

Complete Class Diagram

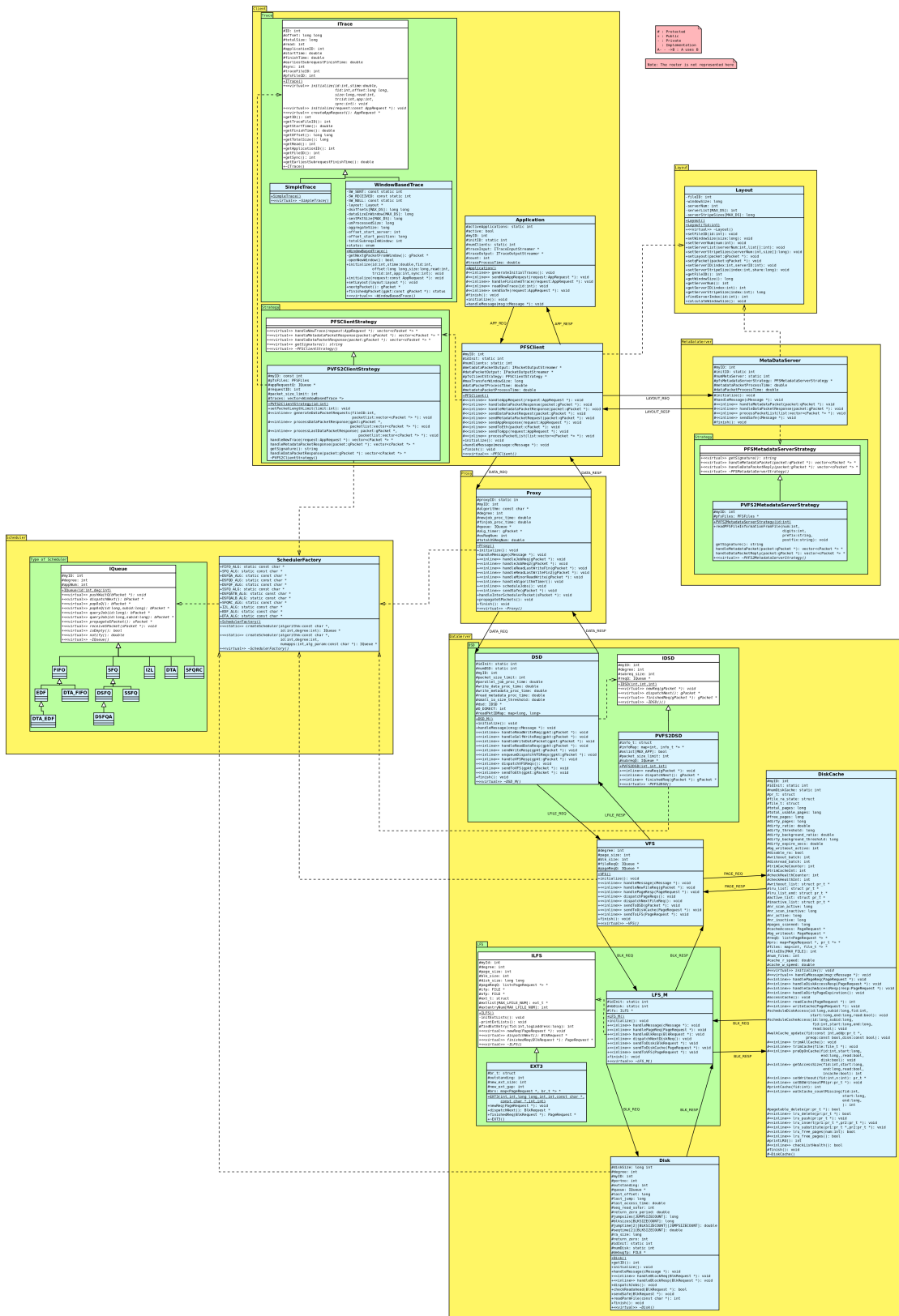


Figure 1: Class Diagram PFSSim

Class Diagram Client Package

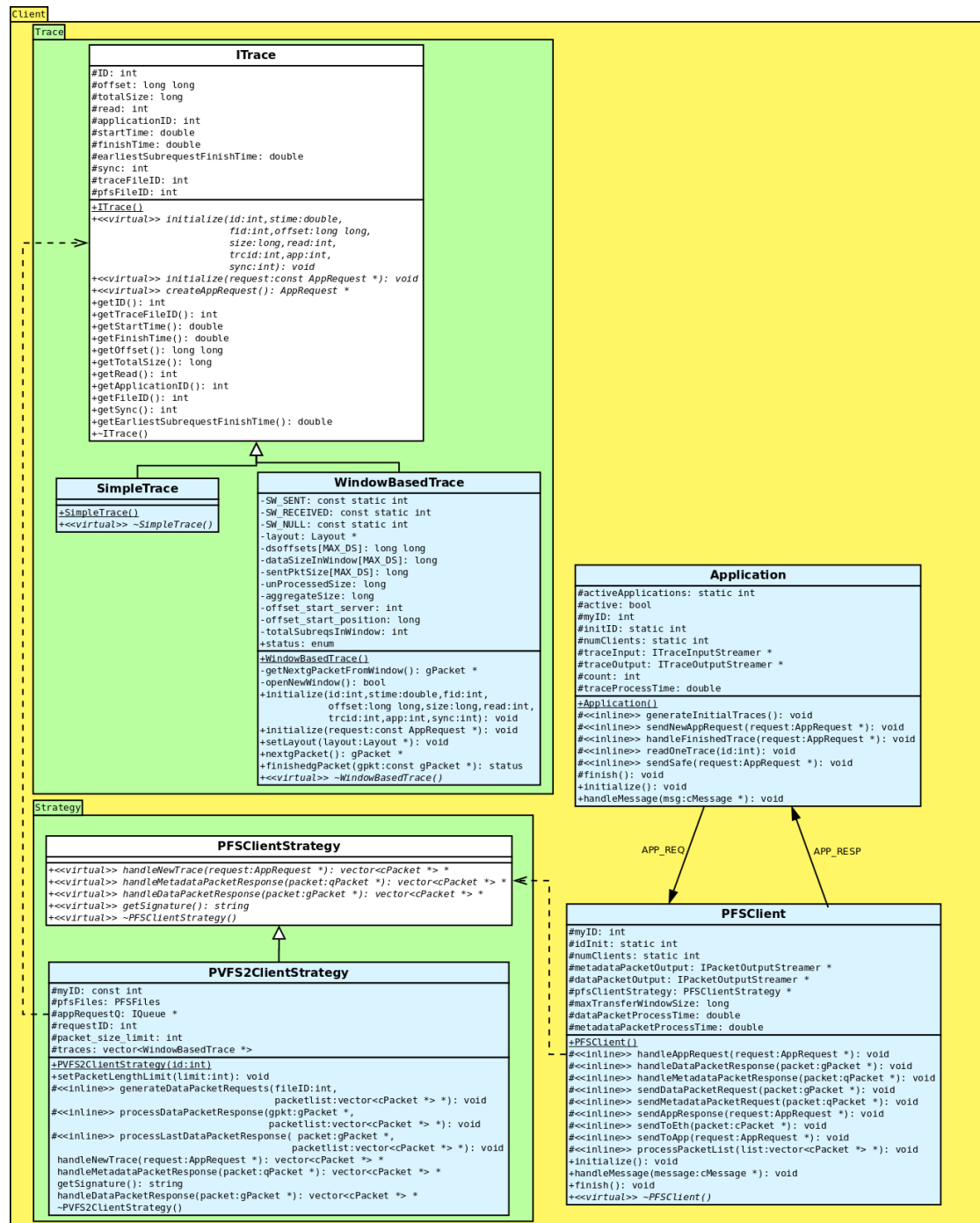


Figure 2: Class Diagram PFSSim Package Client

DataServer

```

graph TD
    DSD[DSD] -- "LFILF_REQ" --> VFS[VFS]
    IDSD[IDSD] -- "LFILF_RESP" --> VFS
    VFS -- "BLK_REQ" --> LFS_M[LFS_M]
    LFS_M -- "BLK_RESP" --> VFS
    VFS -- "PAGE_REQ" --> DC[DiskCache]
    DC -- "PAGE_RESP" --> VFS
    LFS_M -- "BLK_REQ" --> Disk[Disk]
    Disk -- "BLK_RESP" --> LFS_M
    EXT3[EXT3] --> ILFS[ILFS]
    ILFS --> LFS[LFS]
    LFS --> LFS_M
  
```

DSD

```

#idInit: static int
#numDSD: static int
#myID: int
#packet_size limit: int
#parallel_job_proc_time: double
#write_data_proc_time: double
#write_metadata_proc_time: double
#small_io_size_threshold: double
#dso: IDSD
#IO_DIRECT: int
#readPktIDMap: map<long, long>

~DSD_M()
+initialize(): void
+handleMessage(cmsg: CMessage *): void
+<<inline>> handleReadWriteReq(gpkt: GPacket *): void
+<<inline>> handleWriteDataPacket(gpkt: GPacket *): void
+<<inline>> handleReadDataResp(gpkt: GPacket *): void
+<<inline>> sendWriteReq(gpkt: GPacket *): void
+<<inline>> enqueueDispatchVFSReqs(gpkt: GPacket *): void
+<<inline>> handleVFSResp(gpkt: GPacket *): void
+<<inline>> dispatchVFSReqs(): void
+<<inline>> sendToVFS(gpkt: GPacket *): void
+<<inline>> sendToEth(gpkt: GPacket *): void
+finish(): void
+<<virtual>> ~DSD_M()

```

IDSD

```

#myID: int
#degree: int
#subreq_size: int
#reqQ: IOQueue *

+IDSD_Init(int)
+<<virtual>> newReq(gPacket *) : void
+<<virtual>> dispatchNext(): gPacket *
+<<virtual>> finishReq(gPacket *) : gPacket *
+<<virtual>> ~IDSD()

```

PVF2DSD

```

#info_t: struct
+info_t {map: map<int, info_t >*
+oslist[MAX_APP]: bool
+packet_size_limit: int
+subreqQ: IOQueue *}
+PVF2DSD(int, int, int)
+<<inline>> newReq(gPacket *) : void
+<<inline>> dispatchNext(): gPacket *
+<<inline>> finishReq(gPacket *) : gPacket *
+<<virtual>> ~PVF2DSD()

```

VFS

```

#degree: int
#page_size: int
#blk_size: int
#fileReqQ: IOQueue *
#pageReqQ: IOQueue *
+VFS()
+initialize(): void
+<<inline>> handleMessage(cmessage *) : void
+<<inline>> handleFileReq(gPacket *) : void
+<<inline>> handlePageReq(PageRequest *) : void
+<<inline>> dispatchPageReqs(): void
+<<inline>> dispatchNextFileReq(): void
+<<inline>> sendTOSDB(gPacket *) : void
+<<inline>> sendToDiskCache(PageRequest *) : void
+<<inline>> sendToLFS(PageRequest *) : void
+finish(): void
+<<virtual>> ~VFS()

```

LFS

```

#myID: int
#degree: int
#page_size: int
#blk_size: int
#disk_size: long long
#pageReqQ: List<PageRequest *>
#fp: FILE *
#ext_t: struct
+ext_t {MAX_FILE_NUM: ext_t *
+extEntryNum[MAX_FILE_NUM]: int}
+LFS()
+initExtLists(): void
+printExtLists(): void
+findExtEntry(fid: int, logaddress: long): void
+<<virtual>> newReq(PageRequest *) : void
+<<virtual>> dispatchNext(): BLKRequest *
+<<virtual>> finishReq(BLKRequest *) : PageRequest *
+<<virtual>> ~LFS()

```

ILFS

```

#myID: int
#degree: int
#page_size: int
#blk_size: int
#disk_size: long long
#pageReqQ: List<PageRequest *>
#fp: FILE *
#ext_t: struct
+ext_t {MAX_FILE_NUM: ext_t *
+extEntryNum[MAX_FILE_NUM]: int}
+ILFS()
+initExtLists(): void
+printExtLists(): void
+findExtEntry(fid: int, logaddress: long): void
+<<virtual>> newReq(PageRequest *) : void
+<<virtual>> dispatchNext(): BLKRequest *
+<<virtual>> finishReq(BLKRequest *) : PageRequest *
+<<virtual>> ~ILFS()

```

EXT3

```

#pr: t: struct
#outstanding: int
#new_ext_size: int
#new_ext_gaps: int
#prs: map<PageRequest *, pr_t >*
+EXT3(int, int, long, long, int, const char *,
const char *, int, int)
+newReq(PageRequest *) : void
+dispatchNext(): BLKRequest *
+finishReq(BLKRequest *) : PageRequest *
+~EXT3()

```

LFS_M

```

#idInit: static int
#mDisk: static int
#lfs: LFS *
+LFS_M()
+initialize(): void
+<<inline>> handleMessage(cmessage *) : void
+<<inline>> handlePageReq(PageRequest *) : void
+<<inline>> handleBlkReq(BLKRequest *) : void
+<<inline>> dispatchNextFileReq(): void
+<<inline>> sendToDiskCache(PageRequest *) : void
+<<inline>> sendToVFS(PageRequest *) : void
+finish(): void
+<<virtual>> ~LFS_M()

```

DiskCache

```

#myID: int
#idInit: static int
#mDiskCache: static int
#pr: t: struct
#file_ra_state: struct
#file_t: struct
#total_pages: long
#total_usable_pages: long
#free_pages: long
#dirty_pages: long
#dirty_ratio: double
#dirty_threshold: long
#dirty_background_ratio: double
#dirty_background_threshold: long
#dirty_expire_secs: double
#pg_writout_active: int
#disable_ra: bool
#writout_batch: int
#isReadBatch: int
#trimCacheCounter: int
#trimCachetInt: int
#checkHealthCounter: int
#checkHealthInt: int
#writout_list: struct pr_t *
#ru_list: struct pr_t *
#ru_list_end: struct pr_t *
#active_list: struct pr_t *
#inactive_list: struct pr_t *
#nr_scan_active: long
#nr_scan_inactive: long
#nr_active: long
#nr_inactive: long
#pages_scanned: long
#cacheAccess: PageRequest *
#pg_writout: PageRequest *
#reqQ: List<PageRequest *>
#prs: map<PageRequest *, pr_t >*
#files: map<int, file_t >*
#fileIDs(MAX_FILE): int
#num_files: int
#cache_r_speed: double
#cache_w_speed: double
+<<virtual>> initialize(): void
+<<virtual>> handleMessage(msg: CMessage *) : void
+<<inline>> handlePageReq(PageRequest *) : void
+<<inline>> handleLDiskAccessReq(PageRequest *) : void
+<<inline>> handleLDiskAccessResp(resp: PageRequest *) : void
+<<inline>> handleDirtyPageExpiration(): void
+AccessCache(): void
+<<inline>> readCache(PageRequest *) : int
+<<inline>> writeCache(PageRequest *) : void
#scheduleLDiskAccess(id: long, subid: long, fid: int,
start: long, end: long, read: bool): void
#scheduleCacheAccess(id: long, subid: long, fid: int,
start: long, end: long, read: bool): void
#wkCache_update(fid: const int, addp: pr_t *,
preop: const bool, disk: const bool): void
+<<inline>> trimAllCache(): void
+<<inline>> trimCache(file: file_t *, pr_t *: void
end: long, read: bool,
disk: bool): void
+<<inline>> prodOnCache(fid: int, start: long,
end: long, read: bool,
incache: bool): int
+<<inline>> setWritout(fid: int, start: long,
end: long, read: bool,
pr: pr_t *) : void
+<<inline>> setBGWritoutPR(pr: pr_t *) : void
+printCache(fid: int): int
+<<inline>> wkCache_countMissing(fid: int,
start: long,
end: long,
): int
#pagetable_delete(pr: pr_t *) : bool
+<<inline>> ru_delete(pr: pr_t *) : bool
+<<inline>> ru_push(pr: pr_t *) : void
+<<inline>> ru_insert(pr: pr_t *, pr2: pr_t *) : void
+<<inline>> ru_substitute(pr: pr_t *, pr2: pr_t *) : void
+<<inline>> ru_free_pages(num: int): bool
+<<inline>> ru_free_pages(): bool
+printRU(): int
+<<inline>> checkListHealth(): bool
+finish(): void
+~DiskCache()

```

Disk

```

#diskSize: long int
#degree: int
#myID: int
#portno: int
#outstanding: int
#queue: IOQueue *
#last_offset: long
#last_jump: long
#last_access_time: double
#seq_read_sofar: int
#return_zero_period: double
#jumpsizes[JUMPSIZECOUNT]: long
#blksizes[BLKSIZECOUNT]: long
+jumpInel[2][BLKSIZECOUNT][JUMPSIZECOUNT]: double
+seqtime[2][BLKSIZECOUNT]: double
#ra_size: long
#return_zero: int
#idInit: static int
#numDisk: static int
#debugfp: FILE *
+Disk()
+getID(): int
+initialize(): void
+handleMessage(cmessage *) : void
+<<inline>> handleBlockReq(BLKRequest *) : void
+<<inline>> handleBlockResp(BLKRequest *) : void
+dispatchJobs(): void
+checkReadahead(BLKRequest *) : bool
+sendSafe(BLKRequest *) : void
+readParFile(const char *) : int
+finish(): void
+<<virtual>> ~Disk()

```

3

Class Diagram MetaData Server Package

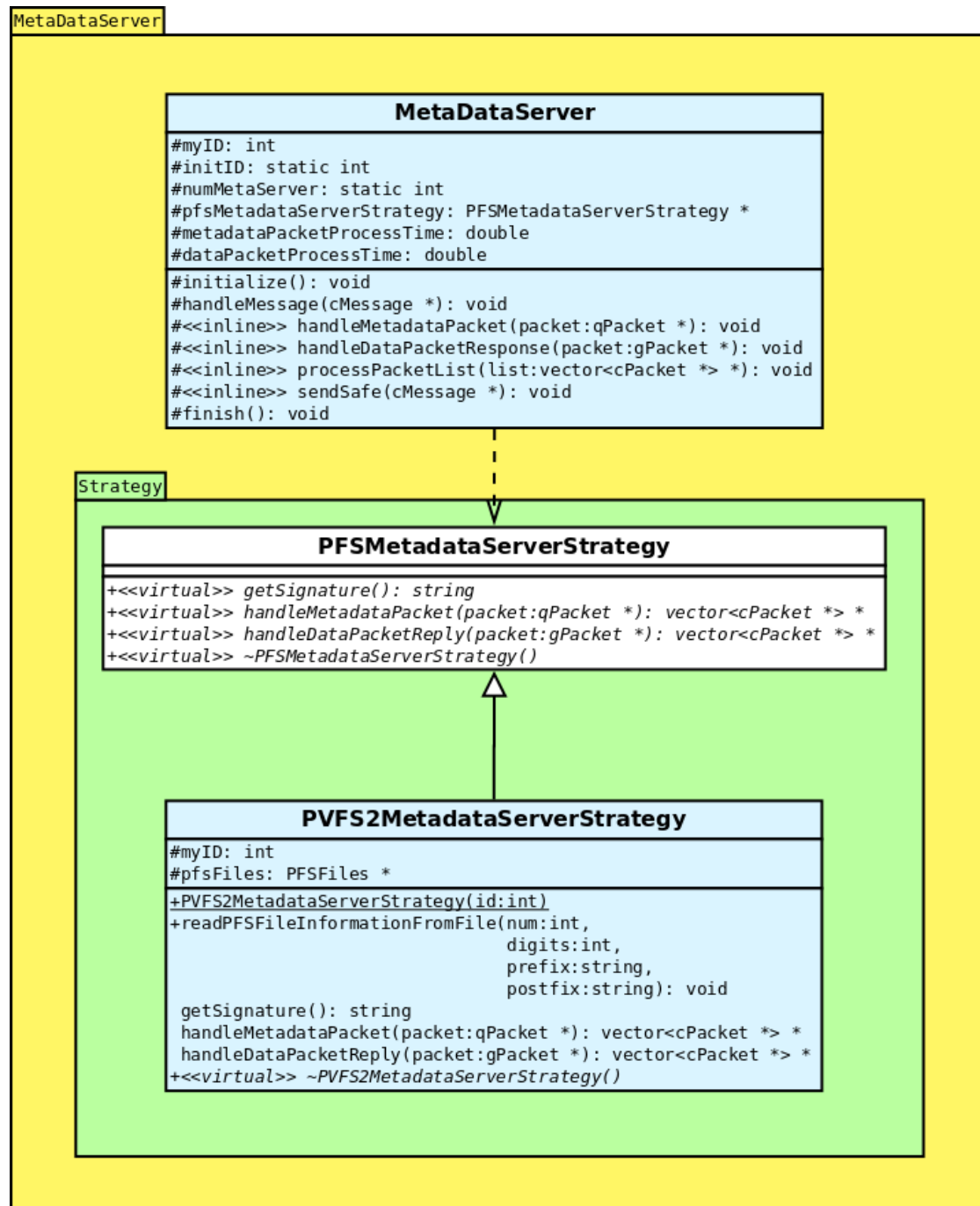


Figure 4: Class Diagram PFSSim Package MetaData Server

Class Diagram Layout Package



Figure 5: Class Diagram PFSSim Package Layout

Class Diagram Proxy Package

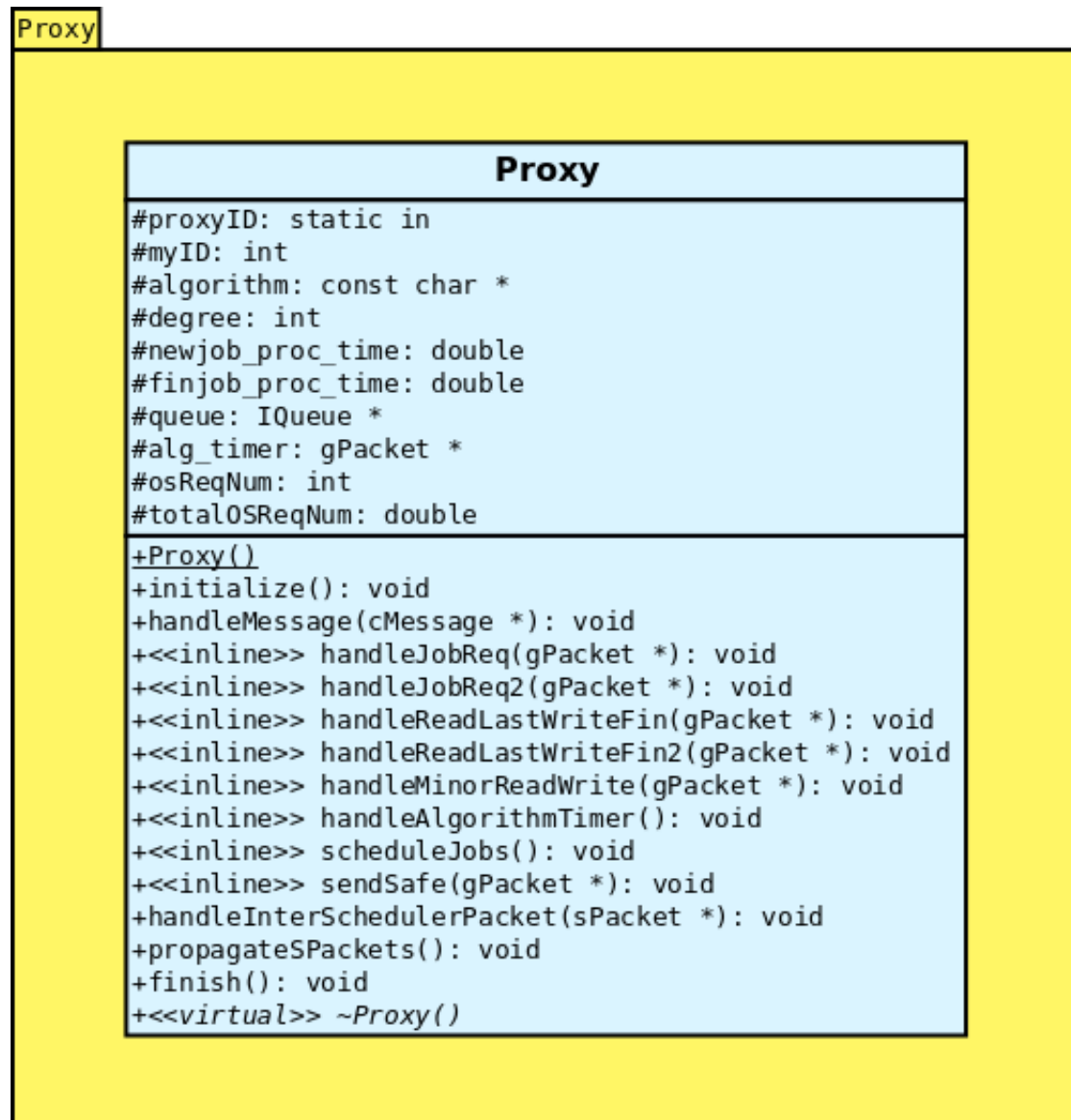


Figure 6: Class Diagram PFSSim Package Proxy

Class Diagram Scheduler Package

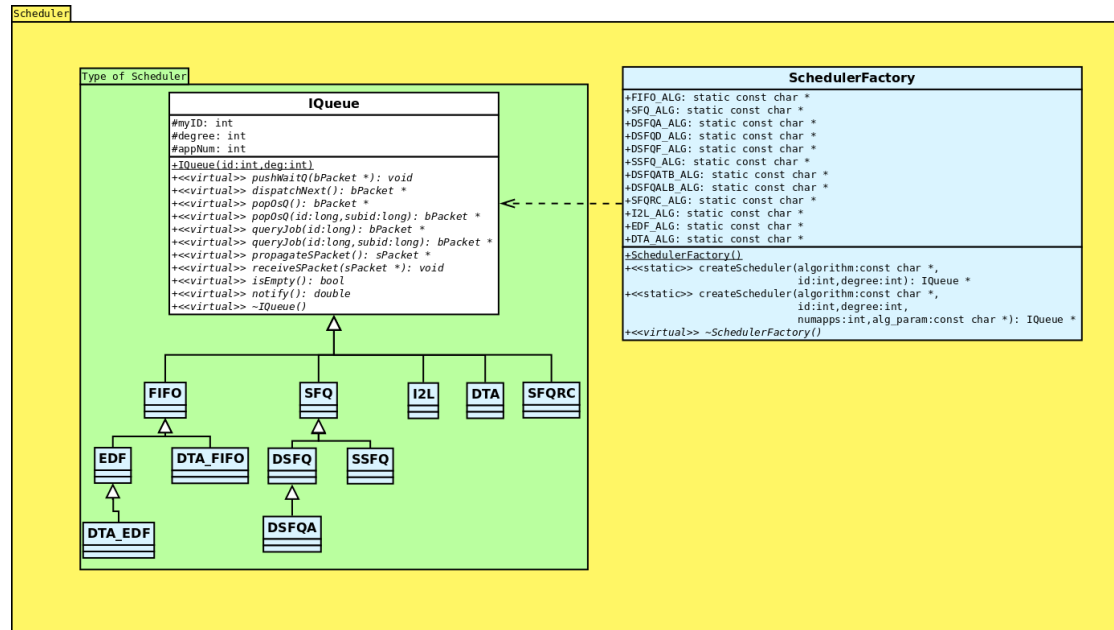


Figure 7: Class Diagram PFSSim Package Schedulers