Building a cross-platform library

Lessons learned

Jonas Vautherin

June 29, 2022

iOS: Coming Soon

Inconsistent

Weird API

Hiding complexity

Hiding complexity

Provide helpers, do not hide low level

```
some_api.add_udp_connection(14540);
```

vs

```
UdpConnection connection(14540);
some_api.add_connection(connection);
```

Auto-generation

An auto-generated API scales and stays consistent

Can we make it ”not weird”?

MAVSDK-Python (asyncio)

```python
from mavsdk import System

drone = System()

await drone.connect()
await drone.action.arm()
await drone.action.takeoff()

async for position in drone.telemetry.position():
    print(position)
```

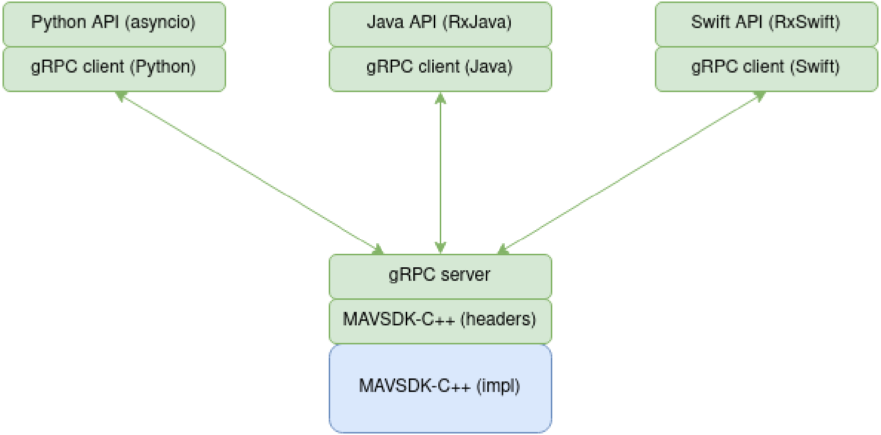
MAVSDK-C++

```cpp
auto system = get_system(mavsdk);
auto action = Action{system};
auto telemetry = Telemetry{system};

action.arm();
action.takeoff();

telemetry.subscribe_position([](Position position) {
    std::cout << position << std::endl;
});
```

MAVSDK-Java (RxJava)

```
System drone = new System();
drone.getAction().arm()
    .andThen(drone.getAction().takeoff())
    .delay(5, TimeUnit.SECONDS)
    .andThen(drone.getAction().land())
    .subscribe();
```

MAVSDK


Python API (asyncio)
gRPC client (Python)
Java API (RxJava)
gRPC client (Java)
Swift API (RxSwift)
gRPC client (Swift)
gRPC server
MAVSDK-C++ (headers)
MAVSDK-C++ (impl)

Proto definition (1)

```
syntax = "proto3";

package meetup.example.api;

// Example API for C++ meetup
service SomeApi {
  // Call a function (that may have side effects)
  rpc Poke(PokeRequest) returns(PokeResponse) {}
  ...
  // Subscribe to 'mode' updates.
  rpc SubscribeMode(SubscribeModeRequest)
                    returns(stream ModeResponse) {}
}
...
```

Proto definition (2)

```
...
message SubscribeModeRequest {}
message ModeResponse {
  Mode mode = 1; // The current mode
}

// Mode
enum Mode {
  MODE_UNKNOWN = 0; // Mode unknown
  MODE_IDLE = 1; // Mode 'idle'
  MODE_ACTIVE = 2; // Mode 'active'
}
...
```

C++ header

```cpp
class SomeApi {
public:
    enum class Mode {
        Unknown, // Mode unknown
        Idle, // Mode 'idle'
        Active, // Mode 'active'
    };

    // Call a function (that may have side effects)
    void poke();

    // Subscribe to 'mode' updates.
    void subscribe_mode(std::function<void(Mode)>& callback);
};
```

C++ template

```
class {{ plugin_name.upper_camel_case }} {
public:
{% for enum in enums %}
    enum class {{ enum.name.upper_camel_case }} {
        {% for value in enum.values %}
        {{ value.name.upper_camel_case }}
        {% endfor %}
    };
{% endfor %}

{% for method in methods %}
    void {{ method.name.lower }}({{ param.type_info.name }});
{% endfor %}
};
```

Pybind11

```
PYBIND11_MODULE(example_library_py, m) {
    py::class_<SomeApi> some_api(m, "SomeApi");
    some_api
        .def(py::init<>())
        .def("poke", &SomeApi::poke)
        .def("mode", &SomeApi::subscribe_mode);

    py::enum_<SomeApi::Mode> mode(some_api, "Mode");
    mode
        .value("Unknown", SomeApi::Mode::Unknown)
        .value("Idle", SomeApi::Mode::Idle)
        .value("Active", SomeApi::Mode::Active);
}
```

Demo

The End

References

protoc-gen-mavsdk
https://github.com/mavlink/mavsdk-proto

protoc-gen-mavsdk example
https://github.com/jonasvautherin/protoc-gen-mavsdk-example

Protobuf
https://github.com/protocolbuffers/protobuf

Protobuf compiler
https://grpc.io/docs/protoc-installation/