Overview

Advanced Identity Management allows you to represent your organization's structure within Matrix and Element: creating a space for all its members, maintaining their membership in rooms and subspaces, managing power levels and more.

It is composed of two main parts:

- **Bridges** connect to an existing data source (LDAP, Azure AD or others) and extract the list of users and groups from it.
  Multiple **Bridges** exists, and more can be added by implementing the `Bridge` interface (see `src/bridging`).
  See Bridging for more details.
- **Provisioner** takes directory produced by a bridge, maps it to matrix spaces (see Space mapping) and enforces its presence on a Matrix server — enforces meaning that it will both create and modify it as needed, but also act as an Matrix Application Service that will automatically react to changes on the Matrix server and check them against the rules established prior.
  Provisioner is ignorant of its data source — it is not aware of the **Bridge** being used and is merely fed data from it.
  See Provisioning for more details.

In addition to that, Advanced Identity Management is also an Application Service. The Provisioner observers the events reported by the AS in case it needs to enforce its rules on entities that it didn't itself create: for example demote a room creator to their expected power level (see LDAP as a source of truth).

Example configuration

```
provisioner:
  # Optional. A list of rooms that'll get automatically created in in managed space.
  # The ID is required to enable GPS to track whether they were already created or not
  # — you can change it, but it'll cause new rooms to be generated.
  default_rooms:
  - id: 'general'
    properties: { name: 'General discussion' }
  # Optional. A list of userid patterns that will not get kicked from rooms
  # even if they don't belong to them according to LDAP.
  # This is useful for things like the auditbot.
  # Patterns listed here will be wrapped in ^ and $ before matching.
  allowed_users:
  - '@adminbot:.*'
```

```
  # Optional. Determines whether users will be automatically invited to rooms (default, public and space-joinable)
  # when they gain access to them. Defaults to true. Users will still get invited to spaces regardless of this setting.
  invite_to_public_rooms: false
  # Optional: A list of remote Advanced Identity Management we'll be federating with. Requests from other remote users will be ignored.
  federation:
    federates_with:
    - '@aim_bot:consultancy.test'
  # Optional. When enabled, spaces that are no longer configured, and rooms belonging to those spaces will be cleaned up.
  # This will likely become enabled by default in the future.
  # When disabled (or omitted), GS will log the rooms and spaces it would clean up if allowed to.
  gc:
    enabled: true

# Optional. If configured, Advanced Identity Management will synchronize user accounts (attributes and account validity)
# found in the data directory to the specified list of targets.
userProvisioner:
  # Optional. Configure to enable user deprovisioning. Disabled by default.
  deprovisioning:
    enabled: false
    # Optional. When users get removed from the directory their accounts will only be deactivated,
    # but their erasure will be delayed by the specified time period, allowing them to be reactivated in the meantime.
    # The format is <amount><unit>, with amount being numeric and unit being one of: [s, m, h, d], for seconds, minutes,
    # hours or days respectively (for example: "24h", "31d" etc.).
    # The specified period will be translated into seconds, so won't account for things like DST, leap seconds etc.
    # Users will be deleted *no sooner* than that, but may be removed a bit later, depending on other Advanced Identity Management operations.
    # By default set to 30 days.
    soft_delete_period: '30d'

# Configure the spaces you want Advanced Identity Management to manage on your Matrix server
spaces:
```

```
  # The internal ID of this space. Don't change it after you set it, or it will create a new
one and abandon the old one.
- id: main
  # The display name of the space, safe to change later.
  name: 'My Company'
  # The list of groups from your user directory to add as members to this space.
  # An empty string is a special name that means "all available users, regardless of group
memberships".
  # You can set a power level for any of the groups. By default it's 0.
  # This space is going to contain all the users in the directory,
  # and those that are present in the "managers" groups will be the moderators (in the space
and its child rooms).
  groups:
    - externalId: ''
    - externalId: 'cn=managers,ou=employees,dc=element,dc=test'
      powerLevel: 50
- id: management
  name: 'Management'
  groups:
    - externalId: 'cn=managers,ou=employees,dc=element,dc=test'
  # You can confgure spaces that'll be federated between multiple GS servers.
  # Each Advanced Identity Management will only manage its local users.
- id: shared
  name: 'Federated space'
  groups:
    - externalId: 'cn=engineering,ou=employees,dc=element,dc=test'
  federatedGroups:
      # The external ID of the group on the foreign server.
      # This will be enforced by the Advanced Identity Management running on consultancy.test,
not this instance configured here.
      # The Advanced Identity Management running on consultancy.test needs to have our MXID
      # (@gpsbot:element.test by default) configured in its provisioner.federates_with config
option.
    - externalId: 'ou=element-contractors,dc=consultancy,dc=test'
      # The MXID of the remote Advanced Identity Management bot.
      agent: '@aim_bot:consultancy.test'

source:
  type: 'ldap'
  # LDAP will be checked for changes every this many seconds
  check_interval_seconds: 60
```

```
  # The following can be copied straight from Synapse's homeserver.yaml
  # if you're already using its LDAP password provider
  uri: "ldap://element.test"
  # The base ou we specify here will become the root space
  base: "ou=employees,dc=element,dc=test"
  # Optional. An LDAP filter to use when searching for entries
  filter: '(!(ou=Domain Controllers))'
  # Make sure the account you use here has enough permissions to perform searches on your
`base`
  bind_dn: "ELEMENT\\administrator"
  bind_password: "donkey.8"
  # Needs `uid` to be able to determine Matrix localparts for users
  # and `name`s to pick the right names for spaces
  attributes:
    uid: "sAMAccountName"
    name: "name"
  # If the LDAP server requires a client certificate, enable this option.
  # cert:
    # The path to the file
    # file: "./my-cert-file.pem"
    # OR the PEM-encoded cert itself
    # cert: "foobar"
    # Passphrase for the cert, if required.
    # passphrase: "passphrase"

# For Microsoft Graph:
# source:
#   type: 'ms-graph-ad'
#
#   # This is the "Tenant ID" from your Azure Active Directory Overview
#   tenant_id: 'b9355cb3-feed-dead-beef-9cc325f0335b'
#
#   # Register your app in "App registrations". This will be its "Application (client) ID"
#   client_id: '5c955b66-18b3-42de-bb5a-13b5a202d4fc'
#
#   # Go to "Certificates & secrets", and click on "New client secret".
#   # This will be the "Value" of the created secret (not the "Secret ID").
#   client_secret: 'yOb7Q~Km~~YMKzpeq73swJj3kOeJpUwXSZamr'
#   # For the bridge to be able to operate correctly, navigate to API permissions and unsure
#   # it has access to GroupMember.Read.All and User.Read.All
#   # Application permissions for Microsoft Graph. Remember to grant the admin consent for
```

```
those.
#
#   # Optional. The url to reach Graph on. Override if your deployment uses a specific graph
endpoint.
#   base_url: 'https://graph.microsoft.com/'
#
#   # Optional. Specific scopes to set for graph to use.
#   scopes: ['https://graph.microsoft.com/.default']

# For SCIM:
#   type: 'scim'
#   # HTTP port that the SCIM server will listen on
#   port: 8040
#   # Optional URL prefix for all routes
#   base_url: '/scim/v2'
#   client:
#     # Unique ID for the SCIM client.
#     # This will be used to keep track of the managed Space and User/Group storage in Matrix.
#     id: 'element-ad'
#     # You can set up multiple client tokens with different permission levels.
#     rbac:
#         # Bearer token for the client, as per RFC 6750
#       - token: 'foo-bar-baz'
#         # What's the token allowed to do: in this case, everything (read+write on all
endpoints).
#         # The format for these is 'access:scope', access being 'read', 'write' or '*' for
both,
#         # scope being 'users', 'groups' or '*' for everything.
#         roles: ['*:*']
#         # You can specify permissions for anyone who presents a valid Matrix access_token
for an admin user
#       - synapse_user: 'admin'
#         # ...and assign more fine-tuned permissions to it
#         roles: ['read:*', 'write:groups']
#     attributeMapping:
#       # The SCIM user attribute that'll be used as the Matrix username for provisioned users
#       username: 'externalId'
#   # Should SCIM user creation register a Matrix account for the user.
#   # Possible values are 'yes', 'no' and 'if-missing'
#   # - 'yes' will register Matrix accounts on the server upon a SCIM create user request,
#   #   and error out if the user with that username already exists.
```

```
#   # - 'if-missing' will register Matrix accounts unless they exist already.
#   #   This is useful if some users have their user accounts created independently before the
SCIM bridge was set up.
#   # - 'no' will not create user accounts, only work with existing ones.
#   register_users: 'no'
#   # Optional: Should SCIM responses wait for Matrix provisioning to complete.
#   # It is recommended to leave it as false. HTTP responses will be sent quicker,
#   # and Matrix provisioning may still fail in the background (to be retried later).
#   synchronous_provisioning: false
#   # Optional: Configure a mailer to send email notifications to newly registered, activated
and deactivated users.
#   # mailer:
#   #   # The email address emails will be sent from
#   #   from: 'element@element.com'
#   #   # Path to a directory with email templates.
#   #   # Each template should be a directory containing 'subject.pug', 'text.pug' and
'html.pug',
#   #   # all using https://pugjs.org/ as a template language.
#   #   # Advanced Identity Management ships with standard, Element-branded templates in
templates/
#   #   templates_path: './templates'
#   #   # SMTP transport configuration, as per https://nodemailer.com/smtp/,
#   #   # except that we default `secure` to `true` and `port` to 465.
#   #   transport:
#   #     host: 'smtp.example.com'
#   #     auth:
#   #       user: 'mailer'
#   #       pass: 'mailerpass.8'

# Optional. Configure this to gather usage statistics.
# See telemetry spec at https://gitlab.matrix.org/new-vector/modular/telemetry-schema
# for details on what's being gathered and sent.
telemetry:
  # Identifier of this Advanced Identity Management instance
  instance_id: 'foo'
  # Every this many seconds (and on startup) telemetry will be recorded (and optionally sent)
  send_interval: 3600
  # Optional: the EMS endpoint to submit telemetry entries to.
  # This is optional as it wouldn't work for airgapped environments,
  # and by default no telemetry is sent (but it is still gathered).
  endpoint: 'https://ems.com/telemetry'
```

```
  # Optional: how many times should we retry sending telemetry if it fails. Defaults to 3
  retry_count: 3
  # Optional: how long should we wait between retries. Defaults to 60, in seconds
  retry_interval: 60

# Optional
logging:
  # Allowed levels are: error, warn, info, http, verbose, debug, silly - case sensitive.
  # "info" will typically notify of all "write" actions (affecting the state of the homeserver),
  # while "debug" will also be reporting checks performed that didn't result in any changes.
  level: "info"
  # Optional. Allowed formats are are:
  # - pretty: the default. A timestamped, colorized output suitable for humans
  # - json:   logging a json object containing a `level`, `message`, `timestamp` and optionally a `label`
  format: "json"
```

Revision #5
Created 2025-05-15 09:21:52 UTC by Gaël Goinvic
Updated 2025-05-27 13:12:00 UTC by Gaël Goinvic