Chat history deletion

Problem

Currently chat history is preserved indefinitely and can only be cleared manually, either individual messages or per chat. This leads to privacy concerns as well as growing memory consumption.

Solution

An option to turn on scheduled deletion of chat history (chat items and files). Initially only as a global setting but we can also consider setting per conversation.

Implementation plan

Scheduled deletion implementation plan:

- Enum ChatItemTTL - None, Day, Week, Month, etc.
- Functions to convert ChatItemTTL to number of seconds for chatItemTTL and expireChatItemsInterval
- Interval can be:
  - TTL / 2
  - depend on TTL (e.g. 1 day for a 1 week TTL) (reuse ExpirationConfig and have fixed configs?)
  - fixed - 30 min?
- iOS is not a long running process so we have to check after start
- To prevent NSE from running this process parameterize startChat to allow starting without scheduled deletion even if it is configured
- Don't update chats and previews?

Chat global expiration

Api:

- API Command - SetChatItemTTL ChatItemTTL, response is CRCmdOk
- API Command - GetChatItemTTL, response is CRChatItemTTL

UI:

- New view in settings, on start GetChatItemTTL to load into model
- When changed in UI - SetChatItemTTL, update in model
- UI options match ChatItemTTL

Core:

- Add expireChatItems to ChatController: TVar (Maybe (Async ())) similar to agentAsync?
- Thread is created/stopped in runtime because interval has to be figured out dynamically when TTL is changed (e.g. if it was changed from 1 week to 30 mins and interval for 1 week is 1 day, we shouldn't wait 1 day before reading new interval)
- Add table settings, field chatitemttl
- On chat start - read settings, convert chatitemttl into chatItemTTL and expireChatItemsInterval (may be Nothing); if not Nothing - run expireMessages thread and put into controller
- On SetChatItemTTL - update settings
  - If Nothing - cancel expireMessages, remove from controller, update setting in store
  - If Just - start expireMessages, put into controller, update setting in store
- expireMessages thread: forever $ do threadDelay interval expiration logic
- Expiration logic:
  - Select all (chat ref, chat item id) older than (current time - TTL), comparing with updatedat (createdat?)
  - Reuse logic from APIDeleteChatItem to delete each item (should messages be deleted or updated to XMsgDeleted?)

Questions

- single thread (don't re-create on change), read flag on each cycle and on each chat item
- if ttl changed from none to some value - first run sync, no delay between chat items on first run
- seconds instead of enum in api / backend
- part of APISetChatSettings api? - unclear can block for long on first deletion
- fixed interval
- if ttl became smaller, set flag to false, then one sync cycle

Per chat expiration

API:

- API Command - SetChatCITTL ChatRef ChatItemTTL, response is CRCmdOk
- API Command - GetChatCITTL ChatRef, response is CRChatItemTTL
- If we do both global and contact API can be SetChatItemTTL (Maybe ChatRef) ChatItemTTL or SetChatItemTTL GlobalOrChatRef ChatItemTTL, same for Get

UI:

- In UI - in ChatInfo views, loaded on opening

Core:

- Add expireChatCIs in ChatController: map [ChatRef, Async ()]
- Added and started/cancelled by chatRef
- Saved in contacts/groups tables
- On chat start - read from contacts/groups
- Expiration logic: select per chat