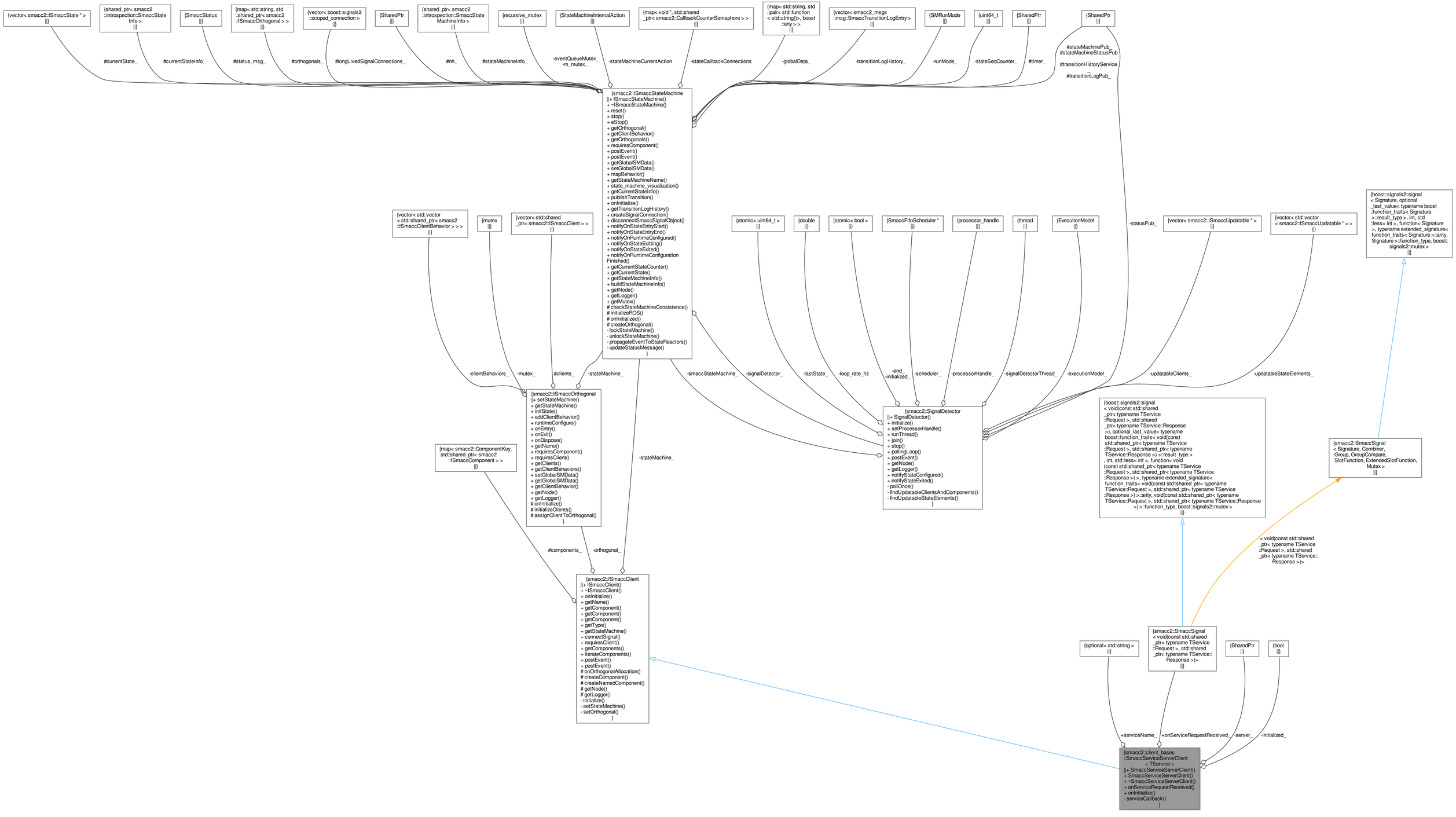
{vector< smacc2::ISmaccState * > }
{shared_ptr< smacc2 ::introspection::SmaccState Info > }
{SmaccStatus }
{map< std::string, std ::shared_ptr< smacc2 ::ISmaccOrthogonal > > }
{vector< boost::signals2 ::scoped_connection > }
{SharedPtr }
{shared_ptr< smacc2 ::introspection::SmaccState MachineInfo > }
{recursive_mutex }
{StateMachineInternalAction }
{map< void *, std::shared _ptr< smacc2::CallbackCounterSemaphore > > }
{map< std::string, std ::pair< std::function < std::string()>, boost ::any > > }
{vector< smacc2_msgs ::msg::SmaccTransitionLogEntry > }
{SMRunMode }
{uint64_t }
{SharedPtr }
{SharedPtr }
#currentState_
#currentStateInfo_
#status_msg_
#orthogonals_
#longLivedSignalConnections_
#nh_
#stateMachineInfo_
-eventQueueMutex_ -m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#timer_
#stateMachinePub_ #stateMachineStatusPub
#transitionHistoryService
#transitionLogPub_
{smacc2::ISmaccStateMachine }
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getClientBehavior()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ disconnectSmaccSignalObject()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExitting()
+ notifyOnStateExited()
+ notifyOnRuntimeConfiguration Finished()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
+ getLogger()
+ getMutex()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
}
{vector< std::vector < std::shared_ptr< smacc2 ::ISmaccClientBehavior > > > }
{mutex }
{vector< std::shared _ptr< smacc2::ISmaccClient > > }
{atomic< uint64_t > }
{double }
{atomic< bool > }
{SmaccFifoScheduler * }
{processor_handle }
{thread }
{ExecutionModel }
-statusPub_
{vector< smacc2::ISmaccUpdatable * > }
{vector< std::vector < smacc2::ISmaccUpdatable * > > }
{boost::signals2::signal < Signature, optional _last_value< typename boost ::function_traits< Signature >::result_type >, int, std ::less< int >, function< Signature >, typename extended_signature< function_traits< Signature >::arity, Signature >::function_type, boost:: signals2::mutex > }
-clientBehaviors_
-mutex_
#clients_
-stateMachine_
-smaccStateMachine_
-signalDetector_
-lastState_
-loop_rate_hz
-end_ -initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
{smacc2::ISmaccOrthogonal }
+ setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
+ getNode()
+ getLogger()
# onInitialize()
# initializeClients()
# assignClientToOrthogonal()
}
{smacc2::SignalDetector }
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ getNode()
+ getLogger()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClientsAndComponents()
- findUpdatableStateElements()
}
{boost::signals2::signal < void(const std::shared _ptr< typename TService ::Request >, std::shared _ptr< typename TService::Response >), optional_last_value< typename boost ::function_traits< void(const std::shared_ptr< typename TService ::Request >, std::shared_ptr< typename TService::Response >) >::result_type > , int, std::less< int >, function< void (const std::shared_ptr< typename TService ::Request >, std::shared_ptr< typename TService ::Response >) >, typename extended_signature< function_traits< void(const std::shared_ptr< typename TService::Request >, std::shared_ptr< typename TService ::Response >) >::arity, void(const std::shared_ptr< typename TService::Request >, std::shared_ptr< typename TService::Response >) >::function_type, boost::signals2::mutex > }
{smacc2::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex > }
{map< smacc2::ComponentKey, std::shared_ptr< smacc2 ::ISmaccComponent > > }
-stateMachine_
< void(const std::shared _ptr< typename TService ::Request >, std::shared _ptr< typename TService:: Response >)>
#components_
-orthogonal_
{smacc2::ISmaccClient }
+ ISmaccClient()
+ ~ISmaccClient()
+ onInitialize()
+ getName()
+ getComponent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
+ iterateComponents()
+ postEvent()
+ postEvent()
# onOrthogonalAllocation()
# createComponent()
# createNamedComponent()
# getNode()
# getLogger()
- initialize()
- setStateMachine()
- setOrthogonal()
}
{smacc2::SmaccSignal < void(const std::shared _ptr< typename TService ::Request >, std::shared _ptr< typename TService:: Response >)> }
{optional< std::string > }
{SharedPtr }
{bool }
+serviceName_
+onServiceRequestReceived_
-server_
-initialized_
{smacc2::client_bases ::SmaccServiceServerClient < TService > }
+ SmaccServiceServerClient()
+ SmaccServiceServerClient()
+ ~SmaccServiceServerClient()
+ onServiceRequestReceived()
+ onInitialize()
- serviceCallback()
}