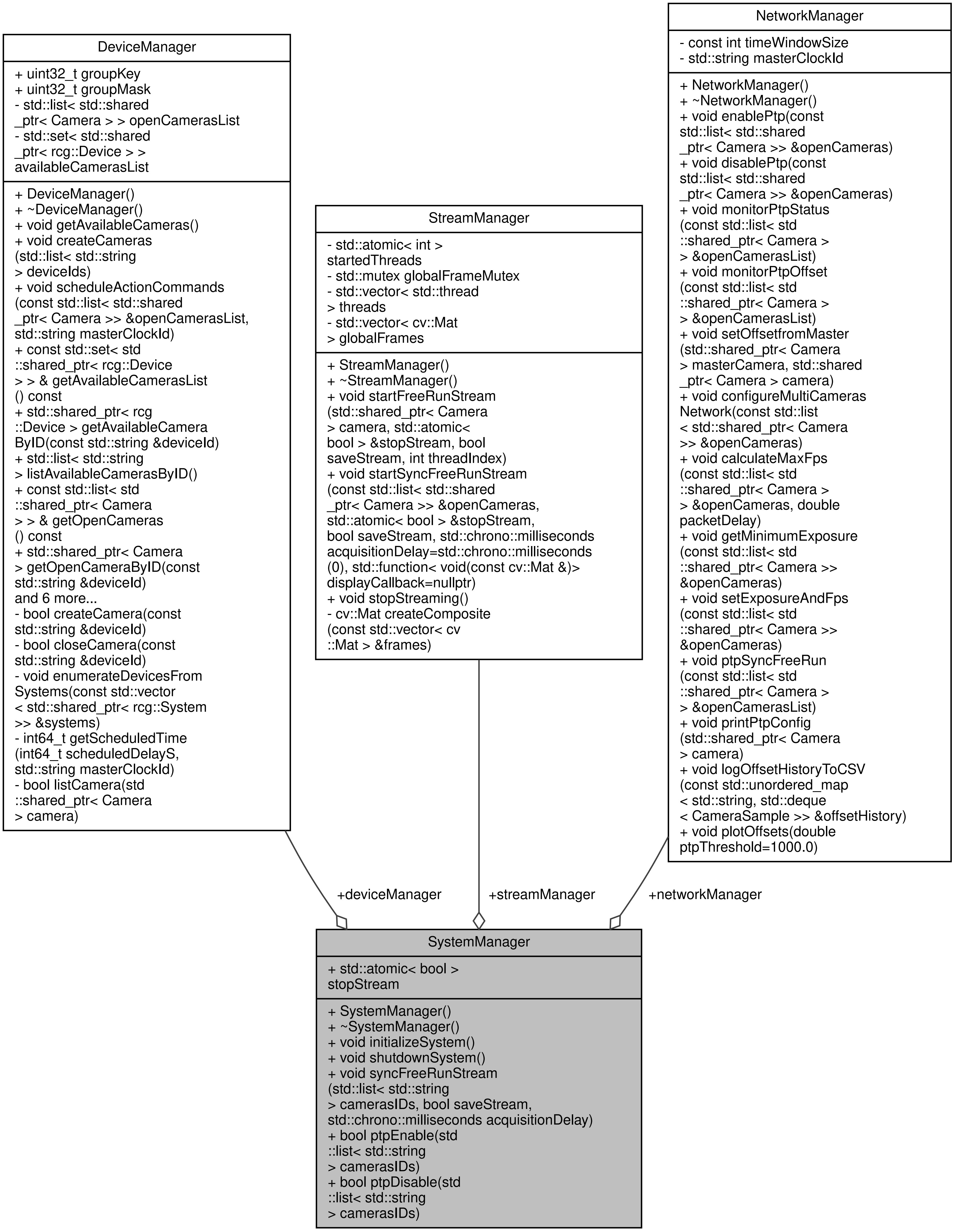
DeviceManager
+ uint32_t groupKey
+ uint32_t groupMask
- std::list< std::shared _ptr< Camera > > openCamerasList
- std::set< std::shared _ptr< rcg::Device > > availableCamerasList
+ DeviceManager()
+ ~DeviceManager()
+ void getAvailableCameras()
+ void createCameras (std::list< std::string > deviceIds)
+ void scheduleActionCommands (const std::list< std::shared _ptr< Camera >> &openCamerasList, std::string masterClockId)
+ const std::set< std ::shared_ptr< rcg::Device > > & getAvailableCamerasList () const
+ std::shared_ptr< rcg ::Device > getAvailableCamera ByID(const std::string &deviceId)
+ std::list< std::string > listAvailableCamerasByID()
+ const std::list< std ::shared_ptr< Camera > > & getOpenCameras () const
+ std::shared_ptr< Camera > getOpenCameraByID(const std::string &deviceId)
and 6 more...
- bool createCamera(const std::string &deviceId)
- bool closeCamera(const std::string &deviceId)
- void enumerateDevicesFrom Systems(const std::vector < std::shared_ptr< rcg::System >> &systems)
- int64_t getScheduledTime (int64_t scheduledDelayS, std::string masterClockId)
- bool listCamera(std ::shared_ptr< Camera > camera)
StreamManager
- std::atomic< int > startedThreads
- std::mutex globalFrameMutex
- std::vector< std::thread > threads
- std::vector< cv::Mat > globalFrames
+ StreamManager()
+ ~StreamManager()
+ void startFreeRunStream (std::shared_ptr< Camera > camera, std::atomic< bool > &stopStream, bool saveStream, int threadIndex)
+ void startSyncFreeRunStream (const std::list< std::shared _ptr< Camera >> &openCameras, std::atomic< bool > &stopStream, bool saveStream, std::chrono::milliseconds acquisitionDelay=std::chrono::milliseconds (0), std::function< void(const cv::Mat &)> displayCallback=nullptr)
+ void stopStreaming()
- cv::Mat createComposite (const std::vector< cv ::Mat > &frames)
NetworkManager
- const int timeWindowSize
- std::string masterClockId
+ NetworkManager()
+ ~NetworkManager()
+ void enablePtp(const std::list< std::shared _ptr< Camera >> &openCameras)
+ void disablePtp(const std::list< std::shared _ptr< Camera >> &openCameras)
+ void monitorPtpStatus (const std::list< std ::shared_ptr< Camera > > &openCamerasList)
+ void monitorPtpOffset (const std::list< std ::shared_ptr< Camera > > &openCamerasList)
+ void setOffsetfromMaster (std::shared_ptr< Camera > masterCamera, std::shared _ptr< Camera > camera)
+ void configureMultiCameras Network(const std::list < std::shared_ptr< Camera >> &openCameras)
+ void calculateMaxFps (const std::list< std ::shared_ptr< Camera > > &openCameras, double packetDelay)
+ void getMinimumExposure (const std::list< std ::shared_ptr< Camera >> &openCameras)
+ void setExposureAndFps (const std::list< std ::shared_ptr< Camera >> &openCameras)
+ void ptpSyncFreeRun (const std::list< std ::shared_ptr< Camera > > &openCamerasList)
+ void printPtpConfig (std::shared_ptr< Camera > camera)
+ void logOffsetHistoryToCSV (const std::unordered_map < std::string, std::deque < CameraSample >> &offsetHistory)
+ void plotOffsets(double ptpThreshold=1000.0)
+deviceManager
+streamManager
+networkManager
SystemManager
+ std::atomic< bool > stopStream
+ SystemManager()
+ ~SystemManager()
+ void initializeSystem()
+ void shutdownSystem()
+ void syncFreeRunStream (std::list< std::string > camerasIDs, bool saveStream, std::chrono::milliseconds acquisitionDelay)
+ bool ptpEnable(std ::list< std::string > camerasIDs)
+ bool ptpDisable(std ::list< std::string > camerasIDs)