

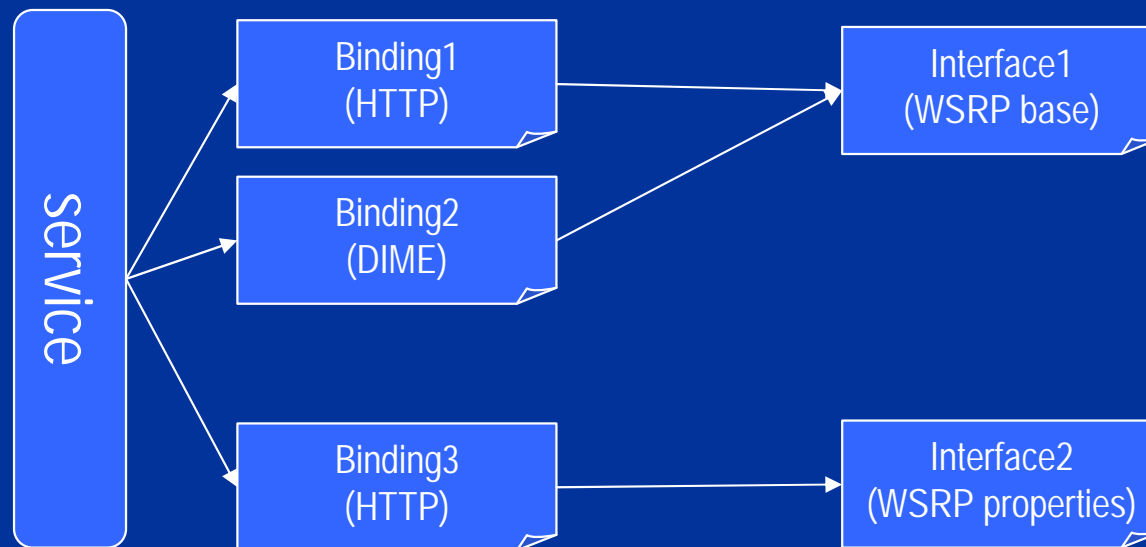
How to create proxies for WSRP

Carsten Leue

IBM Software Group

Producer

- § One WSRP service may implement one or more different interfaces
- § Each interface is defined by an interface WSDL
- § Each binding is defined by a binding WSDL that imports the interface WSDL

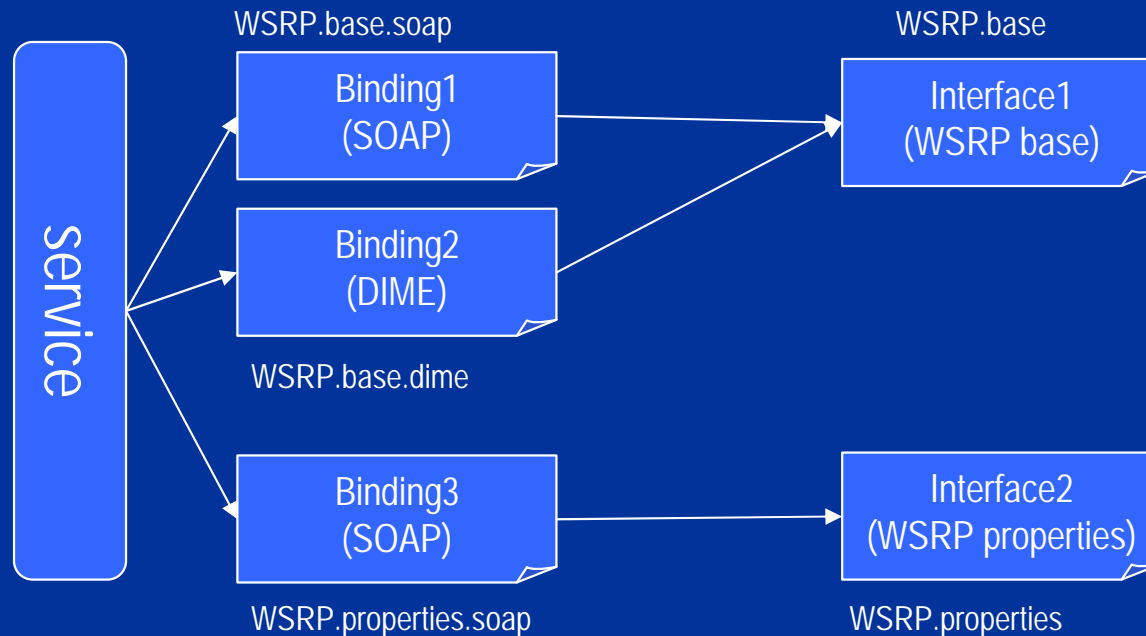


Discovery

- § For each service a consumer can discover all exposed bindings and all exposed interfaces
 - ∅ Via UDDI as tModelInstanceInfos
 - ∅ Via self-description of the service
- § The consumer needs to filter all compatible interfaces and bindings to connect to the producer
- § How can the consumer find out if a binding or interface is the expected one?
 - ∅ By analyzing the WSDL -> tedious and error prone
 - ∅ The WSDL files can be named uniquely so interface and binding can be deferred from the name (next slide)

Naming

- § Define a name that contains the protocol, binding and interface
- § WSRP.<interface>.<binding>
- § The publisher of the WSDL MUST ensure that the content of the WSDL matches the name
- § Naming should include a version number



How a consumer can attach using precompiled proxies

- § The consumer finds out about all bindings and interfaces the service exposes
- § The consumer filters this set by supported interfaces. Per definition the producer must at least expose the WSRP.base interface
- § The consumer filters the resulting set by supported bindings. Per definitions all producers must expose SOAP binding, so the resulting list will contain at least one binding for each interface
- § The consumer sorts the list of bindings per interface by preferred bindings (e.g. DIME > SOAP-Attachments > SOAP)
- § For each interface the consumer selects the best binding

How a consumer can attach using precompiled proxies

- § The consumer instantiates a precompiled WSRP-proxy for each binding/interface and points it to the correct access point
- § In the JAVA case the consumer now instantiates a `java.lang.ref.proxy` object and makes it expose all discovered and supported interfaces
- § The consumer delegates each method invocation to the appropriate WSRP-proxy (e.g. by managing a map between method object and WSRP-proxy)

Object Model

