                      Guide to Internet Calendaring

Status of this Memo

   This memo provides information for the Internet community.  It does
   not specify an Internet standard of any kind.  Distribution of this
   memo is unlimited.

Copyright Notice

Abstract

   This document describes the various Internet calendaring and
   scheduling standards and works in progress, and the relationships
   between them.  Its intent is to provide a context for these
   documents, assist in their understanding, and potentially aid in the
   design of standards-based calendaring and scheduling systems.  The
   standards addressed are RFC 2445 (iCalendar), RFC 2446 (iTIP), and
   RFC 2447 (iMIP).  The work in progress addressed is "Calendar Access
   Protocol" (CAP).  This document also describes issues and problems
   that are not solved by these protocols, and that could be targets for
   future work.

Table of Contents

1. Introduction

   Calendaring and scheduling protocols are intended to aid individuals
   in obtaining calendaring information and scheduling meetings across
   the Internet, to aid organizations in providing calendaring
   information on the Internet, and to provide for organizations looking
   for a calendaring and scheduling solution to deploy internally.

   It is the intent of this document to provide a context for these
   documents, assist in their understanding, and potentially help in the
   design of standards-based calendaring and scheduling systems.

   Problems not solved by these protocols, as well as security issues to
   be kept in mind, are discussed at the end of the document.

1.1 Terminology

   This memo uses much of the same terminology as iCalendar [RFC-2445],
   iTIP [RFC-2446], iMIP [RFC-2447], and [CAP].  The following
   definitions are provided as an introduction; the definitions in the
   protocol specifications themselves should be considered canonical.

Calendar

   A collection of events, to-dos, journal entries, etc.  A calendar
   could be the content of a person or resource's agenda; it could
   also be a collection of data serving a more specialized need.
   Calendars are the basic storage containers for calendaring
   information.

Calendar Access Rights

   A set of rules defining who may perform what operations, such as
   reading or writing information, on a given calendar.

Calendar Service

   A running server application that provides access to a number of
   calendar stores.

Calendar Store (CS)

   A data store of a calendar service.  A calendar service may have
   several calendar stores, and each store may contain several
   calendars, as well as properties and components outside of those
   calendars.

Calendar User (CU)

   An entity (often a human) that accesses calendar information.

Calendar User Agent (CUA)

   Software with which the calendar user communicates with a calendar
   service or local calendar store to access calendar information.

Component

   A piece of calendar data such as an event, a to-do or an alarm.
   Information about components is stored as properties of those
   components.

Delegator

   A calendar user who has assigned his or her participation in a
   scheduled calendar component (e.g.  a VEVENT) to another calendar
   user (sometimes called the delegate or delegatee).  An example of
   a delegator is a busy executive sending an employee to a meeting
   in his or her place.

Delegate

   A calendar user (sometimes called the delegatee) who has been
   assigned to participate in a scheduled calendar component (e.g. a
   VEVENT) in place of one of the attendees in that component
   (sometimes called the delegator).  An example of a delegate is a
   team member sent to a particular meeting.

Designate

   A calendar user authorized to act on behalf of another calendar
   user.  An example of a designate is an assistant scheduling
   meetings for his or her superior.

Local Store

   A CS that is on the same device as the CUA.

Property

   A description of some element of a component, such as a start
   time, title or location.

Remote Store

   A CS that is not on the same device as the CUA.

## 1.2 Concepts and Relationships

   iCalendar is the language used to describe calendar objects.  iTIP
   describes a way to use the iCalendar language to do scheduling.  iMIP
   describes how to do iTIP scheduling via e-mail.  CAP describes a way
   to use the iCalendar language to access a calendar store in real-
   time.

   The relationship between calendaring protocols is similar to that
   between e-mail protocols.  In those terms, iCalendar is analogous to
   RFC 2822, iTIP and iMIP are analogous to the Simple Mail Transfer
   Protocol (SMTP), and CAP is analogous to the Post Office Protocol
   (POP) or Internet Message Access Protocol (IMAP).

## 2. Requirements

## 2.1 Fundamental Needs

   The following scenarios illustrate people and organizations' basic
   calendaring and scheduling needs:

   a] A doctor wishes to keep track of all her appointments.

   Need: To read and manipulate one's own calendar with only one CUA.

   b] A busy musician wants to maintain her schedule with multiple
   devices, such as through an Internet-based agenda and with a PDA.

   Need: To read and manipulate one's own calendar, possibly with
   solutions from different vendors.

   c] A software development team wishes to more effectively schedule
   their time through viewing each other's calendar information.

   Need: To share calendar information between users of the same
   calendar service.

   d] A teacher wants his students to schedule appointments during
   his office hours.

   Need: To schedule calendar events, to-dos and journals with other
   users of the same calendar service.

   e] A movie theater wants to publish its schedule for prospective
   customers.

   Need: To share calendar information with users of other calendar
   services, possibly from a number of different vendors.

   f] A social club wants to schedule calendar entries effectively
   with its members.

   Need: To schedule calendar events and to-dos with users of other
   calendar services, possibly from a number of different vendors.

## 2.2 Protocol Requirements

   Some of these needs can be met by proprietary solutions (a, c, d),
   but others can not (b, e, f).  These latter scenarios show that
   standard protocols are required for accessing information in a
   calendar store and scheduling calendar entries.  In addition, these
   protocols require a common data format for representing calendar
   information.

   These requirements are met by the following protocol specifications.

- Data format: iCalendar [RFC-2445]

iCalendar [RFC-2445] provides a data format for representing
calendar information, to be used and exchanged by other protocols.
iCalendar [RFC-2445] can also be used in other contexts, such as a
drag-and-drop interface, or an export/import feature.  All the
other calendaring protocols depend on iCalendar [RFC-2445], so all
elements of a standards-based calendaring and scheduling systems
will have to be able to interpret iCalendar [RFC-2445].

- Scheduling protocol: iTIP [RFC-2446]

iTIP [RFC-2446] describes the messages used to schedule calendar
events.  Within iTIP messages, events are represented in iCalendar
[RFC-2445] format, and have semantics that identify the message as
being an invitation to a meeting, an acceptance of an invitation,
or the assignment of a task.

iTIP [RFC-2446] messages are used in the scheduling workflow,
where users exchange messages in order to organize things such as
events and to-dos.  CUAs generate and interpret iTIP [RFC-2446]
messages at the direction of the calendar user.  With iTIP [RFC-
2446] users can create, modify, delete, reply to, counter, and
decline counters to the various iCalendar [RFC-2445] components.
Furthermore, users can also request the free/busy time of other
people.

iTIP [RFC-2446] is transport-independent, and has one specified
transport binding: iMIP [RFC-2447] binds iTIP to e-mail.  In
addition [CAP] will provide a real-time binding of iTIP [RFC-
2446], allowing CUAs to perform calendar management and scheduling
over a single connection.

- Calendar management protocol: [CAP]

[CAP] describes the messages used to manage calendars on a
calendar store.  These messages use iCalendar [RFC-2445] to
describe various components such as events and to-dos.  These
messages make it possible to perform iTIP [RFC-2446] operations,
as well as other operations relating to a calendar store such as
searching, creating calendars, specifying calendar properties, and
specifying calendar access rights.

3. Solutions

3.1 Examples

   Returning to the scenarios presented in section 2.1, the calendaring
   protocols can be used in the following ways:

      a] The doctor can use a proprietary CUA with a local store, and
      perhaps use iCalendar [RFC-2445] as a storage mechanism.  This
      would allow her to easily import her data store into another
      application that supports iCalendar [RFC-2445].

      b] The musician who wishes to access her agenda from anywhere can
      use a [CAP]-enabled calendar service accessible over the Internet.
      She can then use any available [CAP] clients to access the data.

      A proprietary system that provides access through a Web-based
      interface could also be employed, but the use of [CAP] would be
      superior in that it would allow the use of third party
      applications, such as PDA synchronization tools.

      c] The development team can use a calendar service which supports
      [CAP], and each member can use a [CAP]-enabled CUA of their
      choice.

      Alternatively, each member could use an iMIP [RFC-2447]-enabled
      CUA, and they could book meetings over e-mail.  This solution has
      the drawback that it is difficult to examine other users' agendas,
      making the organization of meetings more difficult.

      Proprietary solutions are also available, but they require that
      all members use clients by the same vendor, and disallow the use
      of third party applications.

      d] The teacher can set up a calendar service, and have students
      book time through any of the iTIP [RFC-2446] bindings.  [CAP]
      provides real-time access, but could require additional
      configuration.  iMIP [RFC-2447] would be the easiest to configure,
      but may require more e-mail processing.

      If [CAP] access is provided then determining the state of the
      teacher's schedule is straightforward.  If not, this can be
      determined through iTIP [RFC-2446] free/busy requests.  Non-
      standard methods could also be employed, such as serving up
      iCalendar [RFC-2445], HTML, or XML over HTTP.

      A proprietary system could also be used, but would require that
      all students be able to use software from a specific vendor.

e] [CAP] would be preferred for publishing a movie theater's
schedule, since it provides advanced access and search
capabilities.  It also allows easy integration with customers'
calendar systems.

Non-standard methods such as serving data over HTTP could also be
employed, but would be harder to integrate with customers'
systems.

Using a completely proprietary solution would be very difficult,
if not impossible, since it would require every user to install
and use the proprietary software.

f] The social club could distribute meeting information in the
form of iTIP [RFC-2446] messages, sent via e-mail using iMIP
[RFC-2447].  The club could distribute meeting invitations, as
well as a full published agenda.

Alternatively, the club could provide access to a [CAP]-enabled
calendar service.  However, this solution would be more expensive
since it requires the maintenance of a server.

3.2 Systems

   The following diagrams illustrate possible systems and their usage of
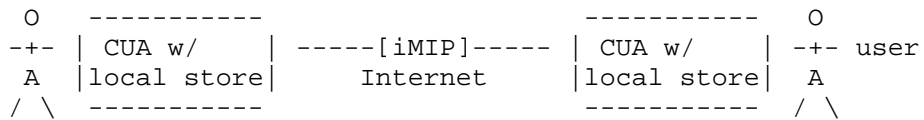   the various protocols.

3.2.1 Standalone Single-user System

   A single user system that does not communicate with other systems
   need not employ any of the protocols.  However, it may use iCalendar
   [RFC-2445] as a data format in some places.

```
        -----------          O
       | CUA w/     |       -+- user
       |local store |        A
        -----------         / \
```
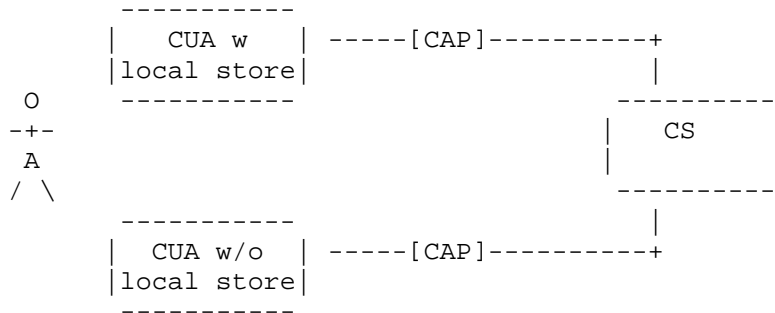
3.2.2 Single-user Systems Communicating

   Users with single-user systems may schedule meetings with each others
   using iTIP [RFC-2446].  The easiest binding of iTIP [RFC-2446] to use
   would be iMIP [RFC-2447], since messages can be held in the users'
   mail queues, which we assume to already exist.  [CAP] could also be
   used.

```
    O   -----------                  -----------   O
   -+- | CUA w/    | -----[iMIP]----- | CUA w/    | -+- user
    A  |local store|     Internet     |local store|  A
   / \  -----------                  -----------   / \
```
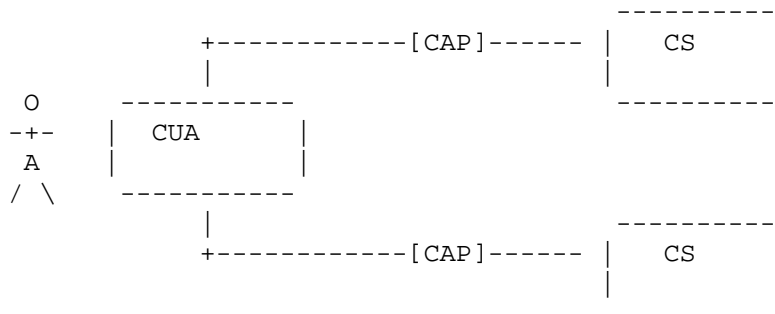
3.2.3 Single-user with Multiple CUAs

   A single user may use more than one CUA to access his or her
   calendar.  The user may use a PDA, a Web client, a PC, or some other
   device, depending on accessibility.  Some of these clients may have
   local stores and others may not.  Those with local stores need to
   synchronize the data on the CUA with the data on the CS.

```
             -----------
            |   CUA w   | -----[CAP]----------+
            |local store|                     |
    O        -----------                    ----------
   -+-                                     |   CS     |
    A                                      |          |
   / \                                      ----------
             -----------                        |
            |   CUA w/o | -----[CAP]----------+
            |local store|
             -----------
```
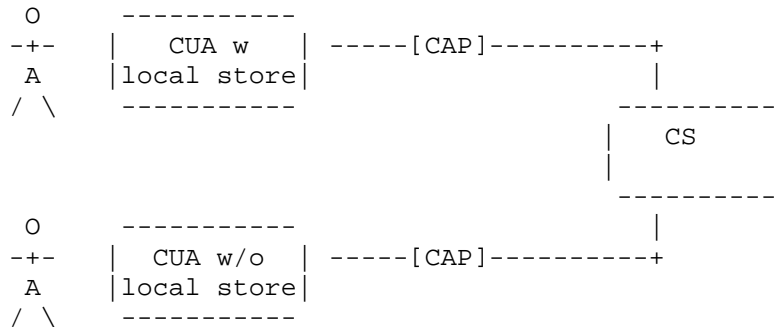
3.2.4 Single-user with Multiple Calendars

   A single user may have many independent calendars; for example, one
   may contain work-related information and another personal
   information.  The CUA may or may not have a local store.  If it does,
   then it needs to synchronize the data of the CUA with the data on
   both of the CS.

```
                                         ----------
            +-----------[CAP]------ |   CS     |
            |                       |          |
    O        -----------             ----------
   -+-      |   CUA     |
    A       |           |
   / \       -----------
            |                           ----------
            +-----------[CAP]------ |   CS     |
                                    |          |
                                     ----------
```
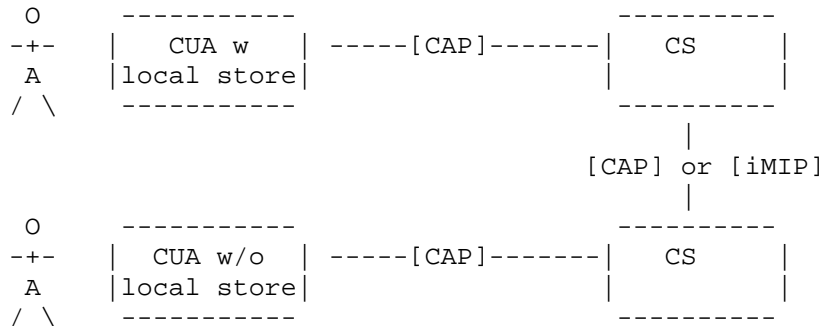
3.2.5 Users Communicating on a Multi-user System

   Users on a multi-user system may schedule meetings with each other
   using [CAP]-enabled CUAs and services.  The CUAs may or may not have
   local stores.  Those with local stores need to synchronize the data
   on the CUAs with the data on the CS.

```
        O      -----------
       -+-    |   CUA w   | -----[CAP]----------+
        A     |local store|                     |
       / \     -----------                     ----------
                                              |   CS     |
                                              |          |
                                               ----------
        O      -----------                        |
       -+-    |  CUA w/o  | -----[CAP]----------+
        A     |local store|
       / \     -----------
```

3.2.6 Users Communicating through Different Multi-user Systems

   Users on a multi-user system may need to schedule meetings with users
   on a different multi-user system.  The services can communicate using
   [CAP] or iMIP [RFC-2447].

```
        O      -----------              ----------
       -+-    |   CUA w   | -----[CAP]-------|   CS     |
        A     |local store|              |          |
       / \     -----------               ----------
                                              |
                                         [CAP] or [iMIP]
                                              |
        O      -----------              ----------
       -+-    |  CUA w/o  | -----[CAP]-------|   CS     |
        A     |local store|              |          |
       / \     -----------               ----------
```

4.  Important Aspects

   There are a number of important aspects of these calendaring
   standards of which people, especially implementers, should be aware.

4.1 Timezones

   The dates and times in components can refer to a specific time zone.
   Time zones can be defined in a central store, or they may be defined
   by a user to fit his or her needs.  All users and applications should
   be aware of time zones and time zone differences.  New time zones may

need to be added, and others removed.  Two different vendors may
describe the same time zone differently (such as by using a different
name).

4.2 Choice of Transport

There are issues to be aware of in choosing between a network
protocol such as [CAP], or a store and forward protocol, such as iMIP
[RFC-2447].

The use of a network ("on-the-wire") mechanism may require some
organizations to make provisions to allow calendaring traffic to
traverse a corporate firewall on the required ports.  Depending on
the organizational culture, this may be a challenging social
exercise.

The use of an email-based mechanism exposes time-sensitive data to
unbounded latency.  Large or heavily utilized mail systems may
experience an unacceptable delay in message receipt.

4.3 Security

See the "Security Considerations" (Section 6) section below.

4.4 Amount of data

In some cases, a component may be very large, for instance, a
component with a very large attachment.  Some applications may be
low-bandwidth or may be limited in the amount of data they can store.
Maximum component size may be set in [CAP].  It can also be
controlled in iMIP [RFC-2447] by restricting the maximum size of the
e-mail that the application can download.

4.5 Recurring Components

In iCAL [RFC-2445], one can specify complex recurrence rules for
VEVENTs, VTODOs, and VJOURNALs.  One must be careful to correctly
interpret these recurrence rules and pay extra attention to being
able to interoperate using them.

5. Open Issues

Many issues are not currently resolved by these protocols, and many
desirable features are not yet provided.  Some of the more prominent
ones are outlined below.

5.1 Scheduling People, not Calendars

   Meetings are scheduled with people; however, people may have many
   calendars, and may store these calendars in many places.  There may
   also be many routes to contact them.  The calendaring protocols do
   not attempt to provide unique access for contacting a given person.
   Instead, 'calendar addresses' are booked, which may be e-mail
   addresses or individual calendars.  It is up to the users themselves
   to orchestrate mechanisms to ensure that the bookings go to the right
   place.

5.2 Administration

   The calendaring protocols do not address the issues of administering
   users and calendars on a calendar service.  This must be handled by
   proprietary mechanisms for each implementation.

5.3 Notification

   People often wish to be notified of upcoming events, new events, or
   changes to existing events.  The calendaring protocols do not attempt
   to address these needs in a real-time system.  Instead, the ability
   to store alarm information on events is provided, which can be used
   to provide client-side notification of upcoming events.  To organize
   notification of new or changed events, clients have to poll the data
   store.

6. Security Considerations

6.1 Access Control

   There has to be reasonable granularity in the configuration options
   for access to data through [CAP], so that what should be released to
   requesters is released, and what shouldn't is not.  Details of
   handling this are described in [CAP].

6.2 Authentication

   Access control must be coupled with a good authentication system, so
   that the right people get the right information.  For [CAP], this
   means requiring authentication before any database access can be
   performed, and checking access rights and authentication credentials
   before releasing information.  [CAP] uses the Simple Authentication
   Security Layer (SASL) for this authentication.  In iMIP [RFC-2447],
   this may present some challenges, as authentication is often not a
   consideration in store-and-forward protocols.

Authentication is also important for scheduling, in that receivers of
scheduling messages should be able to validate the apparent sender.
Since scheduling messages are wrapped in MIME [RFC-2045], signing and
encryption are freely available.  For messages transmitted over mail,
this is the only available alternative.  It is suggested that
developers take care in implementing the security features in iMIP
[RFC-2447], bearing in mind that the concept and need may be foreign
or non-obvious to users, yet essential for the system to function as
they might expect.

The real-time protocols provide for the authentication of users, and
the preservation of that authentication information, allowing for
validation by the receiving end-user or server.

6.3 Using E-mail

Because scheduling information can be transmitted over mail without
any authentication information, e-mail spoofing is extremely easy if
the receiver is not checking for authentication.  It is suggested
that implementers consider requiring authentication as a default,
using mechanisms such as are described in Section 3 of iMIP [RFC-
2447].  The use of e-mail, and the potential for anonymous
connections, means that 'calendar spam' is possible.  Developers
should consider this threat when designing systems, particularly
those that allow for automated request processing.

6.4 Other Issues

The current security context should be obvious to users.  Because the
underlying mechanisms may not be clear to users, efforts to make
clear the current state in the UI should be made.  One example of
this is the 'lock' icon used in some Web browsers during secure
connections.

With both iMIP [RFC-2447] and [CAP], the possibilities of Denial of
Service attacks must be considered.  The ability to flood a calendar
system with bogus requests is likely to be exploited once these
systems become widely deployed, and detection and recovery methods
will need to be considered.

Acknowledgments

Thanks to the following, who have participated in the development of
this document:

   Eric Busboom, Pat Egen, David Madeo, Shawn Packwood, Bruce Kahn,
   Alan Davies, Robb Surridge.

References

    [RFC-2445] Dawson, F. and D. Stenerson, "Internet Calendaring and
               Scheduling Core Object Specification - iCalendar", RFC
               2445, November 1998.

    [RFC-2446] Silverberg, S., Mansour, S., Dawson, F. and R. Hopson,
               "iCalendar Transport-Independent Interoperability Protocol
               (iTIP):  Scheduling Events, Busy Time, To-dos and Journal
               Entries", RFC 2446, November 1998.

    [RFC-2447] Dawson, F., Mansour, S. and S. Silverberg, "iCalendar
               Message-Based Interoperability Protocol - iMIP", RFC 2447,
               November 1998.

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

    [CAP]      Mansour, S., Royer, D., Babics, G., and Hill, P.,
               "Calendar Access Protocol (CAP)", Work in Progress.

Authors' Addresses

   Bob Mahoney
   MIT
   E40-327
   77 Massachusetts Avenue
   Cambridge, MA   02139
   US

   Phone: (617) 253-0774
   EMail: bobmah@mit.edu


   George Babics
   Steltor
   2000 Peel Street
   Montreal, Quebec   H3A 2W5
   CA

   Phone: (514) 733-8500 x4201
   EMail: georgeb@steltor.com


   Alexander Taler

   EMail: alex@0--0.org

Full Copyright Statement

Acknowledgement