



IBM Software Group

# WebSphere ILOG JRules Execution Server Packaging Applications

Franck Delporte ([franck.delporte@us.ibm.com](mailto:franck.delporte@us.ibm.com))

WebSphere ILOG JRules

31 August 2010



WebSphere® Support Technical Exchange



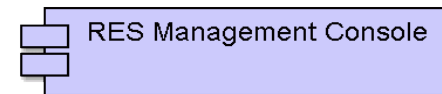
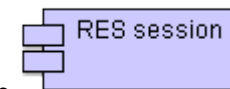
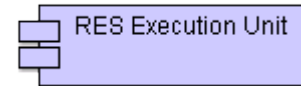
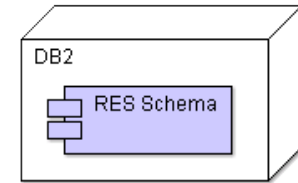
# Agenda

- Introduction : RES Components and Protocols
- Typical Deployments
  - ▶ Local
    - J2EE Single Server
    - J2SE (with and without Tomcat)
  - ▶ Remote
    - J2EE Remote Rule Session
    - HTDS Web Service
    - MTDS Web Service
  - ▶ J2EE Cluster
- Class loading and related issues

# Introduction : RES Components

## RES Principal “Building Blocks”

- ▶ Data store for the ruleapps/rulesets, the RES DB Schema
  - File persistence is an option, not presented here
- ▶ Execution Unit
  - e.g. : jrules-res-xu-WAS7.rar
- ▶ Session/client library ( whether a java utility jar or an ejb-jar )
  - e.g. : jrules-res-session-**java**.jar , jrules-res-session-**WAS7**.jar
- ▶ Optional Management Stack and Console Application
  - e.g. : jrules-res-management-WAS7.ear



## Application specifics:

- ▶ Executable Object Model (XOM)
- ▶ Web Application



# Introduction : RES Protocols and Interfaces

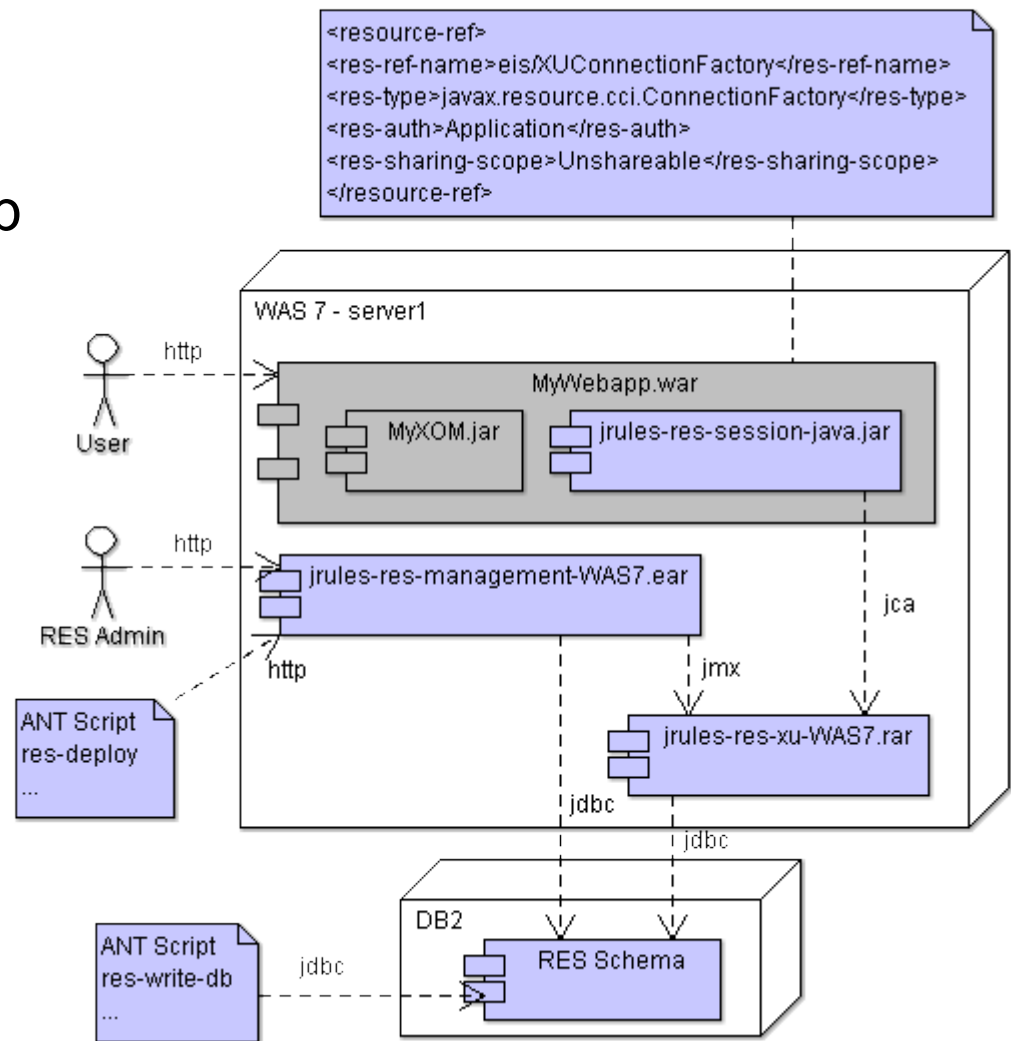
- Protocols and Interfaces used in RES :
  - ▶ HTTP
    - RES Management Console user interface
    - Rule Studio and RES Ant tasks use HTTP to interface with RES Management Console
  - ▶ JDBC : RES Management Console and Execution Unit both access the RES DB Schema
  - ▶ RMI-IIOP / EJB : Remote EJB Rule Sessions
    - Note: other rule session invocations (POJO, J2SE, EJB local) are Local to the JVM
  - ▶ JMS : RES Message Driven Bean (MDB)
  - ▶ JCA : Connection Management between RES Session and XU
  - ▶ JMX : RES Management Console and Execution Unit expose JMX MBeans

# Agenda

- Introduction : RES Components and Protocols
- Typical Deployments
  - ▶ Local
    - J2EE Single Server
    - J2SE (with and without Tomcat)
  - ▶ Remote
    - J2EE Remote Rule Session
    - HTDS Web Service
    - MTDS Web Service
  - ▶ J2EE Cluster
- Class loading and related issues

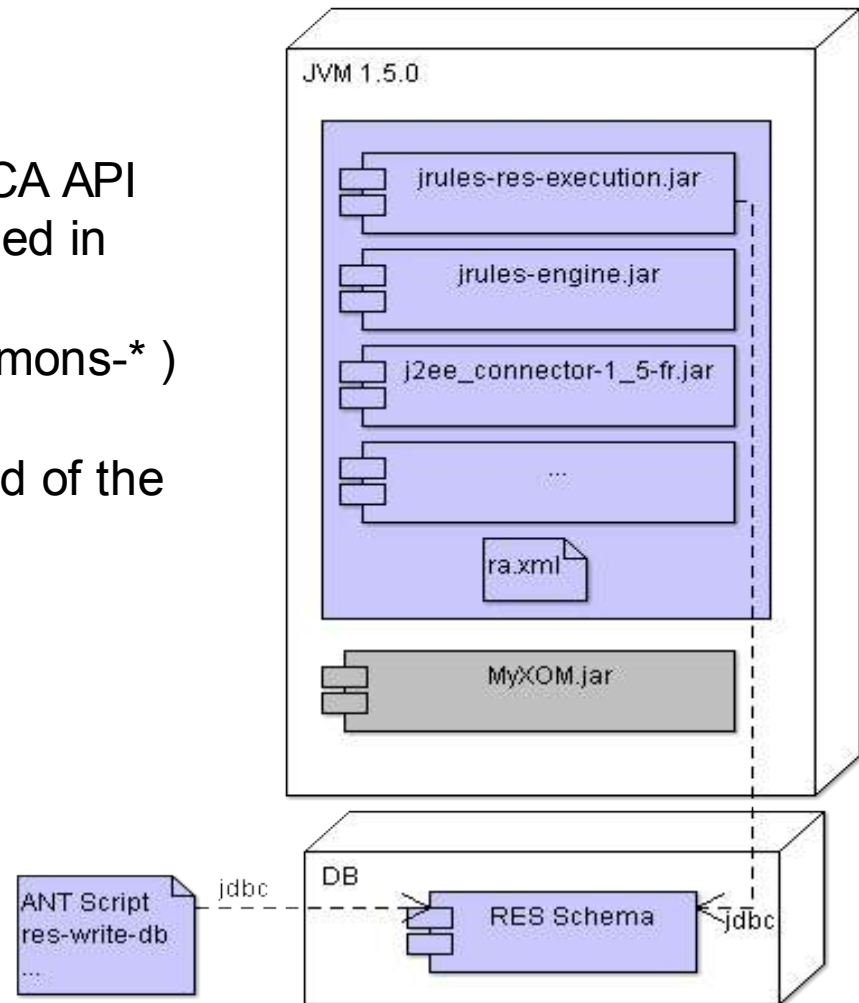
# J2EE Deployment

- Single Server Setup with Web Application
- XOM and RES Session jar in WEB-INF/lib
- Management Console is optional
- OOTB Hot Deployment notifications require the Management Console
- XU configuration through `<XU>/META-INF/ra.xml`



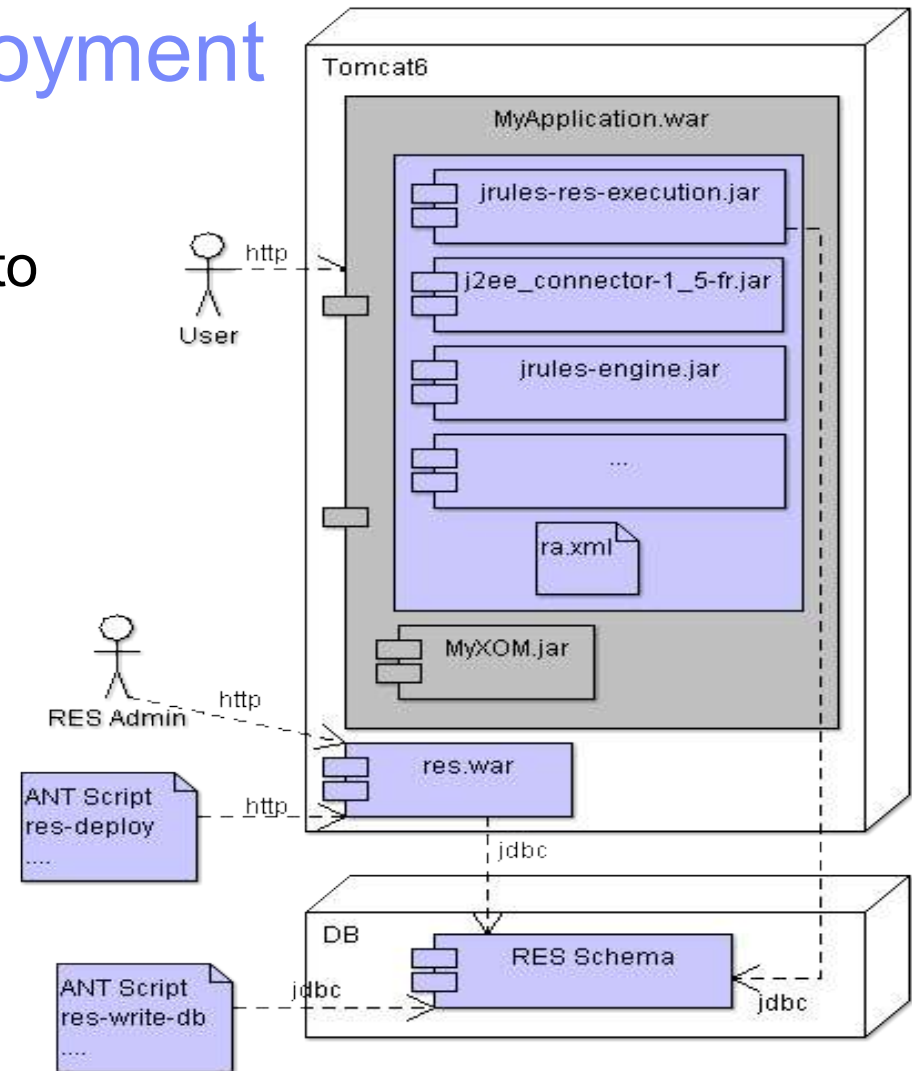
# J2SE Deployment

- **j2ee\_connector-1\_5-fr.jar** provides JCA API
- XU and client session library are merged in **jrules-res-execution.jar**
- XU dependencies (jrules-engine, commons-\*) are added individually on classpath
- **ra.xml** should be on classpath (instead of the bundled <XU>/META-INF/ra.xml)



# J2SE with Tomcat Deployment

- Same as J2SE plus RES Console
- No shared rule engine, XU needs to be deployed with any ruleset invoking Web Application
- No clustering support for management and hot-deploy
- No remote rule session (no EJB container)



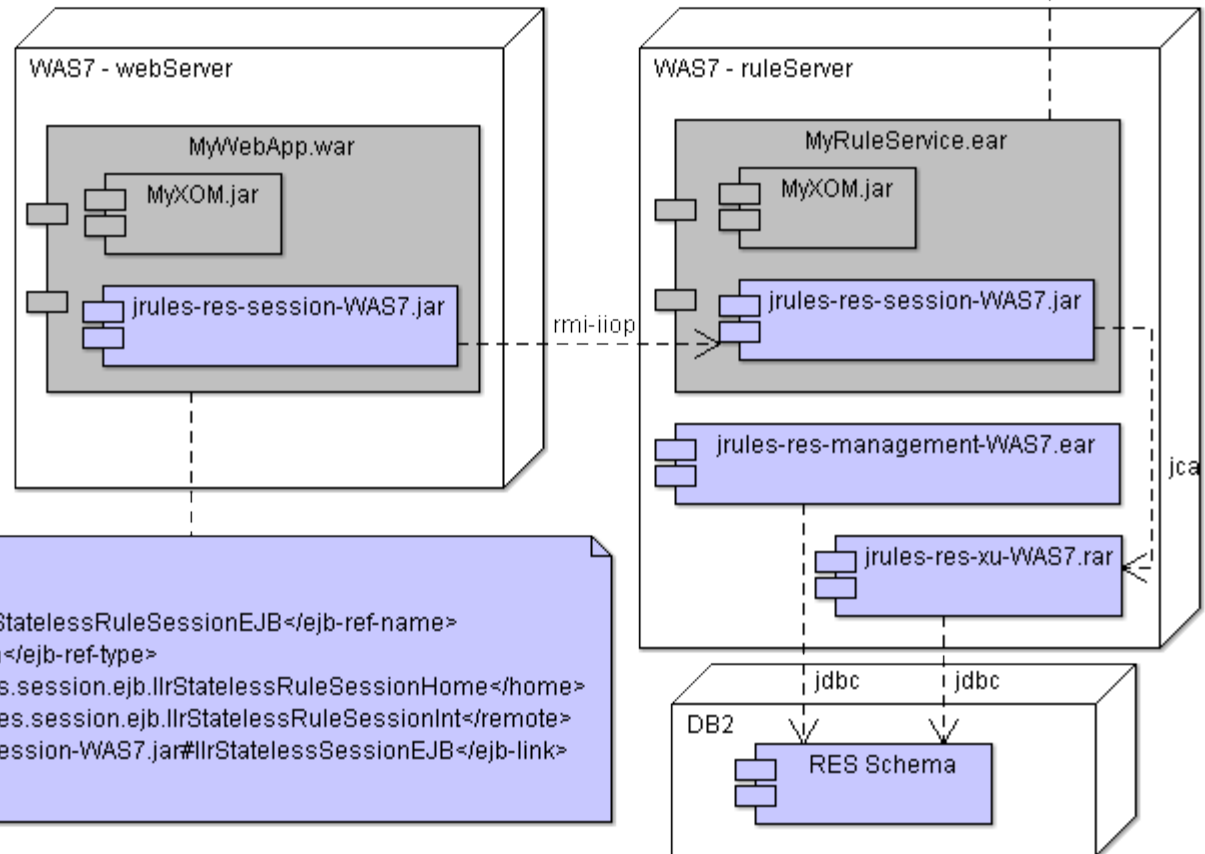


# Agenda

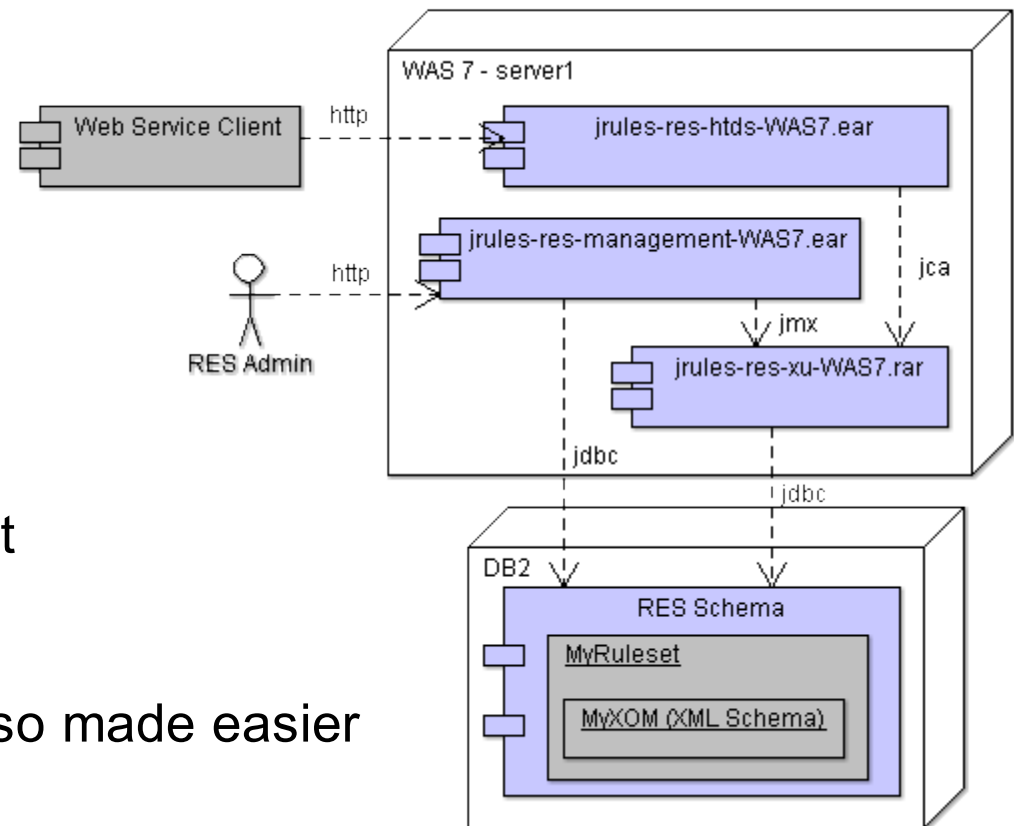
- Introduction : RES Components and Protocols
- Typical Deployments
  - ▶ Local
    - J2EE Single Server
    - J2SE (with and without Tomcat)
  - ▶ Remote
    - J2EE Remote Rule Session
    - HTDS Web Service
    - MTDS Web Service
  - ▶ J2EE Cluster
- Class loading and related issues

# J2EE Remote Rule Session

- Web front-end and Rule Server hosted on different servers
- XOM needs to be packaged on both sides
- Package RES Session EJB-JAR ( instead of the POJO/Java session Jar)



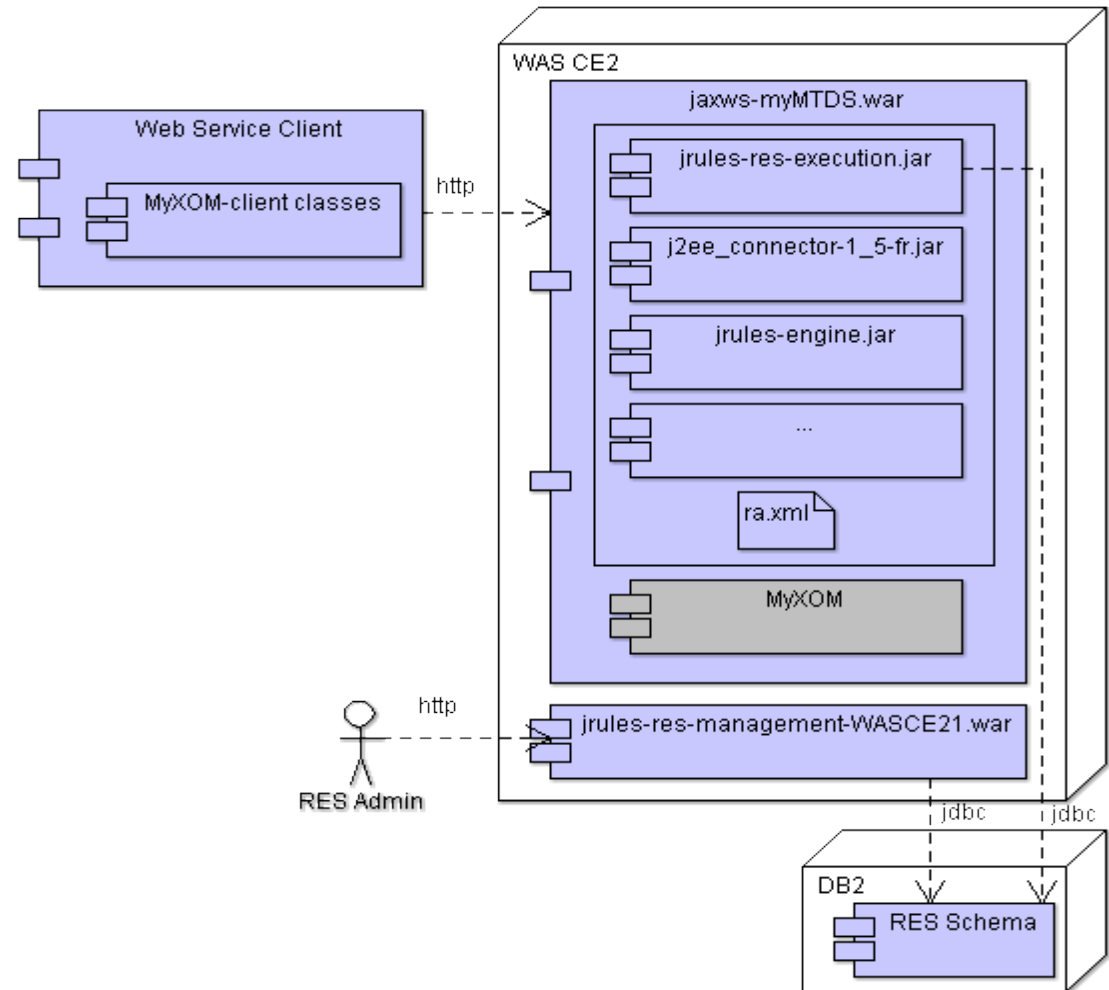
# Hosted Transparent Decision Service (HTDS)



- XOM is bundled with Ruleset
  - ▶ Simplifies packaging
  - ▶ Versioning of XOM is also made easier

# Monitored Transparent Decision Service (MTDS)

- JAX-WS Web Service, WSDL derived from Ruleset signature
- Wizard in Rule Studio generates server and sample client
- Only supported with:
  - WebSphere CE2
  - JBoss
  - Tomcat

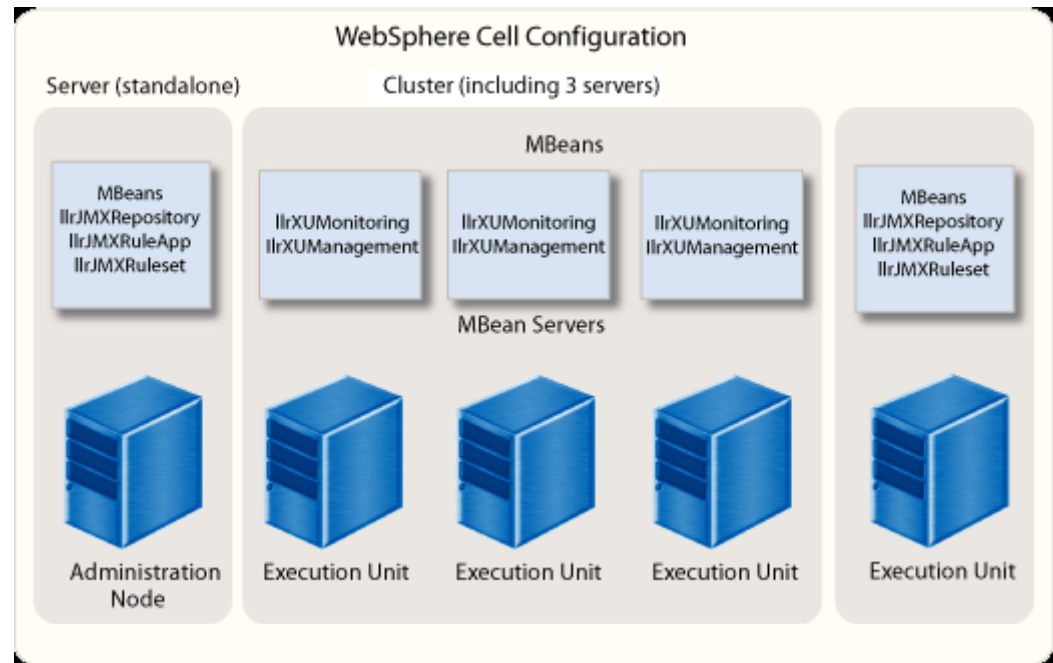


# Agenda

- Introduction : RES Components and Protocols
- Typical Deployments
  - ▶ Local
    - J2EE Single Server
    - J2SE (with and without Tomcat)
  - ▶ Remote
    - J2EE Remote Rule Session
    - HTDS Web Service
    - MTDS Web Service
  - ▶ **J2EE Cluster**
- Class loading and related issues

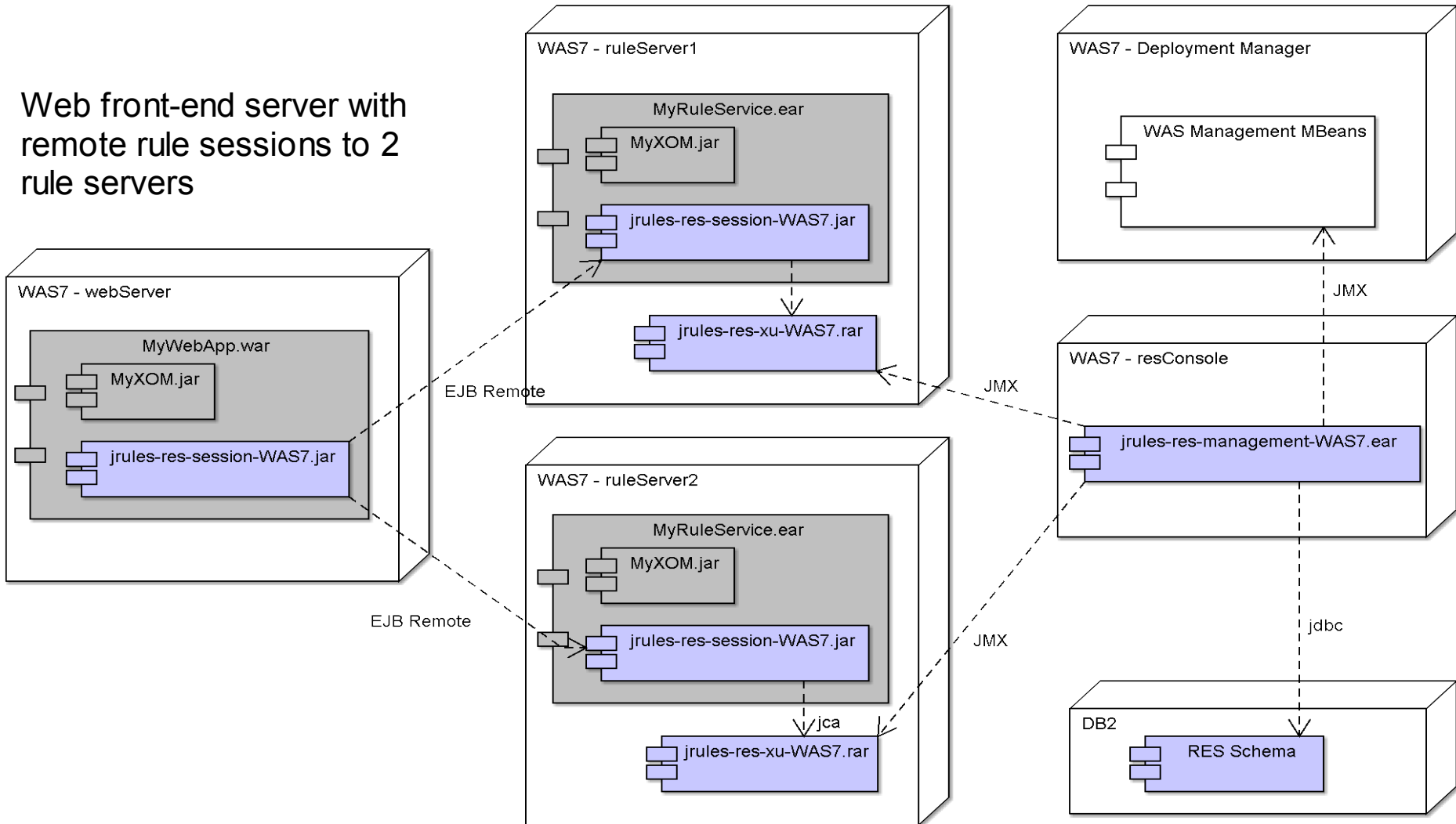
# J2EE Cluster Deployment (1/2)

- By default only one RES Console per Cell / Domain / Partition
  - ▶ Unless they are used to manage separate XUs (with different JMX xuName)
- RES Management stack only spans a single Cell / Domain / Partition



# J2EE Cluster Deployment (2/2)

- Web front-end server with remote rule sessions to 2 rule servers



# Agenda

- Introduction : RES Components and Protocols
- Typical Deployments
  - ▶ Local
    - J2EE Single Server
    - J2SE (with and without Tomcat)
  - ▶ Remote
    - J2EE Remote Rule Session
    - HTDS Web Service
    - MTDS Web Service
  - ▶ J2EE Cluster
- **Class loading and related issues**



# Class loading – Basics / J2SE

- In the JVM classes are associated with a class loader
  - ▶ => (class, class loader) pairs are unique not just classes
- A delegation mechanism is used to lookup classes : each class loader has a single parent

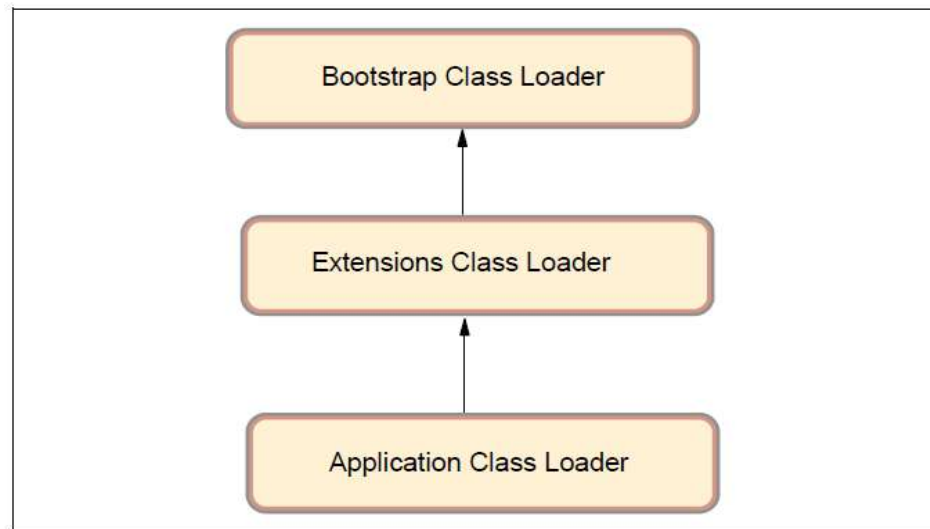


Figure 13-1 Java class loaders hierarchy

- By default Java follows “Parent First” delegation

# Class loading - J2EE

- Typical J2EE class loading hierarchy – WebSphere example

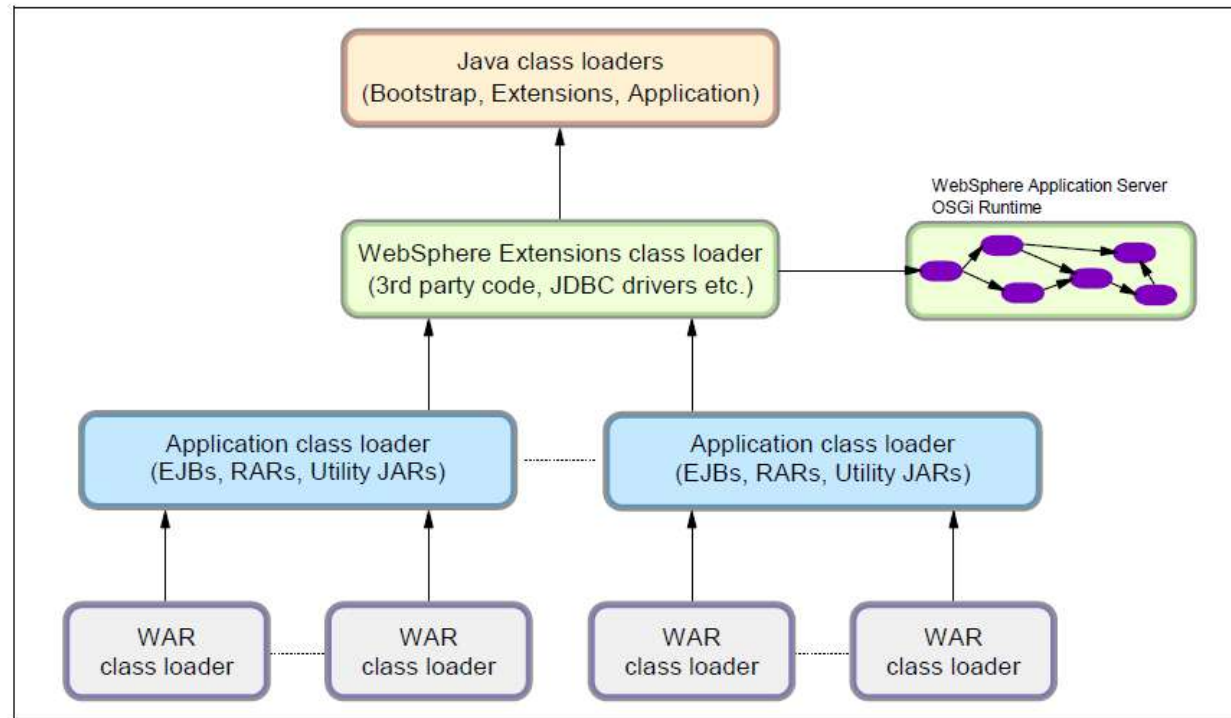
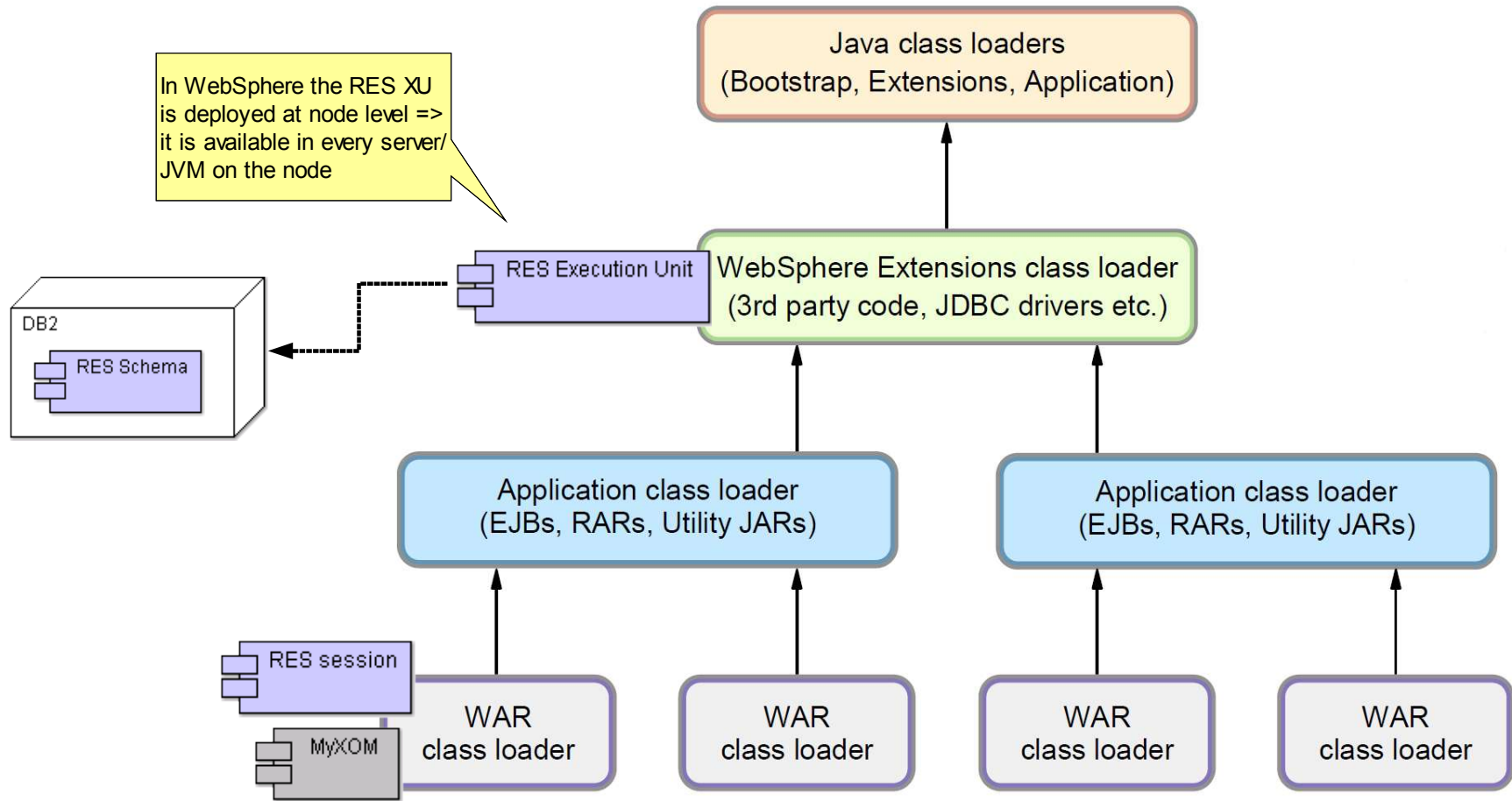


Figure 13-2 WebSphere class loaders hierarchy

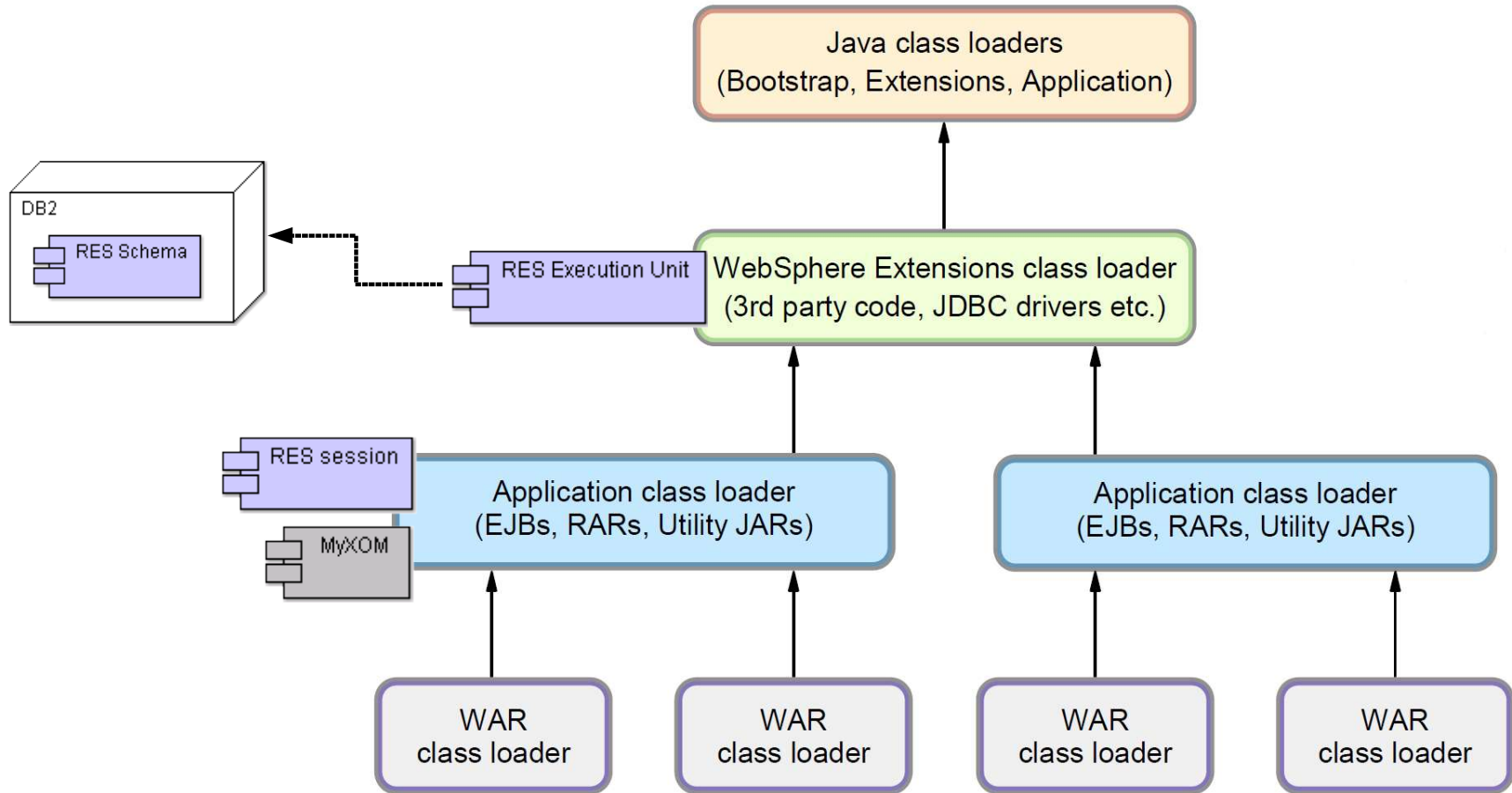
**PARENT\_FIRST** = "Classes loaded with **parent** class loader **first**"

**PARENT\_LAST** = "Classes loaded with **application** class loader **first**"

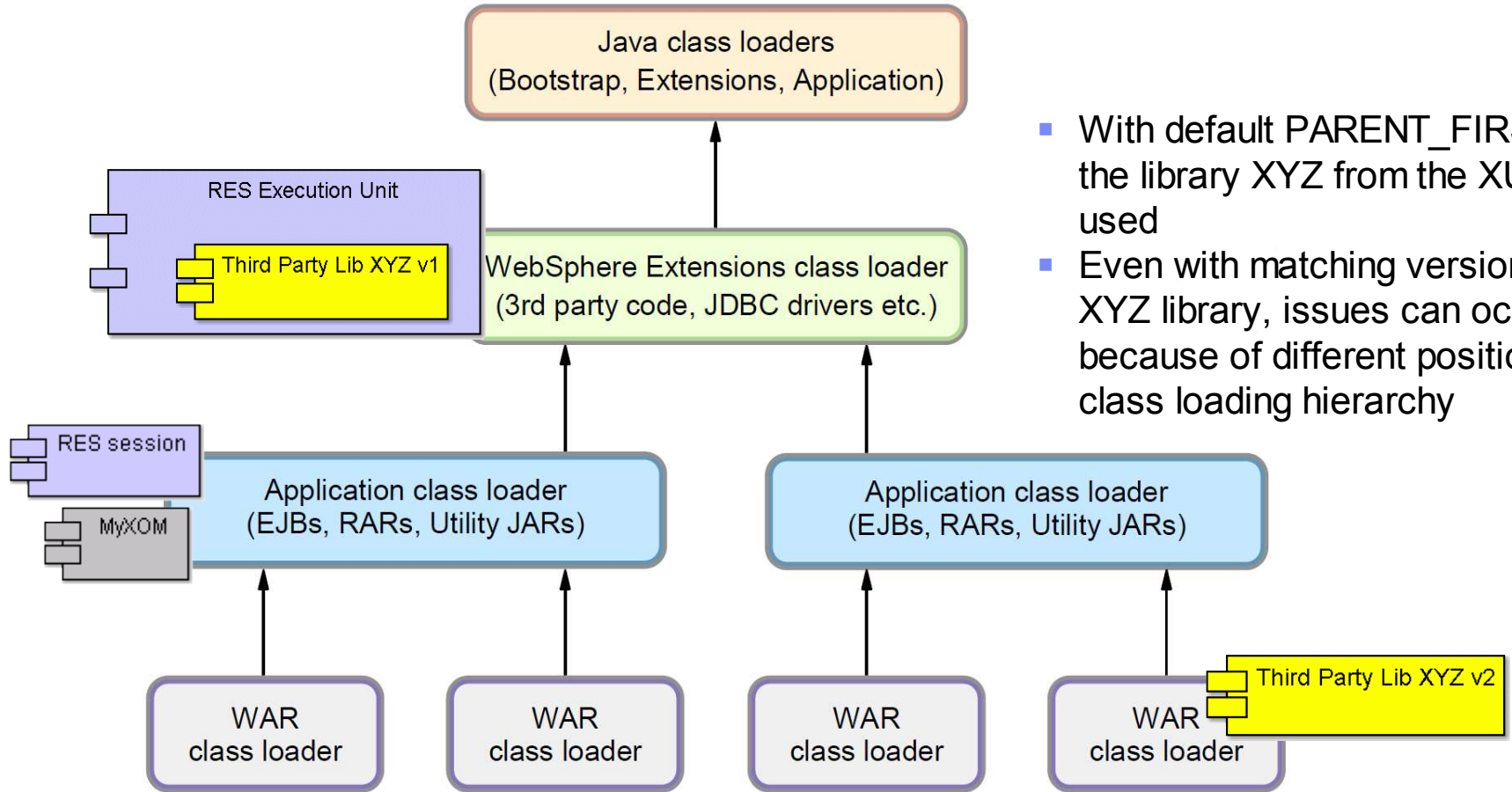
# Class loading – RES J2EE - Web Application



# Class loading – RES J2EE - EJB/MDB

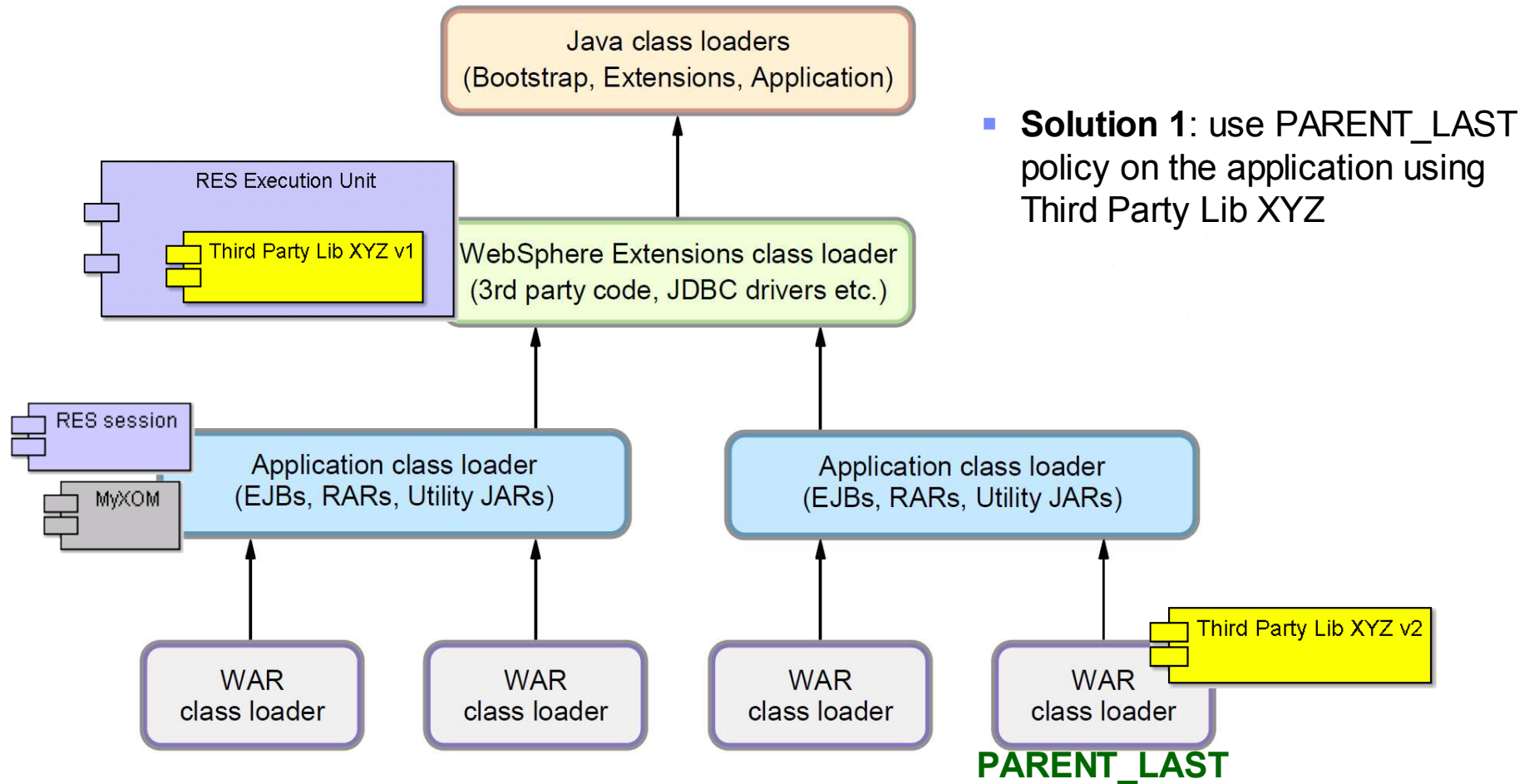


# Class loading – XU Third Party Libraries (1)

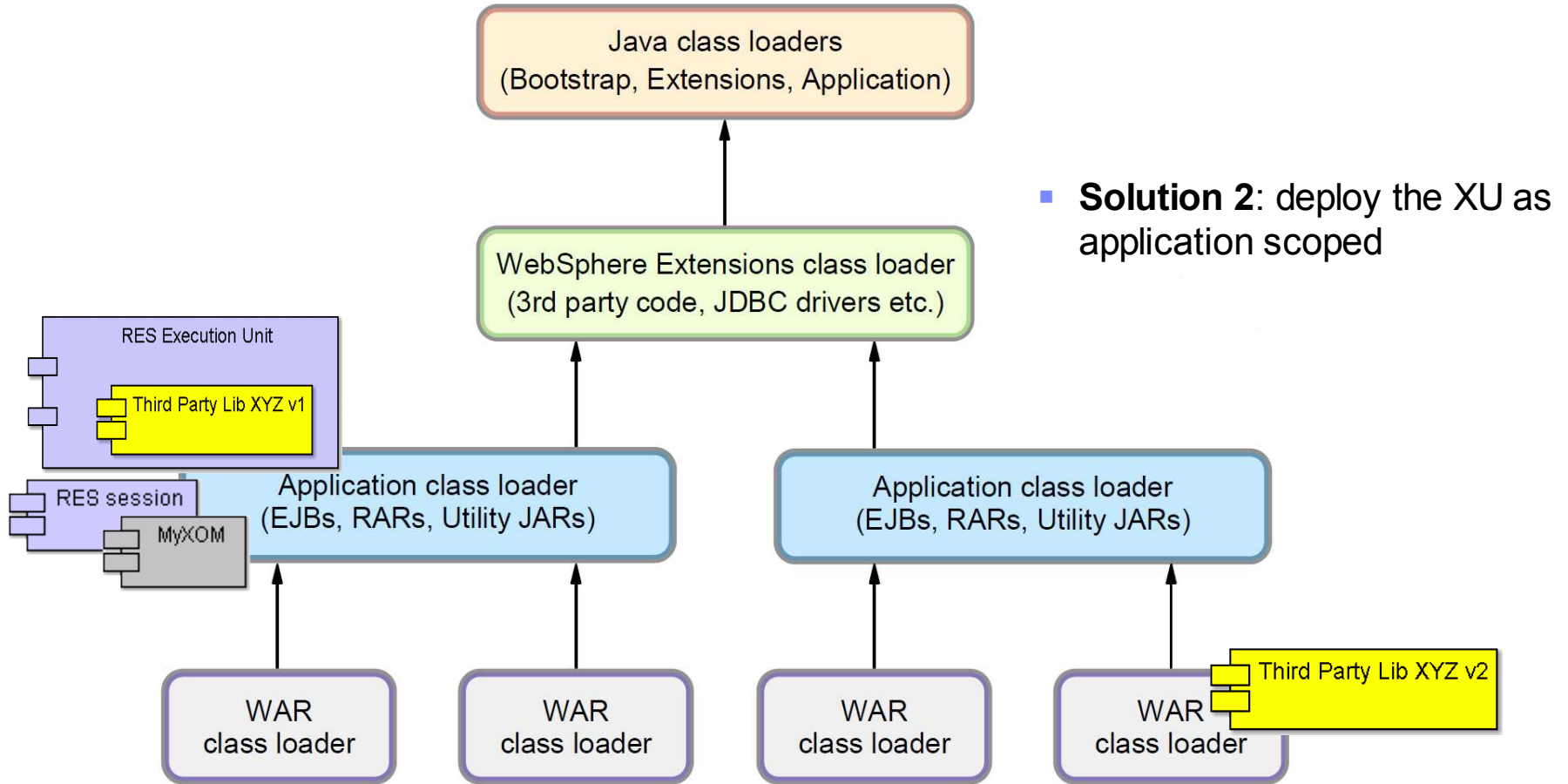


- With default PARENT\_FIRST the library XYZ from the XU is used
- Even with matching versions of XYZ library, issues can occur because of different position in class loading hierarchy

# Class loading – XU Third Party Libraries (2)



# Class loading – XU Third Party Libraries (3)



- **Solution 2:** deploy the XU as application scoped

# Summary

- Identify the right type of RES packaging needed
  - Local vs Remote
  - Java XOM vs Dynamic XOM
  - J2SE vs J2EE
- Package your RES application with all necessary dependencies
  - WEB-INF/lib or also APP-INF/lib
  - META-INF/MANIFEST Class-Path entry
  - EAR java utility module
- Beware of classloading issues
  - ▶ Diagram of the class loading hierarchy
  - ▶ Consider as options:
    - PARENT\_LAST / PARENT\_FIRST
    - Application Scoped RES XU resource adapter





# Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:  
[http://www.ibm.com/software/websphere/support/supp\\_tech.html](http://www.ibm.com/software/websphere/support/supp_tech.html)
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:  
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>

# We Want to Hear From You!

## Tell us about what you want to learn

Suggestions for future topics  
Improvements and comments about our webcasts  
We want to hear everything you have to say!

**Please send your suggestions and comments to:**  
[wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com)

# Questions and Answers