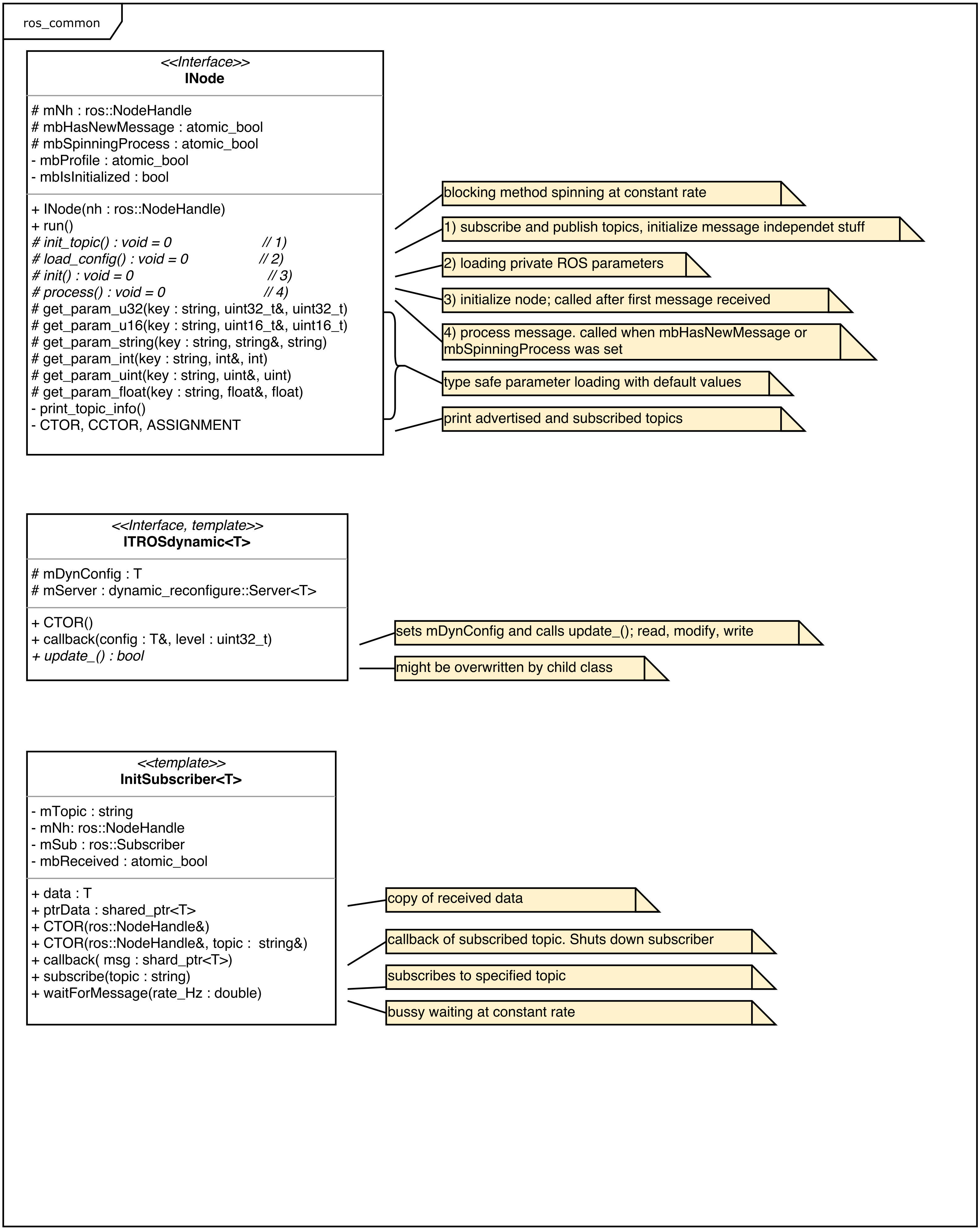

ros_common

<<Interface>>
INode

```
# mNh : ros::NodeHandle
# mbHasNewMessage : atomic_bool
# mbSpinningProcess : atomic_bool
- mbProfile : atomic_bool
- mbIsInitialized : bool
```

```
+ INode(nh : ros::NodeHandle)
+ run()
# init_topic() : void = 0              // 1)
# load_config() : void = 0             // 2)
# init() : void = 0                    // 3)
# process() : void = 0                 // 4)
# get_param_u32(key : string, uint32_t&, uint32_t)
# get_param_u16(key : string, uint16_t&, uint16_t)
# get_param_string(key : string, string&, string)
# get_param_int(key : string, int&, int)
# get_param_uint(key : string, uint&, uint)
# get_param_float(key : string, float&, float)
- print_topic_info()
- CTOR, CCTOR, ASSIGNMENT
```

blocking method spinning at constant rate

1) subscribe and publish topics, initialize message independet stuff

2) loading private ROS parameters

3) initialize node; called after first message received

4) process message. called when mbHasNewMessage or mbSpinningProcess was set

type safe parameter loading with default values

print advertised and subscribed topics

<<Interface, template>>
ITROSdynamic<T>

```
# mDynConfig : T
# mServer : dynamic_reconfigure::Server<T>
```

```
+ CTOR()
+ callback(config : T&, level : uint32_t)
+ update_() : bool
```

sets mDynConfig and calls update_(); read, modify, write

might be overwritten by child class

<<template>>
InitSubscriber<T>

```
- mTopic : string
- mNh: ros::NodeHandle
- mSub : ros::Subscriber
- mbReceived : atomic_bool
```

```
+ data : T
+ ptrData : shared_ptr<T>
+ CTOR(ros::NodeHandle&)
+ CTOR(ros::NodeHandle&, topic :  string&)
+ callback( msg : shard_ptr<T>)
+ subscribe(topic : string)
+ waitForMessage(rate_Hz : double)
```

copy of received data

callback of subscribed topic. Shuts down subscriber

subscribes to specified topic

bussy waiting at constant rate