* Mark Anderson, system manager of EDN, said that embedding the chatbot into EDN would require a Redmine plugin ([tutorial](https://www.redmine.org/projects/redmine/wiki/Plugin_Tutorial)). This would have to be written in [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails).

*Your team would not be allowed to touch the EDN software. That would eventually be up to the staff to do (meaning me). If we did, yes, it would be a Redmine plugin.*

*For the moment, as with all projects, please focus on functionality rather than implementation specifics.*

—Mark Anderson, Jan 13 2022

* Has any team member used Ruby on Rails? Even if not, I wouldn’t rule it out. As Mark Anderson said, we won’t start implementation for a while, so I wouldn’t mind learning the [language](https://www.ruby-lang.org/en/) and [framework](https://rubyonrails.org/) in the meantime and as we go. They are widely used and documented.
* However, this would likely not beat out the Webex bot option (JavaScript on Node.js), which would be slightly less user-friendly but is much more familiar to the team. **Discuss in class.**
  + In fact, Webex bots use a REST API, so we could use any language to make a compatible web server.
  + The above options use webhooks to connect the bot server to Webex. A possibly better alternative is websockets, supported by the JavaScript framework [Node Flint](https://www.npmjs.com/package/node-flint) and the Python package [webex\_bot](https://github.com/fbradyirl/webex_bot) ([tutorial](https://0x2142.com/how-to-building-a-basic-webex-chatbot/)).
* Integrating Node.js JavaScript code into Rails is [impractical](https://stackoverflow.com/a/3527783).