Dash Implementation（Exo） Investigation
—— MPD Implementation in EXO

Format
+id:String
+mimeType:String
+bitrate:int
+width:int
+height:int
+frameRate:float
+audioChannels:int
+audioSamplingRate:int
+codecs:String
+language:String
+Format(String,String,int,int,float,int,int,int)
+Format(String,String,int,int,float,int,int,int,String)
+Format(String,String,int,int,float,int,int,int,String,String)
+hashCode():int
+equals(Object):boolean

<<接口>>
FormatWrapper
+getFormat():Format

DrmInitData.SchemeInitData
+mimeType:String
+data:byte[]
+SchemeInitData(String,byte[])
+equals(Object):boolean
+hashCode():int

ContentProtection
+schemeUriId:String
+uuid:UUID
+data:SchemeInitData
+ContentProtection(String,UUID,SchemeInitData)
+equals(Object):boolean
+hashCode():int

Representation
+contentId:String
+revisionId:long
+format:Format
+presentationTimeOffsetUs:long
-cacheKey:String
-initializationUri:RangedUri
+newInstance(String,long,Format,SegmentBase):Representation
+newInstance(String,long,Format,SegmentBase,String):Representation
-Representation(String,long,Format,SegmentBase,String)
+getFormat():Format
+getInitializationUri():RangedUri
+getIndexUri():RangedUri
+getIndex():DashSegmentIndex
+getCacheKey():String

AdaptationSet
+TYPE_UNKNOWN:int=-1
+TYPE_VIDEO:int=0
+TYPE_AUDIO:int=1
+TYPE_TEXT:int=2
+id:int
+type:int
+representations:List<Representation>
+contentProtections:List<ContentProtection>
+AdaptationSet(int,int,List<Representation>,List<ContentProtection>)
+AdaptationSet(int,int,List<Representation>)
+hasContentProtection():boolean

RangedUri
+start:long
+length:long
-baseUri:String
-referenceUri:String
-hashCode:int
+RangedUri(String,String,long,long)
+getUri():Uri
+getUriString():String
+attemptMerge(RangedUri):RangedUri
+hashCode():int
+equals(Object):boolean

MediaPresentationDescription
+availabilityStartTime:long
+duration:long
+minBufferTime:long
+dynamic:boolean
+minUpdatePeriod:long
+utcTiming:UtcTimingElement
+location:String
+periods: List<Period>
+MediaPresentationDescription(long,long,long,boolean,long,long,UtcTimingElement,String,List<Period>)
+getNextManifestUri():String
+getPeriodCount():int
+getPeriod(int):Period
+getPeriodDuration(int):long

ContentProtectionsBuilder
-adaptationSetProtections: ArrayList<ContentProtection>
-representationProtections: ArrayList<ContentProtection
-currentRepresentationProtections: ArrayList<ContentProtection>
-representationProtectionsSet:boolean
+addAdaptationSetProtection(ContentProtection):void
+addRepresentationProtection(ContentProtection):void
+endRepresentation()
+build():ArrayList<ContentProtection>
-maybeAddContentProtection(List<ContentProtection>,ContentProtection):void

<<接口>>
DashSegmentIndex
+INDEX_UNBOUNDED:int=-1;
+getSegmentNum(long,long):int
+getTimeUs(int):long
+getDurationUs(int,long):long
+getSegmentUrl(int):RangedUri
+getFirstSegmentNum():int
+getLastSegmentNum(long):int
+isExplicit():boolean

Representation.SingleSegmentRepresentation
+uri:Uri
+contentLength:long
-indexUri:RangedUri
-segmentIndex:DashSingleSegmentIndex
+newInstance(String,long,Format,String,long,long,long,String,long):SingleSegmentRepresentation
+SingleSegmentRepresentation(String,long,Format,SingleSegmentBase,String,long)
+getIndexUri():RangedUri
+getIndex():DashSegmentIndex

Representation.MultiSegmentRepresentation
-segmentBase:MultiSegmentBase
+MultiSegmentRepresentation(String,long,Format,MultiSegmentBase,String)
+getIndexUri():RangedUri
+getIndex():DashSegmentIndex
+getSegmentUrl(int):RangedUri
+getSegmentNum(long,long):int
+getTimeUs(int):long
+getDurationUs(int,long):long
+getFirstSegmentNum():int
+getLastSegmentNum(long):int
+isExplicit():boolean

DashSingleSegmentIndex
-uri:RangedUri
+DashSingleSegmentIndex(RangedUri)
+getSegmentNum(long,long):int
+getTimeUs(int segmentNum):long
+getDurationUs(int,long):long
+getSegmentUrl(int):RangedUri
+getFirstSegmentNum();int
+getLastSegmentNum(long):int
+isExplicit():boolean

Period
+id:string
+startMs:long
+adaptationSets: List<AdaptationSet>
+Period(String,long,List<AdaptationSet>)
+getAdaptationSetIndex(int):int

SegmentBase
+initialization:RangedUri
+timescale:long
+presentationTimeOffset:long
+SegmentBase(RangedUri,long,long)
+getInitialization(Representation):RangedUri
+ getPresentationTimeOffsetUs():long

SegmentBase.SingleSegmentBase
+uri:String
+indexStart:long
+indexLength:long
+SingleSegmentBase(RangedUri,long,long,String,long,long)
+SingleSegmentBase(String)
+getIndex():RangedUri

SegmentBase.MultiSegmentBase
+startNumber:int
+duration:long
+segmentTimeline:List<SegmentTimelineElement>
+MultiSegmentBase(RangedUri,long,long,int,long,List<SegmentTimelineElement>)
+getSegmentNum(long,long):int
+getSegmentDurationUs(int,long):long
+getSegmentTimeUs(int):long
+getSegmentUrl(Representation,int):RangedUri
+getFirstSegmentNum():int
+getLastSegmentNum(long):int
+isExplicit():boolean

SegmentTimelineElement
+startTime:long
+duration:long
+SegmentTimelineElement(long,long)

<<接口>>
UriLoadable.Parser<T>
+parse(String,InputStream):T

SegmentBase.SegmentList
+mediaSegments:List<RangedUri>
+SegmentList(RangedUri,long,long,int,long,List<SegmentTimelineElement>,List<RangedUri>)
+getSegmentUrl(Representation,int):RangedUri
+getLastSegmentNum(long):int
+isExplicit():boolean

MediaPresentationDescriptionParser
-FRAME_RATE_PATTERN:Pattern
-contentId:String
-xmlParserFactory:XmlPullParserFactory
+MediaPresentationDescriptionParser()
+MediaPresentationDescriptionParser(String)
+parse(String connectionUrl, InputStream inputStream):MediaPresentationDescription
#parseMediaPresentationDescription(XmlPullParser,String):MediaPresentationDescription
#buildMediaPresentationDescription(long,long,long,boolean,long,long,UtcTimingElement,String,List<Period>):MediaPresentationDescription
#parseUtcTiming(XmlPullParser):UtcTimingElement
#buildUtcTimingElement(String,String):UtcTimingElement
#parsePeriod(XmlPullParser,String,long):Pair<Period,Long>
#buildPeriod(String,long,List<AdaptationSet>):Period
#parseAdaptationSet(XmlPullParser,String,SegmentBase):AdaptationSet
#buildAdaptationSet(int,int,List<Representation>,List<ContentProtection>):AdaptationSet
#parseContentType(XmlPullParser):int
#getContentType(Representation):int
#parseContentProtection(XmlPullParser):ContentProtection
#buildContentProtection(String,UUID,SchemeInitData):ContentProtection
#parseAdaptationSetChild(XmlPullParser):void
#parseRepresentation(XmlPullParser,String,String,String,int,int,float,int,int,String,SegmentBase,ContentProtectionsBuilder):Representation
#buildFormat(String,String,int,int,float,int,int,int,String,String):Format
#buildRepresentation(String,int,Format,SegmentBase):Representation
#parseSegmentBase(XmlPullParser,String,SingleSegmentBase):SingleSegmentBase
#buildSingleSegmentBase(RangedUri,long,long,String,long,long): SingleSegmentBase
#parseSegmentList(XmlPullParser,String,SegmentList):SegmentList
#buildSegmentList(RangedUri,long,long,int,long,List<SegmentTimelineElement>,List<RangedUri>):SegmentList
#parseSegmentTemplate(XmlPullParser,String,SegmentTemplate):SegmentTemplate
#buildSegmentTemplate(RangedUri,long,long,int,long,List<SegmentTimelineElement>,UrlTemplate,UrlTemplat,String):SegmentTemplate
#parseSegmentTimeline(XmlPullParser):List<SegmentTimelineElement>
#buildSegmentTimelineElement(long,long):SegmentTimelineElement
#parseUrlTemplate(XmlPullParser,String,UrlTemplate):UrlTemplate
#parseInitialization(XmlPullParser,String):RangedUri
#parseSegmentUrl(XmlPullParser,String):RangedUri
#parseRangedUrl(XmlPullParser,String,String,String):RangedUri
#buildRangedUri(String,String,long,long):RangedUri
#parseAudioChannelConfiguration(XmlPullParser):int
-checkLanguageConsistency(String,String):String
-checkContentTypeConsistency(int,int):int
#parseFrameRate(XmlPullParser,float):float
#parseDuration(XmlPullParser,String,long):long
#parseDateTime(XmlPullParser,String,long):long
#parseBaseUrl(XmlPullParser, String):String
#parseInt(XmlPullParser,String):int
parseInt(XmlPullParser,String,int):int
#parseLong(XmlPullParser,String):long
#parseLong(XmlPullParser,String,long):long
#parseString(XmlPullParser,String,String):String

MediaPresentationDescriptionParser.ContentProtectionsBuilder
-adaptationSetProtections:ArrayList<ContentProtection>
-representationProtections;ArrayList<ContentProtection>
-currentRepresentationProtections; ArrayList<ContentProtection>
-representationProtectionsSet:boolean
+addAdaptationSetProtection(ContentProtection):void
+addRepresentationProtection(ContentProtection):void
+endRepresentation():void
+build():ArrayList<ContentProtection>
-maybeAddContentProtection(List<ContentProtection>, ContentProtection):void
+compare(ContentProtection,ContentProtection):int

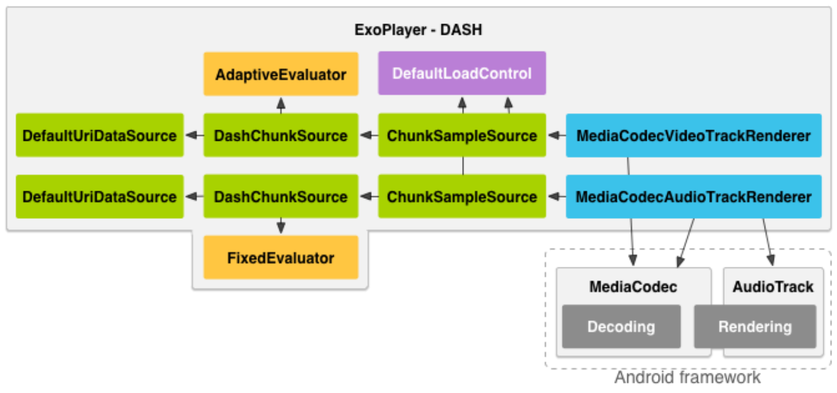
UrlTemplate
-REPRESENTATION:String="RepresentationID"
-NUMBER:String="Number"
-BANDWIDTH:String="Bandwidth"
-TIME:String="Time"
-ESCAPED_DOLLAR:String="$$"
-DEFAULT_FORMAT_TAG;String="%01d"
-REPRESENTATION_ID:int=1
-NUMBER_ID:int=2
-BANDWIDTH_ID:int=3
-TIME_ID:int=4;
-urlPieces:String[]
-identifiers:int[]
-identifierFormatTags:String[]
-identifierCount:int
+compile(template):UrlTemplate
-UrlTemplate(String[],int[],String[],int)
+buildUri(String,int,int,long):String
-parseTemplate(String,String[],int[],String[]):int

Segment.SegmentTemplate
+initializationTemplate:UrlTemplate
+mediaTemplate:UrlTemplate
-baseUrl:String
+SegmentTemplate(RangedUri,long,long,int,long,List<SegmentTimelineElement>,UrlTemplate,UrlTemplate, String)
+getInitialization(Representation):RangedUri
+getSegmentUrl(Representation,int):RangedUri
+getLastSegmentNum(long):int

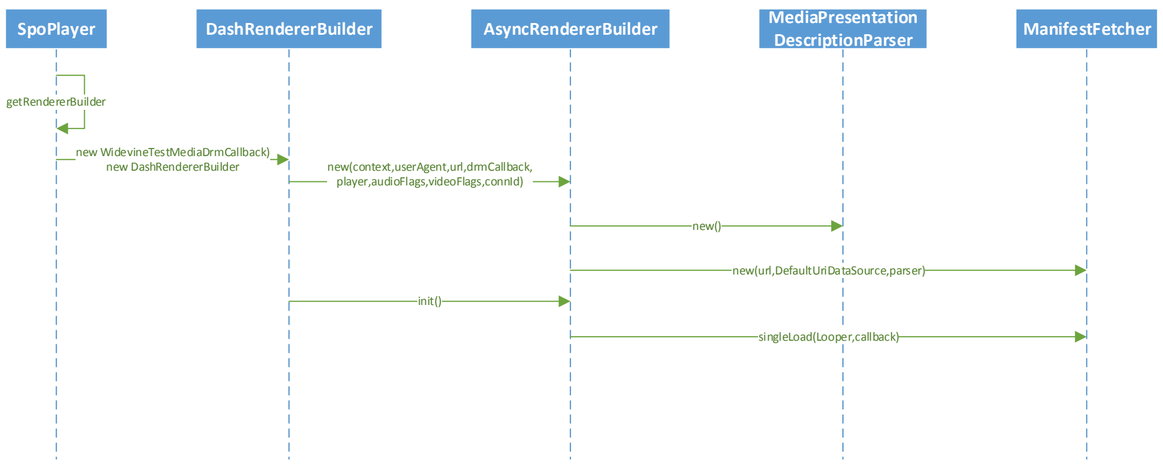
+format
-initializationUri
+representations
+contentProtections
-segmentIndex
+periods
+initialization
+adaptationSets
+mediaSegments
+segmentTimeline
+initializationTemplate
+mediaTemplate

Dash Implementation（Exo） Investigation
—— DASH Model

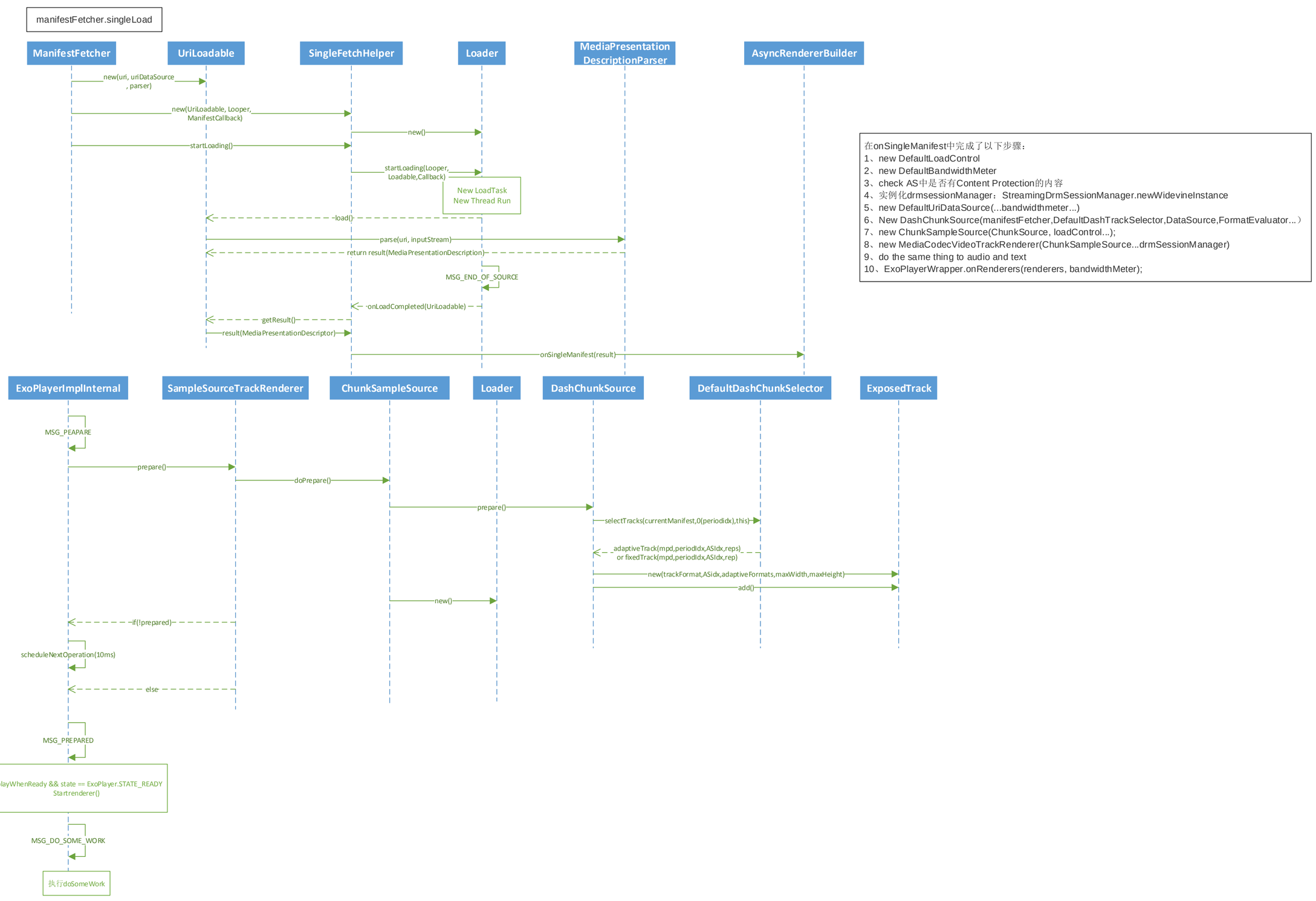
Dash Implementation（Exo） Investigation
—— DASH Playback Flow



Demo创建Player的过程

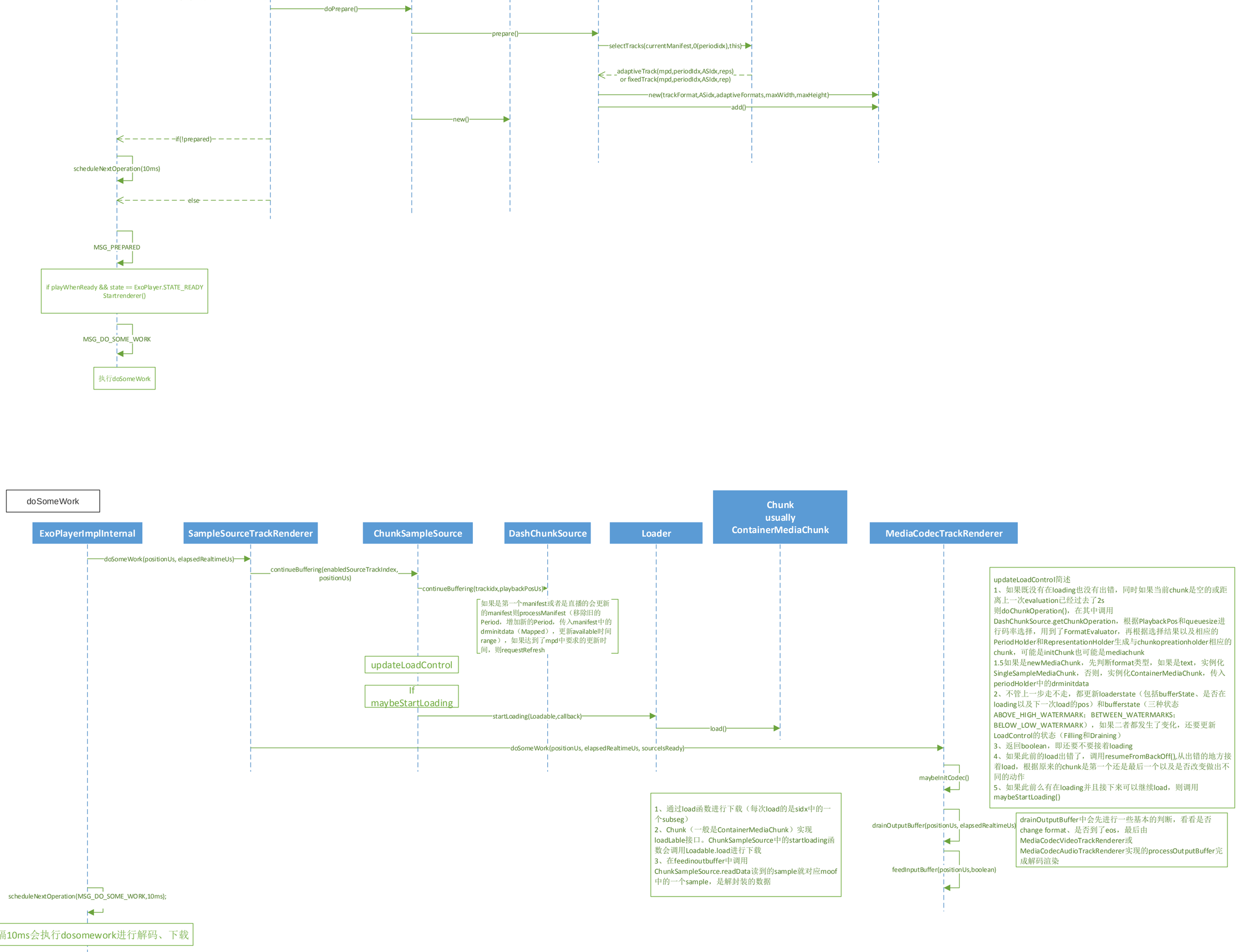


manifestFetcher.singleLoad



在onSingleManifest中完成了以下步骤：
1、new DefaultLoadControl
2、new DefaultBandwidthMeter
3、check AS中是否有Content Protection的内容
4、实例化drmsessionManager：StreamingDrmSessionManager.newWidevineInstance
5、new DefaultUriDataSource(...bandwidthmeter...)
6、New DashChunkSource(manifestFetcher,DefaultDashTrackSelector,DataSource,FormatEvaluator...）
7、new ChunkSampleSource(ChunkSource, loadControl...);
8、new MediaCodecVideoTrackRenderer(ChunkSampleSource...drmSessionManager)
9、do the same thing to audio and text
10、ExoPlayerWrapper.onRenderers(renderers, bandwidthMeter);

doSomeWork



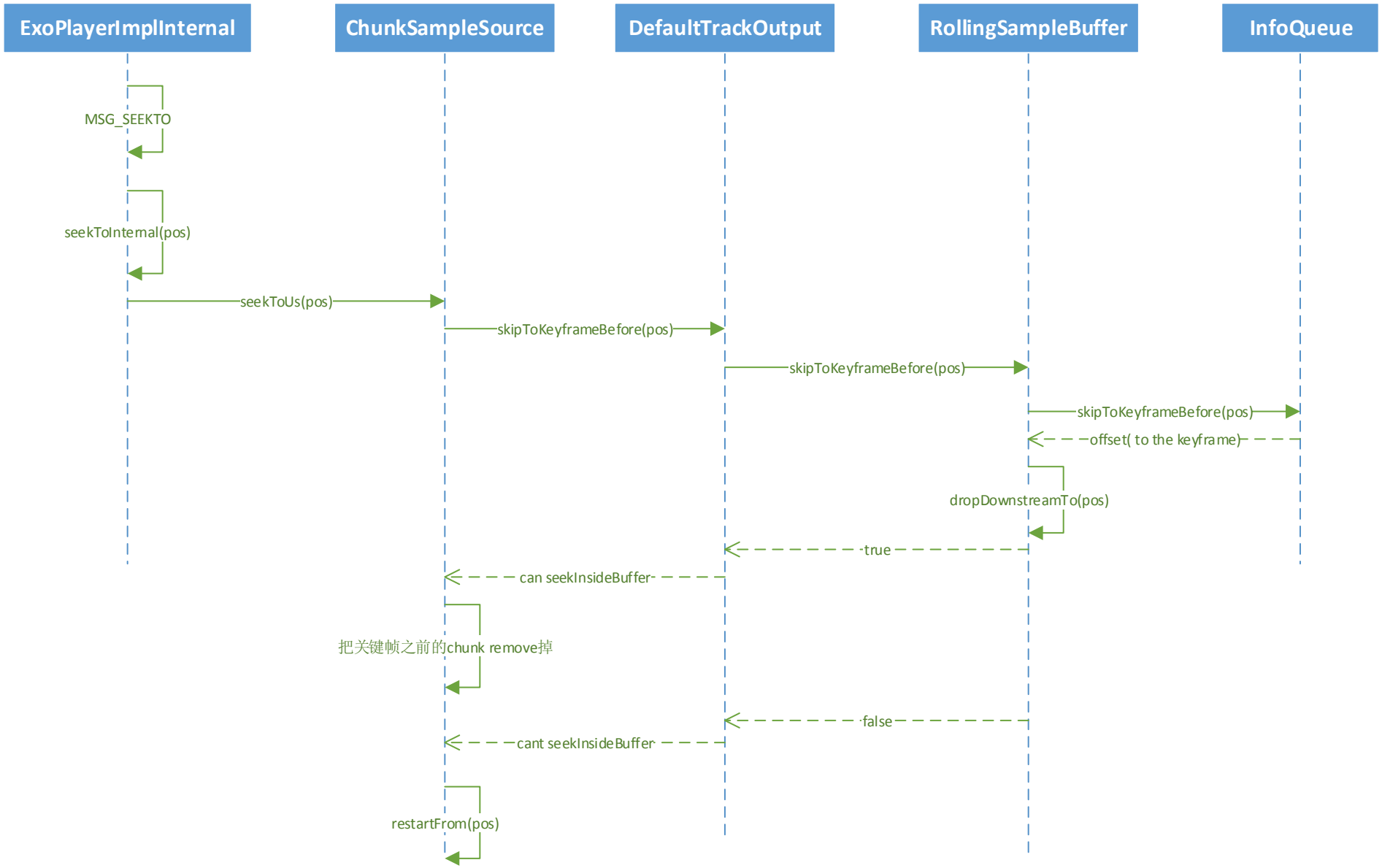
updateLoadControl简述
1、如果既没有在loading也没有出错，同时如果当前chunk是空的或距离上一次evaluation已经过去了2s
则doChunkOperation()，在其中调用DashChunkSource.getChunkOperation，根据PlaybackPos和queuesize进行码率选择，用到了FormatEvaluator，再根据选择结果以及相应的PeriodHolder和RepresentationHolder生成与chunkopreationholder相应的chunk，可能是initChunk也可能是mediachunk
1.5如果是newMediaChunk，先判断format类型，如果是text，实例化SingleSampleMediaChunk，否则，实例化ContainerMediaChunk，传入periodHolder中的drminitdata
2、不管上一步走不走，都更新loaderstate（包括bufferState、是否在loading以及下一次load的pos）和bufferstate（三种状态ABOVE_HIGH_WATERMARK；BETWEEN_WATERMARKS；BELOW_LOW_WATERMARK），如果二者都发生了变化，还要更新LoadControl的状态（Filling和Draining）
3、返回boolean，即还要不要接着loading
4、如果此前的load出错了，调用resumeFromBackOff(),从出错的地方接着load，根据原来的chunk是第一个还是最后一个以及是否改变做出不同的动作
5、如果此前么有在loading并且接下来可以继续load，则调用maybeStartLoading()

1、通过load函数进行下载（每次load的是sidx中的一个subseg）
2、Chunk（一般是ContainerMediaChunk）实现loadLable接口。ChunkSampleSource中的startloading函数会调用Loadable.load进行下载
3、在feedinoutbuffer中调用ChunkSampleSource.readData读到的sample就对应moof中的一个sample，是解封装的数据

drainOutputBuffer中会先进行一些基本的判断，看看是否change format、是否到了eos，最后由MediaCodecVideoTrackRenderer或MediaCodecAudioTrackRenderer实现的processOutputBuffer完成解码渲染

Dash Implementation（Exo） Investigation
—— DASH Seek Flow

Seek流程简述：假设需要seek到pos位置，则先找到在pos之前且距pos最近的keyframe位置，设为posA，如果posA在buffer中，直接切换过去，如果不在，则从posA位置重新开始一个playback流程



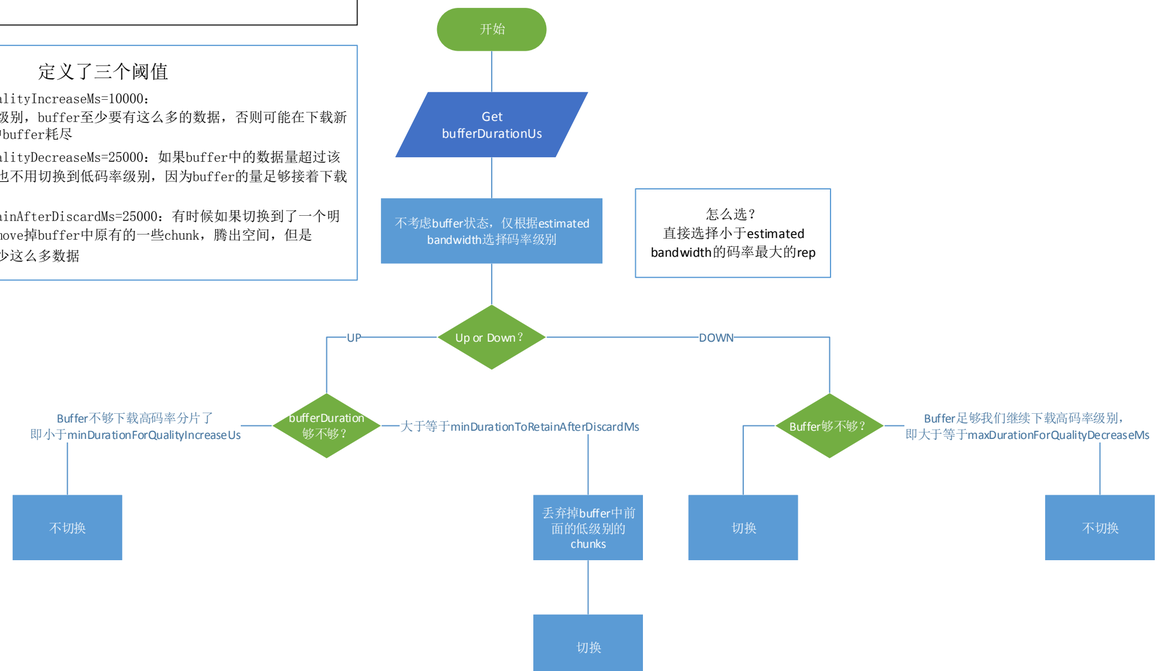
Dash Implementation（Exo） Investigation
—— Default Adaptive Algorithm

定义了三个阈值

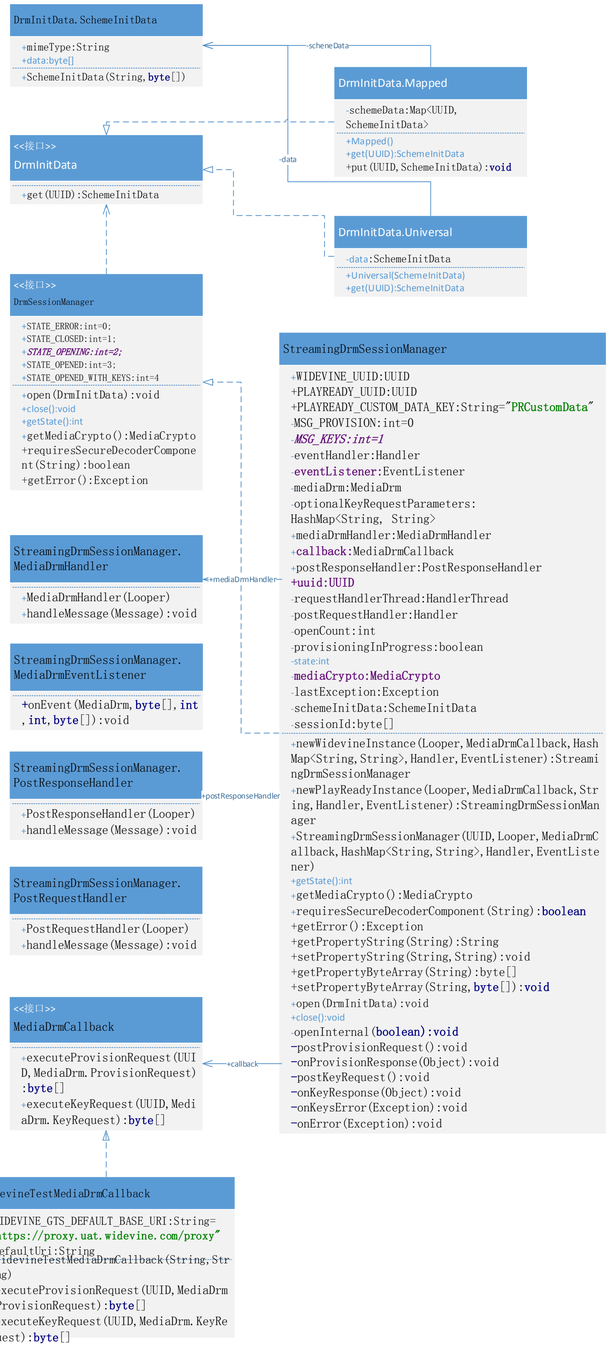
1、minDurationForQualityIncreaseMs=10000：
要想切换更高的format级别，buffer至少要有这么多的数据，否则可能在下载新的更高级别rep的过程中buffer耗尽

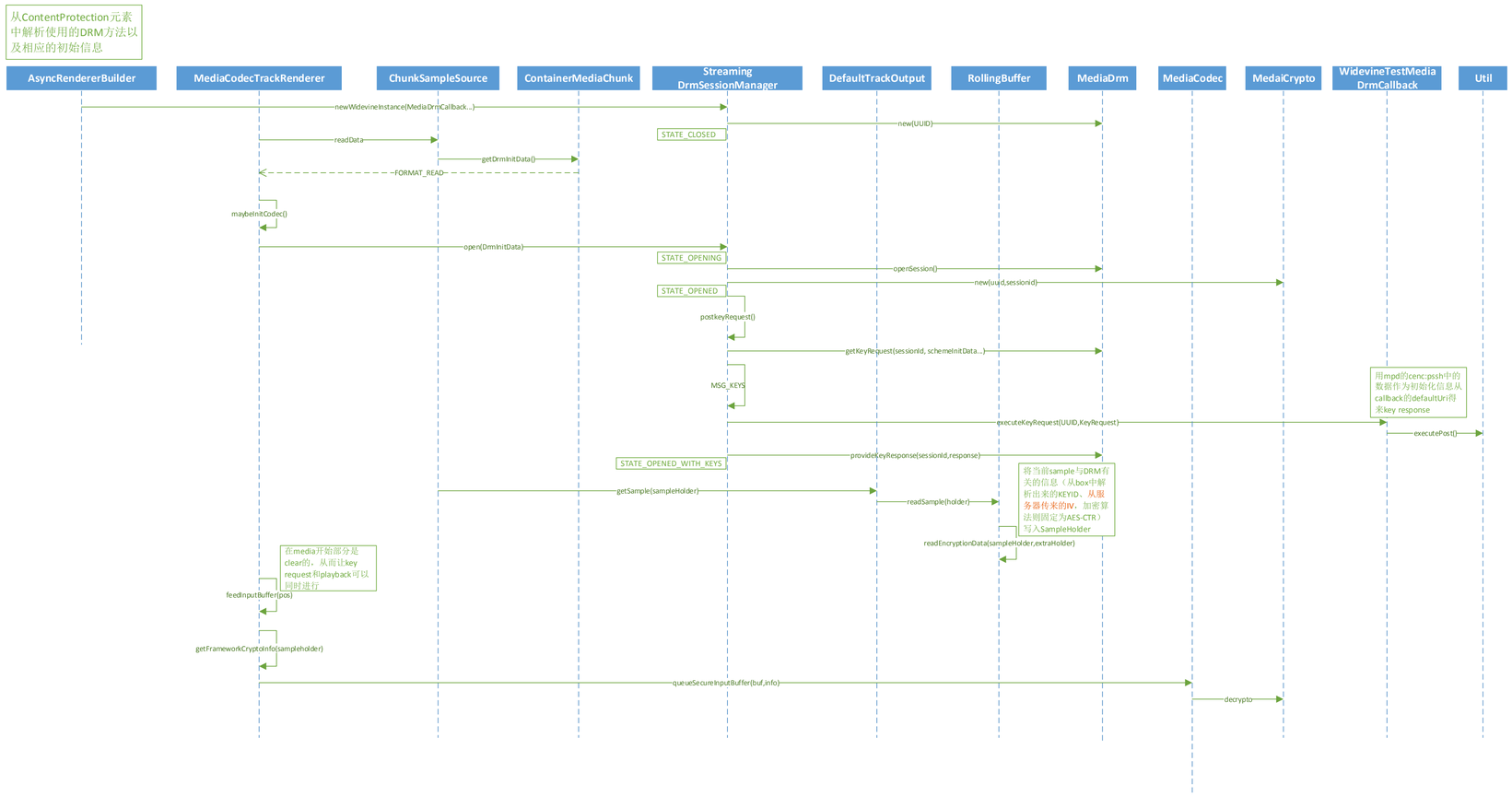
2、maxDurationForQualityDecreaseMs=25000：如果buffer中的数据量超过该值，就算带宽降低了，也不用切换到低码率级别，因为buffer的量足够接着下载高码率级别

3、minDurationToRetainAfterDiscardMs=25000：有时候如果切换到了一个明显更高的级别，需要remove掉buffer中原有的一些chunk，腾出空间，但是buffer中要始终保持至少这么多数据



Dash Implementation（ Exo ） Investigation
—— DRM Related Things





Dash Implementation (Exo) Investigation
—— DASH Extractor and Render