

Si2 Registry Specification

Version 2.0-050100

Copyright © 2000 Silicon Integration Initiative, Inc. All rights reserved worldwide.

Introduction

This specification defines the conceptual model and related constraints and semantics for the Si2 Registry of participants. The detailed requirements and problem statements driving this specification are documented in the “ECIX II Requirements”.

Compliance

An implementation that fails to satisfy any constraint identified herein, or in the relevant normative references, with the keywords “required”, “shall”, or “shall not”, is not compliant to the specification. These keywords, along with the keywords “may”, “should”, “should not”, and “optional”, are to be interpreted as described in RFC 2119.

Notation Conventions

This specification uses the following conventions:

Characters in `courier` are literal values to be used in code or messages exactly as shown.

Characters in *times* are not to be used literally but instead describe how values shall be constructed when used in code or messages.

Information contained in the following notation shall be considered non-normative:

(NOTE: This text is informative only and is not considered part of this specification.)

(EXAMPLE: This text is informative only and is not considered part of this specification.)

Characters bounded by double quotes have a special significance referenced as a whole in the sentence, such as a title, value, name, or other identifier.

An identifier bounded by angular brackets refers to a DTD element object. For example, `<value>` refers to the particular DTD element named “value”. If the letter “s” immediately follows the trailing right angle bracket it indicates one or more such elements.

An identifier bounded on the left by the percent character and on the right by the semicolon character refers to a DTD entity object. For example, %xml.att; refers to the DTD entity named “xml.att”.

An identifier bounded by single quotes refers to a DTD attribute object. For example, ‘NoValueReason’ refers to the DTD attribute named “NoValueReason”.

References

Normative

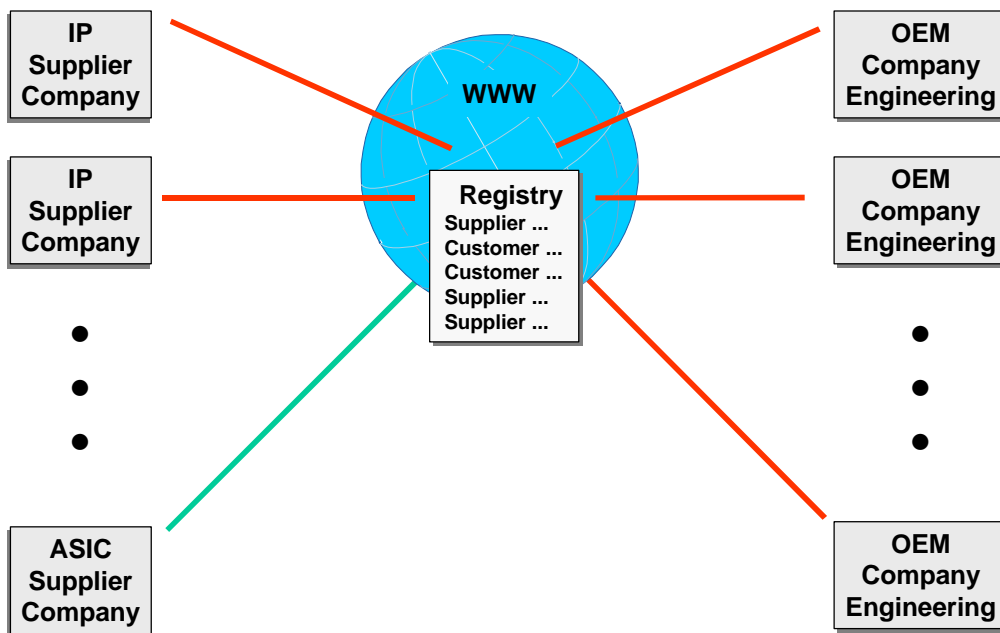
Documents identified as "Normative References" are referenced in part or in their entirety in the current document and the parts referenced shall have normative impact on compliant implementations. For each reference the edition indicated is the authority. All normative documents are subject to revision at different times and by different agencies. Consequently, application of more recent editions of such documents may affect interoperability of implementations and may have additional implications.

Informative

Documents identified as "Informative References" are relevant to the current document, and may assist in comprehension of it, but do not have normative impact on compliant implementations.

Architecture

Figure 1: Automated Connection via Registry



The Registry is a machine-sensible directory of participants capable of sending and/or receiving messages. The

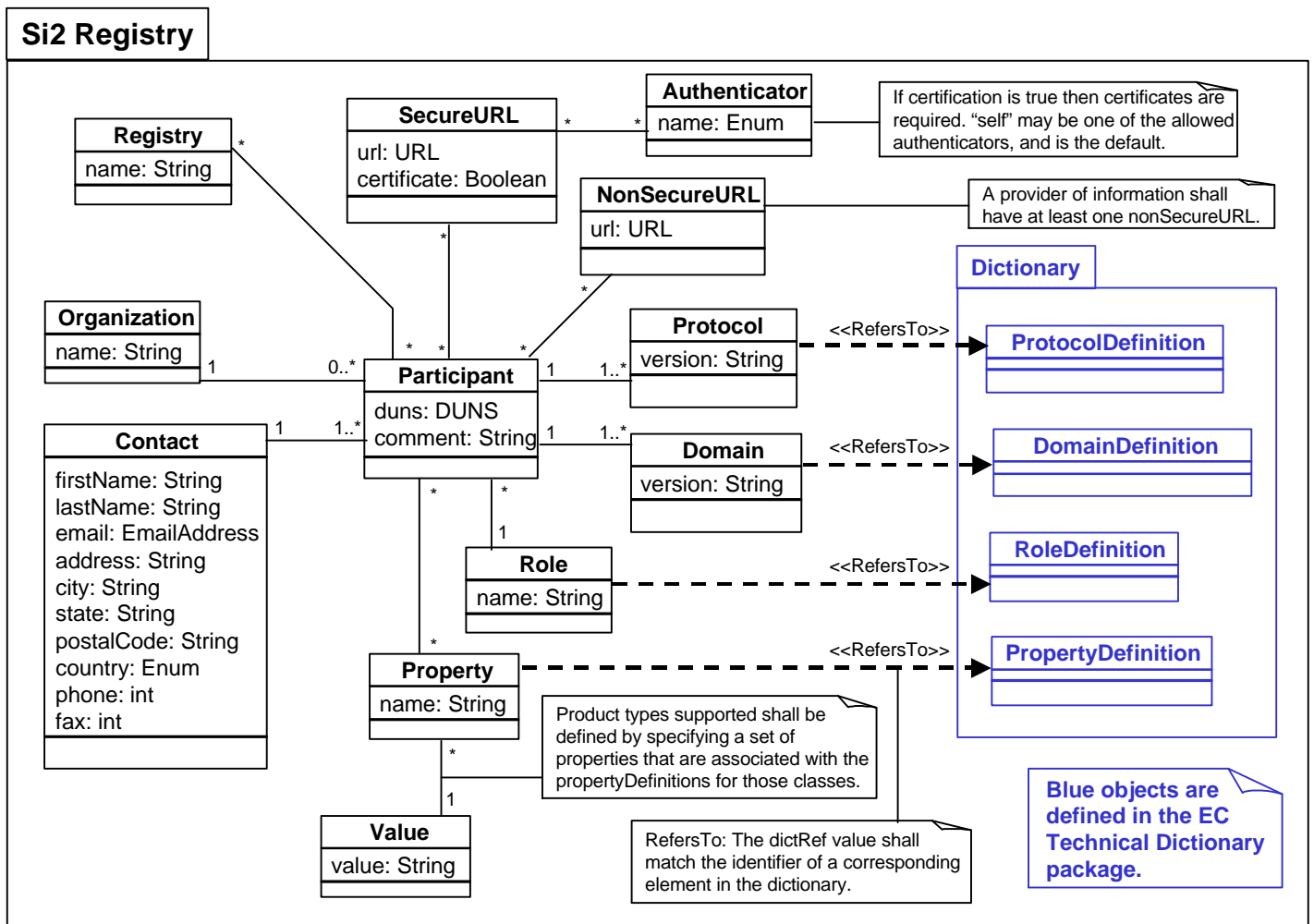
Registry allows senders of messages to locate at any given time all appropriate receivers of those messages without requiring any manual discovery process or human intervention.

The Registry includes information that allows tools to identify the roles of participants (such as sender or receiver), what kinds of messages can be exchanged (ie, the application use domains covered), and communications requirements (including required protocols and their version identification, addresses, and/or security).

Conceptual Model

The UML class diagram in Figures 2 defines the conceptual model for the Registry.

Figure 2: Registry Conceptual Model



Schema and Content

The Si2 Registry consists of a set of participants and information about each participant. This information includes:

- Name (company name)
- DUNS Number
- Contact for reporting technical problems (email, phone, address, etc.)
- Role (supplier, customer, querytool provider, distributor, information provider, etc.)
- Message Protocol version (QuickData, RosettaNet 2A5EC, etc.)
- Application Domain Protocol (eg, QED 1.1, QuickVC 0.9, Electronic Components 2.0, etc.)
- URL (complete address target to which messages are sent)
- Comment (optional)

For example, an instance of the Registry at a point in time is shown in Table 1. Note that this is merely a view of the complete data actually maintained in the Registry repository.

Table 1: Sample Registry Data

Company	DUNS #	Contact	Role	Message	Application	
Silicon Integration Initiative (Si2)	621911718	John Teets teets@si2.org	Query Tool	QD v1.0.1	QED v1.1	...

Table 1 (continued): Sample Registry Data

	URL	Comment
...	http://www.si2.org/ecix/qd/Toolkit	This entry in the Si2 Registry is for the ECIX QuickData Toolkit "customer" side reference implementation. This reference implementation is itself being registered as the first query tool in the official Si2 Registry.

Access to a machine-sensible snapshot of Registry information is available via CGI POST request in accordance with the constraints defined in the "Si2 Registry Maintenance Procedures". The only format for Registry information supported at this time is a snapshot (as of a given point in time) of a view (subset) of the data in the Registry repository, with each record separated by newline (hex 0A) and each field separated by a pound sign (hex 23). The pound sign character is disallowed as content in any field delivered in a snapshot.

The Registry is used to:

- Identify all current participants.
- Bind a role with each participant.
- Bind a particular version of the Message Layer specification with each participant. (This implies a particular version of the DTD and all associated behavioral and semantic constraints.)
- Bind a particular version of the Application Domain Layer specification with each participant. (This implies a particular dictionary and revision level and associated behavioral and semantic constraints.)

The communications protocol address (eg, URL) to which a message may be sent may be obtained from a Registry information instance by the tool that sends the message. This can be the address of any participant whose role is appropriate for receiving queries, and whose protocol and domain are appropriate for the message. Since the Registry is maintained dynamically, any snapshot Registry information instance represents all eligible participants worldwide, at the point in time the snapshot was taken.

Registry Maintenance

Procedures allow add/update/delete of participant records in the master, including some level of validation of compliance in the process. The formal procedures and requirements for access to and change of Registry information are defined by the “Si2 Registry Maintenance Procedures”. Information transfer is accomplished using HTTP CGI POST messages.

Normative References

Requirements

ECIX II Requirements: [requirements.html](#).

Other Specifications

Si2 Registry Maintenance Procedures: <http://www.si2.org/ecix/registryform>.

RFC 2119, “Key words for use in RFCs to Indicate Requirement Levels”:
<http://andrew2.andrew.cmu.edu/rfc/rfc2119.html>

HTTP/1.1 rfc2068: <http://www.w3.org/Protocols/rfc2068/rfc2068>

Common Gateway Interface (CGI) Specification: <http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>

Informative References

SSL V3.0: <http://www17.netscape.com/eng/ssl3>

QuickData Message Specification: [qd_message.doc](#).

QuickData Application: Electronic Components Domain Specification: [qd_application.doc](#).

OMG Unified Modeling Language Specification, Version 1.3, June 1999: <http://www.omg.org/cgi-bin/doc?ad/99-06-08>.