|  |
| --- |
|  |
| : |
|  |
| Author:  Date: **17 August 2017**  Version:  Status: |

The Solution Design is produced during the Definition phase of a project.

The inputs required for *starting* the Solution Design are:

| **Document** | **Responsibility** | **Received?** |
| --- | --- | --- |
| Architecture Options and Recommendations | Architecture Team | Yes |
| Start up Response | SI | **No** |
| High level Architecture | Architecture Team | Yes |
| High Level Requirements | BA Team | Yes |
| Draft Detailed Requirements, including Use Cases, Business Processes and Non-functional Requirements | BA Team | Yes |

The inputs required before *completing* the solution design are:

| **Document** | **Responsibility** | **Received?** |
| --- | --- | --- |
| Signed Off Detailed Requirements document, including Use Cases, Business processes and non-functional requirements | BA Team | **No** |
| Requirements Traceability Matrix | BA Team | **No** |
| Infrastructure High Level Design | Infrastructure Solutions Design | **Yes** |

The document is written by the E2E Solution Designer with input from other impacted specialist Solution Designers, development areas, information security, operations and service management and in consultation with architecture, the business analyst and project manager.

The Solution Design is required for all projects that are delivering changes requiring IT development. The purpose of the document is to describe the design of the solution being delivered in sufficient detail that the developers generating the detailed technical designs need no other starting point to understand what their detailed designs must do.

| **OWNERSHIP & CONFIDENTIALITY** |
| --- |
| The information contained in this communication belongs to, or is licensed to, Virgin Media. The information is confidential and is intended only for use by Virgin Media and authorised third parties, to whom it is provided by Virgin Media. The information is intended for use as part on a specific project. Unauthorised use, disclosure of, or copying of this information, in whole or in part, is strictly prohibited. This information shall be returned to Virgin Media promptly upon request. Virgin Media makes no warranties as to its accuracy or completeness of this information and the recipient shall not rely on it for any reason. Readers should ensure that they have an up to date version of this information by referring to prior to use. Virgin Media reserves all its rights. |

# Table of Contents

[1 Table of Contents 2](#_Toc490728921)

[1.1 Table of figures 6](#_Toc490728922)

[2 Document Control 7](#_Toc490728923)

[2.1 Document Information 7](#_Toc490728924)

[2.2 Document Release Information 7](#_Toc490728925)

[2.3 Preliminary Review 8](#_Toc490728926)

[2.4 For Review/Impact Assessment By 8](#_Toc490728927)

[2.5 Distribution 9](#_Toc490728928)

[2.6 References 9](#_Toc490728929)

[2.7 Contributors to this document 10](#_Toc490728930)

[2.8 Glossary 10](#_Toc490728931)

[3 Introduction 12](#_Toc490728932)

[3.1 Project Overview 12](#_Toc490728933)

[3.1.1 Imperative 12](#_Toc490728934)

[3.1.2 Future State Vision 12](#_Toc490728935)

[3.1.3 Approach 12](#_Toc490728936)

[3.2 Scope 12](#_Toc490728937)

[3.2.1 Day 1 Scope 12](#_Toc490728938)

[3.2.2 Day 2 Scope 12](#_Toc490728939)

[3.2.3 Out of Scope 12](#_Toc490728940)

[3.2.4 Additional CRs 13](#_Toc490728941)

[4 Raid Log 14](#_Toc490728942)

[4.1 Risks 14](#_Toc490728943)

[4.2 Assumptions 14](#_Toc490728944)

[4.3 Issues 14](#_Toc490728945)

[4.4 Dependencies 14](#_Toc490728946)

[4.5 Questions to be addressed 15](#_Toc490728947)

[5 Solution Design 16](#_Toc490728948)

[5.1 Solution Overview 16](#_Toc490728949)

[5.1.1 Manifest Manipulation 17](#_Toc490728950)

[5.1.2 Playout 17](#_Toc490728951)

[5.1.2.1 Pause and Restart 17](#_Toc490728952)

[5.1.2.2 User Error Reporting 18](#_Toc490728953)

[5.2 Solution Design Diagram 19](#_Toc490728954)

[5.3 Solution Elements 20](#_Toc490728955)

[5.4 Solution Physical Architecture 22](#_Toc490728956)

[5.5 Security 22](#_Toc490728957)

[5.5.1 Application 22](#_Toc490728958)

[5.5.2 Network 22](#_Toc490728959)

[5.5.3 Solution Deficiencies 22](#_Toc490728960)

[6 Solution Design Detail 23](#_Toc490728961)

[6.1 Detailed Design Documents 23](#_Toc490728962)

[6.2 Interfaces 24](#_Toc490728963)

[6.3 Content Management 28](#_Toc490728964)

[6.3.1 Content Delivery 28](#_Toc490728965)

[6.3.2 Advert signalling overview 28](#_Toc490728966)

[6.3.3 Packaging Content for Advert Insertion. 29](#_Toc490728967)

[6.3.4 SCTE35 Processing in VSPP 29](#_Toc490728968)

[6.3.5 Asset Management & Metadata 29](#_Toc490728969)

[6.3.5.1 Assets management 29](#_Toc490728970)

[6.3.5.2 Presentation on STB 30](#_Toc490728971)

[6.3.5.3 Interior Break Data 30](#_Toc490728972)

[6.3.5.4 Encoding type table 31](#_Toc490728973)

[6.3.5.5 Asset Identification 31](#_Toc490728974)

[6.3.6 Matching Ad Content to Main Content 31](#_Toc490728975)

[6.4 Content Processing in VSPP 33](#_Toc490728976)

[6.4.1 Flow 33](#_Toc490728977)

[6.4.2 Sequence 34](#_Toc490728978)

[6.4.3 SeaChange to VSPP Messaging 34](#_Toc490728979)

[6.4.3.1 Ingest Notification 34](#_Toc490728980)

[6.4.3.2 Status Poll 35](#_Toc490728981)

[6.4.3.3 Essences 36](#_Toc490728982)

[6.4.3.4 Update Notification 36](#_Toc490728983)

[6.4.3.5 Delete Notification 36](#_Toc490728984)

[6.5 Metadata Delivery to Cadent 37](#_Toc490728985)

[6.5.1 Sequence 38](#_Toc490728986)

[6.5.1.1 SeaChange Changes 39](#_Toc490728987)

[6.5.1.2 Processing and Filtering 39](#_Toc490728988)

[6.5.1.3 Updates 39](#_Toc490728989)

[6.5.1.4 Deletions 40](#_Toc490728990)

[6.5.1.5 Sizing 40](#_Toc490728991)

[6.5.2 Interface File Format 40](#_Toc490728992)

[6.5.2.1 Structure 40](#_Toc490728993)

[6.5.2.2 Key Data 41](#_Toc490728994)

[6.5.2.3 Show and Series 48](#_Toc490728995)

[6.5.2.4 Example 49](#_Toc490728996)

[6.5.3 URL Generation 53](#_Toc490728997)

[6.6 Campaign Management & Tracking 55](#_Toc490728998)

[6.6.1 BlackArrow Campaign Manager 55](#_Toc490728999)

[6.6.2 Tracking (day1) 56](#_Toc490729000)

[6.6.2.1 Ad Placement Tracking ID 56](#_Toc490729001)

[6.6.3 Tracking 56](#_Toc490729002)

[6.6.3.1 Full Quartile Tracking 56](#_Toc490729003)

[6.6.3.2 Trick-Play suppression 56](#_Toc490729004)

[6.6.4 Client Applications 57](#_Toc490729005)

[6.6.5 Placement Tracking URLs flow 57](#_Toc490729006)

[6.6.6 Placement Status Notice (PSN) 57](#_Toc490729007)

[6.6.6.1 Ad IDs in the MPEG-DASH Stream 57](#_Toc490729008)

[6.6.6.2 Sequence Diagram 59](#_Toc490729009)

[6.6.6.3 Example Messages 59](#_Toc490729010)

[6.6.6.3.1 Measurement Notification 59](#_Toc490729011)

[6.6.6.3.2 Placement Status Acknowledgement 62](#_Toc490729012)

[6.6.6.4 Error Handling and Retry 62](#_Toc490729013)

[6.6.6.5 Monitoring 62](#_Toc490729014)

[6.6.6.6 Controlling Parameters 63](#_Toc490729015)

[6.7 Real Time Decisioning 64](#_Toc490729016)

[6.7.1 Cadent Decisioning 64](#_Toc490729017)

[6.7.2 Fallback Decisioning 64](#_Toc490729018)

[6.7.3 Ad Availability 64](#_Toc490729019)

[6.7.4 Ad Decisioning Request – MDC to Cadent 65](#_Toc490729020)

[6.7.4.1 Request Format 65](#_Toc490729021)

[6.7.4.2 SCTE 130-3 Request Body 65](#_Toc490729022)

[6.7.5 Ad Decisioning Response – Cadent to MDC 66](#_Toc490729023)

[6.7.5.1 Response Structure 66](#_Toc490729024)

[6.7.6 Timeout and Retry 71](#_Toc490729025)

[6.7.7 Decisioning Failure 72](#_Toc490729026)

[6.8 Manifest Manipulation 73](#_Toc490729027)

[6.8.1 Manifest manipulator components 73](#_Toc490729028)

[6.8.1.1 Master Central Manager System (CMS) (Arris supplied) 73](#_Toc490729029)

[6.8.1.2 Spectrum MDC Satellites (Arris supplied) 73](#_Toc490729030)

[6.8.1.3 Front-End Load Balancer 73](#_Toc490729031)

[6.8.2 Manifest requesting flow 73](#_Toc490729032)

[6.8.2.1 DASH/TiVo IP VOD 74](#_Toc490729033)

[6.8.3 STB Playout Flow 75](#_Toc490729034)

[6.8.3.1 Leapfrog Setup Flow 76](#_Toc490729035)

[6.8.3.2 Compass VoD Setup Flow 79](#_Toc490729036)

[6.8.4 Playout Flow URLs 81](#_Toc490729037)

[6.8.5 Manifest Structure 86](#_Toc490729038)

[6.8.6 Manifest Requesting flow - Security 87](#_Toc490729039)

[6.8.6.1 Akamai Token & Cookie, & Amendment to URL 87](#_Toc490729040)

[6.8.7 Akamai NetStorage Instances 88](#_Toc490729041)

[6.8.7.1 Akamai Balancing 89](#_Toc490729042)

[6.8.8 Customer MAC Cookie within TLS 90](#_Toc490729043)

[6.9 Adaptive Streaming 91](#_Toc490729044)

[6.9.1 MPEG-DASH 91](#_Toc490729045)

[6.9.2 Ad insertion in MPEG-DASH 91](#_Toc490729046)

[6.9.3 Difference Between Live and VoD Streams in MPEG-DASH 92](#_Toc490729047)

[6.9.4 Server Side Insertion Mode – Virgin Media’s Chosen Method 92](#_Toc490729048)

[6.10 Network & Server infrastructure 93](#_Toc490729049)

[6.10.1 MDC Cluster components 93](#_Toc490729050)

[6.10.1.1 MDC Cluster Component Scaling Considerations 93](#_Toc490729051)

[6.10.2 Blackarrow & SeaChange components 93](#_Toc490729052)

[6.11 Platform Monitoring 96](#_Toc490729053)

[6.11.1 Seachange Platforms 96](#_Toc490729054)

[6.11.2 Ericsson VSPP 96](#_Toc490729055)

[6.11.3 Cadent Platforms 96](#_Toc490729056)

[6.11.4 Arris MDC 96](#_Toc490729057)

[6.11.5 CDN 96](#_Toc490729058)

[6.11.6 Varnish Cache and Load Balancer 96](#_Toc490729059)

[6.11.7 F5 Load Balancers 96](#_Toc490729060)

[7 Non Functional Aspects 97](#_Toc490729061)

[7.1 Volumetrics & Capacity 97](#_Toc490729062)

[7.1.1 Non Functional Requirements 97](#_Toc490729063)

[7.1.2 Data Volumes for Playout 97](#_Toc490729064)

[7.1.2.1 Calculation Parameters 98](#_Toc490729065)

[7.1.2.2 CDN Calculations 98](#_Toc490729066)

[7.1.2.3 Varnish Shield Cache Calculations 99](#_Toc490729067)

[7.1.2.4 VSPP Calculations 99](#_Toc490729068)

[7.1.2.5 MM Calculations 99](#_Toc490729069)

[7.1.2.6 Cadent Calculations 100](#_Toc490729070)

[8 Security Considerations 101](#_Toc490729071)

[8.1.1 Hardware Security 101](#_Toc490729072)

[8.1.2 Application Security 101](#_Toc490729073)

[8.1.3 Network Security 101](#_Toc490729074)

[8.2 Reporting 101](#_Toc490729075)

[8.2.1 Business reporting 101](#_Toc490729076)

[8.2.2 Application reporting 101](#_Toc490729077)

[8.2.3 Error logging 102](#_Toc490729078)

[8.3 Operational Continuity 102](#_Toc490729079)

[8.3.1 Redundancy of MDC/CMS 102](#_Toc490729080)

[8.4 Disaster Recovery 102](#_Toc490729081)

[8.5 Support Model 102](#_Toc490729082)

[8.6 Licensing 102](#_Toc490729083)

[8.7 Operating System / Software Versions 102](#_Toc490729084)

[8.8 Requirements Traceability Audit 103](#_Toc490729085)

[8.9 Impact Assessments 103](#_Toc490729086)

[8.10 OSS Development 103](#_Toc490729087)

[8.11 TV Engineering 103](#_Toc490729088)

[8.12 TiVo and its clients 103](#_Toc490729089)

[8.13 Web in-house development team 103](#_Toc490729090)

[8.14 Network Engineering / Design 103](#_Toc490729091)

[8.15 Infrastructure Design 103](#_Toc490729092)

[8.16 Virgin Media Engineering Security 103](#_Toc490729093)

[9 Appendix A 104](#_Toc490729094)

[9.1 Manifest Manipulation Constraints 104](#_Toc490729095)

[10 Appendix B 105](#_Toc490729096)

[11 Appendix C – MPEG DASH Manifest Example 106](#_Toc490729097)

[12 Appendix D – SCTE 130 Request and Response 114](#_Toc490729098)

[12.1 Request 114](#_Toc490729099)

[12.2 Response 114](#_Toc490729100)

[13 Appendix E – Error Codes for STB 124](#_Toc490729101)

[13.1 S1 – STB Player to Traxis 124](#_Toc490729102)

[13.2 P1, P2 – STB Player to MM via CDN 125](#_Toc490729103)

[13.3 P3 – MM to Varnish/VSPP 125](#_Toc490729104)

[13.4 A4 – MM to Cadent 126](#_Toc490729105)

[13.5 xx – STB to Licence Server 127](#_Toc490729106)

[13.6 C7, C8 – STB to CDN/VSPP 127](#_Toc490729107)

## Table of figures

[Figure 5‑1 - Architecture Overview 16](#_Toc490729108)

[Figure 5‑2 - Solution Diagram overview 19](#_Toc490729109)

[Figure 6‑1 - Interface Diagram 24](#_Toc490729110)

[Figure 6‑2- Chunking the segments 29](#_Toc490729111)

[Figure 6‑3 - Deployment Architecture for VOD QAM 55](#_Toc490729112)

[Figure 6‑4 - Token flow 88](#_Toc490729113)

[Figure 6‑5 - DASH structure 91](#_Toc490729114)

[Figure 6‑6 - MPEG-DASH MPD file 91](#_Toc490729115)

[Figure 6‑7 - MPD manipulation 92](#_Toc490729116)

[Figure 6‑8 - MDC connections 94](#_Toc490729117)

[Figure 6‑9 - Arris, BA & SeaChange connections 95](#_Toc490729118)

# Document Control

## Document Information

| **Title** |  |
| --- | --- |
| **Type** |  |
| **Author / (Last saved by)** | / () |
| **Status (Version)** | ( ) |

## Document Release Information

| **Version** | **Description / Change Comments** | **Who** | **Date** |
| --- | --- | --- | --- |
| 0.25 | *Initial draft* | Keith Wigmore | 02/09/15 |
| 0.26 | *Streaming, housing Input from TV Engineering – Tushar Patel* | Keith Wigmore/Tushar Patel | 10/09/15 |
| 0.29 | *Update after Arris, SeaChange & BlackArrow review* | Keith Wigmore | 21/10/2015 |
| 1 | *Officaly releas* | Keith Wigmore | 23/10/2015 |
| 1.1 | *Update from Arris – mirror fixes* | Keith Wigmore | 22/01/2016 |
| 01\_02 | *Remove references to linear IP and TVA (this is now TiVo STB only).* | Geoff Routledge | 22/03/2017 |
| 01\_03 | *New version to avoid confusion whilst previous is out for preliminary review. Add SCTE 130 PSN details.* | Geoff Routledge | 28/03/2017 |
| 01\_04 | *Interim version, primarily of interest to TiVo. Has updates to sections describing the DASH manifest and the way in which tracking information is encoded in it. Also has changes to the SCTE 130-3 exchange between MM and Cadent.* | Geoff Routledge | 25/04/2017 |
| 01\_05 | *More detail around SeaChange to VSPP content processing.* | Geoff Routledge | 25/04/2017 |
| 01\_06 | *Passing of reporting URL in manifest from MM. Compass and Leapfrog differences in playback. SeaChange to Cadent interface changes.* | Geoff Routledge | 27/04/2017 |
| 01\_07 | *Changes resulting from workshop session.* | Geoff Routledge | 03/05/2017 |
| 01\_08 | *Further incremental changes.Consolidation of approach to SD/HD matching – use MM.* | Geoff Routledge | 10/05/2017 |
| 01\_09 | *Updates as a result of review comments.* | Geoff Routledge | 17/05/2017 |
| 01\_10 | *Expansion of errors appendix. Minor changes to scope statement. Changes arising from review comments.* | Geoff Routledge | 25/05/2017 |
| 02\_00 | *Changes to NFRs. Issue. Include SeaChange comments that were missed.* | Geoff Routledge | 14/06/2017 |
| 02\_01 | *Correction on page 27 – table of interfaces – S2: this should be HTTPS. Updated deployment architecture diagram for Cadent with Cadent’s new version.*  *Additional detail and corrections in description of TVA Transformer file format as a result of review comments from SeaChange.* | Geoff Routledge | 28/06/2017  08/08/2017 |
| 02\_02 | *Add appendix detailing changes to move the VSPP solution to use the Velocix CDN. Also show the co-existence of a Harmonic/Akamai solution and explain how these run side-by-side. Ultimately, this information will be folded back into the main document but for now we will treat this as an appendix and tech note.* | Geoff Routledge | 17/08/2017 |

## Preliminary Review

| **Name** | **Title / Department** | **Approval (Y/N)** |
| --- | --- | --- |
| Ben Hanson | Head of Product Solution Design |  |
| Tushar Patel | Principal Engineer, Content Protection & Distribution |  |
| Jon Haddock | Project Manager |  |
| David Holroyd | Solution/Systems Designer |  |
| Blackarrow | Vendor - ADS |  |
| Arris | Vendor - ADM |  |

## For Review/Impact Assessment By

|  |  | **Approval (Y/N)** | **Impact Assessment**  **(Y/N)** |
| --- | --- | --- | --- |
| Ben Hanson | Head of Product Solution Design |  |  |
| Richard Wood | Head of Entertainment Architecture |  |  |
| Darron Christopher | Head of Data Engineering |  |  |
| Mike Denney | Head of TV Engineering |  |  |
| David Holroyd | Local Integration Architect |  |  |
| David Eatwell | Service Transition Manager |  |  |
| Dylan White | Security Engineering |  |  |
| Tushar Patel | Principal Engineer, Content Protection & Distribution |  |  |
| Mark Ferres | VMTS |  |  |
| Mike Denney | Head of TiVo and Interactive Engineering |  |  |
| Mike Fellows | Network Service Optimisation |  |  |
| Neil Patterson | Lead Business Analyst |  |  |
| Noel Priddey | Senior Project Manager |  |  |
| Mark Vandersluis | Solutions Engagement and Design Manager |  |  |
| Keith Wigmore | Lead Architecture |  |  |
| Dale Woodcock | Principal Engineer, Content Ingest & Management |  |  |
| Stephen Hagerty | Senir Project Manager, Network Delivery |  |  |

Where a reviewer does not approve the document, the written list of changes required must be supplied by that reviewer.

See **Impact Assessments** section below for further details on expected IA deliverables from the Departments/Teams highlighted above

## Distribution

Distribution is to the reviewer list above, plus to the following: (*this is the right list*?)

| **Name** | **Title / Department** |
| --- | --- |
| Adrian Hayes | Business Improvement Manager, Strategy & Development |
| Amery Saint | Multiscreen Applications Support Manager, Middleware & Portals |
| Andrew Wickens | Manager, Problem Management, Incident and Problem Mgmt |
| Darran Tibballs | Enterprise Monitoring & Automation Support Manager, OSS |
| Dean Ralston | CNMC Team Leader, Customer Network Management Centre |
| Dee Samra | Delivery Manager, Enterprise Data Design & Manage |
| Dom DiMambro | OSS Platform Support Manager, OSS |
| Graham Clare | Consumer Systems Support Manager, BSS |
| James Liddiard | Supplier Manager, Vendor Delivery |
| John Craft | OSS Platform Development Manager, OSS |
| Jolene Copeland | Manager, Incident Management, Incident and Problem Mgmt |
| Jorge Martins | Problem Manager, Incident and Problem Mgmt |
| Kevin Crawford | Change Manager, Change |
| Laura Sheriff | Consumer Business Liaison Manager, Service Governance |
| Lesley Waghorn | Data Protection & Privacy Manager, Privacy, IT & Ops Risk |
| Lynne Ewington | Team Leader, BSS |
| Mark Jackman | Multiscreen Software Development Manager, Middleware & Portals |
| Mark Rotheram | Technical Production Control Manager, IT Operations |
| Mark Titus | Data Delivery Manager, IT Technical Support |
| Martin Harper | Network Specialist (Delivery), IT Technical Support |
| Michael Daley | 2nd Line Support Engineer, Network Support |
| Neil Letchford | Online Development Manager, Middleware & Portals |
| Oliver Stanley | Senior Developer, OSS |
| Pardip Sahota | Security Operations Analyst, CIO Office |
| Paul Baker | Configuration Analyst, Operational Governance |
| Rob Pinniger | Senior Manager, Content Security, TV Content Acquisition |
| Stephen Oakham | Service Partner, Service Governance |
| Tracy Redfern | Broadcast Operations Manager, TV NMC |
| Vawns Murphy | Change Specialist, Change |

## References

| **Reference** | **Document** | **Link / Path** |
| --- | --- | --- |
| Ref01 | 1008264: STB VOD Re-launch - SDD | [SharePoint link](http://sharepoint/sites/change/progs/1008264/Workstreams/Architecture%20and%20Solution%20Design/Virgin%20Media/Solution%20Design/STB%20VOD%20Relaunch%20SDD%20v1.1%20Final.docx) |
| Ref02 | STCE35 standard – “Digital Program Insertion Cueing Message for Cable” - ANSI/SCTE 35 2013 | [SharePoint link](https://www.scte.org/documents/pdf/Standards/Top%20Ten/ANSI_SCTE%2035%202013.pdf) |
| Ref03 | [VDE-102 VTVA 2 Simulcast and Playready HLA v1\_0](http://sharepoint/sites/support/Technology/techarch/BoDTVA/Shared%20Documents/Video%20and%20Entertainment%20Architecture/VDE-102%20VTVA%202%20Simulcast%20and%20Playready%20HLA%20v1_0.pdf) | [SharePoint link](http://sharepoint/sites/support/Technology/techarch/BoDTVA/Shared%20Documents/Video%20and%20Entertainment%20Architecture/VDE-102%20VTVA%202%20Simulcast%20and%20Playready%20HLA%20v1_0.pdf) |
| Ref04 | [VDE-051 HLA\_TV\_Simulcast\_v1.42](http://sharepoint/sites/support/Technology/techarch/BoDTVA/Shared%20Documents/Video%20and%20Entertainment%20Architecture/VDE-051%20HLA_TV_Simulcast_v1.42.pdf) | [SharePoint link](http://sharepoint/sites/support/Technology/techarch/BoDTVA/Shared%20Documents/Video%20and%20Entertainment%20Architecture/VDE-051%20HLA_TV_Simulcast_v1.42.pdf) |
| Ref05 | 2.18 MDC 2.7 HA User Guide.pdf – Arris Spectrum MDC | [SharePoint link](http://sharepoint/sites/change/progs/1008264/Workstreams/Forms/AllItems.aspx?RootFolder=%2fsites%2fchange%2fprogs%2f1008264%2fWorkstreams%2fAdvertising%2f03%20Technical%20Documents&FolderCTID=&View=%7bF0AD2FC6%2dB9F7%2d4F86%2d9B27%2d1E0BB56E06AE%7d) |
| Ref06 |  |  |
| Ref07 | VoD Advert Replacement - VoD Advert Replacement  High Level Architecture (HLA\_BlackArrow\_1\_0ca.doc) | [SharePoint link](http://sharepoint/sites/change/progs/1008264/Workstreams/Forms/AllItems.aspx?RootFolder=%2fsites%2fchange%2fprogs%2f1008264%2fWorkstreams%2fAdvertising%2f03%20Technical%20Documents&FolderCTID=&View=%7bF0AD2FC6%2dB9F7%2d4F86%2d9B27%2d1E0BB56E06AE%7d) |
| Ref08 | Virgin Media message flows\_lmc20150421.pdf | [SharePoint link](http://sharepoint/sites/change/progs/1008264/Workstreams/Forms/AllItems.aspx?RootFolder=%2fsites%2fchange%2fprogs%2f1008264%2fWorkstreams%2fAdvertising%2f03%20Technical%20Documents&FolderCTID=&View=%7bF0AD2FC6%2dB9F7%2d4F86%2d9B27%2d1E0BB56E06AE%7d) |
| Ref09 | Digital Video Ad Serving Template (VAST) 3.0 | http://www.iab.net/media/file/VASTv3.0.pdf |
| Ref10 | Digital Video Multiple Ad Playlist (VMAP) 1.0.1 | http://www.iab.net/media/file/VMAP.pdf |
| Ref11 | VirginVirtuousSolutionApproach\_lmc20150505.pdf | [SharePoint link](http://sharepoint/sites/change/progs/1008264/Workstreams/Forms/AllItems.aspx?RootFolder=%2fsites%2fchange%2fprogs%2f1008264%2fWorkstreams%2fAdvertising%2f03%20Technical%20Documents&FolderCTID=&View=%7bF0AD2FC6%2dB9F7%2d4F86%2d9B27%2d1E0BB56E06AE%7d) |
| Ref12 | Ad FFW Reporting -v2.pdf | [SharePoint link](http://sharepoint/sites/change/progs/1008264/Workstreams/Forms/AllItems.aspx?RootFolder=%2fsites%2fchange%2fprogs%2f1008264%2fWorkstreams%2fAdvertising%2f03%20Technical%20Documents&FolderCTID=&View=%7bF0AD2FC6%2dB9F7%2d4F86%2d9B27%2d1E0BB56E06AE%7d) |
| Ref13 | SeaChange Interface Part 2 – Asset ingest + VM Addendum |  |
| Ref14 | Solution Architecture Virgin Media OTT Ad Insertion v0.6 clean | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/Solution%20Architecture%20Virgin%20Media%20OTT%20Ad%20Insertion%20v0.6%20clean.docx) |
| Ref15 | VOD Advertising Manifest Manipulator Infrastructure Design | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/01%20Infrastructure/VOD-Advertising-Infrastructure-Design-v1-4doc.pdf) |
| Ref16 | Virgin Media Advertising on TVA+IPVOD Proposal v11 ISSUED | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Forms/AllItems.aspx?RootFolder=%2Fsites%2Fchange%2Fprogs%2F1008264%2FWorkstreams%2FAdvertising%2F03%20Technical%20Documentshttp://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/Virgin%20Media%20Advertising%20on%20TVA+IPVOD%20Proposal%20v11%20ISSUED.pdf) |
| Ref17 | VMAP\_Response\_Standard.xml | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/VMAP_Response_Standard.xml) |
| Ref18 | VMAP\_Response\_Extended.xml | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/VMAP_Response_Extended.xml) |
| Ref19 | BAIABExtensions\_lmc20150529.pdf | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/BAIABExtensions_lmc20150529.pdf) |
| Ref20 | applePlaylist577610.m3u8 | [SharePoint Link](http://sharepoint/sites/change/progs/1008264/Workstreams/Advertising/03%20Technical%20Documents/applePlaylist577610.m3u8) |
| Ref21 | Technology\_HLA\_IP VoD Ingest(VSPP)\_VMUK\_v1\_2.docx |  |

Please contact if you have trouble accessing any of the referenced documents.

## Contributors to this document

The following people provided input into this document:

| **Name** | **Role / Company** |
| --- | --- |
| Tushar Patel | Tv Engineering |
| David Holroyd | Solution Design |
| Les Carter | Cadent |
| Elvio Caruana | Cadent |
| Dave Romrell | Arris |

## Glossary

The following terms have been used in this document:

| **Term** | **Description** |
| --- | --- |
| TVA | TV Anywhere, a Virgin Media mobile application allowing access to streaming video content and acting as a companion for the TiVo service |
| TVA | TV Anytime. Metadata format relevant to the TVA Transformer interface out of SeaChange Adrenalin. |
| ABR | Adaptive Bite rate |
| MDC | ARRIS Spectrum MDC (Manifest Delivery Controller) |
| CMS | Arris Spectrum CMS (Central Manager System) |
| STB | Set top box |
| OTT | Over-the-top, delivery of content over IP networks |
| CPE | Customer-premises equipment |
| DTV | Digital TeleVision |
| PVR | Personal Video Recorder, a set top box with an internal hard-drive for recording. |
| PID | Provider ID |
| PAID | Provider & Asset ID. The combination of provider and asset ID. Also sometimes referred to as PID/PAID. |
| VOD | Video On Demand |
| LGI | Liberty Global central organisation |
| XML | Extensible markup language |
| ADI | Asset distribution interface |
| IAB | Internet Advertising Bureau |
| SCTE | Society of Cable Telecommunication Engineers |
| VAST | Video ad serving template |
| VMAP | Video multiple ad playlist |
| CDN | Content Distribution Network |
|  |  |
|  |  |
|  |  |
|  |  |

Further glossary for the project can be found on [SharePoint](http://sharepoint/sites/support/Technology/techarch/BoDTVA/Shared%20Documents/on%20Demand%20Architecture/Projects/VTVA2/Technical%20Glossary.xlsx)

# Introduction

## Project Overview

### Imperative

VM’s obligations to serve video on demand advertising have increased significantly with new content providers coming on board, new OTT platforms to address through the TV Anywhere programme and an expectation to accrue increased revenue through targeted advertising.

Cadent Technology Advanced Advertising System (CTAAS), a leading advertising platform currently being deployed across Europe as part of LGI’s VOD Ad Insertion (VAI) initiative has been deployed to serve the QAM VOD ad insertion.

The objective of this project is to extend the Ad insertion capabilities to OTT IP VOD content for TiVo STB for pre-rolls, mid-rolls and post-rolls.

### Future State Vision

Delivery of content and advertising to OTT devices will eventually use similar architectural components but will be delivered as part of the Horizon Go project. This document focuses on delivery of targeted advertising for VoD content to TiVo STBs.

### Approach

Arris MDC (Manifest Delivery Controller) and Cadent/Blackarrow will be used to deliver dynamic advert insertion to the TiVo IP VOD platform.

## Scope

### Day 1 Scope

* Core delivery functionality to serve and play adverts on IP VOD STB
* Ingest advert (video and metadata) and publish on CDN
* Create and manage ad campaigns across all platforms
* Dynamically insert adverts into VOD plays (pre-roll, mid-roll and post-roll)
* Reporting functionality
* Client side reporting of ad impressions through SCTE 130 PSNs
* FFWD Reporting (for TIVo Only)
* Quartile reporting of ad plays (from data supplied by SCTE 130 PSNs)
* Trickplay reporting (from data supplied by SCTE 130 PSNs)

### Day 2 Scope

* Integration into 3rd Party Ad-servers for decisioning data,
* Integration into 3rd Party ad servers for ad tracking,
* Full support for mid-roll ads.

### Out of Scope

* Advert replacement into NGTV QAM based STB Linear Services
* Advert replacement into VOD QAM based STB Services
* Mid Roll & Post roll on VTVA Linear services
* Blackout Insertion
* Client side advert insertion
* 3rd party VAST wrappers – i.e. using an ADS that is not blackarrow

### Additional CRs

Note that the scope of works represented by this project also includes changes to existing functionality described by a number of change requests. These are listed below –

|  |  |
| --- | --- |
| CR | Brief Description |
| CR031 | All existing filters to incorporate ‘And’ and ‘Or’ parameter:  • And – e.g. genre contains ‘Comedy AND Drama’  • Or – e.g. genre contains ‘Comedy OR Drama’ |
| CR045 | Ability to Exclude node filters, rather than just include. Currently the solution enables the inclusion of genres, this CR is a request for the solution to enable the exclusion of genres, e.g. “NOT Genre”. The users require the ability to apply this to any node filter. |
| CR049 | VM require a fail-safe to prevent TVOD Adult content from ever appearing outside of the Adult area of the catalogue: On Demand > More > Adult |
| CR055 | The upload capacity from Adrenalin to Akamai is limited by single threaded property of CDS. This restricts the rate IPVOD assets can be made available below that required by Product Team. This is particularly important for DASH assets which consist of many thousands of short files.  Seachange to make CDS multi-threaded supporting and supports multiple connections to CDN (e.g Akamai). The concurrency and number of connections should be configurable. |
| CR059 | VM require 4 additional ADI fields to be passed in to the 4e usage data and 2b product data |

# Raid Log

## Risks

| **Risk ID** | **Description** | **Probability**  **H / M / L** | **Impact**  **H / M / L** | **Mitigating Actions** |
| --- | --- | --- | --- | --- |
| R1 | MPEG DASH clients not being able to handle multiple periods in MPD | L | H | Change control into TiVo to ensure all clients can deliver multi period support for DASH delivery |
| R2 | SCTE 35 markers in content | M | L | Working with CPs |
| R3 | No true model of scale of adverts to be served | M | M | System scaled to a high level |
| R4 | Client reporting required dev work on clients | H | L | Phase1 CRs raised to make the correct changes, Phase 2 client functionality will be delivered later in another work stream |
| **R5** | SCTE35 marker detection within WFS | H | H | Now using Ericsson VSPP as the packager. |
|  |  |  |  |  |
|  |  |  |  |  |

## Assumptions

| **Assumption ID** | **Assumption Description** |
| --- | --- |
| **A.01** | Virtuous project makes all the necessary changes to ensure the correct streaming protocols are delivered. |
| **A.02** | The Load balancers used within Virtuous (F5) can be used for the manifest requests too |
| **A.05** | Additional BlackArrow services will also not require additional hardware and current platforms are deemed suitably spec’d. |
| **A.06** | All ads will be unencrypted. Encrypting ads would complicate the DRM flow in the STB and result in additional setup delays. |
| **A.07** | Content Aggregators will distinguish Ad content from normal content using separate encoding types. |

## Issues

| **Issue ID** | **Description** | **Impact**  **H / M / L** | **Mitigating Actions** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Dependencies

| **Dependency ID** | **Project**  **ID** | **Project Name / Description** | **Dependency Description** |
| --- | --- | --- | --- |
| **D.01** |  | IPVOD Relaunch STB | Delivery of IPVOD to TiVo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Questions to be addressed

| **Question ID** | **Description** | **To be Covered By** | **Status** | **Notes** |
| --- | --- | --- | --- | --- |
| Q.1 | Communication between TiVo TSC and BlackArrow to get Ads for the Display bar | TiVo-BA | Closed | Make it simple if possible (not VAST or SCTE-130) = Phase 2 |
| Q.2 | Quartile reporting from Players, including TiVo STB | HLA | Closed | Though VAST/VMAP has been chosen, this is reliant on a TiVo PCR. |
| Q.3 | Impression reporting URL notification to players | HLA | Closed | Specific metadata, during session setup, included in MPD/m3u8, via VMAP/VAST & Blob |
| Q.4 | TrickPlay restriction notification to players | HLA | Closed | specific metadata, during session setup, included in MPD/m3u8, via VMAP/VAST & Blob |
| Q.5 | Asset repository for Ad insertion | HLA | Closed | who will manage this repository, where will be located. Will TSC include the asset? Ads held on BML. |
| Q.6 | Integration between Manifest manipulator and BA for Ad insertion | HLA | Closed | Though VAST/VMAP has been chosen, this is reliant on a TiVo PCR |
| ~~Q.7~~ | ~~Manifest manipulator solution~~ | ~~HLA~~ | ~~OPEN~~ | ~~Issue RFI to select vendor~~ |
| Q.8 | SCTE-35 management in Packager | HLA | Closed | Confirm that Packager can manage SCTE-35 marking and create MPD and m3u8 with Ad insertion info.. |
|  |  |  |  |  |

# Solution Design

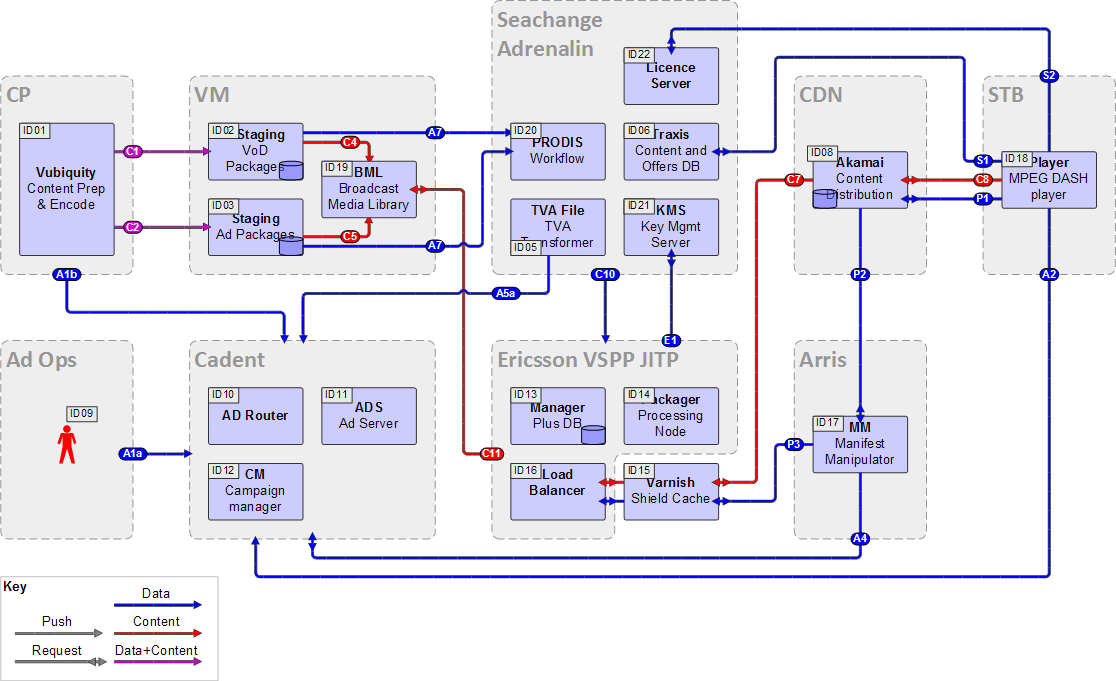


Figure ‑ - Architecture Overview

## Solution Overview

Virgin Media currently employs Blackarrow and SeaChange Adrenalin platform to perform the VOD QAM advert insertion. To extend this advert insertion functionality to the OTT space (TiVo IP VoD), Virgin Media has commissioned a new manifest manipulator platform. This Manifest Manipulator platform is the ARRIS Spectrum MDC (Manifest Delivery Controller). The MDC will perform server side advert insertion in the manifests of ABR streams, by manipulating the streaming manifests with advert manifest for the newly commissioned IP VOD platform for the TiVo platform. The Arris MDC and the Ericsson VSPP packager are the only new functional blocks within this solution, as all others are currently legacy. Changes are required to the STB player to support the PSN messages for impression reporting and to properly gather data from the DASH manifest and player.

The advert insertion placements will be performed by the Arris MDC, but the opportunities of advert insertion will be performed by the Blackarrow (BA) platform (now known as Cadent). The BA platform will be the same instance of the VOD QAM platform. When the VOD QAM BA platform was commissioned it was scaled to the take the future OTT insertion platform. The protocol for communications between the MDC & BA will be the SCTE 130 protocol in keeping with the protocol used for linear ad insertion. This protocol will also add impression reporting to allow the clients to report on customer actions. Impression reporting is carried out using the SCTE 130-3 PSN mechanism allowing Cadent near real-time updates on impressions and the ability to analyse the data to a high degree of granularity.

To enable the MDC to send streaming data and customer details to BA (Cadent) (for placement opportunities) the following information is required from the client -

* VoD PID/PAID information for VOD assets
* Device Identification to target adverts (MAC address)
* Platform type / identification

These changes are detailed further on in the document.

As stated in the scope, VOD Assets will have pre/mid/post roll adverts. To enable VOD mid roll the VOD work flow platform (using the Ericsson VSPP) is being introduced to enable SCTE35 marker detection and packaging. This will allow for a smooth transition between the VOD content and adverts. Full support for mid-roll ads will be part of phase 7b.

Adverts are supplied and managed through the system in much the same way as main content with the lifecycle of this content being managed by SeaChange components. Main content is encrypted by the VSPP platform in conjunction with the SeaChange Key Management Server (KMS) but adverts are presented unencrypted. This approach lessens the load on the STB for DRM licence management since encrypted ads would require additional calls from the STB to the Licence server to enable ad playback. In order for this encryption to work, the STB must be registered by its TSNid in Adrenalin since this is the method by which the Licence Server identifies the STB.

### Manifest Manipulation

As implied earlier in this document, manifest manipulation describes the process of creating custom manifests per content, device or session, in real-time, in response to a device request. This is done much later in the video delivery workflow to enable highly scalable, targeted ad insertion or alternate content placement. This has the unique advantage of creating a virtual session with the end-user device, enabling a long list of real-time control features.

Manifest manipulation could be seen as a simple operation. By changing the “playlist” one or multiple goals can be achieved such as:

* Inserting ads
* Inserting alternate content for blackouts
* Limiting or expanding the number of bit rates available
* Setting initial bit rates
* Specific authentication requirements
* Inserting impression tracking URL’s
* Signalling Trickplay restrictions (require client side support)

However, as with most things, it is never quite as simple as it appears. For a variety of reasons, most protocols do not take kindly to having the manifest rewritten and this has left some operators in significant difficulty.

Therefore it is extremely important that anyone thinking of using some form of manifest manipulation to achieve some specific goal understands the implications of managing and “manipulating” the manifest, and uses technology that has been proven to achieve that goal. The result of a wrong choice is often that the required process will just not work or the user experience will be completely unac­ceptable.

The advert project has inputted the need to choose the correctly streaming protocols for the virtuous project, which will described later in this document.

The MDC being employed for this project is the Arris Spectrum MDC product. This needs to be able to re-write MPEG-DASH manifests in order to merge separate ad and VoD manifests into a single document

### Playout

Playout will be on the Leapfrog and Compass STBs with the navigation and UI elements using the existing navigation.

#### Pause and Restart

The pause and restart behaviour will be similar to that already in place for QAM VoD. On pausing, the playout will pause and the position in the stream be remembered. Upon resume, the playout will resume from exactly the same point in the stream as where it was paused

If the playout is stopped, the position in the stream will be remembered. This is as per existing QAM VoD functionality. However, for IP VoD, when we restart the stream a new ad decision is requested which will result in a new set of ads inserted in the breaks. If the earlier stop point was mid ad-break then when the stream is restarted the restart will be from the start of the ad break.

Note that although the behaviour for IP VoD is the same as for QAM, the implementation will be different.

#### User Error Reporting

There are a number of reasons why the session setup will fail either at startup or once playout is underway. This can be caused by a failure in any one of the components used in the playout chain. When this occurs, the user is to be presented with an error message containing a code that points to the cause of the problem. More detailed information will also be placed in the logs for that STB allowing operations staff to diagnose the problem. Details of the error codes and causes are listed in Appendix E.

.

## Solution Design Diagram

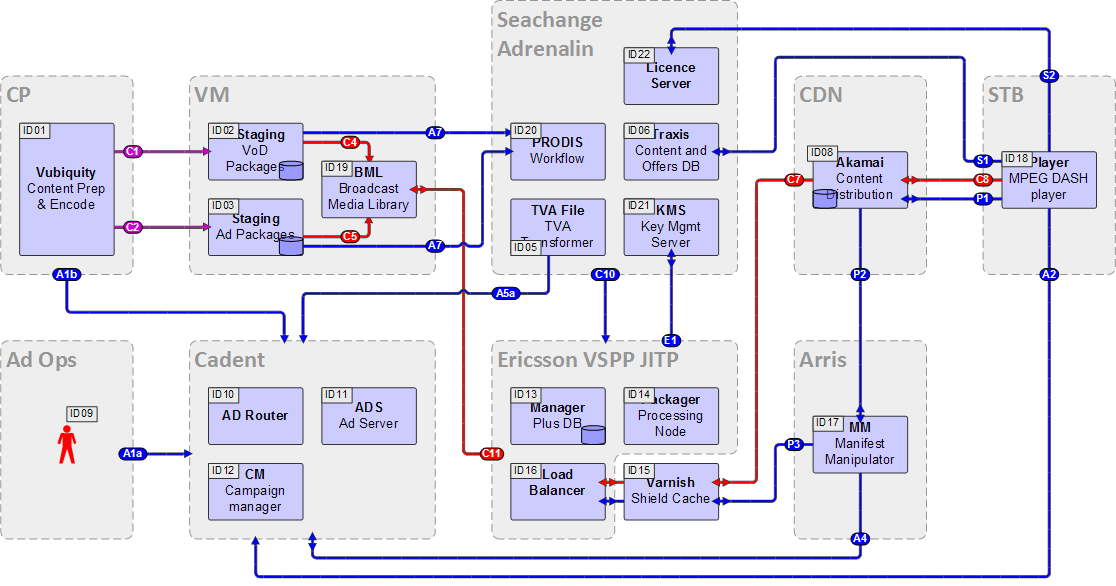


Figure ‑ - Solution Diagram overview

## Solution Elements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | System | Status | Int/Ext | Desctiption |
| ID01 | Vubiquity | Existing | External | System used by the third party content providers to package assets for ingest into VM’s IP VoD platform. Packages assets as an ADI++ package with metadata and different encodings. Encoding is expected to used closed GOPs and be in a suitable format to slice into segments for MPEG-DASH encoding. Where mid-roll breaks are to be used, the break positions must align with the boundary between two closed GOPs and additionally be signalled with a SCTE 35 marker. |
| ID02 | Staging – VoD | Existing | Internal | Staging server used as an ingest point for VoD asset packages. Third parties will provide packages to this location using SFTP. |
| ID03 | Staging – Ads | Existing | Internal | As per the above but for ads. |
| ID05 | TVA Transformer | Modified | External | Existing extract and feed to TVA (TV Anytime) containing details of the full catalogue in Adrenalin. Run on a scheduled basis. Supplies a zipped file of the full catalogue metadata. In this deployment, we are reusing this file to also supply Cadent with required metadata including internal break positions and the mapping between PID/PAID and the SeaChange ContentID. |
| ID06 | Seachange  Traxis | Existing | Internal | Provides functionality for management of subscriptions and purchases. In this deployment, this is the component which manages authorisation for the session to the STB and provides a URL through which the STB can request the stream.  This component also provides information to the TiVo platform through the SeaChange 5fv2 interface allowing for the presentation of assets on the EPG. |
| ID08 | Akamai  CDN | Modified | External | Content Distribution Network outsourced to Akamai. Used for both VTVA & IP VOD TiVo platform. Changes – Domain name routing, SSL connection between Clients and F5 Load balancers. |
| ID09 | Ad Operations | Existing | Internal | This represent the department that manage the creation and management of campaigns on the Cadent Central systems through the Web UI. This is existing functionality and the process will be identical to that of the existing QAM VoD process. |
| ID10 | Cadent  Ad Router | Modified | Internal | The Cadent component that manages requests from clients on to the Cadent Ad Decisioning Server. This component also embellishes requests as they pass through by looking up profile information in the Subscriber Information Server based on the STB ID and adds this to the request that is sent to the ADS. Additional request decoration is performed by using the Cadent CIS and POIS components to determine the program metadata and user experience respectively. |
| ID11 | Cadent  ADS | Modified | Internal | The Cadent component that makes ad insertion decisions in respect of SCTE 130-3 placement requests from the MM. This will supply a list of suitable ads to be inserted as pre, mid and post-roll positions in the main VoD content. |
| ID12 | Cadent  CM | Existing | External | The Cadent component that is used to manage campaigns. Users access this service through a web interface. This is part of the Cadent Campaign suite of components and is hosted in Amsterdam in Cadent’s data centre. |
| ID13 | Ericsson  VSPP Manager | New | Internal | The component within the Ericsson VSPP suite which manages the function of the other components and provides a central point for control. This component also hosts the database which houses the data that controls the VSPP suite and the content therein. |
| ID14 | Ericsson  JITP | New | Internal | Just-in-time Packager. This is the component in the Ericsson VSPP suite that is used to create manifests for VoD and ad asset content as requested by the MM. This component also packages and encrypts segments of the content when segments are requested by the CDN. There are multiple JITP nodes that sit behind a VM-supplied Varnish Shield Cache load balancer. |
| ID15 | Varnish  Shield Cache | New | Internal | Cache and load balancer that sits in front of the Ericsson VSPP components. To be implemented by VM. Balances load onto the VSPP nodes using DNS round-robin on to the IP addresses of the VSPP nodes. |
| ID16 | Ericsson  Load Balancer | New | Internal | Interface point on to the JITP nodes. Provides load balancing of requests on to the pool of JITP nodes. |
| ID17 | Arris  MM MDC | New | Internal | Manifest Manipulator. New component to perform the insertion of ads into MPEG-DASH streams for VoD. Manages the request for a MPEG-DASH manifest from a client by stitching a number of ad manifests into the VoD content manifest as determined by the SCTE 130-3 placement message from Cadent and then supplying this back to the client. |
| ID18 | TiVo  STB | Modified | External | Changes to the STB to allow the playback of IP VoD streams using the MPEG-DASH protocol. Functionality uses the MPEG-DASH player libraries that form part of the STB’s Opera browser. Functionality should be delivered in the TiVo phase 7a code for the STB with full support for mid-rolls in 7b.  Also includes changes to allow the reporting of ad impressions through the SCTE 130 PSN message to the ad router. This functionality is also used in linear ad insertion. |
| ID19 | VM  BML | Modified | Internal | Virgin Media – Broadcast Media Library. Used as a repository of content with content being placed here by SeaChange once ingested into their systems. This location can be used by downstream systems (VSPP in this case) as a source of content. |
| ID20 | SeaChange  Prodis | Modified | Internal | SeaChange workflow management. Will need workflows configured to detect incoming content, ingest, initiate VSPP processing and initiate TVA Transformer export for Gracenote/Cadent. |
| ID21 | SeaChange  KMS | Modified | Internal | SeaChange Key Management Server. Provides PlayReady encryption keys to VSPP. |
| ID22 | SeaChange Licence Server | Existing | Internal | SeaChange Licence Server. Manages DRM for DASH (and other) content on the STB using PlayReady DRM. Implements the PlayReady challenge response with the STB for the content keys. |

Table ‑ – Solution Elements

## Solution Physical Architecture

All servers required for the MDC & CMS platforms will hosted on Cisco UCS Blades via VMWare ESX. The design is inherently resilient, so there is no requirement for vMotion, or any other VMWare optimisation. For this infrastructure the use of VMware is purely for consolidation.

The servers will be in the VOD Domain for Production, Pre-Production and DR. TnT Windows Support already have accounts in these Domains as a result of the Adrenaline platform roll out.

For further details about the physical design please refer to the “VOD Advertising Manifest Manipulator Infrastructure Design”, which can be found at in References:- Ref15

A network design will be required to ensure MDC has access to Akamai/CDN & Blackarrow

## Security

{{Use this section to detail any specific security requirements that must be considered in the detailed design. Ensure that all areas of the solution adhere to any relevant VM policies and baselines with particular focus on those entailing EA DPA requirements for safeguarding personal data, e.g., ‘Cryptographic Controls’, ‘Information Security Policy For Systems Acquisition Dev Test’ at:

[http://sharepoint/CTIO/IS/Information%20Security%20Policies/Forms/AllItems.aspx}](http://sharepoint/CTIO/IS/Information%20Security%20Policies/Forms/AllItems.aspx%7d)}

### Application

Any Application development should ensure that it follows the guidelines at <https://www.owasp.org/index.php/Main_Page>

Leapfrog boxes use HTTPS (SSL) to access the Akamai CDN. The CDN has controls to limit access from non-authorised clients (due to legal content contracts). The communications between the clients, CDN & F5 load balancers will to be upgraded to SSL to protect customer account ID / MAC address of TiVo.

Compass boxes access the CDN and the F5/MM using HTTP. However, the traffic for Compass boxes originates within VM’s networks at the CMTS and is routed through NAT servers. Configuration is required to allow Compass traffic from the NAT servers to route through to the CDN and to the F5 load balancers using HTTP from within the VM network.

As no additional public interfaces are exposed as part of the advert insertion project, so no penetration testing needs to be carried out.

### Network

The Arris MDC servers will be installed within VM private network, though we use public interfaces, which are protected via a VPN to Akamai. This VPN is already present as part of the virtuous project.

A more detail network diagram / layout will be explained later in the document.

### Solution Deficiencies

| **Detailed Req Ref** | **Non/Partial Compliance** | **Compliance Gap** | **Reason for Gap, Proposed Workarounds** |
| --- | --- | --- | --- |
|  |  | The C10 Seachange to VSPP interface (3d in SeaChange nomenclature) allows for content to be distributed when the VSPP reports “IngestingAndPlayable”. Our solution requires distribution only on “Ingested”. Discussions are ongoing to resolve this. |  |

# Solution Design Detail

## Detailed Design Documents

The following documents should be generated from this document during the Detailed Design phase:

| **Document ID** | **Document name** | **Document description** | **To be produced by** |
| --- | --- | --- | --- |
| **DDREF01** | Network infrastructure design | Design document covering the network infrastructure changes, and capacity planning of infrastructure | Data Network Engineering |
| **DDREF02** | Adverts preparation and streaming workflow | Low level design work flow of the SCTE35, streaming URLs, Akamai and routing to MDC cluster | TV Engineering |
| **DDREF03** | Infrastructure Design | Server infrastructure design of the MDC | Infrastructure team |
| **DDREF04** | MDC cluster | Low level design of MDC & configures | Arris |
| **DDREF05** | SCTE 130 MDC to BA | Low level design SCTE 130 user cases and configures | Arris & BA |
| **DDREF06** | SeaChange-SOL-VM\_JITP\_Advertisement | Changes to Adrenalin platform to enable advert insertion.  VSPP IF-3d propagation of content for JITP. | SeaChange |
| **DDREF07** | Metadata flow | Ingress & flow of Advert metadata into SeaChange to Blackarrow | TV Engineering |
| **DDREF08** |  |  |  |

Table ‑ - Design Documents

## Interfaces

The following diagram shows key components and key interfaces that are covered in this solution design.

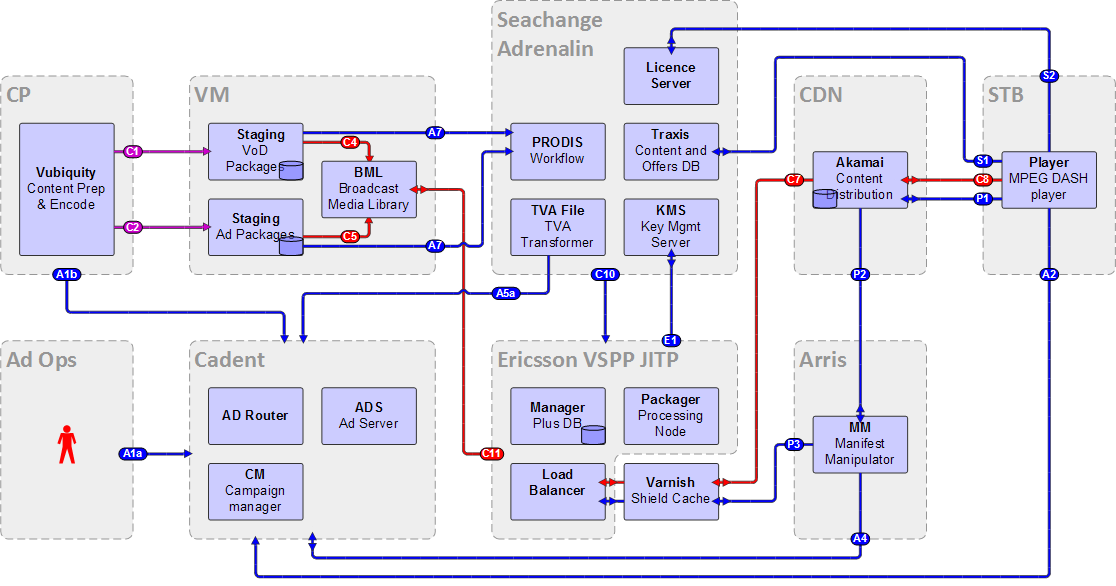


Figure ‑ - Interface Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| I/F | Description | Protocol | Specification |
| A1a | **Ad Campaign through Ad Ops.**  Covers the creation and management of ad campaigns by the ad ops team through the web interface provided by the campaign management module in Cadent Central. | HTTP / Web UI |  |
| A1b | **Ad Campaign Ingest from Third Party.**  Integration point into Cadent campaign management from the third party’s ad sales and management system. This feature will be implemented as part of phase 7b. | <TBC – phase 7b> |  |
| A2 | **Impression Tracking**  The STB reports impressions of ads played back to Cadent whenever it completes playback of an ad or a partial playback. The Placement Status Notification (PSN) message contains details of the ad asset played, the speed and duration of the playback, tracking information and details of the last user interaction with the STB (allowing Cadent to apply rules to infer actual impressions). | HTTP / SCTE 130-3 PSN |  |
| A4 | **MM to Cadent – Ad Template Data**  This is the call that the MM makes when compiling the eventual manifest that will be sent to the STB. Here, the MM asks Cadent for details of ads that are to be inserted into the specified VoD title. Cadent replies with a list of substitutions in a SCTE 130-3 reply plus tracking metadata that it requires the MM to additionally insert into the manifest. | HTTP / SCTE 130-3 |  |
| A5a | **SeaChange TVA (TV-Anytime) Transformer Metadata and Break Data to Cadent**  This is the interface through which SeaChange supply Cadent with the break data for the VoD asset. The break position data will originally have been ingested to Adrenalin through the ADI++ supplied metadata and placed into Adrenalin’s database. This data along with key data including PID/PAID, title and SeaChange back office ID is made available on this interface. | SFTP / TVA | No documentation available. |
| A7 | **SeaChange Adrenalin Metadata Ingest**  This is the ADI metadata from the supplied packages being parsed and ingested into Adrenalin under the control of PRODIS. The package metadata at this stage will reside on the staging servers and will be moved to a different folder by the workflow once ingest complete. | HTTP / ADI++ | SeaChange Interface Part 2 - Asset ingest + Seachange Interface Part 2 - Asset ingest - Addendum for Virgin Media\_v3\_6.pdf |
| C1 | **VoD Package Ingest**  Vubiquity supplies ADI packages for VoD titles by placing them on the staging server. Delivery is by SFTP to a temporary filename with an in-place rename made once the transfer is complete. The ADI package is a zip file containing the various essences plus the ADI metadata document describing the asset, its renditions and break position data. | SFTP / zipped ADI++ | SeaChange Interface Part 2 - Asset ingest + Seachange Interface Part 2 - Asset ingest - Addendum for Virgin Media\_v3\_6.pdf |
| C2 | **Ad Package Ingest**  As per C1 but for ad assets. Uses a different folder on the staging server. | SFTP / zipped ADI++ | Seachange Interface Part 2 - Asset ingest - Addendum for Virgin Media\_v3\_6.pdf |
| C4 | **Ingested Asset to BML**  Once Adrenalin ingest processing is complete, the PRODIS workflow manager moves the asset to the BML where it can be accessed by VSPP. BDL and the staging server provide FTP access to the files and directories to allow this. | FTP |  |
| C5 | **Ingested Ad to BML**  As above but for ads. | CIFS |  |
| C7 | **CDN Pull Content Segments from VSPP**  As MPEG DASH segments are requested by the STB from the CDN, the CDN either satisfies these from cache or forwards the request to VSPP via the Varnish Shield Cache. Varnish will either satisfy the request directly or forward to the VSPP. VSPP packages and encrypts the requested segment on-the-fly and returns to the CDN. | MPEG-DASH |  |
| C8 | **STB Requests Content Segments from CDN**  As the STB plays the content under the direction of the MPEG DASH manifest it fetches the individual segments of content from the CDN. The exact segment variant requested will depend on network conditions. | HTTP / MPEG DASH  HTTP (Compass)  HTTPS (Leapfrog) |  |
| C10 | **Adrenalin Registration of Content in VSPP**  This is the interface from the Adrenalin workflow manager (PRODIS) to VSPP to initiate the ingest of the content into VSPP. Adrenalin provides VSPP with the title, content id, provider id, profile and a URL pointing to the content on the BML where VSPP can pull the content. | HTTP with XML body | SeaChange Interface Part 3d - Content propagation. |
| C11 | **VSPP Pull from BML**  This is the interface where VSPP pulls the essences from the BML using the URL provided in the registration message (as per interface C10). | FTP |  |
| E1 | **VSPP Requests Encryption Key**  As part of the on-the-fly packaging process, VSPP will encrypt the content. This interface allows VSPP to fetch the encryption key from SeaChange. | HTTP - PlayReady |  |
| P1 | **STB to CDN – Request Title Manifest**  On playout of IP VoD content, the STB will request the MPEG DASH manifest for that title from the Manifest Manipulator (MM) via the CDN. The MM will query for ad insertion instructions, gather together the manifests for the title and ads and then stitch all of these into a single combined manifest before returning to the STB. For this call, the CDN is configured to simply pass the request through to the MM; if can’t be cached at the CDN since each request will potentially result in a different set of ads to be inserted. | HTTP / MPEG DASH |  |
| P2 | **CDN to MM – Request Title Manifest**  The CDN to MM part of the process described above. For this request, the CDN doesn’t cache but simply forwards the request to the MM. | HTTP / MPEG DASH |  |
| P3 | **MM to VSPP – Request Manifests for Main Asset and Ads**  In generating the combined manifest for the client, the MM needs to fetch separate manifests for the main title and for each of the ads that the ADS has indicated should be included. The MM satisfies these from its own cache or alternatively requests these from the VSPP (via the Varnish Shield Cache at the VSPP end). When a request hits the VSPP, the VSPP creates the manifest on-the-fly and returns it. | HTTP / MPEG DASH |  |
| S1 | **TiVo to Traxis – Get URL for VoD Title**  The STB, having got a title ID from the IP VoD EPG courtesy of TSC now queries Traxis for a URL for this title that will allow the STB to pull the content from the CDN. Traxis takes care of any authentication and then supplies the URL to the STB. | HTTP | SeaChange Interface Part 5j - Session Management. |
| S2 | **TiVo to SeaChange Licence Server**  Once the STB has the manifest for the content it can start to set up the playback of the content segments. Part of the manifest received will contain PlayReady DRM details that enable the STB to initiate the PlayReady DRM challenge/response with the Licence Server to allow the STB to obtain the content key for the main asset. | HTTPS – PlayReady |  |

Table ‑ - Interfaces descriptions

## Content Management

### Content Delivery

VOD Content providers / encoding houses will be asked to deliver the advert assets, including all the different renditions required by the MPEG DASH protocol. The Advert assets should be H.264 encoded, encapsulated in M2TS file, with Subtitles (STL) and second audio tracks (audio descriptor) (if available). This content will be delivered in the same way as the Virtuous project, STB VOD Re-launch - SDD in References. If mid-rolls are to be used, the content should contain SCTE 35 markers to indicate where the breaks are to take place. Full details of the required format can be found in the “IP VoD Ingest (VSPP) Virgin Media UK” document [Ref21].

The content will be encoded using the advert insertion signalling markers (STCE35 standard – “Digital Program Insertion Cueing Message for Cable” - ANSI/SCTE 35 2013 – in References) to ensure the H264 GOP structures align with the advert breaks. This ensures no encoding artefacts during slicing. The markers (SCTE-35) will also be placed into the MP4 file for use up chain. The files will later be packaged & DRM’ed as required within the VSPP using PlayReady encryption under the control of the SeaChange Key Management Server (KMS).

Adverts will be delivered to the same staging server as the VOD content, and ingested in the same manner. This will ensure the content can be encoded (if not already) with the correct profiles and packaged. All assets (VoD & Advert) will be placed on the correct server to ensure propagation to the BML. VOD & Adverts assets will be uplinked via FTP.

The delivery of VOD to the VSPP platform will be controlled via the work flow manager at SeaChange, this will include both main content and ads. The system will be configured to prevent the addition of ads to the catalogue. Detail of this solution can be found within the Virtuous project, STB VOD Re-launch - SDD. The VSPP platform will manage the packaging and encryption of the content.

### Advert signalling overview

SCTE-35 signalling will be used to indicate the appropriate points where adverts may be inserted. To carry out seamless advert insertion play-out, the Primary Content (typically a TV program or a movie) is switched to Advert content at the appropriate points. Once the advert content has played to completion then the play-out control returns from the advert to the Primary Content.

In SCTE-35 the points (splice-points) at which play-out control can safely exit from the Primary Content are known as Out-Points. The points at which the play-out control can safely return to the primary stream are known as In-Points. SCTE-35 may be used to transport splice-point signalling from the network source to network edge where the adaptive segmentation will be performed.

Within Virgin Media implementation the SCTE35 markers are not currently being transferred over the metadata with the manifest file. This data transfer can also be enhanced via ESAM connection between the Packager and the ADS (in VM case BA). Again, this is out of scope.

This SCTE35 standard supports frame accurate signalling of events in MPEG-2 transport streams along with associated descriptive data. This standard supports the splicing of MPEG-2 transport streams for the purpose of Digital Program Insertion, which includes advertisement insertion and insertion of other content types. An in-stream messaging mechanism is defined to signal splicing and insertion opportunities.

The STCE35 standard – “Digital Program Insertion Cueing Message for Cable” - ANSI/SCTE 35 2013 – in References, will be adhered to ensure the content can be spliced correctly

This standard defines Splice Points. Splice Points in an MPEG-2 transport stream provide opportunities to switch elementary streams from one source to another. They indicate a place to switch or a place in the bit stream where a switch can be made.

### Packaging Content for Advert Insertion.

The SCTE35 markers / annotations placed within the MPEG2TS/MP4 files represent the placement advert boundaries. The SCTE-35 marker will not fall precisely at the natural segment boundaries and therefore the encoder and packager will need to ensure that:

1. The encoder starts a new closed GOP on encountering a SCTE-35 marker designed to indicate an ad avail or programme marker
2. The packager splits the segment at this point, placing the marker between the two resulting segments

Furthermore, to improve stream stability, if possible, the encoder/packager should implement a look ahead mechanism to avoid segments less than 1 seconds in duration (target segment duration as specified by the manifest should be set higher to the modal average).

The diagram below illustrates this with a SCTE-35 marker injection

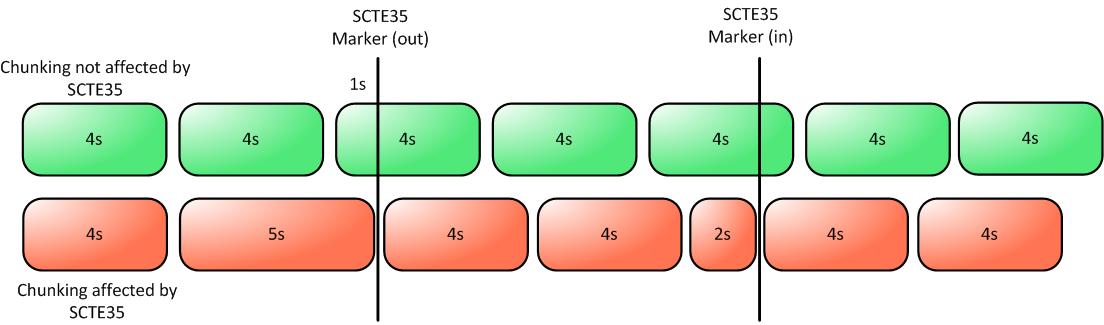


Figure ‑- Chunking the segments

### SCTE35 Processing in VSPP

The Ericsson VSPP packager will be required to honour the SCTE 35 signalling in the incoming assets in order to create package manifests with boundaries as specified by the signalling. In summary, the VSPP will need to ensure that –

* There are closed GOPs surrounding any SCTE 35 markers in the stream,
* Where there are SCTE 35 markers in the stream that these are used to start a new period in the eventual packaged content,
* Segment breaks happen at exactly the same point in time across all renditions allowing clients to seamlessly switch between different bitrate renditions,
* Segment sizes are managed such that they remain roughly at the target duration with small variations being allowed to position SCTE 35 breaks and to ensure that no excessively small segments are delivered.

### Asset Management & Metadata

#### Assets management

All VOD assets (content and adverts) are ingested and provisioned into the SeaChange Video CMS (Adrenalin).

This ensures the knowledge of all video content (including relevant metadata, such as asset IDs) is stored into one single repository.

There is a SeaChange Adrenalin platform for the unified TiVo STB VOD catalogue, this being delivered by the STB VOD Re-launch project.

Content providers will be delivering VOD content encoded specifically for the target STB device. The ad copy metadata (being advert metadata, streaming url, etc) is required by Blackarrow to add to the placement response to Arris.

A good detailed description of relevant interfaces between the SeaChange Adrenalin and other systems in the ecosystem can be found in the documents ***SeaChange Interface Part 2 - Asset ingest*** and ***Seachange Interface Part 2 - Asset ingest - Addendum for Virgin Media\_v3.8***.

#### Presentation on STB

The catalogue of titles that the customer is able to view is managed by the Adrenalin platform with this feeding the downstream TiVo platform to allow for the presentation on the STB UI.

Since ads are now ingested into the platform alongside the main content, these are managed in Adrenalin as assets in their own right. It is important that filters be set up to prevent these assets being offered directly to the STB in the catalogue. Ad assets can be detected by their having an Encoding\_Type of “H264-ADV”.

#### Interior Break Data

Interior break data, if present, will appear in the movie assets in the form:

<App\_Data App="MOD" Name="InteriorBreakNPTs" Value="720,1410" />

This is described in the “Seachange Interface Part 2 - Asset ingest - Addendum for Virgin Media\_v3...” document.

#### Encoding type table

The following encoding types will be used in the preparation of content as per the encoding specification. The Encoding\_Type values will be passed into Adrenalin and from there will be available to Traxis and to the TVA Transformer file output to Cadent. These values will be used to indicate to Cadent the type of content metadata it is receiving from Adrenalin allowing it to target on this information if that is desired. This also allows Cadent to distinguish between VoD assets and ads since these are treated differently in the Cadent system.

VSPP also receives this parameter when being instructed to ingest from Adrenlin/Prodis but doesn’t perform any special processing based on this value.

Note the table also includes a UHD variant. This is not used at present.

|  |  |  |
| --- | --- | --- |
|  | Streaming protocol | Encoding\_type |
| TiVo SD | **DASH iso bmff Live Profile** | **H264-SD** |
| TiVo HD | **DASH iso bmff Live Profile** | **H264-HD** |
| TiVo UHD | **DASH iso bmff Live Profile** | **H264-UHD** |
| TiVo Ads | **DASH iso bmff Live Profile** | **H264-ADV** |

Table ‑ - Encoding types vod & advert assets

SeaChange Adrenalin program assets use Encoding\_Type to identify the encoding type. Note that “Encoding\_Type” refers to the parameter in the ADI XML; this may be named differently in downstream interfaces.

#### Asset Identification

With the metadata for all VOD content (as described above) Cadent will be able to perform replacement opportunities on any content asset.

The manifest manipulator will inform Cadent of the asset ID of the content for which it is requesting ad decisioning within the ad decision request.

The manifest manipulator will extract the relevant IDs from the original manifest request URL. For IP VOD this will be the asset PID/PAID, which SeaChange (TRAXIS) will be required to append to the end of the PlayURL which itself is based on the Adrenalin BackOfficeID.

### Matching Ad Content to Main Content

In order to present a consistent user experience it is required that the ads encoding be matched to that of the main content. This is to ensure that ads are presented in a similar quality to the content that is being viewed. This means that attention must be paid to the main content type – SD, HD and eventually UHD and steps taken to present ad content in a similar resolution and bitrate.

In doing this, we also want to ensure that there is only one asset for the ad in the Cadent system. This is important from a campaign management perspective.

Most of this processing is performed within the Arris MDC since this is geared specifically for this type of manifest manipulation. The MDC receives full manifests for ads and main content from the VSPP with these containing all available renditions (bitrate/resolution/audio variants). The MDC is then able to analyse the main content manifest and filter the advert manifest components such that they contain renditions compatible to the main content being displayed. This matching will be based on bitrate and resolution with separate rules covering the audio to ensure that this is also compatible.

Where ads are not available in the required quality, the MDC will be able to simply omit these from the final manifest. It is imperative therefore that particular attention is paid to ad encoding to ensure that it fully meets the encoding specifications in terms of available renditions.

The Encoding\_Type parameter mentioned above describes the content type but is not used directly in the decision of what appears in the final manifest. However, this is useful metadata for Cadent to have since it potentially allows targeting to be based on a distinction between HD and SD content if this is so desired. The VSPP is also presented with this metadata but takes no specific action on it.

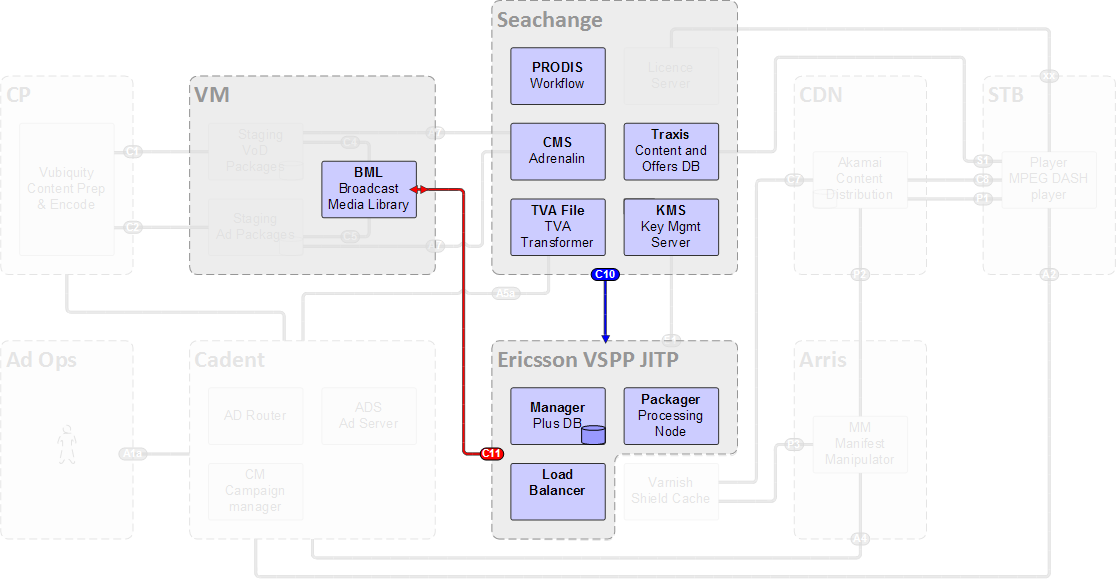
Another part of the mechanism relies on a feature of the VSPP that allow rules to be defined based on a &device=xxx parameter that is attached to the URL used to request a manifest from the VSPP. The MDC uses this parameter to signal to the VSPP whether it is serving an ad or main content. These rules are set up as follows –

|  |  |
| --- | --- |
| &device= | Meaning |
| TiVo | Used for main content. Return whatever layers are present since HD and SD offerings will only have been supplied with the relevant layer types and thus all are relevant to return in this request.  This rule will also require that the content is encrypted and that the returned manifest contains the information required to initiate the PlayReady DRM setup. |
| TiVoAdvert | Return all available renditions (since the MDC will be filtering).  Do not apply encryption; ads are served in the clear. |

The Cadent SCTE 130-3 ad decision URLs (passed back to the Arris MDC) will ensure that the “&device=TiVoAdvert” parameter is present which will later be used by the MDC in ad manifest requests into the VSPP. For main content, the “&device=TiVo” parameter will already be present having been part of the original URL the STB was supplied with by Traxis and with this being passed through the CDN and into the MDC.

## Content Processing in VSPP

The SeaChange Adrenalin platform is used to manage content metadata in the IP VoD platform and propagates this through to the front-end systems that make this available to the STB and manage purchases, rights and subscriptions (Traxis).



However, the content itself, in terms of video file fragments and MPEG DASH manifests is managed by the Ericsson VSPP platform. This platform is composed of multiple subsystem components and is organised into a number of nodes to spread the processing and storage across a number of servers. Details of the hardware components and their interconnectivity are given in the “Network & Server Infrastructure” section of this document.

Although the VSPP will not actually perform encoding of assets, it will be responsible for analysing and distributing them across its internal network such that manifests and content segments are available on-demand to client systems. Client systems in our scenario are the MM (for manifests) and the CDN (for content segments).

### Flow

The process is triggered by the SeaChange Prodis workflow manager once the metadata for the asset is successfully ingested to Adrenalin. Prodis will then inform the VSPP of a package to ingest through interface C10 (3d in SeaChange documentation) through an HTTP PUT with an XML document payload specifying the asset to ingest and the location on the staging servers from which VSPP can pull the essence files.

VSPP will perform a number of QA steps on the asset and then, if successful, will proceed to process the essence files into the VSPP storage nodes and the accompanying manifest metadata into the VSPP database.

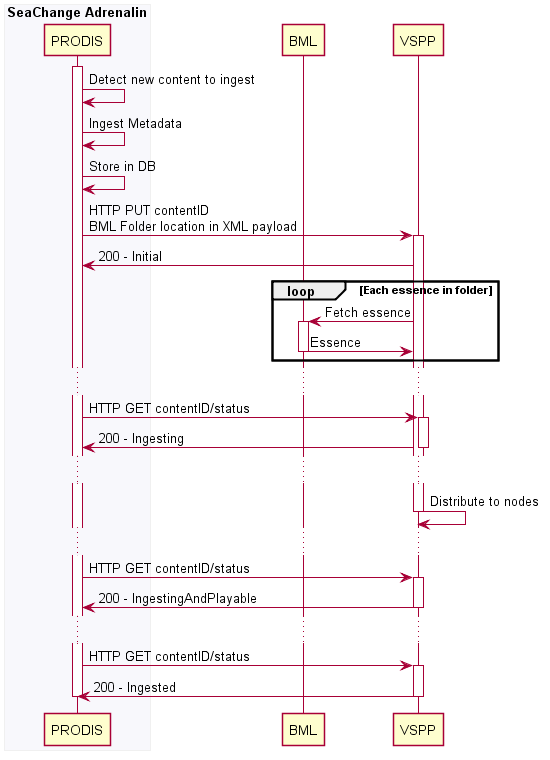
VSPP expects the various bitrate/resolution renditions of the DASH asset to be supplied ready-encoded. This content should have SCTE 35 markers at the internal breaks and should be structured into a series of closed GOPs allowing the VSPP to fragment the asset as per MPEG DASH specifications. These fragments are then distributed across the VSPP storage nodes.

Some content will contain subtitles in STL format. Where these are provided, the VSPP is required to convert these to the WebVTT format for distribution.

During this process, Prodis periodically polls VSPP for the status of the ingest. Through this mechanism, VSPP will indicate whether the asset is still being processed, is ready or if it has failed the VSPP QC. The offer will only be made available by Prodis/Traxis once processing has been successfully completed by VSPP.

### Sequence

The following diagram shows these events in terms of a sequence diagram –



Following the completion of this process, the asset is ready to be notified into Cadent via the TVA Transformer interface (described in a later section).

### SeaChange to VSPP Messaging

#### Ingest Notification

SeaChange will notify the VSPP through an HTTP PUT –

PUT /Content/4d854d85-4e6c-d06f-b15b-d06fa6d85edf HTTP/1.1

The request follows the REST convention in that the URL of the PUT message also specified the content ID.

The body of the message will contain an XML document as follows -

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Content

id=*"4d854d85-4e6c-d06f-b15b-d06fa6d85edf"*

xmlns=*"urn:eventis:cpi:1.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"urn:eventis:cpi:1.0 SeaChange3D.xsd "*>

<Name>Spiderman 3</Name>

<Provider>Sony</Provider>

<SourceUri>ftp://10.0.1.1/spiderman3/</SourceUri>

<ProfileName>H264-HD</ProfileName>

</Content>

Following receipt of this message, the VSPP should reply immediately to acknowledge that it has received and understood the request. SeaChange will poll later for updates on the status of this ingest.

The initial reply will appear something like the following –

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Content

id=*"4d854d85-4e6c-d06f-b15b-d06fa6d85edf"*

xmlns=*"urn:eventis:cpi:1.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"urn:eventis:cpi:1.0 SeaChange3D.xsd "*>

<Status id=*"Initial"*/>

</Content>

#### Status Poll

Periodically, after the initial PUT, Adrenalin will poll the VSPP for the status of the previously initiated ingest. The poll will be in the form of an HTTP GET. This will be composed, in REST format, with a URL containing the ID of the content being queried followed by the string “/Status”.

GET /Content/4d854d85-4e6c-d06f-b15b-d06fa6d85edf/Status HTTP/1.1

The VSPP is expected to reply with an HTTP response with a 200 status code (for successful query) and a body containing the current status.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Content

id=*"4d854d85-4e6c-d06f-b15b-d06fa6d85edf"*

xmlns=*"urn:eventis:cpi:1.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"urn:eventis:cpi:1.0 SeaChange3D.xsd "*>

<Status id=*"Ingesting"*>

<IngestPercentage>20</IngestPercentage>

</Status>

</Content>

For a successful response, the status might be any one of the following –

|  |  |
| --- | --- |
| Initial | The VSPP has registered the asset for ingest but has not yet begun the ingest process. |
| Ingesting | The VSPP is currently processing and <IngestPercentage> indicates progress. |
| IngestingAndPlayable | As above but the content can be streamed out. Not expected to be used in this scenario since for VoD we require that the full manifest is available before it can be used. A partial manifest would not work since the manifest manipulator requires the full manifest to work on. |
| Ingested | The ingest is complete and the asset is ready for use on the VSPP. |

For an unsuccessful call, the status code will indicate an error and the body will contain an error response. For an error response, the status code in the message body may be one of the following –

|  |  |
| --- | --- |
| Error-IngestHostUnreachable | VSPP couldn’t pull the content from the URL specified. |
| Error-IngestSourceNotFound | VSPP couldn’t find one or more of the ingest source files. |
| Error-InsufficientIngestBandwidth | Not relevant in this scenario. |
| Error-OustOfStorageSpace | VSPP out of storage space. |
| Error-Other |  |

Once Prodis has received an “Ingested” status back for the content then it can safely advertise that content to Traxis for use. Traxis will supply a URL to the eventual client that the client can use to access the content, this URL will be based on a number of parameters such as the device type, encoding type and segment duration plus the SeaChange back-office content ID.

#### Essences

Since we are dealing with an ABR system, the assets delivered will have a number of different renderings ranging from low bitrate to high bitrate in accordance with the VM encoding specification. When Adrenalin notifies VSPP of the content through the above message, it simply supplies a folder location that contains all renderings; the VSPP should ingest all of these renderings.

#### Update Notification

The platform as a whole allows for metadata updates following the initial ingest although updates to the content itself are always handled as a delete followed by a new ingest. Since the VSPP is dealing just with content, it will never be required to process an update.

#### Delete Notification

As with updates, control of deletion of content is entirely under the control of SeaChange Adrenalin. The deletion request will come over the same interface as the initial ingest request. Adrenalin is wholly responsible for managing the lifecycle of content on the platform and will base this on the licensing dates in the content as originally delivered in the ADI file. In addition, Adrenalin will add a grace period (3 days) to content before deleting from the platform.

The deletion request is simply an HTTP PUT request without any body (other than standard headers) –

DELETE /Content/4d854d85-4e6c-d06f-b15b-d06fa6d85edf HTTP/1.1

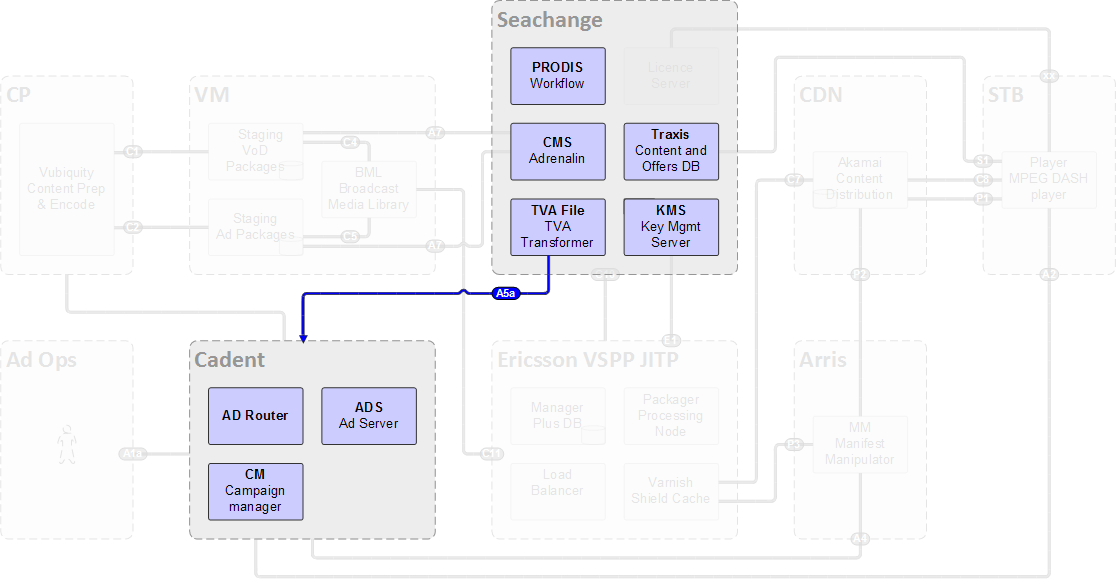
The VSPP is expected to return a successful status code –

HTTP/1.1 200 OK

Once this request is delivered to the VSPP by Adrenalin, Adrenalin considers the content deleted. It is the responsibility of VSPP to manage the deletion process either immediately or when existing requests have been completely satisfied.

## Metadata Delivery to Cadent

As part of the content preparation workflows, the SeaChange Adrenalin platform needs to make certain metadata regarding ingested assets available to Cadent.



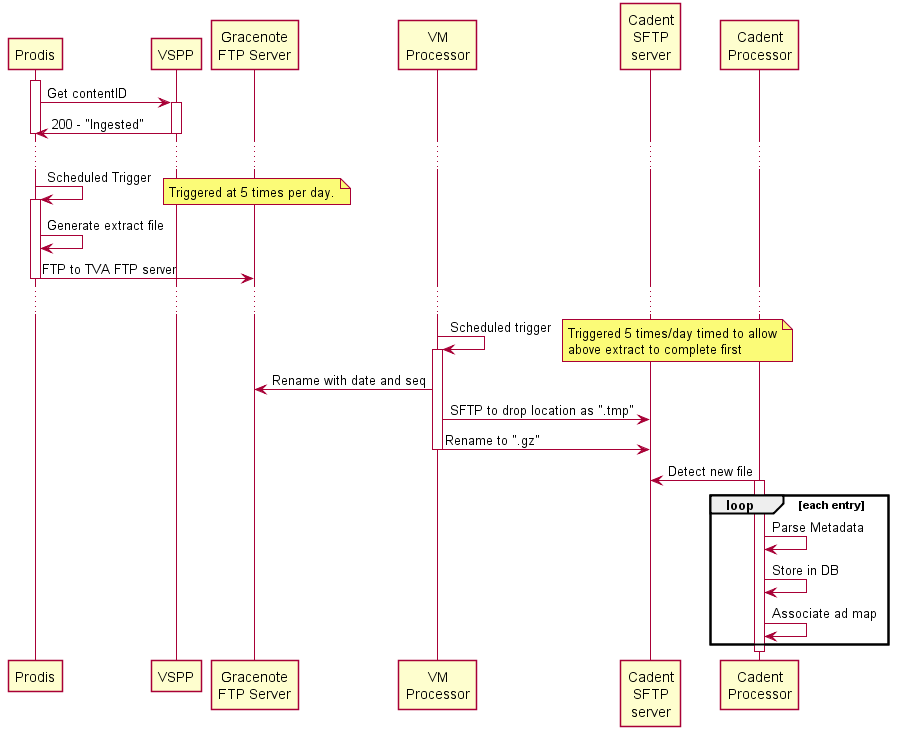
Cadent needs to know the following information –

* The SeaChange content ID (since this is the ID that will be used in the playout workflows),
* The Provider ID and Provider Asset ID (PID/PAID) of the content (since campaign management in Cadent will know the content by these IDs instead of the SeaChange ID),
* The Clock Number of ad content (since this will be needed in Cadent to track campaign impressions),
* Internal break timing data for VoD content that is to have mid-roll ads inserted,
* Encoding Type as passed in on the ADI file so that Cadent can know if the content is HD, SD or an ad,
* Title, episode title and synopses,
* Series and Show name,
* HD indicator,
* Folder name (as in where presented on catalogue),
* Ratings,
* Duration,
* Licensing start and end dates

This data is conveyed to Cadent in an XML file as described below.

### Sequence

The following diagram summarises the process.



The first two steps in the above diagram summarise the SC to VSPP ingest process whereby VSPP prepares the content and distributes it to its internal nodes. Once this process has completed successfully the content can be included in the extract file that is produced on the TVA transformer interface.

The extract is initiated by a scheduled trigger several times a day and then goes on to produce an extract file containing details of all VoD content in that instance of the Adrenalin system. This is created as a single dump file which is then transferred to a drop location known to Cadent through SFTP. Note that this last step is not controlled by SeaChange but is a VM developed process.

The VM process names the file according to a unique name containing the date and time thus allowing processing to target the latest delivery. Filenames will follow the format –

stbvod\_***<environment>***\_yyyymmdd-***<seq>***.xml.gz

Where ***<environment>*** is one of “dev”, “func”, “preprod”, “live” or “dr” and ***<seq>*** is a number from 1 to 5 denoting which of the five daily drops this file is. File sequences/drops are according to the following timetable; the file will be delivered to Cadent some time after its creation but before the next extract.

|  |  |
| --- | --- |
| Seq | Creation Time |
| 1 | 00:01 |
| 2 | 06:00 |
| 3 | 12:00 |
| 4 | 16:00 |
| 5 | 20:00 |

Hence, a typical file might be named as follows –

stbvod\_live\_20170523-3.xml.gz

The file will be transferred by SFTP and will have an extension of “.tmp” during the transfer. Once the transfer is complete, the file is remotely renamed to its final name signalling to Cadent that the file can be processed. After this point, it is up to Cadent to process the file and remove or archive it when done.

Cadent will detect the completion of the transfer to the drop location (temp file and rename) and initiate processing of the data within. The file will contain details of new content plus details of previously notified content. The Cadent process will need to parse the file and compare it with its own view of the world and thus determine which are new assets, which are changed and which are deleted. The Cadent database should be updated accordingly. During this process, Cadent will apply rules to determine how ads should be applied to the new content and will store that data alongside the content.

#### SeaChange Changes

A change is required to the configuration of PRODIS.PA which in turn feeds the TVA Transformer through SeaChange interface 8c. The change allows the interface to map though additional data ensuring that it appears in the TVA Transformer file. The additional fields to be mapped through are –

* InteriorBreakNPTs

#### Processing and Filtering

The TVA Transformer file arrives as a zipped archive containing a single large XML document detailing all the assets on that Adrenalin instance. The Cadent process will need to unzip the file and then process the file with reference to its own existing database to detect updates and additions.

Since the file is a full system extract, the file will contain IP VoD assets and ads but will also contain details of the QAM VoD content that also resides on this platform. The Cadent ingest process will need to apply filtering to pull out just the entries applicable to IP VoD. There are a number of ways in which IP VoD content could be identified in the file. One possible approach would be to use the “Encoding\_Type” attribute and only use the entries where this is “H264-SD”, “H264-HD” and “H264-UHD” for main content and “H264-ADV” for ads.

The TVA Transformer file contains all entries known to Adrenalin; this will include entries that are still not fully ingested onto the VSPP platform. The Cadent ingest process should note where this is the case and ensure that ads that are not present on the VSPP are not placed in ad decisions. For main content, this early visibility is not a problem since there will be no decision requests for content until it has been published on the Traxis platform. The ingest process might use this early information to partially populate its database and note the status, updating this when the full metadata comes through. Full metadata and availability on the platform is indicated by the presence of an <OnDemandProgram> element in the metadata and the details contained within it.

#### Updates

Metadata for assets and ads may be amended and passed through the platform. This is performed by the content provider or Vubiquity submitting an amendment ADI package for ingest. This package will indicate that it is an update and will contain an updated version number allowing the content management system (Adrenalin/Prodis) to detect it as such. In these cases, the amended metadata will appear on the TVA Transformer file. The Cadent ingest process will need to detect the update by comparing the new data with that already stored on its database; there are no indicators in the data to show that this is an update. One possible approach to optimise this process would be to take a hash of certain key fields and store this on the Cadent database thus allowing a change to be more easily detected. However, whatever approach is adopted, it must not assume any ordering or consistency of elements in the XML document other than those explicitly stated in the schema definition (XSD).

Updates to essences, i.e. the raw content contained within the ADI package, will not be processed as an update. If there are any changes to the content, these will be processed as a deletion followed by an ingest of the new content. In these cases, the replacement content will have a new PID/PAID and will also have a new ID on the SeaChange systems.

#### Deletions

For deletion, the absence of a title in the new TVA file should **not** be used to trigger a deletion of that data on Cadent, instead, deletion should be carried out through an expiry of the content as signalled by the licensing window in the metadata. The normal operational method of deletion is to pull in the end date on the licence window and send an update through SeaChange which will then come through on the TVA Transformer file. Note that SeaChange apply a grace period of 3 days before deleting on their systems so the logically deleted item will appear in a number of TVA file drops before disappearing. This method is preferred over simply noting the absence of an item in the TVA file as it protects against the accidental deletion of the whole database in the event that a corrupt null file is received from SeaChange.

In the event that an item is forcibly deleted on the SeaChange system, there will be no notification through this mechanism. This forcible deletion is expected to happen rarely and in these circumstances would require manual intervention to also remove the item from Cadent.

The deletion might also be missed by Cadent if the Licensing End Date is changed to a date further in the past than the grace period. In this case the asset will simply not be present on the TVA file. It might be prudent to report on assets that are thought to be current in the Cadent system but which do not appear in the TVA file. These assets could then be candidates for manual deletion in the Cadent system.

#### Sizing

The TVA Transformer file will potentially contain a large number of entries since it is a complete extract of the assets on the Adrenalin system.

The number of IP VoD entries in the file is expected to grow over time to around 24,000 VoD assets by the year 2020. In addition, there will at all times be up to 1000 ad assets live on the platform.

The ingest process must be able to process this volume of data in a timely manner such that this works within the expected delivery schedule of 5 times per day.

### Interface File Format

The interface file is as per the TVA Transformer stripped TVA.xml file.

#### Structure

The file is divided up into a number of sections each with a number of entries.

The <ProgramInformationTable> element contains <ProgramInformation> elements, each describing a title in the system.

The <GroupInformationTable> table contains a number of <GroupInformation> elements, these describe, amongst other things, the show and season structure and membership.

The <ProgramLocationTable> contains a number of <OnDemandProgram> elements which describe a particular format of an asset (e.g. QAM VoD, IP VoD) that are available for showing. These relate to the <ProgramInformation> elements through the <GroupId> attribute. These elements describe the actual show or movie assets in various forms whereas the <ProgramInformation> element logically provides a title that applies to all these assets.

Conceptually, the file looks like this-

<TVAMain>

<ProgramDescription>

<ProgramInformationTable>

<ProgramInformation/>

<ProgramInformation/>

<ProgramInformation/>

</ProgramInformationTable>

<GroupInformationTable>

<GroupInformation/>

<GroupInformation/>

<GroupInformation/>

</GroupInformationTable>

<ProgramLocationTable>

<OnDemandProgram/>

<OnDemandProgram/>

<OnDemandProgram/>

</ProgramLocationTable>

</ProgramDescription>

</TVAMain>

#### Key Data

The following data is to be extracted from the TVA Transformer file. The descriptions below are given in the form of XPath expressions relative to the root of the TVA file. Note that the information comes from multiple sections in the file and will need to be combined in the import process to get the full picture. For information, the source within the ADI file is also given.

|  |  |
| --- | --- |
| Item | Description |
| ContentID | Generated by SeaChange. This is the ID by which the content is known in the SeaChange ecosystems. This will be the ID that forms part of the URL that the STB uses to request playback and will also be the primary identity of the content on the VSPP. Accessible in the TVA Transformer file for <OnDemandProgram> elements using the following XPath-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /InstanceDescription[1]  /OtherIdentifier[@type="VodBackOfficeId"][1]  Note that there is also a **VodBackOfficeId** that applies to the title of which the above content is a part. This has no relevance for this extract procedure. |
| SC\_crid | Generated by SeaChange. The internal ID by which the content is known in the SeaChange database. Useful here since this value is used to match up the “**ProgramInformation**” and “**OnDemandProgram**” information blocks in the file. Accessible in the TVA file “ProgramInformation” section using the following XPath-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]/@programId  and in the “OnDemandProgram” block as-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /Program[1]/@crid |
| Title | This is the content title as present on the ADI file and passed in through SeaChange. In the case of main content this will likely be a brief string naming the movie title or a particular episode of a series. For ads, this will be a string composed of the ad’s clock number and a brief campaign title, e.g. “Fairy Liquid - xyx8342j2 ”. This field is the main way that assets are ordered in the Cadent system and is the field that will most commonly be used to locate content.  Located in the TVA Transformer file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Title[not(@type)][1]  Originally sourced from the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App=”VOD” and @Name=”Title”]@Value |
| Title\_Sort\_Name | Originally sourced in the ADI package and mapped through to the TVA Transformer file. Specifies the string that the title should be sorted on and usually omits initial words such as “a” and “the” as well as normalising case. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /ShortTitle[@type=\"secondary\"][1]  Originally sourced from the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App=”VOD” and @Name=”Title\_Sort\_Name”]@Value |
| Episode\_Title | Originally sourced in the ADI package and mapped through. Where the title is a series, this field gives the specific episode title. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Title[@type="episodeTitle"][1]  Originally sourced in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App=”VOD” and @Name=”Episode\_Name”]@Value |
| Summary\_Short | From ADI package. The short synopsis. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Synopsis[@length="short" and @xml:lang="en-GB"][1]  Originally sourced in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Summary\_Short"]@Value |
| Summary\_Medium | From ADI package. The short synopsis. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Synopsis[@length="medium" and @xml:lang="en-GB"][1]  Originally sourced in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Summary\_Medium"]@Value |
| Summary\_Long | From ADI package. The short synopsis. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Synopsis[@length="long" and @xml:lang="en-GB"][1]  Originally sourced in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Summary\_Long"]@Value |
| Series\_Name | If the title is a member of a series, it will contain a link to a <GroupInformation> element later in the file that describes that series (including its name). If it is a member of a series, it will have a link to a crid of the <GroupInformation> element describing the series. This can be obtained from the TVA file thus-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /MemberOf[@xsi:type=\"EpisodeOfType\"]/@crid  See the section “Show and Series” on details of how to extract this. |
| Show\_Name | This is a lookup from the series GroupInformation element as described above. See the section “Show and Series” on how to extract this information. |
| Provider | Originally from the ADI file. The name of the content provider. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /CreditsList[1]  /CreditsItem[@role="urn:eventis:metadata:cs:RoleCS:2010:CONTENT-PROVIDER"][1]  /OrganizationName[@xml:lang="en-GB"][1]  Originally sourced from the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /AMS[1]@Provider |
| Title PID | Provider ID in the form of a short (usually 3 characters) code. This is passed through SeaChange and made available on the TVA interface. Used in conjunction with PAID. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /CreditsList[1]  /CreditsItem[@role="urn:eventis:metadata:cs:RoleCS:2010:CONTENT-PROVIDER-ID"][1]  /OrganizationName[@xml:lang="en-GB"][1]  Originally present in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /AMS[1]@Provider\_ID  Note that the Content PID\_PAID field below should be used in preference to this. |
| Title PAID | Provider Asset ID. Identifier that uniquely identifies the asset in the content provider’s ecosystem. Always used in conjunction with the PID hence the common terminology PID/PAID. This is passed through SeaChange and made available on the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /OtherIdentifier[@type="PID\_PAID" and @authority="eventis"][1]  Originally present on the ADI file at-  concat(  /ADI  /Asset[1]  /Metadata[1]  /AMS[1]@Provider\_ID  ,”#”,  /ADI  /Asset[1]  /Metadata[1]  /AMS[1]@Asset\_ID)  Note that the Content PID\_PAID field below should be used in preference to this. |
| Content PID\_PAID | The PID and PAID of the content asset that forms the offer (as distinct from the title to which it belongs). Found in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /InstanceDescription[1]  /OtherIdentifier[@type=\"PID\_PAID\" and @authority=\"eventis\"][1]  This will usually be a concatenation of the above two fields but, significantly, is taken from a different location in the TVA file. This field is more specific than the PID and PAID above in that it applies to the asset offered and should be the preferred source of this data.  Originally present on the ADI file at-  concat(  /ADI  /Asset[Metadata[1]/AMS[1]@AssetClass=”movie”]  /Metadata[1]  /AMS[1]@Provider\_ID  ,”#”,  /ADI  /Asset[Metadata[1]/AMS[1]@AssetClass=”movie”]  /Metadata[1]  /AMS[1]@Asset\_ID) |
| InteriorBreakNPTs | A comma delimited list of mid-roll break timings in milliseconds. Passed in from the ADI file to SeaChange Adrenalin and made available to Cadent via the TVA file at-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /InstanceDescription[1]  /Genre[@href="virginmedia.com:InteriorBreakNPTs" and @type="other"][1]  /Definition[1]  Note that this comes from the “OnDemandProgram” element that has the same SC\_Crid as the main “ProgramInformaion” element.  Present in the ADI file at-  /ADI  /Asset[1]  /Asset[Metadata[1]/App\_Data[@Name=”Type"]=”movie”]  /Metadata  /App\_Data[@Name=”InteriorBreakNPTs”]@Value  Note that the “/ADI/Asset” element contains multiple sub assets, these being for the movie, poster, trailer etc. The expression above selects the movie asset. |
| External\_Reference | An additional identifier that identifies the asset in the content provider’s systems. For main content this is likely to be a CRID (Customer Reference ID). For ads this will be the ad’s clock number. In the TVA file this is-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Genre[@href=\"urn:virginmedia:metadata:cs:CustomTitlePropertyCS:2014:ExternalReference\"][1]  /Definition[1]  In the ADI file this is-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="External\_Reference"]@Value |
| Encoding\_Type | Placed on the ADI file when the content is prepared. We are interested only in assets with the types “**H264-SD**”, “**H264-HD**”, “**H264-UHD**” or “**H264-ADV**”. Provides an indication of the content type to downstream systems who may make decisions based on the type. This is made available on the TVA file to Cadent (and to VSPP on the 3d interface). Cadent may use this to target to HD or SD specifically. Present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /InstanceDescription[1]  /Genre[@href="urn:eventis:metadata:cs:PropertyCS:2010:profile" and @type="other"][1]  /Definition[1]  Note that this comes from the “OnDemandProgram” element that has the same SC\_Crid as the main “ProgramInformaion” element.  Sourced from the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@Name=”External\_Reference”]@Value |
| Is\_Advert | There is no specific field for this indicator but it is possible to derive it from the **Folder\_Location** field. |
| HD Indicator | Indicates if this is an HD asset. Derived from the TVA file with the following expression-  count(  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /InstanceDescription[1]  /Genre[@href="urn:eventis:metadata:cs:PropertyCS:2010:IsHD" and @type="other"][1]  )  This yields 0 for no and 1 for yes and relies on the presence or absence of the attribute in the “OnDemandProgram” element. Note also that this is from the “OnDemandProgram” section.  Originally sourced from the ADI file at-  /ADI  /Asset[1]  /Asset[Metadata[1]/App\_Data[@Name="Type"]="movie"]  /Metadata[1]  /App\_Data[@Name="HDContent"]@Value |
| Folder\_Location | Indicates what content type catchup/archive/kids/ads etc. Relates to a hierarchy in the catalogue and is used by Cadent in classifying content. Retrieved from the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Genre[@href=\"virginmedia.com:Folder\_Location\"][1]  /Definition[1]  Originally in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="PTP" and @Name="Folder\_Location"]@Value  (Note: ‘App=”PTP”’) |
| Licensing\_Window\_Start | The date at which the asset can be made available on the VM service. Assets will generally be loaded ahead of this date and time but availability on the STB will be limited by this value. Available in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /StartOfAvailability[1]  Originally sourced in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Licensing\_Window\_Start"]@Value |
| Licensing\_Window\_End | The date at which the asset ceases to be available on the platform. After this date, the title will no longer be displayed on the EPG. Internally, within Adrenalin, there is a grace period of 3 days following this date after which the title is notified to other systems for deletion. Cadent should use this field as an indicator for deletion, deleting the title when this date passes. Note that Adrenalin may force an early deletion by sending an update of this field with a date in the past. The field is present in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /EndOfAvailability[1]  Originally sourced in the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Licensing\_Window\_End"]@Value |
| Rating | The age or suitability rating as defined by the censors. This is indicated on the file by the presence of an attribute matching one of the ratings. The attribute can be extracted with the following XPath-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /ParentalGuidance  /ParentalRating[starts-with(@href,"urn:schange:metadata:cs:OriginalRatingCS")][1]/@href  Possible attribute values that are returned by this are -   * urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.G * urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.U * urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.18 * urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.12 * urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.15 * urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.PG   These are mapped in Adrenalin from original values in the ADI file specified at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Rating"]@Value  The values here will be just the rating part, i.e. “G”, “U”, “18”, “12”, “15” or “PG”. |
| Content\_Filesize | This value is present in the TVA file although for ABR streamed content it has little meaning since the stream that the STB receives is a subset of the essences supplied as part of the original package. The values is extracted using the following expression-  /TVAMain  /ProgramDescription  /ProgramLocationTable  /OnDemandProgram[1]  /InstanceDescription[1]  /AVAttributes[1]  /FileSize[1]  Sourced from the ADI file at-  /ADI  /Asset[1]  /Asset[Metadata[1]/App\_Data[@Name=”Type"]=”movie”]  /Metadata[1]  /App\_Data[@Name=”Content\_Filesize”]@Value |
| Run\_Time | Duration of the asset. Available in the TVA file at-  /TVAMain  /ProgramDescription  /ProgramInformationTable  /ProgramInformation[1]  /BasicDescription[1]  /Duration[1]  Originally sourced on the ADI file at-  /ADI  /Asset[1]  /Metadata[1]  /App\_Data[@App="VOD" and @Name="Run\_Time"]@Value |

The communication of data over the TVA Transformer file contains all data on the Adrenalin instance. This may include assets that are not yet fully ingested to the VSPP system and will also include assets not applicable to IP VoD (i.e. QAM VoD). The Cadent ingest process must be aware of this and filter accordingly.

#### Show and Series

Some items described by <ProgramInformation> elements may describe episodes of a series, these in turn may belong to a show. Both series and show are described in the TVA Transformer file as <GroupInformation> elements and related through their crid values.

If a <ProgramInformation> element belongs to a series, it will have a <MemberOf> element which can be accessed using the following XPath-

/TVAMain

/ProgramDescription

/ProgramInformationTable

/ProgramInformation[1]

/MemberOf[@xsi:type=\"EpisodeOfType\"]/@crid

An example might appear as follows-

<MemberOf xsi:type=*"EpisodeOfType"* crid=*"crid://schange.com/f10bbee5-3880-4166-855d-c94414c6501d"* index=*"8"* />

The crid value here relates to a <GroupInformation> element that describes the series. For example-

<GroupInformation

groupId=*"crid://schange.com/f10bbee5-3880-4166-855d-c94414c6501d"*

numOfItems=*"10"*>

<GroupType

xsi:type=*"ProgramGroupTypeType"*

value=*"series"* />

<BasicDescription>

<Title xml:lang=*"en-GB"*>Season 2</Title>

<Genre

href=*"urn:eventis:metadata:cs:CustomGroup:2012:Serie"*

type=*"other"* />

</BasicDescription>

<MemberOf

xsi:type=*"MemberOfType"*

crid=*"crid://schange.com/60a984ef-4286-459d-a65d-f1d11ba01ef3"*

index=*"2"* />

<OtherIdentifier

type=*"VodBackOfficeId"*

organization=*"eventis"*

authority=*"eventis"*>f10bbee5-3880-4166-855d-c94414c6501d</OtherIdentifier>

</GroupInformation>

This, in turn relates to a show through another <MemberOf> element link. In this example, the series links to the following show group-

<GroupInformation

groupId=*"crid://schange.com/60a984ef-4286-459d-a65d-f1d11ba01ef3"*

numOfItems=*"2"*>

<GroupType

xsi:type=*"ProgramGroupTypeType"*

value=*"show"* />

<BasicDescription>

<Title xml:lang=*"en-GB"*>Ash Vs Evil Dead</Title>

<Genre

href=*"urn:eventis:metadata:cs:CustomGroup:2012:Serie"*

type=*"other"* />

</BasicDescription>

<OtherIdentifier

type=*"VodBackOfficeId"*

organization=*"eventis"*

authority=*"eventis"*>60a984ef-4286-459d-a65d-f1d11ba01ef3</OtherIdentifier>

</GroupInformation>

Any implementation should be tolerant of links not being present and where the association to a show or series cannot be made, an empty string should be used.

#### Example

The following snippet shows a section from such a file with the fields of interest highlighted –

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<TVAMain xmlns:xsd=*"http://www.w3.org/2001/XMLSchema"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xml:lang=*"en-GB"* publisher=*"eventis.nl"* publicationTime=*"2017-05-19T06:50:03.5994396Z"*

xmlns=*"urn:tva:metadata:2010"*>

<ProgramDescription>

<ProgramInformationTable>

<ProgramInformation programId=*"crid://schange.com/STZ/TITL0000000001198800"*>

<BasicDescription>

<Title xml:lang=*"en-GB"*>Ash Vs Evil Dead (s2): ep 08 (HD)</Title>

<Title type=*"episodeTitle"* xml:lang=*"en-GB"*>Ashy Slashy</Title>

<ShortTitle length=*"30"* xml:lang=*"en-GB"*>Ash Vs Evil Dead (s2): ep 08 (

</ShortTitle>

<ShortTitle length=*"33"* type=*"secondary"* xml:lang=*"en-GB"*>

Ash Vs Evil Dead (s2): ep 08 (HD)

</ShortTitle>

<Synopsis length=*"short"* xml:lang=*"en-GB"*>TEST-NPT Ruby, Kelly and

Pablo go to look for Ash and Baal.

</Synopsis>

<Synopsis length=*"medium"* xml:lang=*"en-GB"*>TEST-NPT Ruby, Kelly and

Pablo go to look for Ash and Baal.

Adult Content, Adult Language,

Graphic Violence, Strong Sexual

Content

</Synopsis>

<Synopsis length=*"long"* xml:lang=*"en-GB"*>TEST-NPT Ruby, Kelly and

Pablo go to look for Ash and Baal

in an abandoned asylum and

encounter some crazy characters. Adult

Content, Adult Language,

Graphic Violence, Strong Sexual Content

</Synopsis>

<Genre href=*"urn:tva:metadata:cs:VirginMediaCS:2010:58"*>

<Name xml:lang=*"en"*>Drama</Name>

</Genre>

<Genre href=*"urn:tva:metadata:cs:VirginMediaCS:2010:219"*>

<Name xml:lang=*"en"*>Sci Fi &amp; Fantasy</Name>

</Genre>

<Genre href=*"urn:eventis:metadata:cs:AdiGenreCS:2010:Series"*

type=*"other"*>

<Name>Series</Name>

</Genre>

<Genre href=*"urn:eventis:metadata:cs:PropertyCS:2010:msoRating"*

type=*"other"*>

<Definition xml:lang=*"en-GB"*>Adult Content, Adult Language,

Graphic Violence, Strong Sexual Content

</Definition>

</Genre>

<Genre href=*"urn:eventis:metadata:cs:PropertyCS:2010:extraData.1"*

type=*"other"*>

<Definition>PC, web, androidPhone, iphone</Definition>

</Genre>

<Genre href=*"urn:eventis:metadata:cs:PropertyCS:2010:extraData.3"*

type=*"other"*>

<Definition>wifi</Definition>

</Genre>

<Genre href=*"virginmedia.com:Folder\_Location"* type=*"other"*>

<Definition xml:lang=*"en-GB"*>Archive</Definition>

</Genre>

<Genre href=*"virginmedia.com:Display\_Provider"* type=*"other"*>

<Definition xml:lang=*"en-GB"*>Starz</Definition>

</Genre>

<Genre href=*"virginmedia.com:UI\_Location"* type=*"other"*>

<Definition xml:lang=*"en-GB"*>Drama</Definition>

</Genre>

<Genre href=*"virginmedia.com:UI\_Location"* type=*"other"*>

<Definition xml:lang=*"en-GB"*>Sci Fi &amp; Fantasy</Definition>

</Genre>

<Genre

href=*"urn:virginmedia:metadata:cs:CustomTitlePropertyCS:2014:ExternalReference"*

type=*"other"*>

<Definition>11139769</Definition>

</Genre>

<ParentalGuidance>

<ParentalRating

href=*"urn:schange:metadata:cs:OriginalRatingCS:2012:bbfc.18"* />

</ParentalGuidance>

<Language>en</Language>

<CreditsList>

<CreditsItem role=*"urn:eventis:metadata:cs:RoleCS:2010:STUDIO-CODE"*>

<OrganizationName xml:lang=*"en-GB"*>STZ</OrganizationName>

</CreditsItem>

<CreditsItem role=*"urn:eventis:metadata:cs:RoleCS:2010:CONTENT-PROVIDER"*>

<OrganizationName xml:lang=*"en-GB"*>Starz

</OrganizationName>

</CreditsItem>

<CreditsItem

role=*"urn:eventis:metadata:cs:RoleCS:2010:CONTENT-PROVIDER-ID"*>

<OrganizationName xml:lang=*"en-GB"*>STZ</OrganizationName>

</CreditsItem>

<CreditsItem role=*"urn:eventis:metadata:cs:RoleCS:2010:DISTRIBUTOR"*>

<OrganizationName xml:lang=*"en-GB"*>vubiquity.co.uk

</OrganizationName>

</CreditsItem>

<CreditsItem role=*"urn:eventis:metadata:cs:RoleCS:2010:STUDIO-NAME"*>

<OrganizationName xml:lang=*"en-GB"*>Starz

</OrganizationName>

</CreditsItem>

</CreditsList>

<RelatedMaterial>

<HowRelated href=*"urn:eventis:metadata:cs:HowRelatedCS:2010:10"* />

<MediaLocator>

<MediaUri>11139769</MediaUri>

</MediaLocator>

</RelatedMaterial>

<RelatedMaterial>

<HowRelated href=*"urn:eventis:metadata:cs:HowRelatedCS:2010:poster"* />

<MediaLocator>

<MediaUri>https://adrenalin.anywhere.mspp.dtv.virginmedia.com:8086/Posteris/6bbc2d43-d95c-499e-ba6f-44e63567c42f-poster.png

</MediaUri>

</MediaLocator>

</RelatedMaterial>

<ProductionDate>

<TimePoint>2016</TimePoint>

</ProductionDate>

<ProductionLocation>US</ProductionLocation>

<Duration>PT27M30S</Duration>

</BasicDescription>

<OtherIdentifier authority=*"eventis"* organization=*"eventis"*

type=*"VodBackOfficeId"*>6bbc2d43-d95c-499e-ba6f-44e63567c42f</OtherIdentifier>

<OtherIdentifier authority=*"eventis"* organization=*"eventis"*

type=*"PID\_PAID"*>STZ#TITL0000000001198800</OtherIdentifier>

<MemberOf crid=*"crid://schange.com/3238e7c7-d005-4020-be09-cfaa63b54a99"*

index=*"76"* xsi:type=*"MemberOfType"* />

<MemberOf crid=*"crid://schange.com/18ec1718-f48f-4676-8a90-9791d608c666"*

index=*"649"* xsi:type=*"MemberOfType"* />

<MemberOf crid=*"crid://schange.com/e4a3222d-ce6d-4d6d-b710-912eed3d26ae"*

index=*"29"* xsi:type=*"MemberOfType"* />

<MemberOf crid=*"crid://schange.com/f10bbee5-3880-4166-855d-c94414c6501d"*

index=*"8"* xsi:type=*"EpisodeOfType"* />

</ProgramInformation>

</ProgramInformationTable>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<GroupInformationTable>

<GroupInformation

groupId=*"crid://schange.com/f10bbee5-3880-4166-855d-c94414c6501d"*

numOfItems=*"10"*>

<GroupType xsi:type=*"ProgramGroupTypeType"* value=*"series"* />

<BasicDescription>

<Title xml:lang=*"en-GB"*>Season 2</Title>

<Genre href=*"urn:eventis:metadata:cs:CustomGroup:2012:Serie"*

type=*"other"* />

</BasicDescription>

<MemberOf xsi:type=*"MemberOfType"*

crid=*"crid://schange.com/60a984ef-4286-459d-a65d-f1d11ba01ef3"* index=*"2"* />

<OtherIdentifier type=*"VodBackOfficeId"* organization=*"eventis"*

authority=*"eventis"*>f10bbee5-3880-4166-855d-c94414c6501d</OtherIdentifier>

</GroupInformation>

<GroupInformation

groupId=*"crid://schange.com/60a984ef-4286-459d-a65d-f1d11ba01ef3"*

numOfItems=*"2"*>

<GroupType xsi:type=*"ProgramGroupTypeType"* value=*"show"* />

<BasicDescription>

<Title xml:lang=*"en-GB"*>Ash Vs Evil Dead</Title>

<Genre href=*"urn:eventis:metadata:cs:CustomGroup:2012:Serie"*

type=*"other"* />

</BasicDescription>

<OtherIdentifier type=*"VodBackOfficeId"* organization=*"eventis"*

authority=*"eventis"*>60a984ef-4286-459d-a65d-f1d11ba01ef3</OtherIdentifier>

</GroupInformation>

</GroupInformationTable>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<ProgramLocationTable>

<OnDemandProgram>

<Program crid=*"crid://schange.com/STZ/TITL0000000001198800"* />

<ProgramURL>vod://415ffbb9-24dd-4b88-a835-6c0d78434226</ProgramURL>

<InstanceMetadataId>imi:d2e25aded5d2f8ee9bb202c7ab38f0e8

</InstanceMetadataId>

<InstanceDescription>

<Genre href=*"urn:eventis:metadata:cs:ScreenFormatCS:2010:2"*

type=*"other"*>

<Name xml:lang=*"en"*>Widescreen</Name>

</Genre>

<Genre href=*"urn:eventis:metadata:cs:ScreenFormatCS:2010:5"*

type=*"other"* />

<Genre href=*"urn:eventis:metadata:cs:PropertyCS:2010:IsHD"* type=*"other"* />

<Genre

href=*"urn:eventis:metadata:cs:EncryptionAndCopyProtectionCS:2010:copyProtection"*

type=*"other"* />

<Genre

href=*"urn:eventis:metadata:cs:EncryptionAndCopyProtectionCS:2010:cgmsA"*

type=*"other"*>

<Definition>3</Definition>

</Genre>

<Genre href=*"virginmedia.com:InteriorBreakNPTs"* type=*"other"*>

<Definition xml:lang=*"en-GB"*>420,825,1650</Definition>

</Genre>

<Genre href=*"urn:eventis:metadata:cs:PropertyCS:2010:profile"*

type=*"other"*>

<Definition xml:lang=*"en-GB"*>H264-HD</Definition>

</Genre>

<Genre href=*"urn:schange:metadata:cs:ResolutionCS:2015:HD"* type=*"other"* />

<PurchaseList>

<PurchaseItem>

<Price currency=*"GBP"*>0</Price>

<DRMDeclaration>

<DRM>urn:schange:metadata:usagerules</DRM>

<LicenseExpression>&lt;?xml version="1.0" encoding="utf-16"?&gt;

&lt;OPL xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="urn:schange:metadata:2013:usagerules"&gt;

&lt;APS

xmlns=""&gt;1&lt;/APS&gt;

&lt;/OPL&gt;</LicenseExpression>

</DRMDeclaration>

</PurchaseItem>

</PurchaseList>

<AVAttributes>

<FileSize>1650962984</FileSize>

<BitRate maximum=*"3750000"*>3750000</BitRate>

<AudioAttributes>

<MixType href=*"urn:eventis:metadata:cs:AudioPresentationCS:2010:3"*>

<Name xml:lang=*"en"*>Stereo</Name>

<Definition xml:lang=*"en"*>Stereographic sound</Definition>

</MixType>

<AudioLanguage purpose=*"urn:tva:metadata:cs:AudioPurposeCS:2007:6"*>en</AudioLanguage>

</AudioAttributes>

<VideoAttributes>

<Coding href=*"urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:3"*>

<Name xml:lang=*"en"*>MPEG-4 Visual</Name>

<Definition xml:lang=*"en"*>MPEG-4 Visual Coding Format

</Definition>

</Coding>

</VideoAttributes>

</AVAttributes>

<OtherIdentifier authority=*"eventis"* organization=*"eventis"*

type=*"VodBackOfficeId"*>415ffbb9-24dd-4b88-a835-6c0d78434226</OtherIdentifier>

<OtherIdentifier authority=*"eventis"* organization=*"eventis"*

type=*"PID\_PAID"*>STZ#ASST0000000001198800</OtherIdentifier>

<OtherIdentifier authority=*"eventis"* organization=*"eventis"*

type=*"Genre:other"*>420,825,1650</OtherIdentifier>

</InstanceDescription>

<PublishedDuration>PT27M30S</PublishedDuration>

<StartOfAvailability>2016-11-21T06:00:00Z</StartOfAvailability>

<EndOfAvailability>2017-10-03T04:59:59Z</EndOfAvailability>

</OnDemandProgram>

</ProgramLocationTable>

</ProgramDescription>

</TVAMain>

The file contains a full dump of the assets on the Adrenalin database and is structured into several parts detailing the metadata and location data separately. Cadent will need to parse this data to relate the two sections.

Since this is a full dump, Cadent will also need to compare the relevant entries here with its view of the world and update accordingly. This approach will also easily facilitate a re-synchronisation run should the two systems come out of sync. Another consideration is that this file is not tailored specifically for Cadent and will contain catalogue data for QAM VoD in addition to the IP VoD data; this must be filtered in the ingest to Cadent.

### URL Generation

Part of the data that Cadent needs in order to build SCTE 130-3 responses is a URL pointing to the manifest of the ads on the VSPP. (The main content URL is not needed since the MM already has that from the original request). Since there is no way for the VSPP to pass back its generated manifest URLs to Adrenalin this means that Adrenalin, in turn, cannot pass the URLs back to Cadent.

Instead, manifest URLs will be based on the SeaChange BackOfficeID with a defined prefix and suffix. The prefix and suffix portions will be static and controlled through configuration parameters in the Cadent system. For performance reasons, this URL generation will be performed as part of the metadata ingest process into Cadent.

A typical generated URL might appear as follows –

<http://multiscreen-mpd.vod.msp.virginmedia.com/sdash/22222222-2222-2222-1111-111111222222/index.mpd/Manifest?device=TiVoAdvert>

Where –

* The purple shaded part represents the prefix,
* The green shaded part represents the SeaChange **BackOfficeID** belonging to the **OnDemandProgram** element and
* The blue shaded part represents the suffix.

## Campaign Management & Tracking

### BlackArrow Campaign Manager

Campaign management will be handled by the BlackArrow Campaign component, which includes inventory manager, creative library and reports, as well as campaign manager. These components are hosted in the BlackArrow VM premises and a European Data Centre (Amsterdam). Further details of this infrastructure, which is in place for VOD QAM, can be found in - VoD Advert Replacement (VoD Advert Replacement (High Level Architecture (HLA\_BlackArrow\_1\_0ca.doc – References)

The campaign manager is a browser based tool and is used to provide the feature set required for addressable dynamic ad insertion. Campaigns and insertion instructions are manually defined within the GUI by VM ad operations, or can be ingested/assigned programmatically (dependant on BlackArrow profile manager and associated data).

This component also provides ad hoc and scheduled reporting of all ad placement related activity.

It should be noted that these components are already deployed for and in use by the QAM VOD BlackArrow project. IP VOD advertising will make use of the same platform and functionality.

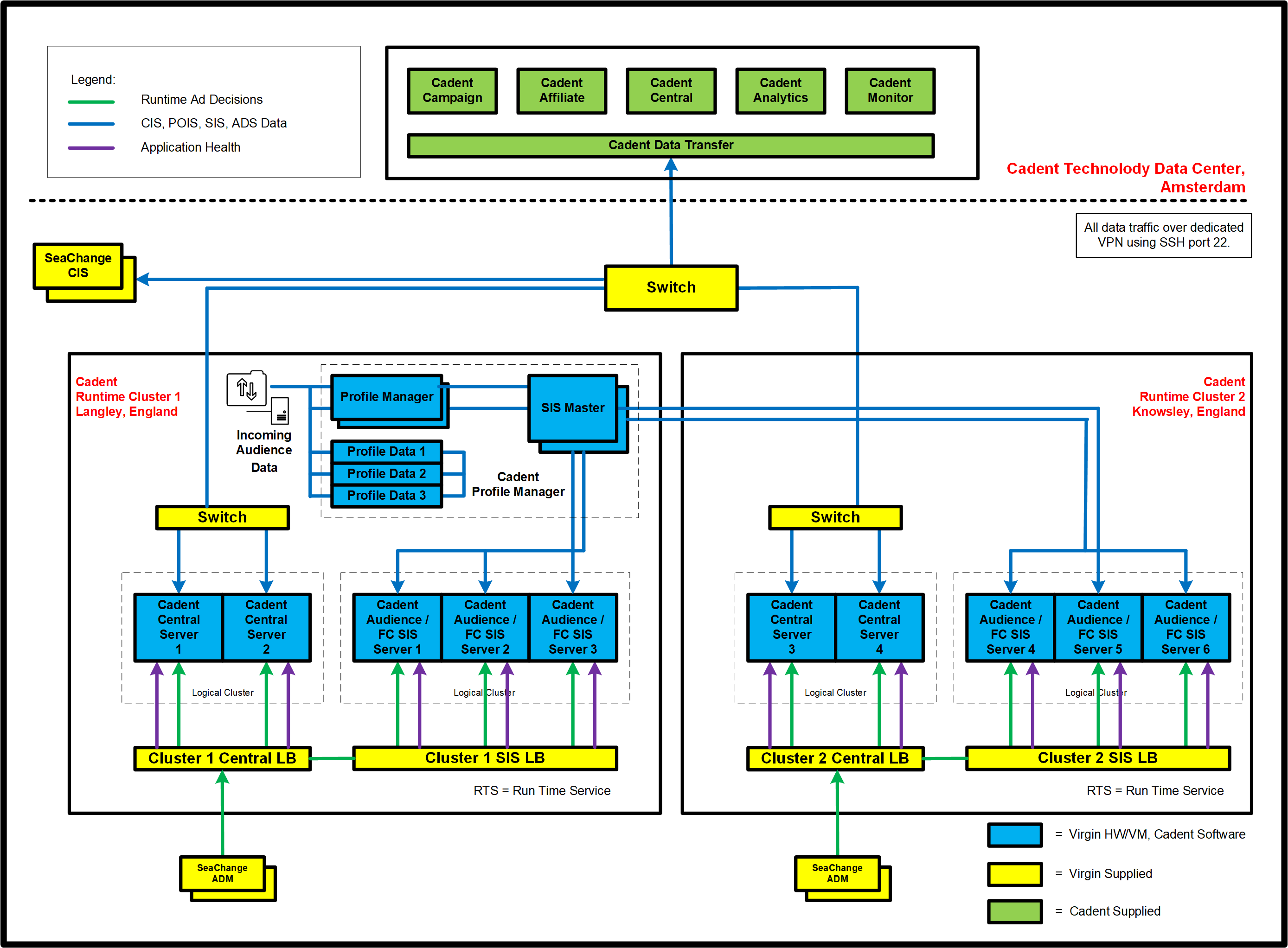


Figure ‑ - Deployment Architecture for VOD QAM

### Tracking (day1)

Tracking of ad placements and playback is critical to ensure accurate reporting of campaign performance and future campaign management. More advanced tracking metrics provide additional detail of user behaviour to further enhance the reporting and allow for more accurate monetisation of content.

The requirements for phase 1 are to provide accurate impression (placement and playback) tracking, as well as a basic fast forward tracking (where a user fast forwards through an advertisement). For later phases full quartile reporting & other functionality are required.

As part of the ad decisioning SCTE 130-3 BlackArrow will include unique IDs for each individual ad placement allowing detailed statistics to be compiled from the SCTE 130 PSN replies from the client whenever the ad is played.

An example of this placement into the manifest can be found in References- Ad FFW Reporting -v2.pdf

The Blackarrow solution approach to Ad Request Workflow (VirginVirtuousSolutionApproach\_lmc20150505.pdf) in References also explains the workflow of adding tracing URLs to the platform

#### Ad Placement Tracking ID

The SCTE 130-3 response message from Cadent to the MM allows a Tracking element to be included in the placement data. This is a unique value associated with the decision and can be used to track impressions against this decision. The tracking element is included in the MPEG-DASH manifest from the MM to the STB when the MM creates the merged manifest and thus the STB can extract this value and use it in the SCTE 130 PSN message back to Cadent whenever the ad is viewed.

Data is sent whenever the ad is viewed and thus there may be multiple instances of the message sent if the content is rewound and the ad viewed again. The PSN included information about the duration and speed of playback and thus the analytics within Cadent can use this information to infer the number of complete and unique impressions made.

### Tracking

#### Full Quartile Tracking

Full quartile reporting will be added in a future phase of the project. In order to support this, the SCTE 130 PSN and manifest extensions, detailed above, will need to be extended further, with relevant player/client support.

In order to calculate the engagement rate of the user with the advertisement, quartile reporting can be used, indicating how many users saw 25%, 50% or 75% of the ad video insertion.

As part of the Ad insertion, it will be needed to include specific tracking IDs to identify the piece of advertisement to be offered to the final user. Black Arrow Ad Insertion system will provide this ID, and needs to be delivered to the device player.

SCTE 130 PSN messages will be used for impression reporting. Events would be used to pass the following information:

1. Time when the Ad asset was entered
2. Time when the viewer has stopped the session or exit the asset
3. Trick mode operations (fastForward”,”rewind”,”play”,”pause”,”stop”.)
4. Engagement metrics (i.e. duration of Ad watched)

#### Trick-Play suppression

There is no requirement for trick play suppression on the STB. If this is required at a later stage then restrictions can be communicated from the ADS to the MM through the SCTE 130-10 Stream Restrictions Data Model specification and placed as additional data in the manifest.

### Client Applications

In order to support some of the ad insertion requirements, client applications will have to support the functionalities described above, currently this client is:-

1. TiVo Opera TV SDK

### Placement Tracking URLs flow

The player/Client will:-

1. Arris MDC will make a SCTE 130-3 request to the Cadent ADS via the Ad Router (covered in section Real Time Decisioning),
2. Cadent ADS will respond with a SCTE 130-3 placement response,
3. Arris MDC will read the tracking elements from the SCTE 130-3 response and place in equivalent structures in the MPEG-DASH manifest,
4. Client will parse the tracking elements from the ad periods,
5. Client will return the tracking IDs in the PSN messages to Cadent via the Ad Router when ads are played,
6. The Cadent Ad Router should respond with an HTTP 200 OK to signal successful receipt of the tracking message,
7. Cadent analytics will use the PSN play data and the tracking IDs to compile analyses of impressions.

To restate, this approach will need player client side changes in order to interpret the relevant tags within the manifest files, call the tracking URLs on the Cadent Ad Router and handle the response.

### Placement Status Notice (PSN)

Placement Status Notifications (PSNs) are sent back to the Ad Router to report the successful playout of a substitution ad. The timing of these is described in detail in the earlier sections of this document. However, in summary, the PSN is sent some time after the ad has played. No particular effort is required here to randomise the timing of PSN messages to Cadent since the playback of VoD content is not synchronised to any linear schedule and for all intents and purposes is effectively randomised by the user.

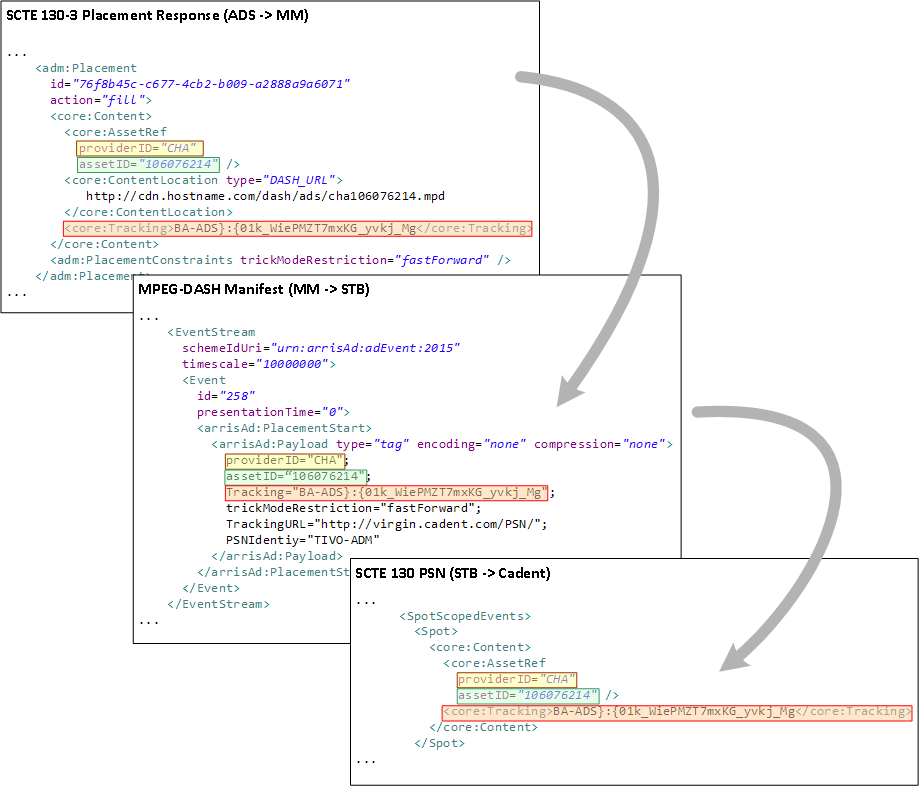
The PSN message should be sent by the STB once the ad has fully played out or once it has partially played and the playhead moved beyond the bounds of the ad. A PSN will also be required for a partially viewed ad if the playout is terminated whilst an ad is playing. This will require additional code in the player app within the STB to detect adverts in the MPEG-DASH stream, extract relevant metadata and place this in the SCTE 130 PSN.

Each PSN corresponds to a placement request/response and is used to report either the successful substitution of an ad or report an error related to this process. Where the PSN is in respect of a PlacementResponse message, the core:Tracking element from that message is included in the PSN to identify the exact decision this relates to. In other cases, such as a null response or an error, the Tracking field is not available but the opportunity ID is still supplied and can be used to identify the context for the failure.

#### Ad IDs in the MPEG-DASH Stream

In order for the STB to be able to report ad impressions back to Cadent, it needs to know the ID of the ad it has just played.

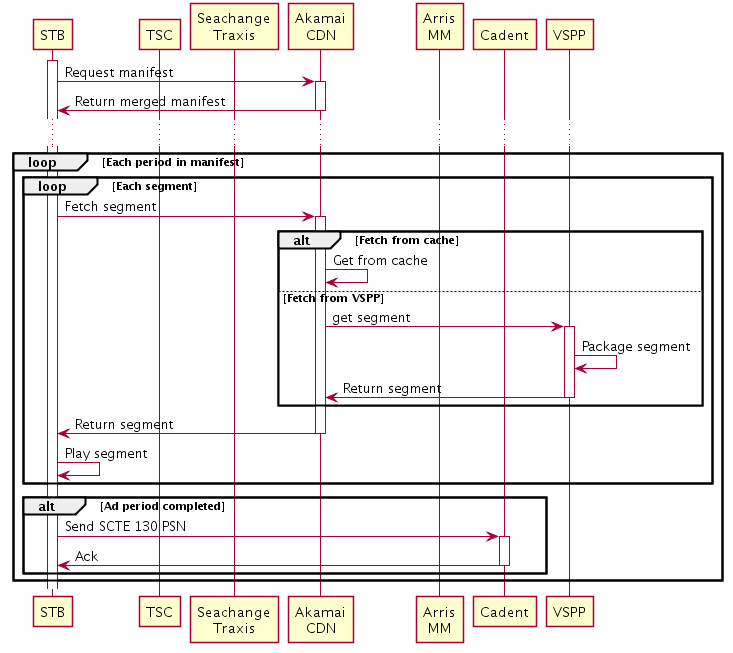
The fields required in the SCTE 130 PSN message to identify the ad are the providerID and assetID attributes in the AssetRef element of the Spot in the Playdata (see example later on). These data need to be sourced from fields within the MPEG-DASH manifest.



The MPEG-DASH manifest as produced by the MM allows for extra data to be embedded and carried in the manifest to the STB. The Arris standard mechanism for carrying this data uses the <EventStream> and <Event> tags of the MPEG-DASH standard and enhances these with definitions from an Arris-specific schema. The MM has existing code to allow data to be embedded in this structure and allows it to be defined through configuration parameters. We use this mechanism here to carry the Cadent ad tracking information to the STB.

In the short-term (phase 7a), the manifest will also carry configuration data that is required by the TiVo client to correctly manage the PSN delivery and retry process. Eventually (phase 7b) this information will be carried in the CPE Configuration Service (CCS) mechanism and the data will be removed from the manifest. When the data appears in both places, the STB should use the CCS as the authoritative source. This data is listed later in this section.

#### Sequence Diagram



Note that the above diagram glosses over the detail of manifest creation and focuses on playout. The manifest creation process is described separately in another section of this document.

The STB plays and processes the content through the use of the MPEG-DASH manifest. This file is divided into periods where each period is a part of the VoD asset or an ad. Whenever the STB completes the playout of an ad period then it should report the playout through the SCTE 130 PSN message sent to Cadent.

#### Example Messages

The following sections provide worked examples of the messages.

##### Measurement Notification

This message is delivered to the Cadent Ad Router for each placement, each time it is viewed.

The message is based on the specification <TBC SCTE130-3 ref> and on the Cadent implementation notes <TBC: cadent ref>.

The message is delivered as an HTTP POST to the endpoint specified in the CPE configuration for the placement request messages. The reply to this will include a body with a PlacementStatusAcknowledgement (PSA) described in a later section of this document.

<PlacementStatusNotification

xmlns=*"http://www.scte.org/schemas/130-3/2013/adm"*

xmlns:core=*"http://www.scte.org/schemas/130-2/2008a/core"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"*

*http://www.scte.org/schemas/130-3/2013/adm SCTE\_130-3\_2013.xsd*

*http://www.scte.org/schemas/130-2/2008a/core SCTE\_130-2\_2008a.xsd"*

version=*"1.1"*

identity=*"TIVO-ADM"*

messageId=*"07af9955-5514-4b9a-a113-3157078cb456"*>

<PlayData>

<Client>

<TerminalAddress type=*"MAC"*>BA:DA:55:CA:FE</TerminalAddress>

</Client>

<EntertainmentScopedEvents>

<SpotScopedEvents>

<Spot>

<core:Content>

<core:AssetRef

providerID=*"ad\_agency.com"*

assetID=*"pre\_roll\_ad\_asset\_1"* />

<core:Tracking>BA-ADS}:{troGgDeYRhClD53sx776aA</core:Tracking>

</core:Content>

</Spot>

<Events>

<PlacementStatusEvent

type=*"startPlacement"*

time=*"2014-04-03T11:02:33.555-08:00"*>

<Client>

<TargetCode key=*"LastKeypressSecs"*>354</TargetCode>

</Client>

<SpotNPT scale=*"1.0"*>BOS</SpotNPT>

</PlacementStatusEvent>

<PlacementStatusEvent

type=*"endPlacement"*

time=*"2014-04-03T11:03:03.555-08:00"*>

<Client>

<TargetCode key=*"LastKeypressSecs"*>367</TargetCode>

</Client>

<SpotNPT scale=*"1.0"*>EOS</SpotNPT>

</PlacementStatusEvent>

</Events>

</SpotScopedEvents>

</EntertainmentScopedEvents>

</PlayData>

</PlacementStatusNotification>

Items of note in this message are -

|  |
| --- |
| **PlacementStatusNotification@identity**  A string that identifies the client to the Cadent back-end. This value is supplied in the configuration value **ba\_psn\_identity** as described in the CPE configuration section of this document. For this implementation will use the agreed value “TIVO-ADM”. |
| **PlacementStatusNotification@messageId**  A value specified by the client that allows the client to match this message with any corresponding reply from the server. The server will return this value in its reply. |
| **PlacementStatusNotification@version**  Always “1.1”, for now at least. |
| **PlacementStatusNotification/PlayData**  Introduces the section that contains playout reports. |
| **PlacementStatusNotification/Client**  **PlacementStatusNotification/Client/TerminalAddress**  These fields provide additional information about the client STB. The Terminal address identifies the STB in terms of its MAC address.. |
| **PlacementStatusNotification/Client/TargetCode**  These fields supply supplemental information about the STB. The HWType and SWVer fields supply additional information that Cadent may use when analysing impression reports. These fields will also be useful for diagnostics and detecting errors with particular builds or hardware versions from an analysis of messages returned.  Note that the TargetCode relating to the last interaction timing field is not present here since this applies to individual replacement events. |
| **PlacementStatusNotification/PlayData/EntertainmentScopedEvents**  **…/Entertainment/core:Content**  **…/Entertainment/core:Content/core:AssetRef**  **…/Entertainment/core:Content/core:AssetId**  For the purposes of IP VoD Ad Insertion, these elements identify the primary asset that is being watched. The values for the Id and Ref are the provider ID and asset ID respectively (PID/PAID). |
| **PlacementStatusNotification/PlayData/EntertainmentScopedEvents/SpotScopedEvents**  **…/Spot/core:Content/core:AssetRef**  **…/Spot/core:Content/core:Tracking**  These elements identify the particular ad spot that was inserted and the ad which occupied it.  The tracking ID supplies a value that was originally passed to the STB from Cadent via the MM in the response to the placement request. The Cadent ad router and back end use this value to match this to an earlier placement request and can use this to correlate impressions to a specific ad decision.  The Asset providerId and assetId identify the ad that was actually inserted. |
| **PlacementStatusNotification/PlayData/EntertainmentScopedEvents/SpotScopedEvents/Events**  **…/PlacementStatusEvent**  **…/PlacementStatusEvent/SpotNPT**  These fields specify the amount of the ad that was played back. The event is described with two PlacementStatusEvent elements, one having a type=”startPlacement” and the other a type=”endPlacement”. These two elements describe the start and end of the playback. Within these, the SpotNPT elements specify the position within the ad with this either being an xsd:duration (e.g. “PT12.5S”) value or the special values “BOS” or “EOS” for the beginning or end of stream. The scale indicates the playback speed.  The time attribute gives the actual calendar time that this placement was viewed.  In a normal playout, there will be a start at “BOS” and an end at “EOS” with a scale of 1.0. However, in trick-play scenarios, the ad may have been partially played either by jumping in part way through or exiting early. Additionally, the ad may have been entered in reverse and/or played at a different speed (either negative or positive). |
| **…/PlacementStatusEvent/Client/TargetCode**  The Client file here simply carries additional client information which here takes the form of a TargetCode element indicating the time since the last user interaction with the STB at the time the impression was played out. This will be of relevance when calculating stats and determining if the event was viewed. |

##### Placement Status Acknowledgement

In response to the above message, the server will reply with an acknowledgement as per below –

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<PlacementStatusAcknowledgement

identity=*"BA-ADS"*

messageId=*"db800468-4d9f-4c7b-83f0-d9901ffc10c6"*

messageRef=*"07af9955-5514-4b9a-a113-3157078cb456"*

version=*"1.1"*

xmlns=*"http://www.scte.org/schemas/130-3/2013/adm"*

xmlns:core=*"http://www.scte.org/schemas/130-2/2008a/core"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"*

*http://www.scte.org/schemas/130-3/2013/adm SCTE\_130-3\_2013.xsd*

*http://www.scte.org/schemas/130-2/2008a/core SCTE\_130-2\_2008a.xsd "*>

<core:StatusCode class=*"0"*>

<core:Note>SUCCESS</core:Note>

</core:StatusCode>

</PlacementStatusAcknowledgement>

#### Error Handling and Retry

If PSNs cannot be delivered for any reason then the STB should attempt redelivery at a later time using a backoff/retry strategy. The redelivery is on a per-message basis with the re-delivered message being exactly the same (including message ID) as the original failed message. This will allow Cadent to identify re-delivered messages and delete duplicates in the event that multiple copies make it through to Cadent.

The STB should periodically retry delivery of these undelivered messages. The retry interval should be configurable and should be set to a value that would minimise impact on the network in the case of multiple failures. In addition, a backoff strategy would be implemented to increase the retry interval in the event that multiple retries were required.

If the STB exceeds a configurable number of retries then eventually, the undelivered message should be dropped. In these cases we would be relying of back-end diagnostics to detect a problem with the PSN delivery mechanism.

The http status code and the status code contained within the PSA will control retry behaviour in the following manner –

* Timeout waiting for response: retry,
* HTTP status something other than 200: retry,
* HTTP status = 200 and PSA status not zero: don’t retry,
* HTTP status = 200 and PSA status = “success”: don’t retry, successful delivery.

#### Monitoring

Since undelivered PSN directly relate to lost revenue (in that we have no reports of substitutions) it is important that we have some way of monitoring the situation. Given that the STB is failing to deliver, we can’t rely on sending a failure message over the network.

One approach to monitor this situation would be to periodically reconcile logs of decisions sent from the ADS with PSN received. If there were STBs with repeated large discrepancies then this could be used to flag an error.

#### Controlling Parameters

The delivery of the PSNs and the retry mechanism as implemented on the STB are controlled by a number of parameters. In phase 7a, many of these will be carried in the MPEG-DASH manifest with these moving to the CPE Configuration Service (CCS) file as part of phase 7b. When present in both places, the values in the CCS should take precedence.

The following table lists these parameters as they appear in both the manifest and the CCS. Also listed are typical values, maxima and minima. The STB may use these values for defaults and validation of supplied values.

|  |  |  |
| --- | --- | --- |
| CCS parameter | MPEG-DASH Parameter | Description |
| ba\_psn\_endpoint | PSNTrackingURL | The URL where the STB should send the PSN messages. |
| ba\_psn\_identity | PSNIdentity | The value that should appear in the @identity attribute in the PSN message. Value should be set to “**TIVO-ADM**” |
| ba\_psn\_timeout | PSNTimeoutMillis | The period after which the STB, having received no response from the URL for reporting PSNs, should consider the request failed. Typical value **2200**.  Minimum: 100; Maximum: 10000. |
| ba\_retry\_delay | PSNRetryInitialSecs | Following a failure, the number of seconds that should elapse before the STB retries the PSN delivery. Typical value **10**.  Minimum: 1; Maximum: 600. |
| lgi\_retry\_delay\_max | PSNRetryMaxSecs | The maximum interval between retries when implementing a backoff strategy. Typical value: **3600**.  Minimum: 10; Maximum: 21600. |
| ba\_max\_retry | PSNMaxRetries | The maximum number of retries for PSN delivery. Typical value: **5**.  Minimum: 0; Maximum: 10. |
| lgi\_retry\_delay\_multiplier | PSNRetryBackoffMultiplier | A multiplier that defines how the retry interval increases with each retry. Typical value: **6**.  Minimum: 1; Maximum: 100. |

## Real Time Decisioning

Real time decisioning occurs between the Arris MDC manifest manipulator and the Cadent ADS, and takes place for all IP VOD stream requests.

Initially the Cadent platform uses information (provided in the request) about the piece of content, the user/device and the time to make a decision as to whether the request is suitable for ad-insertion. If it is, Cadent then uses the same information to make further decisions relating to what advertisements to place. This is driven by campaign management and ad opportunity metadata provided with the content metadata.

The preferred option for the communications between Arris MDC & Cadent is SCTE 130-3 since this will align with the protocol being used for linear targeted ad insertion.

As part of the manifest request flow detailed in the section Manifest requesting flow, the MDC manifest manipulator receives a manifest request from the client/player, which includes details of the content being requested (PID/PAID for VOD content) and device/customer identification information. The Arris MDC uses this information to form a SCTE 130 ad decisioning/placement request to send to the Cadent ADS

### Cadent Decisioning

Cadent forms a playlist based on the provider and the structure of the content and then targets ads into this structure based on rules that work on subscriber data (audience qualifiers). The structure will typically depend on rules that revolve around the provider and the type of content. For instance, there might be a rule that defines that content from Sky with one mid roll will be structured as promo, 2 ads, bumper, part1, 2 ads, part 2, bumper, promo. On top of this, Cadent will use information it knows about the subscriber and the programme type to target ads and promos into the slots that are most likely to appeal to the customer. Customer data is held in Cadent’s Subscriber Information Service (SIS) and is accessed using the STB’s MAC address as the key (this being passed in the SCTE 130-3 request).

There might also be specific campaign targeting to specific content based on the content’s Provider ID and Provider Asset ID (PID/PAID). This information would have been defined in the campaign manager and targeted through the PID/PAID values passed in the SCTE 130-3 request.

### Fallback Decisioning

For the full structuring of breaks and targeting based on specific content, Cadent needs advance information about the content that has been ingested into the system through SeaChange and the structure of the mid-roll breaks in that content. This allows a high degree of tailoring to be performed in the ad decision.

However, there is a possibility that this information does not reach Cadent before it is published in Traxis. This can be the case for Late Exclusives where VM only receive the media a short time before the publishing date and the content needs to be rushed through.

In these cases, Cadent can still perform targeting although with a lesser degree of finesse. In these cases, Cadent won’t know the internal break structure and thus won’t be able to use mid roll breaks but can still structure ads before and after the main event. There will also be rules based on the provider (passed in the SCTE 130-3 request) that will allow Cadent to use a structure suitable for that provider’s content. Personalised targeting can still be used to a degree since Cadent will still have the identity of the requesting STB but will need to use a more generic set of campaigns since the exact type of the content will not be known.

### Ad Availability

Another operational point related to the above is that of ad availability. Cadent can only use ads that it knows are available on the system for playout and this notification mechanism is the same as that used for main content. However, in the case of ads, Cadent can’t use default rules. For main content, Cadent knows that the content exists since it is receiving a SCTE 130-3 request for it. However, for ads, it needs positive confirmation that these are on the platform before it places these into decisions. This limitation can be largely mitigated by ensuring that there is sufficient lead time on ads and content allowing them to be ingested into the platform in good time.

### Ad Decisioning Request – MDC to Cadent

The MDC queries Cadent in order to determine the structure of the content manifest that it needs to build for the end client. The MM queries the Cadent passing in the identity of the client in terms of the client’s MAC address and the identity of the VoD asset to be played in terms of the provider ID and asset ID. This is sent as an HTTP request with an XML payload to the Cadent Ad Router.

The Ad Router will then query the Cadent SIS in order to embellish the request with audience qualifiers (AQs) relevant to the client and then forward the request to the Cadent ADS.

The ADS consults its database to retrieve an ad map relevant to the requested content and then populates the template response with suitable ads based on rules around the AQs. The ad map determines the structure in terms of whether to include pre-rolls, post-rolls and mid-rolls and the length of the breaks in each of these. Where mid-rolls are used, the assumption is that the content itself will have been conditioned with breaks in the appropriate places (as originally specified by SCTE 35 markers) and that the VSPP will partition the related content manifest into periods with the breaks between periods relating to the mid-roll breaks.

The reply is a SCTE 130-3 placement response detailing the full structure of the ads and content to be played out. The response details exactly where the content fragments and ads are to be placed in relation to each other and serves as a template to the MM of how to construct the final manifest.

A full example of a SCTE 130-3 request and response are given in Appendix C; the following paragraphs summarise the salient details.

#### Request Format

The request to Cadent is made as an HTTP GET request with the body of the massage containing an XML document detailing the SCTE 130-3 request.

#### SCTE 130-3 Request Body

The request body will be structured as per the example below –

<PlacementRequest

messageId=*"6dcfe178-5884-47fc-b841-d8aa35586616"*

version=*"1.1"*

identity=*"MDC-ADM"*

system=*"MDC-ADM"*

updatesAllowed=*"false"*>

<SystemContext>

<Session>BC:64:4B:5C:56:73/0143763282</Session>

</SystemContext>

<Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

<core:URI>*<URL of the main asset as from traxis>*</core:URI>

</core:Content>

</Entertainment>

<Client>

<core:CurrentDateTime>2017-04-19T01:03:06+01:00</core:CurrentDateTime>

<TerminalAddress type=*"MAC"*>11:22:33:44:55:66</TerminalAddress>

</Client>

<PlacementOpportunity

id=*”4cfb7aca-2ff2-419d-991e-66ade817e0a3”*

serviceRegistrationRef=*”adsFullReplacement”* />

</PlacementRequest>

Within this request, there are a number of key items of data –

|  |
| --- |
| **Entertainment/core:Content/core:AssetRef@assetID**  **Entertainment/core:Content/core:AssetRef@providerID**  These two fields provide the identity of the VoD asset being viewed. Cadent will use these as a key into its database in order to select a suitable ad map for the content.  **assetID** is the provider asset ID of the VoD content.  **providerID** is the provider ID for the VoD content. |
| **Client/CurrentDateTime**  The current time and date that the request is made. |
| **Client/TerminalAddress**  **Client/TerminalAddress@Type**  These identify the STB through the use of its MAC address. The **@Type** attribute of “MAC” indicates that we’re using the MAC address of the CPE. The contents of the element are the MAC address of the STB. Note that the MAC will originally have been supplied to the STB by Traxis as a parameter on the MDC URL that it originally supplied to the STB. |
| **PlacementOpportunity@id**  Largely irrelevant for IP VoD. A unique ID that the MM can use to track request and response (although it may use other methods rendering this redundant). |
| **ServiceRegistrationRef**  Used here to indicate what type of substitution is being requested. For IP VoD this will always have the same value since we always want a fully populated ad map in return. |

### Ad Decisioning Response – Cadent to MDC

The response back to the MM from Cadent is in the form of a reply message to the original request. A successful reply will have an http status code of 200, will contain a SCTE 130-3 response with a status code of 0 and will have a body consisting of a SCTE 130-3 placement response XML document.

A full example of such a document is included in Appendix C but salient points are summarised below.

#### Response Structure

The response will contain some header information echoing the client information supplied in the request and will then have a number of <PlacementDecision> elements describing the structure of the eventual offering in terms of ads and content sections.

Each <PlacementDecision> element will describe either a section of the main VoD content or a break with one or more ads in it. Ad <PlacementDecision> elements may describe a single ad or may contain multiple ads. Both types may be mixed in the response but the intended positioning of elements will be clear from the attributes included in the <OpportunityBinding> element. It is expected that the order of the ad and main content descriptions in the returned document is the order that they will play out and that the sequence numbers belonging to them are consistent with this.

The following snippet shows the overall structure of a typical response with the detail removed –

<adm:PlacementResponse

identity=*"BA-ADS"*

system=*"BA-SYSTEM"*>

<core:StatusCode class=*"0"* />

<adm:SystemContext>

<adm:Session>BC:64:4B:5C:56:73/0143763282</adm:Session>

</adm:SystemContext>

<adm:Client>

<core:CurrentDateTime>2017-04-19T01:03:06+01:00</core:CurrentDateTime>

<adm:TerminalAddress type=*"MAC"*>11-22-33-44-55-66

</adm:TerminalAddress>

</adm:Client>

<adm:PlacementDecision

id=*"1f9be27a-f012-4ab8-8001-4f71c55fee49"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Preroll"*

opportunityNumber=*"2"*

breakOpportunitiesExpected=*"3"*

breakOpportunitySequence=*"1"*

poGroupIndex=*"1"* />

<adm:Placement/>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"aab5de3d-cf71-40ba-b134-05eec6cabefe"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Preroll"*

opportunityNumber=*"3"*

breakOpportunitiesExpected=*"3"*

breakOpportunitySequence=*"2:3"*

poGroupIndex=*"1"* />

<adm:Placement />

<adm:Placement />

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"26251b21-da98-4b4d-ac89-55af3cd750d0"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content />

</adm:Entertainment>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"6f183965-3109-4a8c-8bef-52d9ec3c5745"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"5"*

breakOpportunitiesExpected=*"6"*

breakOpportunitySequence=*"1"*

poGroupIndex=*"2"* />

<adm:Placement />

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"0b9e38a2-a4c9-436b-8838-be1273123fd4"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"6"*

breakOpportunitiesExpected=*"6"*

breakOpportunitySequence=*"2:3:4:5:6"*

poGroupIndex=*"2"* />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"cfe5f905-27cf-406b-9c59-23660918d503"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content />

</adm:Entertainment>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"f6b0b84f-1ee7-4dd8-bf34-89db6e01310f"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"11"*

breakOpportunitiesExpected=*"7"*

breakOpportunitySequence=*"1:2:3:4:5:6:7"*

poGroupIndex=*"3"* />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"173f4669-908a-459c-82bf-61b307b341ad"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content />

</adm:Entertainment>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"2c46b7dc-9f12-47f7-8bde-9a943a8fac76"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"18"*

breakOpportunitiesExpected=*"7"*

breakOpportunitySequence=*"1:2:3:4:5:6:7"*

poGroupIndex=*"4"* />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

<adm:Placement />

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"ec6896ad-d10f-425a-abc9-edd2e694eee8"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content />

</adm:Entertainment>

</adm:PlacementDecision>

</adm:PlacementResponse>

From this we see that the response is structured into a number of PlacementDecision elements with these containing either parts of the content or ad placements (note that in the above XML, the detail is omitted). The structure is further described by the following attributes on the <OpportunityBinding> element –

* opportunityType – “Preroll”, “Midroll” or “Postroll”,
* opportunityNumber – the opportunity number of the first <Placement> in the <PlacementDecision>,
* breakOpportunitiesExpected – the number of ad placements in this <PlacementDecision> element,
* breakOpportunitySequence – sequence numbers of the ads in this <PlacementDecision>,
* poGroupIndex – where in the ad map the break occurs as a numbered break.

For the example above, we see the structure is as per the following table –

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ads/Content | Op Type | Op Num | Ops Exp | Ops Seq | Grp Index | Content | Period |
| Ads | Preroll | 2 | 3 | 1 | 1 | cha/106076214 | 1 |
| Ads | Preroll | 3 | 3 | 2:3 | 1 | cha/106078364 | 2 |
|  |  |  |  |  |  | cha/106078551 | 3 |
| Content |  |  |  |  |  | cha/106002333 | 4 |
| Ads | Midroll | 5 | 6 | 1 | 2 | cha/106077761 | 5 |
| Ads | Midroll | 6 | 6 | 2:3:4:5:6 | 2 | cha/106078569 | 6 |
|  |  |  |  |  |  | cha/106078626 | 7 |
|  |  |  |  |  |  | cha/106078653 | 8 |
|  |  |  |  |  |  | cha/106078661 | 9 |
|  |  |  |  |  |  | cha/106077841 | 10 |
| Content |  |  |  |  |  | cha/106077761 | 11 |
| Ads | Midroll | 11 | 7 | 1:2:3:4:5:6:7 | 3 | cha/106078653 | 12 |
|  |  |  |  |  |  | cha/106078277 | 13 |
|  |  |  |  |  |  | cha/106077824 | 14 |
|  |  |  |  |  |  | cha/106075700 | 15 |
|  |  |  |  |  |  | cha/106078569 | 16 |
|  |  |  |  |  |  | cha/106078630 | 17 |
|  |  |  |  |  |  | cha/106078314 | 18 |
| Content |  |  |  |  |  | cha/106077761 | 19 |
| Ads | Midroll | 18 | 7 | 1:2:3:4:5:6:7 | 4 | cha/106078314 | 20 |
|  |  |  |  |  |  | cha/106077826 | 21 |
|  |  |  |  |  |  | cha/106078642 | 22 |
|  |  |  |  |  |  | cha/106078773 | 23 |
|  |  |  |  |  |  | cha/106067630 | 24 |
|  |  |  |  |  |  | cha/106078549 | 25 |
|  |  |  |  |  |  | cha/106077218 | 26 |
| Content |  |  |  |  |  | cha/106077761 | 27 |

The MM will be required to turn this into an eventual manifest for the client with 27 periods as shown in the above table.

For ads, the <PlacementDecision> and <Placement> elements are structured as follows –

<adm:PlacementDecision

id=*"1f9be27a-f012-4ab8-8001-4f71c55fee49"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Preroll"*

opportunityNumber=*"2"*

breakOpportunitiesExpected=*"3"*

breakOpportunitySequence=*"1"*

poGroupIndex=*"1"* />

<adm:Placement

id=*"76f8b45c-c677-4cb2-b009-a2888a9a6071"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106076214"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106076214.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01k\_WiePMZT7mxKG\_yvkj\_Mg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

Within this snippet, there are a number of items of note –

|  |
| --- |
| **PlacementDecision/OpportunityBinding**  For an ad type <PlacementDecision> element, this element contains a number of attributes that structure the ads and breaks in the response. Attributes are -  **opportunityType** specifies the type within the overall ad map, i.e. “Preroll”, “Midroll” or “Postroll”,  **opportunityNumber** is the ad opportunity within the overall offering. This won’t necessarily start at 1,  **breakOpportunitiesExpected** specifies the number of ads in this break (regardless of how many <PlacementDecision> elements are used to describe the break),  **breakOpportunitySequence** specifies the sequence within the <PlacementDecision> of the individual ads with each ad being specified by a <Placement> element. Where there are multiple <Placement> elements, this is a list of sequence numbers delimited by colons,  **poGroupIndex** groups <PlacementDecision> and <Placement> elements into logical breaks within the overall offering. |
| **PlacementDecision/Placement**  Specifies a single ad placement. There may be one or more of these in a <PlacementDecision>. Where there are multiple <Placement> elements in a <PlacementDecision> the order in which they appear is significant; it is expected that they will appear in the manifest in the same order. |
| **PlacementDecision/Placement/Content**  Specifies the asset to be used for this ad. The asset is described in terms of its providerID and assetID plus a URL of where to fetch the manifest from the VSPP is also given. |
| **PlacementDecision/Placement/Content/ContentLocation**  Specifies the URL of the DASH manifest that describes this ad. This will point to a manifest as served up by the VSPP. This will also be used by the client as a base URL for the fetching of content segments. |
| **PlacementDecision/Placement/Content/Tracking**  Provides a unique tracking id for this particular placement. This is opaque data that only has meaning to Cadent. This value is passed through on the DASH manifest and is eventually reported back to Cadent by the STB in the PSN message when the ad is viewed. |
| **PlacementDecision/Placement/PlacementConstraints**  This field allows additional playback constraints to be specified for the ad. Although the current implementation plan on the STB does not include playback restrictions, this value will still be passed down to the STB to allow for possible future implementation. |

For <PlacementDecision> elements relating to segments of the main VoD content, the structure is a little different-

<adm:PlacementDecision

id=*"26251b21-da98-4b4d-ac89-55af3cd750d0"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/vod/cha106002333.mpd</core:ContentLocation>

</core:Content>

<adm:EntertainmentNPT scale=*"1.0"*>0.000-575.267

</adm:EntertainmentNPT>

</adm:Entertainment>

</adm:PlacementDecision>

Here, the <PlacementDecision> specifies the ID and the MPEG DASH URL of the main VoD content plus the time range in the overall title of that segment. The structure of the returned manifest is expected be portioned into periods that correspond with the ad map described by the SCTE 130-3 placement response.

### Timeout and Retry

The fetching of the ad decision from the ADS is time critical as the MM needs the decision before it can start fetching the manifests for the ads. With this in mind, there should be a defined period that the MM should allow for a response from the ADS after which the MM should consider the operation a failure and report that back to the client. This period can include retries although a scenario involving a retry is unlikely to meet the timing requirements for this part of the operation.

The time allowed for the response to be received from the ADS should be configurable and will typically be in the order of 500ms.

### Decisioning Failure

If the call to the ADS for an ad decision should fail or time-out, the MM should prepare a manifest for the client containing just the main VoD asset. This situation should be reported in the system logs and should also generate an alarm to alert the Operations team to the situation.

## Manifest Manipulation

This section will cover the manifest manipulation (MM) and the request flow to achieve this manipulation. The actual placement opportunity data, which is gathered from Cadent, is covered in section Real Time Decisioning.

Manifest manipulation will be performed by the ARRIS Spectrum Manifest Delivery Controller (MDC). The MM will not reside in the video delivery plane, but behind the CDN (Akamai). The MM is being placed behind the CDN so as to preserve the token & cookie authority security requirements on theIPVOD STB platform.

The manifest & video delivery of the IPVOD STB platform will require a number of changes to ensure the MM can perform its functions. Also there is a requirement to “boost” the information in requesting Manifest URLs, to ensure the ability to perform targeting adverting.

The solution will support pre-roll, mid-roll and post-roll opportunities.

### Manifest manipulator components

#### Master Central Manager System (CMS) (Arris supplied)

1. Centrally manages policies and client sessions. All policy information is stored and configured on the Master CMS
2. Controls the satellites in the cluster (add/remove from cluster configuration, software upgrades, online/offline status)
3. Monitors the satellites in the cluster (monitors status, collects statistics)
4. Synchronizes with Spectrum MDC system configuration settings
5. Responds to queries from satellites for policies or user sessions
6. The CMS is deployed in 1+1 HA model (Active – Standby)
7. Communicates with the Blackarrow platform

#### Spectrum MDC Satellites (Arris supplied)

1. Receive client requests from front-end load balancer
2. Retrieve content from origin servers in the format required by the player. A client request is processed and returned by the same satellite
3. Maximum 10 satellites per cluster using an N+M model. Should a satellite fail, the traffic served by that satellite will be distributed among the remaining satellites. The distribution is directed by the load balancer and is done without the cooperation of the satellites
4. There is no communication between satellites.

#### Front-End Load Balancer

1. Distributes client requests to the pool of satellites in the cluster. The distribution method is dependent on the load balancer configuration
2. Provide increased reliability and availability. Detects and removes out-of-service satellites from its distribution list, ensuring requests are sent to functioning satellites.
3. Act as a SSL broker for the SSL requests from and to Akamai
4. At time of writing it is assumed that the F5 load balancers being used for the Virtuous project will be efficient to handle the number of requests and loads. Data Volumes and number of requests will be stated later on in this document

A full description of Arris Spectrum MDC product can be found in the Arris documents “MDC 3.3 User Guide.pdf” and “MDC340ReleaseNotes\_1.0.pdf”.

### Manifest requesting flow

The client player will acquire the URL to obtain the manifest (MPD file) from the Adrenalin platform. This URL will also contain the PID PAID and MAC address of the TiVo STB. These references will form part of the information that it will deliver to BA to assist in the targeting of advert opportunities.

The current URL signing token functionality will still be performed by Adrenalin, and the custom data in the URL will not be affected by the signing. SSL will be used on all manifest requests to ensure the customer sensitive data is kept secure. There will be Https between:-

1. Adrenalin to Client (for Leapfrog boxes)
2. Client to Akamai (for Leapfrog boxes)
3. Akamai to F5 Load balancer

Compass boxes use a different route with their traffic originating within the VM network and can thus use HTTP.

*Note – from F5 load balancers to MDC will be HTTP only.*

The Playback URL will also deliver data to the MDC on the device type and streaming protocols

An example of the URL supplied by Adrenalin follows:-

#### DASH/TiVo IP VOD

VOD :

https://multiscreen\_DASH.vod.TiVoSTB.virginmedia.com/<AdrenalinAssetID>/manifest.mpd?token=abc123&pid=someprovider.com&paid=someProviderAssetID&cpeid=01%3a23%3a45%3a67%3a89%3aAB

(note – no Live channels to STB)

where VMyy = Live channel reference (example – VM01, VM02, etc)

where cpeid is always MAC address with the colons URL-escaped to “%3a”

The Client turns around this manifest requesting URL to the CDN (Akamai), where the token will be checked, and if passed the CDN will acquire the manifest from the MDC MM. This is a change to the current solution, where currently the CDN either i) acquires manifest from the Varnish servers for linear, or ii) acquire manifest from its internal cache. The CDN must be set up to ensure all manifests are not cached. If the caching is allowed, then the advert opportunity in the manifest will be missed.

The manifest (acquired now by the MDC MM) will be acquired from the MM via the CDN and F5 load balancers. The CDN does not cache this request since the reply is unique for each STB.

*Note – the request from Spectrum MDC to the Akamai cache must be set-up in such a manner to ensure entry into Akamai is controlled by source IP address, and not require a token.*

On manifest request the MDC MM will check, with Blackarrow, whether an advert should be served in the manifest requests.

The BlackArrow system will perform the request decoration services to provide an Ad response containing ad asset references, to be included in the modified manifest file (placement decision). The MDC MM system will update the manifest files (MPD) with the reference to the Ad content stream. If Blackarrow returns that no adverts are to the served (i.e. no advert opportunity) then the manifest manipulator will return a vanilla manifest. BlackArrow & the MDC MM will communicate via the SCTE 130 XML standard (section Real Time Decisioning).

The client player will use the manifest file (returned from the MDC MM via the CDN) to identify and then collect the appropriate segments from the CDN (main assets and/or Ad content).

The following section describes the STB IP VoD flow.

### STB Playout Flow

The process starts with the STB user browsing the EPG as supplied through the TiVo TSC. Once an IP VoD title is selected for playback, the TSC supplies a content ID to the STB. The STB uses the content ID to contact the SeaChange Traxis system for a URL that it can use to initiate the playout of the content. The Traxis system also manages the purchase and subscription elements to the transaction and only supplies the URL when these are satisfied. The communication with Traxis is over HTTPS for Leapfrog boxes or HTTP for Compass; Traxis returns the required URL which additionally includes appended parameters supplying a session token, the PID and PAID of the content, the STB’s MAC address and a device parameter identifying the TiVo box.

The STB now uses the URL to contact the CDN for the manifest of the title. At this stage, the CDN doesn’t perform any caching since this would inhibit the ad insertion process. Instead, the CDN passes the request on to the Arris Manifest Manipulator (MM). The MM requests the VoD asset manifest from the VSPP, this being the manifest that describes the asset itself without the ads. This manifest will however be composed of periods and the junction between the periods will correspond to mid-roll breaks. This request to the VSPP can be satisfied by a cache hit if present either in the MM or in front of the VSPP. If not cached, the MM requests the VoD asset manifest from the VSPP which will create the manifest on the fly and return it.

At this stage, the MM will also contact the Cadent ADS to request details of ads to be inserted. The request will contain details of the STB MAC address which will allow Cadent to determine the STB’s profile plus details of the ID of the VoD asset. This latter data allows Cadent to determine the positions of the breaks for the content and the map of where ads are to be inserted (ad map). This information will have been supplied to Cadent earlier through the TVA Transformer file interface having been originally supplied through the content provider’s asset metadata contained in the ADI++ format file. In addition to the map, Cadent will use campaign data to determine which ads to insert. The final decisions are delivered back to the Arris MM in the form of a SCTE 130 placement response message.

Now that the MM knows which ads to insert, it again contacts the VSPP for theMPEG-DASH manifests of each of these ads. Again, the MM can satisfy these from its cache or request them from the VSPP if not already present with the VSPP packaging these on-the-fly.

Once the MM has the VoD asset manifest, the ad manifests and the SCTE 130 response, it can merge all of this data into a single MPEG-DASH manifest that contains the ads arranged as separate periods around the VoD asset periods. The MM also performs filtering on the ad manifests to ensure that the renditions offered are compatible with the renditions of the main content thus ensuring a consistent user experience. This single manifest is then returned to the CDN and from there to the STB.

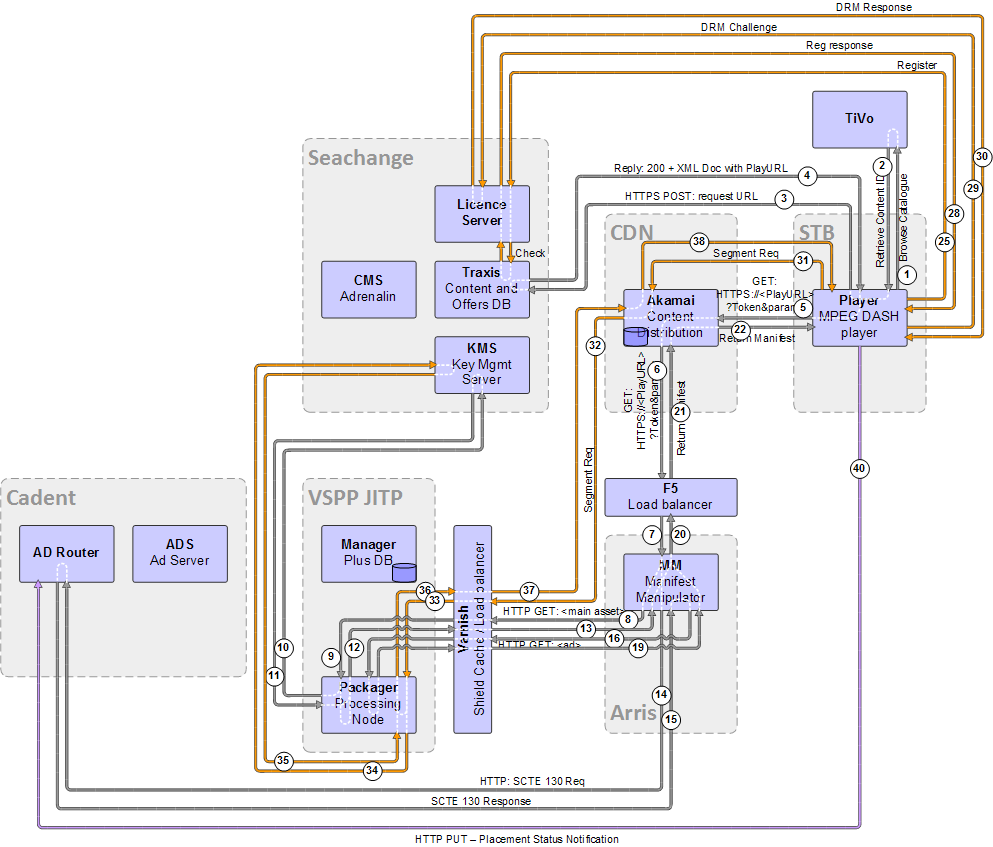
Playout on the STB uses the STB’s MPEG-DASH player libraries. The player parses the manifest to determine the content segments required and then requests these from the CDN. The exact segment selected will depend on the network conditions; the STB will choose a suitably encoded segment based on the available network bandwidth.

The CDN will either serve these directly or request them from the VSPP if not cached. Where a request to the VSPP is made, the VSPP will prepare the segment on-the-fly from its own repository by encrypting it and then returning it to the CDN.

In the manifest, segments are arranged into periods. A period may contain an advertisement or a part of the main asset. Whenever the player completes playback of a period corresponding to an ad (i.e. when it moves on to a new period), it constructs a SCTE 130 Placement Status Notification (PSN) message detailing the manner in which the ad was viewed and sends this back to Cadent. Cadent accumulates and analyses this information and uses it to adjust its ad placement algorithms and to supply impression data back to the third party ad agencies.

#### Leapfrog Setup Flow

The flows for Leapfrog and Compass STBs are slightly different to accommodate the different processing abilities of the two boxes and takes advantage of the different connectivity of these boxes to the network. The Leapfrog flow is described here first.



|  |  |
| --- | --- |
| # | Action |
| **Setup** | |
| 1 | STB browses catalogue supplied by TiVo |
| 2 | TiVo supplies ID of the offer. |
| 3 | Leapfrog box requests streaming manifest from Traxis using HTTPS. |
| 4 | Traxis replies with an XML document containing a URL that the STB can use to request the stream. The URL contains an appended security token and a parameter indicating the STB’s MAC address. This allows the STB to turn this around with no additional changes to the URL. The security token is based on a hash of the playURL plus a shared secret. |
| 5 | The Leapfrog box now uses the supplied URL to request the MPD manifest from the CDN over HTTPS. |
| 6, 7 | The CDN checks the token using the shared secret and, if valid, forwards the request to the MM via the F5 Load Balancer. |
| 8,9 | The MM strips off the superfluous parameters from the URL and then uses this URL to query the VSPP for the manifest of the main VoD content. This traverses the Varnish Shield Cache and Load Balancer to direct the request to one of the VSPP nodes. Alternatively, the MM may satisfy the request form its own cache. |
| 10, 11, 12, 13 | The VSPP either supplies the MPD from its cache or prepares it on-the-fly. This may also involve a trip to the SeaChange KMS to fetch encryption details. |
| 14 | The MM makes a SCTE 130-3 request to Cadent using the SeaChange BackOfficeID of the content and passing in the MAC that was supplied on the original request. |
| 14a | Cadent looks up the SeaChange to PID/PAID mapping and then uses this information to retrieve the ad map and calculate the ad placements required for this title. |
| 15 | Cadent returns the placement response to the MM including a URL for the later reporting of PSNs. |
| 16, 17, 18, 19 | For each of the ads in the response, the MM queries the VSPP for the MPD manifest of that ad via the Shield Cache. These might be satisfied from the MM internal cache, the Shield Cache, the VSPP internal cache or, failing that, will be prepared on-the-fly by the VSPP. This packaging step does not involve any encryption processing since ads are unencrypted. The ad manifest is returned to the MM. |
| 20, 21 | The MM prepares a combined manifest using the structure indicated in the SCTE 130-3 response and interleaves the ad manifest period into the pre, mid and post roll breaks in the main content periods. Tracking information from the SCTE 130-3 message is placed into the ad period headers. This customised manifest is then returned to the CDN via the F5 load balancer. |
| 22 | The CDN returns the manifest and appends a cookie that must be used by the client in the next request. |

|  |  |
| --- | --- |
| # | Action |
| **Playback…** | |
| 25 to 30 | The STB parses the manifest and extracts the PlayReady DRM information. Using this information, the STB makes an HTTPS request to the PlayReady licence server and negotiates the challenge/response. The Licence Server will in turn issue an authorisation check with TRAXIS. At the successful conclusion of this process the STB will have the means of decrypting the content in the segments.  If this is the first time that the STB has requested IP VoD content then the Licence Server will initiate an automatic one-time registration of the STB. |
| 31 | The STB makes an HTTP request to the CDN for a segment as specified in the manifest. The request includes the cookie supplied in the last exchange with the MM. The CDN checks the cookie and then checks its cache for the requested segment. if the request can be satisfied from cache then the data is returned immediately (step 38) otherwise the CDN proceeds to fetch from the VSPP. |
| 32 | For a cache-miss, the CDN requests the segment from the VSPP via the Varnish Shield Cache and Load Balancer. |
| 33 | If the request can be satisfied at the Shield Cache then the data is returned immediately (step 37), otherwise, the Shield Cache proceeds to request the content from the VSPP. |
| 34, 35 | For encrypted segments (i.e. main content) the VSPP Packager needs to encrypt the returned data and thus a call is needed to the SeaChange Key Management Server to fetch the PlayReady encryption keys. Ad content is delivered in the clear and thus does not require this step. |
| 36 | The VSPP packager returns the requested segment to the Shield Cache. |
| 37 | The shield cache returns the requested segment to the CDN. |
| 38 | The CDN returns the content segment in an HTTP response to the STB. The reply includes a new cookie that should be presented with the next request. |
| … | The above continues for all segments in the manifest. |

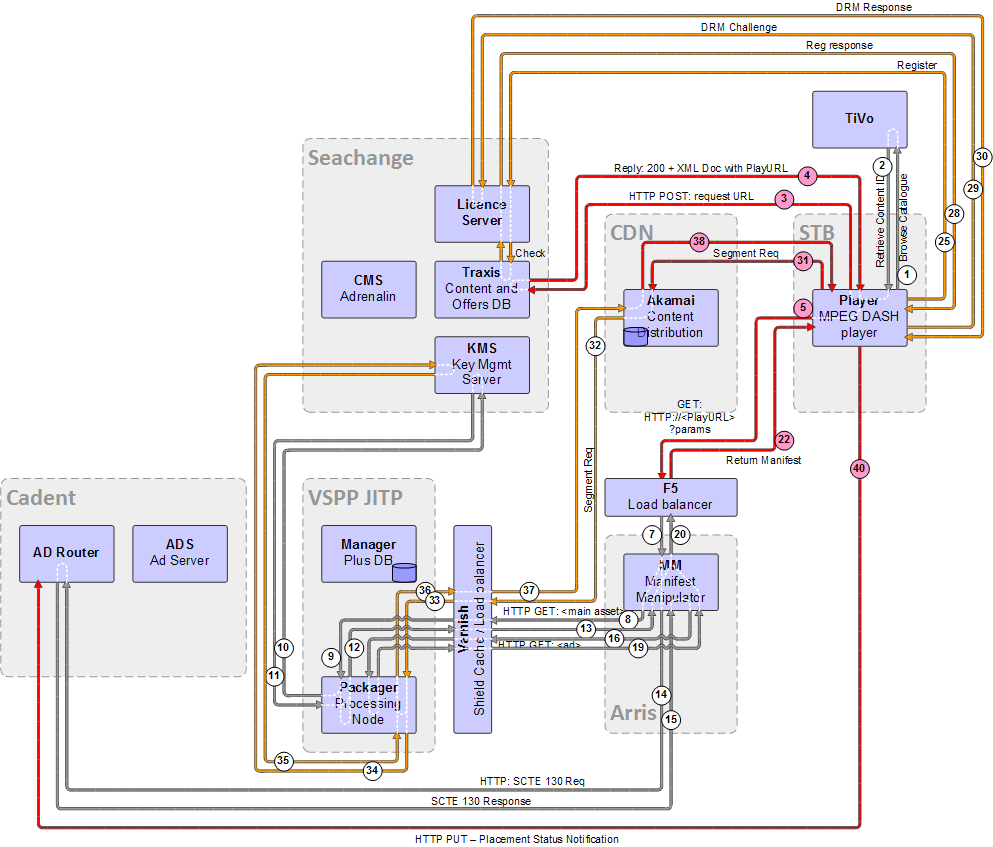
|  |  |
| --- | --- |
| # | Action |
| **Reporting** | |
| 40 | As the STB encounters and plays back adverts (in ad periods), it generates SCTE 130 Placement Status Notifications and sends these back to Cadent via the Ad Router. |

#### Compass VoD Setup Flow

The Compass flow differs slightly from the Leapfrog flow to accommodate the reduced processing power on the Compass box. These optimisations take advantage of the fact that the Compass box connects to the network in a different way to the Leapfrog box allowing security to be relaxed in some parts of the flow.

The Compass flow uses HTTP since HTTPS places a significant load on the limited processing power of the Compass box. This is acceptable from a security standpoint since the CMTS of the Compass STB is within VM’s internal network. This is then routed through the TiVo Secure Perimeter which acts as a NAT connecting the STBs to the network. This flow is different from that used by the Leapfrog box which has a different connection scheme to the network and also has the processing capacity to use HTTPS.

In the descriptions below, the differences from the Leapfrog flow are picked out in pink.



|  |  |  |
| --- | --- | --- |
| # | Action | |
| **Setup** | | |
| 1 | | STB browses catalogue supplied by TiVo |
| 2 | | TiVo supplies ID of the offer. |
| 3 | | Compass box requests streaming manifest from Traxis using HTTP. For Compass, the CMTS is inside Virgin’s network and thus plain HTTP can be used. |
| 4 | | Traxis replies with an XML document containing a URL that the STB can use to request the stream. The URL contains a parameter indicating the STB’s MAC address but omits the security token that is used for the Leapfrog request. The STB need make no changes to this URL before using it to request the content. |
| 5, 7 | | The Compass box now uses the supplied URL to request the MPD manifest from the MM via the F5 load balancer. This is an HTTP request that comes in via a NAT from the CMTS inside VM’s private network. The request is addressed directly to the F5 load balancer rather than the CDN. |
| 8, 9 | | The MM strips off the superfluous parameters from the URL and then uses this URL to query the VSPP for the manifest of the main VoD content. This traverses the Varnish Shield Cache and Load Balancer to direct the request to one of the VSPP nodes. Alternatively, the MM may satisfy this request from its own internal cache. |
| 10, 11, 12, 13 | | The VSPP either supplies the MPD from its cache or prepares it on-the-fly. This may also involve a trip to the SeaChange KMS to fetch encryption details. |
| 14 | | The MM makes a SCTE 130-3 request to Cadent using the SeaChange BackOfficeID of the content and passing in the MAC that was supplied on the original request. |
| 14a | | Cadent looks up the SeaChange to PID/PAID mapping and then uses this information to retrieve the ad map and calculate the ad placements required for this title. |
| 15 | | Cadent returns the placement response to the MM including a URL for the later reporting of PSNs. |
| 16, 17, 18, 19 | | For each of the ads in the response, the MM queries the VSPP for the MPD manifest of that ad via the Shield Cache. These might be satisfied from the MM internal cache, the Shield Cache, the VSPP internal cache or, failing that, will be prepared on-the-fly by the VSPP. This packaging step does not involve any encryption processing since ads are unencrypted. The ad manifest is returned to the MM. |
| 20, 22 | | The MM prepares a combined manifest using the structure indicated in the SCTE 130-3 response and interleaves the ad manifest period into the pre, mid and post roll breaks in the main content periods. Tracking information from the SCTE 130-3 message is placed into the ad period headers. This customised manifest is then returned to the STB via the F5 load balancer. Note that this response does not contain a security cookie as would be the case for a reply from the CDN. |

|  |  |
| --- | --- |
| # | Action |
| **Playback…** | |
| 25 to 30 | The STB parses the manifest and extracts the PlayReady DRM information. Using this information, the STB makes an HTTPS request to the PlayReady licence server and negotiates the challenge/response. The Licence Server will in turn issue an authorisation check with TRAXIS. At the successful conclusion of this process the STB will have the means of decrypting the content in the segments.  If this is the first time that the STB has requested IP VoD content then the Licence Server will initiate an automatic one-time registration of the STB. |
| 31 | The STB makes an HTTP request to the CDN for a segment as specified in the manifest. The CDN checks its cache for the requested segment. if the request can be satisfied from cache then the data is returned immediately (step 38) otherwise the CDN proceeds to fetch from the VSPP. Note that unlike the Leapfrog request, this request doesn’t carry the validation cookie. The CDN knows to accept this request without a cookie since it comes from one of a small number of whitelisted NAT nodes that serve the Compass boxes. |
| 32 | For a cache-miss, the CDN requests the segment from the VSPP via the Varnish Shield Cache and Load Balancer. |
| 33 | If the request can be satisfied at the Shield Cache then the data is returned immediately (step 37), otherwise, the Shield Cache proceeds to request the content from the VSPP. |
| 34, 35 | For encrypted segments (i.e. main content) the VSPP Packager needs to encrypt the returned data and thus a call is needed to the SeaChange Key Management Server to fetch the PlayReady encryption keys. Ad content is delivered in the clear and thus does not require this step. |
| 36 | The VSPP packager returns the requested segment to the Shield Cache. |
| 37 | The shield cache returns the requested segment to the CDN. |
| 38 | The CDN returns the content segment in an HTTP response to the STB. For the Compass box, no cookie is supplied. |
| … | The above continues for all segments in the manifest. |

|  |  |
| --- | --- |
| # | Action |
| **Reporting** | |
| 40 | As the STB encounters and plays back adverts (in ad periods), it generates SCTE 130 Placement Status Notifications and sends these back to Cadent via the Ad Router. For Compass boxes, this is sent using HTTP (rather than HTTPS). |

### Playout Flow URLs

There are a number of URLs used in the playout flow between the STB and the back end components. There are also communications between the back-end components. At each of these stages, there can be some alteration to the URL. The URL is also used to carry parameters through the system and these also get added and removed by various components. The following table summarises these URL journeys for both a Leapfrog STB and a Compass STB and shows where these transformations take place.

The different journeys begin at the point that the STB contacts Traxis with the two different architectures using HTTP and HTTPS respectively. Traxis will use the user-agent string in the request to distinguish requests from the different STBs. The user-agent strings are as follows-

* **Compass**: TiVo VOD Client/TiVo\_Client;1.0
* **Leapfrog**: TiVo VOD Client/TiVo\_Secure\_Client;1.0

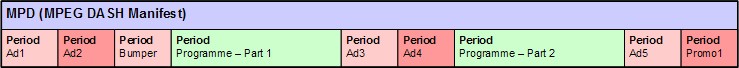
|  |  |  |
| --- | --- | --- |
| **Step** | **Leapfrog** | **Compass** |
| STB to Traxis | **Protocol:**  HTTPS  **Address:**  adrenalin.anywhere.msp.virginmedia.com:8443  **Command:**  POST TRAXIS/Web/Session/Props/Playlist?cpeid=44%3a58%3a29%3a19%3a87%3a6F HTTP/1.1  **Headers:**  Connection: Keep-Alive  Content-Length: ???  Content-Type: application / text; charset = utf – 8  Host: 10.185.64.194  User‑Agent: TiVo VOD Client/TiVo\_Secure\_Client;1.0  **Body:**  <CreateSession>  <ContentId>crid://schange.com/schange.com/TITL1234000001254728,imi:7c8e2a9e33d48daa01c7c948ef302bdd</ContentId>  </CreateSession>  **Note**:  This is an HTTPS request.  In the “msp” part of the domain name, the “p” stands for production.  The XML document returned is described in the SeaChange IF-5j specification.  This is simplified here. The request will actually pass through a load balancer which will add an X-Forwarded-For header. | **Protocol:**  HTTP  **Address:**  adrenalin.anywhere.msp.virginmedia.com  **Command:**  POST TRAXIS/Web/Session/Props/Playlist?cpeid=44%3a58%3a29%3a19%3a87%3a6F HTTP/1.1  **Headers:**  Connection: Keep-Alive  Content-Length: ???  Content-Type: application / text; charset = utf – 8  Host: 10.185.64.194  User‑Agent: TiVo VOD Client/TiVo\_Client;1.0  **Body:**  <CreateSession>  <ContentId>crid://schange.com/schange.com/TITL1234000001254728,imi:7c8e2a9e33d48daa01c7c948ef302bdd</ContentId>  </CreateSession>  **Note**:  This is an HTTP request.  No security token used for Compass.  Any of the parameters that will be sent in XML messages will be appropriately XML escaped.  The XML document returned is described in the SeaChange IF-5j specification.  This is simplified here. The request will actually pass through a load balancer which will add an X-Forwarded-For header. |
| URL returned from Traxis | **URL:**  https://multiscreen-mpd.vod.msp.virginmedia.com/sdash/11111111-1111-1111-1111-111111111111/index.mpd/Manifest  ?\_\_gda\_\_=d7s4jsfdfewwewfewfewr432  &pid=vme  &paid=TITL000000123456789  &cpeid=43%3aae%3a4f%3a2c%3a46%3aFF  &device=TiVo  **Note**:  The first parameter is the security token; the value is 32 chars long.  The pid is the provider ID and is usually 3 characters long but systems should allow for a variable length alphanumeric string here. Defined in the SeaChange IF-2 specification.  The green shaded part of this returned URL represents the part used in the formation of the hashed security token. | **URL:**  http://arris-multiscreen-mpd.vod.msp.virginmedia.com/sdash/11111111-1111-1111-1111-111111111111/index.mpd/Manifest  ?pid=vme  &paid=TITL000000123456789  &cpeid=43%3aae%3a4f%3a2c%3a46%3aFF  &device=TiVo  **Note**:  This points directly to the F5 load balancer that fronts the Arris MDC.  No security token required here as the request is within the VM network and goes to the MDC rather than the CDN. |
| STB to CDN | **Protocol**:  HTTPS  **Address**:  multiscreen‑mpd.vod.msp.virginmedia.com  **Port:**  443  **Command**:  GET sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?\_\_gda\_\_=d7s4jsfdfewwewfewfewr432&pid=vme&paid=TITL000000123456789&cpeid=43%3aae%3a4f%3a2c%3a46%3aFF&device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Content-Length: ???  Content-Type: application / text; charset = utf - 8  Host: multiscreen‑mpd.vod.msp.virginmedia.com  User‑Agent: TiVo VOD Client/TiVo\_Secure\_Client;1.0 |  |
| CDN to F5 / MM | **Protocol**:  HTTPS  **Address**:  *<configured connection to F5>*  **Command**:  GET sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?pid=vme&paid=TITL000000123456789&cpeid=43%3aae%3a4f%3a2c%3a46%3aFF&device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen‑mpd.vod.msp.virginmedia.com  User‑Agent: TiVo VOD Client/TiVo\_Secure\_Client;1.0  X-Forwarded-For: 10.185.64.194  **Note**:  The security token is no longer present, this having been dealt with by the CDN.  The address targeted is the address of the F5 load balancer fronting the Arris MDC.  Note that the full request string with the parameters is now part of the “host” header and has come off the command.  CDN has a permanent configured connection to F5 and hence doesn’t need a domain name on request. This is pre-configured. The F5 will pass through to one of a number of configured MM nodes. |  |
| F5 to MM | **Protocol**:  HTTP  **Address**:  *<Tunnelled to available MM node>*  **Command**:  GET sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?pid=vme&paid=TITL000000123456789&cpeid=43%3aae%3a4f%3a2c%3a46%3aFF&device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen‑mpd.vod.msp.virginmedia.com  User‑Agent: TiVo VOD Client/TiVo\_Secure\_Client;1.0  X-Forwarded-For: 10.185.64.194  **Note**:  F5 has a configured connection to MM cluster and hence doesn’t need a domain name on request. This is configured.  Request into MM is HTTP. The F5 has performed SSL offloading. |  |
| STB to F5 / MM |  | **Protocol**:  HTTP  **Address**:  arris‑multiscreen‑mpd.vod.msp.virginmedia.com  **Command**:  GET /sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?pid=vme&paid=TITL000000123456789&cpeid=43%3aae%3a4f%3a2c%3a46%3aFF&device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host  arris‑multiscreen‑mpd.vod.msp.virginmedia.com  User-Agent: TiVo VOD Client/TiVo\_Client;1.0  **Note**:  Request is directly to the F5 load balancer fronting the Arris MM cluster.  This will be routed to th F5 load balancer in a different way from the Leapfrog request since the Compass box request effectively comes from a NAT within the VM network and uses HTTP rather than HTTPS. |
| F5 to MM |  | **Protocol**:  HTTP  **Address**:  *<Tunnelled to available MM node>*  **Command**:  GET sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?pid=vme&paid=TITL000000123456789&cpeid=43%3aae%3a4f%3a2c%3a46%3aFF&device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen‑mpd.vod.msp.virginmedia.com  User‑Agent: TiVo VOD Client/TiVo\_Secure\_Client;1.0  X-Forwarded-For: 10.185.64.194  **Note**:  F5 has a tunnel to MM cluster and hence doesn’t need a domain name on request. This is configured.  Parameters are now present in the “Host” header and are removed from the command. |
| MM to SHC (Shield Cache)  (For Content) | **Protocol**:  HTTP  **Address**:  <?>  **Command**:  GET /sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen-mpd.vod.msp.virginmedia.com  **Note**:  The cache may be able to service the request directly in which case the request is not forwarded to the VSPP.  Arris will apply a translation rule in the “Host” field ensuring that the requests from Compass and Leapfrog appear the same. This entails stripping of the “:443” suffix from Leapfrog and removing the “arris-“ prefix for Compass. | |
| SHC to VSPP (content) | **Protocol**:  HTTP  **Address**:  vspp\_dnsvip.vod.msp.virginmedia.com  **Command**:  GET /sdash/11111111‑1111‑1111‑1111‑111111111111/index.mpd/Manifest?device=TiVo HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen-mpd.vod.msp.virginmedia.com  **Note**:  The VSPP will ignore any irrelevant parameters on the request.  The “device=TiVo” part is mandatory.  The VSPP cluster manager acts as a DNS allowing the request for **vspp\_dnsvip.vod.msp.virginmedia.com** to resolve to a list of IP addresses pointing to the nodes. | |
| MM to SHC  (ads) | **Protocol**:  HTTP  **Address**:  <?>  **Command**:  GET /sdash/22222222‑2222‑2222‑2222‑222222222222/index.mpd/Manifest?device=TiVoAdvert HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen-mpd.vod.msp.virginmedia.com  **Note**:  The cache may be able to service the request directly in which case the request is not forwarded to the VSPP.  For ads, the “&device” parameter is “TiVoAdvert”. | |
| SHC to VSPP (ads) | **Protocol**:  HTTP  **Address**:  vspp\_dnsvip.vod.msp.virginmedia.com  **Command**:  GET /sdash/22222222‑2222‑2222‑2222‑222222222222/index.mpd/Manifest?device=TiVoAdvert HTTP/1.1  **Headers**:  Connection: Keep-Alive  Host: multiscreen-mpd.vod.msp.virginmedia.com  **Note**:  The “device=TiVoAdvert” part is mandatory and signals to the VSPP that this should be supplied unencrypted. | |

Note the importance of the “&device” parameter; this provides information to the VSPP about what sort of renderings are being requested. Main content can be SD or HD and these have different renderings. However, HD and SD content are stored on the Traxis system as different offerings and hence an offering can only be SD or HD and its corresponding essences will similarly be either SD or HD. Consequently, Traxis does not have to make a distinction between SD and HD and can just supply a parameter of “TiVo”; the VSPP will return whatever renderings it has for that asset. With ads, the story is slightly different. For ads, we have a single preparation which effectively encompasses both SD and HD in terms of bitrates with all renditions having the same aspect ratio. This way, the same ad can be used in either SD or HD playback sessions. For ads, the “&device” parameter is “TiVoAdvert” although this parameter will only be present in the calls between the MM and VSPP since Traxis will not supply URLs for adverts.

### Manifest Structure

The manifest stitching process carried out by the MM will be driven by the response from Cadent with this supplying the required structure of the final manifest and the IDs and positions of the ads. The main content manifest retrieved from VSPP must be consistent with this plan, i.e. The main content manifest must be structured into periods which allow the requisite number of mid-rolls. If only pre and post roll ads are to be inserted, the main manifest should consist of a single period.

In addition to stitching the manifests together, the MM must insert tracking information (originally sourced from Cadent) into the ad periods. The eventual structure will look something like the following illustration –



Ad periods are required to carry ad tracking information that the STB can later use for impression reporting. This information is carried in <EventStream> elements within the <Period> elements representing ads. Main content periods do not need to carry tracking information.

...

<Period

id=*"1"*

duration=*"PT0H4M58.298S"*>

<BaseURL>http://ad.cdnhostname.com/DASH/0563148/</BaseURL>

<EventStream

schemeIdUri=*"urn:arrisAd:adEvent:2015"*

timescale=*"10000000"*>

<Event

id=*"258"*

presentationTime=*"0"*>

<arrisAd:PlacementStart>

<arrisAd:Payload type=*"tag"* encoding=*"none"* compression=*"none"*>

providerID="ad.provider.com";assetID=“ad0571864";Tracking="XXXXXXXXXXXXXDAI-ADS-NoReg}:{97-2696605-02A-I…";TrickModeRestriction="fastForward"; PSNTrackingURL="http://virgin.cadent.com/PSN/";PSNIdentiy="TIVO-ADM";PSNTimeoutMillis="2200";PSNRetryInitialSecs="10";PSNRetryMaxSecs="3600";PSNMaxRetries="10";PSNRetryBackoffMultiplier="6"

</arrisAd:Payload>

</arrisAd:PlacementStart>

</Event>

</EventStream>

<AdaptationSet />

<AdaptationSet />

</Period>

...

An explanation of some of the key elements in this fragment is given below –

|  |
| --- |
| **Period/EventStream**  A period may contain multiple <EventStream> elements. However, in this implementation, we are expecting a <Period> to contain a single <EventStream> with this containing a single <Event> for the tracking data. |
| **Period/EventStream/Event**  For the purposes of IP VoD ad tracking we are expecting each ad period in the manifest to contain an <EventStream> with an <Event> in it that carries the tracking information. In this implementation, we are not interested in timing-related information in the <Event>; the <Event> is related to the ad by virtue of the fact that it is placed in the same <Period> as the ad and that there is a one to one mapping between <Period> and ad. |
| **Period/EventStream/Event/arrisAd:PlacementStart**  **Period/EventStream/Event/arrisAd:PlacementStart/ArrisAd:Payload**  Although in the wider Arris ecosystem these fields have very specific meanings, for our purposes in IP VoD they simply act as a vehicle for carrying tracking information related to the ad. The payload is constructed as a number of keyword/value pairs that were originally conveyed to the MM in the SCTE 130-3 placement response message.  **providerID** is the provider ID of the ad asset. Originally supplied in the SCTE 130-3 message in the <Placement><Content><AssetRef>@providerID attribute.  **assetID** is the ad asset ID originally supplied in the SCTE 130-3 message in the <Placement><Content><AssetRef>@assetID attribute.  **Tracking** is the opaque Cadent tracking value originally carried in the SCTE 130-3 message in the <Placement><Content><Tracking> element.  **TrickModeRestriction** is the value originally carried in the SCTE 130-3 message in the <Placement><PlacementConstraints>@trickModeRestriction attribute.  **PSNTrackingURL** is the URL that the STB should use to report back Placement Status Notification (PSN) messages. The MM knows this since it was configured as part of the MDC setting for the Event payload.  **PSNIdentity, PSNTimeoutMillis, PSNRetryInitialSecs, PSNRetryMaxSecs, PSNMaxRetries, PSNRetryBackoffMultiplier** are described fully in the “PlacementStatus Notice” secion earlier in this document. These fields will move from the manifest to the Cadent CPE Configuration Service as part of phase 7b. These fields are also configured via the MDC setting for the event payload. |

A fuller example of the manifest is given in Appendix C.

### Manifest Requesting flow - Security

#### Akamai Token & Cookie, & Amendment to URL

Traffic carried over the public network using HTTPS also uses additional security features. This applies to Leapfrog boxes only as Compass boxes connect using HTTP, effectively from within the VM private network.

The security functionality installed within the VTVA OTT platform, means any manifest URL sent to the client (created from a streaming request by the client to the Video CMS) is amended with a token. This token (put on by the Video CMS) is used by the CDN (Akamai) to ensure the request for data is legit, and not a 3rd party trying to access VM files. This functionality is legally required by the Content Providers.

Therefore, when adding a manifest manipulating platform into the work flow, this security functionality has to be preserved. To ensure the security requirements are adhered to, the manifest manipulator will be placed behind the CDN. This will also ensure the cookie used for further CDN entry is preserved. Further information of this flow can be found within the HLA for VTVA

Due to the addition of customer account data (STB MAC address) amended to the URL, all connections now must be SSL. This data will be added to the end of URL proceeded by “&”. An example follows:-

https://multiscreen\_DASH.vod.AND.virginmedia.com/<AdrenalinAssetID>/manifest.mpd?tokenabc123&pid=someprovider.com&paid=someProviderAssetID&cpeid=XXXX



Figure ‑ - Token flow

|  |  |
| --- | --- |
| No. | Action |
| 1 | Client Requests Video stream |
| 2 | Video CMS returns Streaming Manifest URL (pointing to Akamai CDN), with Akamai token (URL signing) amended on URL |
| 3 | Client request manifest from CDN, using with URL & Token |
| 4 | CDN checks token, and if passes, passes manifest to manifest manipulator, within SSL |
| 5 | MM manipulators the manifest with the Adverts pointers (data collected from BA) |
| 6 | MM passes new manifest back to CDN. |
| 7 | CDN forwards new manifest back to client with cookie |
| 8 | Client reads manifest and actions pointers (via CDN), with cookie attached. |
| 9 | Client requests further data from CDN with cookie attached |

Table 6‑4 – Token / cookie Flow

*Note – within this mode the Client will always request manifests from the manifest manipulator. This means the manifest manipulator will always keep track what the client is playing. Further detail of this process can be found later on in this section.*

### Akamai NetStorage Instances

Akamai is being used as a caching layer between the STB and the other back-end components.

For Leapfrog STB manifest requests, the CDN is required to validate the security token and then forward the request to the MM. (For Compass the STB talks to the MM directly). The CDN cannot cache this request since each reply will be different and tailored to the specific STB and Cadent campaign targets.

For requests for DASH segments, the CDN acts as a cache between the STB and the VSPP JITP. Requests will be made to the CDN and the CDN must either satisfy these from cache or request the required content from the VSPP.

VSPP instances for caching content will be mapped across environments as follows –

|  |  |
| --- | --- |
| Environment – STB Reluanch | VSPP Instance |
| Dev Adrenalin | <TBC> |
| Functional Adrenalin | <TBC> |
| PreProduction Adrenalin | <TBC> |
| Production Adrenalin | <TBC> |
| DR Adrenalin | <TBC> |

Table ‑ - STB Environments Adreanlin to WFS

#### Akamai Balancing

**Assumption:** Advertising Content Roadmap does not impinge on the following Akamai thresholds.

Uploading to a single NetStorage has the following limitations:

* Maximum recommended storage for a single NetStorage instance is 10 TB

There will be three dedicated Akamai instances used to host advert content:

1. STB VOD DASH Content (SD + HD)

### Customer MAC Cookie within TLS

As stated above, Customer information (MAC address of the TiVo STB) must not be allowed outside Virgin Media internal systems in an unencrypted format. Therefore the customer related data (MAC address) that the Adrenalin platform appends to the URL must be encrypted. To ensure this data is encrypted, a SSL connection will be used between Client & Adrenalin, and Client to CDN. Adrenalin is not supporting SSL directly. SSL is terminated on Load balances in front of Adrenalin.

The above is used for Leapfrog boxes since these connect over public networks. For Compass boxes, the network traffic originates within the VM network and thus allows HTTP to be used for certain operations.

## Adaptive Streaming

Adaptive streaming works via client playback of a sequence of HTTP-requested short video file segments or ‘chunks.’ Adaptive HTTP streaming allows the client to request chunks of the same content, encoded at different bitrates and resolutions (called profiles), so that, as network bandwidth changes, the client can download the best possible quality chunk at any point in time.

There are four popular protocols today – Apple HTTP Live Streaming (HLS), Microsoft Smooth Streaming, Adobe HDS and MPEG DASH

The ABR format used for IP VoD to TiVo STBs will be MPEG-DASH.

The Profiles to be used for the TiVo STB devices can be found in - Appendix B

### MPEG-DASH

MPEG-DASH works nearly identically to the other adaptive streaming protocols. Available stream content is presented to the player in a manifest file called Media Presentation Description (MPD) file, which is in XML format. The MPD is analogous to an HLS m3u8 file, a Smooth Streaming Manifest file or an HDS f4m file.

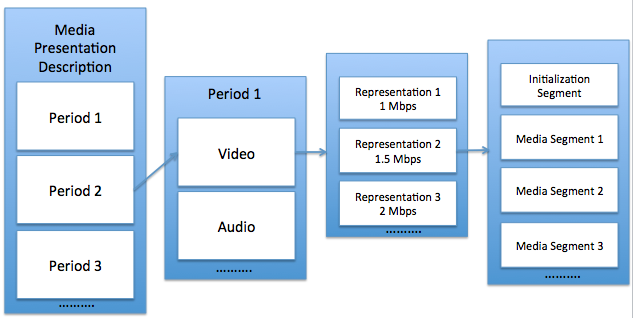


Figure ‑ - DASH structure

Figure 6‑6 - MPEG-DASH MPD file

### Ad insertion in MPEG-DASH

Ad insertion in MPEG-Dash is supported through the use of **Multiple Periods**; this is how it manages discontinuities between main asset and Ad timelines. All players must support multiple periods.

Content URLs in a period are independent; alternate content can exist on completely different servers/hosts from the original asset. Typically the original MPD is “conditioned” to contain information that shows where an alternate content event can occur.

A MPEG-DASH MPD consists of one or multiple periods. A period in DASH is simply a program interval containing the content’s temporal domain. Beyond the timeline the period identifies discontinuities, and also allows for different URL's to be used when the base format is template based.

An actual period really just consists of one or multiple adaptation sets, which provides the information about the various encoded alternatives (e.g. Bitrates and Resolutions) available to the player.

Each adaptation set usually includes multiple representations. Representations can be thought of encoded alternatives of the same media component, varying from other representations by bitrate, resolution, number of channels or other characteristics. Each representation consists of one or multiple segments, which are the media steam chunks in the temporal sequence.

The solution now is to deploy players that support multiple periods and allow Ad Insertion or Alternate Content Switching through simple MPD Manipulation.

Figures below gives i) a very high-level view of the periods and representations in MPEG DASH, ii) example of transitions between the representations (note – this example doesn’t mean switching between content and ads would force a representation change)

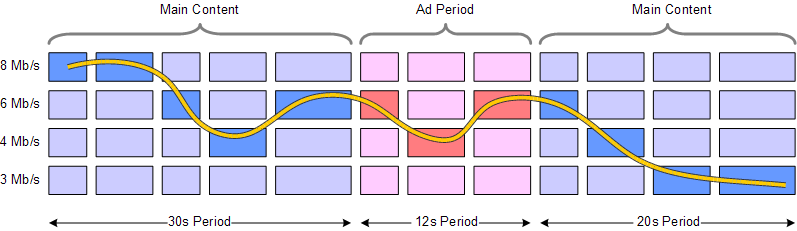


Figure 6‑7 - MPD manipulation

### Difference Between Live and VoD Streams in MPEG-DASH

Live and VOD streams have very similar workflows in HTTP streaming. The only difference is that in the live case, the playlist is modified as new chunks become available and thus the client repeatedly downloads it in order to have the most up-to-date version. In the VOD case, the playlist is created just once with all the references to every chunk included. The mechanisms used to deliver ads in live streams can be used to deliver pre-roll, mid-roll and post-roll Ads in a VOD framework.

For this implementation we will only be using VoD type streams.

### Server Side Insertion Mode – Virgin Media’s Chosen Method

Following this approach, Ad insertion logic (including personalization decisions) are made by the server components.

Clients in this model are not aware of the advert insertion and therefore there is no need to communicate specific advert details to the client (e.g. intended location of the insertion point, insertion rules etc.). This model is simple to implement

In the case of MPEG-DASH server-based model, all Ad-related information is expressed via MPD and segments, and Ad decisions are triggered by client requests for MPDs and for resources described in them (segments, remote periods).

Server-based model is inherently MPD-centric – all data needed to trigger ad decision is concentrated in the MPD. In case where ad break location (i.e., its start time) is unknown at the MPD generation time, it is necessary to rely on MPD update functionality.

## Network & Server infrastructure

### MDC Cluster components

The Spectrum MDC Cluster will have three parts to its components, being

1. MDC Master Central Manager System (CMS) (Arris supplied)
2. MDC Satellites (Arris supplied)
3. Front-End Load Balancer

The Spectrum MDC will be built as a cluster serving IP VoD. The cluster will be built as follows:-

1. 2x primary & 1x redundant – VOD cluster MDC satellites
2. 1x main & 1x redundant - Central Manager System (CMS)
3. 1x main & 1x redundant – Front-End load balancers

The MDC platform will be built at both Langley & Knowsley to ensure disaster recovery operation. For the MDC clusters, failover is effectively automatic since the clusters are being run in an active/active configuration.

#### MDC Cluster Component Scaling Considerations

Determination of the number of MDC cluster components required for the VOD deployments are based on the following:

The specifications of the servers on which each Cluster Manager or Satellite instance shall be loaded are given in References document - Ref15 VOD Advertising Manifest Manipulator Infrastructure,

VOD Sessions:

* 32k concurrent MPEG-DASH sessions
* Average length of each VOD session is 20 minutes
* Length of each DASH media segment is around 4 seconds

### Blackarrow & SeaChange components

Blackarrow platform is already live and was delivered by the VOD QAM advert insertion program. The infrastructure is already scaled to support the anticipated load. The Spectrum MDC platform will talk to the Blackarrow runtime servers within Langley and Knowsley. Documentation of this infrastructure and connectivity can be found within VoD Advert Replacement - VoD Advert Replacement High Level Architecture (HLA\_BlackArrow\_1\_0ca.doc)- in References

The SeaChange Adrenalin platform are already Live (or will be live). Documentation of this infrastructure and connectivity can be found in *1008264: STB VOD Re-launch – in* References

The connections between the systems are as follows:-

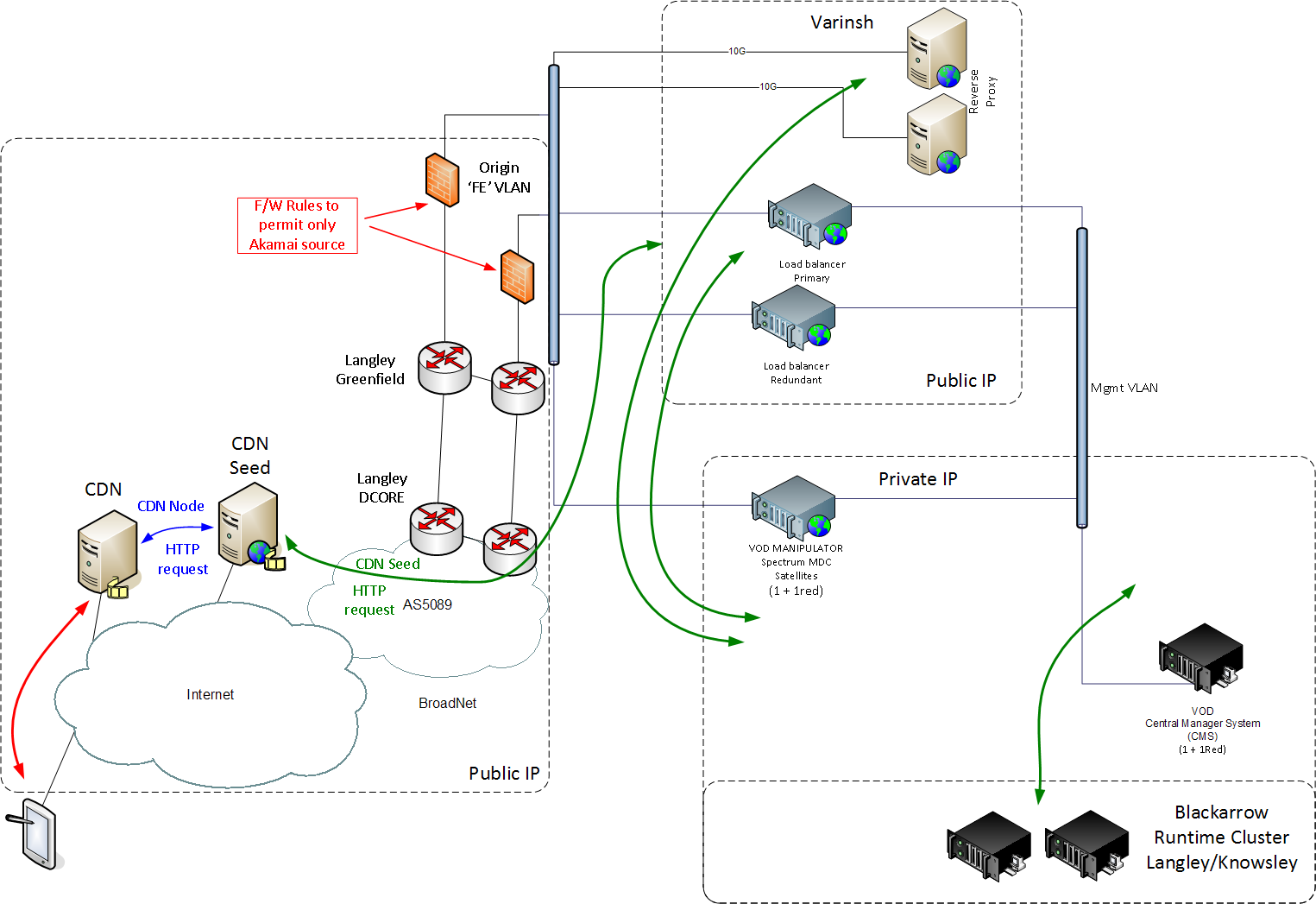


Figure ‑ - MDC connections



Figure ‑ - Arris, BA & SeaChange connections

## Platform Monitoring

All components need to be integrated into Virgin Media’s operational monitoring framework. This is implemented on top of the Nagios and Netcool products. Specifics listed in the following sections.

### Seachange Platforms

This is existing infrastructure that is being extended for IP VoD. Platform monitoring is already in place.

### Ericsson VSPP

Monitoring requirements covered in the ingest specification. [Technology\_HLA\_IP VoD Ingest(VSPP)\_VMUK\_v1\_2.docx]

### Cadent Platforms

This is existing infrastructure that is being extended for IP VoD. Platform monitoring is already in place.

### Arris MDC

Platform monitoring will need to integrate into Nagios and Netcool. At a minimum, the following should be monitored –

* All hardware or VMs should be monitored for general health with active polling from the monitoring system to detect hardware failures,
* Health of all network interfaces to be monitored for general health,
* Key services to be monitored for health,
* SNMP traps to be configured to report alarm conditions to Nagios.

The main monitoring is from LB to satellites for the /rest/healthcheck (remove the node from load distribution if no return of HTTP 200 OK). Each MDC instances (Satellites & CMS servers provide /rest interface with healthcheck and other stats). They can also be configured to return the SNMP traps.

### CDN

This is existing infrastructure that is being extended for IP VoD. Platform monitoring is already in place.

### Varnish Cache and Load Balancer

Platform monitoring will need to integrate into Nagios and Netcool. At a minimum, the following should be monitored –

* All hardware or VMs should be monitored for general health with active polling from the monitoring system to detect hardware failures,
* Health of all network interfaces to be monitored for general health,
* Key services to be monitored for health,
* SNMP traps to be configured to report alarm conditions to Nagios.

### F5 Load Balancers

This is existing infrastructure that is being extended for IP VoD. Platform monitoring is already in place. Where new instances are deployed, these should be monitored in the same way as the existing instances.

# Non Functional Aspects

## Volumetrics & Capacity

### Non Functional Requirements

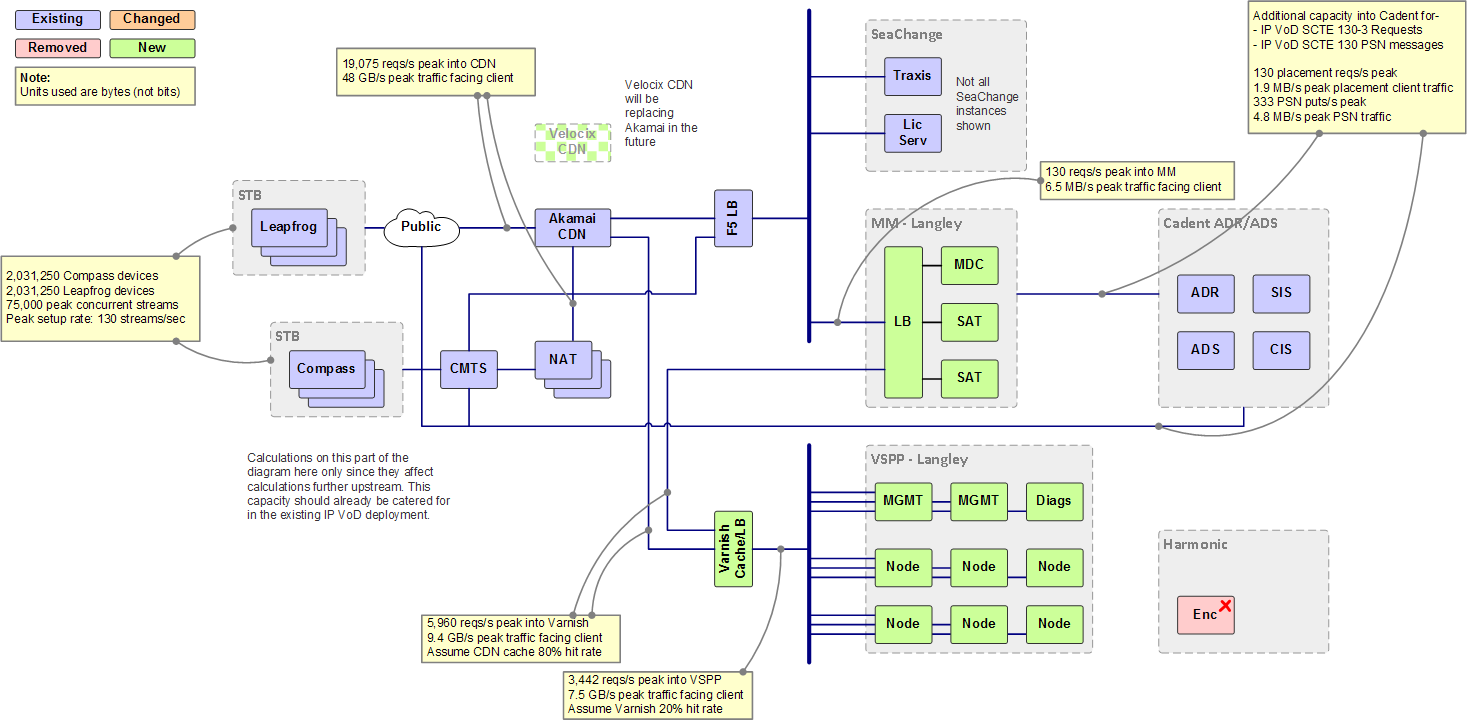
The volume calculations presented in the following sections are driven by the non-functional requirements documentation which is summarised in the ***20170509 TiVo7a – IPVOD consolidated NFRs v0.1.pptx*** slide pack, this being an amalgamation of requirements contained in other documents. Specifically, the requirements (paraphrased) driving these calculations are-

|  |  |  |
| --- | --- | --- |
| Req | Description | Notes |
| NFR.1 | The solution to support up to 4,062,500 total TiVo devices. | For the calculations here, we are looking at a 50/50 split between Compass and Leapfrog boxes, this being the expected position at the end of 2018. |
| NFR.2 | The solution to support setting up 130 streams per second. Based on supporting around 32k peak concurrent IP streams being set up over a 4 minute period. | This requirement relates to the speed at which the streams are set up rather than the maximum number of streams overall. |
| NFR.3 | .. setup rate of 130 streams/sec … | Derived from the above |
| NFR.8 | The solution to support up to 20,000 total live hours of HD content on platform.  It’s noted that some content will be SD, but planning for all HD | Calculations based entirely on HD. |
| NFR.13 | The solution to support an average SD bitrate of profile mix of 2,150 kbs. |  |
| NFR.14 | The solution to support an average HD bitrate of profile mix of 5,000 kbs. |  |
| NFR.19 | The solution to support 75,000 peak concurrent VOD streams being played out at any one time |  |

### Data Volumes for Playout

The calculations are based on the above NFRs. For bitrates, we use the average expected bitrate rather than the absolute maximum as this has a large impact on CDN capacity and it is fair to assume that not all streams will be receiving the maximum rate of ABR. However, as per NFR.8 we do assume that all streams will be HD. Other calculations are based on peak values with the key driver being the setup rate as defined in NFR.2.

The following diagram summarises the playout chain and shows key metrics. More detailed calculations are shown in the following sections.



#### Calculation Parameters

The following parameters are used to drive the calculations.



#### CDN Calculations

Note that these are here for information only in that they are used to drive later calculations. The existing IP VoD project has already reserved CDN capacity.



#### Varnish Shield Cache Calculations

The calculations here are for manifest and segment requests. Note that requests have already passed through the CDN which will have served a percentage of requests from its own cache. The figures here are for what passes through.



#### VSPP Calculations

Traffic hitting VSPP will have already been reduced by the shield cache and the CDN. These calculations reflect that.



#### MM Calculations

The data volumes here are small since the MM is dealing only with manifests. However, the number of requests is significant since there is a lot of processing required to generate the manifest.



#### Cadent Calculations

The calculations are for the additional capacity required by IP VoD. It should be noted that, over time, the load here will be partially mitigated by the corresponding decreased load from QAM VoD.



# Security Considerations

### Hardware Security

The new components to the virgin media infrastructure are the Arris Spectrum MDC CMSs, MDC Satellites & Load balancers. These units will need the normal hardening and antivirus measures.

The new Ericsson VSPP infrastructure will operate solely within Virgin’s networks and thus should need no special measures.

### Application Security

Due to the nature of targeting adverting, any request for a manifest (hence routed to MDC), must identify itself with an account id of some sort. For the TiVo STB this will be the MAC address. This data will be attached to the requesting URL (amended at the end), where the request will be using SSL to protect the data

### Network Security

Traffic to the MDC can come from the Akamai CDN via the F5 load balancer or directly from the CMTS (for Compass boxes) via the F5 load balancer. In the case the Akamai route, protection is provided by the existing Akamai infrastructure and Virgin Media firewalls. For the CMTS route, the traffic effectively originates within VM’s network under the control of the application on the STB. This route is already protected and thus no additional measures are required.

Requests for content segments also come via the CDN and thus are subject to the same protections. Again, traffic for Compass boxes traverses a different route, effectively originating in VM’s network at the CMTS. Special routing rules allow this traffic to route to the CDN.

STBs will now be making placement status notifications directly into the Cadent Ad Router. This will thus require a public interface and will need to be protected by the appropriate firewalls.

Other new network elements are all within VMs internal network.

## Reporting

### Business reporting

Spectrum MDC’s web User Interface provides a dashboard with an overview of manifest-level statistics, such as active devices, total streams by type, and manifest requests. Session detail records (SDRs) are recorded for each client session and include information such as maximum and minimum bitrate. Session information can also be viewed for active client sessions while they occur.

Further details can be found in – MDC 2.7 HA User Guide.pdf – in References

| **REQ ID** | **Requirement** | **Solution Proposed** |
| --- | --- | --- |
|  | TBC |  |

### Application reporting

{{This section details the reporting requirements at the time of release and the proposal in terms of solution where this is not indicated therein}]

| **REQ ID** | **Requirement** | **Solution Proposed** |
| --- | --- | --- |
|  | TBC |  |

### Error logging

{{This section details the reporting requirements at the time of release and the proposal in terms of solution where this is not indicated therein}}

| **REQ ID** | **Requirement** | **Solution Proposed** |
| --- | --- | --- |
|  | TBC |  |

## Operational Continuity

### Redundancy of MDC/CMS

Redundancy of the MDS & CMS will be explained in the Arris design documentation, though the following in redundancy for MDC/CMS systems have been commissioned

1. MDC
   1. VOD – 1 Primary + 1 redundant
2. CMS
   1. VOD – 1 primary + 1 redundant

## Disaster Recovery

As in NGTV & VTVA, the disaster recovery site is within Knowsley. The disaster recovery system is built with exactly the same nodes as the primary site.

The disaster recovery platform, as with NGTV & VTVA, is not an automatic fail over, controlled by manual intervention

## Support Model

This is define by Partner Operations, please refer to this department for further information

## Licensing

The MDS/CMS software is a license base model, though its one off license.

## Operating System / Software Versions

Both the MDC & CMS is run on Centos Linux. The Ericsson VSPP system runs on Linux OS (SD in non-specific about distributions or versions).

## Requirements Traceability Audit

{{Insert a copy of the Requirements Traceability Matrix here (as produced by the Business Analysis team), and for each requirement in the matrix cross-reference back to the section(s) within the Solution Design which address that requirement. This will demonstrate that the Solution Design meets the stakeholders’ needs. Highlight requirements for which the Solution Design is Partially Compliant or Not Compliant. Clearly mark requirements which are not in scope for Technology}}

## Impact Assessments

The following groups/individuals have been identified as needing to produce an Impact assessment (IA) as a response to this document. Further details on expected deliverables within the IA are provided below. Note, this list is not exhaustive and other groups may also need to provide an Impact Assessment, as requested by the SI

## OSS Development

Adding Arris MDC & CMS into Nagios OSS platform, (though not limited too)

## TV Engineering

Addressing changes for Akamai CDN, SeaChange Adrenalin, ADI input of metadata for Adverts interface for populating data in Blackarrow, Addition of SCTC35 marker detection in WFS, Work flow of Adverts through WFS, (though not limited to)

## TiVo and its clients

Adding of advert tracking functionality into all clients, multi period (MPD), Discontinuities (HLS), (though not limited too).

## Web in-house development team

The desktop web application to support advert tracking functionality (though not limited too).

## Network Engineering / Design

Network design to support connectivity between all stated components, additional of MDC & CMS, load through-put on F5 load balancers, Firewall configurations (though not limited too)

## Infrastructure Design

Design document covering infrastructure changes required to support the new connectivity required to add Arris MDC & CMS (though not limited too)

## Virgin Media Engineering Security

Confirmation that passing the MAC address or vmTVAccountID can be passed amended to the URL, though in SSL (though not limited too).

# Appendix A

## Manifest Manipulation Constraints

Manifest manipulation is more complex than it at first may seem.

Manifest manipulation is most suited for protocols based on the MPEG-2 TS specification.

* Apple HLS
* MPEG-DASH (not MPEG-DASH ISO Base media file format)

Dynamic manifest manipulation for MPEG-DASH ISO BMFF as they are problematic to support:

* Discontinuities in the timeline indicate out of order segment appending.
* Not supported today in Smooth Streaming, HDS and mpeg DASH-MP4 players
* Some implementations of HLS players on Android do not support discontinuity
* Manifest manipulation does not support the packaging or repackaging of any content or content fragments
* Manifest manipulation cannot encrypt or otherwise secure content, however it can swap out the DRM key URL’s in the manifest to tailor a syndication experience
* Session Management capabilities in dynamic manifest manipulation are limited to the manifest level only:
* Cannot affect fragment size / duration and the associated quality controls
* Cannot provide analytics related to content fragment delivery; Statistics are restricted to manifest requests

# Appendix B



Table ‑ - Streaming profiles

# Appendix C – MPEG DASH Manifest Example

The following manifest example shows the structure that we might expect the MM to send to the STB (via the CDN). The example shows how the manifest is structured into periods with each period representing either an ad or a section of the content.

The following example shows a manifest with three periods representing a simple playout of a VoD title with a single pre-roll ad, a single chapter of content and a single post-roll ad.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<MPD

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"urn:mpeg:dash:schema:mpd:2011"*

xmlns:arrisAd=*"arris-dash-advertising.xsd"*

xsi:schemaLocation=*"urn:mpeg:dash:schema:mpd:2011 DASH-MPD-edition2.xsd"*

type=*"static"*

profiles=*"urn:mpeg:dash:profile:isoff-live:2011"*

minBufferTime=*"PT2.002S"*

mediaPresentationDuration=*"PT0H14M54.894S"*

>

<!-- Preroll with segment timeline for video and audio -->

<Period

id=*"1"*

duration=*"PT0H4M58.298S"*>

<BaseURL>http://ad.cdnhostname.com/DASH/0563148/</BaseURL>

<EventStream

schemeIdUri=*"urn:arrisAd:adEvent:2015"*

timescale=*"10000000"*>

<Event

id=*"258"*

presentationTime=*"0"*>

<arrisAd:PlacementStart>

<arrisAd:Payload type=*"tag"* encoding=*"none"* compression=*"none"*> providerID="ad.provider.com";assetID=“ad0571864";Tracking="XXXXXXXXXXXXXDAI-ADS-NoReg}:{97-2696605-02A-I…";trickModeRestriction="fastForward"; PSNTrackingURL="https://virgin.cadent.com/PSN/";PSNIdentity="TIVO-ADM";PSNTimeoutMillis="2200";PSNRetryInitialSecs="10";PSNRetryMaxSecs="3600";PSNMaxRetries="10";PSNRetryBackoffMultiplier="6"

</arrisAd:Payload>

</arrisAd:PlacementStart>

</Event>

</EventStream>

<AdaptationSet

id=*"1"*

contentType=*"video"*

group=*"1"*

par=*"16:9"*

maxWidth=*"1280"*

maxHeight=*"720"*

maxFrameRate=*"989/33"*

segmentAlignment=*"true"*

bitstreamSwitching=*"true"*

mimeType=*"video/mp4"*

startWithSAP=*"1"*

maxPlayoutRate=*"29.969999999999999"*>

<SegmentTemplate

initialization=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(video=Init)"*

timescale=*"10000000"*

media=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(video=$Time$)"*

startNumber=*"0"*>

<SegmentTimeline>

<S

t=*"0"*

d=*"20020000"*

r=*"148"* />

</SegmentTimeline>

</SegmentTemplate>

<Representation

id=*"video.1.L1"*

width=*"480"*

height=*"270"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.42c01f"*

bandwidth=*"600029"* />

<Representation

id=*"video.1.L2"*

width=*"640"*

height=*"360"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"1400033"* />

<Representation

id=*"video.1.L3"*

width=*"960"*

height=*"540"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"2400032"* />

<Representation

id=*"video.1.L4"*

width=*"1280"*

height=*"720"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"3400032"* />

<Representation

id=*"video.1.L5"*

width=*"1280"*

height=*"720"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.640029"*

bandwidth=*"4400037"* />

</AdaptationSet>

<AdaptationSet

id=*"2"*

contentType=*"audio"*

group=*"2"*

lang=*"und"*

segmentAlignment=*"true"*

bitstreamSwitching=*"true"*

mimeType=*"audio/mp4"*>

<AudioChannelConfiguration

schemeIdUri=*"urn:mpeg:dash:23003:3:audio\_channel\_configuration:2011"*

value=*"2"* />

<SegmentTemplate

initialization=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(audio\_482\_und=Init)"*

timescale=*"22050"*

media=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(audio\_482\_und=$Time$)"*

startNumber=*"0"*>

<SegmentTimeline>

<S t=*"0"* d=*"45056"* />

<S t=*"45056"* d=*"44032"* r=*"7"* />

<S t=*"397312"* d=*"45056"* />

<S t=*"442368"* d=*"44032"* r=*"7"* />

<S t=*"794624"* d=*"45056"* />

<S t=*"839680"* d=*"44032"* r=*"7"* />

<S t=*"1191936"* d=*"45056"* />

<S t=*"1236992"* d=*"44032"* r=*"7"* />

<S t=*"1589248"* d=*"45056"* />

<S t=*"1634304"* d=*"44032"* r=*"7"* />

<S t=*"1986560"* d=*"45056"* />

<S t=*"2031616"* d=*"44032"* r=*"7"* />

<S t=*"2383872"* d=*"45056"* />

<S t=*"2428928"* d=*"44032"* r=*"7"* />

<S t=*"2781184"* d=*"45056"* />

<S t=*"2826240"* d=*"44032"* r=*"8"* />

<S t=*"3222528"* d=*"45056"* />

<S t=*"3267584"* d=*"44032"* r=*"7"* />

<S t=*"3619840"* d=*"45056"* />

<S t=*"3664896"* d=*"44032"* r=*"7"* />

<S t=*"4017152"* d=*"45056"* />

<S t=*"4062208"* d=*"44032"* r=*"7"* />

<S t=*"4414464"* d=*"45056"* />

<S t=*"4459520"* d=*"44032"* r=*"7"* />

<S t=*"4811776"* d=*"45056"* />

<S t=*"4856832"* d=*"44032"* r=*"7"* />

<S t=*"5209088"* d=*"45056"* />

<S t=*"5254144"* d=*"44032"* r=*"7"* />

<S t=*"5606400"* d=*"45056"* />

<S t=*"5651456"* d=*"44032"* r=*"7"* />

<S t=*"6003712"* d=*"45056"* />

<S t=*"6048768"* d=*"44032"* r=*"8"* />

<S t=*"6445056"* d=*"45056"* />

<S t=*"6490112"* d=*"44032"* r=*"1"* />

</SegmentTimeline>

</SegmentTemplate>

<Representation

id=*"audio\_482\_und.2.L1"*

audioSamplingRate=*"22050"*

codecs=*"mp4a.40.2"*

bandwidth=*"64000"* />

</AdaptationSet>

</Period>

<!-- Entertainment with segment timeline for video and audio -->

<Period

id=*"2"*

duration=*"PT0H4M58.298S"*>

<AssetIdentifier schemeIdUri=*"urn:org:dashif:asset-id:2013"* />

<BaseURL>http://ch.cdnhostname.com/HLS/ch552530/</BaseURL>

<AdaptationSet

id=*"1"*

contentType=*"video"*

group=*"1"*

par=*"16:9"*

maxWidth=*"1280"*

maxHeight=*"720"*

maxFrameRate=*"989/33"*

segmentAlignment=*"true"*

bitstreamSwitching=*"true"*

mimeType=*"video/mp4"*

startWithSAP=*"1"*

maxPlayoutRate=*"29.969999999999999"*>

<SegmentTemplate

initialization=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(video=Init)"*

timescale=*"10000000"*

media=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(video=$Time$)"*

startNumber=*"0"*>

<SegmentTimeline>

<S

t=*"0"*

d=*"20020000"*

r=*"148"* />

</SegmentTimeline>

</SegmentTemplate>

<Representation

id=*"video.1.L1"*

width=*"480"*

height=*"270"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.42c01f"*

bandwidth=*"600029"* />

<Representation

id=*"video.1.L2"*

width=*"640"*

height=*"360"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"1400033"* />

<Representation

id=*"video.1.L3"*

width=*"960"*

height=*"540"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"2400032"* />

<Representation

id=*"video.1.L4"*

width=*"1280"*

height=*"720"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"3400032"* />

<Representation

id=*"video.1.L5"*

width=*"1280"*

height=*"720"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.640029"*

bandwidth=*"4400037"* />

</AdaptationSet>

<AdaptationSet

id=*"2"*

contentType=*"audio"*

group=*"2"*

lang=*"und"*

segmentAlignment=*"true"*

bitstreamSwitching=*"true"*

mimeType=*"audio/mp4"*>

<AudioChannelConfiguration

schemeIdUri=*"urn:mpeg:dash:23003:3:audio\_channel\_configuration:2011"*

value=*"2"* />

<SegmentTemplate

initialization=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(audio\_482\_und=Init)"*

timescale=*"22050"*

media=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(audio\_482\_und=$Time$)"*

startNumber=*"0"*>

<SegmentTimeline>

<S t=*"0"* d=*"45056"* />

<S t=*"45056"* d=*"44032"* r=*"7"* />

<S t=*"397312"* d=*"45056"* />

<S t=*"442368"* d=*"44032"* r=*"7"* />

<S t=*"794624"* d=*"45056"* />

<S t=*"839680"* d=*"44032"* r=*"7"* />

<S t=*"1191936"* d=*"45056"* />

<S t=*"1236992"* d=*"44032"* r=*"7"* />

<S t=*"1589248"* d=*"45056"* />

<S t=*"1634304"* d=*"44032"* r=*"7"* />

<S t=*"1986560"* d=*"45056"* />

<S t=*"2031616"* d=*"44032"* r=*"7"* />

<S t=*"2383872"* d=*"45056"* />

<S t=*"2428928"* d=*"44032"* r=*"7"* />

<S t=*"2781184"* d=*"45056"* />

<S t=*"2826240"* d=*"44032"* r=*"8"* />

<S t=*"3222528"* d=*"45056"* />

<S t=*"3267584"* d=*"44032"* r=*"7"* />

<S t=*"3619840"* d=*"45056"* />

<S t=*"3664896"* d=*"44032"* r=*"7"* />

<S t=*"4017152"* d=*"45056"* />

<S t=*"4062208"* d=*"44032"* r=*"7"* />

<S t=*"4414464"* d=*"45056"* />

<S t=*"4459520"* d=*"44032"* r=*"7"* />

<S t=*"4811776"* d=*"45056"* />

<S t=*"4856832"* d=*"44032"* r=*"7"* />

<S t=*"5209088"* d=*"45056"* />

<S t=*"5254144"* d=*"44032"* r=*"7"* />

<S t=*"5606400"* d=*"45056"* />

<S t=*"5651456"* d=*"44032"* r=*"7"* />

<S t=*"6003712"* d=*"45056"* />

<S t=*"6048768"* d=*"44032"* r=*"8"* />

<S t=*"6445056"* d=*"45056"* />

<S t=*"6490112"* d=*"44032"* r=*"1"* />

</SegmentTimeline>

</SegmentTemplate>

<Representation

id=*"audio\_482\_und.2.L1"*

audioSamplingRate=*"22050"*

codecs=*"mp4a.40.2"*

bandwidth=*"64000"* />

</AdaptationSet>

</Period>

<!-- Postroll with segment timeline for video and audio -->

<Period

id=*"3"*

duration=*"PT0H4M58.298S"*>

<BaseURL>http://ad.cdnhostname.com/DASH/ad0571358/</BaseURL>

<EventStream

schemeIdUri=*"urn:arrisAd:adEvent:2015"*

timescale=*"10000000"*>

<Event

id=*"514"*

presentationTime=*"0"*>

<arrisAd:PlacementStart>

<arrisAd:Payload type=*"tag"* encoding=*"none"* compression=*"none"*>

providerID="ad.provider.com";

assetID="ad0571358”;

Tracking="XXXXXXXXXXXXXDAI-ADS-NoReg}:{97-2866605-04A-I…";

trickModeRestriction="";

PSNTrackingURL="https://virgin.cadent.com/PSN/";

PSNIdentity="TIVO-ADM";

PSNTimeoutMillis="2200";

PSNRetryInitialSecs="10";

PSNRetryMaxSecs="3600";

PSNMaxRetries="10";

PSNRetryBackoffMultiplier="6"

</arrisAd:Payload>

</arrisAd:PlacementStart>

</Event>

</EventStream>

<AdaptationSet

id=*"1"*

contentType=*"video"*

group=*"1"*

par=*"16:9"*

maxWidth=*"1280"*

maxHeight=*"720"*

maxFrameRate=*"989/33"*

segmentAlignment=*"true"*

bitstreamSwitching=*"true"*

mimeType=*"video/mp4"*

startWithSAP=*"1"*

maxPlayoutRate=*"29.969999999999999"*>

<SegmentTemplate

initialization=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(video=Init)"*

timescale=*"10000000"*

media=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(video=$Time$)"*

startNumber=*"0"*>

<SegmentTimeline>

<S

t=*"0"*

d=*"20020000"*

r=*"148"* />

</SegmentTimeline>

</SegmentTemplate>

<Representation

id=*"video.1.L1"*

width=*"480"*

height=*"270"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.42c01f"*

bandwidth=*"600029"* />

<Representation

id=*"video.1.L2"*

width=*"640"*

height=*"360"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"1400033"* />

<Representation

id=*"video.1.L3"*

width=*"960"*

height=*"540"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"2400032"* />

<Representation

id=*"video.1.L4"*

width=*"1280"*

height=*"720"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.4d401f"*

bandwidth=*"3400032"* />

<Representation

id=*"video.1.L5"*

width=*"1280"*

height=*"720"*

sar=*"1:1"*

frameRate=*"989/33"*

codecs=*"avc1.640029"*

bandwidth=*"4400037"* />

</AdaptationSet>

<AdaptationSet

id=*"2"*

contentType=*"audio"*

group=*"2"*

lang=*"und"*

segmentAlignment=*"true"*

bitstreamSwitching=*"true"*

mimeType=*"audio/mp4"*>

<AudioChannelConfiguration

schemeIdUri=*"urn:mpeg:dash:23003:3:audio\_channel\_configuration:2011"*

value=*"2"* />

<SegmentTemplate

initialization=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(audio\_482\_und=Init)"*

timescale=*"22050"*

media=*"S!d2EIZGFzaF9vdHQSBFT-....AZ8\_/QualityLevels($Bandwidth$)/Fragments(audio\_482\_und=$Time$)"*

startNumber=*"0"*>

<SegmentTimeline>

<S t=*"0"* d=*"45056"* />

<S t=*"45056"* d=*"44032"* r=*"7"* />

<S t=*"397312"* d=*"45056"* />

<S t=*"442368"* d=*"44032"* r=*"7"* />

<S t=*"794624"* d=*"45056"* />

<S t=*"839680"* d=*"44032"* r=*"7"* />

<S t=*"1191936"* d=*"45056"* />

<S t=*"1236992"* d=*"44032"* r=*"7"* />

<S t=*"1589248"* d=*"45056"* />

<S t=*"1634304"* d=*"44032"* r=*"7"* />

<S t=*"1986560"* d=*"45056"* />

<S t=*"2031616"* d=*"44032"* r=*"7"* />

<S t=*"2383872"* d=*"45056"* />

<S t=*"2428928"* d=*"44032"* r=*"7"* />

<S t=*"2781184"* d=*"45056"* />

<S t=*"2826240"* d=*"44032"* r=*"8"* />

<S t=*"3222528"* d=*"45056"* />

<S t=*"3267584"* d=*"44032"* r=*"7"* />

<S t=*"3619840"* d=*"45056"* />

<S t=*"3664896"* d=*"44032"* r=*"7"* />

<S t=*"4017152"* d=*"45056"* />

<S t=*"4062208"* d=*"44032"* r=*"7"* />

<S t=*"4414464"* d=*"45056"* />

<S t=*"4459520"* d=*"44032"* r=*"7"* />

<S t=*"4811776"* d=*"45056"* />

<S t=*"4856832"* d=*"44032"* r=*"7"* />

<S t=*"5209088"* d=*"45056"* />

<S t=*"5254144"* d=*"44032"* r=*"7"* />

<S t=*"5606400"* d=*"45056"* />

<S t=*"5651456"* d=*"44032"* r=*"7"* />

<S t=*"6003712"* d=*"45056"* />

<S t=*"6048768"* d=*"44032"* r=*"8"* />

<S t=*"6445056"* d=*"45056"* />

<S t=*"6490112"* d=*"44032"* r=*"1"* />

</SegmentTimeline>

</SegmentTemplate>

<Representation

id=*"audio\_482\_und.2.L1"*

audioSamplingRate=*"22050"*

codecs=*"mp4a.40.2"*

bandwidth=*"64000"* />

</AdaptationSet>

</Period>

</MPD>

# Appendix D – SCTE 130 Request and Response

The following sections present full examples of the SCTE 130-3 request from the MM to the Cadent ADS and the response given. Parts of these examples are discussed in the relevant sections of these documents but the full examples are given here.

## Request

Request made as an HTTP GET from MM to Cadent. Body of the message is an XML document as shown below –

<PlacementRequest

xmlns=*"http://www.scte.org/schemas/130-3/2008a/adm"*

xmlns:xsd=*"http://www.w3.org/2001/XMLSchema"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:core=*"http://www.scte.org/schemas/130-2/2008a/core"*

xsi:schemaLocation=*"*

*http://www.scte.org/schemas/130-3/2008a/adm SCTE\_130-3\_2008a.xsd*

*http://www.scte.org/schemas/130-2/2008a/core SCTE\_130-2\_2008a.xsd"*

messageId=*"6dcfe178-5884-47fc-b841-d8aa35586616"*

version=*"1.1"*

identity=*"AF8284CD-CB68-4F64-A651-92EBE459BDEB"*

system=*"Site: 0 AdAgent Instance: 1 Cluster: 35913 NodeGroup: 8104 BindId: 0x3ba9f6587c00bc644b5c5673"*

updatesAllowed=*"false"*>

<SystemContext>

<Session>BC:64:4B:5C:56:73/0143763282</Session>

</SystemContext>

<Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

</core:Content>

</Entertainment>

<Client>

<core:CurrentDateTime>2017-04-19T01:03:06+01:00</core:CurrentDateTime>

<TerminalAddress type=*"MAC"*>11-22-33-44-55-66

</TerminalAddress>

</Client>

<PlacementOpportunity

id=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*

serviceRegistrationRef=*"adsFullReplacement"* />

</PlacementRequest>

## Response

HTTP response from Cadent to MM. Body of the message contains an XML document as shown below.

<adm:PlacementResponse

xmlns:adm=*"http://www.scte.org/schemas/130-3/2008a/adm"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:core=*"http://www.scte.org/schemas/130-2/2008a/core"*

messageRef=*"6dcfe178-5884-47fc-b841-d8aa35586616"*

messageId=*"886c46fb-1a7b-4e07-a00b-d99cc716787e"*

version=*"1.1"*

identity=*"BA-ADS"*

system=*"BA-SYSTEM"*

xsi:schemaLocation=*"*

*http://www.scte.org/schemas/130-3/2008a/adm SCTE\_130-3\_2008a.xsd*

*http://www.scte.org/schemas/130-2/2008a/core SCTE\_130-2\_2008a.xsd"*

>

<core:StatusCode class=*"0"* />

<adm:SystemContext>

<adm:Session>BC:64:4B:5C:56:73/0143763282</adm:Session>

</adm:SystemContext>

<adm:Client>

<core:CurrentDateTime>2017-04-19T01:03:06+01:00</core:CurrentDateTime>

<adm:TerminalAddress type=*"MAC"*>11-22-33-44-55-66

</adm:TerminalAddress>

</adm:Client>

<adm:PlacementDecision

id=*"1f9be27a-f012-4ab8-8001-4f71c55fee49"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Preroll"*

opportunityNumber=*"2"*

breakOpportunitiesExpected=*"3"*

breakOpportunitySequence=*"1"*

poGroupIndex=*"1"* />

<adm:Placement

id=*"76f8b45c-c677-4cb2-b009-a2888a9a6071"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106076214"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106076214.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01k\_WiePMZT7mxKG\_yvkj\_Mg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"aab5de3d-cf71-40ba-b134-05eec6cabefe"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Preroll"*

opportunityNumber=*"3"*

breakOpportunitiesExpected=*"3"*

breakOpportunitySequence=*"2:3"*

poGroupIndex=*"1"* />

<adm:Placement

id=*"ee9b6026-410f-469b-a002-dc1b5d1ef835"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078634"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078634.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{017BwVOEqORo241oCbMZr-eQ</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"be2bc36c-c509-47eb-b598-99c2712e0051"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078551"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078551.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01z6tr\_xgiTPCrn8bBGsXtjA</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"26251b21-da98-4b4d-ac89-55af3cd750d0"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/vod/cha106002333.mpd</core:ContentLocation>

</core:Content>

<adm:EntertainmentNPT scale=*"1.0"*>0.000-575.267

</adm:EntertainmentNPT>

</adm:Entertainment>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"6f183965-3109-4a8c-8bef-52d9ec3c5745"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"5"*

breakOpportunitiesExpected=*"6"*

breakOpportunitySequence=*"1"*

poGroupIndex=*"2"* />

<adm:Placement

id=*"c1ef17bf-79b9-4ca1-91e7-a0efdc7e7828"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106070761"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106070761.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01-2oWvJG6Q9qWnJEqZ15V0A</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"0b9e38a2-a4c9-436b-8838-be1273123fd4"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"6"*

breakOpportunitiesExpected=*"6"*

breakOpportunitySequence=*"2:3:4:5:6"*

poGroupIndex=*"2"* />

<adm:Placement

id=*"9da4da78-f4ef-41cb-a56a-aa64565406ed"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078569"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078569.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{018FeWiJRnRwa4ltRq6hYK-A</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"33fe07f0-7f30-47d3-8563-5c0739ba7187"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078626"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078626.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01Qr\_n2z2HSh2HJr-khArqXA</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"065f5a06-4d95-446d-8185-a79d75804a34"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078653"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078653.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01JShHECMySISPEqBoMUervQ</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"c3533946-c793-4d64-abf2-af159c47db48"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078661"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078661.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01pom8zv-vReevXuPiAa6cmg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"d6d30ae8-8928-499b-b7dc-ddfcb641b924"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106077849"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106077849.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01DZw5pE8ITHylXTbV\_yQFNg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"cfe5f905-27cf-406b-9c59-23660918d503"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/vod/cha106002333.mpd</core:ContentLocation>

</core:Content>

<adm:EntertainmentNPT scale=*"1.0"*>575.267-1297.186

</adm:EntertainmentNPT>

</adm:Entertainment>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"f6b0b84f-1ee7-4dd8-bf34-89db6e01310f"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"11"*

breakOpportunitiesExpected=*"7"*

breakOpportunitySequence=*"1:2:3:4:5:6:7"*

poGroupIndex=*"3"* />

<adm:Placement

id=*"f386939d-02c1-4b5c-b2df-3d4cb8082d64"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078653"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078653.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01tOHhqAd-Q5y9utKrUxQTXw</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"5416371f-6a54-40cf-9aaa-a82d9956e2fa"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078277"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078277.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01px9RoMkNS4Sjeou0MmdgUw</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"cedbcf6c-5a7a-440d-b367-9bc66283066d"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106077824"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106077824.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01S-kfOxKUTZKi6A5vPOQp9g</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"bcc263f4-dfdc-4828-8fa0-f8e508790e96"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106075700"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106075700.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01\_8vMde49RSi0OpgpzUlHPA</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"9616c4c7-c314-41ba-891d-3c72dd34d762"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078569"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078569.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01rlW2XqhwTxu5rSPEbWJwnw</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"9b37363a-e86c-4277-aecb-d0ccddc52cdf"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078630"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078630.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01wuXxMmDyQdmdnSq75mZmNA</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"146d78dc-075f-49c9-9cec-a93c722a57c9"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078314"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078314.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01ryw4NavsSKShY7nwRTvBTg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"173f4669-908a-459c-82bf-61b307b341ad"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/vod/cha106002333.mpd</core:ContentLocation>

</core:Content>

<adm:EntertainmentNPT scale=*"1.0"*>1297.186-2251.426

</adm:EntertainmentNPT>

</adm:Entertainment>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"2c46b7dc-9f12-47f7-8bde-9a943a8fac76"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:OpportunityBinding

opportunityType=*"Midroll"*

opportunityNumber=*"18"*

breakOpportunitiesExpected=*"7"*

breakOpportunitySequence=*"1:2:3:4:5:6:7"*

poGroupIndex=*"4"* />

<adm:Placement

id=*"04a404c2-379b-4ab3-89ac-4ecc4b26b21e"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078314"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078314.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01H3i-DaL0QPak9Qr1zKrlGw</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"2f149d78-a12c-4b76-b94f-ebe1ef9b35bd"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106077826"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106077826.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01Co1j\_-XHTPCOBSemsSKZbA</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"32eabc7d-4f8c-4281-8c39-7cff719c2f59"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078642"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078642.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01Iso70y8DSWu86e-rrn68eA</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"7805f0fa-b7c7-4fe7-b7a8-58cf59a76f0d"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078773"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078773.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01YJzt92PhTe-EBEa3Jf\_HMg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"4f05fb44-cd6a-4ba7-9140-3b81f593fb47"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106067630"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106067630.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01lV2G0xs1RoibwiBLbj8gLQ</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"c9c69e73-aa25-4d9b-8b3f-b20af2f7105e"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106078549"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106078549.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01TGkyCw1DSTWcc02DqYbSLw</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

<adm:Placement

id=*"94acdb6d-0aa6-4ad9-8e3c-791678544005"*

action=*"fill"*>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106077218"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/ads/cha106077218.mpd</core:ContentLocation>

<core:Tracking>BA-ADS}:{01O4RoOgt-RYCGtDFFtoDsmg</core:Tracking>

</core:Content>

<adm:PlacementConstraints trickModeRestriction=*"fastForward"* />

</adm:Placement>

</adm:PlacementDecision>

<adm:PlacementDecision

id=*"ec6896ad-d10f-425a-abc9-edd2e694eee8"*

placementOpportunityRef=*"4cfb7aca-2ff2-419d-991e-66ade817e0a3"*>

<adm:Entertainment>

<core:Content>

<core:AssetRef

providerID=*"CHA"*

assetID=*"106002333"* />

<core:ContentLocation type=*"DASH\_URL"*>http://cdn.hostname.com/dash/vod/cha106002333.mpd</core:ContentLocation>

</core:Content>

<adm:EntertainmentNPT scale=*"1.0"*>2251.426-

</adm:EntertainmentNPT>

</adm:Entertainment>

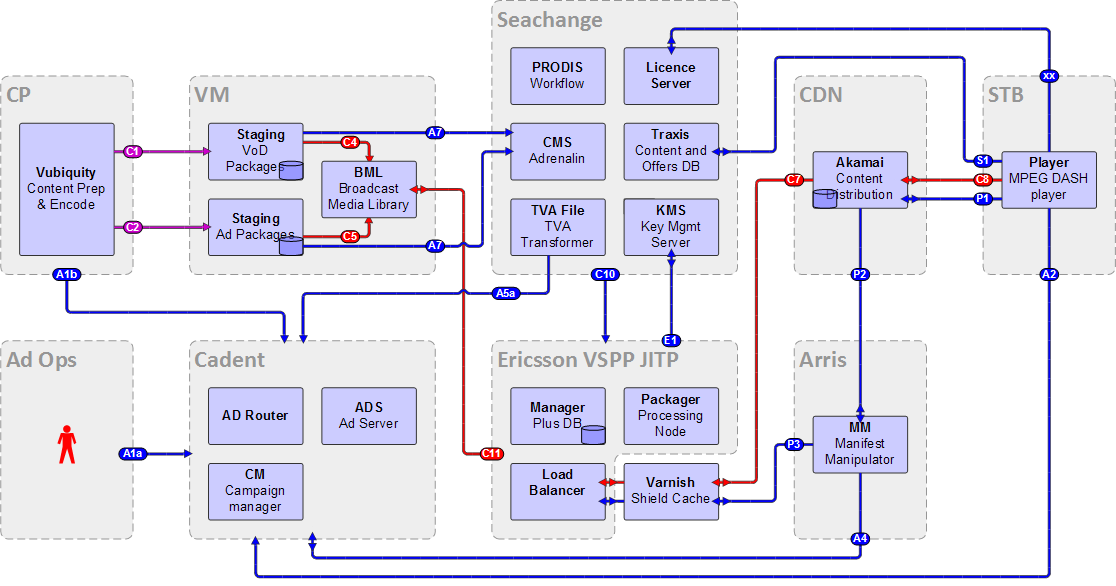
</adm:PlacementDecision>

</adm:PlacementResponse>

# Appendix E – Error Codes for STB

This section lists the error codes that appear on the STB and the cause behind it. These messages are also supported by failure log information either on the STB (accessible in TiVo Mind) or on any of the other components.

The failures listed below are organised by interface as per the diagram below –



## S1 – STB Player to Traxis

This interface used during the initial part of the session setup. STB queries Traxis for the URL required to initiate playback of the MPEG DASH content.

| Error Code | Failure | Additional Information / Log Location |
| --- | --- | --- |
| <TBC> | **STB Can’t Connect to Traxis**  This encompasses a range of errors that arise from network connectivity into Traxis. Possible causes including –   * Leapfrog connectivity problems through hub, * Compass connectivity through CMTS, * Compass connectivity through NAT and secure perimeter, * Failure at F5 load balancer, * Unresponsive Traxis, * Timeout waiting for Traxis. |  |
| <TBC> | **Error in Authorisation from Traxis**  Any errors arising from the STB being ineligible to view the content being requested. This error based on an explicit response from Traxis with a refusal. |  |
| <TBC> | **Bad Response from Traxis**  A response is received from Traxis but it cannot be parsed by the STB. This might be due to a number of issues including –   * Corruption, * Software version incompatibilities. |  |

## P1, P2 – STB Player to MM via CDN

This refers to the part of the setup where the STB requests a manifest from the MM and receives a customised MPEG DASH manifest in return. The request from the Leapfrog box uses HTTP and goes via the CDN and the F5 load balancer; the Compass box goes directly from the CMTS/NAT to the F5 load balancer and MM.

| Error Code | Failure | Additional Information / Log Location |
| --- | --- | --- |
| <TBC> | **STB Can’t Connect to CDN**  This encompasses any number of failures from the STB to the MM. Some of these are broken out in more detail below. |  |
| <TBC> | **Leapfrog Security Token Failure**  Authorisation failure on the token passed from Traxis. Possible causes include –   * Different shared secret, * Incorrect calculation of token. | Check CDN logs. |
|  | **Compass to F5 Failure**  Will be seen by the STB as a connection failure. | Check F5 logs for details. |
| <TBC> | **CDN to F5 Failure**  Will be seen by the STB as a connection failure. | Check CDN logs. |
|  | **F5 to MM Failure**  Will be seen at the STB as a connection failure. | Check F5 logs for details. |
|  | **Timeout at STB Waiting for Reply**  Reply took too long. |  |
|  | **Unintelligible Response from MM**  A reply is received from the MM but the STB cannot parse it. Causes might include –   * Corruption, * Software version incompatibilities. |  |
|  | **Error Response from MM**  This covers the situation where the MM sends a readable reply but that reply indicates an error in the setup. These errors are likely due to upstream errors and are listed under the sections detailing those interfaces. |  |

## P3 – MM to Varnish/VSPP

This is the interface that the MM uses to request manifests from the VSPP via the Varnish cache. The request can either be as a result of the initial PlayURL or a request related to an ad as indicated in the SCTE 130 response. Where possible we want to see an error code on the STB indicating the nature of the failure.

| Error Code | Failure | Additional Information / Log Location |
| --- | --- | --- |
|  | **Main Content Not Found**  The request to the VSPP for the manifest of the main VoD content is not present on the VSPP. | MM returns an http 404 to the STB client. |
|  | **Main Content Manifest Partially Complete**  This can occur if the main content is still ingesting into the VSPP. The VSPP will return a dynamic partial manifest. | MM should treat as an error and return http 404 to the STB client. |
|  | **Main Content Unintelligible to MM**  The VSPP returns a manifest that cannot be parsed by the MM. This might be due to corruption or software incompatibilities. | STB should see this as an http 400.  MM Logs will indicate nature of failure. Failure will need reporting to operations through some sort of alarm. |
|  | **Timeout from Varnish/VSPP**  Failure of the VSPP to provide the main manifest within the configured interval. | MM will return to STB as http 408.  Error should also be visible in MM logs. |
|  | **Ad Manifest not Found**  The ad referenced in the SCTE 130 response cannot be found on the VSPP. Although an error, the MM will handle this by skipping the ad. | STB will see no error; the ad will simply be omitted from playback.  MM logs should record the error. Possibly need to raise an alarm to ops. |
|  | **Ad Manifest Timeout**  Timeout retrieving the ad manifest from the VSPP. The MM should log the error and omit the ad from the final manifest. | STB will see no error; the ad will simply be omitted from playback.  MM logs should record the error. Possibly need to raise an alarm to ops. |
|  | **Ad Manifest Unintelligible**  The manifest cannot be parsed by the VSPP. | STB will see no error; the ad will simply be omitted from playback.  MM logs should record the error.  Alarm raised. |

## A4 – MM to Cadent

This is the SCTE 130-3 interface used by the MM to request the playlist of content and ads from the Ad Router and ADS. The SCTE 130-3 reply contains details of the ads, content segments and order in which they are to be played.

| Error Code | Failure | Additional Information / Log Location |
| --- | --- | --- |
|  | **Connection Failure MM to Ad Router**  This might be caused by the Ad Router being down or through a network failure. The MM behaviour is to attempt to play the content “in the raw” with no ads. | STB will see content without ads.  MM logs will record the error.  Alarm to be raised. |
|  | **Timeout from Ad Router or ADS**  The response from the ad router does not come within the configured time window. The MM behaviour is to attempt to play the content “in the raw” with no ads. | STB will see content without ads.  MM logs will record the error.  Alarm to be raised. |
|  | **Error Response from Ad Router or ADS**  SCTE 130-3 response indicates an error. MM behaviour as above. | STB will see content without ads.  MM logs will record the error.  Alarm to be raised. |
|  | **Response from Ad Router Unintelligible**  The SCTE 130-3 response cannot be parsed. MM behaviour as above. | STB will see content without ads.  MM logs will record the error.  Alarm to be raised. |

## xx – STB to Licence Server

Interface used during the initiation of the playback. The STB uses the information contained in the manifest to request a decryption key from the Licence Server.

| Error Code | Failure | Additional Information / Log Location |
| --- | --- | --- |
|  | **Connection Failure**  This encompasses the error case where the STB is unable to connect to the Licence Server. Reasons might include –   * Network problems between STB and F5 load balancer, * Network problems between F5 and Licence Server, * F5 down, * Licence Server down. | STB displays error to user.  No playback possible. |
|  | **Connection Timeout**  STB times out waiting for reply from Licence Server. | STB displays error to user.  No playback possible. |
|  | **Licence Response Unintelligible**  The reply received by the STB cannot be parsed. | STB displays error.  No playback possible.  Licence Server logs to indicate the exchange from the Licence Server end. |
|  | **Licence Request Rejected**  The reply from the Licence Server indicates that the STB is not authorised to play the content. | STB displays error.  No playback possible. |

## C7, C8 – STB to CDN/VSPP

Interface from the STB to the CDN to retrieve content segments. In the event of a cache miss, the CDN will request the content just-in-time from the VSPP. VSPP will package and encrypt the segment on-the-fly.

| Error Code | Failure | Additional Information / Log Location |
| --- | --- | --- |
|  | **Communication Failure toward CDN**  This covers the case where the STB is unable to contact the CDN. Reasons might include –   * Network failure, * Firewall configuration. | STB will attempt retry a number of times before giving up. May result in a pause and buffering message. |
|  | **Cookie Validate Failure toward CDN**  The cookie passed in the request to the CDN does not validate correctly. Cookies have a time-to-live after which they will fail at the CDN. | STB will stop playback with an error.  CDN logs should record the validation failure. |
|  | **Segment Not Found**  This will be an error returned by the CDN to the STB after the CDN has tried, unsuccessfully, to retrieve the requested segment from the VSPP. | STB will stop playback with an error. Error returned to STB will be http 404.  CDN logs should record the error.  Alarm required to ops. <confirm> |
|  | **Segment Corrupt**  Segment cannot be parsed by the STB. | STB will stop playback with an error. |
|  | **Decryption Failure**  STB is unable to decrypt the returned segment. | STB will stop playback with an error. |

# Appendix F – Harmonic/Akamai and VSPP/Velocix

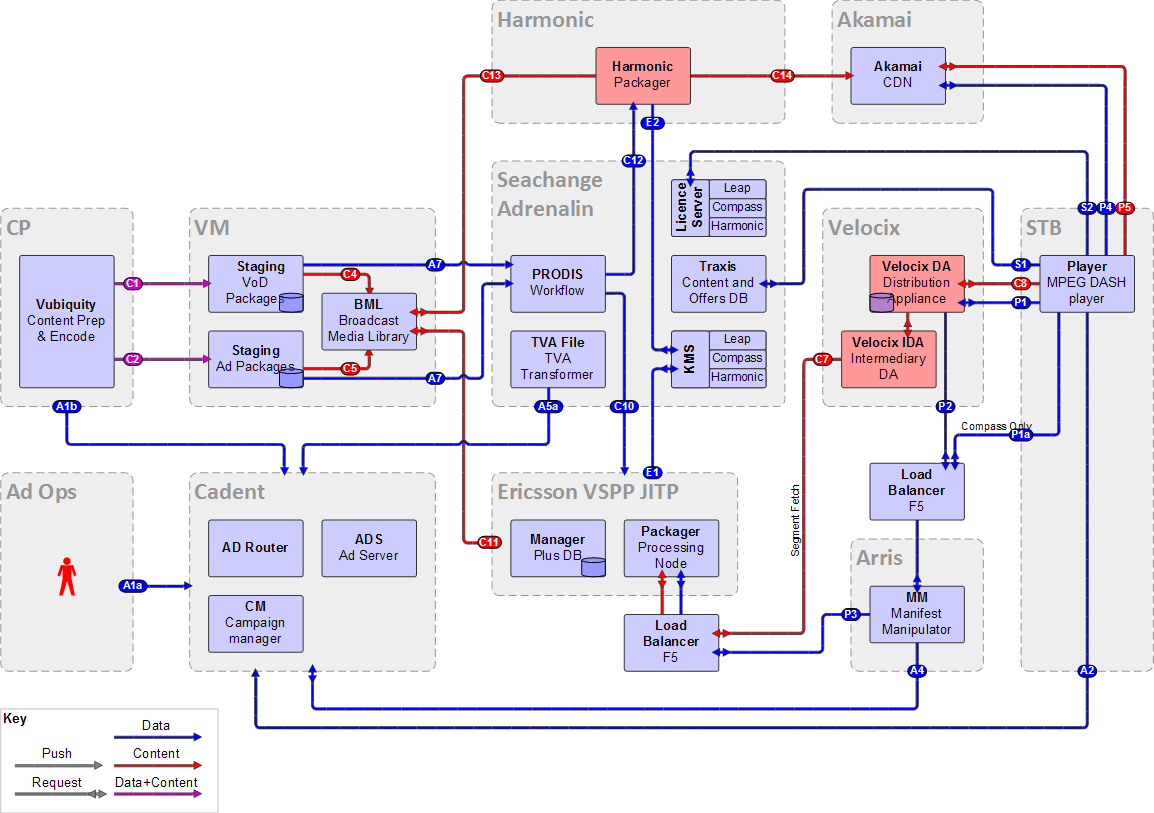
This appendix also forms the basis of a tech note on Confluence and details a number of changes from the main document to accommodate the following –

* Use of Velocix as the primary CDN for the eventual long-term solution using VSPP and targeted advertising,
* Use of Akamai and the Harmonic packager to provide a short-term IP VoD solution without advertising,
* Notes on the configuration of load balancers and firewalls that front the VSPP and MM components,
* Notes on changes required in the SeaChange Adrenalin components to accommodate the ingest and playout of both paths.

Later revisions of this document will incorporate these elements into the main body of the document.

## Revised Functional Block Diagram

The following diagram shows the revised overall functional blocks with elements supporting both the Harmonic and VSPP approaches.



The system boxes highlighted in red in the above diagram show the new Velocix CDN replacing the Akamai CDN for the VSPP playout and the Harmonic packager using the existing Akamai CDN.

Note also that the Varnish shield cache fronting the VSPP has been removed and is now replaced by a load balancer only.

### Interfaces (New and Changed)

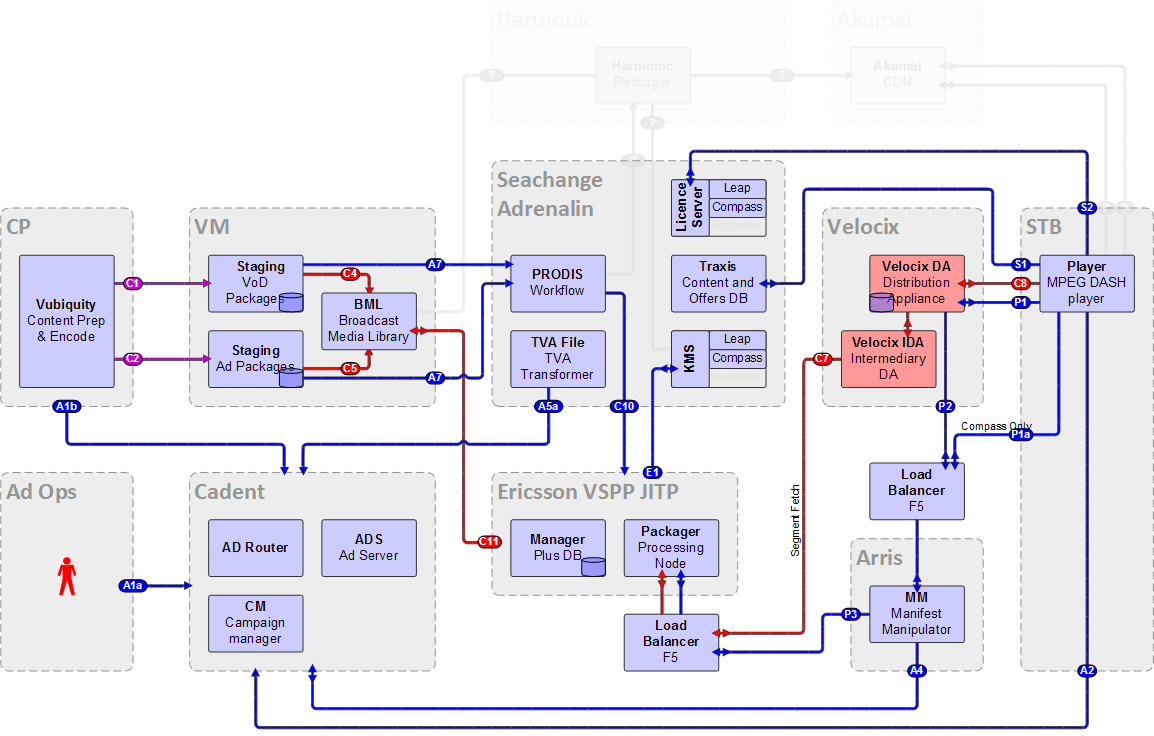
The following interfaces are either new as of this appendix or are changed from the main body of the document above. For other interfaces please see the main body of the document.

|  |  |  |  |
| --- | --- | --- | --- |
| I/F | Description | Protocol | Specification |
| C7 (Changed) | **CDN Pull Content Segments from VSPP**  As MPEG DASH segments are requested by the STB from the CDN, the CDN either satisfies these from cache or forwards the request to VSPP via the load balancer. VSPP packages and encrypts the requested segment on-the-fly and returns to the CDN.  Changed source of the request: now originated from the Velocix IDA.  Changed destination: Now directed to the load balancer rather than Varnish. | MPEG-Dash |  |
| C8 (Changed) | **STB Requests Content Segments from CDN**  As the STB plays the content under the direction of the MPEG DASH manifest it fetches the individual segments of content from the CDN. The exact segment variant requested will depend on network conditions.  Changed destination: The request is now directed to the Velocix DA instead of Akamai. | HTTP / MPEG DASH  HTTP (Compass)  HTTPS (Leapfrog) |  |
| C12 (New) | **PRODIS to Harmonic Packager – Ingest (3d)**  The SeaChange PRODIS workflow manager directs the Harmonic packager to ingest content through the SeaChange 3d interface. | HTTP with XML body. | SeaChange Interface Part 3d - Content propagation. |
| C13 (New) | **Harmonic Packager pulls content from BML**  This is the interface where the Harmonic Packager pulls the essences from the BML using the URL provided in the registration message. | FTP |  |
| C14 (New) | **Harmonic Packager delivers content to Akamai Origin**  Harmonic Packager sends the processed MPEG DASH manifests and segments to the Akamai CDN for distribution. | FTP |  |
| E2 (New) | **Harmonic Packager requests encryption key from KMS**  As part of the packaging process, the content is encrypted using PlayReady DRM. This interface allows the SeaChange Key Management Server to supply the key and metadata for embedding in the manifest. | HTTP - PlayReady |  |
| P1 (Changed) | **STB to CDN – Request Title Manifest**  On playout of IP VoD content, the STB will request the MPEG DASH manifest for that title from the Manifest Manipulator (MM) via the CDN. The MM will query for ad insertion instructions, gather together the manifests for the title and ads and then stitch all of these into a single combined manifest before returning to the STB. For this call, the CDN is configured to simply pass the request through to the MM; if can’t be cached at the CDN since each request will potentially result in a different set of ads to be inserted.  Changed destination: The player is now requesting content from the Velocix CDN rather than Akamai. | HTTP / MPEG DASH |  |
| P2 (Changed) | **CDN to MM – Request Title Manifest**  The CDN to MM part of the process described above. For this request, the CDN doesn’t cache but simply forwards the request to the MM.  Changed source: The request now comes from the Velocix DA rather than Akamai. | HTTP / MPEG DASH |  |
| P4 (New) | **STB requests DASH manifest from Akamai**  This is part of the Harmonic playout. The STB pulls the manifest from the Akamai CDN. No manipulation of the manifest is performed as there is no ad insertion. Token processing applies on the CDN to verify this is a valid client. | MPEG DASH |  |
| P5 (New) | **STB requests DASH segments from Akamai**  This is part of the Harmonic playout. The STB pulls the MPEG DASH content segments from the Akamai CDN. | MPEG DASH |  |

## VSPP Playout

This is the solution as described in the main body of the IP VoD Solution Design (this document). However, there are some changes to this to use the Velocix CDN instead of Akamai. These elements are shown in the diagram below, these being largely the original architecture.

Notes on the changes are described in the following sections, these being largely concerned with the setup flow for Leapfrog and Compass boxes now being through Velocix rather than Akamai. There are also changes to the way load balancing and firewalling is carried out in front of the Arris MDC and Ericsson VSPP components.



The notes that follow detail changes from the main body of the Solution Design document (this document).

### Ingest Flow Notes

Ingest is as per the processes described in the main body of the document above. The Encoding\_Type field in the ADI file is used to mark this content as being for the VSPP platform. The encoding type is use are –

|  |  |
| --- | --- |
| H264-SD | SD content for the VSPP platform |
| H264-HD | HD content for the VSPP platform |
| H264-UHD | Not currently used |
| H264-ADV | Advert content for the VSPP platform containing renditions spanning both SD and HD. |

ADI files for the VSPP platform will not have a derived asset section.

### CDN Token Processing

Access into the Velocix CDN by the STB for the manifest request is over https. Additionally, the STB is required to supply with the request a token that was generated by TRAXIS and encrypted using a key shared between TRAXIS and the CDN. This token verifies to the CDN that the request comes from a valid STB. This token process is identical to that implemented on the Akamai CDN and described in the main body of the document above.

Additionally, subsequent segment requests into the CDN will contain a cookie that was supplied to the STB on its previous interaction with the CDN. Again, this verifies that the request if from a valid client. Again, this is described in the main body of the document above in relation to the Akamai CDN. It should be noted that this cookie is an additional security measure that protects against DDoS attacks; the returned content segments are themselves strongly encrypted using the DRM scheme.

### Playout (Leapfrog and Compass)

The playout flow is largely as described in section 6 of the main body of the document above except for the following differences –

* Requests to the CDN will now be targeted at the Velocix CDN instead of Akamai,
* Manifest requests via the CDN will break out of the DA and be directed to the MM whereas other requests will either be satisfied internally or by a request from the IDA,
* The IDA will either satisfy requests from the DA internally or will fetch the required data from the F5/VSPP,
* Requests to VSPP will be via an F5 load balancer instead of the Varnish shield cache, the only caching in this path will be by the CDN and the MM internal cache (for manifests)

### Arris MDC Firewalling and Load Balancing

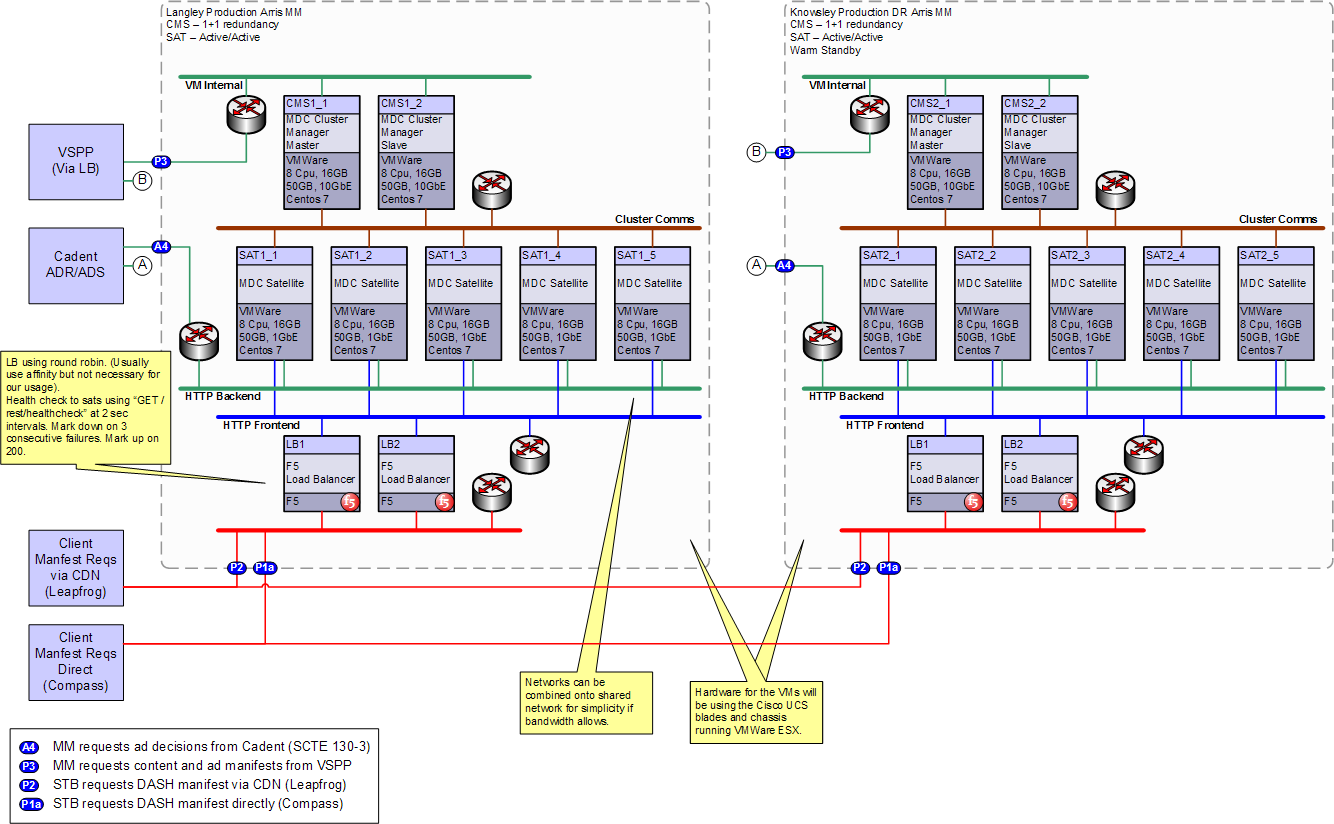
**Full details to be finalised by CN&O team**. For guidance, the diagram below is derived from the Arris project documentation and shows the components that comprise the clusters in both Langley and Knowsley.

The F5 load balancers are shown as an indication that some form of firewalling and load balancing is required onto the platform although this does not have to be through F5 units; CN&O will determine the exact configuration here. One point to note here is that this architecture supports both the Leapfrog boxes which come into the MDC via the CDN and the Compass boxes which bypass the CDN and use http instead of https. This suggests that separate load balancers might be required here to allow load balancing for both routes. If the CDN were to do load balancing then traffic from the Compass boxes would also need to loop through the CDN and the CDN be configured to allow this traffic originating within the VM network and using http. Final decision to be made by CN&O.

Another feature shown here is the DR solution. This has a separate cluster of MDC components available at Knowsley. Switchover to the Knowsley cluster would require an operational change to change the network routing to point clients towards these services rather than the Langley cluster. Traffic from the cluster components to the Cadent and VSPP originates within the Arris components and would not require any reconfiguration on switchover to the DR cluster. Arris have suggested that any of the following methods could be used to perform this switchover-

* DNS update to provide new IP address to access point on secondary site,
* Virtual IP or Route Health Injection to migrate existing TCP/IP traffic to secondary site or
* Frontend LB or CDN edge that is updated to route traffic to the secondary site.

Alternatively, Arris have indicated that the Knowsley cluster could be operated active/active with the Langley cluster. In this configuration, there would need to be operational processes or load balancing in place to allow one of the clusters to pick up the full load in case of a failure or for scheduled maintenance.



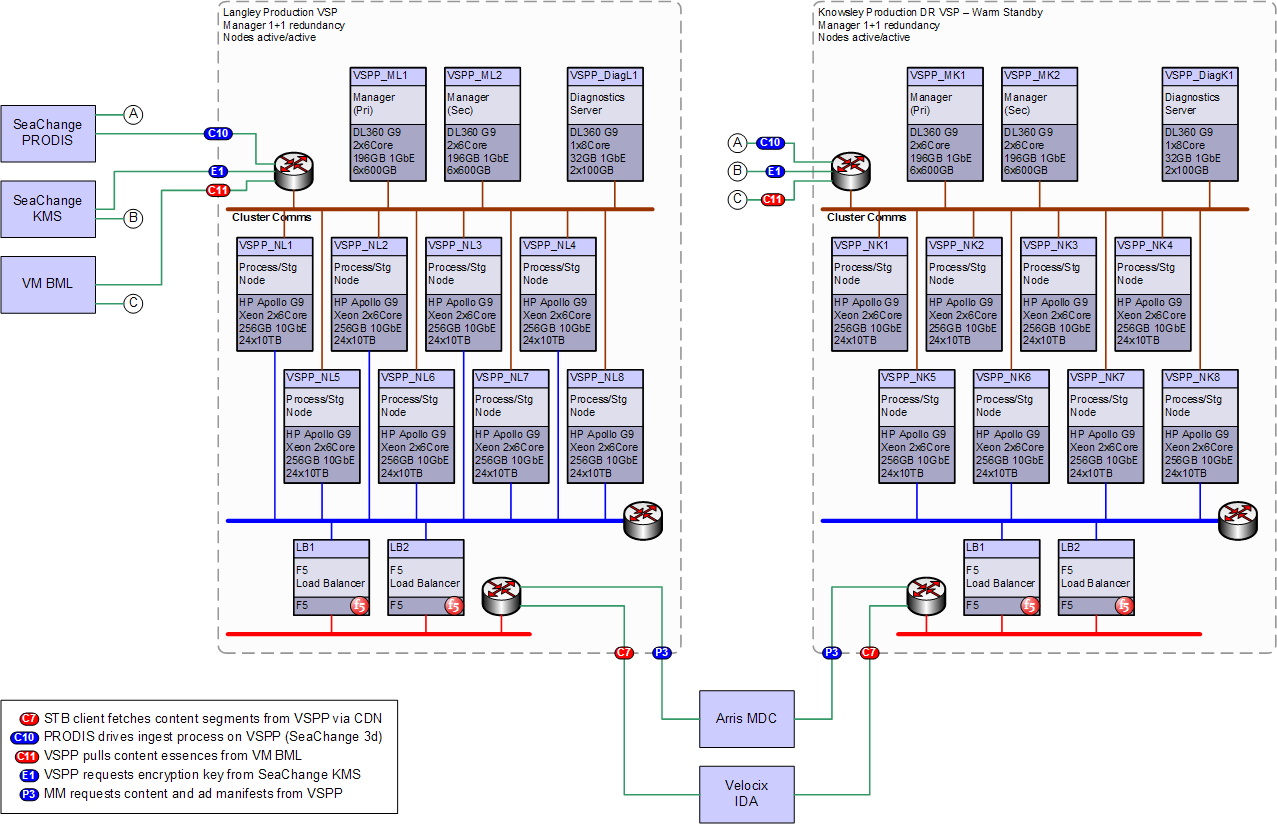
### VSPP Firewalling and Load Balancing

**Full details to be finalised by CN&O team**. For guidance, the diagram below is derived from the Ericsson project documentation and shows the components that comprise the clusters in both Langley and Knowsley.

The F5 load balancers are shown as an indication that some form of firewalling and load balancing is required onto the platform although this does not have to be through F5 units; CN&O will determine the exact configuration here.

The VSPP cluster needs to support both manifest requests coming in via the MDC and segment requests coming in through theVelocix IDA CDN. Load balancing is required onto the VSPP processing nodes from both of these routes and thus suggests the use of load balancers fronting the cluster. If the CDN were to do load balancing then traffic from the Arris MDC would also need to loop through the CDN and the CDN be configured to allow this traffic originating within the VM network.

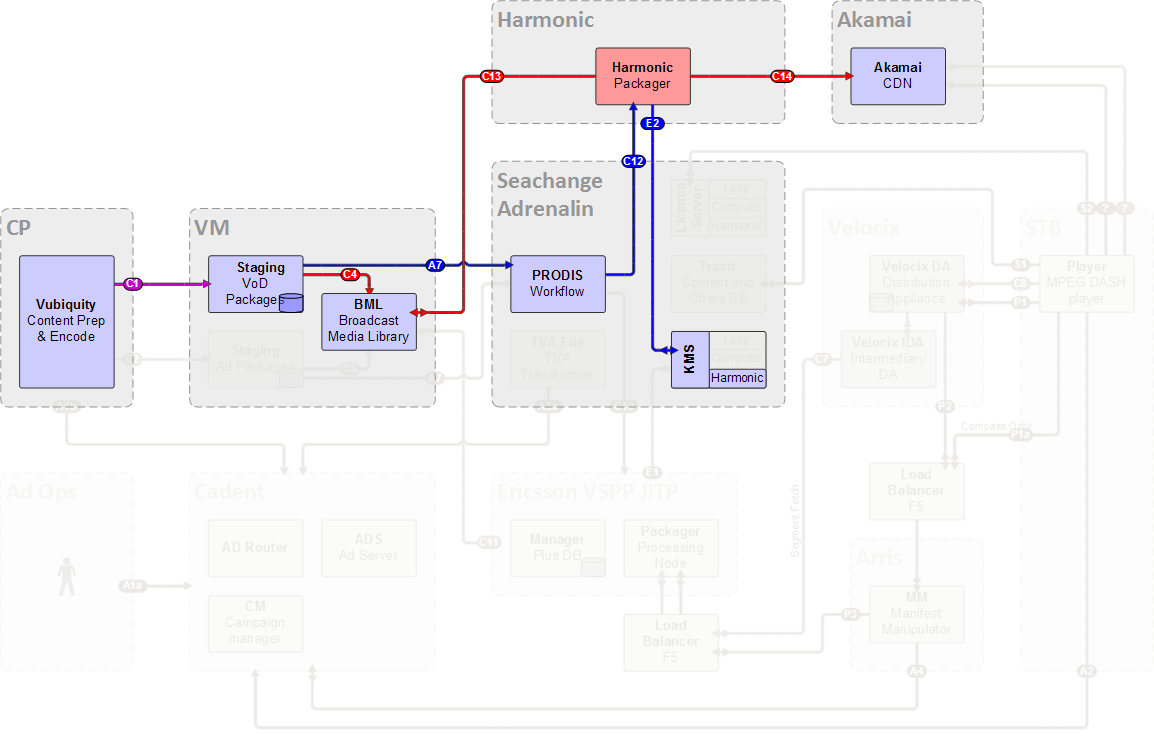
**Final decision to be made by CN&O**.



## Harmonic Ingest

The Harmonic approach is based on a simplified architecture with ingested content being statically packaged into an MPEG DASH manifest and all required MPEG DASH content segments. These files are then pushed to a CDN origin, Akamai in this case, whence they can be served to clients.

As with the VSPP approach, content for this method of delivery is ingested onto a staging server (on a different folder) and is then managed by the SeaChange PRODIS workflow onto the Harmonic packager and the metadata onto the SeaChange Adrenalin system. As with the VSPP ingest, Harmonic is informed of content to ingest over an HTTP interface (3d in the SeaChange nomenclature). The Harmonic packager then pulls the content from the BML, processes it and delivers the results to the Akamai origin.



### Content Preparation

Content preparation is largely as per the content for use on the VSPP platform but with a different Encoding\_Type value being used to mark this content as being intended for the Harmonic platform. Encoding types in use here are –

|  |  |
| --- | --- |
| DASH-STB-HD | HD content for Harmonic platform |
| DASH-STB-SD | SD content for Harmonicplatform |

There are no ads to consider in this solution since this service is ad-free. Consequently, only main content needs to the ingested and ADI packages are sent to a single ingest point on the staging server.

Content for the Harmonic platform will also contain a Derived Asset <Asset> element with this stating the Encoding\_Type specified above as this is what the current SeaChange PRODIS platform expects.

### Staging Server

<TBC – Possibly different ingest folder or possibly discriminate on Encoding\_Type>

### Workflow

The SeaChange PRODIS workflow needs to detect that this is Harmonic content and drive a separate ingest workflow from that of the VSPP content.

The Harmonic workflow uses the SeaChange 3d interface (C12 in the above diagram) to initiate an ingest to the Harmonic Platform. The request will contain details of the SeaChange ID of the content and the location on the BML from which the Harmonic packager can collect the content.

As with the VSPP process, the Harmonic packager will be required to respond to status requests over this interface indicating the progress of the packaging operation.

### Harmonic Processing

The Harmonic packager is required to package the supplied content and its various renditions into an MPEG-DASH manifest and related MPEG-DASH content segments at the various bitrates.

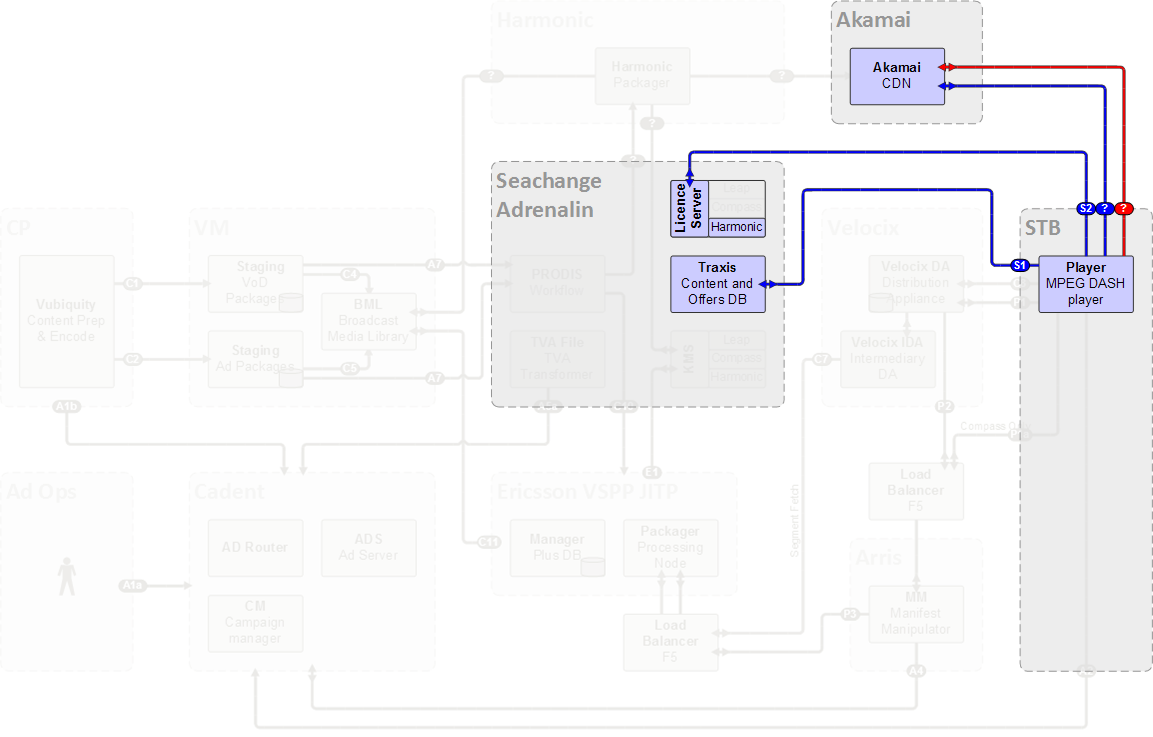
All generated content elements are then pushed by the Harmonic packager onto the Akamai CDN origin for distribution.

Content is protected by Microsoft PlayReady DRM as applied by the packager and with relevant metadata embedded in the manifest. The Harmonic packager talks to a SeaChange Key Management Server (KMS) to provide the encryption keys and metadata. This KMS is a separate instance to that used for the VSPP content and the related metadata embedded in the manifest also points to a separate related licence server. The KMS instance may however run on the same server as the KMS for the VSPP instances.

## Harmonic Playout

Harmonic playout from the STB point of view is very similar to that of the VSPP approach. However, the back end is much simpler.

As with VSPP, the STB asks SeaChange Traxis for details of the playout URL. Traxis, will deliver a URL pointing to the content manifest on the Akamai CDN. The STB will fetch the manifest, negotiate the DRM with the Licence Server and then fetch and play the content segments from the CDN.



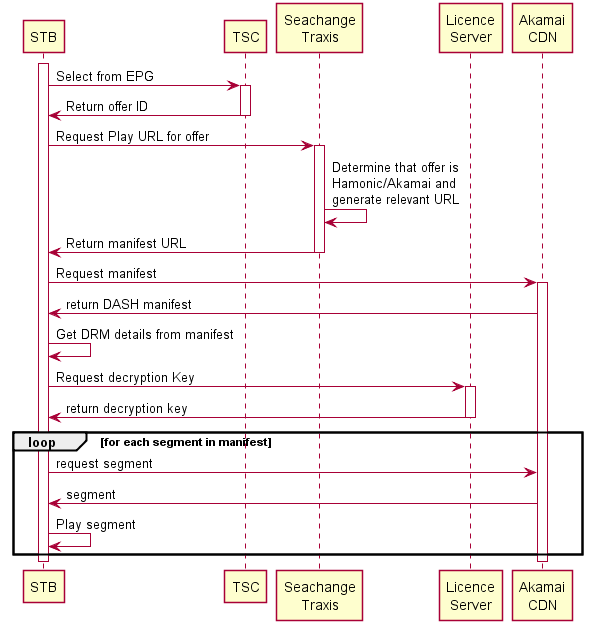
### Play URL

The key difference in the playout setup of Harmonic content from VSPP content is the initial URL that the STB uses to request the manifest. The STB obtains this URL from the SeaChange TRAXIS system and then uses that URL to start the MPEG-DASH streaming playback.

TRAXIS is required to examine the SeaChange Content ID of the content being requested and use this to determine whether this content is on the Harmonic platform or the VSPP platform. An appropriate URL, pointing to relevant servers for the platform, should then be constructed and sent back to the STB client. TRAXIS should base this decision on metadata derived from the Encoding\_Type field that was originally sent in the ADI file. The platform will be operated in such a way that content is loaded to either the Harmonic platform or the VSPP platform and this this mechanism can be safely used to discriminate.

### Playout Flow

Playout for harmonic content is much simpler than for VSPP content since there are no ads and no dynamic generation of the manifest. The manifest and segments are simply served statically from the CDN.



### STB Requirements

From the STB point of view, processing should be largely identical to that used in the VSPP playout flow. The STB will contact Traxis as per the VSPP flow and will receive a URL that it can use to fetch the manifest for the content. The fact that the manifest (and the content pointed to in the manifest) comes from a different server should be irrelevant to the STB.

The manifest fetched by the STB will contain DRM information which will entail a trip to the Licence server as per the VSPP flow. The address of this server is contained in the manifest and the STB should need no additional logic to access this.

The Harmonic manifest will differ from the MDC/VSPP manifest in that it will contain no adverts and will consist of a single period. The manifest will contain no data related to the creation or delivery of PSN messages since there will be no ads in the stream. The STB will not be required to deliver any PSN notifications in respect of this content.

## Volumetrics

The figures here are based on the published NFRs but without the 50% uplift noted in that document giving us a peak capacity of 50,000 concurrent sessions. Setup rate is still as per the NFRs.

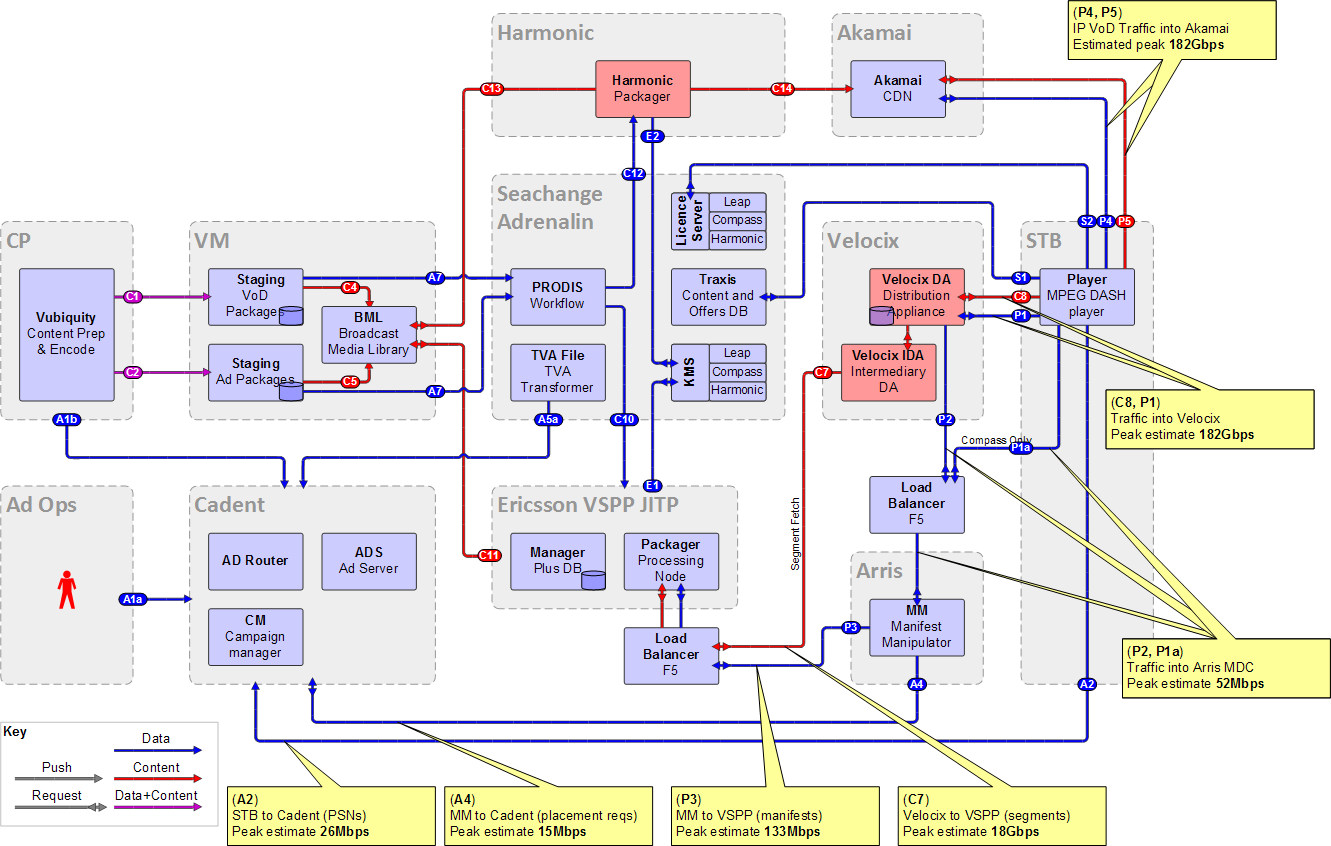
Volume into the VSPP cluster is fronted by the CDN and here we are assuming a high cache hit rate of 90% based on the assumption that the majority of traffic will be for popular catch-up titles thus driving a high hit rate. <TBC – revise VSPP calcs based on real QAM viewing data>.

The peak figure relates to overall IP VoD activity delivered over both the Harmonic/Akamai and VSPP/Velocix delivery platforms. The platform will initially be deployed using the Harmonic platform. The VSPP platform will come on line later and will then gradually ramp up as the Harmonic platform ramps down. Content will expire on Harmonic and migrate to VSPP. Overall, the combined capacity of the Harmonic and VSPP solutions will be fairly constant but after the initial deployment on Harmonic, the balance of traffic will shift towards the VSPP solution.

The estimates in the following sections are based on these assumptions.

### Interface Volumes

The following diagram shows the interfaces used for the capacity estimates in the following sections in terms of their position in the functional block diagram. These interface numbers are also shown on the physical diagrams earlier in this appendix.



**Note:** that these calculations should be treated as a rough estimate only. Full detailed estimation of capacities will be carried out by the relevant teams and will replace these estimates.