

Ideas for midPoint development

- [Progressive user profile maintenance](#)
- [Self-service management of orgs](#)
- [User relationships](#)
- [Smart identity merge/split](#)
- [See Also](#)

This page summarizes ideas that were not yet "shaped" into a [planned feature](#).

Progressive user profile maintenance

Typical enterprise use case is that the IDM will get almost all user profile data from the HR. But that is not the case for other environments, such as customer IDM or academia. In that case we usually want to collect just a minimum amount of information from the user at the beginning. Yet there may be services that require more personal data.

The idea is to create user profile as needed in a "progressive" way. E.g. when user requests a service that requires additional data item, the user has to enter that additional data item as part of the request. Similarly when access to a system was assigned to a user, the user has to manually enter the data item (and possibly consent to data processing).

This feature is partially implemented by the ability to include custom form in [approval](#) GUI that can be used to fill out user profile. But that is not yet a complete feature as this is only aimed at administrators. There is no support for user to fill out the missing data.

This also relates to [Management of Lawful Bases for Data Processing \(GDPR\)](#) and [Data Provenance](#), e.g. we may need to make erasure of the entered data once there is no longer need to process that data.

Self-service management of orgs

MidPoint has quite a nice set of features for organizational management and especially delegated organizational management (see [Authorization](#)). However, the user interface is mostly concerned about enterprise use cases and power users that maintain the organizational structure.

The idea is to make org management more "democratic" or "ad-hoc". E.g. we would like an ability for every user to create an ad-hoc work group. The user that creates the group will become an owner and manager, he can add more owners and managers, add members and so on. This process should be lightweight, it should scale well and it should be available for almost all ordinary users.

The tricky part is that those ad-hoc groups may need special parameters or data. But that can be solved by improvements to parametric roles and/or archetypes.

User relationships

MidPoint is good to maintain formal relationships, such as owner or manager. Those are used a lot in enterprises and they are essentially the foundation of any organizational identity management. However, there are also informal and self-managed relationship between users, such as the relation of "collaborator", "friend" or "family member". Those relations may be essential for future identity management, e.g. ad-hoc collaboration on a project, cooperation on non-business activities, parents paying bills for children and so on.

The users should be able to set up those relationships with existing users, invite new users and so on.

MidPoint should allow to set up and manage those relationship. This is not that unlike the way how social networks work. However, the goal is quite different. We do not want to entertain the users and sell them ads. We want to use those relationships to allow real cooperation.

Note: there is a data protection aspect to consider.

Smart identity merge/split

Traditional IDM approach is to gather all identity data, correlate them with existing database and decide about identity matching at the time of new user enrollment.

But that method will cease to work as we will go deeper to progressive user profiles. We may not have enough data to correlate a user at the beginning - and we may not even want to do it (e.g. for data protection reasons).

The right moment to merge identities may come way later. At that moment the identities may be operating independently for quite some time. Therefore we may need to do some kind of smart identity merge, most like a user-assisted identity merge. E.g. user may "proof" the merge by logging in into both accounts.

There will be mistakes in the merge process. Therefore there may be a need for identity split. While merge will not be easy, split may be even more difficult.

See Also

- [Planned Features](#)