

Network Working Group
ietf-draft-calsch-mod-03.txt
Category: INTERNET DRAFT

J. Noerenberg
Qualcomm, Inc
Expires: May 1998

Internet Calendar Model Specification

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months. Internet-Drafts may be updated, replaced, or made obsolete by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as a "working draft" or "work in progress".

To learn the current status of any Internet-Draft, please check the `1id-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ds.internic.net` (US East Coast), `nic.nordu.net` (Europe), `ftp.isi.edu` (US West Coast), or `munni.oz.au` (Pacific Rim).

Distribution of this document is unlimited.

Abstract

Internet Calendaring and Scheduling protocols require the definition of objects to encapsulate an event to be scheduled, a calendar on which the event will be stored, and means for exchanging these objects across the Internet between calendars on behalf of people for whom the calendars are meaningful. This document gives an abstract model of the objects and the protocols necessary to accomplish this kind of information exchange. It will establish the context in which other Calendaring and Scheduling RFCs can be interpreted. Included are brief introductions to the other components of Internet calendar protocols and definitions of nomenclature common to all documents defining these protocols. Reading this document will enable implementors and users of Internet Calendaring and Scheduling protocols to understand the goals and constraints chosen for related protocols.

Table of Contents

<u>1.</u> Document framework	3
<u>1.1</u> Model Specification	3
<u>1.2</u> iCalendar: Core Object Specification	3
<u>1.3</u> iTIP: Transport Independent Interoperability Protocol	4
<u>1.4</u> iRIP: Binding of iTIP to a session protocol	4
<u>1.5</u> iMIP: Binding of iTIP to E-mail	4
<u>1.6</u> CAP: Calendar Access Protocol Specification	4
<u>2.</u> Abstract Model	5
<u>3.</u> Principal Model Components	6
<u>3.1</u> Calendar User	7
<u>3.2</u> Calendar	7
<u>3.3</u> Calendar User Agent (CUA)	9
<u>3.4</u> Calendar Service	9
<u>3.5</u> Calendar Domain	9
<u>3.6</u> Calendar Access Protocol (CAP)	9
<u>3.7</u> Transport Independent Interoperability Protocol (iTIP)	10
<u>4.</u> Calendar Object Transport	10
<u>4.1</u> Direct Access	12
<u>4.2</u> Calendar Service Mediation	12
<u>4.3</u> Interdomain Exchange	13
<u>4.4</u> Node-Foreign Domain Exchange	13
<u>5.</u> Security considerations	13
<u>6.</u> Copyright	14
<u>7.</u> References	14
<u>8.</u> Acknowledgments	15
<u>9.</u> Author's address	15
<u>10.</u> Appendix -- Calendar protocol nomenclature	16

Noerenberg

Expires May 1998

[Page 2]

Introduction

The Internet Calendar Model specification provides a framework for the set of protocols that define Calendaring and Scheduling for the Internet. The protocols specify the contents of calendars, and how the objects stored on calendars are represented during transmission across the Internet. These protocols also define the interaction between a calendar user agent and a calendar store, as well as the actions performed by calendar transfer agents that facilitate communication between calendar stores. These terms will be defined more precisely in the section on nomenclature below. The protocols are the:

"Core Object Specification" [[iCalendar](#)]

"iCalendar Interoperability Protocol" [[iTIP](#)]

"Calendar Access Protocol" [[CAP](#)]

[1.](#) Document framework

Calendar and Scheduling Protocols are contained in a series of related documents. This section describes the relationship between the documents. [Section 2](#) presents the abstract model for Internet Calendaring and Scheduling.

Following sections amplify the principal concepts defined in the abstract model, provide a schematic representation of information flow in Internet Calendaring and Scheduling, and supply other, useful background information.

[1.1](#) Model Specification

This document - see abstract and introduction above.

[1.2](#) iCalendar: Core Object Specification

The Core Object Specification is the dictionary for Internet Calendaring and Scheduling protocols. It provides the authoritative definition of all properties that may be used in the Internet Calendar and Scheduling protocols as well as the rules for encoding and representing the objects that are constructed from those properties. iCalendar also specifies the method to be used to define new properties.

Because it is a reference, some readers will find it easier to read the

transport protocols first, and use iCalendar to learn details of the objects transmitted by the transport protocols. If any discrepancies exist between the use of properties in iCalendar and other Calendaring and Scheduling protocols, iCalendar's definition always takes

precedence.

The properties of a calendar object can be thought of as attributes. The objects that are defined in iCalendar are referred to as Calendar components. In this document, and others regarding Calendaring and Scheduling, properties and attributes may be regarded as synonymous, and components are equivalent to objects.

1.3 iTIP: Transport Independent Interoperability Protocol

This document specifies how calendaring systems use iCalendar objects to interoperate with other calendar systems. It does so in a general way so as to allow multiple methods of communication between systems.

1.4 iRIP: Binding of iTIP to a Session Protocol

This document specifies a session-layer iTIP protocol. Multiple bindings are possible. This WG will specify and foster implementation of at least one binding.

1.5 iMIP: Binding of iTIP to E-mail

This document specifies an iTIP protocol over Internet e-mail. Internet e-mail protocols are given by RFCs 821, 822, 2045-2049 [[RFC-821](#)] [[RFC-822](#)] [[RFC-2045](#)] [[RFC-2046](#)] [[RFC-2047](#)]. [[RFC-2048](#)] [[RFC-2049](#)]. See the references for details for constructing Internet e-mail messages.

1.6 CAP: Calendar Access Protocol Specification

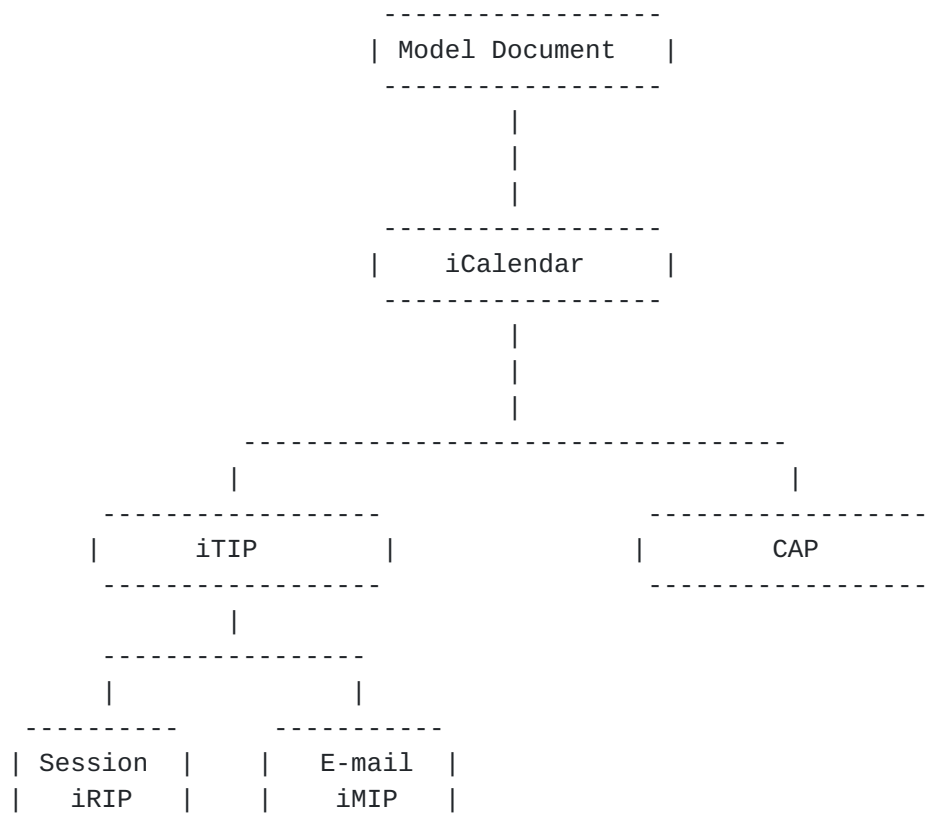
This document specifies how a Calendar User Agent (CUA) will interact with a Calendar Service using iCalendar objects.

Noerenberg

Expires May 1998

[Page 4]

A graphical representation of the relationship between the documents is shown below:



2. Abstract Model

A CALENDAR is a collection of logically related OBJECTs.

OBJECTs include EVENTS, TODOs, JOURNALS, ALARMS, TIMEZONES and FREEBUSYs. OBJECTs are also termed Calendar Components.

Each OBJECT is described using a set of OBJECT PROPERTIES such as date&time, attendees, resources and statuses.

The complete set of OBJECT PROPERTIES and their representation is defined in [[iCalendar](#)].

A specific CALENDAR has a unique CALENDAR IDENTIFIER (CI).

A CALENDAR USER (CU) views and modifies a CALENDAR using a CALENDAR USER AGENT (CUA). Via a CUA a CU can modify a CALENDAR by adding or modifying OBJECTs stored in the CALENDAR. When a CU creates an OBJECT, the CU becomes the OWNER and ORGANIZER for that OBJECT. The CU may delegate OWNER and ORGANIZER properties to other CUs by changing the

OBJECT PROPERTIES and distributing the OBJECT to other CUs.

A CUA uses the services of a CALENDAR SERVICE (CS) via the CALENDAR ACCESS PROTOCOL (CAP) to distribute OBJECTS from and reconcile changes

Noerenberg

Expires May 1998

[Page 5]

to a CALENDAR owned by a CU. A CS delivers messages to a CUA containing OBJECTs from other CALENDAR USERS via CAP. A CUA uses the iCalendar Transport-Independent Interoperability Protocol (iTIP) to deliver OBJECTs to a CS to be distributed to other CUAs.

A CS stores a set of CALENDARs which are accessible according to ACCESS RULES maintained by the CS. CAP enables the CUA and CS to request access to CALENDARs according to the ACCESS RULES. A CUA using CAP MUST NOT use a plaintext password to gain access to a CALENDAR stored by a CS. CAP also enables a CUA and CS to modify the current ACCESS RULES for particular CALENDAR USERS and particular CALENDARs.

iTIP supports the exchange of OBJECTs without the use of ACCESS RULES. It enables the exchange of objects between CSs, as well as between CUAs and CSs. OBJECTs exchanged via iTIP SHOULD be encrypted for the recipients and signed by the sender. A set of CSs may cooperate in a CALENDAR DOMAIN. A CALENDAR DOMAIN appears to be a single CS through iTIP, from the point of view of another CS.

An INTERNET CALENDAR SYSTEM comprises a CUA, a CS and the services of CAP and iTIP, all of which which may, or may not, be integrated together in a given implementation.

NOTE: It is not required that interacting CUA, CS, CAP and iTIP entities be matched implementations, though it is required that all implementations must comply with the specified CAP and iTIP.

A CS is responsible for locating the appropriate CALENDAR DOMAIN for CIs specified in OBJECTs to be transmitted between CALENDAR DOMAINS. A CUA is also required to locate the appropriate CALENDAR DOMAIN in order to use iTIP. When OBJECTs are transmitted, they are encapsulated in a CALENDAR PROTOCOL DATA UNIT (CPDU).

CPDUs are MIME encoded objects that specify a requested action and/or response and carry associated data. Thus, the format of all information exchanged among CALENDAR SYSTEM ENTITIES is defined in terms of MIME CONTENT-TYPES with associated PARAMETER VALUES and MIME BODY PART CONTENT. CONTENT is defined in terms of iCalendar.

The complete set of CPDUs and the corresponding finite state machine for unauthenticated exchange make up iTIP. iTIP is defined in [[iTIP](#)]. iTIP is capable of using a variety of transport mechanisms such as Internet Mail ([[RFC821](#)], [[RFC822](#)]), and session-layer protocols, such as those derived from iTIP ([iMIP], [iRIP]).

ACCESS RULES and the corresponding finite state machine for authenticated exchange make up CAP. CAP is defined in [[CAP](#)].

3. Principal Model Components

There are several principal components in a Calendaring and Scheduling system. Their relationship can be seen in Figure 2 below. This

section identifies some of the salient features of the components. The syntax and semantics for creating and transmitting these objects are completely specified in [[iCalendar](#)], [[CAP](#)], and [[iTIP](#)].

[3.1](#) Calendar User

A calendar user interprets objects on a calendar, creates them, and exchanges them with other calendar users. A calendar user may be a person (Ken Caminiti), a group of people (the San Diego Padres baseball club), or a resource (Qualcomm Stadium). From the point of view of other calendar users, groups and resources appear as a single Calendar user, regardless of their composition in the physical world. Calendar users that are resources generally contain properties that identify them as inanimate objects -- anything from a fruit bowl to rubber bats to settle disputes during a meeting.

A calendar user owns his own calendar, and can manipulate objects stored there via a CUA. Access control properties condition access to calendars and their components and properties.

A calendar user can also manipulate the contents of other calendar users' calendars by sending messages containing calendar objects to them. For example, The San Diego Padres sends calendar events for the **[1997 season to Ken Caminiti, so he knows when to show up at the](#)** ballpark. The Padres sends calendar events for games to be played at home to the Qualcomm Stadium calendar so the concessionaires can order hot dogs.

A calendar user can also organize and own events. When a calendar user creates an event object, that user becomes the organizer and the owner. The organizer can delegate ownership and the role of organizer to others. Only organizers and owners may alter any property of an event object. Calendar users assigned other Attendees roles may not change event properties.

[3.2](#) Calendar

[3.2.1](#) Collection of objects

Calendar users own calendars. This is a one to many relationship. A single calendar user may have multiple calendars. However, each calendar must have a unique identifier. A calendar is an information store containing information about events, to-dos, alarms, journals and free time, the objects stored in it. Within the context of a Calendar, these objects are called components. Also stored in a calendar are properties that are global to all of the objects in the calendar. An example of a global property is the CALSCALE property that identifies the type of calendar year used by objects in a calendar. Global properties such as this establish the context used to interpret the

objects stored in the calendar. The principal structural features of a calendar are described below in [section 3.2.3](#). When objects or properties of a calendar are exchanged between actors in a calendaring and scheduling network (Calendar User Agents and Calendar Services),

they are expressed in the form defined in [[iCalendar](#)]. Figure 1 below is a schematic representation of a calendar.

[3.2.2 Properties](#)

Properties are attributes of an object or a calendar. They consist of a name and a value. Properties are strongly typed. Some properties are multivalued. A property may have parameters that distinguish between related properties. Some properties may occur multiple times in the same object instance, and may be gathered into a logical group. Some properties may be unique to a particular calendar or object.

[3.2.3 Objects](#)

Objects are collections of property values. A particular set of values for the properties of an object define a particular object. Some objects may contain certain other objects. The set of objects in a calendar are identified below.

[3.2.3.1 Events](#)

Event objects are defined for specific starting date-time, have a duration on a calendar, and a description. Other properties of an event may specify a location or other attributes that define the event, such as resources that are part of the event. Events may also contain an Alarm object.

[3.2.3.2 To-do](#)

While like an event, a To-do doesn't reserve a specific block of time on a calendar. A To-do component must have a starting date-time that identifies its first appearance on the calendar. It must also have a date-time that specifies when the To-do expires. A To-do must have a description. It may also contain an alarm object.

[3.2.3.3 Alarms](#)

An alarm object may occur in an Event or To-do. It contains a date-time. When present, and the date-time is passed, it will cause a CUA or CS to notify the user the date-time has passed.

[3.2.3.4 Journal](#)

A journal object is a textual item that is associated with a particular date. As its name suggests, its purpose is to record information meaningful about the date, but not necessarily tied to other calendar objects on a calendar.

[3.2.3.5 Timezone](#)

Timezone objects encapsulate rules for calculating local time from UTC. Including this object in an event object enables a receiver to calculate the universal time value for time values expressed in the sender's local time. This object is especially useful for expressing recurring events whose instances span a change in the time reference

such as the transition between Standard time and Daylight Savings time. Time values expressed in local time are always interpreted in the receiver's local time. The sender must provide another context using

UTC values and Timezone objects if this is not the interpretation intended by the sender.

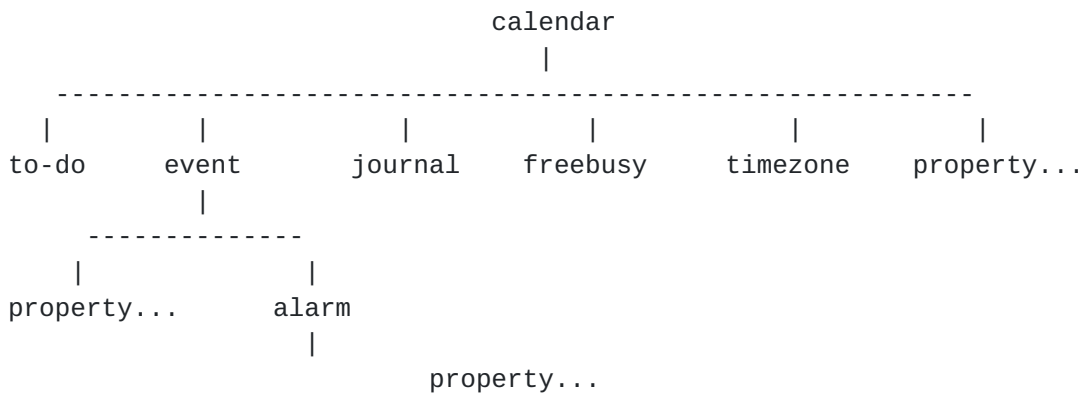


Figure 1: Calendar Object Model

3.3 Calendar User Agent (CUA)

A CUA mediates the interactions between a calendar user and his calendar. It represents the information stored in the calendar to the user, and enables the user to manipulate it. This is a particular instance of the interactive process used by a calendar user.

3.4 Calendar Service

A Calendar Service (CS) stores a collection of calendars and interacts with the Calendar User Agent (CUA) via the Calendar Access Protocol (CAP).

3.5 Calendar Domain

A collection of calendars that can be grouped together constitutes a Calendar Domain. The relation used to bound the group is arbitrary. Frequently membership in an organization will be used to define the domain, but it could be a shared Internet address domain, as well. A Calendar Domain provides a contiguous address space for all the calendars, CSs and CUAs contained in the domain. It must be possible for any Calendar User (via the facilities of a CUA and/or CS) to determine whether they are members of a particular domain, or if other Calendar Users are members. CSs and CUAs can take advantage of domain information when routing event messages.

3.6 Calendar Access Protocol (CAP)

When calendar users need to manipulate calendars that are not stored on the same computer where the CUA executes, the CUA will use the Calendar

Access Protocol to exchange objects with the Calendar Service (CS). CAP specifies the beginning and ending of the session between the CUA and the CS. Using CAP, the CUA will mediate authentication of the user to the service. CAP requires calendar objects and calendar properties

to be expressed in the on-the-wire data format defined in [[iCalendar](#)]. A CUA must not be required to maintain a connection to a CS via CAP in order to display a Calendar for a Calendar User or accept commands from a user to change a Calendar's content. By using CAP a CUA need not have specific information on how a particular CS stores a Calendar and vice versa. This enables specification and exchange of objects and properties independently of Calendar storage models adopted by particular CUAs or CSs.

3.7 Transport Independent Interoperability Protocol (iTIP)

CSs in a domain or across domains exchange objects and properties using iTIP. Like CAP, the objects exchanged with iTIP are iCalendar objects. iTIP defines the beginning and ending of the exchange session, as well the users for whom the messages are intended. iTIP permits unauthenticated delivery of objects to a CS. An CS may accept or reject delivery without interaction with a user. But a CS may require further confirmation of receipt of a object before it is acted upon by the CS.

4. Calendar Object Transport

There are several transport modes in this architecture. The figures below illustrate the different modes that are allowed. Four modes are required to handle the different kinds of calendar exchanges across the Internet, person to person, group interactions local to a particular network, and exchanges between networks, and exchanges between an individual node in one network and the foreign network.

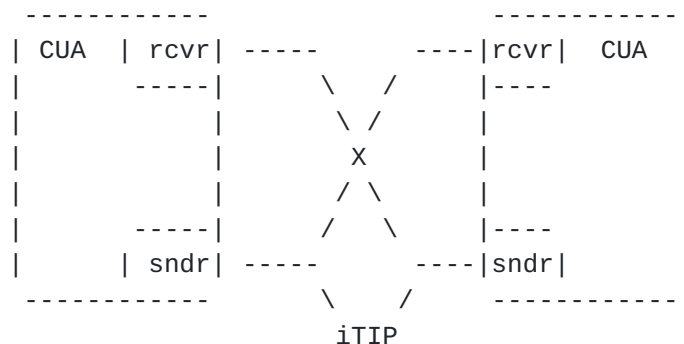


Figure 2: Direct Access

Noerenberg

Expires May 1998

[Page 10]

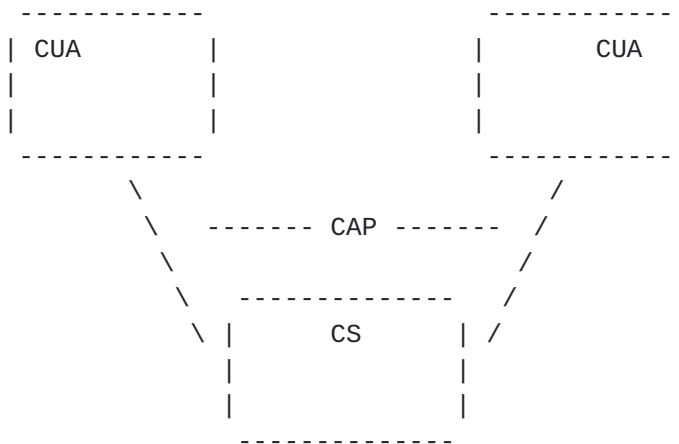


Figure 3: Calendar Service Mediation

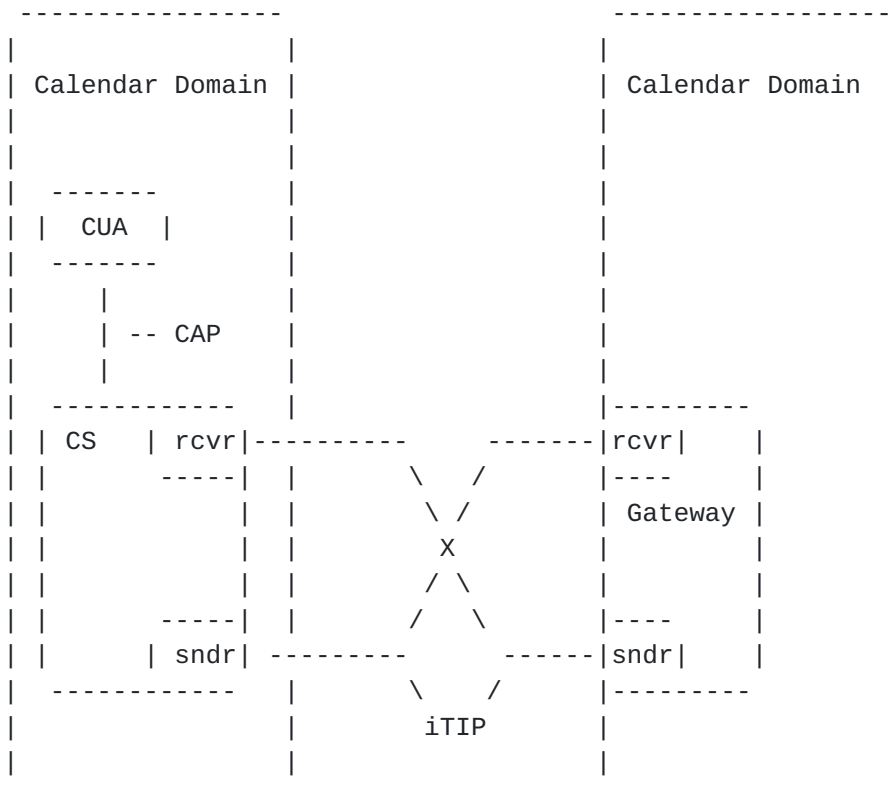


Figure 4: Interdomain Exchange

Noerenberg

Expires May 1998

[Page 11]

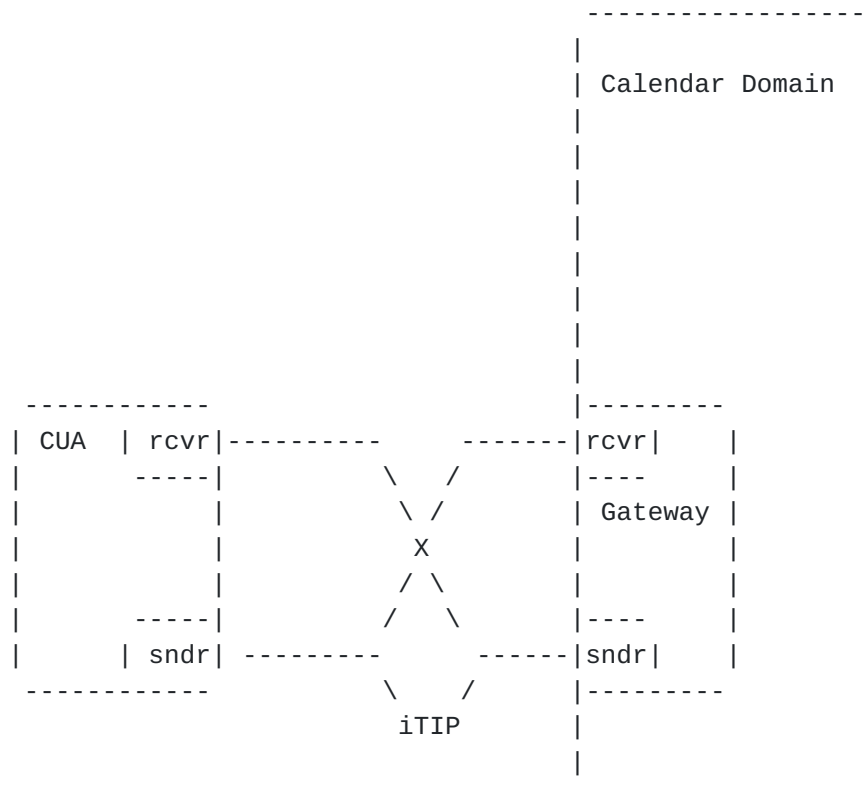


Figure 5: Node-Foreign Network Exchange

[4.1](#) Direct Access

For direct access, two calendar user agents (CUA) exchange calendar components by using iTIP. Each CUA provides an iTIP sender and receiver. As is generally the case, the methods used by the CUA to store calendar data locally are not relevant to the transport model. For this mode, calendar users must be uniquely identifiable across the Internet, and access to CUAs must conform with privacy and security considerations. To insure privacy and/or authenticity, CUAs should use the cryptographic wrappers provided by iTIP.

[4.2](#) Calendar Service Mediation

Using Calendar Service Mediation a CUA interoperates with a Calendar Service (CS) using CAP to exchange calendar components. The CS takes responsibility for mediating receipt and delivery of components with other collaborating CUAs. The principal difference between this mode and Interdomain Exchange is that CSs do not need to use iTIP to facilitate exchange of components. A consequence of this mode is that CUAs and CSs need not use addresses that are unique across the Internet. However, consistency with other modes makes a consistent address model an obvious simplification for implementors. Moreover, because CAP access provides authentication, objects exchanged through a

CAP channel need not carry authentication information.

Noerenberg

Expires May 1998

[Page 12]

4.3 Interdomain Exchange

With Interdomain Exchange a Calendar Service (CS) supporting a group of users in one domain can exchange calendar components with a different calendar domain. Domains may or may not be within the same Internet network domain. Like Direct Access, iTIP is the vehicle which permits component exchange. In the figure, one domain is illustrated with a Calendar Service providing iTIP service. The 2nd domain in this figure has a distinct iTIP sender and receiver. In order to provide end-to-end privacy components must be wrapped in a cryptographically secure wrapper to insure only the intended correspondents can interpret the components. This wrapper is not required unless privacy must be assured. In order to provide backward compatibility with existing calendar and scheduling systems, a privacy wrapper cannot be a required aspect of the component exchange.

4.4 Node-Foreign Domain Exchange

Figure 5 describes the interaction between some particular Calendar Domain, and a node which is not part of that domain. Like Interdomain Exchange and Direct Access, iTIP is used to mediate the exchange of objects between the CUA and the Calendar Domain. Similar to those two modes, objects exchanged in this mode should be enclosed in a cryptographic wrapper to assure the privacy and authenticity of the exchange.

5. Security considerations

There are four classes of threats with which Calendaring and Scheduling protocols must be concerned. These threats can be characterized as Spoofs, Eavesdropping, Floods, and Malicious Changes. A Spoof occurs when a node masquerades as another node enabling it to receive objects to which it would not otherwise be entitled or send unauthorized objects. Eavesdropping occurs when an intermediary node is able to retain and interpret an object exchanged between a sender and receiver without their knowledge or permission. A Flood occurs when a node acts to send more messages to a receiver than the receiver can process, prohibiting the receiver from receiving other messages. Malicious Changes occur if a CU gains access to some Calendar not owned by the CU or some CPDU in transit and makes unauthorized changes to the Calendar or to the CPDU. Simply gaining unauthorized access via the protocols outlined by this model may be considered malicious, as well.

iCalendar must provide the means to classify the intended sensitivity level of an event, to-do or journal calendar component (i.e., PUBLIC, PRIVATE, or CONFIDENTIAL).

CAP must provide a description of the elements of the calendaring system access model and the protocol elements for creating and manipulating the access control to a calendar. This protocol must also describe the authentication procedure between a CUA and CS. This will

mitigate Malicious Changes.

iTIP must provide a means to wrap all components in an exchange in a cryptographically secure envelope so that only the intended correspondents can decode the message. This will mitigate the threats

Noerenberg

Expires May 1998

[Page 13]

of Spoofs and Eavesdropping. It must also provide a means for a receiver to throttle the messages of a sender to prevent Flooding.

6. Copyright

Copyright (C) The Internet Society (Oct 1997). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

7. References

[[RFC-821](#)] Postel, J., "Simple Mail Transfer Protocol", STD 10, [RFC 821](#), USC/Information Sciences Institute, August 1982.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, [RFC 822](#), UDEL, August 1982.

[RFC-1983] Malkin, G., "Internet Users' Glossary", [RFC 1983](#), Aug 1996

[RFC-2026] Bradner, S., "The Internet Standards Process -- Revision 3", [RFC 2026](#), Oct 1996

[RFC-2045] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, Nov 1996

[RFC-2046] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions MIME) Part Two: Media Types", [RFC 2046](#), Nov 1996

[RFC-2047] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", [RFC 2047](#), Nov 1996

[RFC-2048] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", [RFC 2048](#), Nov 1996

[RFC-2049] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, Nov 1996

Noerenberg

Expires May 1998

[Page 14]

[RFC-2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), Mar 1997

[iCalendar] Dawson, F. & Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification" ietf-calsch-ical-01.txt, March, 1997

[iTIP] Silverberg, S., Mansour, S., Dawson, F., & Hopson, R., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", [draft-ietf-calsch-itip-00.txt](#), Jul 1997

[CAP] Author U., "Calendar Access Protocol", unwritten, Oct 1997

8. Acknowledgments

The author is extremely grateful for the assistance, patience and tolerance of the members of the CalSch working group. Their ideas and suggestions are crucial to making this a useful document. In particular, the author would like to thank

Anik Ganguly
Derik Stenerson
Frank Dawson
Gilles Fortin
Einar Stefferud
Steve Mansour
Steve Silverberg

Their comments and ideas were particularly important in the formulation of this draft. I would also like to thank Qualcomm, Incorporated for allowing the time necessary to bring this effort to fruition.

9. Author's address

For more information, please contact the author via Internet Mail.

John W. Noerenberg, II
Qualcomm, Incorporated
6455 Lusk Blvd
San Diego, CA 92131
USA

email: jwn2@qualcomm.com
ph: +1 619 658 3510

Noerenberg

Expires May 1998

[Page 15]

The "Internet Calendar Model Specification" is the work of the Internet Engineering Task Force Working Group on Calendaring and Scheduling. The chairman of that group, Anik Ganguly, may be reached at:

Anik Ganguly
Campbell Services, Inc.
[21700 Northwestern Highway](#)
10th Floor
Southfield, MI 48075
email: anik@ontime.com

[10.](#) Appendix -- Calendar protocol nomenclature

Calendaring and Scheduling uses a rich lexicon of terms that are specific to the problems of scheduling events and reconciling conflicting requests for time and resources. This document will identify the major components of these systems, and show component relationships. However, for the sake of completeness and to serve as an introduction to the protocols in general, a more extensive list of terms, and brief definitions are included here. Essential parts of the system model have expanded definitions in this document where the components of the model are introduced.

[10.1](#) Calendaring lexicon

Alarm, Reminder

An asynchronous mechanism providing feedback for a pending or past event or to-do.

Busy Time

A period of time that is not available for scheduling.

Calendar

A particular collection of calendar objects.

Calendar Component

A Calendar Object.

Calendar Object

The objects that can be found in a calendar. The objects are events, to-dos, journals, free/busies, time zones and alarms. Objects also include properties and may include other objects. A calendar object is identified by unique delimiters.

Calendar Date

A day identified by its position within the calendar year.

Calendar Scale

A particular type of calendar year, for example, Gregorian, Buddhist Era, Japanese Emperor Era, Chinese Lunar, Islamic, and Jewish

Calendars.

Noerenberg

Expires May 1998

[Page 16]

Calendar Service

A Calendar Service (CS) stores a collection of calendars and interacts with the Calendar User Agent (CUA) via the Calendar Access Protocol (CAP).

Calendar User Agent (CUA)

A CUA mediates the interactions between a calendar user and his calendar. It represents the information stored in the calendar to the user, and enables the user to manipulate it. This is a particular instance of the interactive process used by a calendar user.

Coordinate Universal Time (UTC)

The time scale maintained by the Bureau International de l'Heure (International Time Bureau). UTC is often incorrectly referred to as GMT.

Daylight Saving Time (DST)

An adjustment to local time to accommodate annual changes in the number of daylight hours. DST is also known as Advanced Time, Summer Time, or Legal Time. Daylight Saving Time adjustments in the Southern Hemisphere are opposite to those in the Northern Hemisphere.

Event

A calendar object that defines a scheduled activity, minimally specified by a start and end calendar date and time of day and a description.

Free Time

A period of time available for scheduling on a calendar.

FreeBusy

FreeBusy objects describe blocks of allocated and unallocated time on a calendar. They do not contain a description why the time is allocated.

Gregorian Calendar

A calendar scale in general use beginning in 1582. The Gregorian Calendar scale is based on a solar calendar consisting of common years made up of 365 days and leap years made up of 366 days; both divided into 12 sequential months.

Journal

A calendar object that defines a collection of information intended for human presentation and is minimally specified by a calendar date and one or more descriptions.

Local Time

The clock time in public use in a locale. Local time is often referenced by the customary name for the time zone in which it is located. The relationship between local time and UTC is based on the

offset(s) that are in use for a particular time zone. In general, the formula is as follows:

$$\text{local time} = \text{UTC} + (\text{offset})$$

Period

A duration of time, specified as either a defined length of time or by its beginning and end points.

Properties

Attributes that apply to calendar objects or calendars. A calendar object is a named set of properties. Properties can also be used to produce variants to suit a particular purpose.

Recurrence Rule

A notation used to represent repeating occurrences, or the exceptions to such a repetition of an event or a to-do. The recurrence rule can also be used in the specification of a time zone description.

Repeating Event or To-do

An event or to-do that repeats for one or more additional occurrences. The recurrence may be defined with discrete dates and times and/or with a recurrence rule.

Standard Time

Introduced by Sir Sanford Fleming and others around 1870, standard time is a scheme for dividing the world into zones where the same time would be kept. The original proposal was to divide the world into 24 zones, each zone having a width of 15 degrees of longitude. The center zone was originally the meridian passing through Greenwich, England, called Greenwich Mean Time (GMT). The time in the zones was decremented by one hour per zone going westwards and was incremented by one hour per zone going eastwards from GMT. Changes have been made to the original proposal to accommodate political boundaries. In addition, some countries and regions specify 30 or 45 minute offsets, rather than the full 60 minute offset. Standard time is also known as Winter Time in some regions.

GMT and UTC are generally equivalent. However, by international agreement, the GMT term is discouraged in favor of the term UTC for all general time keeping.

Time Zone

A geographic region to which a specified offset from UTC applies. While offsets can frequently be calculated by knowing the longitudinal distance from Greenwich, England, local conventions frequently alter the calculation, complicating the determination of local time. Local convention may also assign a label to identify the time zone. There is no world-wide standard for labels.

To-do

A calendar object that defines an action item and is minimally

specified by an effective calendar date and time of day, a due calendar date and time of day, a priority and a description.

Noerenberg

Expires May 1998

[Page 18]