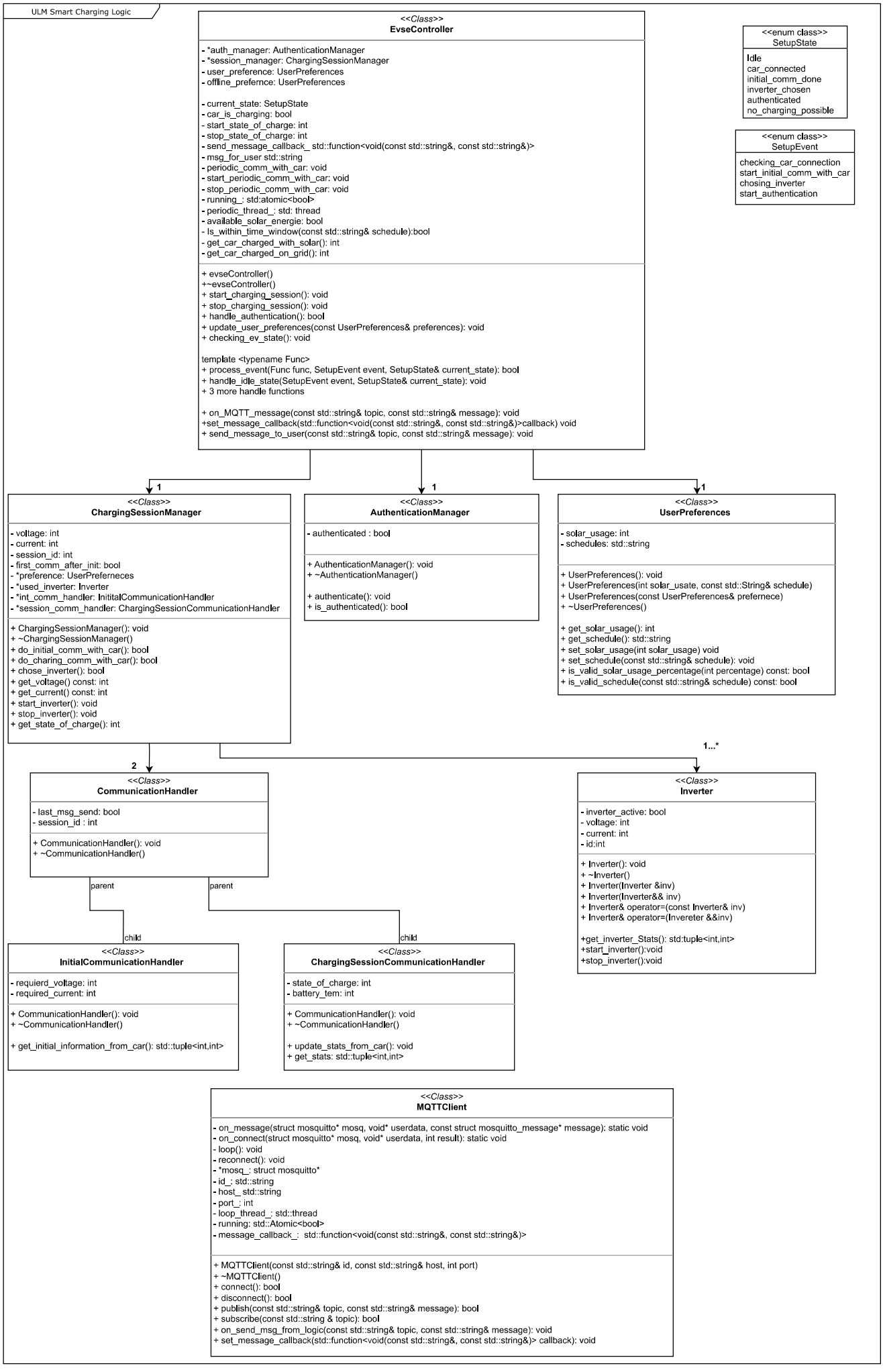
ULM Smart Charging Logic
<<Class>>
EvseController
- *auth_manager: AuthenticationManager
- *session_manager: ChargingSessionManager
- user_preference: UserPreferences
- offline_prefernce: UserPreferences
- current_state: SetupState
- car_is_charging: bool
- start_state_of_charge: int
- stop_state_of_charge: int
- send_message_callback_ std::function<void(const std::string&, const std::string&)>
- msg_for_user std::string
- periodic_comm_with_car: void
- start_periodic_comm_with_car: void
- stop_periodic_comm_with_car: void
- running_: std:atomic<bool>
- periodic_thread_: std: thread
- available_solar_energie: bool
- Is_within_time_window(const std::string& schedule):bool
- get_car_charged_with_solar(): int
- get_car_charged_on_grid(): int
+ evseController()
+~evseController()
+ start_charging_session(): void
+ stop_charging_session(): void
+ handle_authentication(): bool
+ update_user_preferences(const UserPreferences& preferences): void
+ checking_ev_state(): void
template <typename Func>
+ process_event(Func func, SetupEvent event, SetupState& current_state): bool
+ handle_idle_state(SetupEvent event, SetupState& current_state): void
+ 3 more handle functions
+ on_MQTT_message(const std::string& topic, const std::string& message): void
+set_message_callback(std::function<void(const std::string&, const std::string&)>callback) void
+ send_message_to_user(const std::string& topic, const std::string& message): void
<<enum class>>
SetupState
Idle
car_connected
initial_comm_done
inverter_chosen
authenticated
no_charging_possible
<<enum class>>
SetupEvent
checking_car_connection
start_initial_comm_with_car
chosing_inverter
start_authentication
1
1
1
<<Class>>
ChargingSessionManager
- voltage: int
- current: int
- session_id: int
- first_comm_after_init: bool
- *preference: UserPrefernece
- *used_inverter: Inverter
- *int_comm_handler: InititalCommunicationHandler
- *session_comm_handler: ChargingSessionCommunicationHandler
+ ChargingSessionManager(): void
+ ~ChargingSessionManager()
+ do_initial_comm_with_car(): bool
+ do_charing_comm_with_car(): bool
+ chose_inverter(): bool
+ get_voltage() const: int
+ get_current() const: int
+ start_inverter(): void
+ stop_inverter(): void
+ get_state_of_charge(): int
<<Class>>
AuthenticationManager
- authenticated : bool
+ AuthenticationManager(): void
+ ~AuthenticationManager()
+ authenticate(): void
+ is_authenticated(): bool
<<Class>>
UserPreferences
- solar_usage: int
- schedules: std::string
+ UserPreferences(): void
+ UserPreferences(int solar_usate, const std::String& schedule)
+ UserPreferences(const UserPreferences& prefernece)
+ ~UserPreferences()
+ get_solar_usage(): int
+ get_schedule(): std::string
+ set_solar_usage(int solar_usage) void
+ set_schedule(const std::string& schedule): void
+ is_valid_solar_usage_percentage(int percentage) const: bool
+ is_valid_schedule(const std::string& schedule) const: bool
2
1...*
<<Class>>
CommunicationHandler
- last_msg_send: bool
- session_id : int
+ CommunicationHandler(): void
+ ~CommunicationHandler()
<<Class>>
Inverter
- inverter_active: bool
- voltage: int
- current: int
- id:int
+ Inverter(): void
+ ~Inverter()
+ Inverter(Inverter &inv)
+ Inverter(Inverter&& inv)
+ Inverter& operator=(const Inverter& inv)
+ Inverter& operator=(Invereter &&inv)
+get_inverter_Stats(): std:tuple<int,int>
+start_inverter():void
+stop_inverter():void
parent
parent
child
child
<<Class>>
InitialCommunicationHandler
- requierd_voltage: int
- required_current: int
+ CommunicationHandler(): void
+ ~CommunicationHandler()
+ get_initial_information_from_car(): std::tuple<int,int>
<<Class>>
ChargingSessionCommunicationHandler
- state_of_charge: int
- battery_tem: int
+ CommunicationHandler(): void
+ ~CommunicationHandler()
+ update_stats_from_car(): void
+ get_stats: std::tuple<int,int>
<<Class>>
MQTTClient
- on_message(struct mosquitto* mosq, void* userdata, const struct mosquitto_message* message): static void
- on_connect(struct mosquitto* mosq, void* userdata, int result): static void
- loop(): void
- reconnect(): void
- *mosq_: struct mosquitto*
- id_: std::string
- host_ std::string
- port_: int
- loop_thread_: std::thread
- running: std::Atomic<bool>
- message_callback_: std::function<void(const std::string&, const std::string&)>
+ MQTTClient(const std::string& id, const std::string& host, int port)
+ ~MQTTClient()
+ connect(): bool
+ disconnect(): bool
+ publish(const std::string& topic, const std::string& message): bool
+ subscribe(const std::string & topic): bool
+ on_send_msg_from_logic(const std::string& topic, const std::string& message): void
+ set_message_callback(std::function<void(const std::string&, const std::string&)> callback): void