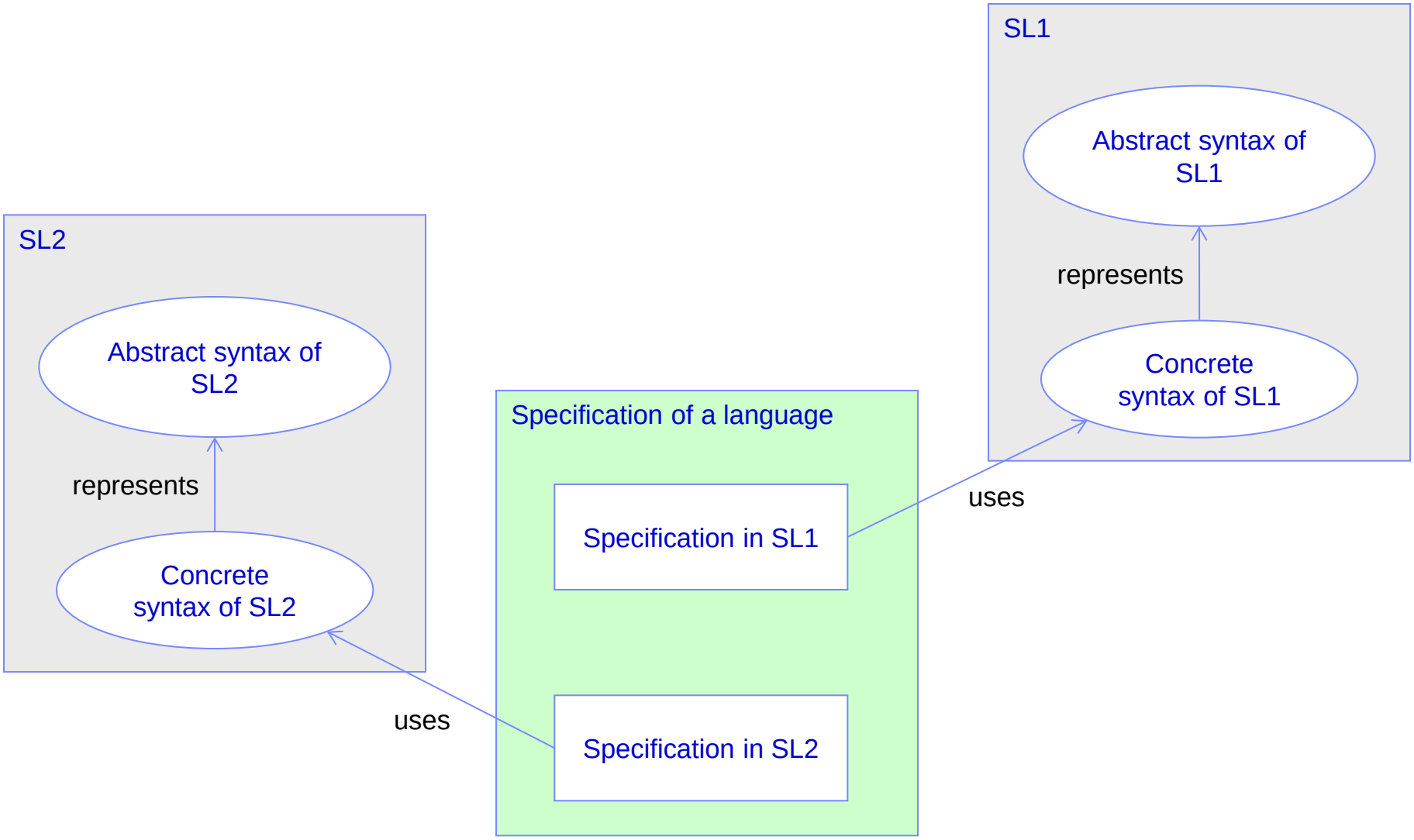


WebSphere software

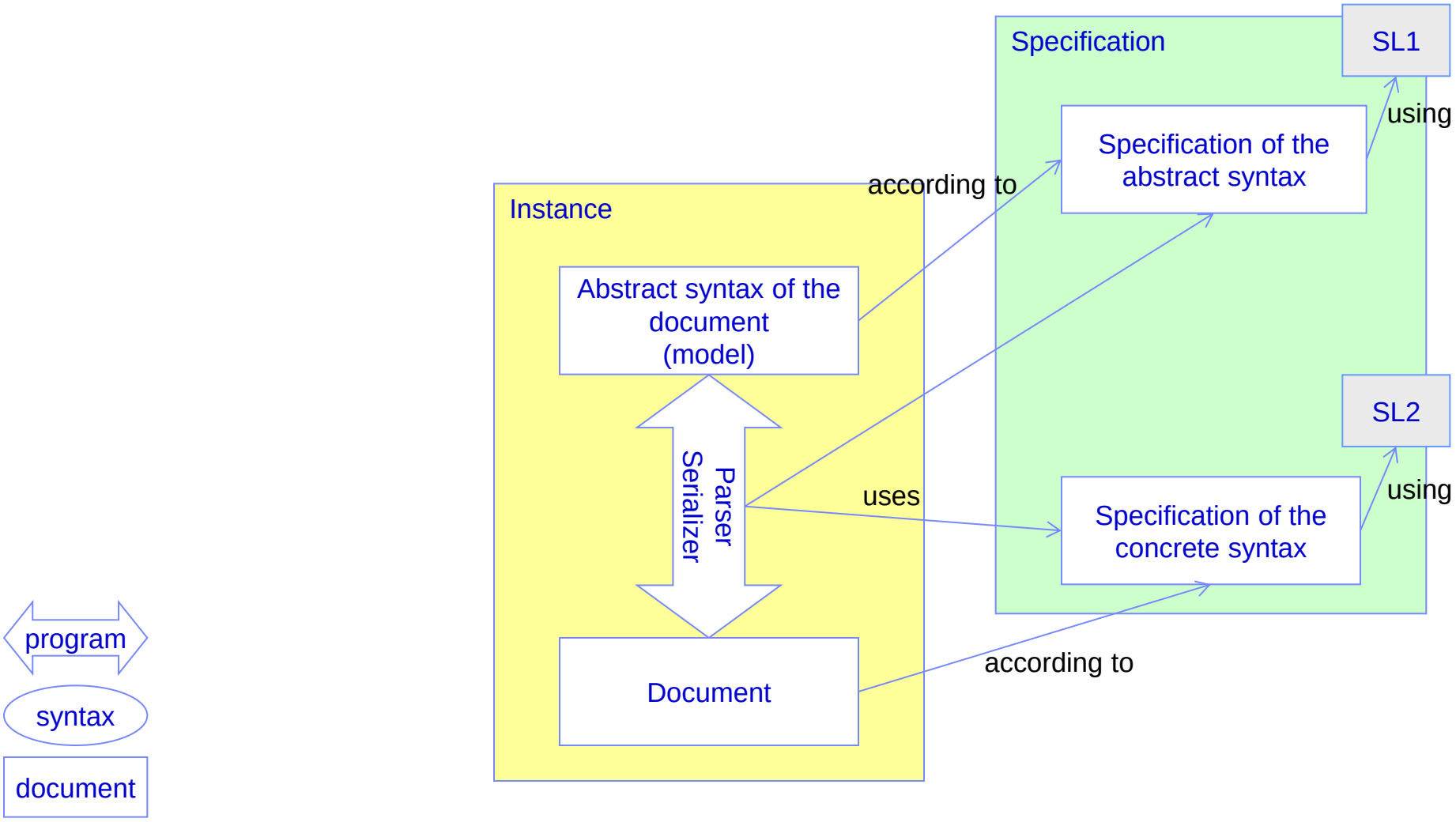
Specification of the syntax of a language



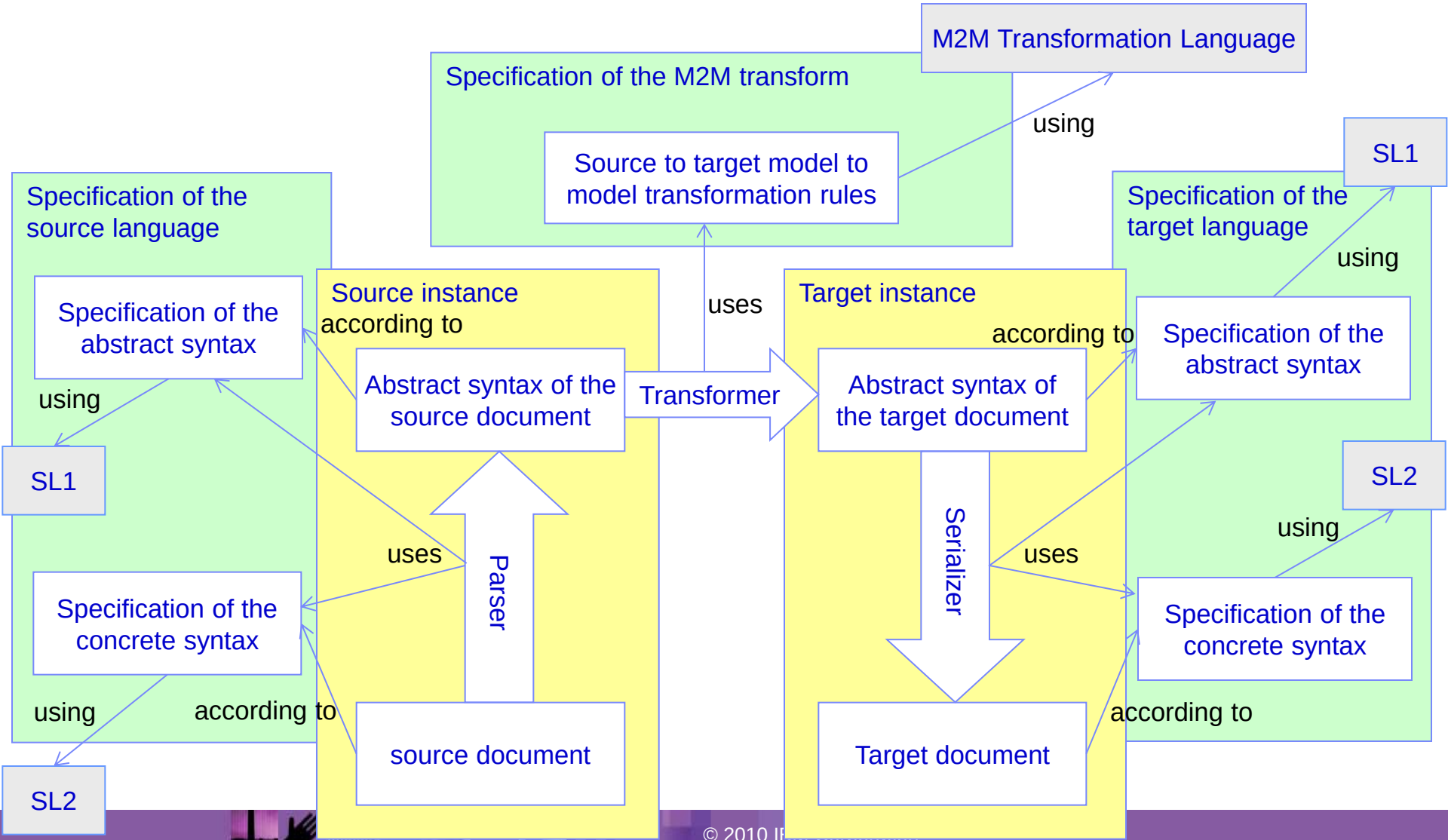
NB: SL1 and SL2 are languages



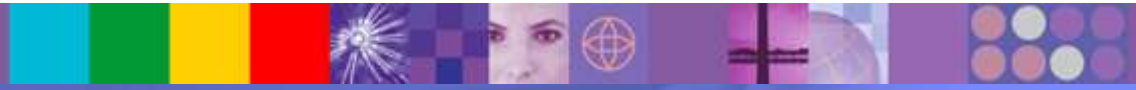
Using the specification for parsing and serialization



Using the specification for translating documents



Rule standards



WebSphere software

Standards in the rules life cycle

Requirements analysis

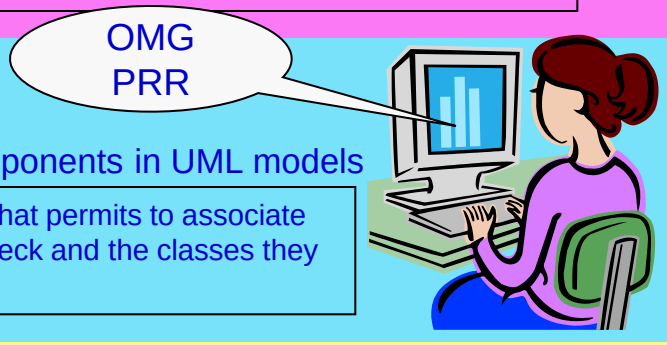
Model business rules and business vocabularies in a verifiable and communicable way

OMG SBVR defines the vocabulary and rules for documenting the semantics of business vocabularies, business facts, and business rules

Application design

Include rules as 1st class components in UML models

OMG PRR is the UML extension that permits to associate each rule with the classes they check and the classes they impact



Deployment and operation

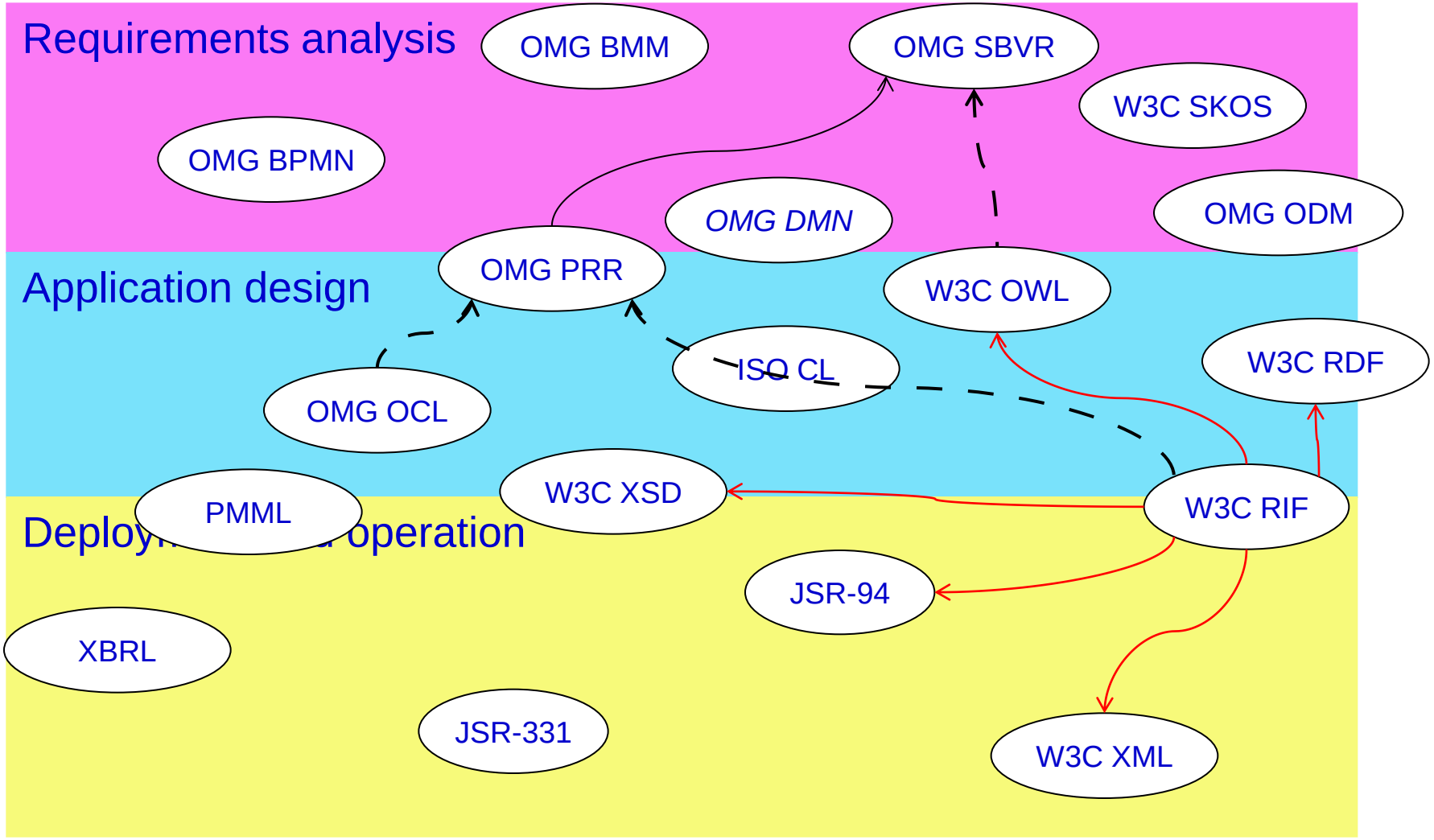
Write rules once, use them everywhere
Share rules with business partners
Publish/retrieve rules

W3C RIF is the standard XML serialization for rules

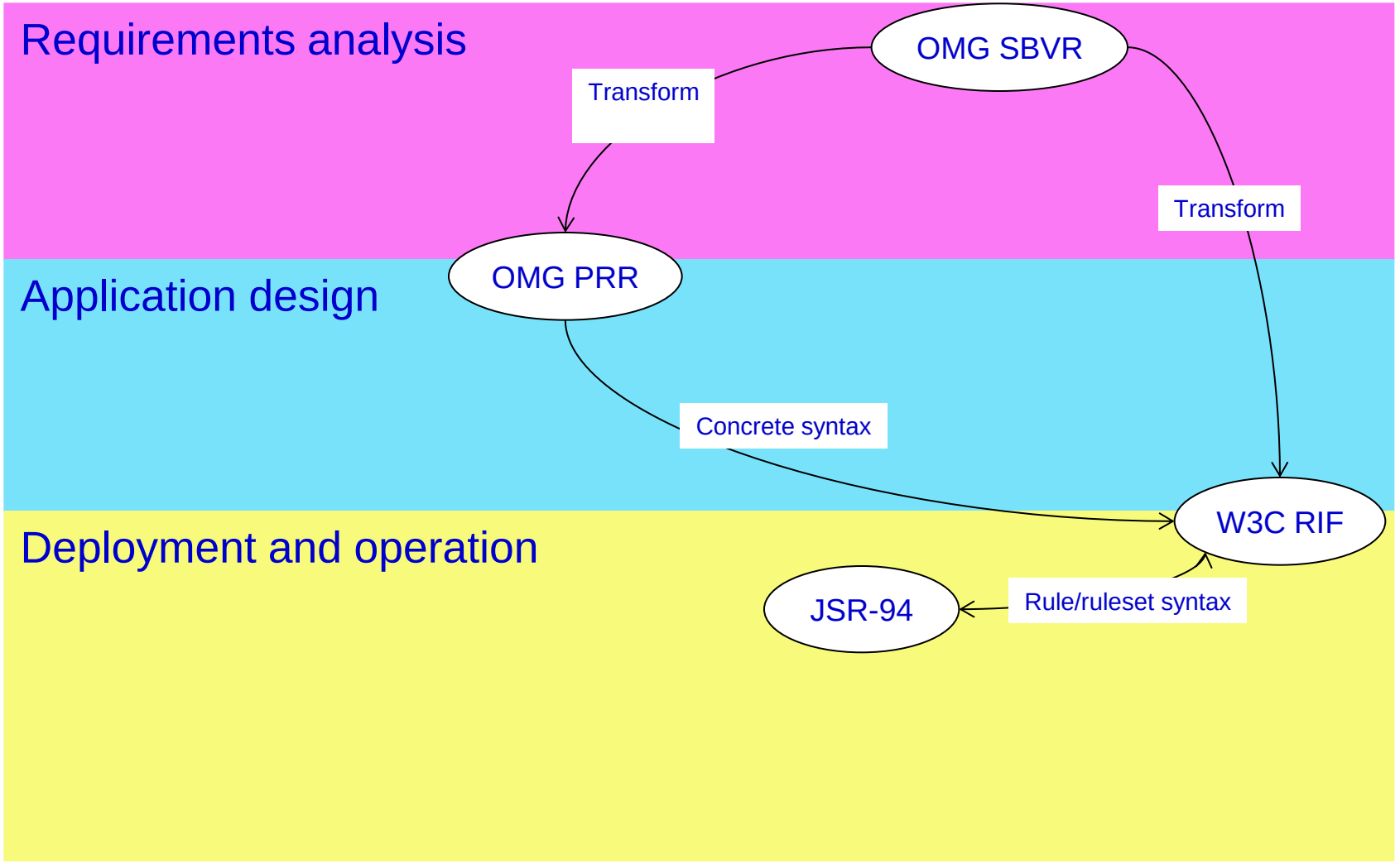
Embed rule engines in your Java applications

JSR-94 is the Java API for rule engines

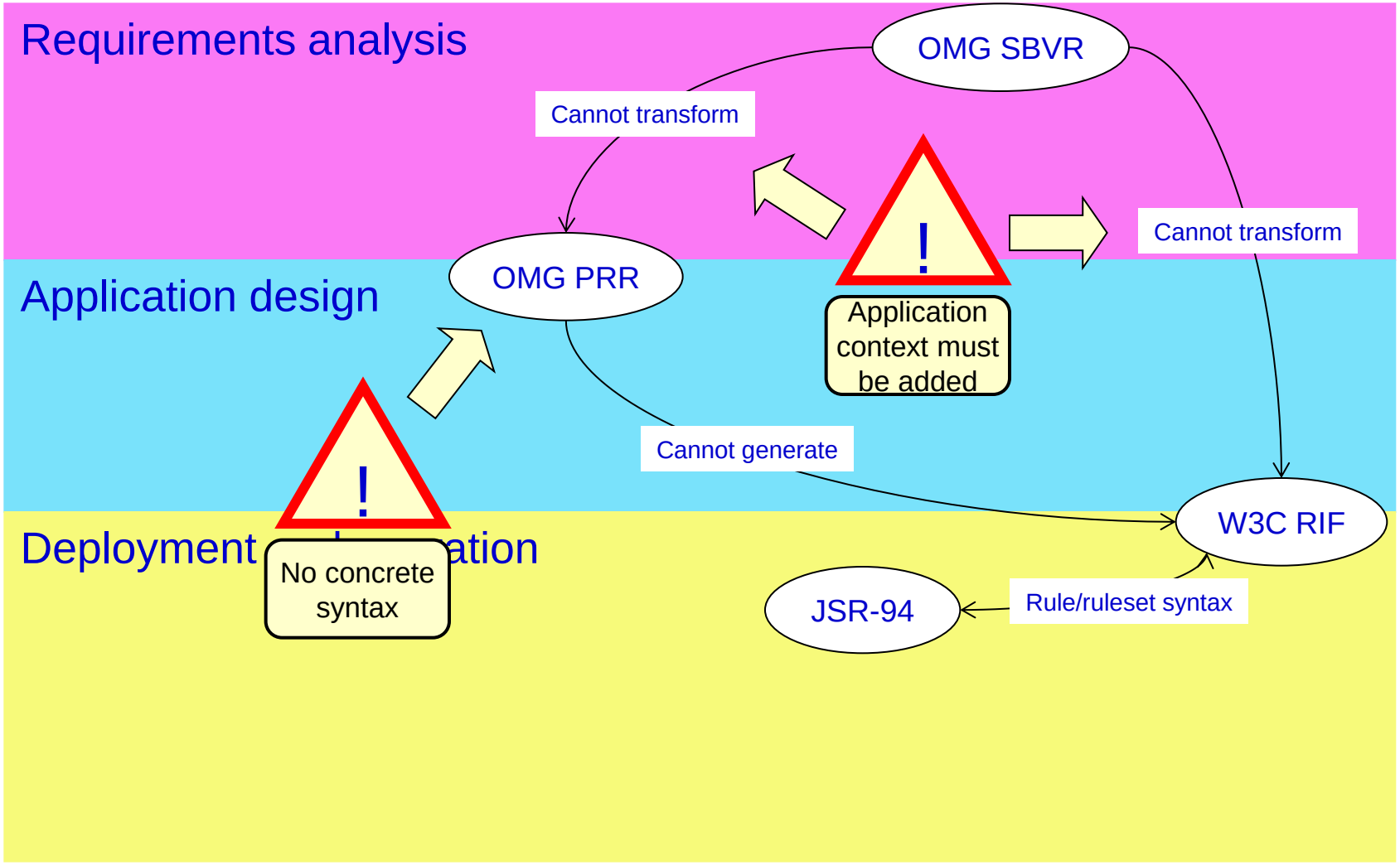
Rule standards and surrounding



What's next? Convergence



What's next? Convergence



What's next? Challenges