## # stateMachine # components + ISmaccClient() + ~ISmaccClient() + initialize() + setStateMachine() + getName() + postEvent() + postEvent() + getComponent() + configureEventSourceTypes() + getType() smacc::client\_bases ::SmaccSubscriberClient < MessageType > + topicName + queueSize + onFirstMessageReceived\_ + onMessageReceived\_ + postMessageEvent + postInitialMessageEvent # nh\_ - sub firstMessage\_ - initialized\_ + SmaccSubscriberClient() + ~SmaccSubscriberClient() + onMessageReceived() + onFirstMessageReceived() + configureEventSourceTypes() + initialize() messageCallback() < sensor msgs::LaserScan > smacc::client\_bases ::SmaccSubscriberClient < sensor\_msgs::LaserScan > multirole sensor client + topicName ::CIMultiroleSensor< Message + queueSize Type > + onFirstMessageReceived\_ + onMessageReceived\_ + onMessageTimeout + postTimeoutMessageEvent + postMessageEvent + postInitialMessageEvent + timeout # nh timeoutTimer\_ initialized\_ - sub - firstMessage\_ + CIMultiroleSensor() initialized + onMessageTimeout() + SmaccSubscriberClient() + configureEventSourceTypes() + ~SmaccSubscriberClient() + initialize() + onMessageReceived() # resetTimer() + onFirstMessageReceived() timeoutCallback() + configureEventSourceTypes() + initialize() messageCallback() < sensor\_msgs::LaserScan > multirole\_sensor\_client ::CIMultiroleSensor< sensor \_msgs::LaserScan > + onMessageTimeout\_ + postTimeoutMessageEvent + timeout - timeoutTimer\_ initialized + CIMultiroleSensor() + onMessageTimeout() + configureEventSourceTypes() + initialize() # resetTimer() timeoutCallback() sm\_dance\_bot::cl\_lidar ::CILaserSensor

+ CILaserSensor()

smacc::ISmaccClient