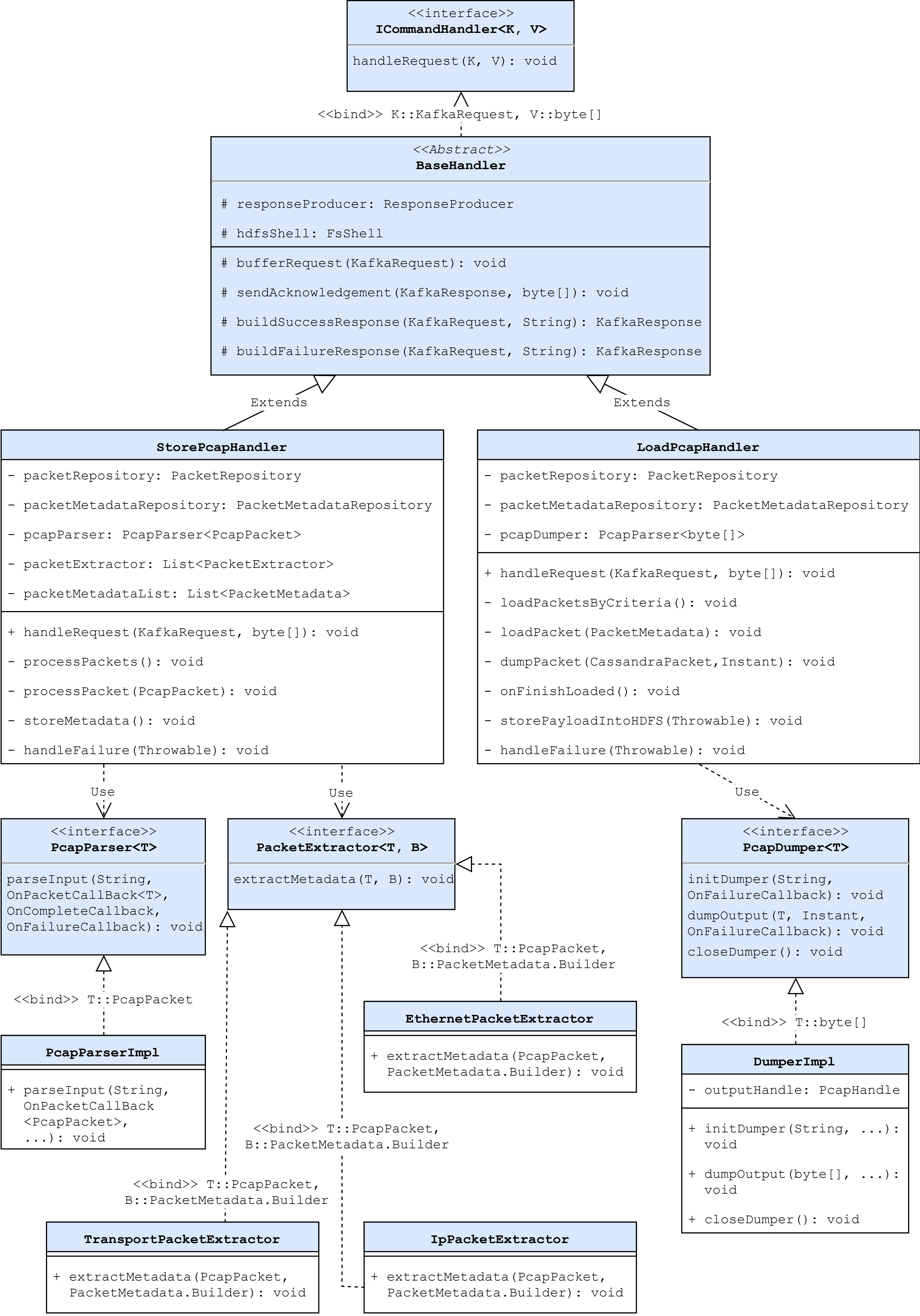
<<interface>>
ICommandHandler<K, V>
handleRequest(K, V): void
<<bind>> K::KafkaRequest, V::byte[]
<<Abstract>>
BaseHandler
# responseProducer: ResponseProducer
# hdfsShell: FsShell
# bufferRequest(KafkaRequest): void
# sendAcknowledgement(KafkaResponse, byte[]): void
# buildSuccessResponse(KafkaRequest, String): KafkaResponse
# buildFailureResponse(KafkaRequest, String): KafkaResponse
Extends
Extends
StorePcapHandler
- packetRepository: PacketRepository
- packetMetadataRepository: PacketMetadataRepository
- pcapParser: PcapParser<PcapPacket>
- packetExtractor: List<PacketExtractor>
- packetMetadataList: List<PacketMetadata>
+ handleRequest(KafkaRequest, byte[]): void
- processPackets(): void
- processPacket(PcapPacket): void
- storeMetadata(): void
- handleFailure(Throwable): void
LoadPcapHandler
- packetRepository: PacketRepository
- packetMetadataRepository: PacketMetadataRepository
- pcapDumper: PcapParser<byte[]>
+ handleRequest(KafkaRequest, byte[]): void
- loadPacketsByCriteria(): void
- loadPacket(PacketMetadata): void
- dumpPacket(CassandraPacket,Instant): void
- onFinishLoaded(): void
- storePayloadIntoHDFS(Throwable): void
- handleFailure(Throwable): void
Use
Use
Use
<<interface>>
PcapParser<T>
parseInput(String,
OnPacketCallBack<T>,
OnCompleteCallback,
OnFailureCallback): void
<<interface>>
PacketExtractor<T, B>
extractMetadata(T, B): void
<<interface>>
PcapDumper<T>
initDumper(String,
OnFailureCallback): void
dumpOutput(T, Instant,
OnFailureCallback): void
closeDumper(): void
<<bind>> T::PcapPacket,
B::PacketMetadata.Builder
<<bind>> T::PcapPacket
<<bind>> T::byte[]
EthernetPacketExtractor
+ extractMetadata(PcapPacket,
PacketMetadata.Builder): void
PcapParserImpl
+ parseInput(String,
OnPacketCallBack
<PcapPacket>,
...): void
DumperImpl
- outputHandle: PcapHandle
+ initDumper(String, ...):
void
+ dumpOutput(byte[], ...):
void
+ closeDumper(): void
<<bind>> T::PcapPacket,
B::PacketMetadata.Builder
<<bind>> T::PcapPacket,
B::PacketMetadata.Builder
TransportPacketExtractor
+ extractMetadata(PcapPacket,
PacketMetadata.Builder): void
IpPacketExtractor
+ extractMetadata(PcapPacket,
PacketMetadata.Builder): void