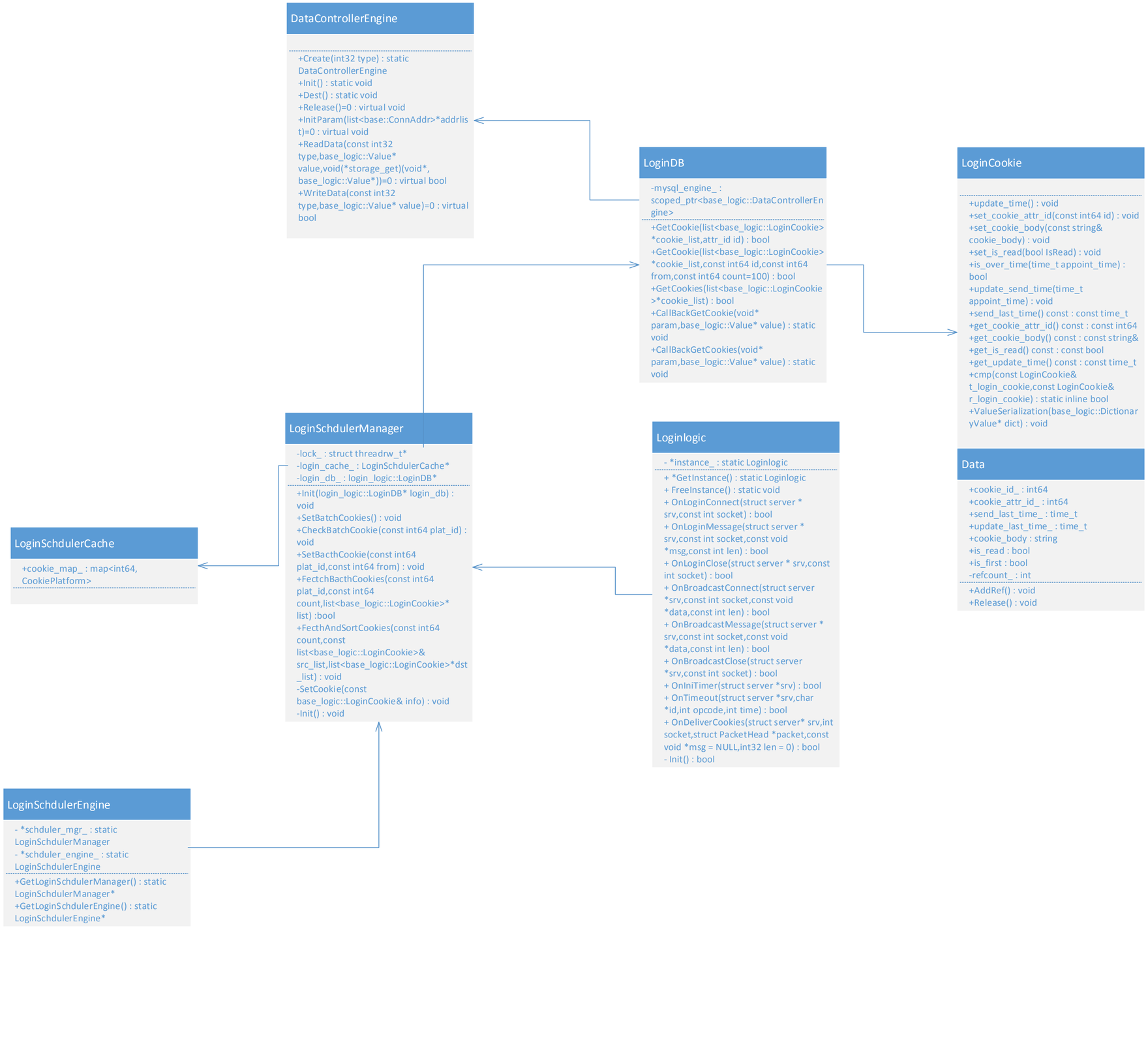
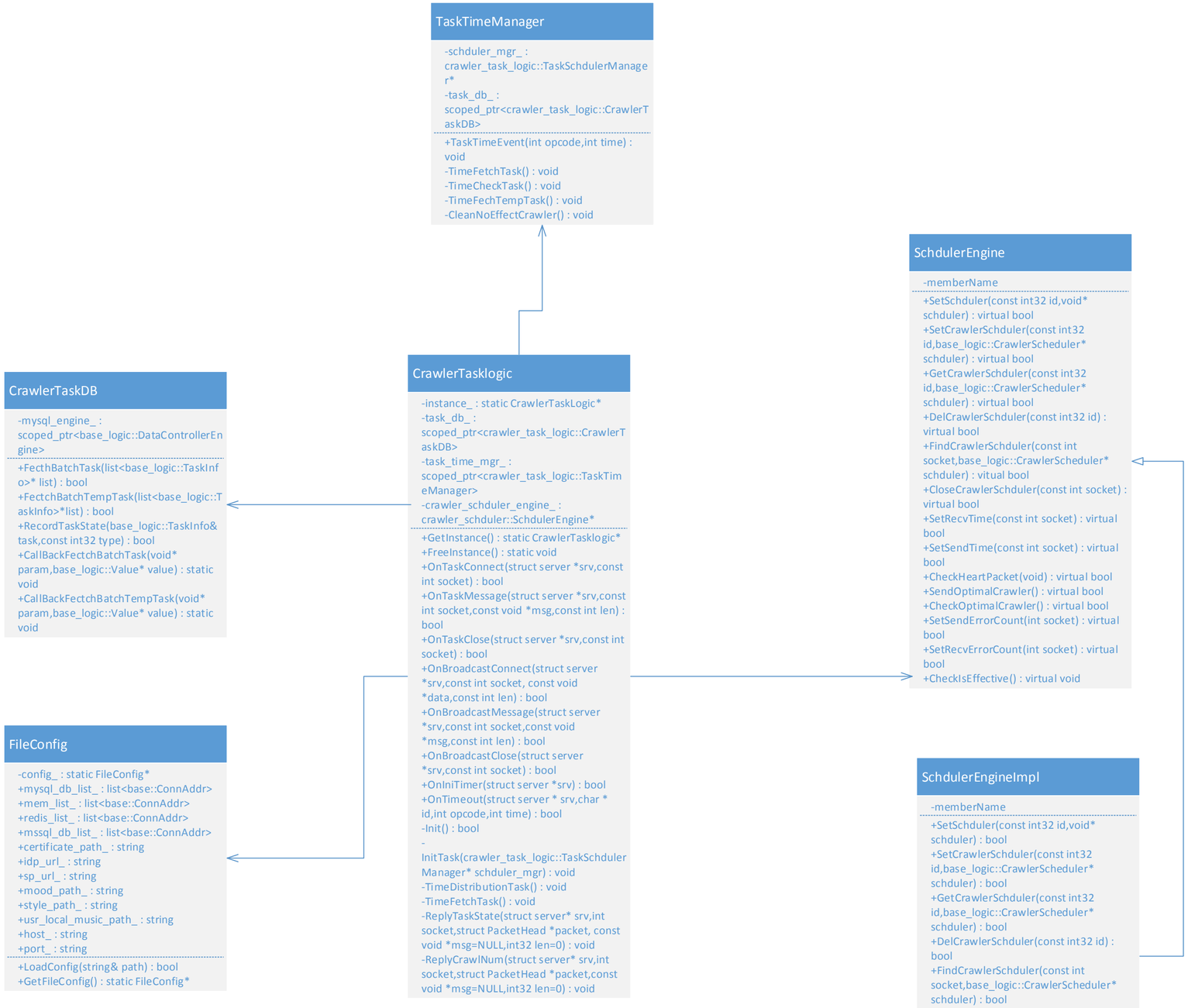
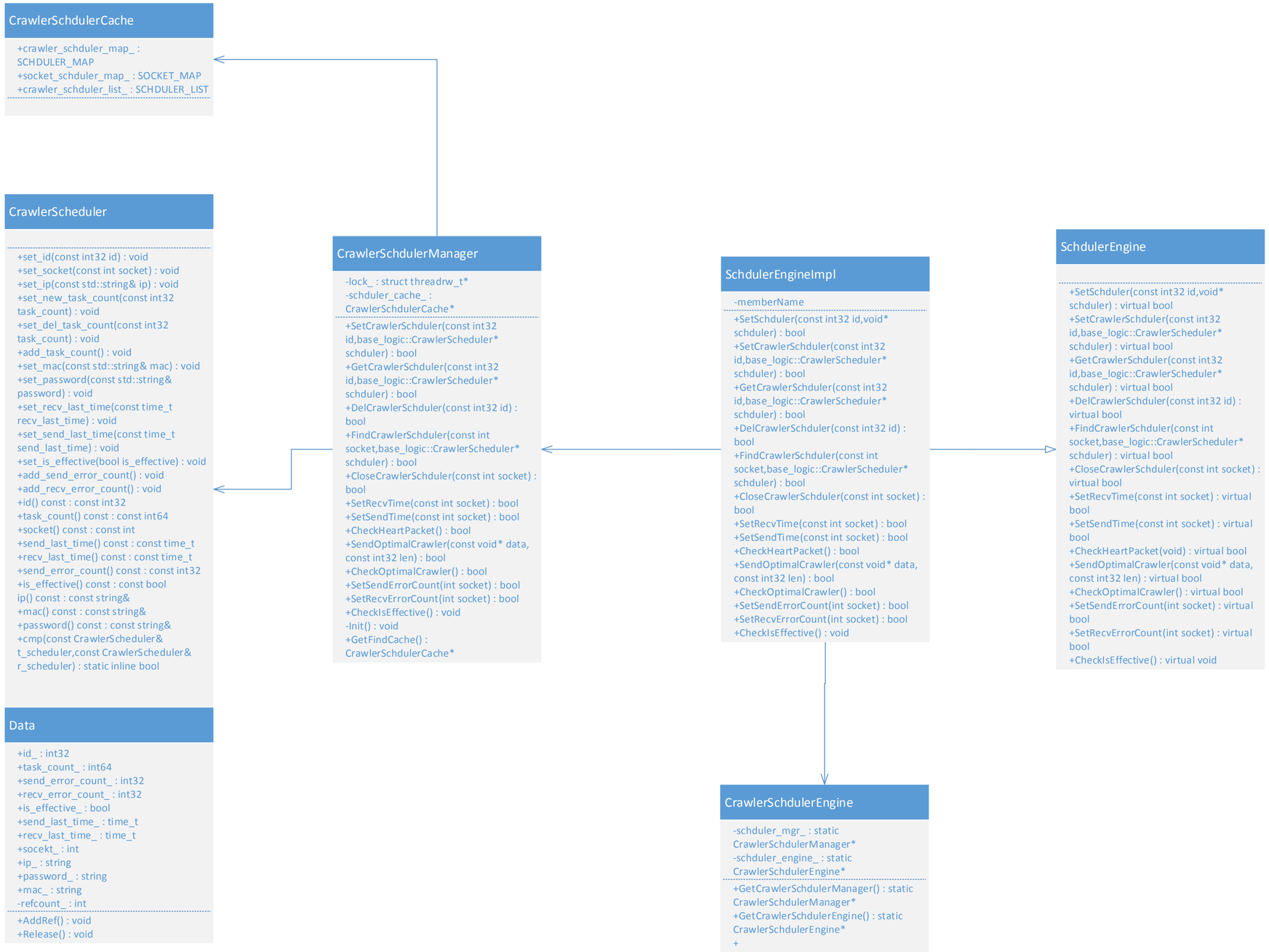
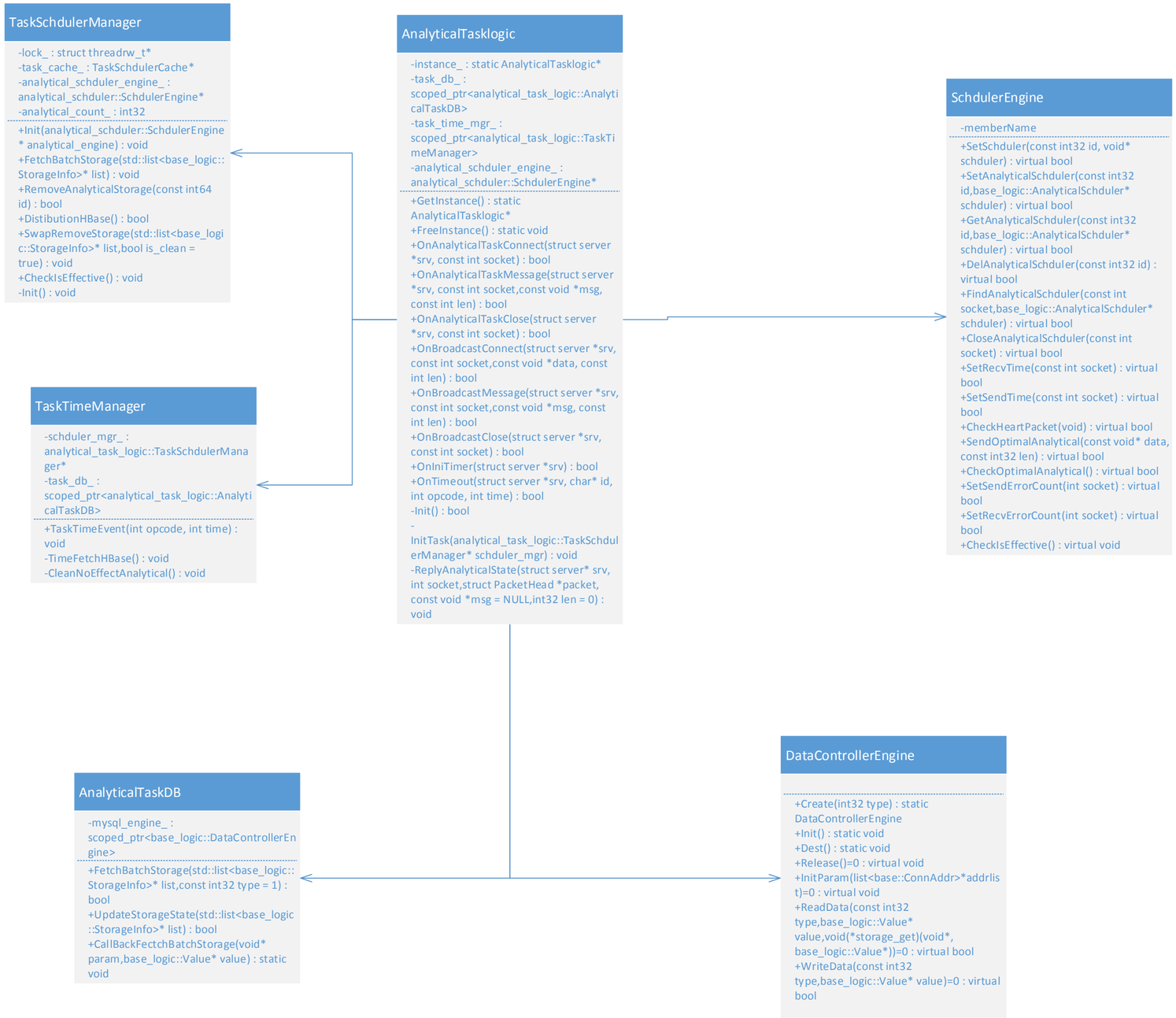
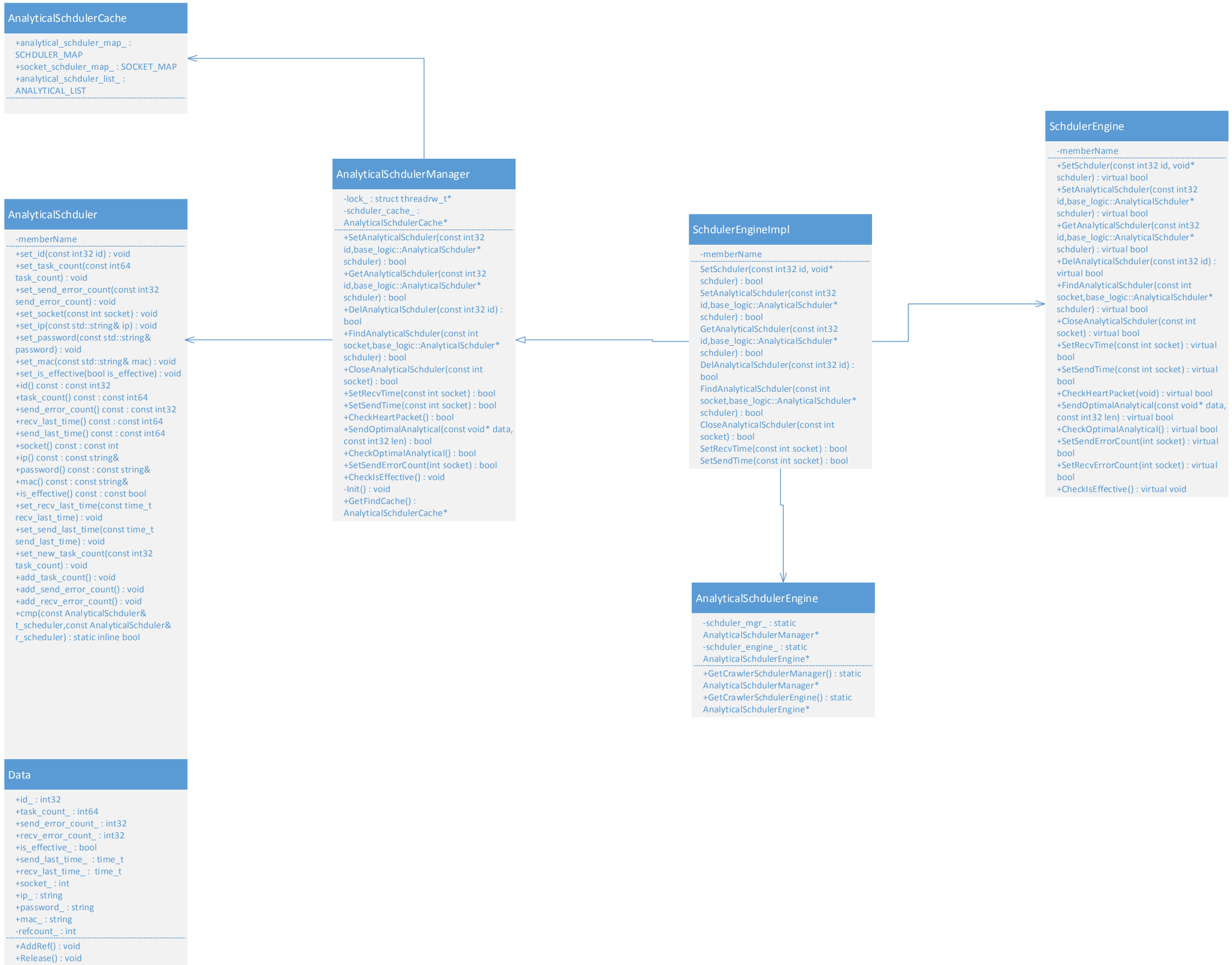
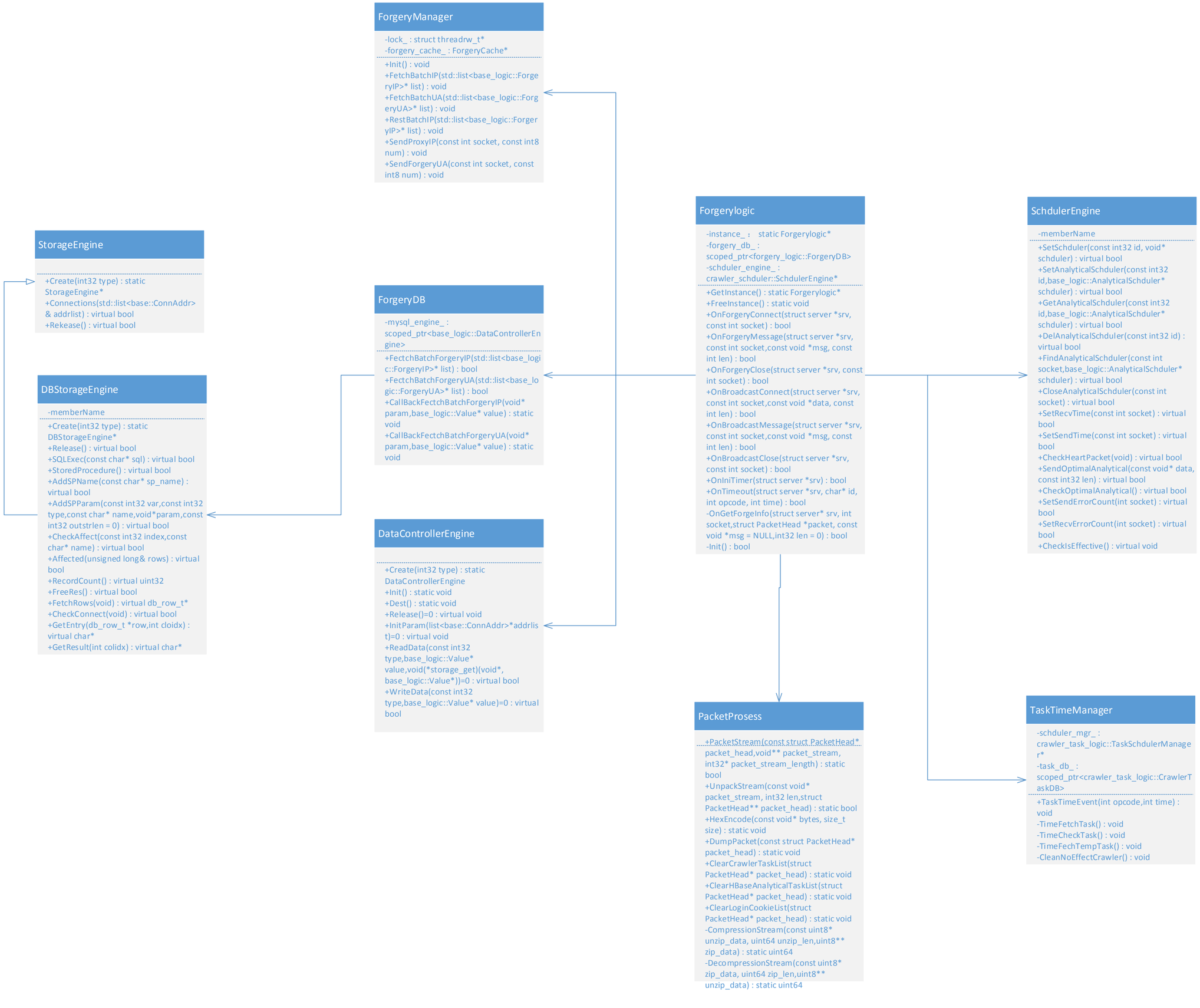

DataControllerEngine
+Create(int32 type) : static DataControllerEngine
+Init() : static void
+Dest() : static void
+Release()=0 : virtual void
+InitParam(list<base::ConnAddr>*addrlist)=0 : virtual void
+ReadData(const int32 type,base_logic::Value* value,void(*storage_get)(void*, base_logic::Value*))=0 : virtual bool
+WriteData(const int32 type,base_logic::Value* value)=0 : virtual bool
LoginDB
-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+GetCookie(list<base_logic::LoginCookie>*cookie_list,attr_id id) : bool
+GetCookie(list<base_logic::LoginCookie>*cookie_list,const int64 id,const int64 from,const int64 count=100) : bool
+GetCookies(list<base_logic::LoginCookie>*cookie_list) : bool
+CallBackGetCookie(void* param,base_logic::Value* value) : static void
+CallBackGetCookies(void* param,base_logic::Value* value) : static void
LoginCookie
+update_time() : void
+set_cookie_attr_id(const int64 id) : void
+set_cookie_body(const string& cookie_body) : void
+set_is_read(bool IsRead) : void
+is_over_time(time_t appoint_time) : bool
+update_send_time(time_t appoint_time) : void
+send_last_time() const : const time_t
+get_cookie_attr_id() const : const int64
+get_cookie_body() const : const string&
+get_is_read() const : const bool
+get_update_time() const : const time_t
+cmp(const LoginCookie& t_login_cookie,const LoginCookie& r_login_cookie) : static inline bool
+ValueSerialization(base_logic::DictionaryValue* dict) : void
Data
+cookie_id_ : int64
+cookie_attr_id_ : int64
+send_last_time_ : time_t
+update_last_time_ : time_t
+cookie_body : string
+is_read : bool
+is_first : bool
-refcount_ : int
+AddRef() : void
+Release() : void
LoginSchdulerManager
-lock_ : struct threadrw_t*
-login_cache_ : LoginSchdulerCache*
-login_db_ : login_logic::LoginDB*
+Init(login_logic::LoginDB* login_db) : void
+SetBatchCookies() : void
+CheckBatchCookie(const int64 plat_id) : void
+SetBacthCookie(const int64 plat_id,const int64 from) : void
+FectchBacthCookies(const int64 plat_id,const int64 count,list<base_logic::LoginCookie>* list) :bool
+FecthAndSortCookies(const int64 count,const list<base_logic::LoginCookie>& src_list,list<base_logic::LoginCookie>*dst_list) : void
-SetCookie(const base_logic::LoginCookie& info) : void
-Init() : void
Loginlogic
- *instance_ : static Loginlogic
+ *GetInstance() : static Loginlogic
+ FreeInstance() : static void
+ OnLoginConnect(struct server * srv,const int socket) : bool
+ OnLoginMessage(struct server * srv,const int socket,const void *msg,const int len) : bool
+ OnLoginClose(struct server * srv,const int socket) : bool
+ OnBroadcastConnect(struct server *srv,const int socket,const void *data,const int len) : bool
+ OnBroadcastMessage(struct server * srv,const int socket,const void *data,const int len) : bool
+ OnBroadcastClose(struct server *srv,const int socket) : bool
+ OnIniTimer(struct server *srv) : bool
+ OnTimeout(struct server *srv,char *id,int opcode,int time) : bool
+ OnDeliverCookies(struct server* srv,int socket,struct PacketHead *packet,const void *msg = NULL,int32 len = 0) : bool
- Init() : bool
LoginSchdulerCache
+cookie_map_ : map<int64, CookiePlatform>
LoginSchdulerEngine
- *schduler_mgr_ : static LoginSchdulerManager
- *schduler_engine_ : static LoginSchdulerEngine
+GetLoginSchdulerManager() : static LoginSchdulerManager*
+GetLoginSchdulerEngine() : static LoginSchdulerEngine*

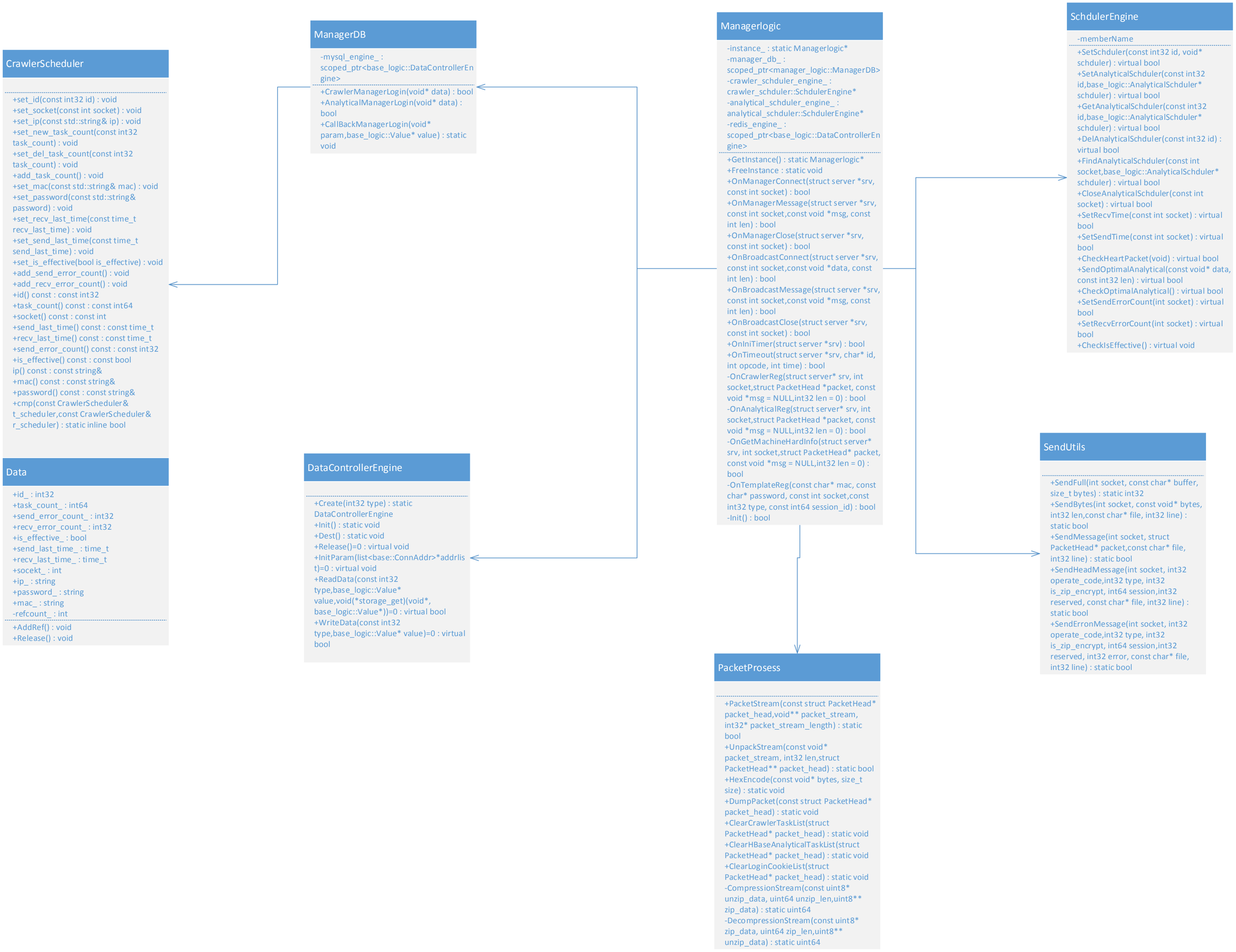

TaskTimeManager
-schduler_mgr_ : crawler_task_logic::TaskSchdulerManager*
-task_db_ : scoped_ptr<crawler_task_logic::CrawlerTaskDB>
+TaskTimeEvent(int opcode,int time) : void
-TimeFetchTask() : void
-TimeCheckTask() : void
-TimeFechTempTask() : void
-CleanNoEffectCrawler() : void
SchdulerEngine
-memberName
+SetSchduler(const int32 id,void* schduler) : virtual bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+DelCrawlerSchduler(const int32 id) : virtual bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : vitual bool
+CloseCrawlerSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalCrawler() : virtual bool
+CheckOptimalCrawler() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
CrawlerTaskDB
-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+FecthBatchTask(list<base_logic::TaskInfo>* list) : bool
+FectchBatchTempTask(list<base_logic::TaskInfo>*list) : bool
+RecordTaskState(base_logic::TaskInfo& task,const int32 type) : bool
+CallBackFectchBatchTask(void* param,base_logic::Value* value) : static void
+CallBackFectchBatchTempTask(void* param,base_logic::Value* value) : static void
CrawlerTasklogic
-instance_ : static CrawlerTaskLogic*
-task_db_ : scoped_ptr<crawler_task_logic::CrawlerTaskDB>
-task_time_mgr_ : scoped_ptr<crawler_task_logic::TaskTimeManager>
-crawler_schduler_engine_ : crawler_schduler::SchdulerEngine*
+GetInstance() : static CrawlerTasklogic*
+FreeInstance() : static void
+OnTaskConnect(struct server *srv,const int socket) : bool
+OnTaskMessage(struct server *srv,const int socket,const void *msg,const int len) : bool
+OnTaskClose(struct server *srv,const int socket) : bool
+OnBroadcastConnect(struct server *srv,const int socket, const void *data,const int len) : bool
+OnBroadcastMessage(struct server *srv,const int socket,const void *msg,const int len) : bool
+OnBroadcastClose(struct server *srv,const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server * srv,char * id,int opcode,int time) : bool
-Init() : bool
-InitTask(crawler_task_logic::TaskSchdulerManager* schduler_mgr) : void
-TimeDistributionTask() : void
-TimeFetchTask() : void
-ReplyTaskState(struct server* srv,int socket,struct PacketHead *packet, const void *msg=NULL,int32 len=0) : void
-ReplyCrawlNum(struct server* srv,int socket,struct PacketHead *packet,const void *msg=NULL,int32 len=0) : void
FileConfig
-config_ : static FileConfig*
+mysql_db_list_ : list<base::ConnAddr>
+mem_list_ : list<base::ConnAddr>
+redis_list_ : list<base::ConnAddr>
+mssql_db_list_ : list<base::ConnAddr>
+certificate_path_ : string
+idp_url_ : string
+sp_url_ : string
+mood_path_ : string
+style_path_ : string
+usr_local_music_path_ : string
+host_ : string
+port_ : string
+LoadConfig(string& path) : bool
+GetFileConfig() : static FileConfig*
SchdulerEngineImpl
-memberName
+SetSchduler(const int32 id,void* schduler) : bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+DelCrawlerSchduler(const int32 id) : bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : bool
+CloseCrawlerSchduler(const int socket)
+SetRecvTime(const int socket) : bool
+SetSendTime(const int socket) : bool
+CheckHeartPacket() : bool
+SendOptimalCrawler(const void* data,const int32 len) : bool
+CheckOptimalCrawler() : bool
+SetSendErrorCount(int socket) : bool
+SetRecvErrorCount(int socket) : bool
+CheckIsEffective() : void

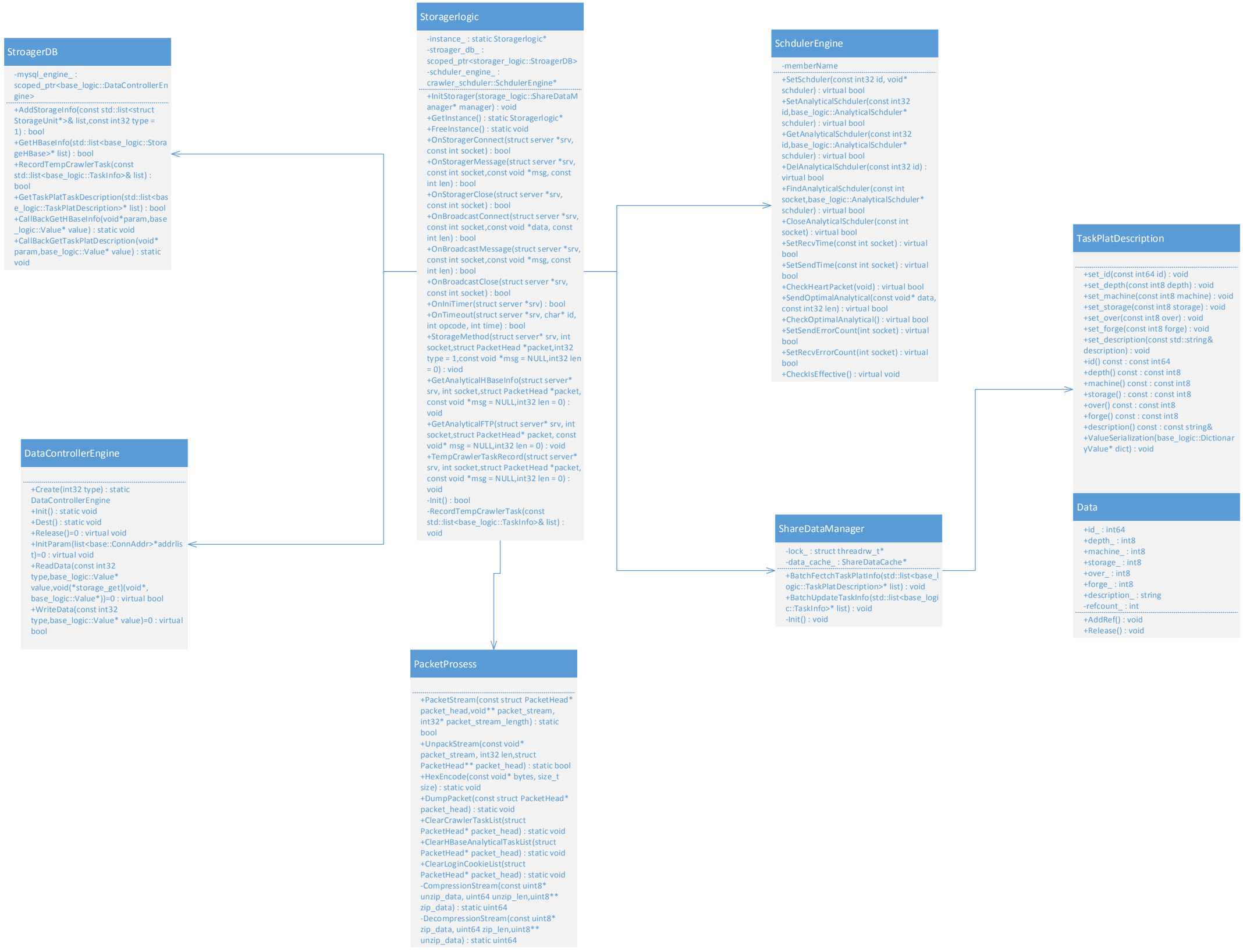

CrawlerSchdulerCache
+crawler_schduler_map_ : SCHDULER_MAP
+socket_schduler_map_ : SOCKET_MAP
+crawler_schduler_list_ : SCHDULER_LIST
CrawlerScheduler
+set_id(const int32 id) : void
+set_socket(const int socket) : void
+set_ip(const std::string& ip) : void
+set_new_task_count(const int32 task_count) : void
+set_del_task_count(const int32 task_count) : void
+add_task_count() : void
+set_mac(const std::string& mac) : void
+set_password(const std::string& password) : void
+set_recv_last_time(const time_t recv_last_time) : void
+set_send_last_time(const time_t send_last_time) : void
+set_is_effective(bool is_effective) : void
+add_send_error_count() : void
+add_recv_error_count() : void
+id() const : const int32
+task_count() const : const int64
+socket() const : const int
+send_last_time() const : const time_t
+recv_last_time() const : const time_t
+send_error_count() const : const int32
+is_effective() const : const bool
ip() const : const string&
+mac() const : const string&
+password() const : const string&
+cmp(const CrawlerScheduler& t_scheduler,const CrawlerScheduler& r_scheduler) : static inline bool
Data
+id_ : int32
+task_count_ : int64
+send_error_count_ : int32
+recv_error_count_ : int32
+is_effective_ : bool
+send_last_time_ : time_t
+recv_last_time_ : time_t
+socekt_ : int
+ip_ : string
+password_ : string
+mac_ : string
-refcount_ : int
+AddRef() : void
+Release() : void
CrawlerSchdulerManager
-lock_ : struct threadrw_t*
-schduler_cache_ : CrawlerSchdulerCache*
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+DelCrawlerSchduler(const int32 id) : bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : bool
+CloseCrawlerSchduler(const int socket) : bool
+SetRecvTime(const int socket) : bool
+SetSendTime(const int socket) : bool
+CheckHeartPacket() : bool
+SendOptimalCrawler(const void* data, const int32 len) : bool
+CheckOptimalCrawler() : bool
+SetSendErrorCount(int socket) : bool
+SetRecvErrorCount(int socket) : bool
+CheckIsEffective() : void
-Init() : void
+GetFindCache() : CrawlerSchdulerCache*
SchdulerEngineImpl
-memberName
+SetSchduler(const int32 id,void* schduler) : bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+DelCrawlerSchduler(const int32 id) : bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : bool
+CloseCrawlerSchduler(const int socket) : bool
+SetRecvTime(const int socket) : bool
+SetSendTime(const int socket) : bool
+CheckHeartPacket() : bool
+SendOptimalCrawler(const void* data, const int32 len) : bool
+CheckOptimalCrawler() : bool
+SetSendErrorCount(int socket) : bool
+SetRecvErrorCount(int socket) : bool
+CheckIsEffective() : void
CrawlerSchdulerEngine
-schduler_mgr_ : static CrawlerSchdulerManager*
-schduler_engine_ : static CrawlerSchdulerEngine*
+GetCrawlerSchdulerManager() : static CrawlerSchdulerManager*
+GetCrawlerSchdulerEngine() : static CrawlerSchdulerEngine*
+
SchdulerEngine
+SetSchduler(const int32 id,void* schduler) : virtual bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+DelCrawlerSchduler(const int32 id) : virtual bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : virtual bool
+CloseCrawlerSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalCrawler(const void* data, const int32 len) : virtual bool
+CheckOptimalCrawler() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void


TaskSchdulerManager
-lock_ : struct threadrw_t*
-task_cache_ : TaskSchdulerCache*
-analytical_schduler_engine_ : analytical_schduler::SchdulerEngine*
-analytical_count_ : int32
+Init(analytical_schduler::SchdulerEngine * analytical_engine) : void
+FetchBatchStorage(std::list<base_logic::StorageInfo>* list) : void
+RemoveAnalyticalStorage(const int64 id) : bool
+DistibutionHBase() : bool
+SwapRemoveStorage(std::list<base_logic::StorageInfo>* list,bool is_clean = true) : void
+CheckIsEffective() : void
-Init() : void
AnalyticalTasklogic
-instance_ : static AnalyticalTasklogic*
-task_db_ : scoped_ptr<analytical_task_logic::AnalyticalTaskDB>
-task_time_mgr_ : scoped_ptr<analytical_task_logic::TaskTimeManager>
-analytical_schduler_engine_ : analytical_schduler::SchdulerEngine*
+GetInstance() : static AnalyticalTasklogic*
+FreeInstance() : static void
+OnAnalyticalTaskConnect(struct server *srv, const int socket) : bool
+OnAnalyticalTaskMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnAnalyticalTaskClose(struct server *srv, const int socket) : bool
+OnBroadcastConnect(struct server *srv, const int socket,const void *data, const int len) : bool
+OnBroadcastMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnBroadcastClose(struct server *srv, const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server *srv, char* id, int opcode, int time) : bool
-Init() : bool
-InitTask(analytical_task_logic::TaskSchdulerManager* schduler_mgr) : void
-ReplyAnalyticalState(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : void
SchdulerEngine
-memberName
+SetSchduler(const int32 id, void* schduler) : virtual bool
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+DelAnalyticalSchduler(const int32 id) : virtual bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : virtual bool
+CloseAnalyticalSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalAnalytical(const void* data, const int32 len) : virtual bool
+CheckOptimalAnalytical() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
TaskTimeManager
-schduler_mgr_ : analytical_task_logic::TaskSchdulerManager*
-task_db_ : scoped_ptr<analytical_task_logic::AnalyticalTaskDB>
+TaskTimeEvent(int opcode, int time) : void
-TimeFetchHBase() : void
-CleanNoEffectAnalytical() : void
AnalyticalTaskDB
-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+FetchBatchStorage(std::list<base_logic::StorageInfo>* list,const int32 type = 1) : bool
+UpdateStorageState(std::list<base_logic::StorageInfo>* list) : bool
+CallBackFectchBatchStorage(void* param,base_logic::Value* value) : static void
DataControllerEngine
+Create(int32 type) : static DataControllerEngine
+Init() : static void
+Dest() : static void
+Release()=0 : virtual void
+InitParam(list<base::ConnAddr>*addrlist)=0 : virtual void
+ReadData(const int32 type,base_logic::Value* value,void(*storage_get)(void*, base_logic::Value*))=0 : virtual bool
+WriteData(const int32 type,base_logic::Value* value)=0 : virtual bool


AnalyticalSchdulerCache
+analytical_schduler_map_ : SCHDULER_MAP
+socket_schduler_map_ : SOCKET_MAP
+analytical_schduler_list_ : ANALYTICAL_LIST
AnalyticalSchdulerManager
-lock_ : struct threadrw_t*
-schduler_cache_ : AnalyticalSchdulerCache*
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : bool
+DelAnalyticalSchduler(const int32 id) : bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : bool
+CloseAnalyticalSchduler(const int socket) : bool
+SetRecvTime(const int socket) : bool
+SetSendTime(const int socket) : bool
+CheckHeartPacket() : bool
+SendOptimalAnalytical(const void* data, const int32 len) : bool
+CheckOptimalAnalytical() : bool
+SetSendErrorCount(int socket) : bool
+CheckIsEffective() : void
-Init() : void
+GetFindCache() : AnalyticalSchdulerCache*
AnalyticalSchduler
-memberName
+set_id(const int32 id) : void
+set_task_count(const int64 task_count) : void
+set_send_error_count(const int32 send_error_count) : void
+set_socket(const int socket) : void
+set_ip(const std::string& ip) : void
+set_password(const std::string& password) : void
+set_mac(const std::string& mac) : void
+set_is_effective(bool is_effective) : void
+id() const : const int32
+task_count() const : const int64
+send_error_count() const : const int32
+recv_last_time() const : const int64
+send_last_time() const : const int64
+socket() const : const int
+ip() const : const string&
+password() const : const string&
+mac() const : const string&
+is_effective() const : const bool
+set_recv_last_time(const time_t recv_last_time) : void
+set_send_last_time(const time_t send_last_time) : void
+set_new_task_count(const int32 task_count) : void
+add_task_count() : void
+add_send_error_count() : void
+add_recv_error_count() : void
+cmp(const AnalyticalSchduler& t_scheduler,const AnalyticalSchduler& r_scheduler) : static inline bool
Data
+id_ : int32
+task_count_ : int64
+send_error_count_ : int32
+recv_error_count_ : int32
+is_effective_ : bool
+send_last_time_ : time_t
+recv_last_time_ : time_t
+socket_ : int
+ip_ : string
+password_ : string
+mac_ : string
-refcount_ : int
+AddRef() : void
+Release() : void
SchdulerEngineImpl
-memberName
SetSchduler(const int32 id, void* schduler) : bool
SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : bool
GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : bool
DelAnalyticalSchduler(const int32 id) : bool
FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : bool
CloseAnalyticalSchduler(const int socket) : bool
SetRecvTime(const int socket) : bool
SetSendTime(const int socket) : bool
SchdulerEngine
-memberName
+SetSchduler(const int32 id, void* schduler) : virtual bool
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+DelAnalyticalSchduler(const int32 id) : virtual bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : virtual bool
+CloseAnalyticalSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalAnalytical(const void* data, const int32 len) : virtual bool
+CheckOptimalAnalytical() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
AnalyticalSchdulerEngine
-schduler_mgr_ : static AnalyticalSchdulerManager*
-schduler_engine_ : static AnalyticalSchdulerEngine*
+GetCrawlerSchdulerManager() : static AnalyticalSchdulerManager*
+GetCrawlerSchdulerEngine() : static AnalyticalSchdulerEngine*


ForgeryManager
-lock_ : struct threadrw_t*
-forgery_cache_ : ForgeryCache*
+Init() : void
+FetchBatchIP(std::list<base_logic::ForgeryIP>* list) : void
+FetchBatchUA(std::list<base_logic::ForgeryUA>* list) : void
+RestBatchIP(std::list<base_logic::ForgeryIP>* list) : void
+SendProxyIP(const int socket, const int8 num) : void
+SendForgeryUA(const int socket, const int8 num) : void
StorageEngine
+Create(int32 type) : static StorageEngine*
+Connections(std::list<base::ConnAddr> & addrlist) : virtual bool
+Rekease() : virtual bool
ForgeryDB
-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+FectchBatchForgeryIP(std::list<base_logic::ForgeryIP>* list) : bool
+FectchBatchForgeryUA(std::list<base_logic::ForgeryUA>* list) : bool
+CallBackFectchBatchForgeryIP(void* param,base_logic::Value* value) : static void
+CallBackFectchBatchForgeryUA(void* param,base_logic::Value* value) : static void
Forgerylogic
-instance_ : static Forgerylogic*
-forgery_db_ : scoped_ptr<forgery_logic::ForgeryDB>
-schduler_engine_ : crawler_schduler::SchdulerEngine*
+GetInstance() : static Forgerylogic*
+FreeInstance() : static void
+OnForgeryConnect(struct server *srv, const int socket) : bool
+OnForgeryMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnForgeryClose(struct server *srv, const int socket) : bool
+OnBroadcastConnect(struct server *srv, const int socket,const void *data, const int len) : bool
+OnBroadcastMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnBroadcastClose(struct server *srv, const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server *srv, char* id, int opcode, int time) : bool
-OnGetForgeInfo(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : bool
-Init() : bool
SchdulerEngine
-memberName
+SetSchduler(const int32 id, void* schduler) : virtual bool
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+DelAnalyticalSchduler(const int32 id) : virtual bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : virtual bool
+CloseAnalyticalSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalAnalytical(const void* data, const int32 len) : virtual bool
+CheckOptimalAnalytical() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
DBStorageEngine
-memberName
+Create(int32 type) : static DBStorageEngine*
+Release() : virtual bool
+SQLExec(const char* sql) : virtual bool
+StoredProcedure() : virtual bool
+AddSPName(const char* sp_name) : virtual bool
+AddSPParam(const int32 var,const int32 type,const char* name,void*param,const int32 outstrlen = 0) : virtual bool
+CheckAffect(const int32 index,const char* name) : virtual bool
+Affected(unsigned long& rows) : virtual bool
+RecordCount() : virtual uint32
+FreeRes() : virtual bool
+FetchRows(void) : virtual db_row_t*
+CheckConnect(void) : virtual bool
+GetEntry(db_row_t *row,int cloidx) : virtual char*
+GetResult(int colidx) : virtual char*
DataControllerEngine
+Create(int32 type) : static DataControllerEngine
+Init() : static void
+Dest() : static void
+Release()=0 : virtual void
+InitParam(list<base::ConnAddr>*addrlist)=0 : virtual void
+ReadData(const int32 type,base_logic::Value* value,void(*storage_get)(void*, base_logic::Value*))=0 : virtual bool
+WriteData(const int32 type,base_logic::Value* value)=0 : virtual bool
PacketProsess
+PacketStream(const struct PacketHead* packet_head,void** packet_stream, int32* packet_stream_length) : static bool
+UnpackStream(const void* packet_stream, int32 len,struct PacketHead** packet_head) : static bool
+HexEncode(const void* bytes, size_t size) : static void
+DumpPacket(const struct PacketHead* packet_head) : static void
+ClearCrawlerTaskList(struct PacketHead* packet_head) : static void
+ClearHBaseAnalyticalTaskList(struct PacketHead* packet_head) : static void
+ClearLoginCookieList(struct PacketHead* packet_head) : static void
-CompressionStream(const uint8* unzip_data, uint64 unzip_len,uint8** zip_data) : static uint64
-DecompressionStream(const uint8* zip_data, uint64 zip_len,uint8** unzip_data) : static uint64
TaskTimeManager
-schduler_mgr_ : crawler_task_logic::TaskSchdulerManager*
-task_db_ : scoped_ptr<crawler_task_logic::CrawlerTaskDB>
+TaskTimeEvent(int opcode,int time) : void
-TimeFetchTask() : void
-TimeCheckTask() : void
-TimeFechTempTask() : void
-CleanNoEffectCrawler() : void


CrawlerScheduler
+set_id(const int32 id) : void
+set_socket(const int socket) : void
+set_ip(const std::string& ip) : void
+set_new_task_count(const int32 task_count) : void
+set_del_task_count(const int32 task_count) : void
+add_task_count() : void
+set_mac(const std::string& mac) : void
+set_password(const std::string& password) : void
+set_recv_last_time(const time_t recv_last_time) : void
+set_send_last_time(const time_t send_last_time) : void
+set_is_effective(bool is_effective) : void
+add_send_error_count() : void
+add_recv_error_count() : void
+id() const : const int32
+task_count() const : const int64
+socket() const : const int
+send_last_time() const : const time_t
+recv_last_time() const : const time_t
+send_error_count() const : const int32
+is_effective() const : const bool
ip() const : const string&
+mac() const : const string&
+password() const : const string&
+cmp(const CrawlerScheduler& t_scheduler,const CrawlerScheduler& r_scheduler) : static inline bool
ManagerDB
-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+CrawlerManagerLogin(void* data) : bool
+AnalyticalManagerLogin(void* data) : bool
+CallBackManagerLogin(void* param,base_logic::Value* value) : static void
Managerlogic
-instance_ : static Managerlogic*
-manager_db_ : scoped_ptr<manager_logic::ManagerDB>
-crawler_schduler_engine_ : crawler_schduler::SchdulerEngine*
-analytical_schduler_engine_ : analytical_schduler::SchdulerEngine*
-redis_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+GetInstance() : static Managerlogic*
+FreeInstance : static void
+OnManagerConnect(struct server *srv, const int socket) : bool
+OnManagerMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnManagerClose(struct server *srv, const int socket) : bool
+OnBroadcastConnect(struct server *srv, const int socket,const void *data, const int len) : bool
+OnBroadcastMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnBroadcastClose(struct server *srv, const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server *srv, char* id, int opcode, int time) : bool
-OnCrawlerReg(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : bool
-OnAnalyticalReg(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : bool
-OnGetMachineHardInfo(struct server* srv, int socket,struct PacketHead* packet, const void *msg = NULL,int32 len = 0) : bool
-OnTemplateReg(const char* mac, const char* password, const int socket,const int32 type, const int64 session_id) : bool
-Init() : bool
SchdulerEngine
-memberName
+SetSchduler(const int32 id, void* schduler) : virtual bool
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+DelAnalyticalSchduler(const int32 id) : virtual bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : virtual bool
+CloseAnalyticalSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalAnalytical(const void* data, const int32 len) : virtual bool
+CheckOptimalAnalytical() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
Data
+id_ : int32
+task_count_ : int64
+send_error_count_ : int32
+recv_error_count_ : int32
+is_effective_ : bool
+send_last_time_ : time_t
+recv_last_time_ : time_t
+socekt_ : int
+ip_ : string
+password_ : string
+mac_ : string
-refcount_ : int
+AddRef() : void
+Release() : void
DataControllerEngine
+Create(int32 type) : static DataControllerEngine
+Init() : static void
+Dest() : static void
+Release()=0 : virtual void
+InitParam(list<base::ConnAddr>*addrlist)=0 : virtual void
+ReadData(const int32 type,base_logic::Value* value,void(*storage_get)(void*, base_logic::Value*))=0 : virtual bool
+WriteData(const int32 type,base_logic::Value* value)=0 : virtual bool
PacketProsess
+PacketStream(const struct PacketHead* packet_head,void** packet_stream, int32* packet_stream_length) : static bool
+UnpackStream(const void* packet_stream, int32 len,struct PacketHead** packet_head) : static bool
+HexEncode(const void* bytes, size_t size) : static void
+DumpPacket(const struct PacketHead* packet_head) : static void
+ClearCrawlerTaskList(struct PacketHead* packet_head) : static void
+ClearHBaseAnalyticalTaskList(struct PacketHead* packet_head) : static void
+ClearLoginCookieList(struct PacketHead* packet_head) : static void
-CompressionStream(const uint8* unzip_data, uint64 unzip_len,uint8** zip_data) : static uint64
-DecompressionStream(const uint8* zip_data, uint64 zip_len,uint8** unzip_data) : static uint64
SendUtils
+SendFull(int socket, const char* buffer, size_t bytes) : static int32
+SendBytes(int socket, const void* bytes, int32 len,const char* file, int32 line) : static bool
+SendMessage(int socket, struct PacketHead* packet,const char* file, int32 line) : static bool
+SendHeadMessage(int socket, int32 operate_code,int32 type, int32 is_zip_encrypt, int64 session,int32 reserved, const char* file, int32 line) : static bool
+SendErronMessage(int socket, int32 operate_code,int32 type, int32 is_zip_encrypt, int64 session,int32 reserved, int32 error, const char* file, int32 line) : static bool


StroagerDB
-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>
+AddStorageInfo(const std::list<struct StorageUnit*>& list,const int32 type = 1) : bool
+GetHBaseInfo(std::list<base_logic::StorageHBase>* list) : bool
+RecordTempCrawlerTask(const std::list<base_logic::TaskInfo>& list) : bool
+GetTaskPlatTaskDescription(std::list<base_logic::TaskPlatDescription>* list) : bool
+CallBackGetHBaseInfo(void*param,base_logic::Value* value) : static void
+CallBackGetTaskPlatDescription(void* param,base_logic::Value* value) : static void
Storagerlogic
-instance_ : static Storagerlogic*
-stroager_db_ : scoped_ptr<storager_logic::StroagerDB>
-schduler_engine_ : crawler_schduler::SchdulerEngine*
+InitStorager(storage_logic::ShareDataManager* manager) : void
+GetInstance() : static Storagerlogic*
+FreeInstance() : static void
+OnStoragerConnect(struct server *srv, const int socket) : bool
+OnStoragerMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnStoragerClose(struct server *srv, const int socket) : bool
+OnBroadcastConnect(struct server *srv, const int socket,const void *data, const int len) : bool
+OnBroadcastMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnBroadcastClose(struct server *srv, const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server *srv, char* id, int opcode, int time) : bool
+StorageMethod(struct server* srv, int socket,struct PacketHead *packet,int32 type = 1,const void *msg = NULL,int32 len = 0) : viod
+GetAnalyticalHBaseInfo(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : void
+GetAnalyticalFTP(struct server* srv, int socket,struct PacketHead* packet, const void* msg = NULL,int32 len = 0) : void
+TempCrawlerTaskRecord(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : void
-Init() : bool
-RecordTempCrawlerTask(const std::list<base_logic::TaskInfo>& list) : void
SchdulerEngine
-memberName
+SetSchduler(const int32 id, void* schduler) : virtual bool
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+DelAnalyticalSchduler(const int32 id) : virtual bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : virtual bool
+CloseAnalyticalSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalAnalytical(const void* data, const int32 len) : virtual bool
+CheckOptimalAnalytical() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
TaskPlatDescription
+set_id(const int64 id) : void
+set_depth(const int8 depth) : void
+set_machine(const int8 machine) : void
+set_storage(const int8 storage) : void
+set_over(const int8 over) : void
+set_forge(const int8 forge) : void
+set_description(const std::string& description) : void
+id() const : const int64
+depth() const : const int8
+machine() const : const int8
+storage() : const : const int8
+over() const : const int8
+forge() const : const int8
+description() const : const string&
+ValueSerialization(base_logic::DictionaryValue* dict) : void
Data
+id_ : int64
+depth_ : int8
+machine_ : int8
+storage_ : int8
+over_ : int8
+forge_ : int8
+description_ : string
-refcount_ : int
+AddRef() : void
+Release() : void
ShareDataManager
-lock_ : struct threadrw_t*
-data_cache_ : ShareDataCache*
+BatchFectchTaskPlatInfo(std::list<base_logic::TaskPlatDescription>* list) : void
+BatchUpdateTaskInfo(std::list<base_logic::TaskInfo>* list) : void
-Init() : void
DataControllerEngine
+Create(int32 type) : static DataControllerEngine
+Init() : static void
+Dest() : static void
+Release()=0 : virtual void
+InitParam(list<base::ConnAddr>*addrlist)=0 : virtual void
+ReadData(const int32 type,base_logic::Value* value,void(*storage_get)(void*, base_logic::Value*))=0 : virtual bool
+WriteData(const int32 type,base_logic::Value* value)=0 : virtual bool
PacketProsess
+PacketStream(const struct PacketHead* packet_head,void** packet_stream, int32* packet_stream_length) : static bool
+UnpackStream(const void* packet_stream, int32 len,struct PacketHead** packet_head) : static bool
+HexEncode(const void* bytes, size_t size) : static void
+DumpPacket(const struct PacketHead* packet_head) : static void
+ClearCrawlerTaskList(struct PacketHead* packet_head) : static void
+ClearHBaseAnalyticalTaskList(struct PacketHead* packet_head) : static void
+ClearLoginCookieList(struct PacketHead* packet_head) : static void
-CompressionStream(const uint8* unzip_data, uint64 unzip_len,uint8** zip_data) : static uint64
-DecompressionStream(const uint8* zip_data, uint64 zip_len,uint8** unzip_data) : static uint64

TaskDB

-mysql_engine_ : scoped_ptr<base_logic::DataControllerEngine>

+FecthBatchTask(std::list<base_logic::TaskInfo>* list,const bool is_new = false) : bool
+FetchBatchHBase(std::list<base_logic::StorageHBase>* list) : bool
+UpdateHBaseState(std::list<base_logic::StorageHBase>* list) : bool
+CallBackFectchBatchTask(void*param,base_logic::Value* value) : static void
+CallBackFectchBatchHBase(void* param,base_logic::Value* value) : static void

Tasklogic

-instance_ : static Tasklogic
-task_db_ : scoped_ptr<task_logic::TaskDB>
-task_time_mgr_ : scoped_ptr<task_logic::TaskTimeManager>
-crawler_schduler_engine_ : crawler_schduler::SchdulerEngine*
-analytical_schduler_engine_ : analytical_schduler::SchdulerEngine*

+GetInstance() : static Tasklogic*
+FreeInstance() : static void
+OnTaskConnect(struct server *srv, const int socket) : bool
+OnTaskMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnTaskClose(struct server *srv, const int socket) : bool
+OnBroadcastConnect(struct server *srv, const int socket,const void *data, const int len) : bool
+OnBroadcastMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnBroadcastClose(struct server *srv, const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server *srv, char* id, int opcode, int time) : bool
-Init() : bool
-InitTask(task_logic::TaskSchdulerManager* schduler_mgr) : void
-TimeDistributionTask() : void
-TimeFetchTask() : void
-TimeDistributionHBase() : void
-TimeFetchHBase() : void
-ReplyTaskState(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : void
-ReplyAnalyticalState(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : void
-RelpyCrawlNum(struct server* srv, int socket,struct PacketHead *packet, const void *msg = NULL,int32 len = 0) : void

SchdulerEngine

+SetSchduler(const int32 id,void* schduler) : virtual bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+DelCrawlerSchduler(const int32 id) : virtual bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : virtual bool
+CloseCrawlerSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalCrawler(const void* data, const int32 len) : virtual bool
+CheckOptimalCrawler() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void

TaskTimeManager

-schduler_mgr_ : task_logic::TaskSchdulerManager*
-task_db_ : scoped_ptr<task_logic::TaskDB>

+TaskTimeEvent(int opcode, int time) : void
-TimeFetchTask() : void
-TimeFetchHBase() : void
-CleanHBase() : void
-CleanNoEffectCrawler() : void

DataControllerEngine

+Create(int32 type) : static DataControllerEngine
+Init() : static void
+Dest() : static void
+Release()=0 : virtual void
+InitParam(list<base::ConnAddr>*addrlist)=0 : virtual void
+ReadData(const int32 type,base_logic::Value* value,void(*storage_get)(void*, base_logic::Value*))=0 : virtual bool
+WriteData(const int32 type,base_logic::Value* value)=0 : virtual bool

SendUtils

+SendFull(int socket, const char* buffer, size_t bytes) : static int32
+SendBytes(int socket, const void* bytes, int32 len,const char* file, int32 line) : static bool
+SendMessage(int socket, struct PacketHead* packet,const char* file, int32 line) : static bool
+SendHeadMessage(int socket, int32 operate_code,int32 type, int32 is_zip_encrypt, int64 session,int32 reserved, const char* file, int32 line) : static bool
+SendErronMessage(int socket, int32 operate_code,int32 type, int32 is_zip_encrypt, int64 session,int32 reserved, int32 error, const char* file, int32 line) : static bool

TaskSchdulerEngine

-schduler_mgr_ : static TaskSchdulerManager*
-schduler_engine_ : static TaskSchdulerEngine*

+GetTaskSchdulerManager() : static TaskSchdulerManager*
+GetTaskSchdulerEngine() : static TaskSchdulerEngine*

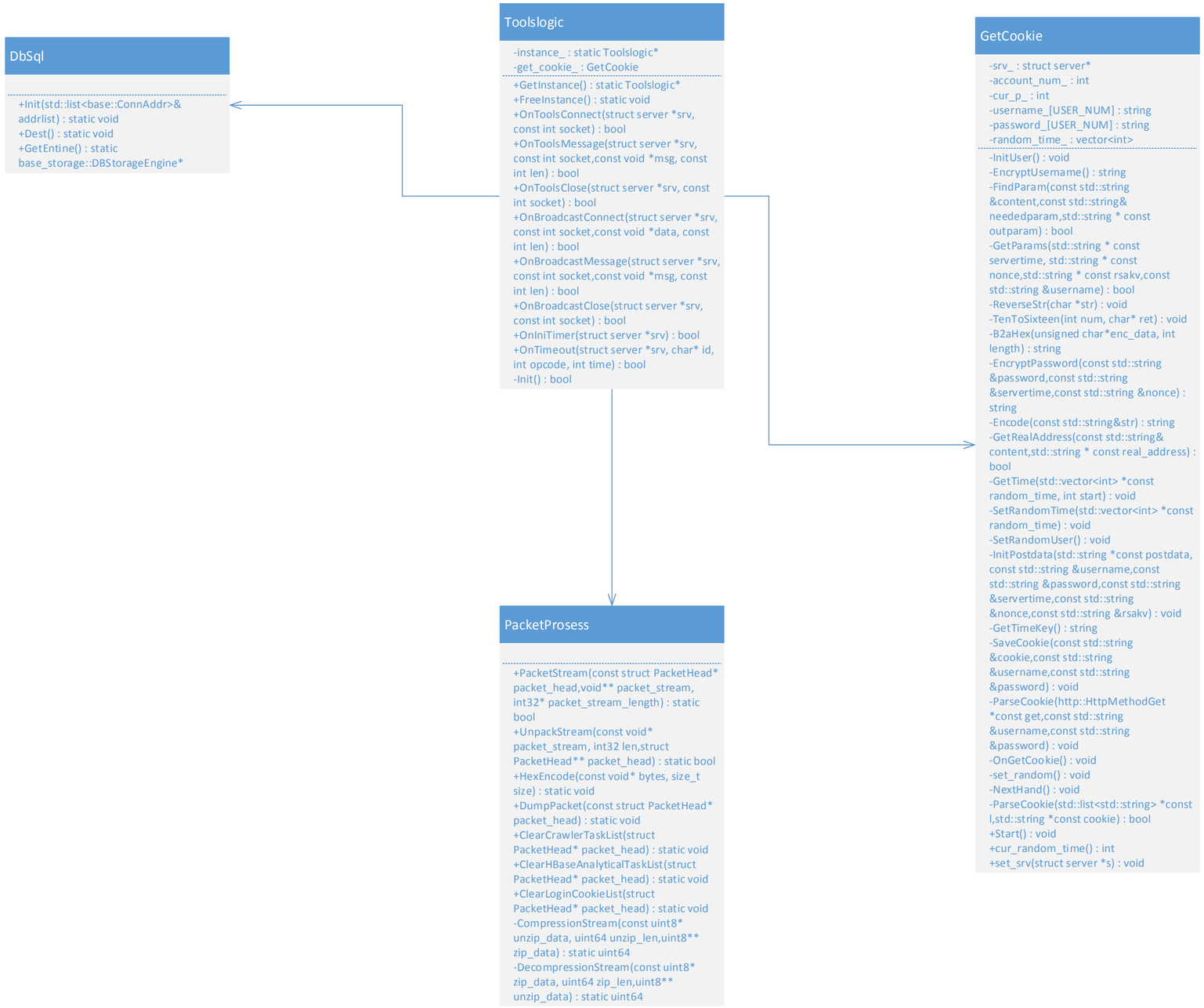
TaskSchdulerManager

-lock_ : struct threadrw_t*
-task_cache_ : TaskSchdulerCache*
-crawler_schduler_engine_ : crawler_schduler::SchdulerEngine*
-analytical_schduler_engine_ : analytical_schduler::SchdulerEngine*
-crawler_count_ : int32
-analytical_count_ : int32

+Init(crawler_schduler::SchdulerEngine* crawler_engine,analytical_schduler::SchdulerEngine* analytical_engine) : void
+FetchBatchTask(std::list<base_logic::TaskInfo>* list,bool is_first = false) : void
+FetchBatchHbase(std::list<base_logic::StorageHBase>* list) : void
+RemoveAnalyticalHBase(const int64 id) : bool
+DistributionTask() : bool
+DistibutionHBase() : bool
+SwapRemoveHBase(std::list<base_logic::StorageHBase>* list,bool is_clean = true) : void
+RecyclingTask() : void
+AlterTaskState(const int64 task_id, const int8 state) : bool
+AlterCrawlNum(const int64 task_id, const int64 num) : bool
+CheckIsEffective() : void
-Init() : void

SchdulerEngine

+SetSchduler(const int32 id, void* schduler) : virtual bool
+SetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+GetAnalyticalSchduler(const int32 id,base_logic::AnalyticalSchduler* schduler) : virtual bool
+DelAnalyticalSchduler(const int32 id) : virtual bool
+FindAnalyticalSchduler(const int socket,base_logic::AnalyticalSchduler* schduler) : virtual bool
+CloseAnalyticalSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalAnalytical(const void* data, const int32 len) : virtual bool
+CheckOptimalAnalytical() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void


DbSql
+Init(std::list<base::ConnAddr>& addrlist) : static void
+Dest() : static void
+GetEntine() : static base_storage::DBStorageEngine*
Toolslogic
-instance_ : static Toolslogic*
-get_cookie_ : GetCookie
+GetInstance() : static Toolslogic*
+FreeInstance() : static void
+OnToolsConnect(struct server *srv, const int socket) : bool
+OnToolsMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnToolsClose(struct server *srv, const int socket) : bool
+OnBroadcastConnect(struct server *srv, const int socket,const void *data, const int len) : bool
+OnBroadcastMessage(struct server *srv, const int socket,const void *msg, const int len) : bool
+OnBroadcastClose(struct server *srv, const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server *srv, char* id, int opcode, int time) : bool
-Init() : bool
GetCookie
-srv_ : struct server*
-account_num_ : int
-cur_p_ : int
-username_[USER_NUM] : string
-password_[USER_NUM] : string
-random_time_ : vector<int>
-InitUser() : void
-EncryptUsername() : string
-FindParam(const std::string &content,const std::string& neededparam,std::string * const outparam) : bool
-GetParams(std::string * const servertime, std::string * const nonce,std::string * const rsakv,const std::string &username) : bool
-ReverseStr(char *str) : void
-TenToSixteen(int num, char* ret) : void
-B2aHex(unsigned char*enc_data, int length) : string
-EncryptPassword(const std::string &password,const std::string &servertime,const std::string &nonce) : string
-Encode(const std::string&str) : string
-GetRealAddress(const std::string& content,std::string * const real_address) : bool
-GetTime(std::vector<int> *const random_time, int start) : void
-SetRandomTime(std::vector<int> *const random_time) : void
-SetRandomUser() : void
-InitPostdata(std::string *const postdata, const std::string &username,const std::string &password,const std::string &servertime,const std::string &nonce,const std::string &rsakv) : void
-GetTimeKey() : string
-SaveCookie(const std::string &cookie,const std::string &username,const std::string &password) : void
-ParseCookie(http::HttpMethodGet *const get,const std::string &username,const std::string &password) : void
-OnGetCookie() : void
-set_random() : void
-NextHand() : void
-ParseCookie(std::list<std::string> *const l,std::string *const cookie) : bool
+Start() : void
+cur_random_time() : int
+set_srv(struct server *s) : void
PacketProsess
+PacketStream(const struct PacketHead* packet_head,void** packet_stream, int32* packet_stream_length) : static bool
+UnpackStream(const void* packet_stream, int32 len,struct PacketHead** packet_head) : static bool
+HexEncode(const void* bytes, size_t size) : static void
+DumpPacket(const struct PacketHead* packet_head) : static void
+ClearCrawlerTaskList(struct PacketHead* packet_head) : static void
+ClearHBaseAnalyticalTaskList(struct PacketHead* packet_head) : static void
+ClearLoginCookieList(struct PacketHead* packet_head) : static void
-CompressionStream(const uint8* unzip_data, uint64 unzip_len,uint8** zip_data) : static uint64
-DecompressionStream(const uint8* zip_data, uint64 zip_len,uint8** unzip_data) : static uint64


TaskTimeManager
-schduler_mgr_ : crawler_task_logic::TaskSchdulerManager*
-task_db_ : scoped_ptr<detail_task_logic::DetailTaskDB>
+TaskTimeEvent(int opcode,int time) : void
-TimeFetchTask() : void
-TimeCheckTask() : void
-CleanNoEffectCrawler() : void
SchdulerEngine
-memberName
+SetSchduler(const int32 id,void* schduler) : virtual bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : virtual bool
+DelCrawlerSchduler(const int32 id) : virtual bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : vitual bool
+CloseCrawlerSchduler(const int socket) : virtual bool
+SetRecvTime(const int socket) : virtual bool
+SetSendTime(const int socket) : virtual bool
+CheckHeartPacket(void) : virtual bool
+SendOptimalCrawler() : virtual bool
+CheckOptimalCrawler() : virtual bool
+SetSendErrorCount(int socket) : virtual bool
+SetRecvErrorCount(int socket) : virtual bool
+CheckIsEffective() : virtual void
DetailTaskDB
-kafka_consumer_news_ : kafka_consumer
-task_platform_ : TASKPLAT_MAP
-task_platform_inited_ : bool
-login_info_map_ : LOGININFO_MAP
+FecthBatchTask(list<base_logic::TaskInfo>* list) : bool
+RecordTaskState(base_logic::TaskInfo& task,const int32 type) : bool
+GetTaskPlatTaskDescription(list<base_logic::TaskDescription>*list) : bool
+BatchFectchTaskPlatInfo(list<base_logic::TaskPlatDescription>* list) : void
+BatchUpdateTaskInfo(list<base_logic::TaskInfo>* list)
+GetLoginDescription() : void
+CallBackGetTaskPlatDescription(void* param,base_logic::Value* value) : static void
+CallBackGetLoginDescription(void* param,base_logic::Value* value) : static void
DetailTasklogic
-instance_ : static DetailTaskLogic*
-task_db_ : scoped_ptr<detail_task_logic::DetailTaskDB>
-task_time_mgr_ : scoped_ptr<detail_task_logic::TaskTimeManager>
-detail_schduler_engine_ : crawler_schduler::SchdulerEngine*
+GetInstance() : static CrawlerTasklogic*
+FreeInstance() : static void
+OnTaskConnect(struct server *srv,const int socket) : bool
+OnTaskMessage(struct server *srv,const int socket,const void *msg,const int len) : bool
+OnTaskClose(struct server *srv,const int socket) : bool
+OnBroadcastConnect(struct server *srv,const int socket, const void *data,const int len) : bool
+OnBroadcastMessage(struct server *srv,const int socket,const void *msg,const int len) : bool
+OnBroadcastClose(struct server *srv,const int socket) : bool
+OnIniTimer(struct server *srv) : bool
+OnTimeout(struct server * srv,char * id,int opcode,int time) : bool
-Init() : bool
-InitTask(crawler_task_logic::TaskSchdulerManager* schduler_mgr) : void
-TimeDistributionTask() : void
-TimeFetchTask() : void
-ReplyDetailState(struct server* srv,int socket,struct PacketHead *packet, const void *msg=NULL,int32 len=0) : void
-ReplyCrawlNum(struct server* srv,int socket,struct PacketHead *packet,const void *msg=NULL,int32 len=0) : void
FileConfig
-config_ : static FileConfig*
+mysql_db_list_ : list<base::ConnAddr>
+mem_list_ : list<base::ConnAddr>
+redis_list_ : list<base::ConnAddr>
+mssql_db_list_ : list<base::ConnAddr>
+certificate_path_ : string
+idp_url_ : string
+sp_url_ : string
+mood_path_ : string
+style_path_ : string
+usr_local_music_path_ : string
+host_ : string
+port_ : string
+LoadConfig(string& path) : bool
+GetFileConfig() : static FileConfig*
SchdulerEngineImpl
-memberName
+SetSchduler(const int32 id,void* schduler) : bool
+SetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+GetCrawlerSchduler(const int32 id,base_logic::CrawlerScheduler* schduler) : bool
+DelCrawlerSchduler(const int32 id) : bool
+FindCrawlerSchduler(const int socket,base_logic::CrawlerScheduler* schduler) : bool
+CloseCrawlerSchduler(const int socket)
+SetRecvTime(const int socket) : bool
+SetSendTime(const int socket) : bool
+CheckHeartPacket() : bool
+SendOptimalCrawler(const void* data,const int32 len) : bool
+CheckOptimalCrawler() : bool
+SetSendErrorCount(int socket) : bool
+SetRecvErrorCount(int socket) : bool
+CheckIsEffective() : void