

USER AGENTS:

1. BOCA_USER
- a. Sends request to t5_base_hf agent.

b. Handles responses from the t5_base_hf agent.

c. Sends boca_message to chat partner upon response messages

NON-USER AGENTS:

1. BOCA_MATCH_MAKER (Hosted Service)
- a. Matches chat participants

b. Handles queue of participant chat requests
2. T5_BASE_HF (Hosted Service)
- a. Handles translation requests and responses to-and-from the remote HF model.

PROTOCOLS:

1. boca_match_maker
2. t5_base

MESSAGES:

1. MatchMakerRequest
- a. Request chat to match_maker
2. MatchMakerResponse
- a. Request chat response from match_maker
3. MatchMakerUpdateRequest
- a. Update match request to match_maker
4. MatchMakerUpdateResponse
- a. Update match request response from match_maker
5. TranslationRequest
- a. Translation request to t5_base
6. TranslationResponse
- a. Translation request response from t5_base
7. Error
- a. Error response from t5_base

8. BocaMessage
- a. Message sent to chat partner agent from user's agent

Flow:

1. The user enters the BocaLibre website.
2. The user connects their Fetch.ai wallet.
3. The user adds the following environment variables:

a. Native language (the language they will be typing in)

b. Desired/Translation language (optional)

c. Their HuggingFace API Token
4. The user starts their **BOCA_USER** agent.
5. The user makes a chat request to the **BOCA_MATCH_MAKER** agent.

a. If the Desired/Translation language was supplied by the user, the match maker will look for a partner with the corresponding native language, otherwise, the agent will match the user with a random partner with a different native language.
6. The **BOCA_MATCH_MAKER** agent matches two user's who have matching criteria to initiate a chat between the two agents.
7. The **BOCA_MATCH_MAKER** agent responds to the user's agents with the address of the user's agent with whom they have been matched.

a. Once the match request has been fulfilled, the two agents are cleared from the **BOCA_MATCH_MAKER** agent's storage, so it does not continuously match the same user(s).

i. This means that each distinct user will have to make a new request each time they wish to be paired with a new chat partner.
8. The user's **BOCA_USER** agent stores the chat request response in its internal storage.
9. A chat is started between the two partners.
10. A user inputs some text as a message to their chat partner.
11. The user's agent sends a TranslationRequest to the hosted **T5_BASE_HF** agent with the appropriate headers.
12. The **T5_BASE_HF** agent receives the request, and makes a subsequent **TranslationRequest** to the remote HuggingFace t5-base model.
13. The **T5_BASE_HF** agent awaits a reponse from the model.
14. Upon receiving the translation response from the remote HF model, the **T5_BASE_HF** agent sends the **TranslationResponse** message to the user's **BOCA_USER** agent.
15. Upon receiving this **TranslationResponse**, the user's **BOCA_USER** agent sends a **BocaMessage** to their chat partner.
16. The partner's **BOCA_USER** agent receives the **BocaMessage**, stores the message, and logs/displays it.

