

Use `TextArea` for Chat Message Content

Context and Problem Statement

This decision record concerns the UI component that is used for rendering the content of chat messages.

Decision Drivers

- Looks good (renders Markdown)
- User can select and copy text
- Has good performance

Considered Options

- Use `TextArea`
- Use a [third-party package](#)
- Use a Markdown parser and convert AST nodes to JavaFX `TextFlow` elements
- Use a Markdown parser to convert content into HTML and use a `WebView` for one message
- Use a Markdown parser and `WebView` for the whole chat history

Decision Outcome

Chosen option: “Use `TextArea`”. All other options require more time to implement. Some of the options do not support text selection and copying, which for now we value more than Markdown rendering.

Pros and Cons of the Options

Use `TextArea`

- Good, because it is easy to implement
- Good, because it supports text selection and copying
- Bad, because it does not offer rich text. Thus, Markdown can only be displayed in a plain text form.
- Bad, because default JavaFX’s `TextArea` shrinks

Use a third-party package

There seems to be [only one package](#) for JavaFX that provides a ready-to-use UI node for Markdown rendering.

- Good, because it is easy to implement
- Good, because it renders Markdown
- Good, because it renders Markdown to JavaFX nodes (does not use a `WebView`)
- Good, because complex elements from Markdown are supported (tables, code blocks, etc.)
- Bad, because it has very strange issues and architectural flaws with styling
- Bad, because it does not support text selection and copying (because of underlying JavaFX `Text` nodes)

Use a Markdown parser and convert AST nodes to JavaFX TextFlow elements

- Good, because we will support Markdown
- Good, because no need to write a Markdown parser from scratch
- Good, because does not use a `WebView`
- Good, because easy styling
- Bad, because we need some time to implement Markdown AST -> JavaFX nodes converter
- Bad, because rendering tables and code blocks may be hard
- Bad, because it will not support text selection and copying

Use a Markdown parser to convert content into HTML and use a `WebView` for one message

- Good, because there are libraries to convert Markdown to HTML
- Good, because may be easier to implement than other choices (except `TextArea`)
- Good, because it supports text selection and copying
- Bad, because it may be a problem to connect JavaFX CSS to `WebView`
- Bad, because one `WebView` for one message is resourceful

Use a Markdown parser and `WebView` for the whole chat history

- Good, because there are libraries to convert Markdown to HTML
- Good, because it supports text selection and copying
- Bad, because it may be a problem to connect JavaFX CSS to `WebView`
- Bad, because it may be a problem to correctly communicate with Java code and `WebView` to add new messages

More Information

This ADR is highly linked to [ADR-0042](#).

Actually we used an `ExpandingTextArea` from `GemsFX` package so the content can occupy as much space as it needs in the `ScrollPane`.

About the selection and copying, this goes down to fundamental issue from JavaFX. `Text` and `Label` as a whole or a part [cannot be selected and/or copied](#).
