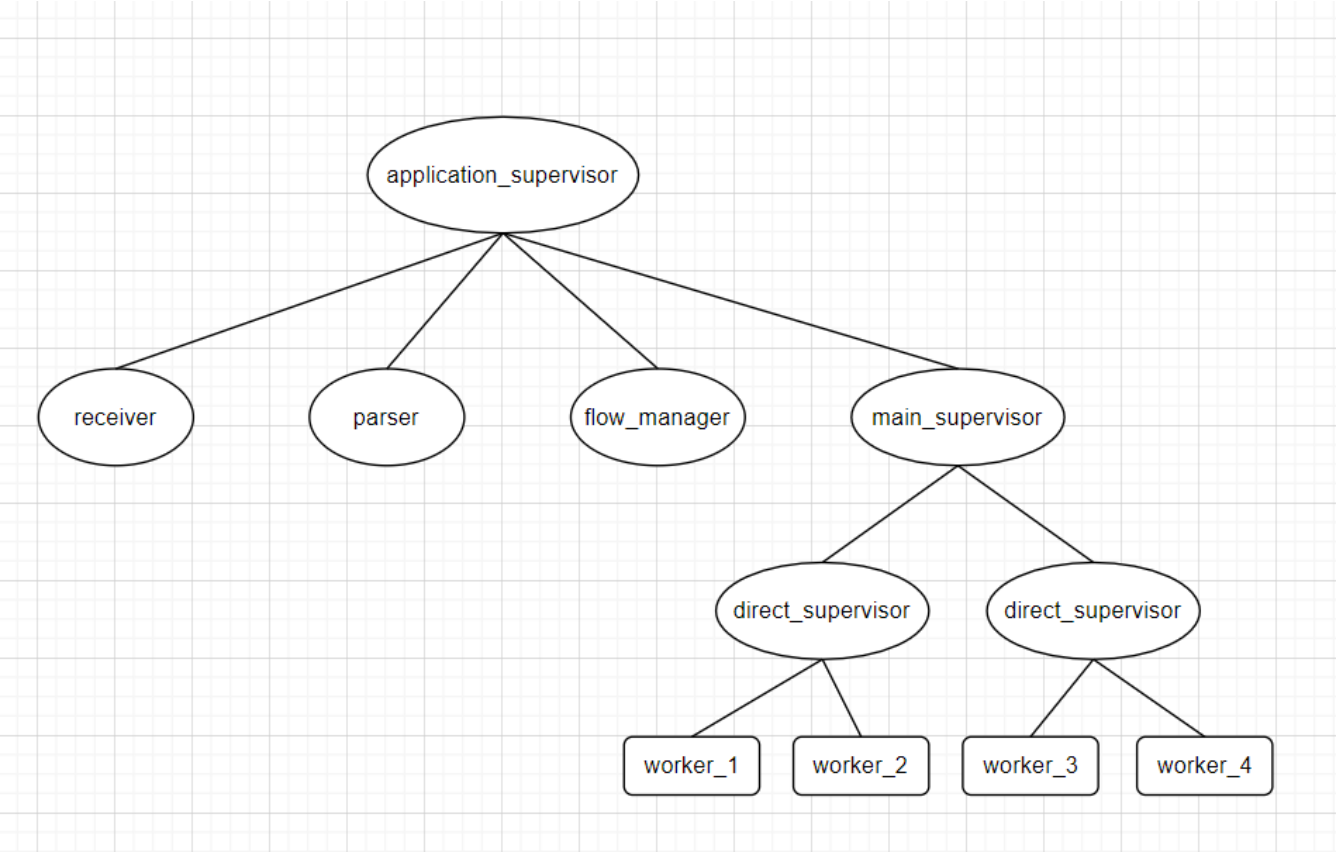
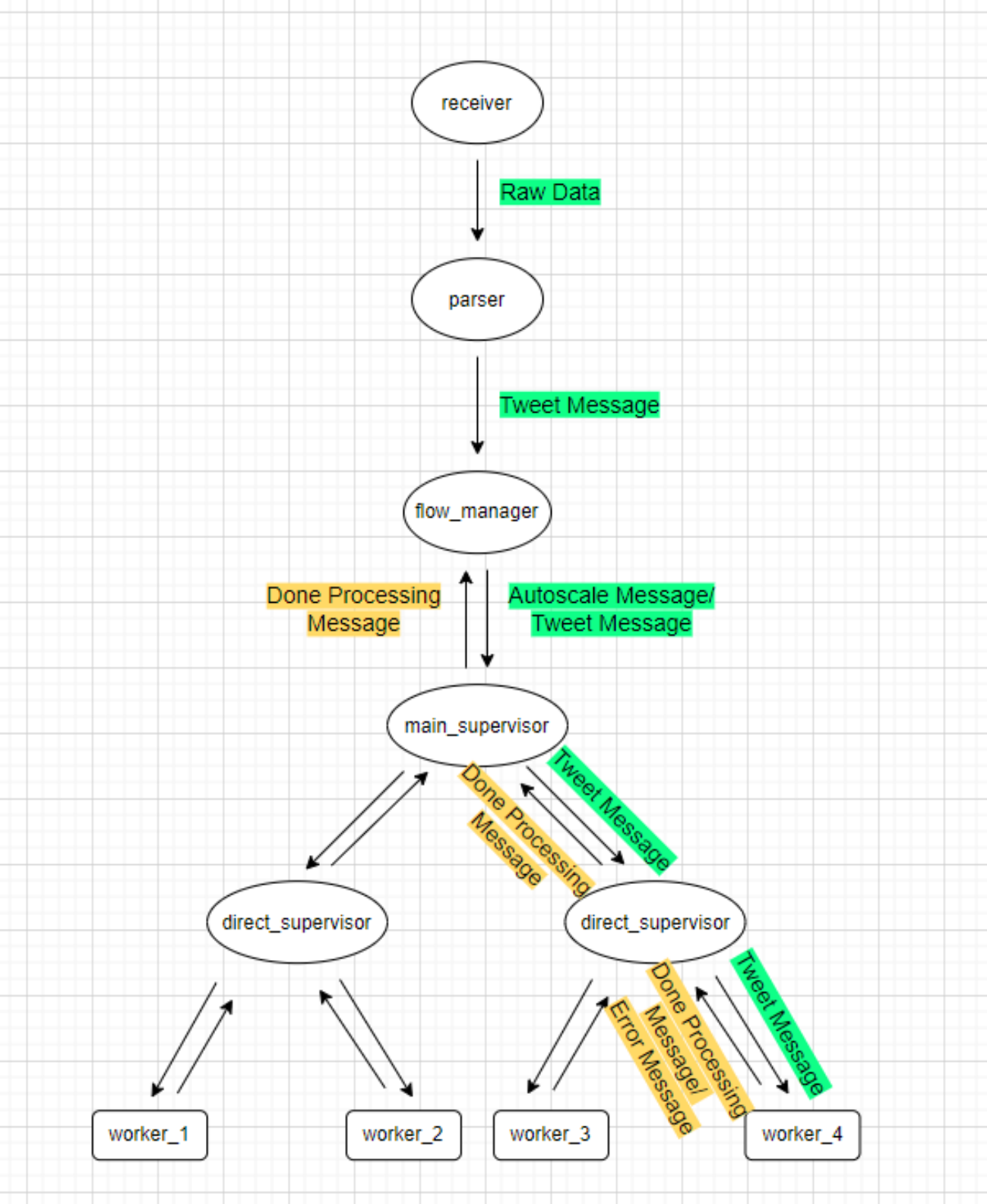
PTR checkpoint nr.1

Supervision tree:



Message flow:



Actors:

receiver + parser:

- They can be taken as a single actor (this will be decided later).
- Receive SSE stream data and does parsing in order to get operable data .

flow_manager:

- receives parsed messages
- containts counter with the nr of messages in process (based on that will autoscale)
- responsible for autoscale (will send messages to main_supervisor with how many “sub-supervisors” should be provided)

main_supervisor

- receives messages and load balances them using Round Robin strategy
- will autoscale (add/remove children actors) based on the messages from the flow_manager

direct_supervisor

- responsible for sending the messages to the workers
- responsible for handling the stops
- can be scaled

worker

- responsible for printing the message or stopping
- sends back a message when the tweet is processed
- sends back an error message
- can be scaled

Used language: *Elixir*

Short description + reflections:

The SSE stream information is processed by the receiver and parser components. They provide operable data (messages). The flow_manager receives the messages and modifies the counter based on the nr. of received messages. Also, based on that, it decides whether the main_supervisor should autoscale. By autoscaling, means creating a new direct_supervisor with some workers or deleting one. The message is sent to the direct_supervisor and then to the worker.

The thing about the direct_supervisor is that it was added as an additional layer in case that a worker fails so that it won't affect the main_supervisor and the other workers, but my perception may be wrong. As I know, in Elixir exists the restarting strategy: one_for_one which should not affect others and this may get rid of the direct_supervisor, but I think it will be clearer in the development process.

So, after a tweet is processed, a message is sent back to the flow_manager in order to decrease the counter and based on that perform again the necessary scaling.