

# Simple OSLC RI (SORI)

Jim Conallen and Dave Johnson  
IBM Rational

# Agenda

- Motivation for SORI
- What is SORI
- How to get and build SORI
- SORI architecture
- Demonstration

# Motivation

- Provide minimal reference implementation of the OSLC specifications
- Provide tool for provider and consumer implementations to reference and experiment with

# What is SORI?

- Simple OSLC Reference Implementation
- Written in Java with minimal dependencies
- Apache Software License v2
- Home is <http://sf.net/projects/oslc-tools>




# What SORI is not

- Not a full implementation of OSLC
- Not a full featured ALM tool
- Not a performance benchmark
- Not a framework or SDK

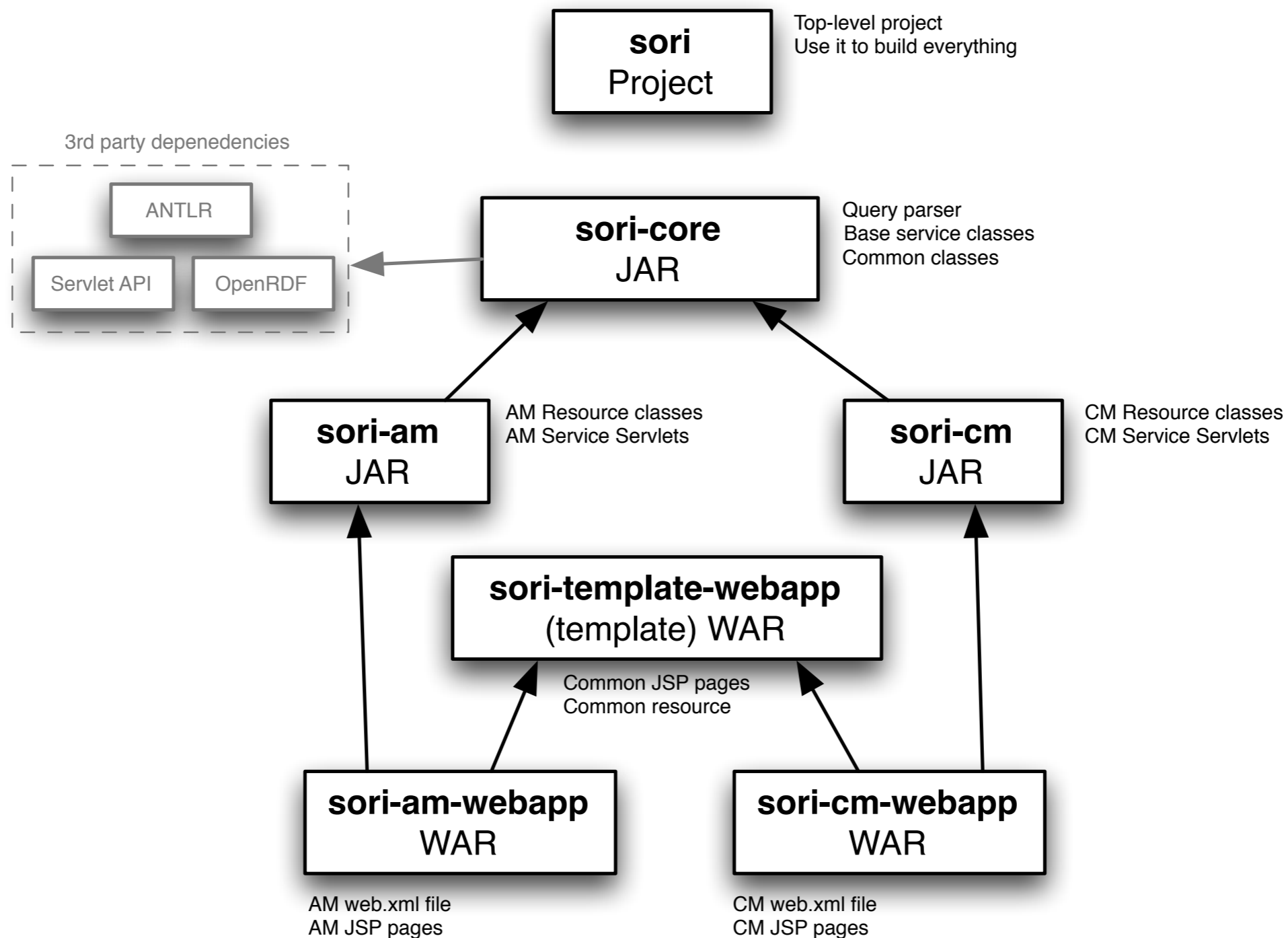
# SORI architecture

- Java EE web application(s)
- Minimal dependencies
  - Java Servlet API and JSP
  - OpenRDF / Sesame RDF parser & store
  - ANTLR parser generator
- Next couple of slides summarize this:
  - <http://open-services.net/bin/view/Main/SimpleOslcRi>

# SORI modules

- ▶  sori (OSLC Main Workspace – ReferenceImpl)
- ▶  sori-am (OSLC Main Workspace – ReferenceImpl)
- ▶  sori-am-webapp (OSLC Main Workspace – ReferenceImpl)
- ▶  sori-cm (OSLC Main Workspace – ReferenceImpl)
- ▶  sori-cm-webapp (OSLC Main Workspace – ReferenceImpl)
- ▶  sori-core (OSLC Main Workspace – ReferenceImpl)
- ▶  sori-template-webapp (OSLC Main Workspace – ReferenceImpl)

# SORI dependencies





# Getting SORI

- Initially:
  - Use Subversion to get code
  - Use Maven to build (and run)
- Later: we'll provide releases

# Build & run via Maven

- You will need: Java JDK 6 and Maven 3
- For example...
- To build and run SORI AM
  - `cd sori`
  - `mvn install`
  - `cd ../sori-am-webapp`
  - `mvn jetty:run-exploded`
- Then navigate to <http://localhost:8080/sori-am>

# Build & run via Tomcat

- Build with Maven
  - `cd sori`
  - `mvn install`
- Find WAR in `sori-am-webapp/target`
- Copy WAR to Tomcat's webapps directory
- Start Tomcat, go to <http://localhost:8080/sori-am>

# Demonstration

- Source and build
- Install and setup
- Service discovery documents
- REST operations
- Using creation factories
- Delegated UIs for selection and creation

# REST Resource operations

- GET a resource
- PUT changes to resource
- DELETE the resource

# Creation Factory

- Create a new Resource through the factory
- Create a new resource from binary

# Simple Query

- Currently incomplete
- OSLC.where

# Delegated UI

- Creator
- Selector