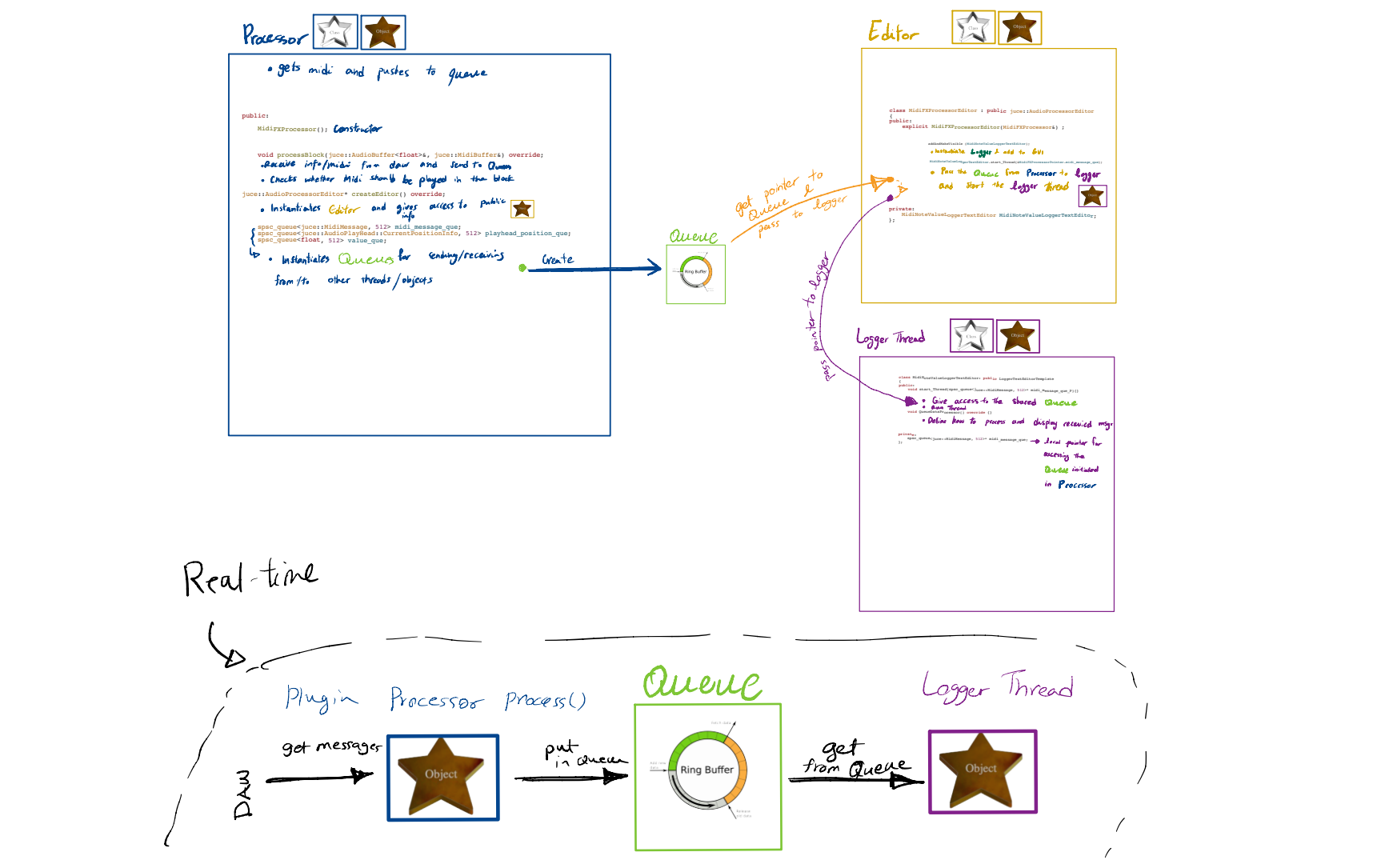
Processor
• gets midi and pastes to queue
public:
MidiFXProcessor(); Constructor
void processBlock(juce::AudioBuffer<float>&, juce::MidiBuffer&) override;
• Receive info/midi from daw and send to Queue
• Checks whether Midi should be played in the block
juce::AudioProcessorEditor* createEditor() override;
• Instantiates Editor and gives access to public info
spsc_queue<juce::MidiMessage, 512> midi_message_que;
spsc_queue<juce::AudioPlayHead::CurrentPositionInfo, 512> playhead_position_que;
spsc_queue<float, 512> value_que;
• Instantiates Queues for sending/receiving from/to other threads / objects
Create
Queue
Ring Buffer
get pointer to Queue & pass to logger
Editor
class MidiFXProcessorEditor : public juce::AudioProcessorEditor
public:
explicit MidiFXProcessorEditor(MidiFXProcessor&) ;
• Instantiate Logger & add to GUI
• Pass the Queue from Processor to Logger and start the Logger thread
private:
MidiNoteValueLoggerTextEditor MidiNoteValueLoggerTextEditor;
pass pointer to logger
Logger Thread
• Give access to the shared Queue
• Runs Thread
• Defines how to process and display received msgs
local pointer for accessing the Queue initialized in Processor
Real-time
Plugin Processor process()
DAW
get messages
Object
put in queue
Queue
Ring Buffer
get from Queue
Logger Thread
Object