OpenACS/TCL Conf 2025, Bologna

The OpenACS CalDAV package

Markus Moser

WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED · AACSB ACCREDITED · ASSOCIATION AMBA ACCREDITED

Project Motivation

- CalDAV is an IETF standard from 2007 which allows a client to access and manage calendar data.
- HTTP-based, extends WebDAV protocol, uses XML.
- Aim: Users should be able to manage their OpenACS calendar feed from their preferred client.
- Focus on implementing essential CalDAV features ensuring compatibility with widely used calendar clients.

Why is CalDAV non-trivial?

RFC 4918 (WebDAV) and RFC 5545 (iCalendar) are foundational.

RFC 4791 (CalDAV) depends on both WebDAV and iCalendar.

Extensions:

RFC 6638 (Scheduling Extensions) builds on CalDAV and iTIP.

RFC 6764 (Discovery) depends on CalDAV.

RFC 5546 (iTIP) depends on iCalendar.

RFC 6352 (CardDAV) depends on WebDAV.

RFC 6578 (Collection Synchronization) depends on WebDAV.

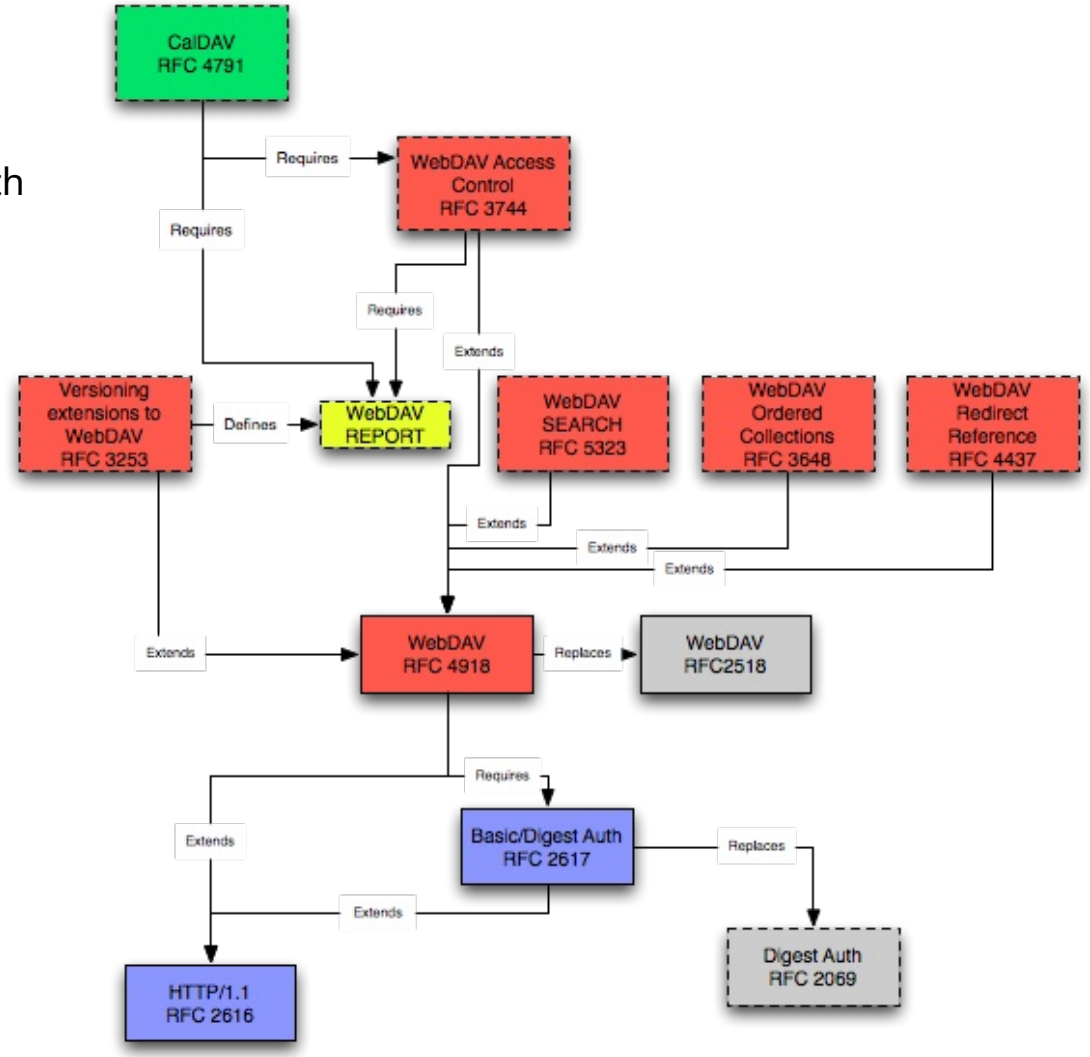


Image src: https://evertpot.com/227/

How does CalDAV work?

- A CalDAV server organizes calendars as WebDAV collection.
- Each event is a separate iCalendar (.ics) resource.
- Multiple calendars per user are possible.
- Clients perform HTTP methods on resources to manage them.

Common Clients for CalDAV

- CalDAV will not work out of the box on Windows and Android OS
  - Plugins must be installed

  Desktop Clients
  - Thunderbird / Lightning
  - Outlook (with CalDAV Plugin)
  - Apple iCal
  - Evolution
- Mobile Clients
  - DAVx5 (Android)
  - CalendarSync (Android)
  - iOS
  - BlackBerry

Developers should be aware that client implementations differ in supported features, vendors often rely on custom extensions, and OS version changes can affect compatibility.

Dependencies on the OpenACS side

- Calendar package
  - https://github.com/openacs/calendar
- Xotcl-core
  - 50-protocol-handler-procs.tcl
    - Implements HTTP callable methods with authentication
  - ical-procs.tcl
    - Importing and exporting of calendar items in ical (.ics) format
  - https://github.com/openacs/xotcl-core
- tDOM for parsing XML
- CalDAV Implementation requirements
  - Implement a subset of CalDAV sufficient to support major clients
  - HTTP callable methods for CalDAV: GET, PUT, DELETE, PROPFIND, REPORT, OPTIONS
  - Support iCal VEVENTs for describing calendar entries (ical-procs.tcl)

Implementation challenges

- Access Control
  - In the OpenACS/Dotlrn world, a user typically has a private calendar with read/write access.
  - Dotlrn communities can have calendars, users get read access to the calendars of communities they are members of.
  - But for CalDAV we want **one aggregated virtual calendar for each user** instead of many to avoid syncing each calendar in a potentially large list.
  - New calendar items can be written to the private calendar.
  - Trying to modify an event on a read-only calendar will result in permission errors.
  - CalDAV uses Principals to define what a user can access (resource discovery)

Implementation details – supported HTTP methods 1/2

- GET
  - Individual calendar items (must end with .ics)

    /caldav/calendar/12872/myevent.ics
  - Complete aggregated calendar for this user (potentially large)

    /caldav/calendar
- OPTIONS
  - Determines and returns capabilities of CalDAV server
- PROPFIND
  - Discover location of CalDAV service and properties of resources, including calendar events.
  - Example properties: principal-URL, calendar-home-set, resourcetype, sync-token, getetag …

Implementation details – supported HTTP methods 2/2

- PUT
  - Update/create a single calendar item denoted by a uid
- REPORT
  - Request specific calendar data, such as events within a time range
    - Currently, three report types are supported
      - calendar-multiget
      - calendar-query
      - sync-collection using a sync token
    - Partial retrieves allow to request only specific parts instead of fetching the whole calendar.

Sample PROPFIND request

```
PROPFIND /caldav/ HTTP/1.1
Content-Type: application/xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8" ?>
<D:propfind xmlns:D="DAV:"
xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:current-user-principal/>
  </D:prop>
</D:propfind>
```

Response

```
<multistatus xmlns="DAV:">
  <response>
    <href>/caldav/</href>
    <propstat>
      <prop>
        <current-user-principal>
          <href>/caldav/principal/</href>
        </current-user-principal>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

Sample PUT request – create an item

```
PUT /caldav/calendar/12345789.ics HTTP/1.1
Content-Type: text/calendar; charset=utf-8
If-None-Match: *

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Caldav Client//EN
BEGIN:VEVENT
UID:123456789
DTSTART;TZID=Europe/Vienna:20250710T120000
DTEND;TZID=Europe/Vienna:20250710T122500
SUMMARY:Meeting
LOCATION:Office
DESCRIPTION:A regular meeting
END:VEVENT
END:VCALENDAR
```

Sample REPORT request – retrieve data from calendar

```
REPORT /caldav/calendar/ HTTP/1.1
Depth: 1
Prefer: return-minimal
Content-Type: application/xml; charset=utf-8

<c:calendar-query xmlns:d="DAV:" xmlns:c="urn:ietf:params:xml:ns:caldav">
   <d:prop>
      <d:getetag />
      <c:calendar-data />
   </d:prop>
   <c:filter>
      <c:comp-filter name="VCALENDAR" />
      <c:comp-filter name="VEVENT">
      <c:time-range start="20250221T000000Z"/>
   </c:filter>
</c:calendar-query>
```

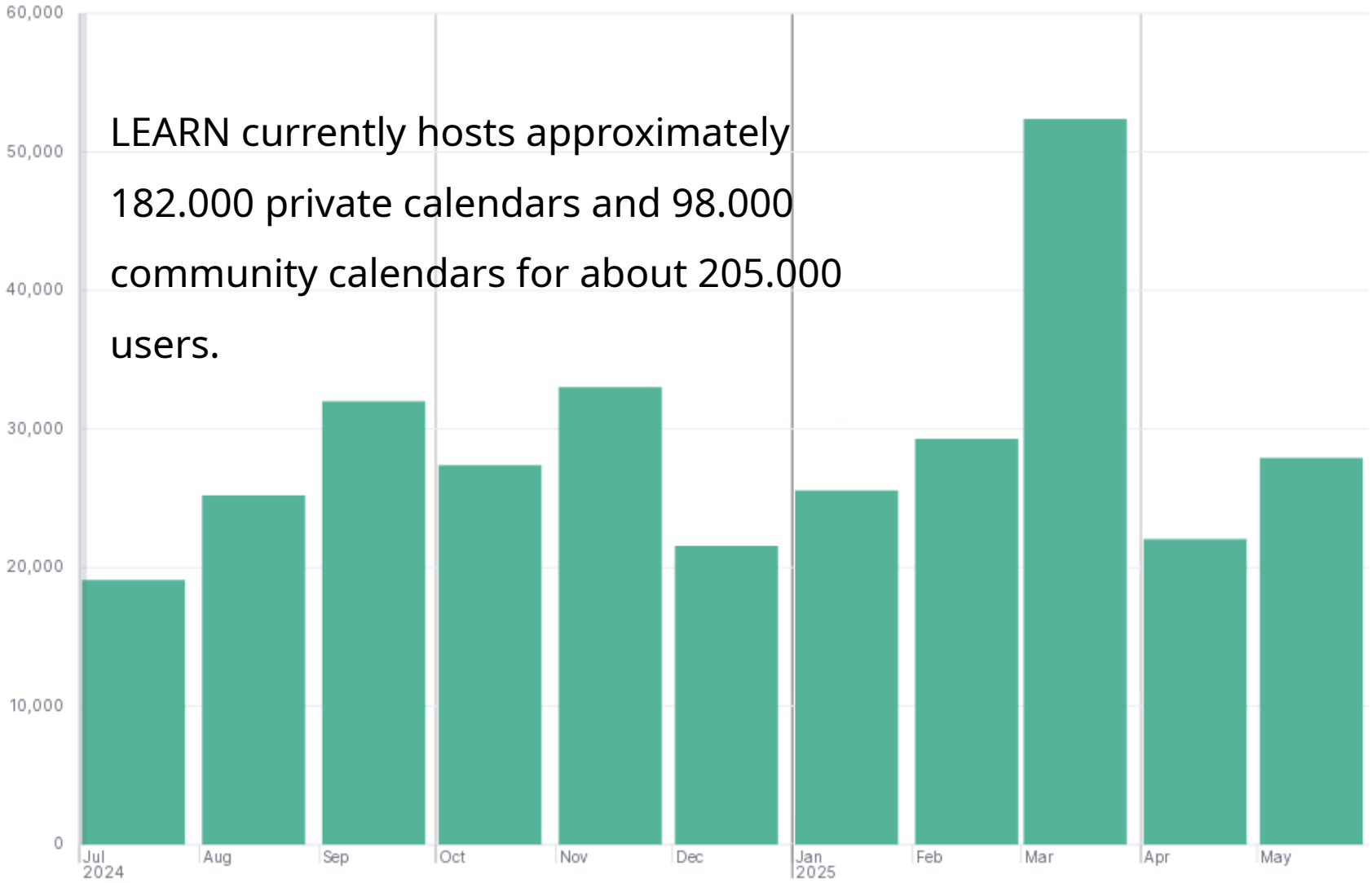
Sample response for a REPORT request

```
<?xml version="1.0" encoding="utf-8" ?>
  <D:multistatus xmlns:D="DAV:"
          xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:response>
      <D:href>https://..../caldav/calendar/12345.ics</D:href>
      <D:propstat>
        <D:prop>
          <D:getetag>"fffff-abcd2"</D:getetag>
          <C:calendar-data>BEGIN:VCALENDAR
....
          END:VCALENDAR
          </C:calendar-data>
          </D:prop>
          <D:status>HTTP/1.1 200 OK</D:status>
        </D:propstat>
    </D:response>
```

Automated Tests

- 394 automated tests for regression testing
- Includes simulating interaction with
  - Thunderbird
  - iOS
  - Android
  - MacOS

Request statistics for CalDAV on learn.wu.ac.at from July 2024 to May 2025



Limitations

- Only a predefined list of iCalendar attributes is supported, all other attributes are ignored.
- More complex syncing operations such as incremental updates.
- Invitations and appointments are not supported.
- For large calendar aggregates, queries can become expensive.

Outlook and potential future work

- Basic HTTP authentication is supported, other authentication methods could be added.
- Sync-collection is not fully supported, since we need to persist the token of previous syncs and calculate the delta between syncs.

- Interested? Your contribution is welcome:
  - https://github.com/openacs/caldav

Thank You for Your Time - Questions Welcome

Institute for Complex Networks
Welthandelsplatz 1, 1020 Vienna, Austria

Markus Moser

markus.moser@wu.ac.at
www.wu.ac.at

Special thanks to all contributors to the caldav package:
Hubert Stickelberger, Nermin Topcagic, Gustaf Neumann, Antonio Pisano