# 19th ICCRTS

**Title of Paper**

# Enabling service discovery in a federation of systems: WS-Discovery case study

**Topic**

Topic 6: Cyberspace, Communications, and Information Networks

**Name of Authors**

Trude H. Bloebaum and Frank T. Johnsen

Norwegian Defence Research Establishment (FFI)

**Point of Contact**

Trude H. Bloebaum
Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller
Norway

E-mail: Trude-Hafsoe.Bloebaum@ffi.no

This page is intentionally left blank

# Enabling service discovery in a federation of systems: WS-Discovery case study

## Abstract

NATO has identified Web services as the key enabling technology for NATO Network Enabled Capability (NNEC). The technology provides a means to build loosely coupled distributed systems following the principles of Service-Oriented Architecture (SOA). Interoperability is of paramount importance in a federated environment like a coalition network. As a consequence, systems built using the technology should follow the civil interoperability profiles from WS-I as well as the NATO-specific initiatives like the SOA baseline and the Service Interoperability Profiles (SIPs) that are being developed in the TIDE community. The service invocation paradigms (i.e., request/response and publish/subscribe) are mature and well understood. Also, through initiatives like NATO STO/IST-090, it has been shown that the technology can also be used in tactical environments provided certain adaptations are made. However, there are still challenges related to the discovery of services in the tactical environment. In this paper we discuss the importance of service discovery in a federated system. Further, we look at the Web services discovery standards and discuss their suitability for use in a federated tactical network. Finally, we present a prototype proof-of-concept solution for discovery in a federation leveraging the WS-Discovery standard. This work has been performed in the context of NATO STO/IST-118 "SOA recommendations for disadvantaged grids in the tactical domain", the follow-on group to IST-090.

## 1. Introduction

Realizing the NATO Network Enabled Capabilities (NNEC) vision, where information is available to those that require it, independent of where or how they are connected to the network infrastructure, requires advanced mechanisms for information sharing. The NNEC information infrastructure, called NII, builds upon the Service-Oriented Architecture (SOA) principle. This paradigm implies that the functionality is broken down into small, reusable pieces of functionality, known as services. In order for users to get access to the functionality they require they need to be able to know which services are available to them at any given time, and how to use these services. This process is known as service discovery.

In order to achieve the seamless information exchange required by the NNEC vision, information exchange, and thus service discovery, must be available across network boundaries. This means that the service discovery mechanism must be able to interact with each other without the need for manual configuration. In a static environment, one potential approach to service discovery across network boundaries is through the use of service registries, which can be configured to share information using a replication or federation mechanism built into the registries themselves. This method can be used to give access to full metadata descriptions of the services. However, in more dynamic environments services registry federation can be problematic, as the availability of the registry becomes a single point of failure in the network. Because of this problem, services discovery in dynamic networks is better done with a decentralized mechanism. Such decentralized service discovery mechanisms are mostly intended for use within a Local Area Network (LAN), and do not always scale well to larger scenarios. In addition, the difference in capabilities between different network types means that one is likely to encounter scenarios where different distributed services discovery mechanisms are required to work together. This problem has been explored previously, and has been shown to be solvable through the use of gateways [8].

However, the issue of scalability with respect to achieving cross-boundary dynamic service discovery remains unsolved.  In many cases, nodes operating closely together geographically can be connected to different networks, and in order for these nodes to become aware of each other's' services, the services information might have to traverse multiple other networks. Extending the reach of a standardized LAN discovery mechanism across a WAN can be done in multiple ways, but remains a major challenge when attempting to achieve pervasive service discovery across heterogeneous networks.

One example of a scenario in which cross-boundary dynamic service discovery was required was the Coalition for Secure Information Sharing (CONSiS) experiment [15]. In this experiment, two tactical units from two different nations were part of the same coalition network,, and they needed to share information with each other. Both units used the same service discovery mechanism for exposing their services to users outside their own domain, but the mechanism used relied on multicast messages to be transmitted between the two coalition networks. Since these networks were not always directly connected to each other, the service information would have to traverse other networks in order to reach its intended recipients. In the CONSiS experiments this was solved by setting up manually configured multicast tunnels on the routing layer, which allowed multicast messages from one network to reach the other network. This solution works in a small scale scenario, but it has a number of downsides; it requires manual configuration, it does not support selective sharing of services, and it does not scale to larger scenarios.

In this paper we present a federation mechanism for dynamic service discovery which addresses these issues by extending the reach of the locally scoped multicast-based ad hoc mode of WS-Discovery across heterogeneous networks by employing Peer-to-Peer[1] (P2P) techniques. We discuss how this can be achieved on a conceptual level, before looking into one specific implementation - the functional prototype we have implemented in Java.

---

[1] In the survey [1], the authors argue that *Peer-to-Peer (P2P) network overlays provide a good substrate for creating large-scale data sharing, content distribution and application-level multicast applications.* This is because *P2P networks potentially offer an efficient routing architecture that is self-organizing, massively scalable, and robust in the wide-area, combining fault tolerance, load balancing and explicit notion of locality.*

## 2. Background

The SOA reference model [2] states that

> A service is a mechanism to enable access to a set of one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. A service is provided by one entity – the service provider – for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider.
>
> A service is accessed by means of a service interface, where the interface comprises the specifics of how to access the underlying capabilities. There are no constraints on what constitutes the underlying capability or how access is implemented by the service provider. Thus, the service could carry out its described functionality through one or more automated and/or manual processes that themselves could invoke other available services. A service is opaque in that its implementation is typically hidden from the service consumer except for
>
> 1. the information and behavior models exposed through the service interface and
> 2. the information required by service consumers to determine whether a given service is appropriate for their needs.
>
> The consequence of invoking a service is a realization of one or more real world effects (i.e. the actual result of using a service).

In summary, service discovery is the process performed by a service consumer to find (i.e., discover) service providers. As discussed in [7], a service consumer may need to perform service discovery in two phases of its lifetime; at design-time or at run-time.

At design-time, the developer of a software component needs to discover services that provide functionality required by the program. This process could be automatic or manual, depending on the design process. For example, the programmer could search through a registry, or use a Web search engine to discover services with the given functionality. Similarly, helper programs provided by a development tool or middleware may attempt to automatically discover available services and present them as software components available to the developer.

During run-time, the application is responsible for discovering the services it requires, possibly with help from the user. The programmer, or designer, is no longer involved. An example of run-time service discovery is when a printer is discovered on the local network. The application is not designed to use a specific printer, but may work with any service that follows the service definition it was created for during design time.

Simply put, service discovery in the context of Web services is the process of discovering a service endpoint in the form of an URL that points to the address where a service implementing a given service interface is deployed. The complete service definition is represented as a WSDL [9]. One way to find the service definition is to use a registry service, such as Universal Description Discovery and Integration (UDDI) [10] or electronic business using XML (ebXML) [11], to search for services during design-time. At run-time, the discovery process can be simplified by not requiring a full service definition, provided that it can be determined that discovered services implement the same interfaces that were used during design-time.

In tactical networks such as disadvantaged grids, service discovery can be difficult to perform due to the nature of such networks. Examples of this include frequent changes in the network topology when nodes move in and out of each other's' radio ranges.  The network is also likely to become partitioned. Service discovery is especially important in these environments as the set of services that are available to each consumer may change often and repeatedly. If a new service becomes available, the service consumers are unable to use it until they are aware of its existence. Conversely, a service may be discovered, but no longer be reachable due to a network change. Using a central service registry is not a viable option as participants in temporarily isolated areas of the network would be unable to discover each other. An alternative is to use a decentralized discovery protocol. For an approach to service discovery at different operational levels, see our in-depth discussion in [5].

A standardized protocol for decentralized discovery of Web services without a registry is WS-Discovery [12]. WS-Discovery uses multicast to distribute service information, and is further described in [6], where we have evaluated WS-Discovery combined with XML compression as a means to optimize it for use in the tactical domain.

## 3. Design and implementation

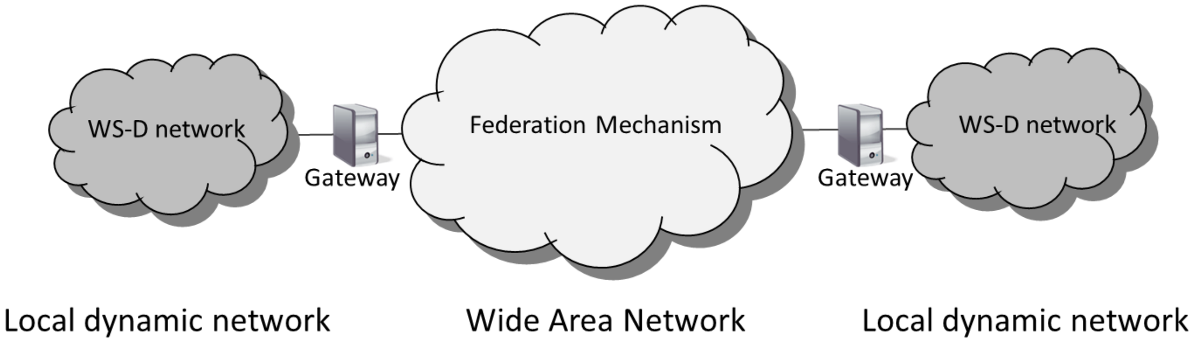

Figure 1: Federation of two independent WS-Discovery (WS-D) domains

The general idea is to extend the reach of a service discovery protocol employed in a LAN across a WAN using a suitably efficient and scalable mechanism. This led us to consider P2P networks for the distribution mechanism, as such overlays can scale to a large number of participating nodes and can function across a WAN. P2P networks support basic overlay network functionality such as message routing, and managing nodes that join or leave the network. Being an overlay, a P2P network introduces its own addressing scheme, creating an abstraction layer above the underlying physical network(s). In addition to the basic functionality involving bootstrapping and maintaining the network, we need a means of efficiently employing one-to-many dissemination of messages. The reason for this is that we may have one (or more) LANs running different service discovery protocols that need to be interconnected across a WAN. If there is a case of heterogeneous discovery protocols being involved, then the discrepancies between the protocols' *expressive power* needs to be mitigated through a metadata repository or through some other means of obtaining the missing information when translating from one protocol to another. This issue has been shown to be solvable through the use of interoperability gateways [8], so we do not pursue that issue further here. It suffices to say that interoperability gateways are a concept than can be combined with the design we are about to describe, if there are different discovery protocols involved. Figure 1 illustrates this concept. For the sake of argument in this paper, we focus only on WS-Discovery [12] which is an OASIS standard for Web services discovery.

Thus, looking at a single node which has the task of extending the reach of WS-Discovery from the LAN across a WAN, we get the conceptual layer model shown in Figure 2 (this is a simplified layered model collapsing some of the layers in the 7 layer OSI model).

| | |
|---|---|
| Application layer | WS-Discovery service descriptions |
| P2P Overlay | Scalable one-to-many message dissemination<br>(includes network maintenance and routing) |
| Network | “Everything over IP” mindset from NATO means that we focus on IP: IP network (IP routing, IP addresses) |
| Physical | Network hardware |

Figure 2: Simplified layered model showing our conceptual approach

The WS-Discovery protocol has been implemented in different frameworks and operating systems, for example it is supported by Apache CXF [13], in the WSO2 ESB [14], and in recent versions of Microsoft Windows (Vista and newer). For our prototype we have chosen to leverage the Java open source implementation[2] of WS-Discovery.

There are many different P2P overlay networks (see e.g., [1] for a recent survey). Our goal is to investigate ways to extend the reach of WS-Discovery, so we have not sought out to evaluate many different P2P overlays in order to identify the overall “best” approach. Rather, we have chosen a solution to employ in our prototype based on the following observations:

- Many P2P approaches are only implemented for simulated environments, and have not been tested for real-world applications. Though interesting from an academic viewpoint, these solutions are of little value when attempting to build a functional prototype. Thus, we need *a solution that exists as a software library*.
- We want *a solution that can scale to a large number of nodes*, and that solves the overlay maintenance issues in an efficient manner.
- The solution *must be able to provide efficient one-to-many communication*, as that forms the basis for our prototype design.

Given the above observations, we found that *Pastry* [3] coupled with SCRIBE [4] provides everything we require from the overlay network: Pastry nodes form a decentralized, self-organizing and fault-tolerant overlay network which provides efficient request routing, deterministic object location, and load balancing in an application-independent manner. Furthermore, Pastry provides mechanisms that support and facilitate application-specific object replication, caching, and fault recovery. Add SCRIBE to Pastry, and you get a generic, scalable and efficient group communication and event notification system providing application level multicast and anycast. Though not perfect (for example, bootstrapping the system requires prior knowledge of one node’s address) this

---

[2] The WS-Discovery library can be obtained from http://code.google.com/p/java-ws-discovery/

combination, as implemented by the open source project *Freepastry*,[3] solves all the important aspects outlined above adequately. Thus, in relation to Figure 2, the P2P overlay constitutes SCRIBE over Pastry for our prototype.

## 4. Results and recommendations

Our prototype has been tested in our lab facilities, and functions to illustrate the proof of concept of our design. We think the approach is viable, and could be refined in an attempt to realize federated service discovery across coalition force networks.

The TIDE community is involved with creating and evaluating service interoperability profiles in context of the next generation Afghan Mission Network, a concept they interchangeably refer to as the *Future Mission Network* or *Federated Mission Network* (FMN for short). We are currently working on a TIDE proposal where we identify WS-Discovery as a service discovery mechanism that can be used in tactical networks where UDDI, the current candidate for service discovery, is insufficient. Along with a sound approach to federating the services across different networks, as well as interoperability with UDDI in backbone networks through the use of gateways, we anticipate that this concept could cover most needs for federated service discovery in a NATO coalition force.

This work is being performed in context of the NATO STO/IST-118 group which focuses on identifying what we call “tactical SOA foundation services” [16]. By this we mean which core enterprise services we need support for in the tactical domain. Examples include the messaging service, publish/subscribe service, and service discovery service. In other words, we aim to investigate how services from the SOA baseline can be extended for use in tactical networks. In this paper we have addressed the service discovery service, as seen by IST-118 (i.e., leverage WS-Discovery (or perhaps other decentralized discovery protocols) rather than UDDI in disadvantaged grids) along with a solution for federation.

## 5. Conclusion and future work

In this paper we have presented our approach to federated service discovery, a concept we think can be realized by employing P2P techniques, interoperability gateways, and existing standards for service discovery. As a proof-of-concept, we have implemented a prototype solving federated service discovery between networks leveraging the WS-Discovery standard. The prototype has been evaluated in our lab.

We plan further experiments in a coalition network context much like the CONSiS experiment that we presented in the introduction to this paper. In this future experiment, we will further evaluate the solution with respect to resilience and bandwidth requirements. Also, we are working on a proposal to the TIDE community involving WS-Discovery in tactical networks. We think it could be a valuable contribution to the future FMN, where it could be combined with UDDI in the backbone for added operational effect.

---

[3] Freepastry is available for download at http://www.freepastry.org/FreePastry/

## References

[1] A. Anitha, J. JayaKumari ,and G.V. Mini, “A survey of P2P overlays in various networks”, in proceedings of the International Conference on Signal Processing, Communication, Computing and Networking Technologies 2011, pages 277-281

[2] OASIS, “Reference Model for Service Oriented Architecture”, Committee draft 1.0, 2006, http://www.oasis-open.org/committees/download.php/16587/wd-soa-rm-cd1ED.pdf

[3] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.

[4] A. Rowstron, A-M. Kermarrec, M. Castro and P. Druschel, "SCRIBE: The design of a large-scale event notification infrastructure", NGC2001, UCL, London, November 2001.

[5] Frank T. Johnsen, Trude Hafsøe and Magnus Skjegstad, “Web Services and Service Discovery in Military Networks”, 14th ICCRTS, Washington, DC, USA, June 15-17, 2009

[6] Frank T. Johnsen and Trude Hafsøe, “Adapting WS-Discovery for use in tactical networks”, 16th ICCRTS, Québec City, Québec, Canada, June 21–23, 2011

[7] Frank T. Johnsen, “Pervasive web services discovery and invocation in military networks”, FFI report 2011/00257, http://rapporter.ffi.no/rapporter/2011/00257.pdf

[8] Frank T. Johnsen, Joakim Flathagen, and Trude Hafsøe, “Pervasive service discovery across heterogeneous networks”, IEEE MILCOM, Boston, MA, USA, 18-21 Oct. 2009

[9] World Wide Web Consortium (W3C), “Web Services Description Language (WSDL) 1.1”, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl

[10] OASIS UDDI Specification TC, “UDDI v3.0.2 specification”, OASIS Standard, http://uddi.org/pubs/uddi_v3.htm

[11] OASIS, “ebXML”, OASIS standard, http://www.ebxml.org/

[12] OASIS, “Web Services Dynamic Discovery (WS-Discovery) Version 1.1”, OASIS Standard, 1 July 2009, http://docs.oasis-open.org/ws-dd/discovery/1.1/wsdd-discovery-1.1-spec.html

[13] Apache Software Foundation, “Apache CXF: An Open-Source Services Framework”, http://cxf.apache.org/

[14] WSO2 Inc., “WSO2 Enterprise Service Bus”, http://wso2.com/products/enterprise-service-bus/

[15] Trude H. Bloebaum and Ketil Lund, “CoNSIS: Demonstration of SOA Interoperability in Heterogeneous Tactical Networks”, 2012 Military Communications and Information Systems Conference (MCC), Gdansk, Poland, 8-9 Oct 2012

[16] F.T. Johnsen, T.H. Bloebaum, P.-P. Meiler, I. Owens, C. Barz, and N. Jansen, “IST-118 – SOA recommendations for Disadvantaged Grids in the Tactical Domain”, 18th ICCRTS, Alexandria, VA, USA, June 19-21, 2013