Use Case Diagram of Uploader

Uploader API

- __init__
- start
- stop
- get_report
- is_active
- wait_for_uploaded

API end-user

is_active

final uploading report

Uploader implementation

Uploader Instance

Threads of the main process

- imap_next_thread
- processing_thread

Processes pool

Upload processes

- Process 1
- Process 2
- Process N
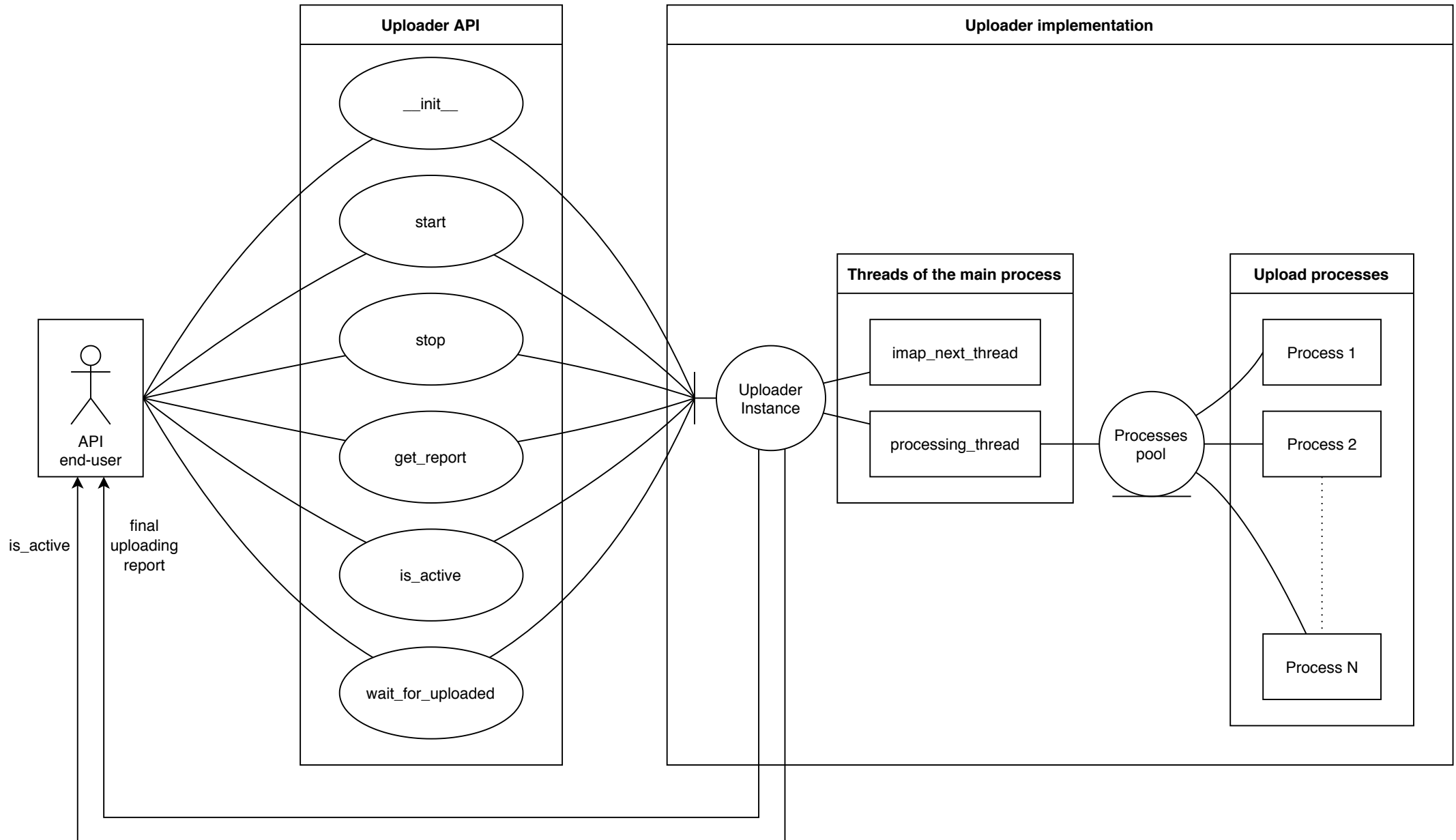
Flowchart of imap_next_thread

imap_next_thread input data
iter queue
result queue

imap_next_thread

START

wait for new value in iter_queue

get value from iter_queue

value is None

TRUE → END

FALSE

put next(value) to result_queue
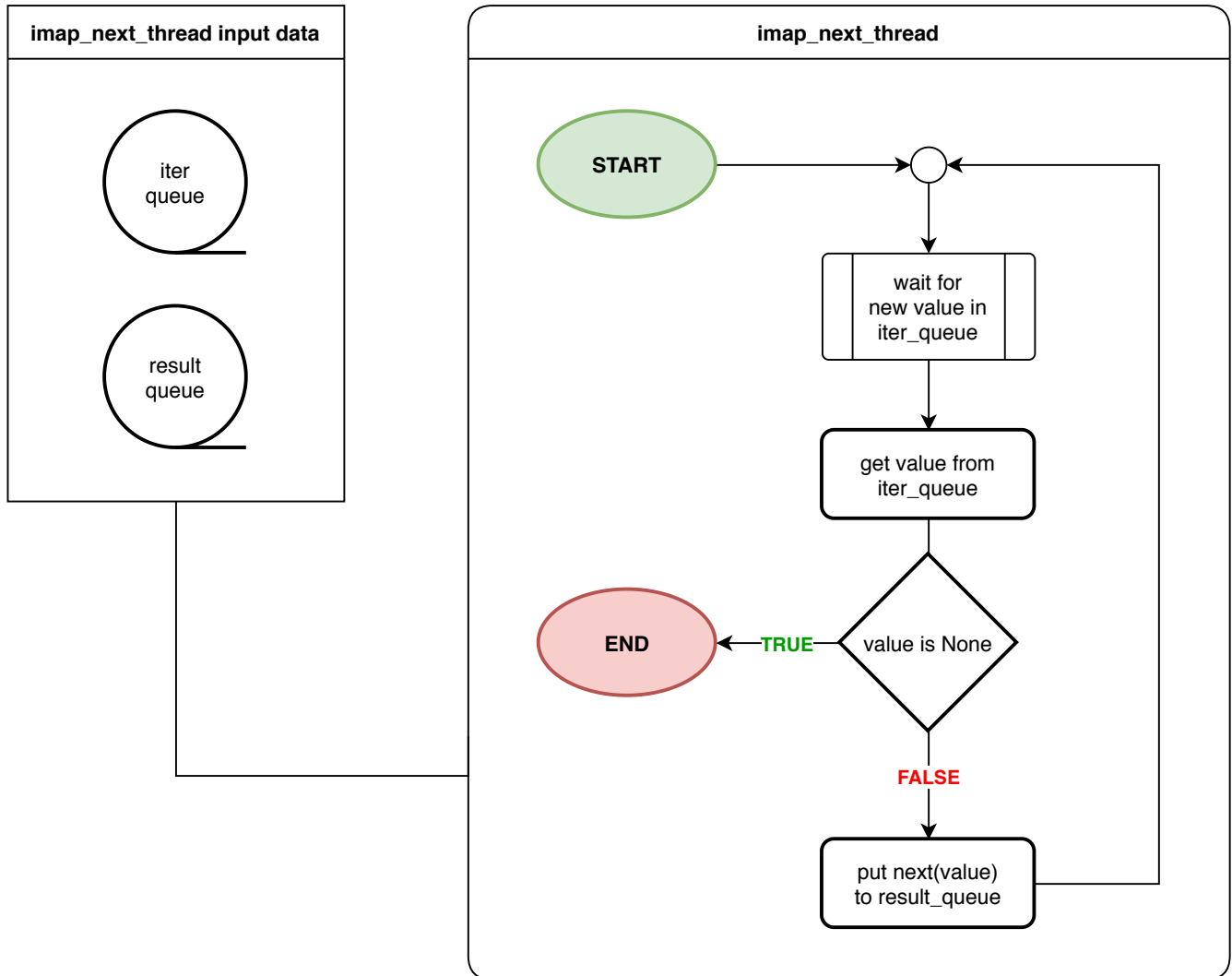
Flowchart of processing_thread

processing_thread input data

- filenames
- number of processes
- loading progress queue
- iter queue
- result queue
- stop event

processing_thread

START

unset stop_event

init progress_value

init processes_pool

processes_pool.imap to result_iter

○

stop_event is set — **TRUE** →

FALSE

put result_iter to iter_queue

○

result_queue is empty — **TRUE** → stop_event is set — **FALSE** ↑

FALSE

get result from result_queue

result is None — **TRUE** → put progress_value to loading_progress_queue → set stop_event

FALSE

result[1] — **TRUE** → ++progress_value.done → append result[0] to uploaded_files

FALSE → ++progress_value.error

put progress_value to loading_progress_queue

TRUE (from stop_event is set) → ○

put None to iter_queue

close processes_pool

END