

PLAI Visitor Management

High Level Technical Document

Version 0.6

Table of Contents

Version History	3
Abbreviations used in this document	4
1. Introduction	5
2. Problem Statement	5
3. High Level Flow	5
4. Visitor access flow	6
4.1 Prerequisite	6
4.2 Example use case	7
5. Detailed Workflow	7
6. Reuse of existing PSIA specifications	7
6.1 Service Model	7
6.2. CSEC	8
6.3. Area Control	8
7. New resources for PLAI Visitor Management profile	8
7.1. /PSIA/AreaControl/Meeting/info	8
8. Workflow scenarios	10
8.1. A new credential holder is added in LDAP/HR system.	10
8.2. Credential holder record updated	11
8.3. Credential holder deleted from LDAP/HR system	11
8.4. A new meeting invite created by a credential holder.	11
8.5. A recurring meeting which will be recurring weekly.	12
8.6. A recurring meeting which will be recurring Daily.	13
8.7. A recurring meeting which will be recurring monthly on a specific day each month(Absolute monthly).	15
8.8. A recurring meeting which will be recurring monthly(Relative monthly).	16
8.9. A recurring meeting which will be recurring Yearly(Absolute yearly)	17
8.10. A recurring meeting XML is sent ,which will be recurring Yearly(Relative yearly)	19
9. Appendix	20
9.1. PSIA XSD change	20

Version History

Version	Date	Author	Summary
0.1	16-April-2018	Praveen Jha	Initial Draft
0.2	30-April-2018	Shivansh Rawat	Added scenarios for different meeting occurrence types
0.3	10-May-2018	Praveen Jha	Added PSIA XSD changes for ICalUID
0.4	15-May-2018	Praveen Jha	Support for non LDAP authoritative sources
0.5	21-May-2018	Praveen Jha	Fixed Type in the time zone name
0.6	24-May-2018	Praveen Jha	Changed all references of PLAI Connect with PLAI Visitor Management

Abbreviations used in this document

PSIA	Physical Security Interoperability Alliance
PLAI	Physical/Logical Access Interoperability
PACS	Physical Access Control System
LDAPv3	Lightweight Directory Access Protocol Version 3
REST	REpresentational State Transfer
CSEC	PSIA Common Security Service Specification
AS	Authoritative Source
HTTPS	Hypertext Transport Protocol (HTTP) Secure
VMS	Visitor Management System
PVM	PLAI Visitor Management
iCalendar/iCal	Standard file format for meeting requests
xCal	xCalendar format equivalent for iCalendar (.ics)

1. Introduction

PLAI Visitor Management (PVM) uses PLAI specification to take a company's employee records from an LDAP v3 compatible system, such as Microsoft Active Directory, or any other authoritative source and syncs them to a Visitor Management System (VMS). The VMS would expose a PLAI endpoint that can be used to sync employee record and handle employee on or off-boarding.

This high level technical document presents use-cases and solutions for corporations that are integrating visitor management system(s) (VMS) and physical access control system(s) (PACS) as an integral part of their strategy for enhanced security across their facility and building perimeter.

This document uses PLAI specifications instead of developing something from scratch.

Corporations that are already using PLAI Agent to sync users and roles with PACS can enhance PLAI Agent to sync changes with VMS without having to install a separate tool.

PLAI Visitor Management should also be made available as a separate application for corporations that wish to use it only for VMS integration. For corporations that don't use an LDAP v3 compatible authoritative source, they would have to build their own agent to sync employee records with VMS.

2. Problem Statement

For a corporation that span multiple locations, people often need to travel from one office location to another. These locations might span different geographies. Each of these office locations might have different visitor management system in place, which is a hindrance in smooth movement of employees. The absence of any system integration requires visitor information to be manually entered in multiple VMS systems, creating redundancy and potentially increasing visitor registration time.

With PLAI Visitor Management, employees can schedule access to remote locations with the ease of sending calendar invite. PLAI compatible VMS system would understand the calendar invite used by popular messaging systems such as Microsoft Outlook, Google Gmail, and many others.

iCalendar is a standard file format which allows Internet users to send meeting requests and tasks to other Internet users by sharing or sending files in this format through various methods. The files usually have an extension of .ics.

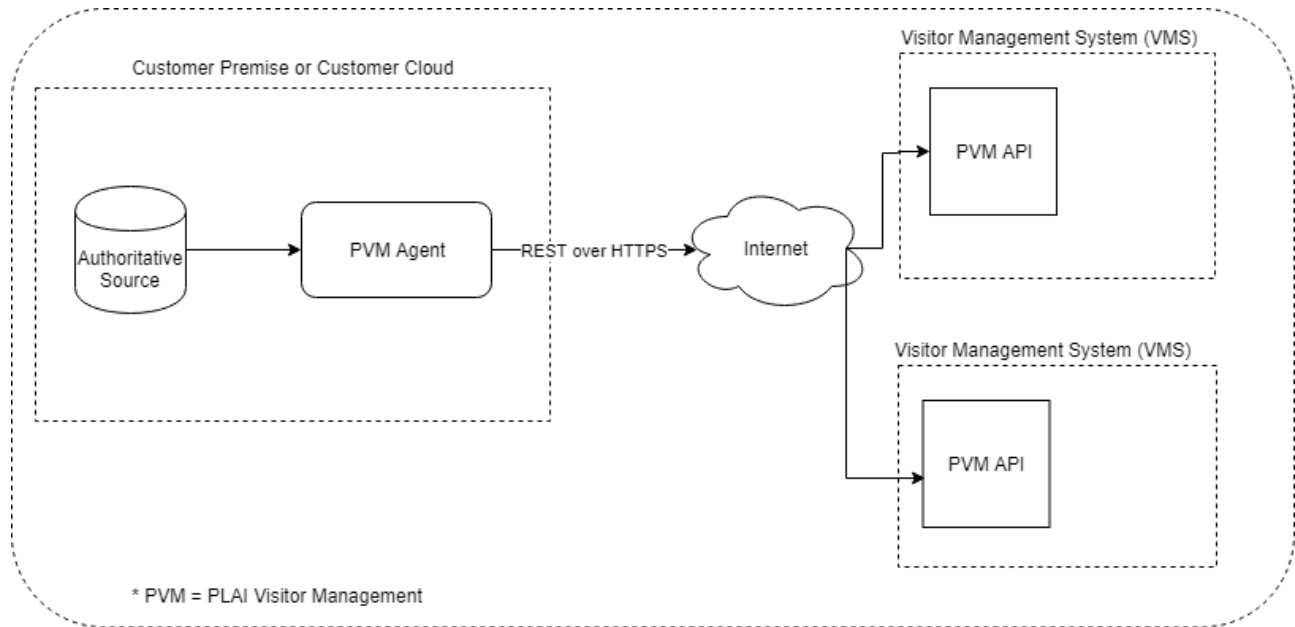
VMS would process this iCalendar meeting request and provide visitor access to the employee, which would allow time bound easy movement of employees.

3. High Level Flow

PLAI Visitor Management uses the PLAI specification, published by the Physical Security Interoperability Alliance (PSIA) to synchronize employee data from a company's authoritative source with Visitor Management System.

This authoritative source can be any LDAP v3 system such as Microsoft Active Directory or any HR system. Key employee data is pushed to VMS such as their objectguid (for future references), name, and email, which then appear in VMS as a visitor record available for assigning credentials.

The transmission of the employee's email address is important as VMS requires an accurate and current email address from the authoritative source to facilitate visitor access.



- 6.1. VMS would expose a public facing (internet accessible) endpoint implementing PLAI Visitor Management profile that would allow PLAI Visitor Management Agent to sync any changes.
- 6.2. PLAI Visitor Management Agent would be installed on the customer's premise or within the customer's cloud on a PC that can access the Authoritative Source (PLAI Visitor Management agent only requires read permissions).
- 6.3. PLAI Visitor Management Agent would poll authoritative source of any changes.
- 6.4. Any changes would be securely transmitted via a REST web service secured using HTTPS (TLS 1.2+) pushed to all the registered VMS.

4. Visitor access flow

4.1 Prerequisite

Corporations needs to setup distinct email address for each location.

For example, a corporation with office locations in New York and Chicago needs to set-up separate email for each location, like VisitNewYork@contoso.com and VisitChicago@contoso.com.

Or they can set-up a single email for visitor access (e.g. visitor@contoso.com) with the "LOCATION" attribute provided in .ics file to identify the visiting office location. In this case the "LOCATION" attribute of .ics file would have address of their New York or Chicago office as the case might be.

If the credential holder plans to visit another office location and require visitor access, they need to create a new meeting invite.

4.2 Example use case

Contoso Inc. has offices in New York and Chicago and has setup PLAI Visitor Management. John Doe is an employee of New York office. With PLAI Visitor Management, the employee basic information is shared with the VMS at their New York and Chicago offices.

John is now planning to visit Chicago office. If Contoso Inc. has separate email address created for each location, John can send an email to VisitChicago@contoso.com through Outlook or any other email client they use.

The HR or visitor management software at Contoso Inc. would send the iCalendar (.ics file) of this meeting to PLAI Visitor Management API implemented by the VMS system at their Chicago office. The VMS system would then allow visitor access to John to their Chicago office.

If additional level of authentication is required, the VMS can trigger an automated email to authorizer of the building for them to confirm the visit before access is granted to John.

5. Detailed Workflow

- PLAI Visitor Management Agent Setup (one time): Admin, typically IT, initially uses configuration manager to define connection settings and filters to get users from LDAP v3 or any other HR system. PLAI Visitor Management Agent would communicate with LDAP or HR system with the settings provided by configuration manager and pull any changes.
- Any modification would be updated in local database.
- PLAI Visitor Management Agent would notify all the registered VMS of any changes like a new user added, a user updated or deleted.
- VMS service would GET updates from Authoritative Source.

6. Reuse of existing PSIA specifications

These resources are already part of PLAI profile and has been reused in PLAI Visitor Management profile.

6.1 Service Model

PLAI Visitor Management Agent and PLAI Visitor Management API (web service) would use **Representational State Transfer (REST)** architecture with XML as the format for data exchange.

HTTPS with basic authentication would be used to support privacy of data. IP based restrictions can be applied at each end point to further enhance the security.

Following resources from Service Model specification would be reused in PLAI profile.

Resource	GET	PUT	POST	DELETE
/PSIA/index	✓			
/PSIA/indexr	✓			
/PSIA/profile	✓			
/PSIA/description	✓			

6.2. CSEC

Following resources from CSEC specification would be reused in PLAI.

Resources	GET	PUT	POST	DELETE
/PSIA/CSEC/deviceOwnership	✓			✓
/PSIA/CSEC/deviceOwnership/scope	✓	✓		
/PSIA/CSEC/deviceOwnership/status	✓			

(Note: Mandatory **Index** and **Description** resource would be supported by each service)

6.3. Area Control

Following resources from PSIA Area Control specification would be reused:

Resources	GET	PUT	POST	DELETE
/PSIA/AreaControl/CredentialHolders/info/<credentialHolderID>	✓	✓		✓
/PSIA/AreaControl/CredentialHolders/state/<credentialHolderID>	✓	✓		

(Note: Mandatory **Index** and **Description** resource would be supported by each service)

7. New resources for PLAI Visitor Management profile

Resources	GET	PUT	POST	DELETE
/PSIA/AreaControl/Meeting/info	✓	✓		✓

7.1. /PSIA/AreaControl/Meeting/info

URI	/PSIA/AreaControl/Meeting/info	Type	Resource
Function	Used to create a new meeting request or to update an existing meeting request		
Methods	Query String(s)	Inbound Data	Return Result
GET			<iCalendar>
PUT		<iCalendar>	<ResponseStatus>
PUT		iCalendar (.ics) file	
DELETE			<ResponseStatus>

Notes:

To uniquely identify a meeting, each meeting calendar invite (.ics file) should mandatorily have a UID that conforms to RFC 2445.

Sample .ics file content (the content type of attached .ics file should be "text/calendar")

```
BEGIN:VCALENDAR
...
BEGIN:VEVENT
CREATED:20180419T021727Z
DTEND;TZID=US/Pacific:20180515T110000
DTSTAMP:20180419T021727Z
```


DTSTART;TZID=US/Pacific:20180515T100000
LAST-MODIFIED:20180419T021727Z
RRULE:FREQ=DAILY;UNTIL=20180519T035959Z
SEQUENCE:0
SUMMARY:Meeting
TRANSP:OPAQUE
UID:21B97459-D97B-4B23-AF2A-E2759745C299
END:VEVENT
END:VCALENDAR

Sample XML:

```
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>
            <text>Organizer@Contoso.com</text>
          </organizer>
          <summary>
            <text>Subject</text>
          </summary>
          <dtstamp>
            <date-time>2008-02-05T19:12:24Z</date-time>
          </dtstamp>
          <tzname>
            <text>PST</text>
          </tzname>
          <dtstart>
            <date>2000-04-04T02:00:00</date>
          </dtstart>
          <dtend >
            <date>2000-04-04T02:30:00</date>
          </dtend>
          <location >
            <text>Location</text>
          </location>
        </properties>
      </vevent>
    </components>
  </vcalendar>
</icalendar>
```

```

<attendee>
  <text>XYZ1@gmail.com</text>
</attendee>
<attendee>
  <text>XYZ2@gmail.com</text>
</attendee>
<uid>
  <text>4088E990AD89CB3DBB484909</text>
</uid>
</properties>
</vevent>
</components>
</vcalendar>
</icalendar>

```

The information of a meeting can be obtained using a GET request passing in the UID. The same meeting can be updated using a PUT request either passing in the .ics file again or using XML data. A meeting can be deleted by issuing a DELETE request with the UID in URI.

8. Workflow scenarios

Below are the different workflow scenarios and how these would be handled.

8.1. A new credential holder is added in LDAP/HR system.

PLAI Visitor Management Agent call PUT to /PSIA/AreaControl/CredentialHolders/info and pass the credential holder information to all the registered VMS. The mandatory information that needs to be passed to VMS are the ObjectGUID (LDAP unique identifier), Email, Last Name (Surname) and First Name (GivenName).

```

<CredentialHolderInfo version="1.0" xmlns="urn:psialliance-org">
  <ID>0</ID>
  <UID>{e58bae7a-06c8-4283-90e0-82ccf2a4b114}</UID>
  <Name>Doe, John</Name>
  <GivenName>John</GivenName>
  <Surname>Doe</Surname>
  <State>Active</State>
  <RoleIDList />
  <AttributeList>
    <Attribute>
      <Name>psialliance.org/AreaControl.CredentialHolderAttribute/Email</Name>
      <Value>John.Doe@contoso.com</Value>
    </Attribute>
  </AttributeList>
</CredentialHolderInfo>

```

8.2. Credential holder record updated

PLAI Visitor Management agent would call PUT to /PSIA/AreaControl/CredentialHolders/info to update credential holder record in VMS.

8.3. Credential holder deleted from LDAP/HR system

PLAI Visitor Management agent would call DELETE to /PSIA/AreaControl/CredentialHolders/info/{GUID} to delete credential holder record from VMS.

8.4. A new meeting invite created by a credential holder.

PLAI Visitor Management Agent call PUT to /PSIA/AreaControl/Meeting/info and pass the meeting information to the registered VMS. Meeting invite would be sent to VMS either as .ics file (as file attachment) or passing the XML document.

The information would be in XML format as shown below.

```
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>
            <text>Organizer@Contoso.com</text>
          </organizer>
          <summary>
            <text>Subject</text>
          </summary>
          <dtstamp>
            <date-time>2008-02-05T19:12:24Z</date-time>
          </dtstamp>
          <tzname>
            <text>PST</text>
          </tzname>
          <dtstart>
            <date>2000-04-04T02:00:00</date>
          </dtstart>
          <dtend >
            <date>2000-04-04T02:30:00</date>
          </dtend>
        </properties>
      </vevent>
    </components>
  </vcalendar>
</icalendar>
```

```

    <location >
      <text>Location</text>
    </location>
  <attendee>
    <text>XYZ1@gmail.com</text>
  </attendee>
  <attendee>
    <text>XYZ2@gmail.com</text>
  </attendee>
  <uid>
    <text>4088E990AD89CB3DBB484909</text>
  </uid>
</properties>
</vevent>
</components>
</vcalendar>
</icalendar>

```

8.5. A recurring meeting which will be recurring weekly.

PLAI Visitor Management agent would call PUT to /PSIA/AreaControl/Meeting/info to pass the meeting information to the registered VMS. The below meeting Occurs every 3 weeks on Sunday, Tuesday, and Friday for 10 recurrences.

```

<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>
            <text>Organizer@Contoso.com</text>
          </organizer>
          <summary>
            <text>Subject</text>
          </summary>
          <dtstamp>

```

```

        <date-time>2008-02-05T19:12:24Z</date-time>
    </dtstamp>
    <tzname>
        <text>PST</text>
    </tzname>
    <dtstart>
        <date>2000-04-04T02:00:00</date>
    </dtstart>
    <dtend >
        <date>2000-04-04T02:30:00</date>
    </dtend>
    <location >
        <text>Location</text>
    </location>
    <attendee>
        <text>XYZ1@gmail.com</text>
    </attendee>
    <attendee>
        <text>XYZ2@gmail.com</text>
    </attendee>
    <uid>
        <text>4088E990AD89CB3DBB484909</text>
    </uid>
    <rrule>
        <recur>
            <freq>Weekly</freq>
            <byday>SU,TU,FR</byday>
            <interval>3</interval>
            <count>10</count>
        </recur>
    </rrule>
</properties>
</vevent>
</components>
</vcalendar>
</icalendar>

```

8.6. A recurring meeting which will be recurring Daily.

PLAI Visitor Management agent would call PUT to /PSIA/AreaControl/Meeting/info to pass the meeting information to the registered VMS. The sample XML is shown below for meeting occurring every 3 days for 5 recurrences. Count is the number of recurrences.

```

<icalendar xmlns="urn:iETF:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
    <calscale>

```

```

<text>GREGORIAN</text>
</calscale>
<prodid>
<text>-//Example Inc.//Example Calendar//EN</text>
</prodid>
<version>
<text>2.0</text>
</version>
</properties>
<components>
<event>
<properties>
<organizer>
  <text>Organizer@Contoso.com</text>
</organizer>
<summary>
  <text>Subject</text>
</summary>
<dtstamp>
  <date-time>2008-02-05T19:12:24Z</date-time>
</dtstamp>
<tzname>
  <text>PST</text>
</tzname>
<dtstart>
  <date>2000-04-04T02:00:00</date>
</dtstart>
<dtend >
  <date>2000-04-04T02:30:00</date>
</dtend>
<location >
  <text>Location</text>
</location>
<attendee>
  <text>XYZ1@gmail.com</text>
</attendee>
<attendee>
  <text>XYZ2@gmail.com</text>
</attendee>
<uid>
  <text>4088E990AD89CB3DBB484909</text>
</uid>
<rrule>
  <recur>
    <freq>DAILY</freq>
    <interval>3</interval>
    <count>5</count>
  </recur>
</rrule>
</properties>

```

```
    </vevent>
  </components>
</vcalendar>
</icalendar>
```

8.7. A recurring meeting which will be recurring monthly on a specific day each month(Absolute monthly).

PLAI Visitor Management agent would call PUT to /PSIA/AreaControl/Meeting/info to pass the meeting information to the registered VMS. The below example Occurs day 8 of every 3 months effective for 10 recurrences.

```
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>
            <text>Organizer@Contoso.com</text>
          </organizer>
          <summary>
            <text>Subject</text>
          </summary>
          <dtstamp>
            <date-time>2008-02-05T19:12:24Z</date-time>
          </dtstamp>
          <tzname>
            <text>PST</text>
          </tzname>
          <dtstart>
            <date>2000-04-04T02:00:00</date>
          </dtstart>
          <dtend >
            <date>2000-04-04T02:00:00</date>
          </dtend>
          <location >
```

```

        <text>Location</text>
      </location>
    <attendee>
      <text>XYZ1@gmail.com</text>
    </attendee>
    <attendee>
      <text>XYZ2@gmail.com</text>
    </attendee>
    <uid>
      <text>4088E990AD89CB3DBB484909</text>
    </uid>
    <rrule>
      <recur>
        <freq>MONTHLY</freq>
        <interval>3</interval >
        <bymonthday>8</bymonthday>
        <count>10</count >
      </recur>
    </rrule>
  </properties>
</vevent>
</components>
</vcalendar>
</icalendar>

```

8.8. A recurring meeting which will be recurring monthly(Relative monthly).

PLAI Visitor Management agent would call PUT to /PSIA/AreaControl/Meeting/info to pass the meeting information to the registered VMS. The relative monthly pattern causes an event to repeat on the same day of the week in the same relative position in the month, based on the number of months between each occurrence. The below example, show a meeting occurring Occurs the fourth Tuesday of every 1 month.

```

<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>

```



```

        <text>Organizer@Contoso.com</text>
    </organizer>
    <summary>
        <text>Subject</text>
    </summary>
    <dtstamp>
        <date-time>2008-02-05T19:12:24Z</date-time>
    </dtstamp>
    <tzname>
        <text>PST</text>
    </tzname>
    <dtstart>
        <date>2000-04-04T02:00:00</date>
    </dtstart>
    <dtend >
        <date>2000-04-04T02:00:00</date>
    </dtend>
    <location >
        <text>Location</text>
    </location>
    <attendee>
        <text>XYZ1@gmail.com</text>
    </attendee>
    <attendee>
        <text>XYZ2@gmail.com</text>
    </attendee>
    <uid>
        <text>4088E990AD89CB3DBB484909</text>
    </uid>
    <rrule>
        <recur>
            <freq>MONTHLY</freq>
            <bysetpos>4</bysetpos>
            <byday>TU</ byday >
            <count>10</ count >
        </recur>
    </rrule>
    </properties>
    </vevent>
    </components>
</vcalendar>
</icalendar>

```

8.9. A recurring meeting which will be recurring Yearly(Absolute yearly)

PLAI Visitor Management agent would call PUT to /PSIA/AreaControl/Meeting/info to pass the meeting information to the registered VMS.

The absolute yearly pattern causes an event to repeat on the same month and day based on the number of years between each occurrence. Example below Occurs every 3 years on June 8 for 10 occurrences

```
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>
            <text>Organizer@Contoso.com</text>
          </organizer>
          <summary>
            <text>Subject</text>
          </summary>
          <dtstamp>
            <date-time>2008-02-05T19:12:24Z</date-time>
          </dtstamp>
          <tzname>
            <text>PST</text>
          </tzname>
          <dtstart>
            <date>2000-04-04T02:00:00</date>
          </dtstart>
          <dtend >
            <date>2000-04-04T02:00:00</date>
          </dtend>
          <location >
            <text>Location</text>
          </location>
          <attendee>
            <text>XYZ1@gmail.com</text>
          </attendee>
          <attendee>
            <text>XYZ2@gmail.com</text>
          </attendee>
          <uid>
            <text>4088E990AD89CB3DBB484909</text>
          </uid>
        </properties>
      </vevent>
    </components>
  </vcalendar>
</icalendar>
```

```

    <rrule>
      <recur>
        <freq>YEARLY</freq>
        <interval>3</interval>
        <bymonthday>8</bymonthday >
        <bymonth>6</bymonth>
        <count>10</count>
      </recur>
    </rrule>
  </properties>
</vevent>
</components>
</vcalendar>
</icalendar>

```

8.10. A recurring meeting XML is sent ,which will be recurring Yearly(Relative yearly)

The relative yearly pattern causes an event to repeat on the same day of the week in the same relative position in a specific month, based on the number of years between each occurrence. For example the below XML shows a meeting Occurs the third Wednesday of May.

```

<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>
        <text>GREGORIAN</text>
      </calscale>
      <prodid>
        <text>-//Example Inc.//Example Calendar//EN</text>
      </prodid>
      <version>
        <text>2.0</text>
      </version>
    </properties>
    <components>
      <vevent>
        <properties>
          <organizer>
            <text>Organizer@Contoso.com</text>
          </organizer>
          <summary>
            <text>Subject</text>
          </summary>
          <dtstamp>
            <date-time>2008-02-05T19:12:24Z</date-time>
          </dtstamp>
          <tzname>
            <text>PST</text>
          </tzname>
        </properties>
      </vevent>
    </components>
  </vcalendar>
</icalendar>

```

```

<dtstart>
  <date>2000-04-04T02:00:00</date>
</dtstart>
<dtend >
  <date>2000-04-04T02:00:00</date>
</dtend>
<location >
  <text>Location</text>
</location>
<attendee>
  <text>XYZ1@gmail.com</text>
</attendee>
<attendee>
  <text>XYZ2@gmail.com</text>
</attendee>
<uid>
  <text>4088E990AD89CB3DBB484909</text>
</uid>
<rrule>
  <recur>
    <freq>YEARLY</freq>
    <byday>WE</byday>
    <bymonth>5</bymonth>
    <bysetpos>3</ bysetpos >
  </recur>
</rrule>
</properties>
</vevent>
</components>
</vcalendar>
</icalendar>

```

9. Appendix

Each calendar invite mandatorily requires an UID property. [RFC7986](#) provides several suggestions to generate a UID. UID can contain any random string to the max length of 255 characters.

9.1. PSIA XSD change

Introduced a new ICalUID type to ReferenceID in psiaCommonTypes.xsd

```

<xs:simpleType name="ICalUID">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The representation of a ICal UID as defined in RFC7986. Can contain any string to
      the max length of 255 characters
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">

```

```
    <xs:minLength value="0"/>
    <xs:maxLength value="255"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="ReferenceID">
  <xs:sequence>
    <xs:element name="ID" minOccurs="0" maxOccurs="1" type="LocalID" />
    <xs:element name="GUID" minOccurs="0" maxOccurs="1" type="GlobalID" />
    <xs:element name="UID" minOccurs="0" maxOccurs="1" type="ICalUID" />
    <xs:element name="Name" minOccurs="0" maxOccurs="1" type="xs:string" />
    <xs:element name="CustomExtension" minOccurs="0" maxOccurs="unbounded"
type="CommonTypeCustomExtension" />
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```