

**ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA**

SCUOLA DI INGEGNERIA E ARCHITETTURA
Corso di Laurea in Ingegneria Elettronica, Informatica e Telecomunicazioni

PROTOCOLLI DI GESIONE

CalDAV e CardDAV

Elaborato in
RETI DI TELECOMUNICAZIONI

Tesi di Laurea di
GIACOMO ORLATI

Relatore:
Prof. FRANCO CALLEGATI

Anno Accademico 2014/15
III Sessione di Laurea

Sommario

Introduzione	5
Capitolo 1 Access Protocol CalDAV (RFC 4791)	7
1.1 Modello dati.....	7
1.2 Privilegi di accesso (Access Control)	10
1.3 Sicurezza.....	13
1.4 Sincronizzazione.....	15
1.5 Ricerca	22
1.6 Memorizzazione	24
Capitolo 2 Scheduling protocol CalDAV (RFC 6638).....	27
2.1 Pianificazione delle operazioni.....	27
2.1.1 Identificare risorse oggetto di scheduling	27
2.1.2 Gestione delle risorse oggetto di scheduling.....	28
2.1.5 Obbligare il server a inviare messaggi di scheduling.....	30
2.2 Elaborazione messaggi in arrivo	30
2.2.1 Elaborazione delle risorse, aggiunte e cancellate dall'organizzatore.....	31
2.3 Richiesta di informazioni "BUSY TIME"	32
Capitolo 3 iCalendar (RFC 5545) e vCard (RFC 6350)	33
3.1 Grammatica	34
3.1.1 iCalendar	34
3.2 Creazione oggetti.....	39
3.2.1 Oggetto iCalendar	39
3.2.2 Oggetto vCard	40
Capitolo 4 Conclusioni	41

Ringraziamenti	43
Bibliografia	44

Introduzione

La specifica CalDAV è stata pubblicata nel 2003 come progetto internet, presentato alla Internet Engineering Task Force (IETF) da Lisa Dusseault.

Nel marzo 2007, la specifica CalDAV è stata pubblicata da IETF come RFC4791, scritto da Cyrus Daboo (di Apple Inc), Bernard Desruissaux (Oracle), e Lisa Dusseault (CommerceNet).

CalDAV è stato progettato per l'implementazione con qualsiasi software collaborativo, client o server, che ha bisogno di mantenere sincroni i dati del proprio calendario.

Si è sviluppato come standard aperto per favorire l'interoperabilità tra software differenti. Si estende la specifica WebDAV (basata su HTML protocollo per la manipolazione dei dati) e si utilizza il formato iCalendar per i dati.

Il protocollo di accesso è definito da RFC 4791, esso consente l'accesso di più client alle stesse informazioni consentendo in tal modo la pianificazione cooperativa e la condivisione delle informazioni, mentre è stato definito come RFC 6638 lo scheduling protocol.

L'architettura CalDAV organizza i dati (eventi, attività, note) in directory (collection), dove più elementi (risorse) risiedono.

Le risorse e le collection sono accessibili da uno o più utenti, utilizzando la semantica HTTP e DAV standard per rilevare eventuali modifiche, o per fornire informazioni richiesta dall'utente.

Per il controllo degli accessi viene utilizzato ACL (access control list), in modo che ogni operazione (visualizzare, modificare, cancellare, ecc) può essere negata o concessa per il singolo utente.

Pertanto, la specifica richiede che i server CalDAV supportino "WebDAV Access Control Protocol" altrimenti questa interazione non è possibile.

Le risorse del calendario devono utilizzare il formato iCalendar, che consente al server di comprendere ed elaborare i dati in maniera standardizzata e permettere al server di sostenere un certo numero di operazioni specifiche di calendario.

Grazie a questa funzionalità, un utente può sincronizzare il proprio calendario a un server CalDAV, e condividerlo tra più dispositivi o con altri utenti. Il protocollo supporta anche i calendari non personali, come i calendari per siti o organizzazioni.

La specifica CardDAV è stata proposta come standard IETF con RFC6352 nel mese di agosto 2011 da C. Daboo (di Apple Inc).

CardDAV è un protocollo client / server della rubrica, progettato per consentire agli utenti di accedere ai contatti condivisi su un server e renderli sincroni su più dispositivi differenti.

Anche per CardDAV si estende la specifica WebDAV (basata su HTML protocollo per la manipolazione dei dati) e si utilizza il formato vCard (RFC 6352) per i dati.

Come per CalDAV anche qui occorre un controllo degli accessi tramite ACL (access control list) per gestire l'autenticazione degli utenti.

Capitolo 1

Access Protocol CalDAV (RFC 4791)

1.1 Modello dati

Prima di parlare di CalDAV occorre spiegare il funzionamento del protocollo WebDAV, il cui acronimo sta per Web-based Distributed Authoring and Versioning. Questo protocollo ha lo scopo di rendere il WWW (World Wide Web) un mezzo di comunicazione tra client e server che tramite un set di istruzioni del protocollo HTTP permettono la gestione di file in maniera collaborativa, permettendo la creazione, modifica e lo spostamento di documenti situati su server (tipicamente chiamati Web Server).

Proprietà molto importanti che contraddistinguono il protocollo WebDAV sono un sistema di blocco (ovvero una protezione contro la sovrascrittura dei file), di proprietà (creazione, rimozione e richieste di informazioni sulla data di modifica, ecc), e di collection (creazione, rimozione ed elencazione file simili a delle directory).

Molti sistemi operativi moderni (anche di basi differenti come Windows, Mac o Linux) hanno un supporto integrato per il protocollo WebDAV e con il giusto client ed una rete veloce, la gestione dei file su server WebDAV è facile quasi quanto una normale gestione dei file sul proprio dispositivo.

Il protocollo consiste in un set di nuovi metodi e intestazioni da usare in ambiente HTTP e si è distinto per essere stato il primo protocollo in assoluto ad utilizzare l'XML.

- **PROPFIND** – usato per ottenere proprietà da risorse o strutture delle collection, sotto forma XML.
- **PROPPATCH** – usato per cambiare e cancellare diverse proprietà mediante una singola operazione;

- **MKCOL** – usato per creare delle collection (directory);
- **COPY** – usato per copiare risorse;
- **MOVE** – usato per spostare delle risorse;
- **LOCK** – usato per bloccare una risorsa. WebDAV supporta sia blocchi per gruppi sia blocchi esclusivi;
- **UNLOCK** – rimuove il blocco da una risorsa;
- **SEARCH** – avvia una ricerca sul server.

Da qui nasce CalDAV, un protocollo che permette un accesso ai calendari mediante il WebDAV. Gli elementi dei calendari sono risorse memorizzate nel formato iCalendar (che riprenderemo in maniera dettagliata nel capitolo 3) ed i calendari che contengono gli eventi sono delle normali collezioni del protocollo WebDAV.

Una delle caratteristiche che ha reso WebDAV un protocollo di successo è il suo modello di dati. Questo lo rende un utile quadro di riferimento per le altre applicazioni che gestiscono il calendario.

Questa specifica segue lo stesso schema, sviluppando tutte le caratteristiche basate su un ben descritto modello di dati.

Una breve panoramica: un calendario CalDAV è modellato come una raccolta WebDAV, con una struttura definita; ogni “calendar collections” contiene una serie di risorse che rappresentano oggetti del calendario come la sua risorse figli.

Ogni risorsa che rappresenta un oggetto calendario (evento, una voce di diario o di altri componenti di calendario) è chiamato “calendar object resource”.

Ogni “calendar object resource” e ciascuna raccolta calendario, può essere bloccata singolarmente e avere tutte delle individuali proprietà WebDAV.

Un server CalDAV è un motore di calendario-aware combinato con un WebDAV repository.

Un WebDAV repository è un insieme di collezioni WebDAV, contenente altre risorse WebDAV, all'interno di uno spazio con nomi URL unificati.

Ad esempio, il repository "http://www.prova.it/webdav/" può contenere collezioni WebDAV e risorse, che hanno tutti gli URL inizia con "http://www.prova.it/webdav/". Si noti che la radice URL, "http://www.prova.it/", non può essere di per sé un repository WebDAV (per esempio, se il supporto WebDAV è attuato mediante una servlet o estensione del server Web).

Un repository WebDAV "MAGGIO" include i dati del calendario in alcune parti del suo "namespace" URL, e non dati di calendario in altre parti.

Questo potrebbe significare che il "calendario dati" è diffuso in tutto il repository e mescolato con "NON-calendario dati" in collezioni vicine (ad esempio, dati del calendario possono essere trovati in: /home/giacomo/calendari/ così come in /home/elisa/calendari/ e NON-calendario dati in /home/elisa/contatti/), oppure potrebbe significare che dati del calendario si possono trovare solo in alcune sezioni del repository (ad esempio, /calendar/).

Funzioni di calendario sono richieste solo nelle sezioni repository che sono o contengono risorse oggetto calendario.

Pertanto, un repository può mettere i dati del calendario in /calendar/, avrebbe solo bisogno di supportare le funzionalità richieste CalDAV all'interno di tale raccolta.

Il server CalDAV o il repository è la posizione canonica per dati del calendario e le informazioni di stato.

I clienti possono memorizzare oggetti di calendario in modo offline e tentare la sincronizzazione in un secondo momento, non appena è presente una connessione al server. I clienti possono presentare richieste per modificare i dati o scaricare dati.

Tuttavia, i client devono essere preparati per i dati di calendario sul server per passare da momento della sincronizzazione a quando si cerca un aggiornamento.

I dati di calendario possono essere condivisi e accessibili da molteplici client e tutto questo è reso possibile da Tag che rendono questo possibile.

1.2 Privilegi di accesso (Access Control)

Per supportare i privilegi di accesso, il server CalDAV deve rispettare i requisiti di WebDAV ACL (RFC3744) il quale fornisce un quadro per un insieme estensibile di privilegi che possono essere applicate alle collezioni WebDAV e ordinarie risorse.

Il server CalDAV deve supportare i privilegi: “read – free – busy”.

Le persone che possiedono un calendario spesso desiderano dare la possibilità a utenti di vedere i loro “impegni” senza visualizzare altri dettagli del calendario (ad esempio, posizione, note o i partecipanti ad un determinato evento).

Questo permette un alto livello di privacy pur consentendo ad altri utenti di pianificare riunioni e sapere quando l’utente sarà libero.

I privilegi “CALDAV:read-free-busy” come “calendar collections”, “regular collections” e “calendar object resources” sono esaminati quando viene elaborata una richiesta di “CALDAV:free-busy-query REPORT”.

VEVENT		VFREEBUSY	
TRANSP	STATUS	FBTYPE	
OPAQUE (default)	CONFIRMED (default)	BUSY	
	CANCELLED	FREE	
	TENTATIVE	BUSY-TENTATIVE	
	x-name	BUSY or x-name	
TRANSPARENT	CONFIRMED CANCELLED TENTATIVE x-name	FREE	

Tabella di query

Questo privilegio può essere concesso su “calendar collections”, “regular collections” e “calendar object resources”, e i server devono supportare questo privilegio a tutte.

<!ELEMENT read-free-busy EMPTY>

Il “CALDAV:read-free-busy privilege” deve essere aggregato nel “DAV:Read privilege”. I server devono permettere che il “CALDAV:read-free-busy privilege” sia concesso senza che il “DAV:read privilege” sia concesso.

I client dovrebbero tener presente che quando è stato concesso solo il “CALDAV:read-free-busy privilege” su una risorsa allora l’accesso alle operazioni GET, HEAD, OPTIONS e PROPFIND non è implicito (siccome queste operazioni sono regolate dal “DAV:read privilege”).

```

>> Request <<

REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:free-busy-query xmlns:C="urn:ietf:params:xml:ns:caldav">
  <C:time-range start="20060104T140000Z"
                end="20060105T220000Z"/>
</C:free-busy-query>

>> Response <<

HTTP/1.1 200 OK
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: text/calendar
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
BEGIN:VFREEBUSY
DTSTAMP:20050125T090000Z
DTSTART:20060104T140000Z
DTEND:20060105T220000Z
FREEBUSY;FBTYPE=BUSY-TENTATIVE:20060104T150000Z/PT1H
FREEBUSY:20060104T190000Z/PT1H
END:VFREEBUSY
END:VCALENDAR

```

Esempio pratico di “Richiesta” e “Risposta” processata tramite protocollo DalDAV

Nell’interazione Client-to-client sono ammesse solo un certo numero di operazioni (affinché il server non restituisca degli errori), ma può degradare l’interoperabilità con altre implementazioni client che vogliono accedere allo stesso dato.

Ad esempio, una regola di ricorrenza potrebbe essere sostituita con una serie di date ricorrenti, un singolo evento ricorrente potrebbe essere sostituiti con un insieme di risorse indipendenti per rappresentare ogni ricorrenza, o l’inizio/fine dei valori di tempo possono essere tradotti dal fuso orario originale ad un altro fuso orario.

1.3 Sicurezza

La sicurezza informatica è un problema molto sentito in ambito tecnico-informatico per via della crescente informatizzazione a livello aziendale/pubblico e privato, ma anche della parallela nascita di soggetti intenti alla manomissione e furto di tali dati (chiamati in gergo Cracker).

La nostra rete deve essere in grado di impedire il furto diretta o indiretta delle informazioni, sia da parte di utenti non autorizzati che da eventi accidentali.

In questo ambito possiamo distinguere il concetto di sicurezza passiva e attiva.

Per sicurezza passiva normalmente si intendono tutte quelle soluzioni tecnico-pratiche che servono a tenere protette le infrastrutture in cui i dati sono conservati (tipicamente chiamati data center) con l'ausilio di un controllo di accesso biometrico, allarmi, video sorveglianza e tutte quelle tecniche di sicurezza che permettono il controllo della identità di un individuo.

Per sicurezza attiva si intendono, invece, tutte quelle tecniche e gli strumenti tramite i quali i dati vengono resi sicuri da eventuali attacchi hacker che ne possono compromettere la loro integrità.

È evidente che la sicurezza passiva e quella attiva sono tra loro complementari e sono entrambi necessari per ottenere un grado di sicurezza elevato.

Le possibili tecniche di attacco di pirata informatico sono moltissime, perciò è necessario usare contemporaneamente diverse tecniche difensive per proteggere un sistema informatico, creando più livelli di protezione.

Per quanto riguarda l'ambito della sicurezza calendari, le operazioni di protocollo HTTP sono inviate in chiaro sulla rete a meno che non venga negoziata la protezione da snooping. Questo può essere realizzato mediante l'uso di TLS (Transport Layer Security), come definito in RFC2818.

In pratica viene creato un canale di comunicazione criptato tra il client e il server attraverso uno scambio di certificati; una volta stabilito questo canale al suo interno viene utilizzato il protocollo HTTP per la comunicazione.

Questo tipo di comunicazione garantisce che solamente il client e il server siano in grado di conoscere il contenuto della comunicazione.

In particolare, l'autenticazione di base HTTP non deve essere utilizzato a meno che non sia presente un Layer di trasporto sicuro come TLS.

Il server DEVE prendere precauzioni adeguate per garantire che clienti malevoli non possano “consumare” le risorse del server (Cpu, Ram, Hard Disk, ecc) attraverso i crafted reports.

Ad esempio, un cliente potrebbe caricare un evento con una regola di ricorrenza che specifica un ricorrente evento verificatosi ogni secondo per i prossimi 100 anni, che dovrebbe tradursi in circa 3×10^9 casi.

Un report che chiede di essere ampliato oltre tale intervallo sarebbe probabilmente da considerarsi un attacco denial-of-service sul server.

Durante la creazione di nuove risorse (tra cui le calendar collections) i client devono garantire che il nome di quest'ultime (l'ultimo segmento percorso della risorsa URI) non esponga alcun dato presente all'interno della iCalendar stessa o informazioni sulla natura di una “calendar collections”.

Ciò è necessario per garantire che un componente specifico iCalendar o la natura dei componenti in una collezione, non si possa dedurre in base al nome di una risorsa.

Quando si fanno richieste reiterate di “Free-Busy”, si accumulano informazioni sugli eventi di un utente, questa pratica potrebbe esporre i dati sensibili ad eventuali attacchi. Un approccio conservativo alla privacy dei dati del calendario lato server è quello di unificare le richieste, restituendo una sola risposta e limitando così l'esposizione dei dati.

“procedure alarms” sono un rischio per la sicurezza sia per i clienti che per i server, in particolare quando l'allarme è stato creato da un'altra agente. In questa casistica client e server non sono tenuti ad eseguire tale “Procedure alarms”.

Al di là di quanto detto in precedenza, CalDAV non solleva considerazioni di sicurezza che non sono presenti già in HTTP e WebDAV.

1.4 Sincronizzazione

WebDAV offre già funzionalità necessarie per sincronizzare una collections o un insieme di collections, di apportare modifiche non in linea, e fornisce un modo semplice per risolvere i conflitti quando si perde la connessione e si esegue un riconnessione.

ETags sono la chiave per fare questo lavoro, ma questi non sono richiesti da tutti server WebDAV. Dal momento che la funzionalità offline è più importante nelle applicazioni di calendario, rispetto ad alcune altre applicazioni WebDAV, il server CalDAV deve sostenere ETags.

Possiamo vedere diversi report da applicare sul nostro calendario:

- *Restringere la fascia oraria*: i “reports provided” in CalDAV può essere utilizzati dai client per ottimizzare le loro prestazioni in termini di utilizzo della larghezza di banda della rete e del consumo di risorse sul computer client locale.

Queste sono certamente importante considerazioni per i dispositivi mobili o palmari con capacità limitata di prestazioni, ma sono rilevanti anche per le applicazioni client desktop in casi dove le “calendar collections” contengono grandi quantità di dati.

In genere i client attuali possono recuperare i dati di calendario considerando un periodo di tempo limitato quindi, quando possibile, i clienti dovrebbero

recuperare i dati del calendario dal server tramite “CALDAV:calendar-query report”, combinato con un “CALDAV:time-range element”, per limitare la set di componenti, restituendo solo quelli necessari per compilare la visualizzazione corrente.

- *Sincronizzazione per intervalli di tempo*: Tipicamente in un calendario i dati storici (eventi, cose da fare, ecc, che sono già scaduti in linea temporale) non cambiano, anche se possono essere cancellato.

Come risultato, un client può accelerare il processo di sincronizzazione considerando soltanto i dati per il tempo presente e il futuro fino a un limite ragionevole di tempo (ad esempio, una settimana, un mese).

Se l'utente tenta quindi di esaminare una porzione del calendario l'intervallo che è stato sincronizzato, il client può eseguire un'altra operazione di sincronizzazione sul nuovo intervallo di tempo considerato.

Questa sincronizzazione "just-in-time" è in grado di ridurre al minimo la larghezza di banda della rete e del consumo di risorse per i comuni comportamenti di interazione dell'utente.

L'esempi sottostante mostrano la procedura di recupero parziale di eventi in un intervallo di tempo:

```
>> Request <<

REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:getetag/>
    <C:calendar-data>
      <C:comp name="VCALENDAR">
        <C:prop name="VERSION"/>
        <C:comp name="VEVENT">
          <C:prop name="SUMMARY"/>
          <C:prop name="UID"/>
          <C:prop name="DTSTART"/>
          <C:prop name="DTEND"/>
          <C:prop name="DURATION"/>
          <C:prop name="RRULE"/>
          <C:prop name="RDATE"/>
          <C:prop name="EXRULE"/>
          <C:prop name="EXDATE"/>
          <C:prop name="RECURRENCE-ID"/>
        </C:comp>
        <C:comp name="VTIMEZONE"/>
      </C:comp>
    </C:calendar-data>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VEVENT">
        <C:time-range start="20060104T000000Z"
                      end="20060105T000000Z"/>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>
```

Richiesta

```

>> Response <<
  HTTP/1.1 207 Multi-Status
  Date: Sat, 11 Nov 2006 09:32:12 GMT
  Content-Type: application/xml; charset="utf-8"
  Content-Length: xxxx
<?xml version="1.0" encoding="utf-8" ?>
  <D:multistatus xmlns:D="DAV:"
    xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:response>
      <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
      <D:propstat>
        <D:prop>
          <D:getetag>"ffff-abcd2"</D:getetag>
          <C:calendar-data>BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VTIMEZONE
LAST-MODIFIED:20040110T032845Z
TZID:US/Eastern
BEGIN:DAYLIGHT
DTSTART:20000404T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZNAME:EDT
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
END:DAYLIGHT
BEGIN:STANDARD
DTSTART:20001026T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZNAME:EST
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
END:VTIMEZONE
BEGIN:VEVENT
DTSTART;TZID=US/Eastern:20060102T120000
DURATION:PT1H
RRULE:FREQ=DAILY;COUNT=5
SUMMARY:Event #2
UID:00959BC664CA650E933C892C@example.com
END:VEVENT
BEGIN:VEVENT
DTSTART;TZID=US/Eastern:20060104T140000
DURATION:PT1H
RECURRENCE-ID;TZID=US/Eastern:20060104T120000
SUMMARY:Event #2 bis
UID:00959BC664CA650E933C892C@example.com
END:VEVENT
BEGIN:VEVENT
DTSTART;TZID=US/Eastern:20060106T140000
DURATION:PT1H
RECURRENCE-ID;TZID=US/Eastern:20060106T120000
SUMMARY:Event #2 bis bis
UID:00959BC664CA650E933C892C@example.com
END:VEVENT
END:VCALENDAR
</C:calendar-data>

```

```

</D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
</D:response>
<D:response>
    <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
    <D:propstat>
        <D:prop>
            <D:getetag>"ffff-abcd3"</D:getetag>
            <C:calendar-data>BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VTIMEZONE
LAST-MODIFIED:20040110T032845Z
TZID:US/Eastern
BEGIN:DAYLIGHT
DTSTART:20000404T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZNAME:EDT
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
END:DAYLIGHT
BEGIN:STANDARD
DTSTART:20001026T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZNAME:EST
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
END:VTIMEZONE
BEGIN:VEVENT
DTSTART;TZID=US/Eastern:20060104T100000
DURATION:PT1H
SUMMARY:Event #3
UID:DC6C50A017428C5216A2F1CD@example.com
END:VEVENT
END:VCALENDAR
</C:calendar-data>
</D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
</D:response>
</D:multistatus>

```

Risposta

- *Processo di Sincronizzazione*: Se un client vuole supportare la sincronizzazione dei dati del calendario, invece di scaricare i dati del calendario ogni volta che è necessario, il client ha bisogno di memorizzare il suo “calendar object resource” ovvero URI e i ETag, insieme ai dati del calendario attuale.

Mentre l'URI rimane statico per tutta la durata della “calendar object resource”, l'ETag cambierà con ogni cambiamento successiva della “calendar object resource”.

Così, per sincronizzare una cache locale di dati calendario con il server, il client può prendere prima le coppie URI/ETag per l'intervallo di tempo considerato, e confrontare i risultati con i dati memorizzati nella cache.

Qualsiasi componente cache con ETag diversa da quella sul server deve essere aggiornata.

Al fine di rilevare correttamente le variazioni di dati tra il server e il client, il client avrà bisogno di tenere un registro di quali “calendar object” sono state create, modificato o cancellato dall'ultima operazione di sincronizzazione, in modo che possa conciliare tali modifiche con i dati sul server.

Di seguito un esempio pratico di come eseguire una fase di sincronizzazione: il client invia un “CALDAV:calendar-query REPORT – query” per un intervallo di tempo specifico e chiede solo la proprietà “DAV:getetag property”.

```
REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:getetag/>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VEVENT">
        <C:time-range start="20040902T000000Z"
                      end="20040903T000000Z"/>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>
```

Il client utilizza quindi i risultati per determinare quali “calendar object resource” sono stati cambiati, creati, o eliminati sul server.

```
REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-multiget xmlns:D="DAV:"
                    xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:getetag/>
    <C:calendar-data/>
  </D:prop>
  <D:href>/bernard/work/abcd1.ics</D:href>
  <D:href>/bernard/work/mtg1.ics</D:href>
</C:calendar-multiget>
```

1.5 Ricerca

Gran parte del tempo, un client (o agente) andrà alla ricerca di un calendario fornito direttamente con l'URL di riferimento.

Per esempio, un utente digita il proprio indirizzo di calendario, il client dovrà solo confermare che l' URL punta a una risorsa che è una raccolta di calendario.

Il client può anche essere in grado di navigare collezioni WebDAV per trovare il suo “calendar collections”.

La scelta di URL HTTP significa che le “calendar object resource” sono compatibili con il software esistenti, ma che hanno lo svantaggio che il software esistente, di solito, non sanno guardare le opzioni di risposta a tale URL per determinare cosa si può fare con esso. Questo è un po' “una barriera” per l'uso di WebDAV nonché l'utilizzo di CalDAV.

Per i casi di condivisione del calendario e sull'uso di programmazione di eventi, si potrebbe avere la necessità di trovare calendari appartenente a un altro utente.

Se l'utente ha un calendario nello stesso repository, questo può essere trovata utilizzando lo spazio dei nomi principale richiesto dal supporto WebDAV ACL.

Per tutte le altre casistiche, i creatori dell'RFC non hanno trovato alcuna soluzione universale, ma i programmatori possono considerare la possibilità di usare vCard (RFC2426) insieme con il “calendar attribute” (RFC2739) per ovviare al problema.

Perché le richieste al server CalDAV supportino WebDAV ACL, devono comprendere “namespaces”, con l'aggiunta del “CALDAV:calendar-home-set property”, ci sono un paio di opzioni di CalDAV che possono usare i clienti per trovare il proprio calendario o calendario di un altro utente. In questo caso, una “DAV:principal-match REPORT “ viene utilizzato per trovare un nome di proprietà (il “CALDAV:calendar-home-set”) sul principio URL del 'utente attuale.

Con questo, un client WebDAV può imparare *"chi sono io"* e *"Dove sono i miei calendari"*.

Il corpo della richiesta "REPORT" può essere formulata in questa maniera:

```
<?xml version="1.0" encoding="utf-8" ?>
<D:principal-match xmlns:D="DAV:">
  <D:self/>
  <D:prop>
    <C:calendar-home-set
      xmlns:C="urn:ietf:params:xml:ns:caldav"/>
  </D:prop>
</D:principal-match>
```

Per trovare i calendari di altri utenti, il "DAV:principal-property-search REPORT" può essere utilizzato per filtrare alcune proprietà e restituirne altre.

Per cercare un calendario di proprietà di un utente denominato "Laurie", il "report request" sarebbe simile a quanto riportato di seguito:

```
<?xml version="1.0" encoding="utf-8" ?>
<D:principal-property-search xmlns:D="DAV:">
  <D:property-search>
    <D:prop>
      <D:displayname/>
    </D:prop>
    <D:match>Laurie</D:match>
  </D:property-search>
  <D:prop>
    <C:calendar-home-set
      xmlns:C="urn:ietf:params:xml:ns:caldav"/>
    <D:displayname/>
  </D:prop>
</D:principal-property-search>
```

Il server esegue una ricerca (case-sensitive o case-less) per un corrispondente stringa "Laurie" all'interno del "DAV: displayname property".

Così, il server potrebbe restituire "Laurie Dusseault", "Laurier Desruisseaux ", o "Wilfrid Laurier ", come "DAV:displayname Values" e restituiscono i valore di calendari per ciascuna di esse.

1.6 Memorizzazione

Si noti che tutte le "calendar collections" CalDAV (compresi quelli che l'utente potrebbe trattare come calendari pubblici o di gruppo) possono contenere allarmi su eventi e cose da fare.

Gli utenti possono sincronizzare un calendario tra più dispositivi e decidere di avere allarmi eseguiti su un dispositivo diverso da quello che ha creato l'allarme.

Non tutti i tipi di azione di allarme sono completamente interoperabili tra vari dispositivi (ad esempio, quelli che eseguono un file audio preciso da riprodurre al momento dell'allarme).

Quando l'azione è *AUDIO* e il client è configurato per l'esecuzione dell'allarme, il cliente dovrebbe svolgere il suono suggerito se è disponibile o riprodurre un altro suono, ma non dovrebbe riscrivere l'allarme solo per sostituire il suono suggerito con un suono che è localmente a disposizione.

Quando l'azione è *DISPLAY* e il client è configurato per l'esecuzione dell'allarme, il cliente deve avviare a un allarme di visualizzazione, visualizzando secondo la descrizione suggerita, ma **NON DEVE** riscrivere l'allarme per la propria convenienza.

Quando l'azione è *E-MAIL* e il client è in grado di inviare e-mail, è dovrebbe ignorare l'allarme, ma deve continuare a sincronizzare l'allarme stesso.

Le informazioni di allarme non interoperabili (ad esempio, quando qualcuno definisce un colore da utilizzare in un allarme display) dovrebbero essere messo in “non-standard properties” all'interno del “VALARM component” al fine di mantenere la base di allarme utilizzabile su tutti i dispositivi.

I client che consentono modifiche al “calendar object resource” devono sincronizzare i dati allarme che già esistono nelle risorse.

I client possono eseguire gli allarmi che vengono scaricati in questo modo, possibilmente in base alle preferenze dell'utente.

Se un client sta facendo operazioni di sola lettura su un calendario (non sta modificando e creando altri eventi) non vi è il rischio che l'evento venga perso, quindi quest'ultimo può scartare le informazioni di allarme.

Questa specifica non fa alcun tentativo di fornire allarmi multi-utente su calendari di gruppo o scoprire, per quali l'allarme è destinato.

Affrontare questi problemi potrebbe richiedere una estensioni di programmazione a “iCalendar”; ad esempio, per memorizzare gli allarmi per utente, o per indicare per quale utente di un “VALARM” era destinato.

Capitolo 2

Scheduling protocol CalDAV (RFC 6638)

2.1 Pianificazione delle operazioni

Quando una “calendar object resource” è creata, modificata, o rimossa da una “calendar collection”, il server esamina i dati del calendario e controlla se il dato rappresenta una “scheduling object resource”.

In caso affermativo, il server tenta automaticamente di trasmettere un messaggio di scheduling agli utenti di calendario appropriati. In questo caso possono verificarsi diversi tipi di operazioni di scheduling, equivalenti a operazioni di iTIP "REQUEST", "REPLY", "CANCEL" e "ADD".

2.1.1 Identificare risorse oggetto di scheduling

Le risorse oggetto calendario sulle quali il server esegue operazioni di scheduling sono indicate come risorse oggetto di scheduling. Ci sono due tipi di risorse oggetto di scheduling : le risorse oggetto di scheduling *organizzatore* e le risorse oggetto di scheduling *partecipante*.

Una “calendar object resource” è considerata un organizzatore valido se la proprietà iCalendar "ORGANIZER" è presente in tutti i componenti del calendario e impostata a un valore che corrisponde a uno dei “calendar user address” del proprietario della “calendar collection”.

Una “calendar object resource” è considerata un partecipante valido se la proprietà iCalendar "ORGANIZER" è presente in tutti i componenti del calendario e impostata in tutti i componenti allo stesso valore, il quale non corrisponde a uno dei “calendar

user address” del proprietario della “calendar collection”, e se almeno uno dei valori della proprietà iCalendar "ATTENDEE" corrisponde a uno dei “calendar user address” del proprietario della “calendar collection”.

Le risorse oggetto di scheduling tipicamente sono create dal server e sono create nella calendar collection appropriata.

2.1.2 Gestione delle risorse oggetto di scheduling

Il comportamento del server quando elabora una “scheduling object resource” dipende a seconda che sia di proprietà “dell’organizzatore” o del “partecipante”, specificato nelle proprietà del calendario.

RISORSE OGGETTO DI SCHEDULING DELL’ORGANIZZATORE

Un organizzatore può creare, modificare o rimuovere una “scheduling object resource”, è soggetto a privilegi di accesso, precondizioni e restrizioni (definite nel RFC4791).

L’organizzatore di un componente del calendario può inoltre essere anche un partecipante dello stesso componente, in questo caso il server non invia messaggi di scheduling al partecipante che corrisponde con l’organizzatore.

Se il valore del parametro della proprietà “SCHEDULE-AGENT” non è presente o impostato al valore “SERVER”, allora il server dovrebbe respingere ogni tentativo di impostare il valore del parametro della proprietà iCalendar “PARTSTAT” della proprietà iCalendar “ATTENDEE” di altri utenti nella “calendar object resource” a un qualsiasi valore che non sia “NEEDS-ACTION”.

Il server può respingere tentativi di creare una “scheduling object resource” che specifichi un valore della proprietà “UID” già specificato in una “scheduling object resource” contenuta in un’altra calendary collection dell’organizzatore.

RISORSE OGGETTO DI SCHEDULING DEL PARTECIPANTE

Un partecipante può creare, modificare, o rimuovere una “scheduling object resource”. I partecipanti hanno il permesso di effettuare alcuni cambiamenti sulle risorse oggetto di scheduling, anche se proprietà chiave come tempo di inizio, tempo di fine, luogo e sommario sono tipicamente sotto il controllo dell’organizzatore.

Il server permette ai partecipanti di fare i seguenti cambiamenti sui dati iCalendar, sempre soggetti a restrizioni, come privilegi di accesso e precondizioni :

- 1 – cambiare il valore del parametro della loro proprietà “PARTSTAT”.
- 2 – aggiungere, modificare, o rimuovere una proprietà iCalendar “TRANSP”.
- 3 – aggiungere, modificare, o rimuovere una proprietà iCalendar “PERCENT-COMplete”.
- 4 – aggiungere, modificare, o rimuovere una proprietà iCalendar “COMPLETED”.
- 5 – aggiungere, modificare, o rimuovere una proprietà iCalendar “VALARM”.
- 6 – aggiungere, modificare, o rimuovere una proprietà iCalendar “CALSCALE” all’interno del componente “VCALENDAR” superiore.
- 7 – modificare la proprietà iCalendar “PRODID” all’interno del componente “VCALENDAR” superiore.
- 8 – aggiungere la proprietà iCalendar “EXDATE” e, eventualmente, rimuovere i componenti per le istanze di ricorrenza sovrascritte.
- 9 – aggiungere, modificare, o rimuovere una proprietà iCalendar “CREATED”, “DTSTAMP” e “LAST-MODIFIED”.
- 10 – aggiungere, modificare, o rimuovere parametri della proprietà iCalendar “SCHEDULE-STATUS” delle proprietà “ATTENDEE” che hanno il parametro “SCHEDULE-AGENT” impostato su “CLIENT”.

11 – aggiungere nuovi componenti che rappresentino istanze di ricorrenza sovrascritte, a condizione che le uniche modifiche alle istanze di ricorrenza seguano le regole descritte sopra.

Quando il partecipante tenta di cambiare i dati iCalendar in maniera proibita dal server, quest'ultimo deve ritornare un errore con il codice di preconditione CALDAV:allowed-attendeescheduling-object-change.

2.1.5 Obbligare il server a inviare messaggi di scheduling

Il parametro della proprietà iCalendar “SCHEDULE-FORCE-SEND” può essere usato da un utente per forzare il server a inviare un messaggio di scheduling a un partecipante o a un organizzatore in una situazione in cui il server normalmente non invierebbe un messaggio di scheduling.

Per esempio, un organizzatore potrebbe usare questo parametro della proprietà per chiedere a un partecipante, che in precedenza aveva rifiutato un invito, di riconsiderare il suo stato della partecipazione, senza dover modificare l'evento.

2.2 Elaborazione messaggi in arrivo

Operazioni di scheduling possono comportare la consegna di messaggi di scheduling alla “scheduling Inbox collection” dell'organizzatore o dei partecipanti.

I server devono automaticamente elaborare messaggi in arrivo utilizzando le regole definite dal RFC5546, creando o aggiornando le corrispondenti risorse oggetto di scheduling nei calendari dei proprietari della “Inbox collection”.

Inoltre il messaggio viene depositato nella “Inbox collection” per indicare al client che è stata eseguita un'operazione di scheduling.

Tipicamente i messaggi vengono rimossi dalla "Inbox collection" dal client quando il calendario è stato aggiornato, prendendo atto del cambiamento.

Durante l'elaborazione di messaggi in arrivo, il server deve tener conto dei privilegi della "Inbox collection", al fine di determinare se la consegna del messaggio è permessa. I privilegi dei calendari contengono ogni corrispondenza di "scheduling object resource" che non è considerato in questo caso (ad esempio, un messaggio da un altro utente provoca modifiche a risorse nelle "calendar collections" a cui gli altri utenti non hanno ancora avuto accesso, in lettura o scrittura). Inoltre, nell'invio di messaggi di scheduling, i server devono tener conto di eventuali precondizioni della "Inbox collection" o di simili precondizioni di eventuali "calendar collection" che contengono, o potrebbero contenere, la corrispondente "scheduling object resource".

2.2.1 Elaborazione delle risorse, aggiunte e cancellate dall'organizzatore

Per messaggi di scheduling inviati da un organizzator, il server tenta per prima cosa di localizzare la "scheduling object resource" corrispondente al partecipante. Se non esiste tale corrispondenza, il sever tratta il messaggio come un nuovo messaggio, altrimenti è trattato come un aggiornamento.

In caso di nuovo messaggio il server elabora il messaggio e crea una nuova "scheduling object resource". In caso di un aggiornamento il server elabora il messaggio e aggiorna la "scheduling object resource" appartenente al partecipante al fine di applicare i cambiamenti inviati dall'organizzatore. In ogni caso, il messaggio deve figurare nell "Inbox collection" del partecipante fintanto che tutte le elaborazioni automatiche non sono terminate.

2.3 Richiesta di informazioni “BUSY TIME”

Un'informazione di “busy time” di uno o più utenti del calendario può essere determinata facendo una richiesta POST alla “scheduling Outbox collection” dell'utente del calendario che ha richiesto l'informazione (l'organizzatore).

Per realizzare ciò, il corpo della richiesta deve contenere un componente del calendario “VFREEBUSY” con la proprietà iCalendar “METHOD” impostata al valore “REQUEST”, come specificato nel RFC5546.

La risorsa identificata dall'URI della richiesta deve essere una “resource collection” del tipo CALDAV:schedule-outbox. Il valore della proprietà “ORGANIZER” nel componente “VFREEBUSY” deve corrispondere con uno degli indirizzi dell'utente del calendario proprietario della “Outbox collection”.

Una risposta a una richiesta di “busy time” che indica lo stato di uno o più utenti del calendario deve essere un documento XML con un elemento XML CALDAV:scheduleresponse come elemento di root. Questo elemento deve contenere elemento CALDAV:response per ogni utente del calendario, con ciascuno di questi elementi che deve a sua volta contenere elementi che indichino a quale utente del calendario essi corrispondano, lo stato di scheduling per quell'utente, eventuali codici di errore, e una descrizione opzionale. Per una richiesta “busy time” riuscita è presente anche un elemento CALDAV:calendar-data per ogni utente, contenente le informazioni “busy time” attuali.

Capitolo 3

iCalendar (RFC 5545) e vCard (RFC 6350)

Spesso viene utilizzata la dicitura iCal per indicare il formato standard iCalendar, nato nel 1998 (RFC2445), aggiornato nel 2009 (RFC5545) e utilizzato dalla maggior parte dei software con funzioni di calendario e/o agenda.

iCalendar è progettato per essere indipendente dal protocollo di trasporto. Ad esempio, alcuni eventi possono essere inviati via e-mail tradizionale o file di calendario, ed essere condivisi e modificati utilizzando un server WebDAV.

I server Web semplici (utilizzando solo il protocollo HTTP) sono spesso usati per distribuire i dati iCalendar di un evento e per pubblicare impegni importanti di un individuo.

La vCard fu proposta originariamente nel 1995 dal consorzio Versit, che era composto da Apple, AT&T Technologies, IBM e Siemens.

Nel dicembre 1996, il possesso del formato fu passato all'Internet Mail Consortium, una associazione commerciale di compagnie con un interesse nell'email.

vCard è un formato di file per identificare i biglietti da visita elettronici e possono essere allegati a messaggi email, ma possono essere scambiati anche in altri modi, generalmente su internet tramite server dedicati o tra diversi e differenti dispositivi mobili tramite Bluetooth, sms (iMessage su dispositivi iOS) o mms.

Questi oggetti vCard possono contenere informazioni tipiche di un biglietto da visita come nome, cognome, indirizzo, numeri di telefono, email, url, immagini e perfino file audio.

3.1 Grammatica

3.1.1 iCalendar

iCalendar è un formato di file con estensione .ICS, appositamente creato per l'ambito informatico che permette agli utenti della rete di inviare richieste di riunione e di attività ad altri utenti tramite Internet, e-mail, o la condivisione di file.

Per la grammatica, le parole chiave "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", e "OPTIONAL" in questo oggetto devono essere interpretati come descritto nell' RFC2119 e la notazione usata è descritta in ABNF (RFC5234).

Tutti i nomi, parametri, valori dei parametri di proprietà, sono "case-insensitive". Però, tutti gli altri valori sono "case-sensitive", se non diversamente specificato.

Il formato per l'oggetto iCalendar si basa sulla sintassi del text/directory media descritta in RFC2425.

Proviamo ora a descrivere i tipi di componenti che possiamo usare all'interno del nostro oggetto iCalendar:

"VEVENT" descrive un evento che ha un periodo di tempo prestabilito, nel quale l'utente è da considerarsi impegnato ma può anche essere utilizzato per gli eventi del calendario senza un momento specifico, come anniversari e promemoria quotidiani.

Può includere un "VALARM" che permette di avvertire l'utente tramite un allarme, tali eventi hanno un "DTSTART" che stabilisce un tempo di inizio e un "DTEND" che definisce un tempo finale. Se l'evento del calendario è ricorrente, "DTSTART" imposta l'inizio del primo evento ricorrente.

Di seguito un esempio di codice "VALARM" (con promemoria di 1 giorno prima):

```
BEGIN:VALARM
TRIGGER:-PT1440M
ACTION:DISPLAY
DESCRIPTION:Reminder
END:VALARM
```

Se è necessario inviare una cancellazione di un evento, l' "UID" dovrebbe essere lo stesso come l'evento originale, e le sue proprietà devono essere settate in "CANCEL". Per l'invio di un "UPGRADE" di un evento il "UID" deve corrispondere all'originale UID, l'altra proprietà del componente da impostare è: "SEQUENCE: <Numero di aggiornamento>" vale a dire che per il primo aggiornamento va scritto "SEQUENCE:1"

"VTODO" spiega una voce "to-do" (fare), cioè un action-item di assegnazione. Non tutte le applicazioni di calendario riconoscono elementi "VTODO". In particolare, Outlook non esporta Operazioni come elementi "VTODO", e ignora gli elementi "VTODO" nei calendari importati.

Un esempio di un to-do il 15 aprile 1998.

Un allarme acustico è stato specificato per ricordare il giorno prima all'utente dell'appuntamento di mezzogiorno e dovrà ripetere ogni ora per quattro volte. Il codice di seguito mostra che l'evento è già stato modificato due volte dalla data di creazione.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VTODO
DTSTAMP:19980130T134500Z
SEQUENCE:2
UID:uid4@example.com
DUE:19980415T235959
STATUS:NEEDS-ACTION
SUMMARY:Submit Income Taxes
BEGIN:VALARM
ACTION:AUDIO
TRIGGER:19980414T120000
ATTACH;FMTTYPE=audio/basic:http://example.com/pub/audio-
files/ssbanner.aud
REPEAT:4
DURATION:PT1H
END:VALARM
END:VTODO
END:VCALENDAR
```

Esempio

“VJOURNAL” è una voce di diario. Esse attribuiscono testo descrittivo a una particolare data del calendario, possono essere utilizzati per registrare un record giornaliero delle attività o descrivere il progresso con un relativo ingresso to-do.

Un componente del calendario "VJOURNAL" non prende il tempo sul calendario, quindi non ha alcun effetto sul tempo libero o occupato (proprio come le voci trasparenti).

```

BEGIN:VCALENDAR
VERSION:2.0
PROPID:--//ABC Corporation//NONSGML My Product//EN
BEGIN:VJOURNAL
DTSTAMP:19970324T120000Z
UID:uid5@example.com
ORGANIZER:MAILTO:jsmith@example.com
STATUS:DRAFT
CLASS:PUBLIC
CATEGORIES:Project Report, XYZ, Weekly Meeting
DESCRIPTION:Project xyz Review Meeting Minutes\n
  Agenda\n1. Review of project version 1.0 requirements.\n2.
  Definition
  of project processes.\n3. Review of project schedule.\n
  Participants: John Smith, Jane Doe, Jim Dandy\n-It was
  decided that the requirements need to be signed off by
  product marketing.\n-Project processes were accepted.\n
  -Project schedule needs to account for scheduled holidays
  and employee vacation time. Check with HR for specific
  dates.\n-New schedule will be distributed by Friday.\n-
  Next weeks meeting is cancelled. No meeting until 3/23.
END:VJOURNAL
END:VCALENDAR

```

Esempio di una voce di diario

“VFREEBUSY” è una richiesta di “free/busy time” , è una risposta a una richiesta “quando sei occupato”.

```

BEGIN:VCALENDAR
VERSION:2.0
PROPID:--//RDU Software//NONSGML HandCal//EN
BEGIN:VFREEBUSY
DTSTAMP:20151013T080000Z
UID:uid6@example.com
ORGANIZER:MAILTO:jsmith@example.com
DTSTART:19980313T141711Z
DTEND:19980410T141711Z
FREEBUSY:19980314T233000Z/19980315T003000Z
FREEBUSY:19980316T153000Z/19980316T163000Z
FREEBUSY:19980318T030000Z/19980318T040000Z
URL:http://www.example.com/calendar/busytime/jsmith.ifb
END:VFREEBUSY
END:VCALENDAR

```

Esempio

di seguito due tabella riassuntiva dei TAG utilizzabili su VCALENDAR e VEVENT:

Tabella 1. Proprietà iCalendar VCALENDAR

Tag name	Descrizione articolo
PRODID	Identificativo del prodotto che ha creato l'oggetto calendario. Questa proprietà è obbligatoria.
VERSION	La versione dell'oggetto calendario. Questa proprietà è obbligatoria.
UID	Il nome univoco. Viene convertito in lettere maiuscole e gli spazi vengono convertiti in caratteri di sottolineatura (_). Non può contenere più di 40 caratteri.
DESCRIPTION	Descrizione dell'utente. Non può contenere più di 120 caratteri.
VTIMEZONE	Il fuso orario da utilizzare per interpretare i valori delle date.
X-DEFAULTDTSTART	Il valore predefinito per VEVENT.DTSTART. Il valore ora/data.
X-DEFAULTDTEND	Il valore predefinito per VEVENT.DTEND. Il valore ora/data.
X-DEFAULTUNTIL	Il valore predefinito per VEVENT.RRULE.UNTIL. Il valore ora/data.
X-CALAVAILABILITYVALUE	Stringa che può essere richiamata dalle API TSS.
X-CALFREESATURDAY	Esclude il sabato dal calendario delle disponibilità. I valori supportati sono TRUE e FALSE.
X-CALFREESUNDAY	Esclude la domenica dal calendario delle disponibilità. I valori supportati sono TRUE e FALSE.

Tabella 2. Proprietà iCalendar VEVENT

Nome proprietà	Descrizione articolo
UID	Il nome univoco. Viene convertito in lettere maiuscole e gli spazi vengono convertiti in caratteri di sottolineatura (_).
DESCRIPTION	Descrizione dell'utente. Non può contenere più di 120 caratteri.
DTSTART	La data e l'ora di inizio dell'intervallo di validità. Sovrascrive il valore definito per X-DEFAULTDTSTART.
DTEND	La data e l'ora di fine dell'intervallo di validità. Sovrascrive il valore definito per X-DEFAULTDTEND.
RRULE	Regola applicata (si esclude a vicenda con RDATE). Le frequenze supportate sono: <ul style="list-style-type: none"> • DAILY • WEEKLY • MONTHLY • YEARLY <p>È possibile specificare la proprietà UNTIL per indicare la data di fine dell'intervallo di validità. Sovrascrive il valore definito per X-DEFAULTUNTIL.</p>
RDATE	Elenco delle date precise da includere (si esclude a vicenda con RRULE).
EXDATE	Elenco delle date precise da escludere.

3.2 Creazione oggetti

3.2.1 Oggetto iCalendar

In genere, la creazione di un oggetto sarà composta da un singolo iCalendar. Tuttavia, più oggetti iCalendar possono essere raggruppati insieme.

Proviamo ad analizzare il nostro codice riportato nella figura sottostante che identifica la creazione di un oggetto iCalendar, la prima linea di codice deve essere "BEGIN:VCALENDAR", e l'ultima riga deve essere "END:VCALENDAR"; i contenuti tra queste linee è chiamato il "icalbody".

La seconda linea "VERSION: 2.0" indica che i dati sono in formato iCalendar.

Il corpo dell'oggetto iCalendar (il icalbody) è costituito da un elenco di proprietà di calendario e uno o più componenti di calendario. Le proprietà valgono per l'intero calendario.

Ad esempio, il componente del calendario può specificare un evento, un "to-do list", una "journal entry", "time zone information", o informazioni sul tempo, "free/busy time", o un allarme. Righe vuote non sono ammesse in alcune versioni di utilizzo odierno (il calendario di Google per esempio).

Ecco un semplice esempio di un oggetto iCalendar, "Bastille Day Party" evento che si verifica il 14 luglio 1997 17:00 (UTC) fino al 15 Luglio 1997 03:59:59 (UTC):

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
UID:uid1@example.com
DTSTAMP:19970714T170000Z
ORGANIZER;CN=John Doe:MAILTO:john.doe@example.com
DTSTART:19970714T170000Z
DTEND:19970715T035959Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

3.2.2 Oggetto vCard

Anche in questo caso proviamo ad analizzare il nostro codice, riportato nella figura sottostante, che identifica la creazione di un oggetto vCard, la prima linea di codice del nostro oggetto vCard deve essere "BEGIN:VCARD" , e l'ultima riga deve essere "END:VCARD"; i contenuti tra queste linee è chiamato il "vcardbody".

La seconda linea "VERSION: 2.1" indica che i dati sono in formato vCard.

Il corpo dell'oggetto è costituito da un elenco di proprietà di contatto e indicano i dati comunemente richiesti come: il nome "N", il cognome "FN", il numero di telefono ed il suo attributo "TEL;CELL", l'indirizzo con il suo attributo "ADR;HOME" e tutti i dati che comunemente vengono utilizzati per il salvataggio di un contatto sul nostro dispositivo (URL, EMAIL, ecc).

```
BEGIN:VCARD
VERSION:2.1
N:Ongaro;Giorgio;sig
FN:Giorgio Ongaro
NICKNAME:Il Giorgio
TEL;HOME;VOICE:123456789
TEL;CELL;VOICE:123456789
TEL;HOME;FAX:123456789
ADR;HOME;;;Via Pesa del Lino, 2;Monza;MI;20052;Italia
LABEL;HOME;ENCODING=QUOTED-PRINTABLE:Via Pesa del Lino, 2=0D=0AMonza, Mi
00100=0D=0AItalia
URL;HOME:http://www.ilgiorgio.it
EMAIL;PREF;INTERNET:ongaro.giornio@hobby.center.ong
REV:20080817T172534Z
END:VCARD
```


Capitolo 4

Conclusioni

Ognuno di noi ha sempre sperato che la tecnologia ci desse la possibilità di gestire calendario, contatti e appuntamenti in maniera semplice, magari dal nostro dispositivo mobile.

L'avvento dell'informatica ha permesso di facilitare enormemente questo compito, ampliandone le prospettive. Si tratta oggi di una innovazione talmente richiesta e comune che molti sistemi operativi la includono direttamente nei loro sistemi, senza richiedere installazioni di software aggiuntivi (ad esempio Apple Mac OSX e il suo iCalendario oppure Microsoft con Outlook) e molti produttori di smartphone ne integrano le funzionalità di condivisione in maniera nativa, senza l'ausilio di software supplementari (iPhone con iOS, Microsoft con Windows Mobile o Google con Android).

Le funzionalità richieste per un servizio come questo sono, oltre la gestione dei propri impegni personali, la possibilità di avere più calendari, di metterli in condivisione con altre persone, di richiedere ad altri la disponibilità per certi giorni ed orari, di organizzare riunioni con più partecipanti, e di avere queste informazioni a portata di mano anche quando si è fuori dall'ufficio o da casa.

Lo stesso discorso è valido anche per il mondo della rubrica telefonica, che con il passare degli anni si è evoluta.

Nei primi tempi i nostri contatti erano presenti solo sulla nostra SIM CARD, con il rischio che allo smarrimento del telefono tutto andasse perduto; ora possiamo consultarli e modificarli da più dispositivi grazie al protocollo CardDAV che ne permette la sincronizzazione tra più dispositivi senza più il rischio di perdite di dati.

Tutto questo è stato reso possibile grazie alla creazione di sistemi Cloud, ovvero accesso a Web Server dedicati e di proprietà delle grandi distribuzioni (Google con

Drive, Apple con iCloud e Microsoft con OneDrive) che conservano i nostri dati e ci permettono l'interazione senza mai correre il rischio di perdere le nostre informazioni.

Questo metodo ha cambiato enormemente tutta la visione globale di sincronizzazione di dati personali ma ha certamente reso più facile per l'utente inesperto il salvataggio.



Ringraziamenti

Siamo giunti ai ringraziamenti.

Ci sono tante persone da ringraziare e poco tempo/spazio per farlo, ma comunque ci si prova.

Il primo va alla mia famiglia che mi ha sostenuto per tanto tempo ed ha portato molta pazienza, ma soprattutto alla mia compagna che nei momenti di vera difficoltà ha saputo tirare fuori il meglio di me e rendere tutto questo possibile.

Nella mia lunga carriera universitaria sono state presenti moltissime persone senza le quali tutto questo non sarebbe potuto succedere, voglio ringraziare Chiara, Pei, Francesco, Casa Alto, Cristiano i primi veri amici dell'università ma non posso fare a meno che ringraziare anche tutti quelli che nel tempo si sono divertiti nel tentare l'impossibile, ovvero farmi LAUREARE, e qui come posso non ricordare Foschini, Michele e Riccardo, tre persone che come minimo si meritano metà della mia laurea per la pazienza che sono riusciti ad avere nei miei confronti ripetendo allo sfinimento le cose fino a farle entrare nella mia testa.

Voglio ringraziare anche il Prof. F. Callegati per la sua vera disponibilità dimostrata durante questo percorso.

Insomma, grazie... di tutto!

Bibliografia

- [1] <http://caldav.calconnect.org/>
- [2] <http://eu.wikipedia.org/>
- [3] <http://carddav.calconnect.org/>
- [4] <http://www.ietf.org/rfc>
- [5] <http://www.google.it/>