

Groups member contacts

Problem

Ability to send direct messages to group members, w/t creating additional direct connections, while keeping existing UX of separate conversations.

Solution

Protocol

Same changes on chat protocol level as for direct messages:

```
```haskell data MessageScope = MSGroup | MSDirect
-- ExtMsgContent extended with scope data ExtMsgContent =
ExtMsgContent { ... scope :: Maybe MessageScope } ```
```

Changes to MsgRef are not necessary - it would be impossible to quote group message directly to member contact.

### Model

If member doesn't have existing contact (e.g. legacy, merged or "member contact" created before), button "Send direct message" / "Open direct chat" in UI would create a new "member contact" record in contacts table that would use the same connection as group member.

New API:

```
haskell APICreateMemberContact GroupId GroupMemberId
```

- would create a new contact record and assign contactid to groupmember record
- should be unmergeable so that the same connection is not used for member across groups
  - flag
  - or groupmemberid in contacts table
- member removal (member leaves or removed) and deletion (group is deleted) should check if "member contact" exists for the same connection, if yes connection should be kept
- contact deletion should also check if member exists
- due to ON DELETE CASCADE constraints entity id should be first set to null before deleting contact/member record

On receiving group message with MSDirect scope, if "member contact" doesn't exist it should be created (same as above).

What if member record already had regular contact assigned? (a contact with a separate connection and no groupmemberid) It can happen if merge completed for these members previously, and one of the members deleted contact; or if merge has only completed for one of members.

- Assigning chat items received as "member contact" to a regular contact would mix messages from different connections in the same conversation. This would practically be indistinguishable in UI, but would further complicate understanding of connection level errors.
- Replacing contactid for groupmembers record would seem to user as if previous messages were lost when chat is entered via "open direct chat" button, unless there's an indication inside this new chat that an old contact exists separately (requires yet additional field on contact - maincontactid?).

When next messages are received read connection entity based on message scope? (parameterize getConnectionEntity, toConnection with message scope). Move message scope on a level above ExtMsgContent? If it's on AChatMsgEvent level - this would allow only MSG agent messages and not make it a fully fledged connection entity. Should "member contact" be able to receive any messages other than XMsgNew and messages referring by shared msg id? If not, maybe it shouldn't be treated as connection entity and instead "member contact" record should be read ad-hoc when processing these messages (for example see processGroupScopeMsg for group direct message). For example, should "member contacts" be available for inviting to other groups? Probably not - if they were, connection established in one group would be reused in another group.

We could also make distinction between regular contacts and "member contacts" in chat list - e.g. show both group and member avatar/icon, or include group name in conversation name. This would also improve clarity for users which contacts were created for which reason and to better understand consequences to deleting it.

TODO:

We should also double check that removed members messages are dropped if received - right now their connections are deleted so it's practically not possible for them to send after being removed; if their connection is kept for "member contact" purpose though, they would still be able to send group messages.