



# Draft Requirements Document for the Development of the ICE2 Specification

**Published by:**

**The ICE-AG**

**February 2002**

This effort is a sanctioned engagement of the Information and Content Exchange Authoring Group (ICE-AG), which is part of IDEAlliance.

The following dates are relevant to these draft requirements:

- **February 15, 2002.** Comments due from ICE-AG to editor.
- **February 18, 2002.** Comments incorporated by editor and document is readied for release to general public.
- **February 21, 2002.** Release of DRs to general public.
- **March 7, 2002.** General community comments due editor.
- **March 11, 2002.** ICE-AG face-to-face at the Weston Seattle, WA during Open Publish.
- **March 11, 2002.** Re-casting of DRs and the formulation of the Requirements Document for ICE2.

Those contributing to the creation and development of the ICE2 specification to date are as follows:

- [Laird Popkin](#), CTO: 3Path and Chairman of the ICE-AG
- [Dr. Richard D. Martin](#), EVP and CTO: Active Data Exchange, Inc. and Subcommittee Chairman of the ICE2 Specification Development Effort
- Alok Srivastava, Sr. Project Manager: Oracle
- [Dianne Kennedy](#): Founder and Partner, XMLxperts and IDEAlliance
- Bill Kearney: Independent Consultant
- Jay Brodsky: Tribune Media Services
- Jim Menard: Independent Consultant

The editor wishes to thank the above individuals for the many dedicated hours spent in the creation of these requirements.

## **Relationship of ICE to Web Services:**

For the next version of the ICE protocol, it is critical to consider a relationship and/or usage of Web Services related standard for the ICE2 definition. ICE is an application level protocol, related to reliable content delivery process and it does not have to be tied down to the lowest level of transport, repository or description standards. It is valuable for ICE to depend upon the growing Web Services standards from the implementation point of view. If we are able to layer ICE2 on top of different web services standards, then developing an ICE client or finding a listing of

providers will become a familiar and trivial task. Given that ICE is inherently distributed and that Web Services standards are squarely focused on distributed computing space, there is enough synergy between the two sets of standards that can be exploited to make ICE2 more complete and easier to adopt.

There are three major web services standards that ICE2 needs to consider. These are WSDL, SOAP and UDDI. From In the remainder of this document, a discussion of the requirements for each of these standards is presented.

#### WSDL:

WSDL is an XML based description language that currently describes RPC based end-points. This is currently being developed by W3C for extending it to enable messaging style program end-points. ICE currently has an XML based protocol for conversation between client and server. For ICE2, we need to define ICE end-points with WSDL (either message oriented or RPC based or both). Having done that, we will eliminate the need for ICE client packages. Any WSDL to Java or any other programming language based generator will be able to generate ICE client interfaces in that programming language. This will also enable customer applications to embed ICE capabilities within their applications as web services and their web services management infrastructure can manage their client. ICE2 will guide and perform all content related operations and transfers at the application layer.

#### SOAP:

Simple Object Access Protocol is widely being used as transport for web-services related RPC. Most of the current implementations of SOAP only support session-less RPC style invocations but that is changing soon. SOAP is gearing towards the messaging infrastructure. It is the belief that ICE2 should layer its communications on SOAP. This matches the previous discussion on the use of WSDL for "description". ICE2 shall define the characteristics of the communication over SOAP (understanding the current and evolving standards). This definition will help maintain content integrity and guarantees even though SOAP does not provide this type of support. ICE2 will have to consider evolution in HTTP as well as SOAP standards that may cover security, session and other features that ICE may need today. If ICE2 can take advantage of SOAP and keep itself flexible to adopt changes that are going to happen in SOAP space, it allows ICE developers and users to take advantage of their existing communication infrastructure and management services while taking advantage of ICE for their content distribution applications or content subscription activities.

#### UDDI:

There are major problems that ICE content providers face today and that is related with discovery. How people find that there is a server that can deliver content to their ICE client. UDDI solves that problem for web services end-point today. UDDI is fairly open and generic. ICE2 has to recommend a practice and perhaps detail the usage pattern for finding content providers from UDDI. This may simply be a best practices white paper presented to ICE users who want to use UDDI. It should also discuss how users should interpret this information. We may want to utilize publishing-industry-specific ontology to categorize offers and contents that one can subscribe.

Draft Requirement	Description
DR1	<p><b>XML Namespaces:</b></p> <p>The requirement is to eliminate element collisions by moving all ICE-defined elements into one or more ICE namespaces.</p> <p><b>Discussion:</b> XML Namespaces enable us to define a set of unique element names within a given context. Namespaces prevent element collisions and enable computers to unequivocally determine exact points of reference. Such unique addressing is critical to reliable messaging between Web services.</p>
DR2	<p><b>XML Schema:</b></p> <p>Since ICE is a protocol, it requires features such as type definitions found in XML Schemas but not supported by XML DTDs.</p> <p><b>Discussion:</b> XML Schema provides a superset of the specification capabilities of the XML DTD. Only XML Schema enables the specification of type that is expected by Web services. This difference is so critical that the SOAP specification specifically states that a SOAP message "MUST NOT" contain a DTD.</p> <p>Relates to DR9, which also requires XML Schemas for ICE.</p>
DR3	<p><b>HTTPR:</b></p> <p><b>No requirement shall be written</b></p> <p><b>Discussion:</b> Reliable Hypertext Transfer Protocol is a new protocol that offers the reliable delivery of HTTP packets between the server and client. This solves a number of issues that are evident in current HTTP and opens the way to reliable messaging between Web services and improves upon the already message intensive nature of ICE.</p> <p>DR3 has been eliminated as a requirement because ICE continues to be transport agnostic and wants to encourage the employment of transport mapping as needs require. HTTPR mapping will be mentioned in the transport portion of the spec.</p>

DR4	<p><b>ICE-OPEN Catalog:</b></p> <p>The requirement of DR4 is to support an open catalog framework. Relates to DR11 that specifies UDDI compatibility. We will keep this DR open for further discussion.</p> <p><b>Discussion:</b> It has been evident from the beginning that once an installation of ICE is completed, there will evolve several new (unforeseen) opportunities for use in content syndication. The thought here is to provide a referential and voluntary open catalog <u>framework</u>, which can be surfed (browsed) by a human, or searched by an ICE apparatus to recover content of interest to the subscriber.</p> <p>It is anticipated that the ice open catalog would become a universal referential repository by topic (as defined by some metadata or classification construct) that would provide payment, use and other business rules supplied by the syndicator.</p>
DR5	<p><b>Negotiation of non-ice delivery, FTP and simple “get” mechanism</b></p> <p>ICE2 shall be able to handle current and future delivery vehicles, and an apparatus needs to be considered to allow for such delivery.</p> <p>The requirement has been extended to require that the transport discussions be held exploring the following four topics, including their relationship to other DRs.</p> <p><b>DR5A:</b> In-band delivery transport with behavior defined (leads into DR7)  <b>DR5B:</b> Out-of-band delivery transport with behavior defined (leads into DR7)  <b>DR5C:</b> In-band negotiation transport with behavior defined (leads into DR7)  <b>DR5D:</b> Out-of-band negotiation transport with behavior defined (leads into DR7)</p> <p><b>Discussion:</b> A multiplicity of implementations may call for a method of delivering ICE packages outside of the ICE protocol. These may include, but are not limited to: email, FTP (push/pull), traditional mail, or WAP/wireless devices. ICE2 needs to be able to handle current and future delivery vehicles, and an apparatus needs to be considered to allow for such delivery.</p>
DR6	<p><b>Express ICE as a Web Services (WSDL):</b></p> <p>There is a requirement to define the end points of the ICE conversation as WSDL, either message-oriented, RPC-oriented or both, on top of SOAP.</p> <p><b>Discussion:</b> This incorporation of WSDL will enable customer applications to embed ice capabilities within their applications as web services and their web services management infrastructure can manage their client. ICE2 will guide and perform all content related operations and transfers at the application layer.</p>
DR7	<p><b>Modularity of Specification:</b></p> <p>There shall be a requirement to break ICE into modules in a manner that maintains interoperability.</p>

	<p><b>Discussion</b> Divide ICE Specification into a series of modular standards so that implementers can have clearly defined modules to implement so that they can implement as much of ICE as they need in a compliant manner.</p> <p>In doing so, aspects that add complexity to the specification (e.g. scheduling, auditing and logging, lightweight syndicator/subscriber) can be removed from the "core" ICE specification and made clearly optional add-on modules. We must have a well-defined mechanism for determining capabilities for interoperability between implementations.</p> <p>An initial set of modules proposed are:</p> <ul style="list-style-type: none"> <li>▪ ICE/Packaging (see RSS in ICE)</li> <li>▪ ICE/Transport: <ul style="list-style-type: none"> <li>▪ HTTP/ice-payload, HTTP/SOAP, multicast, email, ...</li> </ul> </li> <li>▪ ICE/Subscription Management (Basic Negotiation, Scheduling) <ul style="list-style-type: none"> <li>▪ ICE/Advanced Negotiation</li> <li>▪ ICE/Delivery</li> <li>▪ With Full Updates, Push/Pull, Basic Scheduling, Ping,</li> </ul> </li> </ul> <p>Notification</p> <ul style="list-style-type: none"> <li>▪ ICE/Incremental Updates</li> <li>▪ ICE/Advanced Scheduling</li> <li>▪ ICE/Auditing&amp;Logging</li> <li>▪ ICE/Security: SSL/GPG/Kerberos</li> <li>▪ ICE/Lightweight Syndicator</li> </ul> <p>ICE/Lightweight Subscriber</p>
DR8	<p><b>Multilevel Distribution:</b></p> <p>Standardize mechanisms and data formats to allow tracking of content and its usage through multiple levels of content distribution.</p> <p><b>Discussion:</b> The ICE-AG belief is that support of DOI's (DR10) may be a part of this DR, rather than a separate DR. Also, there are many open requirements issues here. (For example, do I attempt to track modified versions of documents (e.g. I receive an AP story, rewrite it to suit my needs, and syndicate that, do we track it as a new document or a version of the AP article?)</p>
DR9	<p><b>ICE and SOAP</b></p> <p>This requirement has been combined with DR6 and DR11</p> <p><b>Discussion:</b> ICE2 needs to layer its communications on SOAP. This goes along with the use of WSDL for description. ICE2 needs to define the characteristics of the communication over SOAP (understanding the current and evolving standards). This will help maintain content integrity and guarantees even though SOAP does not provide any support for that yet. ICE2 will have to leave room for evolution in HTTP as well as SOAP standards that may cover security, session and other features that ICE may need today. If ICE2 can take advantage of SOAP and keep itself flexible to adopt changes that are going to happen in SOAP space, it allows ICE developers and users to take advantage of their existing communication infrastructure and management services while taking advantage of ICE for their content distribution applications or content subscription activities.</p>

DR10	<p><b>Incorporate the notion of DOI</b></p> <p>ICE must incorporate the notion of ice-item tracking in multi-level distribution. This is only a possible implementation detail of DR8. DR10 will become part of the DR8 discussion.</p>
DR11	<p><b>Web Services with UDDI</b></p> <p>This is combined with DR6 and DR9.</p> <p><b>Discussion:</b> ICE2 has to recommend a practice and perhaps detail the usage pattern for finding content providers from UDDI. This may simply be a best practices white paper presented to ICE users who want to use UDDI. It should discuss how users should interpret this information. We may want to utilize publishing-industry-specific ontology to categorize offers and contents that one can subscribe.</p>
DR12	<p><b>General Comments</b></p> <p>We will recast this as the general requirements preamble for the combined DR6, 11, and 9.</p>
DR13	<p><b>Syndicating Web Services:</b></p> <p>A structure shall exist within the ICE specification to syndicate the use of Web services and the results of these services.</p> <p><b>Discussion:</b> It is also required for ICE to define the layered standard with the help of ICE2 to clearly define a protocol/mechanism for syndicating web services with ICE2 constructs. This is an activity that is critical and very important for syndication users. This set of construct must be layered on top of web services without affecting the core nature of web services but they should clearly define how one can syndicate web services if they chose to after they have implemented web services and exposed them to their customers.</p>
DR14	<p><b>Asynchronous Communication</b></p> <p>ICE must be able to support Asynchronous Communication.</p> <p><b>Discussion:</b> The ability to provide messaging in an asynchronous forms permits a number of different enhancements to the ICE protocol. The ability to deal with wireless devices and transient systems (non-persistence) mandates that Asynchronous Communication elements are considered within the ICE2 effort.</p>
DR15	<p><b>Allow User Transformation Specification (such as language, data format)</b></p> <p>This is not a DR but certainly is a required discussion, best practices recommendation.</p>