Web Services for Remote Portlets (WSRP)

WSRP Overview

Web Services for Remote Portlets (WSRP) is a web services standard that allows to "plug-n-play" visual, user-facing web services with portals or other intermediary web applications. It allows to create a repository of services that can be reference to surface applications in their portlets or to consume applications from WSRP-compliant Producers.

WSRP Goals

- Enable interactive, presentation-oriented web services to be easily plugged into standards-compliant portals

- Ensure concepts and data exchanged are aligned with other standards in both the portal and web service arenas.

- Make the Internet a marketplace of visual web services, ready to be integrated into portals

- Easy integration of remote portlets, applications, and content into the portal

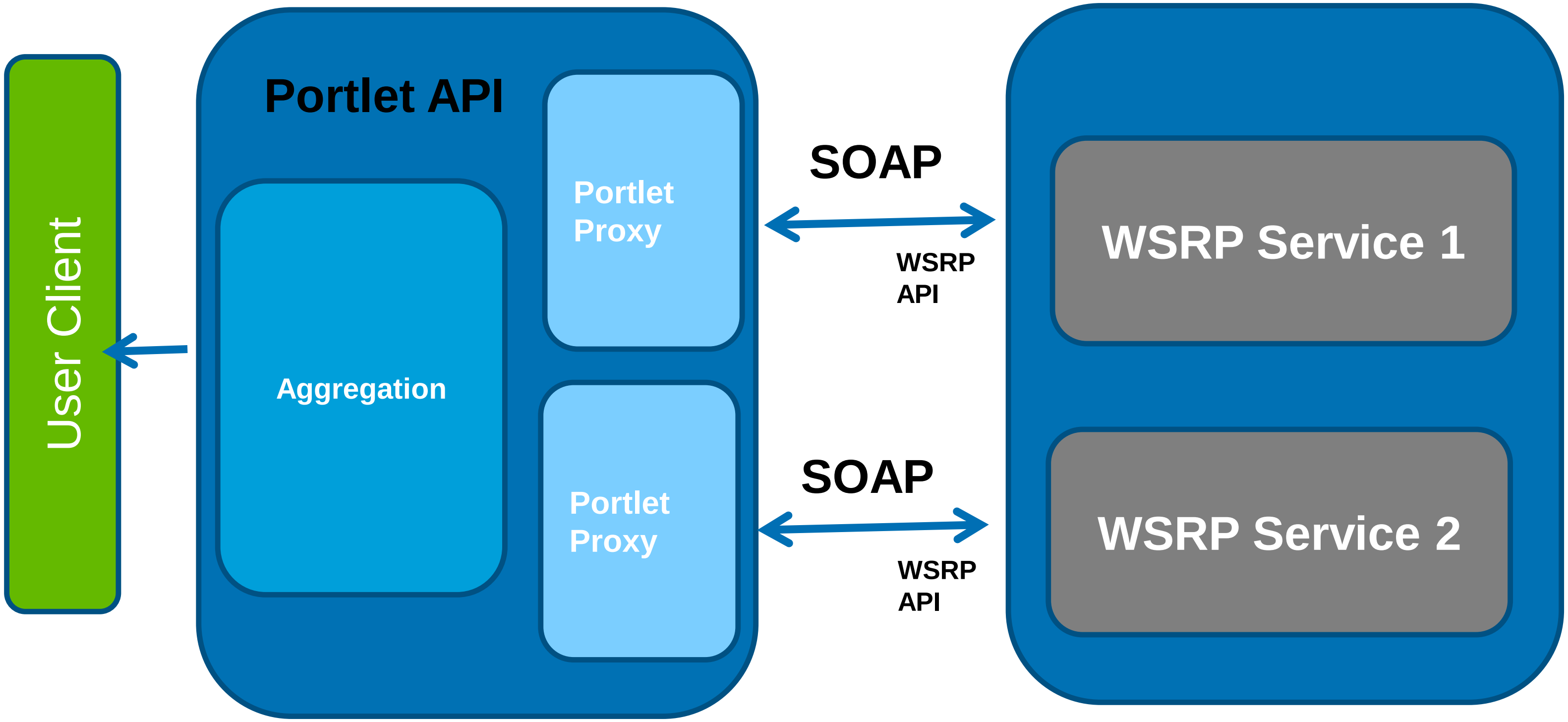
How WSRP works

- WSRP Services are provided by Producers for Consumers

- Producers provide a set of interfaces as defined in the WSRP specification
  - ❖ Service Description
  - ❖ Markup
  - ❖ Portlet Management (optional)
  - ❖ Registration (optional)

- Consumers
  - ❖ Integrates WSRP services
  - ❖ Consumes WSRP services as remote portlets
  - ❖ Receives markup from remote WSRP service and presents it alongside local portlet

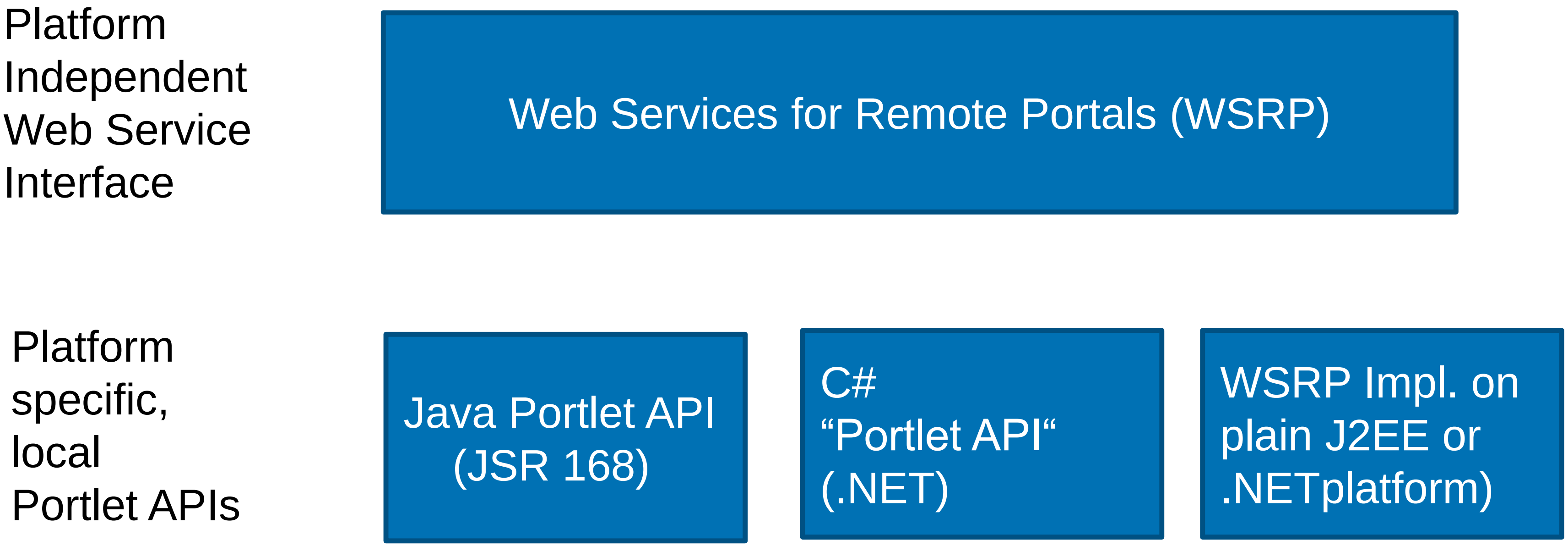
WSRP Architecture



Portals can aggregate presentation from many WSRP services

WSRP services can be aware of portal context

WSRP and Portlet API(s)



Platform Independent Web Service Interface

Web Services for Remote Portals (WSRP)

Platform specific, local Portlet APIs

Java Portlet API (JSR 168)

C# “Portlet API“ (.NET)

WSRP Impl. on plain J2EE or .NETplatform)

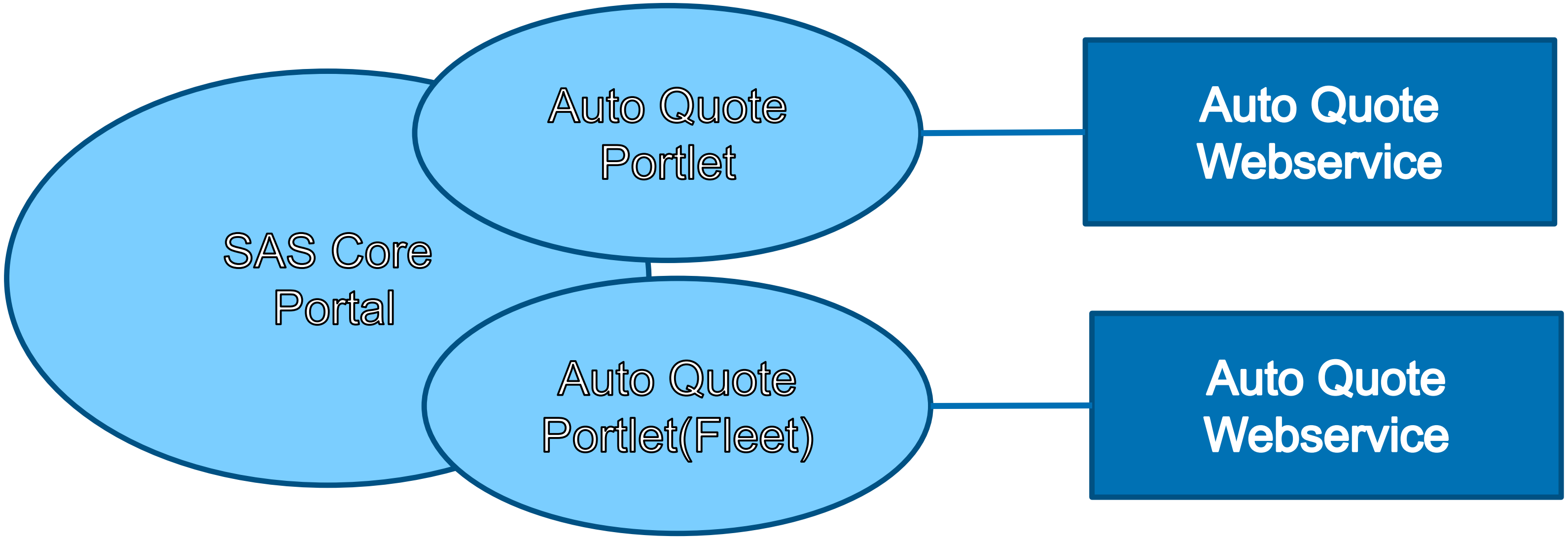
Portlet APIs may be defined for different programming languages;

WSRP can bridge between the different platforms, leveraging platform independence of Web services

Goal:
Portlets written using Portlet API(s) can be published as WSRP services

Web Services vs. WSRP

- Traditional approach
- Web Services define data
- Portlet provides presentation logic

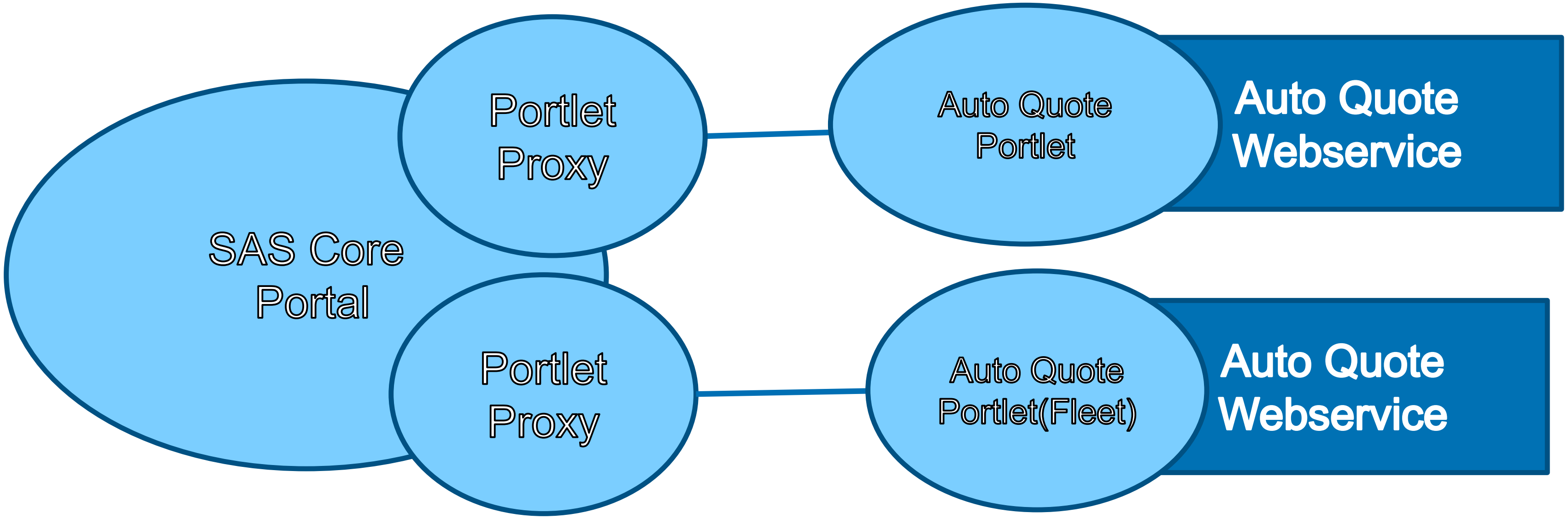


Disadvantages of Traditional approach

- All Portlets must be physically installed on each Portal
- Potentially, multiple instances across a distributed Portal
- Higher admin costs
- Making new portlets is tedious and time consuming
- Presentation layer must be developed for each portlet

WSRP Approach

Web Services for Remote Portlets defines data and presentation logic, built on traditional Web Services



Advantages of WSRP Approach

- One source
- No re-implementation of presentation layer
- No duplication of resources
- Less administrative overload
- Services can be dynamically added to a portal
- Transparent to the administrator
- Behave no differently to local portlets

WebSphere Portal Support for WSRP

WebSphere Portal supports both WSRP Version 1.0 and 2.0 specification

The important task performed by the WSRP components are:

Producer Tasks

Provides local Portlet as a WSRP Service
– ‘Advertises’ via WSRP Service description (WSDL)
– Web Services provided
Withdraws local Portlet as a WSRP Service
Generates group ID’s and handles

Consumer Tasks

Creates (Registers) a Producer
Consumes a WSRP Service

•WSRP configuration is possible via Admin Console or XMLAccess (auto configuration script)

• WSRP supports LTPA or SSL

•WebSphere Portal supports WRSP V 2.0 and can therefore consume all three types of portlets: JSR 168, JSR 286, and IBM portlet API.

WebSite References:

IBM implementation of WSRP

http://infolib.lotus.com/resources/portal/8.0.0/doc/nl_NL/PT800ACD002/admin/wsrpc_plan.html

General Specifications:

http://oasis-open.org/committees/wsrp

Technology for better business outcomes

MPHASIS
an HP company