



Contracts for Services: Needs and Nonsense!

Mark Perreira, Chief Scientist
Talking Blocks





- Web services and WSDL, today.
- Defining a contract-based system.
- The architecture and benefits of a contract-based system.
- Demonstration.





- Business function or capability (**Web Service**)
- Accessed via Internet Protocols (**Web Service**)
- Implementation and protocol neutral, based on open standards for description, discovery & invocation
- Programmatically connects business processes together (heads down, not heads up)
- Typically transactional, requiring integration with existing systems
- Enables information to be queried and manipulated in real-time



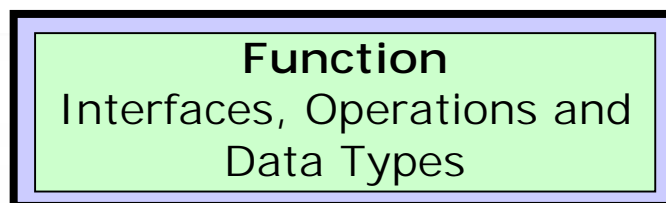
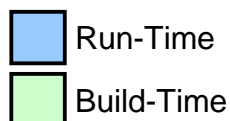


- Web Service Definition Language (WSDL)
- WSDL is the latest in a long history of Interface Definition Languages (IDL):
 - CORBA IDL – implementation neutral, protocol biased.
 - Java Interfaces – implementation open, protocol biased.
 - COM Interfaces – implementation biased, protocol biased.
- WSDL's advantages
 - Implementation neutral AND protocol neutral.
 - Allows for the definition of message bindings.
- WSDL's disadvantages
 - Today, "protocol neutral" generally means SOAP/HTTP.
 - Definition language is complex.

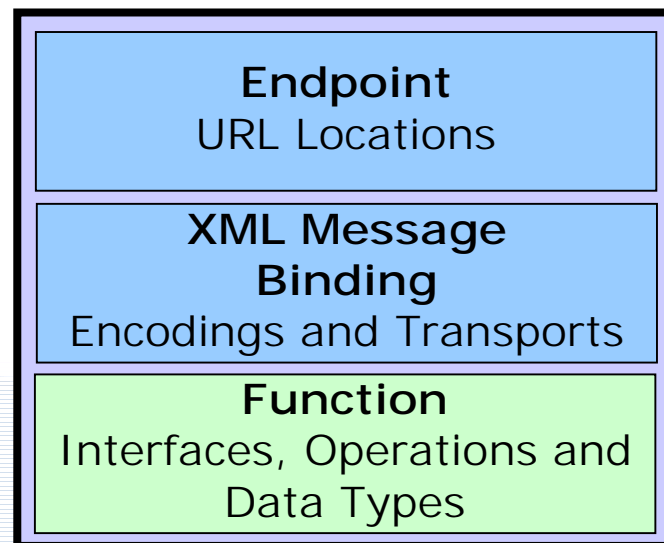




- WSDL mixes build-time and run-time information in the same definition language.



CORBA IDL, Java, COM



WSDL



- Providers of WSDL generally build their functional implementations first, and define them in WSDL second.
 - The IDL That Isn't - <http://www.xml.com/pub/a/2002/01/16/endpoints.html>
- Consumers of WSDL generally generate/create rigid, hard-coded 3GL stubs.
- WSDL defines no role for itself after build-time and is thrown away.
 - WSIF from IBM - <http://www.alphaworks.ibm.com/tech/wsif>
- Building systems the "Microsoft Way".
 - Avoiding DLL Hell: Introducing Application Metadata in the Microsoft .NET Framework - <http://msdn.microsoft.com/msdnmag/issues/1000/metadata/metadata.asp>





● Defining contract:

- A contract is a collection of information that describes accurately and fully a technically binding agreement between two or more interacting systems.

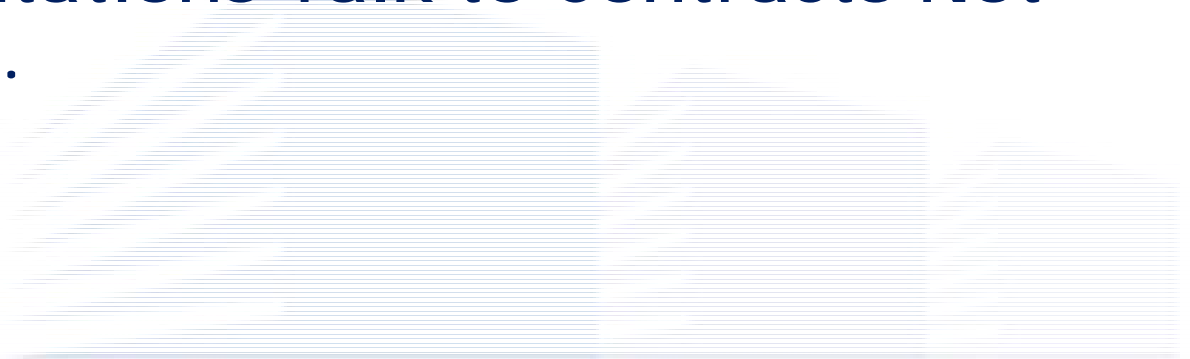
● Information that contracts may contain:

- Name, version and compatibility definition.
- Design descriptions.
- Function definitions.
- Process Sequence and response description/definition.
- Architectural requirements description/definition.
- XML Message binding definitions.
- Binary Message binding definitions.
- Language binding 3GL interfaces.
- Any other supporting description/definition material.

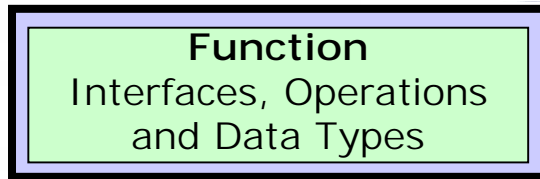
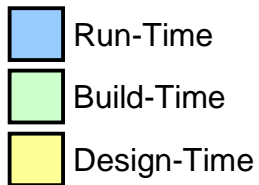




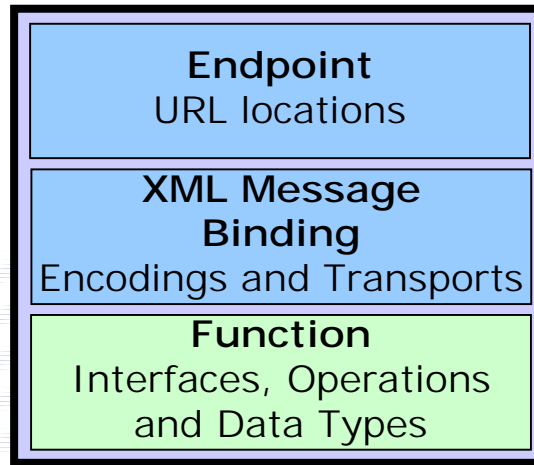
- Contracts come first, implementations come second.
- Contracts are used from design through build-time, deployment and even in the run-time systems.
- Consumers and Providers define system interaction in terms of contracts.
- **Implementations Talk to Contracts Not Endpoints.**



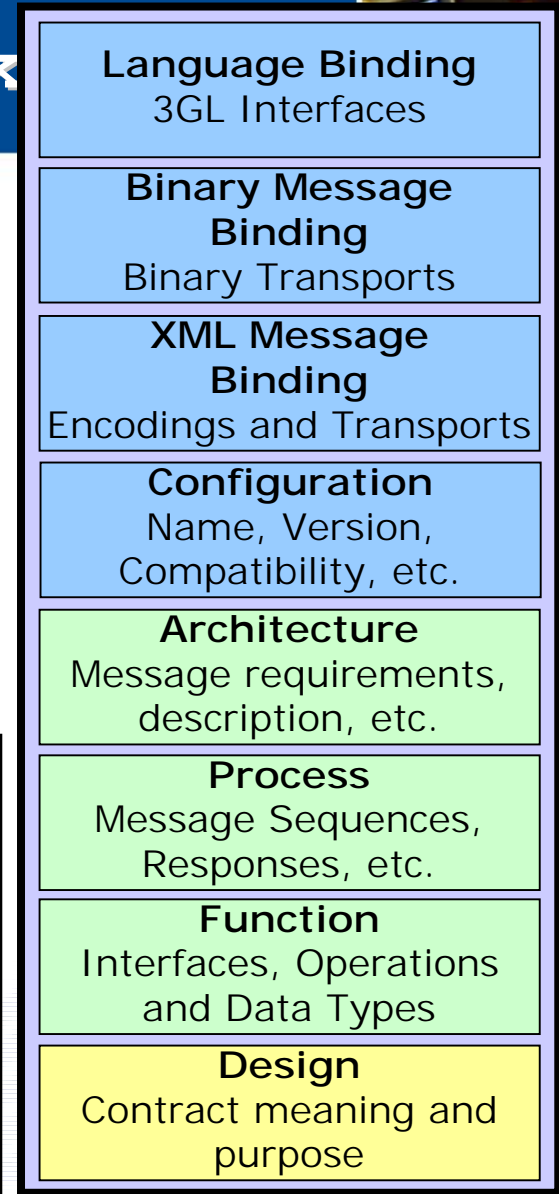
- Contracts can have large collections of information describing a service, but do not contain the service endpoints like WSDL.



CORBA IDL, Java, COM



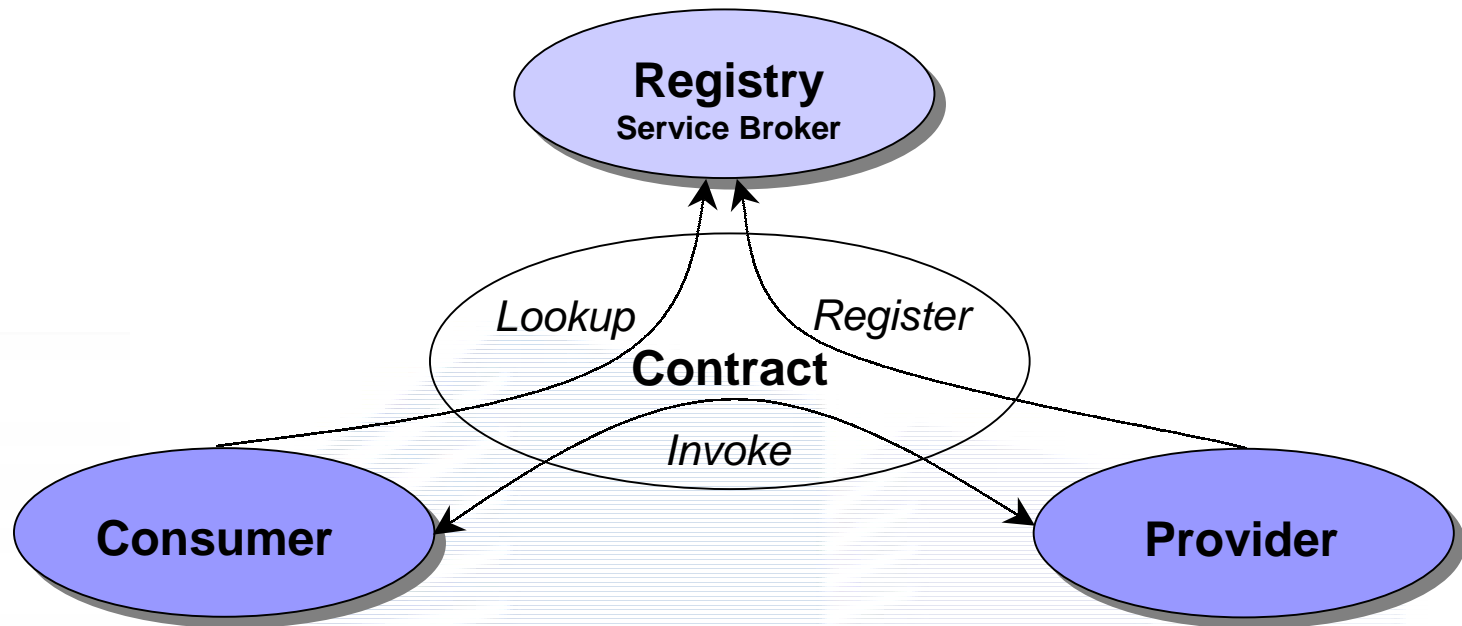
WSDL



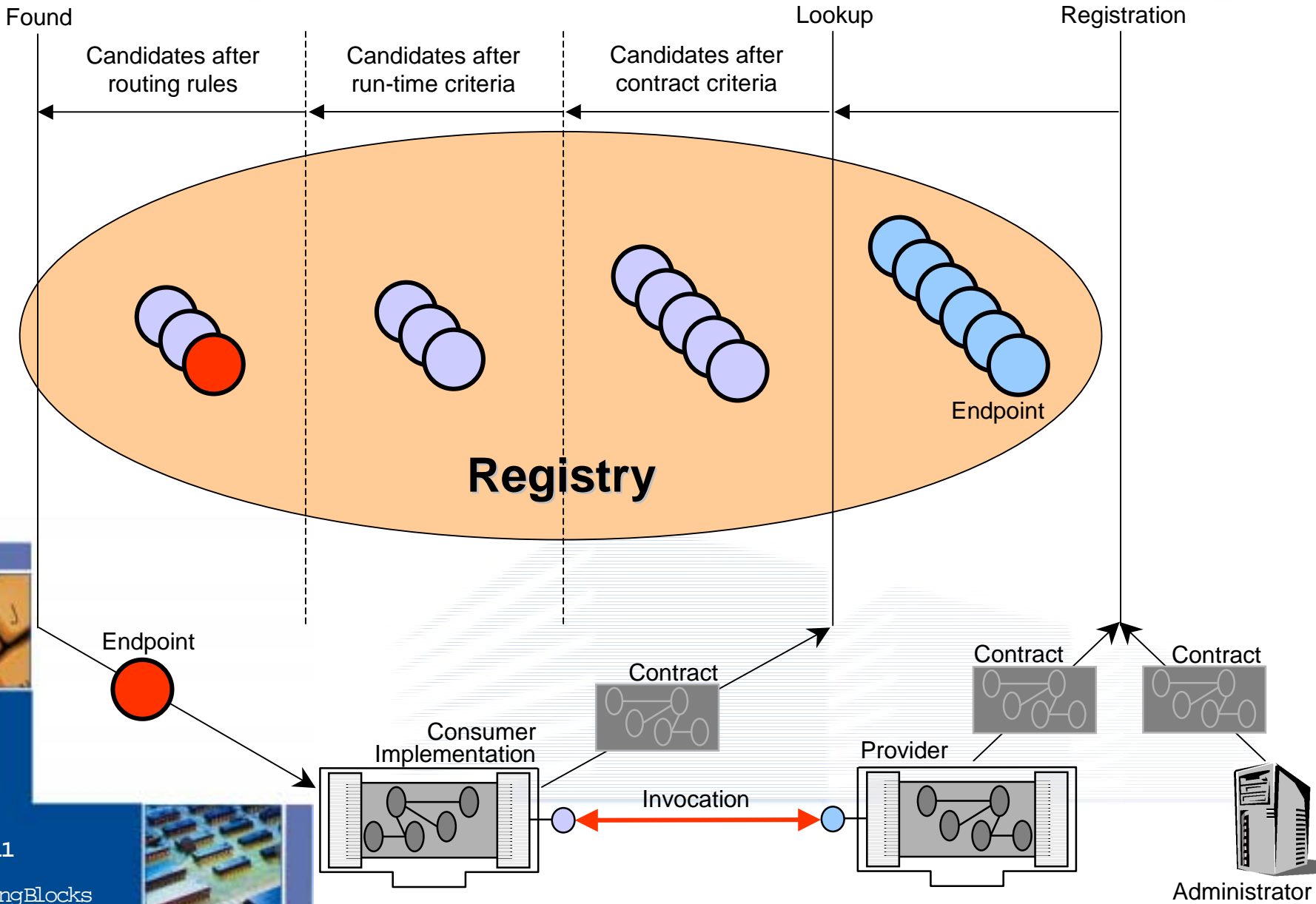
Contract



- The registry plays as an intermediary between the consumer and provider.
- Resolves all interactions at run-time using contract information.



How are contracts found?





- Centralized management of contracts and services.
- Load-balancing and fail-over of contracts independent of service implementation differences.
- Upgraded and side-by-side contract resolution automatically managed by registry.
- Automatic end-point error handling and leasing by registry.
- Equally useful for consumers and providers of services using XML or binary protocols.





Contracts for Services: Needs and Nonsense!

Demonstration

