

Описание решения

**«Интерактивное телевидение Ростелеком»**

Содержание

1. Введение 12

1.1. О документе 12

1.2. Глоссарий 12

1.2.1. Бизнес-термины и сокращения 12

1.2.2. Технические определения и сокращения 15

1.2.3. Системы и компоненты 18

2. Анализ текущего решения Ростелеком ИТВ 20

2.1. Общее описание решения 20

2.2. Состав решения 21

2.3. Техническое описание Платформы интерактивного   
телевидения 24

2.3.1. Назначение 24

2.3.2. Возможности платформы 24

2.3.3. Архитектура 24

2.3.3.1 Развертывание компонентов 27

2.3.4. Форматы данных 28

2.3.4.1 Информационные сущности 28

2.3.4.2 Файлы контента 29

2.3.4.3 Структура баз данных 29

2.3.4.4 Форматы данных внутреннего API 29

2.3.4.5 Форматы данных внутреннего API 29

2.3.5. API взаимодействия с клиентскими продуктами 29

2.3.5.1 Методы управления учетными записями 30

2.3.5.2 Методы тарификации услуг 30

2.3.5.3 ОТТ-методы 31

2.3.5.4 Методы для работы с CoD 32

2.3.5.5 Методы для работы с ТВ-каналами 32

2.3.5.6 Методы для работы с избранными ТВ-каналами 32

2.3.5.7 Методы для работы с EPG 32

2.3.5.8 Методы для работы с сериалами и сезонами 33

2.3.5.9 Методы для получения вспомогательных справочников 33

2.3.5.10 Методы работы с профилями клиентских приложений 33

2.3.5.11 Методы для работы с рекомендациями 33

2.3.5.12 Методы для работы с информационными сервисами 33

2.3.5.13 Методы для работы с рекламой 34

2.3.5.14 Методы для работы с промокодами 34

2.3.5.15 Методы для работы с UGC 34

2.3.5.16 Методы для работы с nPVR 34

2.3.5.17 Методы для работы с сохраненными позициями просмотра 34

2.3.5.18 Методы для работы с социальными сетями 35

2.3.5.19 Методы сбора статистики 35

2.3.5.20 Методы для приема обратной связи от абонентов 35

2.3.5.21 Методы для работы с платежной системой 35

2.3.5.22 Справочник методов, вызываемых со стороны SDP 35

2.3.6. Взаимодействие с внешними системами 36

2.3.7. Внутренние потоки данных 36

2.3.7.1 Загрузка VoD-контента 37

2.3.7.2 IPTV-вещание ТВ-эфира 39

2.3.7.3 Получение PvR-контента прямого эфира 39

2.3.7.4 Вещание CoD-контента 39

2.3.7.5 Вещание PVR-контента 39

2.3.7.6 Обновление данных SDP 39

2.3.7.7 Сбор данных о пользовательских действиях 39

2.3.7.8 Использование услуг 39

2.3.7.9 Отправка нотификаций пользователям 39

2.3.7.10 Администрирование решения 39

2.3.7.11 Отправка файлов интерфейса SmartTV 39

2.3.8. Состав платформы 39

2.3.8.1 Система доставки контента 39

2.3.8.2 Система управления контентом 42

2.3.8.3 Общесистемные сервисы 45

2.3.9. Аппаратное обеспечение 46

2.3.9.1 Состав оборудования 46

2.3.9.2 Стендовая инсталяция 48

2.4. Описание услуг, предоставляемых решением Ростелеком ИТВ 48

2.4.1. Состав услуг 48

2.4.2. Основные сценарии использования 48

2.4.2.1 Регистрация и авторизация устройства 49

2.4.2.2 Регистрация пользователя 49

2.4.2.3 Авторизация пользователя 49

2.4.2.4 Просмотр пользователем IPTV-контента в прямом эфире 49

2.4.2.5 Просмотр пользователем OTT-контента в прямом эфире 49

2.4.2.6 Просмотр пользователем OTT-контента CoD 49

2.4.2.7 Загрузка нового контента 49

2.4.2.8 Биллинговое обслуживание 49

2.5. Описание клиентских продуктов решения Ростелеком ИТВ 49

2.5.1. IPTV/RTSP-приставки 50

2.5.1.1 Предоставляемые услуги 50

2.5.1.2 Описание работы 50

2.5.1.3 Взаимодействие с серверной платформой 50

2.5.1.4 Структура пользовательского интерфейса 50

2.5.2. IPTV/OTT-приставки 50

2.5.2.1 Предоставляемые услуги 50

2.5.2.2 Описание работы 50

2.5.2.3 Взаимодействие с серверной платформой 50

2.5.2.4 Структура пользовательского интерфейса 50

2.5.3. OTT-приставки 51

2.5.3.1 Функции 51

2.5.3.2 Предоставляемые услуги 51

2.5.3.3 Описание работы 51

2.5.3.4 Взаимодействие с серверной платформой 51

2.5.3.5 Структура пользовательского интерфейса 51

2.5.3.6 Архитектура 51

2.5.4. Android-приложение Забава 51

2.5.4.1 Предоставляемые услуги 51

2.5.4.2 Описание работы 51

2.5.4.3 Взаимодействие с серверной платформой 51

2.5.4.4 Структура пользовательского интерфейса 51

2.5.5. iOS-приложение Забава 52

2.5.5.1 Предоставляемые услуги 52

2.5.5.2 Описание работы 52

2.5.5.3 Взаимодействие с серверной платформой 52

2.5.5.4 Структура пользовательского интерфейса 52

2.5.6. Веб-плеер Забава 52

2.5.6.1 Предоставляемые услуги 52

2.5.6.2 Описание работы 52

2.5.6.3 Взаимодействие с серверной платформой 52

2.5.6.4 Структура пользовательского интерфейса 52

2.6. Недостатки текущего решения 52

2.6.1. Выявленные недостатки платформы «Интерактивное телевидение Ростелеком» 52

2.6.1.1 Общесистемные 53

2.6.1.2 Управление работой и контентом 53

2.6.1.3 Доставка контента 56

2.6.1.4 База данных SDP 56

2.6.2. Выявленные недостатки услуг решения 57

2.6.3. Выявленные недостатки клиентских продуктов 57

3. Концепция решения Ростелеком ИТВ 2.0 58

3.1. Предпосылки 58

3.2. Постановка задачи 59

3.2.1. Услуги, предоставляемые решением 59

3.2.1.1 Перечень услуг 59

3.2.1.2 Ценообразование и структура услуг 61

3.2.2. Клиентские продукты решения 61

3.2.2.1 Абонентские устройства 61

3.2.2.2 Клиентские приложения 62

3.2.3. Платформа интерактивного телевидения 2.0 62

3.2.3.1 Система управления платформой 62

3.2.3.2 Система доставки контента 62

3.3. Основные сценарии использования решения 62

3.3.1. Пользовательские сценарии 62

3.3.1.1 Подключение IPTV-приставки 63

3.3.1.2 Подключение OTT-приставки 63

3.3.1.3 Установка приложения 64

3.3.1.4 Активация устройства 64

3.3.1.5 Деактивация устройства 64

3.3.1.6 Авторизация пользователя 65

3.3.1.7 Объединение клиентских продуктов в группы 65

3.3.1.8 Управление профилями 65

3.3.1.9 Удаление клиентских продуктов из группы 66

3.3.1.10 Смена тарифного плана 66

3.3.1.11 Приобретение подписки на ТВ-пакет 67

3.3.1.12 Просмотр ТВ-канала 67

3.3.1.13 Приобретение/аренда VoD, VoD-пакета 67

3.3.1.14 Просмотр VoD контента 68

3.3.1.15 Загрузка контента 68

3.3.1.16 Просмотр контента без подключения к интернету 68

3.3.1.17 Поиск 69

3.3.1.18 Кинорулетка 69

3.3.1.19 Родительский контроль 70

3.3.1.20 SecondScreen 70

3.3.1.21 Детский пульт 70

3.3.1.22 Подключение дополнительного IPTV-приставки 71

3.3.1.23 Дополнительная файловая квота 71

3.3.1.24 Просмотр файлов c USB-носителя 72

3.3.1.25 Добровольная блокировка 72

3.3.2. Сторонние сервисы 73

3.3.3. Обращение в пользовательскую поддержку 73

3.3.4. Административные сценарии 73

3.3.4.1 Регистрация администратора 73

3.3.4.2 Авторизация администратора 74

3.3.4.3 Загрузка и настройка VoD 74

3.3.4.4 Создание и настройка ТВ-канала 75

3.3.4.5 Создание сериалов 75

3.3.4.6 Объединение VoD в пакеты 76

3.3.4.7 Объединение ТВ-каналов в пакеты 76

3.3.4.8 Ведение соглашений с правообладателями 76

3.3.4.9 Ведение справочников 77

3.3.5. Управление услугами 77

3.3.5.1 Создание и настройка услуги 77

3.3.5.2 Создание и настройка правил тарификации 78

3.3.5.3 Создание и настройка тарифного плана 78

3.3.5.4 Управление скидками и промоакциями 79

3.3.6. Управление клиентскими продуктами 79

3.3.6.1 Создание групп пользовательских продуктов 79

3.3.6.2 Обновление прошивок устройств/версий приложений 79

3.3.6.3 Создание групп пользователей 80

3.3.7. Сбор статистики её аналитика 81

3.3.7.1 Создание периодических отчетов 81

3.3.7.2 Формирование периодических отчетов 81

3.3.7.3 Выгрузка данных 82

3.3.7.4 Формирование аналитических отчетов по запросу 82

3.3.8. Проведение A/B-тестирования 83

3.3.9. Мониторинг 83

3.3.9.1 Мониторинг работоспособности аппаратного обеспечения 83

3.3.9.2 Мониторинг доступности внешних систем 83

3.3.9.3 Мониторинг работоспособности компонентов платформы 84

3.3.9.4 Мониторинг качества предоставления услуг 84

3.3.9.5 Задание параметров мониторинга 85

3.3.10. Поддержка пользователей 85

3.3.10.1 Создание заявки 85

3.3.10.2 Обработка заявки 86

3.3.10.3 Формирование отчетности 86

3.3.11. Управление системой рекомендаций 87

3.4. Основные требования к развитию решения ИТВ 87

3.4.1. Бизнес-требования 87

3.4.2. Финансово-юридические требования 87

3.4.3. Требования к процессу разработки 88

3.4.4. Требования к процессу тестирования 88

3.4.5. Требования к информационной безопасности 88

3.4.6. Требования к документированию 88

3.4.7. Требования к процессу поддержки 89

3.4.8. Требования к услугам 91

3.4.8.1 Контентные услуги 91

3.4.8.2 Сервисные услуги 92

3.4.8.3 Оплата услуг 94

3.4.8.4 Коммуникационные сервисы 94

3.4.9. Требования к клиентским продуктам 95

3.4.10. Требования к клиентскому оборудованию 95

3.4.11. Требования к Платформе интерактивного телевидения 2.0 95

3.4.11.1 Основные нефункциональные требования 95

3.4.11.2 Основные функциональные требования 96

3.5. Возможные риски 114

3.5.1. Правовые риски 115

3.5.1.1 Обработка персональных данных 115

3.5.1.2 Лицензионные риски 116

3.5.2. Операционные риски 117

3.5.2.1 Выход из строя оборудования 117

3.5.2.2 Рост нагрузки на платформу 118

3.5.2.3 Увольнение, временная нетрудоспособность ключевых специалистов 118

3.5.2.4 Получение несанкционированного доступа к персональной информации 119

3.5.2.5 Мошенничество 119

3.5.2.6 Права на контент 120

3.5.3. Технические риски 120

3.5.3.1 Смена технических решений 120

3.5.3.2 Некорректные, неактуальные и неполные данные 121

3.5.4. Организационные риски 121

3.5.4.1 Недостаточное и нестабильное финансирование 121

3.5.4.2 Риски изменения содержания проекта 121

3.5.5. Политические риски 122

3.5.5.1 Введение санкций 122

3.5.6. Валютный риск 123

3.5.7. Отраслевой риск 123

4. Техническое описание Платформы интерактивного   
телевидения 2.0 124

4.1. Концепция архитектуры платформы 124

4.2. Описание API пользовательских устройств 126

4.2.1. Цель создания 127

4.2.1.1 Описание 127

4.3. Описание информационной структуры 128

5. Приложение. Справочник API клиентских продуктов   
платформы «Интерактивное телевидение Ростелеком» 129

5.1. Порядок работы c API клиентских продуктов 129

5.1.1. Авторизация пользователя 131

5.1.2. Получение источников вещания ТВ-каналов 134

5.1.3. Получение описаний фильмов 135

5.1.4. Реализация логики работы с правами доступа 137

5.1.5. Обработка состояний учётной записи 137

5.1.6. Управление профилем 139

5.1.7. Медиапозиции 141

5.1.8. Работа с устройствами в группе 144

5.1.9. Нотификации сервера 147

5.1.10. Получение списка услуг 149

5.1.11. Покупка услуг 153

5.1.11.1 Абонементы 153

5.1.11.2 Разовые услуги 156

5.1.11.3 Сезоны и сериалы 157

5.1.12. Разделение услуг по типам устройств 159

5.1.13. Логика видимости услуг в интерфейсе устройств 160

5.1.14. Получение описаний ТВ-каналов 161

5.1.15. Получение описаний фильмов 162

5.1.16. Периодические услуги в SDP 162

5.1.17. Групповые услуги 163

5.1.18. Закачка купленных фильмов (оффлайн-просмотр) 164

5.1.18.1 Алгоритм загрузки фильма: 164

5.1.19. UID OTT-устройств 167

5.1.20. Работа с дельтами 167

5.1.21. Работа с платежными системами 169

5.2. Описние методов API клиентских приложений платформы «Интерактивное телевидение Ростелеком» 173

5.2.1. /ServiceAccount 173

5.2.1.1 /ServiceAccount/web\_authorize 173

5.2.1.2 /ServiceAccount/list\_services 175

5.2.1.3 /ServiceAccount/checkSession 178

5.2.1.4 /ServiceAccount/list\_purchased 179

5.2.1.5 /ServiceAccount/get\_status 181

5.2.1.6 /ServiceAccount/subscribe\_off 182

5.2.1.7 /ServiceAccount/subscribe\_on 183

5.2.1.8 /ServiceAccount/web\_subscribe 184

5.2.1.9 /ServiceAccount/get\_billing\_info 186

5.2.1.10 /ServiceAccount/list\_available\_profiles 187

5.2.1.11 /ServiceAccount/list\_all\_profiles 188

5.2.2. /VodPackage 189

5.2.2.1 /VodPackage/list\_purchased\_movies 189

5.2.2.2 /VodPackage/list\_purchased\_by\_acc 192

5.2.2.3 /VodPackage/list\_movies 196

5.2.2.4 /VodPackage/get\_delta 201

5.2.2.5 /VodPackage/list\_purchase\_period 202

5.2.3. /EstVodPackage 203

5.2.3.1 /EstVodPackage/list\_purchased\_by\_acc 203

5.2.4. /ContentBundlePackage 205

5.2.4.1 /ContentBundlePackage/list\_purchased\_bundles 205

5.2.4.2 /ContentBundlePackage/list\_purchased\_by\_acc 208

5.2.4.3 /ContentBundlePackage/list\_bundles 212

5.2.4.4 /ContentBundlePackage/get\_delta 213

5.2.4.5 /ContentBundlePackage/list\_purchase\_period 214

5.2.5. /EstContentBundlePackage 215

5.2.5.1 /EstContentBundlePackage/list\_purchased\_by\_acc 215

5.2.6. /SeriesPackage 216

5.2.6.1 /SeriesPackage/list\_purchased\_series 216

5.2.6.2 /SeriesPackage/list\_purchased\_by\_acc 218

5.2.6.3 /SeriesPackage/list\_series 220

5.2.6.4 /SeriesPackage/get\_delta 222

5.2.7. /ProductOffer 222

5.2.7.1 /ProductOffer/list\_services\_terminal 222

5.2.8. /WebPortal 225

5.2.8.1 /WebPortal/authorize\_san 226

5.2.8.2 /WebPortal/bind 226

5.2.8.3 /WebPortal/bind\_san 228

5.2.8.4 /WebPortal/login\_exists 229

5.2.8.5 /WebPortal/change\_password 230

5.2.8.6 /WebPortal/service\_account\_add\_request 230

5.2.8.7 /WebPortal/add\_service\_account 231

5.2.8.8 /WebPortal/set\_login\_and\_pwd 232

5.2.8.9 /WebPortal/service\_account\_phone\_get 232

5.2.8.10 /WebPortal/npvr\_create 233

5.2.9. /elk 234

5.2.9.1 /elk/authorize 234

5.2.10. /Terminal 235

5.2.10.1 /Terminal/send\_confirmation\_mail 235

5.2.11. /VideoMovie 236

5.2.11.1 /VideoMovie/web\_purchase 236

5.2.11.2 /VideoMovie/web\_purchase\_by\_abonement 237

5.2.11.3 /VideoMovie/purchase\_by\_franchise 237

5.2.12. /ContentBundle 238

5.2.12.1 /ContentBundle/web\_purchase 238

5.2.13. /Series 239

5.2.13.1 /Series/web\_purchase 239

5.2.14. /Content 240

5.2.14.1 /Content/web\_purchase 240

5.2.14.2 /Content/portal\_purchase 241

5.2.14.3 /Content/mediaPositionList 243

5.2.14.4 /Content/addMediaPosition 244

5.2.14.5 /Content/deleteMediaPosition 245

5.2.15. /Franchise 246

5.2.15.1 /Franchise/purchase\_no\_content 246

5.2.16. /ServiceUsage 246

5.2.16.1 /ServiceUsage/listByTransaction 246

5.2.16.2 /ServiceUsage/list\_purchase 247

5.2.17. /ServiceSpec 248

5.2.17.1 /ServiceSpec/list\_terminal\_type\_rules 248

5.2.18. /ott 250

5.2.18.1 /ott/get\_network\_config 250

5.2.18.2 /ott/ott\_authorize 251

5.2.18.3 /ott/sn\_authorize 254

5.2.18.4 /ott/email\_authorize 256

5.2.18.5 /ott/link\_email\_stb 258

5.2.18.6 /ott/billing\_check\_number 258

5.2.18.7 /ott/link\_to\_group 259

5.2.18.8 /ott/link\_to\_group\_by\_email 259

5.2.18.9 /ott/list\_group\_devices 260

5.2.18.10 /ott/get\_max\_device\_type\_count 261

5.2.18.11 /ott/unlink\_from\_group 262

5.2.18.12 /ott/set\_trusted 263

5.2.18.13 /ott/link\_snetwork 263

5.2.18.14 /ott/preregister\_email 264

5.2.18.15 /ott/email\_check\_token 264

5.2.18.16 /ott/email\_set\_password 265

5.2.18.17 /ott/get\_cur\_geolocation 265

5.2.18.18 /ott/update\_uid 266

5.2.18.19 /ott/get\_account\_additional\_billing\_info 266

5.2.18.20 /ott/set\_account\_additional\_billing\_info 267

5.2.18.21 /ott/checkPayment 267

5.2.18.22 /ott/checkContentPayment 268

5.2.19. /DeviceType 269

5.2.19.1 /DeviceType/list 269

5.2.20. /OttSNetwork 271

5.2.20.1 /OttSNetwork/list 271

5.2.21. /GenreDictionary 273

5.2.21.1 /GenreDictionary/list 273

5.2.22. /KaraokePackage 274

5.2.22.1 /KaraokePackage/search 274

5.2.22.2 /KaraokePackage/count 276

5.2.22.3 /KaraokePackage/list\_karaoke 278

5.2.23. /FranchisePackage 279

5.2.23.1 /FranchisePackage/list\_package 279

5.2.24. /PerformerDictionary 280

5.2.24.1 /PerformerDictionary/list 280

5.2.25. /KaraokeGenreDictionary 281

5.2.25.1 /KaraokeGenreDictionary/list 281

5.2.26. /MovieFolder 282

5.2.26.1 /MovieFolder/list\_movie\_folders 282

5.2.26.2 /MovieFolder/get\_dsc 283

5.2.27. /ChannelSubject 284

5.2.27.1 /ChannelSubject/list 284

5.2.28. /ChannelPackage 284

5.2.28.1 /ChannelPackage/list\_channels\_ott 284

5.2.28.2 /ChannelPackage/list\_incompatible\_packages 288

5.2.28.3 /ChannelPackage/simple\_list?field=internalId 289

5.2.29. /ProgramGenre 289

5.2.29.1 /ProgramGenre/list 289

5.2.30. /EPG 290

5.2.30.1 /EPG/get\_by\_Chnnl 290

5.2.30.2 /EPG/get\_by\_pkg 291

5.2.30.3 /EPG/get\_delta 292

5.2.30.4 /EPG/search 293

5.2.31. /ContentBundleGenre 295

5.2.31.1 /ContentBundleGenre/list 295

5.2.32. /AccessLevelDictionary 295

5.2.32.1 /AccessLevelDictionary/list 295

5.2.33. /AudioPID 296

5.2.33.1 /AudioPID/list 296

5.2.34. /AssetQuality 297

5.2.34.1 /AssetQuality/list 297

5.2.35. /Profile 299

5.2.35.1 /Profile/setFavouriteChannel 299

5.2.35.2 /Profile/setFavouriteChannelsList 300

5.2.35.3 /Profile/getFavouriteChannel 301

5.2.35.4 /Profile/update\_new 301

5.2.35.5 /Profile/insert\_new 305

5.2.35.6 /Profile/delete\_new 306

5.2.35.7 /Profile/copy 307

5.2.36. /Recommender 308

5.2.36.1 /Recommender/get\_vod\_by\_profile 308

5.2.37. /City 308

5.2.37.1 /City/list\_stb 308

5.2.38. /ForecaWeather 309

5.2.38.1 /ForecaWeather/get\_foreca\_forecast 309

5.2.39. /PromoSchedule 310

5.2.39.1 /PromoSchedule/list\_for\_interface 310

5.2.40. /Promo 311

5.2.40.1 /Promo/get 311

5.2.41. /PromoCodeGroup 312

5.2.41.1 /PromoCodeGroup/checkCode 312

5.2.41.2 /PromoCodeGroup/useCode 313

5.2.42. /RTUGC 313

5.2.42.1 /RTUGC/create 314

5.2.42.2 /RTUGC/setStatus 314

5.2.42.3 /RTUGC/listByAcc 315

5.2.42.4 /RTUGC/delete 316

5.2.43. /RTUGCQuota 317

5.2.43.1 /RTUGCQuota/get\_service\_account\_quotas 317

5.2.44. /RecordedProgram 317

5.2.44.1 /RecordedProgram/list\_by\_acc 318

5.2.44.2 /RecordedProgram/cancel 320

5.2.45. /OttContentAsset 320

5.2.45.1 /OttContentAsset/updateName 320

5.2.46. /OttPreviewAsset 321

5.2.46.1 /OttPreviewAsset/updateName 321

5.2.47. /SocialNetworks 322

5.2.47.1 /SocialNetworks/add\_content\_feedback 322

5.2.48. /ottSpy 322

5.2.48.1 /ottSpy 322

5.2.49. /asset 323

5.2.49.1 /asset/setQuality 323

5.2.50. /PaymentSystem 324

5.2.50.1 /PaymentSystem/createPayment 324

5.2.50.2 /PaymentSystem/createContentPayment 325

5.2.50.3 /PaymentSystem/confirmPayment 326

5.2.50.4 /PaymentSystem/getPaymentStatus 327

5.2.50.5 /PaymentSystem/listPayments 327

5.2.50.6 /PaymentSystem/createTopUpPayment 328

5.2.51. /change\_service\_account 328

5.2.51.1 /change\_service\_account 328

5.3. Параметры возвращаемых XML-данных 329

# Введение

## О документе

Этот документ содержит описание текущего решения интерактивного телевидения Ростелекома (решения ИТВ) и концепцию его развития (решение ИТВ 2.0).

Документ разбит на главы:

* Анализ текущего решения ИТВ. Глава содержит техническое описание серверной платформы «Интерактивное телевидение Ростелеком» как основного компонента решения, описание клиентских продуктов и услуг, предоставляемых решение.
* Концепция решения ИТВ 2.0. Глава содержит постановку задачи и основные технические и бизнес-требования развития решения ИТВ, сценарии использования решения; описывает потенциальные риски и способы их уменьшения.
* Техническое описание платформы «Интерактивное телевидение Ростелеком» 2.0. Глава содержит описание высокоуровневой архитектуры проектируемой платформы.

В конце документа содержатся приложения, подробнее раскрывающие содержимое отдельных разделов документа.

## Глоссарий

### Бизнес-термины и сокращения

CoD — Content on Demand — услуга предоставления контента по запросу пользователя.

DRM — Digital Rights Management — процедуры, программные или программно-аппаратные средства, которые ограничивают либо затрудняют действия с данными и контентом в электронной форме (копирование, модификацию, просмотр и т.д.), позволяют отследить такие действия.

EPG — Electronic Program Guide – электронная программа телепередач.

nPVR — Network Personal Video Recorder — услуга интерактивного телевидения, реализующая функции «видеомагнитофона»; позволяет выбрать и записать определённые ТВ-передачи.

OSS/BSS-система — Operation Support System/Business Support System — система поддержки внутренних бизнес-процессов компании.

PauseLive — услуга интерактивного телевидения, позволяющая поставить транслируемую телепередачу на паузу и продолжить воспроизведение с остановленного момента.

Second Screen — услуга позволяющая использовать дополнительное устройство (такое как планшет или смартфон) для получения и просмотра информации, связанной с показываемым основным устройством контентом.

TSTV (Timeshift) — услуга интерактивного телевидения, позволяющая записать транслируемую телепередачу и посмотреть её позже, просмотреть уже завершенную передачу.

VoD — Video on Demand — услуга предоставления видеоконтента по запросу пользователя.

Абонент — пользователь, с которым заключен договор об оказании услуг, имеющий уникальный идентификатор.

Абонентское устройство — клиентский продукт, предоставляющий пользователям услуги решения, состоящий из ТВ-приставки и установленного на нём программного обеспечения (прошивки).

Администратор — сотрудник оператора платформы интерактивного телевидения, обеспечивающий её функционирование.

Администраторская роль — тип учётной записи администратора, имеющей определённые права доступа к системе управления платформой для выполнения своих должностных обязанностей; определены следующие роли:

* контент-менеджер — управление контентом;
* продуктолог — управление услугами, тарификацией;
* сотрудник мониторинга — мониторинг работоспособности платформы и качества предоставления услуг;
* сотрудник поддержки — техническая поддержка пользователей и администраторов платформы, управление абонентами;
* аналитик — формирование отчетности, анализ данных, управление системой рекомендаций;
* системный администратор — управление конфигурациями, управление настройками платформы, управление клиентскими продуктами.

АСР — Автоматизированная система расчетов, осуществляющая поддержку бизнес-процессов биллинга; предназначена для автоматизации процессов регистрации и учёта физических и юридических лиц в качестве абонентов, учёта оказанных услуг связи, выполнения расчётов стоимости предоставленных услуг, печати платежных документов, приема, регистрации и контроля платежей за предоставленные услуги, информационно-справочного обслуживания абонентов, а также формирования аналитической и статистической отчётности по оказанным услугам, произведённой по ним оплате, финансовому состоянию лицевых счетов абонентов; содержит информацию о клиентах, операторах, договорах, контактных лицах.

Биллинг — комплекс процессов и решений, предназначенных для сбора информации об использовании услуг, их тарификации, выставление счетов абонентам, обработку платежей.

Воронка продаж — принцип распределения клиентов по стадиям процесса продаж от первого контакта до покупки.

ЕКП — Единый контентный портал — веб-сайт, предоставляющий пользователям услуги решения посредством браузера.

ЕЛК — Единый личный кабинет — информационная система, предоставляющая клиентам-физическим лицам ПАО «Ростелеком» способ удаленного управления услугами, настройками и пользовательскими данными.

ИСПД — информационная система персональных данных.

ИТВ — набор услуг интерактивного телевидения ПАО «Ростелком».

Клиент — физическое или юридическое лицо пользующееся решением в рамках договорных отношений или без них.

Клиентский продукт — абонентское устройство или клиентское приложение, размещаемое у клиента и предоставляющее ему услуги решения.

Клиентское приложение — программное обеспечение, устанавливаемое на собственное устройство клиента и предоставляющее ему услуги решения.

Контент — совокупность различных медиаданных, предоставляемых пользователям в рамках оказания услуг.

Макрорегион (МРФ) — территориальная единица предоставления услуг решения, соотвествует структуре макрорегиональных филиалы ПАО «Ростелеком».

Модель оказания услуги — способ предоставления услуги пользователю в клиентском продукте.

Мультискрин — передача контента и услуг, а также сохраненных позициях просмотра контента между различными клиентскими продуктами одного абонента.

ОИС — объект интеллектуальной собственности.

Пакет услуг — набор услуг, которые предоставляются абонентам совместно.

Пользователь — лицо или организация, которое использует решение.

Проприетарная технология — технология, являющееся частной собственностью авторов или правообладателей.

Профиль — набор настроек, параметров и других данных привязанных к определенному пользователю.

Родительский контроль — услуга, позволяющая пользователю ограничивать доступ к доступным контенту и услугам.

РТК, Ростелеком — ПАО «Ростелеком».

Тариф — условия и параметры оплаты определённой услуги/пакета услуг.

Транскодирование — процесс конвертации одного медиафайла в другой, отличающийся от исходного форматом и/или параметрами потока.

Услуга — деятельность Ростелекома, направленная на удовлетворение определенной потребности потребителя.

ШПД — услуга предоставления широкополосного (высокоскоростного) доступа в интернет.

### Технические определения и сокращения

AAC — Advanced Audio Coding — формат аудиофайлов и алгоритм кодирования аудиоданных.

API — Application Programming Interface — интерфейс программирования приложений; набор готовых классов, процедур, функций, структур и констант, предоставляемых сервисом для его использования другими сервисами.

AVC — Advanced Video Coding, H.264 — лицензируемый стандарт кодирования видеоизображения.

CDN — Content Delivery Network — географически распределённая сетевая инфраструктура, позволяющая оптимизировать доставку контента конечным пользователям.

CSV — Comma-Separated Values — текстовый формат представления табличных данных.

ETL-процедуры — Extract, Transform and Load — процессы извлечения из внешних источников, трансформации и загрузки данных в хранилищах данных.

HDMI — High Definition Multimedia Interface — интерфейс для передачи цифровых видеоданных высокой четкости и многоканальных аудиосигналов.

HLS — HTTP Live Streaming — протокол потоковой передачи медиаданных посредством прикладного сетевого протокола HTTP.

IGMP — Internet Group Management Protocol — протокол управления групповой (multicast) передачей данных в IP-сетях, используется для организации сетевых устройств в группы.

IPTV — Internet Protocol Television — набор технологий передачи телевизионного контента в сетях передачи данных по протоколу IP посредством сетей, управляемых оператором связи.

ITL — Interested Transaction List — структура базы данных Oracle, хранящая информацию о списке транзакций, блокирующих определенные строки блока данных («заинтересованных» в этих строках).

JSON — JavaScript Object Notation — структурированный текстовый формат обмена данными, состоящими из пар ключ-значение и упорядоченных наборов значений.

MAC-адрес — уникальный идентификатор сетевого устройства в сети передачи данных.

MPEG-TS — MPEG transport stream — медиаконтейнер, предназначенный для хранения и передачи аудио- и видеоданных.

Multicast — способ передачи данных при котором получателям сетевого пакета является группа клиентов.

NFS — Network File System — протокол сетевого доступа к файловым системам.

PIM — Protocol Independent Multicast — набор протоколов многоадресной маршрутизации, работающий на основе традиционных маршрутных протоколов.

PLC-адаптер — адаптер для передачи данных через сеть электрического питания.

RTCP — Real-Time Transport Control Protocol — сетевой протокол транспортного уровня для получения статистики и управления передачей информации в рамках RTP-подключения; используется совместно с протоколом RTP.

RTP — Real-time Transport Protocol — сетевой протокол прикладного уровня, предназначенный для передачи данных в реальном времени; обычно используется совместно с протоколом RTSP.

RTSP-вещание — процесс передачи медиаданных при помощи сетевых протоколов RTSP, RTP и RTCP.

RTSP — Real-time Streaming Protocol — протокол потоковой передачи данных, позволяющий клиенту удалённо управлять потоком данных с сервера.

STB — Set-top Box — телевизионная приставка (ресивер), обеспечивающая доступ к услугам цифрового телевидения.

UDP — User Datagram Protocol — транспортный сетевой протокол передачи данных в IP-сетях; протокол обеспечивает передачу пакетов данных без контроля их получения, используется для передачи с малой задержкой данных не критичных к потере части содержимого.

UID — уникальный идентификатор устройства; для клиентских продуктов STB и SmartTV — MAC-адрес, для мобильных приложений — UUID приложения (Universally Unique Identifier), для веб-клиента — идентификатор сессии.

Unicast — способ передачи данных при котором получателям сетевого пакета является один определённый клиент.

XML — eXtensible Markup Language — расширяемый язык разметки, структурированный текстовый формат обмена данными.

Витрина данных — срез хранилища данных, представляющий собой массив тематической, узконаправленной информации.

Медиаконтейнер — формат хранения (файла) или передачи (потока) медиаданных.

ОТТ — Over-the-Top — набор технологий передачи цифрового видеоконтента через интернет посредством сетей, неуправляемых оператором связи.

Токен — электронный маркер, уникальным образом идентифицирующих пользователей, устройства, сессии и т.д.

ЦОД — центр обработки данных.

### Системы и компоненты

OEBS R12 — Oracle E-Business Suite Release 12 — комплекс программного обеспечения производства компании Oracle, включающий функциональные блоки ERP, CRM, PLM, предназначен для автоматизации основных направлений деятельности предприятий.

АСР «Онима» — автоматизированная система расчетов компании «Онима».

АСР «Петер-сервис» — автоматизированная система расчетов компании «Петер-сервис».

АСР «Старт.IP» — автоматизированная система расчетов компании «Старт2Ком».

ЕССП — Единая система приема платежей — централизованная система приема платежей ПАО «Ростелеком».

Панель управления платформой — компонент СУП решения ИТВ 2.0, предоставляющий администраторам набор инструментов для управления работой платформы в виде веб-приложения.

САУ «Лира» — система абонентского учёта «Лира» — информационная система ПАО «Ростелеком», обслуживающая абонентов решения интерактивного телевидения макрорегионального филиала «Урал».

СДК — Система доставки контента – набор компонентов платформы «Интерактивное телевидение Ростелеком» и платформы «Интерактивное телевидение Ростелеком» 2.0, отвечающих за управление контентом и услугами, предоставляющих графический интерфейс администраторам платформы, обеспечивающих взаимодействие платформы и пользовательских продуктов.

СУК — Система управления контентом – набор компонентов платформы «Интерактивное телевидение Ростелеком», отвечающих за доставку контента пользователям посредством различных механизмов передачи данных.

СУП — Система управления платформой — набор компонентов платформы «Интерактивное телевидение Ростелеком» 2.0, отвечающих за управление продуктами, контентом, оплатой, абонентами, системой рекомендаций; сбор статистики и аналитики; мониторинг и взаимодействие с технической поддержкой.

Узел доступа — архитектурный компонент платформы «Интерактивное телевидение Ростелеком», осуществляющий доставку контента пользователям части макрорегиона.

Узел макрорегиона — архитектурный компонент платформы «Интерактивное телевидение Ростелеком», обслуживающий пользователей макрорегиона.

Центральный узел — архитектурный компонент платформы «Интерактивное телевидение Ростелеком», обслуживающий работу всей платформы и узлов макрорегинов.

# Анализ текущего решения Ростелеком ИТВ

## Общее описание решения

Решение ИТВ базируется на работе серверной Платформы интерактивного телевидения (платформы «Интерактивное телевидение Ростелеком» текущей версии) и пользовательских продуктов.

Платформа «Интерактивное телевидение Ростелеком» и пользовательские продукты обеспечивают предоставление пользователям услуг решения и показ контента, осуществление платежей посредством внешних системам биллинга и процессинга.

Платформа «Интерактивное телевидение Ростелеком» обеспечивает:

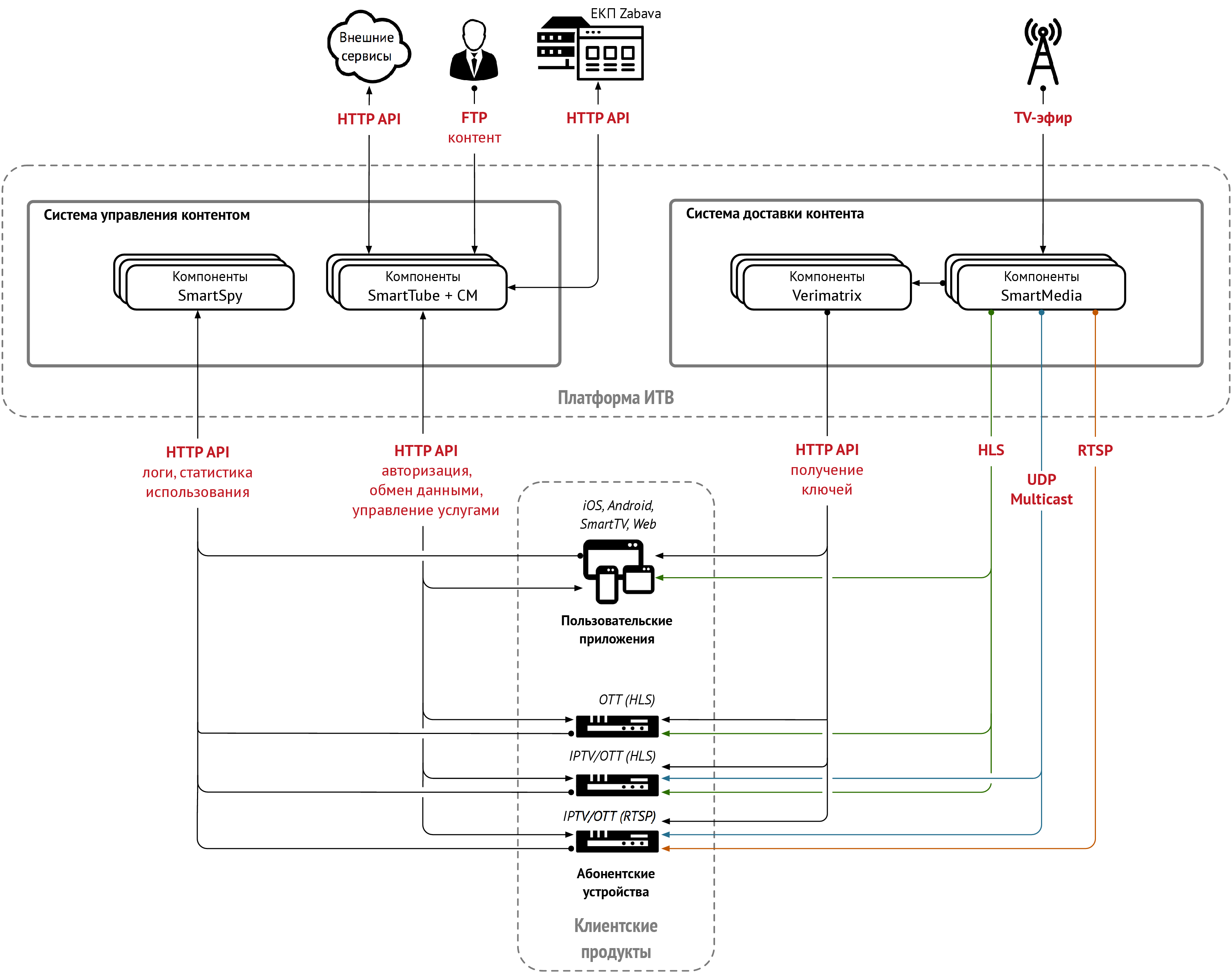
* приём и передачу контента телеканалов (федеральных и местных) пользователям;
* запись и временное хранение записанного контента телеканалов;
* приём и хранение файлов контента от правообладателей;
* транскодирование файлов контента;
* шифрование файлов контента и контента телеканалов;
* трансляцию контента от правообладателей пользователям;
* авторизацию и регистрацию пользователей, абонентов и пользовательских продуктов;
* отправку нотификационных сообщений пользователям;
* взаимодействие с внешними системами (АСР, OSS/BSS, САУ, CDN);
* хранение данных о пользователях, контенте и услугах;
* приём и хранение логов и статистической информации от пользовательских продуктов.

Пользовательские продукты обеспечивают:

* получение контента от платформы «Интерактивное телевидение Ростелеком» и отображение его пользователю;
* получение команд и данных от платформы «Интерактивное телевидение Ростелеком»;
* отправку команд и данных платформе «Интерактивное телевидение Ростелеком»;
* отображение пользовательского интерфейса на основе данных и контента, полученных от платформы «Интерактивное телевидение Ростелеком».

## Состав решения

Решение ИТВ состоит из серверной платформы «Интерактивное телевидение Ростелеком» и набора клиентских продуктов: пользовательских приложений и абонентских устройств.



Состав решения ИТВ

Серверная платформа «Интерактивное телевидение Ростелеком» состоит из набора компонентов — программного обеспечения (сервисов, программных продуктов и комплексов) развернутом на аппаратном обеспечении, связанным в единую информационную сеть.

Компоненты платформы «Интерактивное телевидение Ростелеком» можно объединить в две группы по их функциональному назначению:

* Система доставки контента — набор компонентов, отвечающих за доставку контента пользователям посредством различных механизмов передачи данных.
* Система управления контентом — набор компонентов, отвечающих за управление продуктами, контентом, оплатой, абонентами, системой рекомендаций; сбором статистики и аналитики.

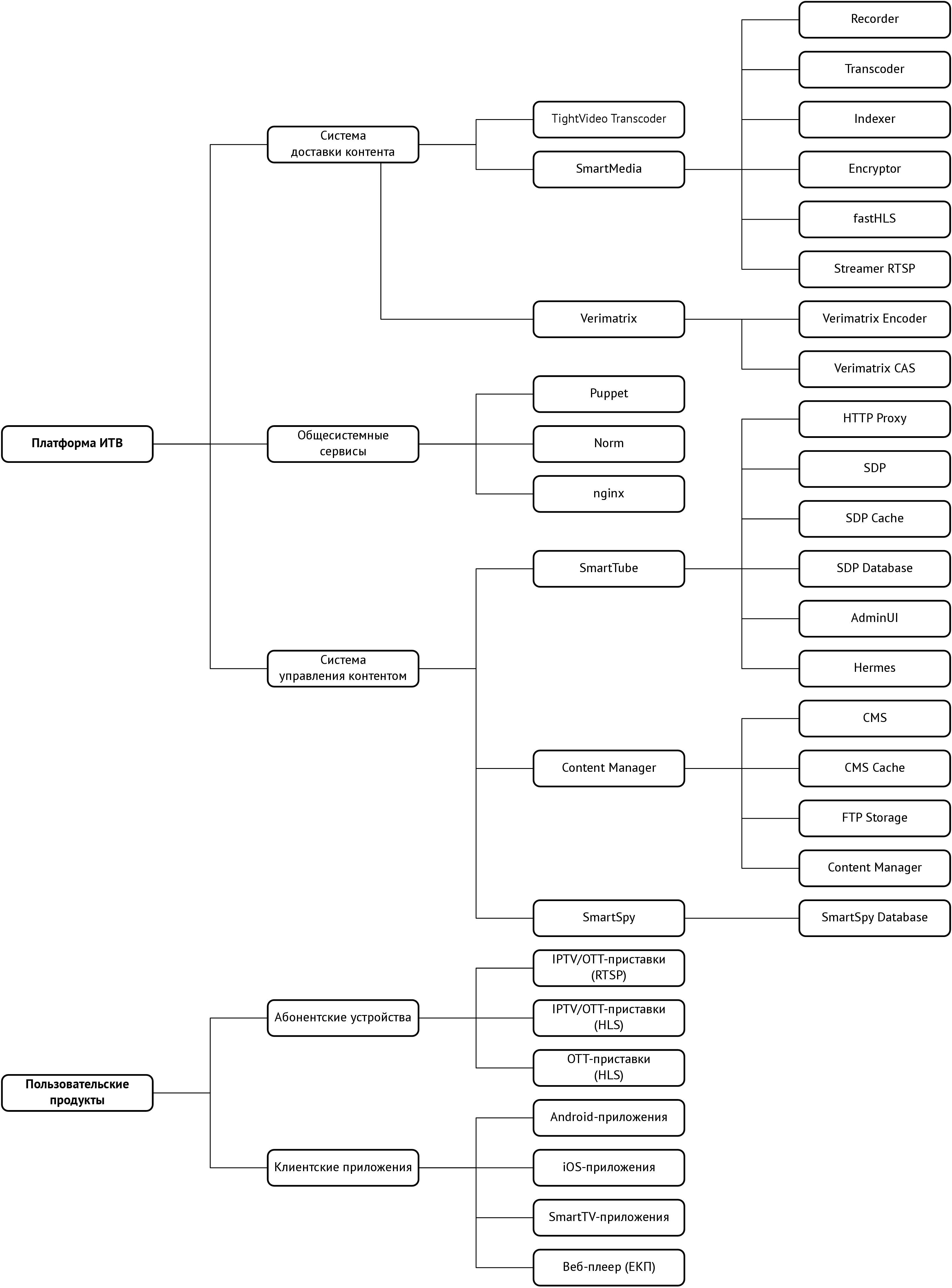
Системы компонентов в свою очередь делятся на подсистемы — наборы компонентов, выполняющих единые функции; подсистемы компонентов разделяются на отдельные компоненты и группы компонентов:

* Система управления контентом:
  + Подсистема SmartTube — осуществляет управление работой, контентом и данными платформы:
    - Группа SmartTube SDP (Service Delivery Platform) — отвечает за хранение, доступ и управление данных;
    - SmartTube Hermes — отвечает за отправку сообщений и нотификаций пользователям.
    - SmartTube HTTP Proxy — кеширует запросы ко внешними системам;
    - SmartTube AdminUI — предоставляет администратору графический интерфейс управления платформой (в виде веб-сайта).
  + Подсистема SmartSPY — осуществляет сбор и агрегацию логов и статистики использования с клиентских продуктов.
  + Подсистема Content Manager — осуществляет приём и управление файлами CoD-контента.
* Система доставки контентом:
  + Подсистема SmartMedia — осуществляет хранение, обработку и передачу контента;
  + Подсистема Verimatrix — осуществляет DRM-шифрование/дешифрование контента.
  + Компонент TightVideo Transcoder — осуществляет транскодирование ТВ-эфира.

Кроме компонентов систем в состав платформы также входят общесистемные сервисы, такие как nginx (кеширующий прокси-сервер), Puppet (централизованная система обновления программного обеспечения) и Norm (сервер распространения прошивок абонентских устройств).

Клиентские продукты делятся на два вида:

* Клиентские приложения — программное обеспечение для установки на устройства (планшеты, смартфоны) клиентов, а также веб-сайт ЕКП (Zabava):
  + iOS-приложения для планшетов;
  + Android-приложения для планшетов;
  + Smart-TV-приложения;
  + веб-сайт ЕКП.
* Абонентские устройства — ТВ-приставки и установленные на них программного обеспечения:
  + IPTV/OTT-приставки (RTSP);
  + IPTV/OTT-приставки (HLS);
  + OTT-приставки (HLS, для абонентов не Ростелекома).



Иерархия компонентов решения ИТВ

## Техническое описание Платформы интерактивного телевидения

### Назначение

Платформа «Интерактивное телевидение Ростелеком» предназначена для предоставления пользователям услуг интерактивного телевидения посредством пользовательских продуктов; управления контентом, услугами и данными пользователей; взаимодействия с внешними системами.

Платформа «Интерактивное телевидение Ростелеком» работает во взаимодействии с клиентскими продуктами, осуществляющими отображение контента и данных пользователям; принимающими от пользователей команды ввода и передающими их в платформу; отправляющими в платформу данные о событиях (логи) и системные сообщения.

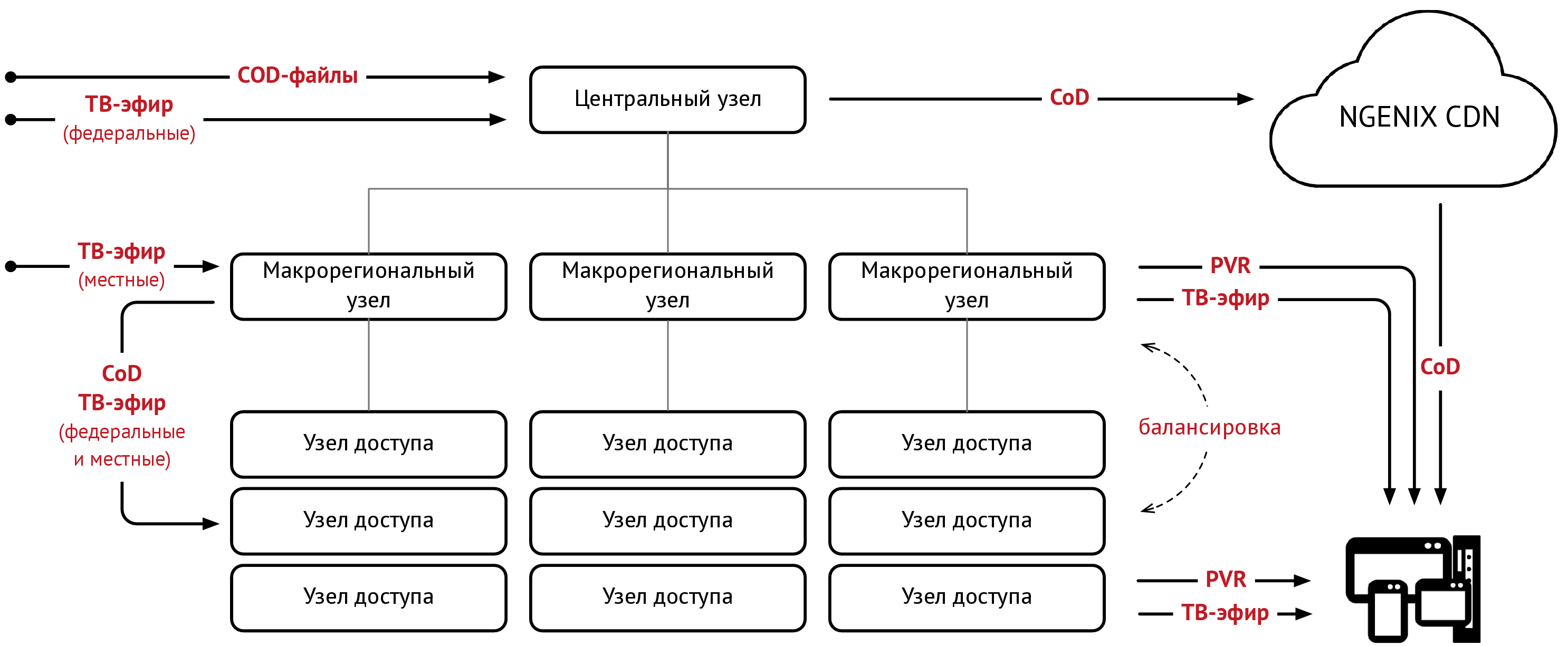
Платформа «Интерактивное телевидение Ростелеком» предоставляет администраторам инструменты управления контентом, данными и параметрами работы решения.

### Возможности платформы

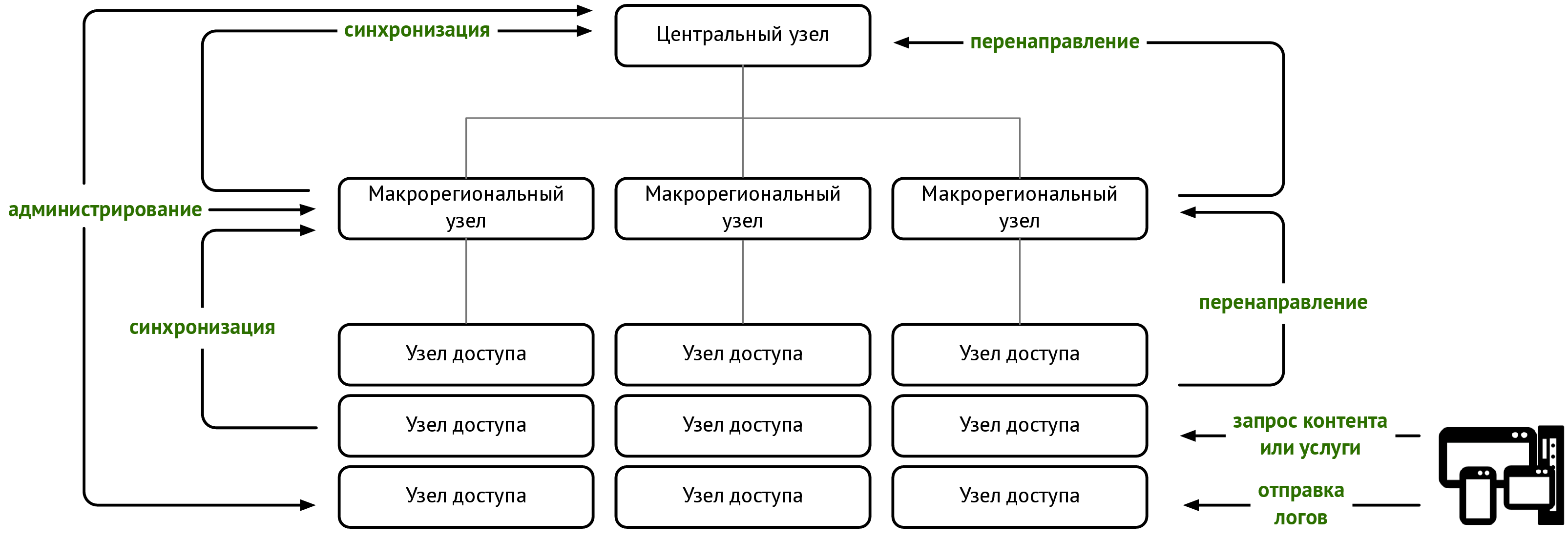
*[Список основных функции, реализуемых платформой]*

### Архитектура

Архитектура платформы «Интерактивное телевидение Ростелеком» состоит из набора компонентов систем доставки и управления контентом.



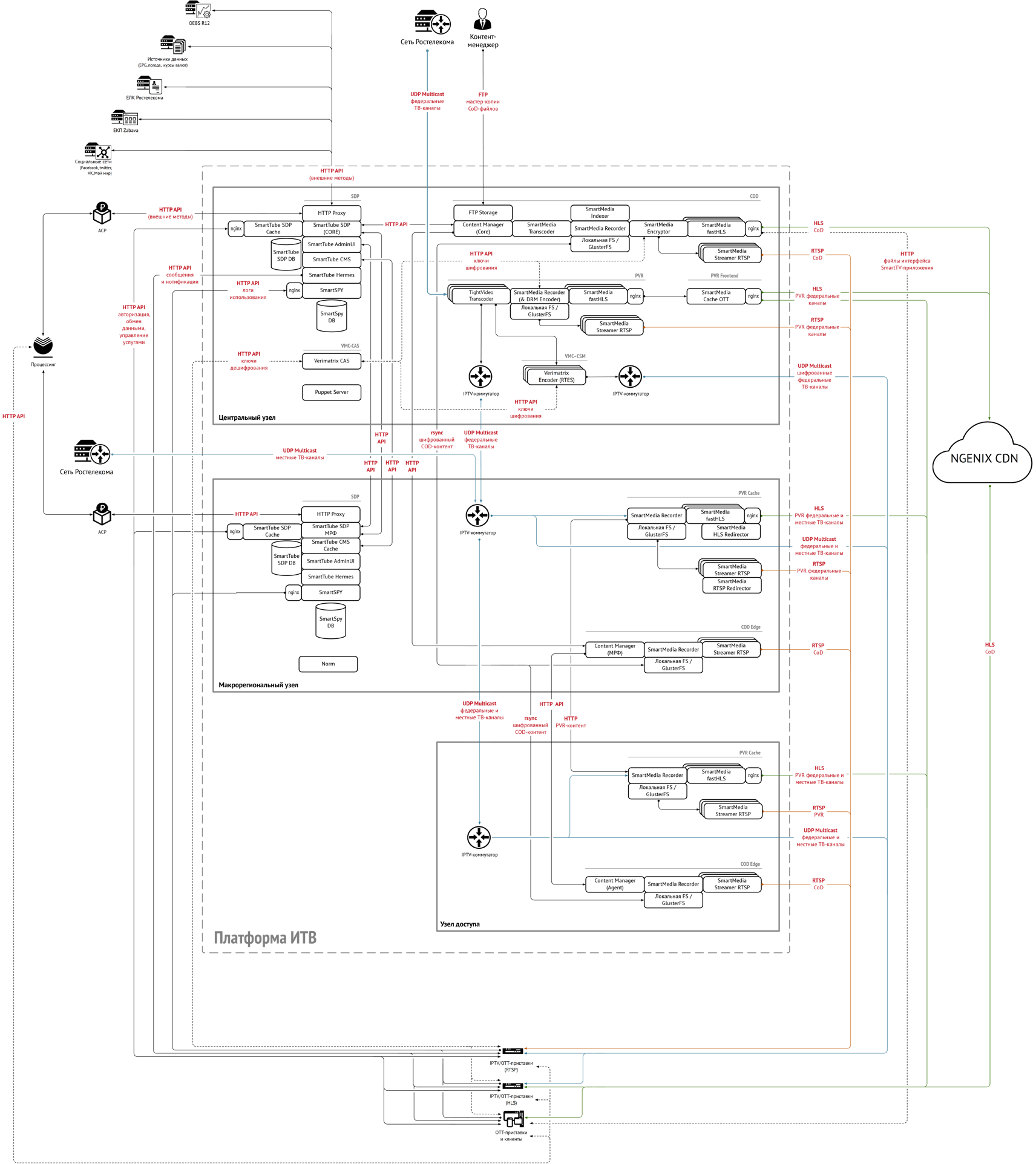
Уровни узлов (контент)



Уровни узлов (данные и запросы)

Компоненты располагаются в нескольких территориальных группах трёх уровней:

* центрального узла — получение контента, управление услугами, агрегация данных, доставка контента.
* макрорегионов — управление услугами, доставка контента.
* узлов доступа — доставка контента.



Логическая архитектура платформы «Интерактивное телевидение Ростелеком»

Некоторые компоненты дублируются между уровнями для увеличения надёжности и ускорения ответов на запросы пользователей; часть контента регулярно синхронизируется между компонентами разных уровней.

Платформа содержит компоненты, отвечающие за доставку контента:

* посредством HLS-протокола (VoD и прямой эфир);
* посредством RTSP-протокола (VoD и прямой эфир);
* посредством IPTV-инфраструктуры — UDP multicast (только прямой эфир).

Платформа получает медиаконтент из двух источников:

* посредством загрузки файлов в FTP-хранилище администраторами;
* посредством записи телеэфира прямого вещания.

Загруженные файлы транскодируются, шифруются и хранятся: файловые хранилища разных уровней (файловые хранилища компонентов SmartMedia Recorder) периодические синхронизируются.

Телеэфир, полученный из системы цифровой телетрансляции Ростелекома, передаётся посредством сетевой IPTV-инфраструктуры соответствующим клиентским продуктам, а также сохраняется в хранилищах в виде файлов — для использования в рамках услуг nPVR, TSTV, PauseLive.

Платформа «Интерактивное телевидение Ростелеком» осуществляет приём ТВ-эфира на двух уровнях:

* центральный узел — эфир федеральных и общедоступных каналов;
* макрорегиональный узел — эфир местных телеканалов (предоставляется только пользователям соответствующего макрорегиона).

Платформа «Интерактивное телевидение Ростелеком» может осуществлять шифрование получаемого контента: как видеофайлов, так и ТВ-эфира; контент хранится и передаётся зашифрованным; для расшифрования клиентские продукты запрашивают DRM-ключи у компонента Verimatrix CAS центрального узла.

#### Развертывание компонентов

Компоненты платформы «Интерактивное телевидение Ростелеком» и дополнительные сервисы, обеспечивающие их работу, разворачиваются на виртуальных/физических машинах при помощи системы Puppet. Каждой такой машине присваивается уникальный сетевой идентификатор — hostname.

Hostname для машины назначается по схеме rt-<mrf>-<rf><sub>-<role><num>, где:

* <mrf> – обозначение макрорегионального филиала:
  + core — центральный узел;
  + ott — сегмент OTT центрального узла;
  + mos — МРФ “Москва”;
  + ct — МРФ “Центр”;
  + nw — МРФ “Северо-Запад”;
  + ural — МРФ “Урал;
  + sib — МРФ “Сибирь”;
  + dv — МРФ “Дальний Восток”;
  + sth — МРФ “Юг”;
  + vlg — МРФ “Волга”.
* <rf> – обозначение регионального филиала или узла доступа (например, nn — Нижний Новогород)
* <sub> – цифровое обозначение субрегиона (например, 01)
* <role> — роль сервера:
  + fe — Frontend;
  + ottfe — OTT Frontend;
  + sdp — SmartTUBE SDP;
  + pub — Content Manager;
  + cod — SmartMEDIA VoD;
  + db — сервер баз данных Oracle;
  + stor — хранилище данных;
  + pvr, mstor, mfe — SmartMEDIA nPVR;
  + clr — Collector ;
  + vmx-csm-\* – Verimatrix CSM;
  + vmx-rtes – Verimatrix RTES;
  + vmx-vpp – Verimatrix VPP;
  + spy — база данных SPY;
* <num> – номер сервера в кластере (например, 01).

### Форматы данных

#### Информационные сущности

*[Описание основных информационных сущностей]*

#### Файлы контента

##### Мастер-копии

##### Транскодированное видео

#### Структура баз данных

*[ERD-диаграмма, описание основных таблиц]*

#### Форматы данных внутреннего API

*[Форматы обмена данных между компонентами]*

#### Форматы данных внутреннего API

*[Форматы обмена данных со внешними системами]*

### API взаимодействия с клиентскими продуктами

API клиентских продуктов обеспечивает взаимодействие абонентских приставок и клиентских приложений с серверами SmartTube (SDP и SPY) посредством HTTP/HTTPS-запросов и ответов.

API клиентских продуктов содержит набор методов, собранных в группы по функциональному назначению. Детальное описание порядка работы с API клиентских продуктов и доступных методов приведено в приложении [*«Справочник API клиентских приложений платформы «Интерактивное телевидение Ростелеком»»*](#приложение.-справочник-api-клиентских-продуктов-платформы-итв)).

В GET-запросах к серверу данные передаются в виде URL-параметров. Сервер отвечает на запросы HTTP-сообщениями, содержащими данные в XML-формате.

В запросах к серверу используется стандартный механизм HTTP-сессий. При запросе авторизации сервер сохраняет ID сессии в базу данных для авторизовавшегося клиентского продукта .

Если в какой-то момент данная учетная запись авторизуется повторно и для неё генерируется новый ID сессии, он перезаписывается в БД.

После этого запросы от устройства со старым ID сессии SDP не обрабатывает и возвращает ошибку (HTTP-код 403). Таким образом, сервер принимает и обслуживает запросы только от того устройства, которое авторизовалось последним (последний записанный в базу данных ID сессии). Устройствам с устаревшим ID сессии сервер возвращает ошибку 403.

API-методы взаимодействия серверов SmartTube SDP и клиентских продуктов можно разделить на наборы в зависимости от сценариев их использования.

Использование методов из различных групп в одном наборе сценария позволяет производить операции, необходимые для выполнения определенных задач: управления учетной записью, подключения услуг и т.д.

#### Методы управления учетными записями

Набор методов, использующихся для авторизации, проверки состояния, получения списков подключенных услуг и других параметров учетной записи пользователя.

* ServiceAccount/web\_authorize;
* ServiceAccount/list\_services;
* ServiceAccount/checkSession;
* ServiceAccount/list\_purchased;
* VodPackage/list\_purchased\_movie;
* VodPackage/list\_purchased\_by\_acc;
* EstVodPackage/list\_purchased\_by\_acc;
* ContentBundlePackage/list\_purchased\_bundles;
* ContentBundlePackage/list\_purchased\_by\_acc;
* EstContentBundlePackage/list\_purchased\_by\_acc;
* SeriesPackage/list\_purchased\_series;
* SeriesPackage/list\_purchased\_by\_ac;
* ServiceAccount/get\_status;
* ProductOffer/list\_services;
* ProductOffer/list\_services\_terminal;
* WebPortal/authorize\_san;
* WebPortal/bind;
* WebPortal/bind\_san;
* WebPortal/login\_exists;
* WebPortal/change\_password;
* WebPortal/service\_account\_add\_request;
* WebPortal/add\_service\_account;
* WebPortal/set\_login\_and\_pwd;
* WebPortal/service\_account\_phone\_get;
* elk/authorize;
* Terminal/send\_confirmation\_mail.

#### Методы тарификации услуг

Набор методов для получения списка покупок пользователя, оформления подписки на услуги, проверки состояния оплаты и других действий, связанных с приобретением контента.

* VideoMovie/web\_purchase;
* ContentBundle/web\_purchase;
* Series/web\_purchase;
* Content/web\_purchase;
* Content/portal\_purchas;
* VideoMovie/web\_purchase\_by\_abonement;
* Franchise/purchase\_no\_content;
* VideoMovie/purchase\_by\_franchise;
* ServiceAccount/subscribe\_off;
* ServiceAccount/subscribe\_on;
* ServiceAccount/web\_subscribe;
* ServiceUsage/listByTransaction;
* ServiceUsage/list\_purchase;
* ServiceSpec/list\_terminal\_type\_rules;
* ServiceAccount/get\_billing\_info.

#### ОТТ-методы

Набор методов для работы с ОТТ-устройствами: авторизации, привязки к пользовательскому профилю, подключения к группам и т.д.

* ott/get\_network\_config;
* ott/ott\_authorize;
* ott/sn\_authorize;
* ott/email\_authorize;
* ott/link\_email\_stb;
* ott/billing\_check\_number;
* ott/link\_to\_group;
* ott/link\_to\_group\_by\_email;
* ott/list\_group\_devices;
* ott/get\_max\_device\_type\_count;
* ott/unlink\_from\_group;
* ott/set\_trusted;
* ott/link\_snetwork;
* ott/preregister\_email;
* ott/email\_check\_token;
* ott/email\_set\_password ;
* ott/get\_cur\_geolocation;
* ott/update\_uid;
* DeviceType/list;
* OttSNetwork/list ;
* ott/get\_account\_additional\_billing\_info;
* ott/set\_account\_additional\_billing\_info ;
* ServiceAccount/get\_status.

#### Методы для работы с CoD

Набор методов для работы с контентом по запросу: списками и описаниями сериалов и фильмов, группировки контента и т.д.

* GenreDictionary/list;
* VodPackage/list\_movies;
* VodPackage/list\_rules;
* VodPackage/get\_delta;
* KaraokePackage/search;
* KaraokePackage/count;
* KaraokePackage/list\_karaoke;
* FranchisePackage/list\_franchise;
* PerformerDictionary/list;
* KaraokeGenreDictionary/list;
* VodPackage/list\_purchase\_period;
* MovieFolder/list\_movie\_folders;
* MovieFolder/get\_dsc.

#### Методы для работы с ТВ-каналами

Набор методов для получения информации о ТВ-каналах.

* ChannelSubject/list;
* ChannelPackage/list\_channels\_ott;
* ChannelPackage/list\_incompatible\_packages;
* ChannelPackage/simple\_list?field=internalId.

#### Методы для работы с избранными ТВ-каналами

Набор методов для работы c избранными ТВ-каналами.

* Profile/setFavouriteChannel;
* Profile/setFavouriteChannelsList;
* Profile/getFavouriteChannelsList.

#### Методы для работы с EPG

Набор методов для работы с электронной программой передач.

* ProgramGenre/list;
* EPG/get\_by\_Chnnl;
* EPG/get\_by\_pkg;
* EPG/get\_delta;
* EPG/search.

#### Методы для работы с сериалами и сезонами

Набор методов для получения списков сериалов, сезонов и серий.

* ContentBundleGenre/list;
* ContentBundlePackage/list\_bundles;
* ContentBundlePackage/list\_rules;
* ContentBundlePackage/get\_delta;
* SeriesPackage/list\_series;
* SeriesPackage/list\_rules;
* SeriesPackage/get\_delta;
* ContentBundlePackage/list\_purchase\_period.

#### Методы для получения вспомогательных справочников

Набор методов для получения справочников, данные из которых могут использоваться в других запросах.

* AccessLevelDictionary/list;
* AudioPID/list;
* AssetQuality/list.

#### Методы работы с профилями клиентских приложений

Набор методов, с помощью которых осуществляется управление профилями клиентских приложение: изменение, удаление, получение списка профилей и т.д.

* ServiceAccount/list\_available\_profiles;
* ServiceAccount/list\_all\_profiles;
* Profile/update\_new;
* Profile/insert\_new;
* Profile/delete\_new;
* Profile/copy.

#### Методы для работы с рекомендациями

Метод для получения списка рекомендованных фильмов.

* Recommender/get\_vod\_by\_profile.

#### Методы для работы с информационными сервисами

Методы для получения дополнительной информации, полученной из внешних сервисов данных: списка городов и прогноза погоды.

* City/list;
* City/list\_stb;
* ForecaWeather/get\_foreca\_forecast.

#### Методы для работы с рекламой

Набор методов для работы с рекламными материалам и условиями их показа.

* PromoSchedule/list\_for\_interface;
* Promo/get.

#### Методы для работы с промокодами

Набор методов для проверки и активации промокодов.

* PromoCodeGroup/checkCode.
* PromoCodeGroup/useCode.

#### Методы для работы с UGC

Набор методов для работы с пользовательским контентом: загрузки, получения списков уже загруженного контента и информации о квоте на пользовательский контент.

* RTUGC/create;
* RTUGC/setStatus;
* RTUGC/listByAcc;
* RTUGC/delete;
* RTUGCQuota/get\_service\_account\_quotas.

#### Методы для работы с nPVR

Методы для работы с записями телевизионных передач.

* RecordedProgram/list\_by\_acc;
* WebPortal/npvr\_create;
* RecordedProgram/cancel;
* OttContentAsset/updateName;
* OttPreviewAsset/updateName.

#### Методы для работы с сохраненными позициями просмотра

Набор методов для работы с сохраненными позициями просмотра: просмотра списка созданных позиций просмотра, их создания и удаления.

* Content/mediaPositionList;
* Content/addMediaPosition;
* Content/deleteMediaPosition.

#### Методы для работы с социальными сетями

Метод для оценки контента в базе данных для заданного профиля.

* SocialNetworks/add\_content\_feedback.

#### Методы сбора статистики

Метод для передачи на сервер SDP сообщения о событиях в клиентском приложении, связанных с просмотром контента.

* ottSpy.

#### Методы для приема обратной связи от абонентов

Метод для передачи на сервер данных о низком качестве видеофрагментов.

* asset/setQuality.

#### Методы для работы с платежной системой

Набор методов для проведения платежей за контент и проверки их состояния.

* ott/checkPayment;
* ott/checkContentPayment;
* PaymentSystem/createPayment;
* PaymentSystem/createContentPayment;
* PaymentSystem/confirmPayment;
* PaymentSystem/getPaymentStatus;
* PaymentSystem/listPayments;
* PaymentSystem/createTopUpPayment.

#### Справочник методов, вызываемых со стороны SDP

Метод, вызываемый SDP на ЕКП Zabava для передачи нотификации об изменении.

* change\_service\_account.

### Взаимодействие с внешними системами

* Биллинг Ростелекома (АСР «Петер-Сервис» ?).
* Сервисы Ростелекома: САУ «Лира», ЕССА, OSS «Ростелеком», ЕКП, ЕСПП.
* Ngenix.
* Соцсети, яндекс-карты и т.д.
* Сайт Забава.

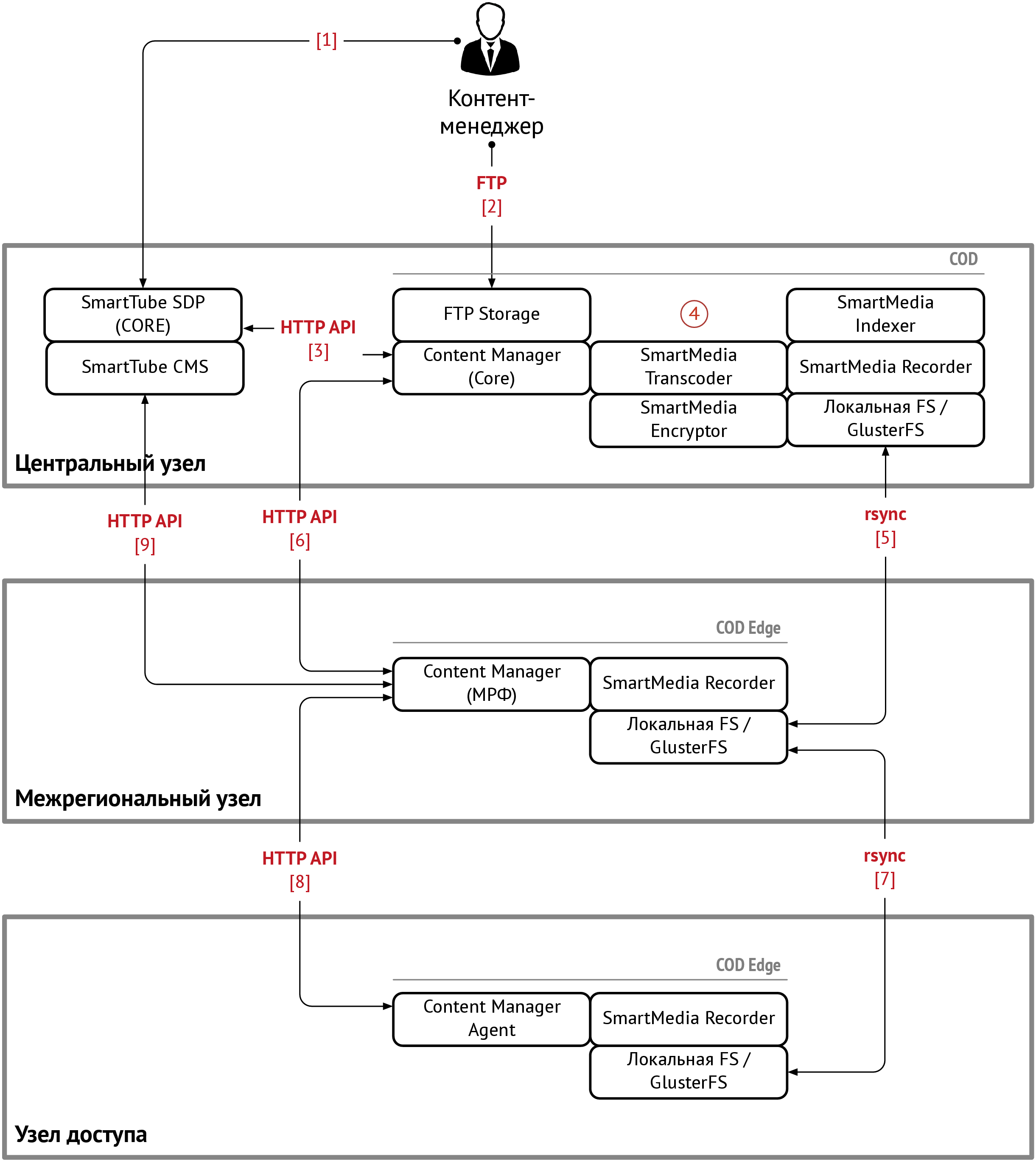
*[Как определять какой абонент откуда по кодам МРФ]*

*[Описание workflow внешнего API]*

### Внутренние потоки данных

*[Описание потоков данных между компонентами платформы при выполнении разных функций. Dataflow-схемы, описание происходящих шагов.]*

#### Загрузка VoD-контента



Загрузка VoD-контента

1. Контент-менеджер при помощи интерфейса AdminUI заводит единицу контента в базе данных SDP:
   1. Загружает описание контента.
   2. Загружает метаданные контента (постеры и описания).
   3. Указывает категории контента.
   4. Указывает регионы доступа контента.
   5. Указывает имя файла мастер-копии контента (ассета).
   6. Указывает необходимость шифрования контента.
   7. Указывает тип публикации контента: только IPTV, IPTV/OTT, только OTT.
2. Контент-менеджер/поставщик контента загружает мастер-копии CoD-файлов в папку файловой системы компонента Content Manager FTP Storage.
3. Компонент Content Manager отслеживает изменениями в файловой системе Content Manager FTP Storage. При появлении нового файла или при изменении существующего файла запрашивает параметры (ассет) соответствующего контента с сервера SmartTube SDP Core. Если ассет с таким именем существует, Content Manager Server передаёт мастер-копии серверам SmartMedia Transcoder по протоколу NFS.
4. Серверы SmartMedia Transcoder с интервалом в 10 минут проверяют наличие новых файлов в NFS-директории и при появлении добавляют их в свою внутреннюю очередь. Затем по имени загруженного файла происходит выбор набора профилей для кодирования:
   * hd\_\* — MPEG4 HD, HD OTT;
   * sd\_\* — MPEG2 SD (при отсутствии суффикса m4o в имени файла, используется только для приставок Amino 110 в МРФ ДВ), MPEG4 SD, SD OTT;
   * sd\_\*karaoke\_film — MPEG4 SD (специальный профиль для файлов караоке), SD OTT. Файлы, предназначенные для вещания на OTT-устройства, шифруются компонентом SmartMedia Encryptor (утилита etsEncrypt). Готовые файлы помещаются в хранилище (локальная NFS-директория/GlutterFS) компонента SmartMedia Recorder. Компонент SmartMedia Indexer создает индексные файлы для транскодированного контента.
5. Content Manager центрального узла отслеживает изменения файловой системы компонентов SmartMedia Transcoder. Транскодированные файлы передаются макрорегиональным компонентам SmartMedia Transcoder посредством протокла rsync.
6. Content Manager центрального узла отправляет макрорегиональным компонентам Content Manager данные об обновлении контента.
7. Этап 5 повторяется для компонентов узлов доступа.
8. Этап 6 повторяется для компонентов узлов доступа.
9. По окончании подготовки контента хотя бы на одном из серверов, Content Manager макрорегионального узла отсылает уведомление о доступности файла в регионе в SmartTube CMS центрального узла. Контент публикуется в соответствии с настроенными в SmartTube SDP правилами тарификации и доступности по регионам. Информация об опубликованном контенте обновляется в SmartTube SDP Cache в каждом макрорегионе при следующем обновлении данных на кеш-серверах. Период обновления настраивается, по умолчанию — 1 раз в час.

#### IPTV-вещание ТВ-эфира

##### Федеральные телеканалы

##### Местные телеканалы

#### Получение PvR-контента прямого эфира

#### Вещание CoD-контента

##### HLS-вещание посредством CDN

##### RTSP-вещание

#### Вещание PVR-контента

#### Обновление данных SDP

#### Сбор данных о пользовательских действиях

#### Использование услуг

#### Отправка нотификаций пользователям

#### Администрирование решения

#### Отправка файлов интерфейса SmartTV

### Состав платформы

*[Описания компонентов детализируются и уточняются]*

#### Система доставки контента

Компоненты системы доставки контента отвечают за получение, обработку и передачу пользовательским продуктом видеоконтента.

В состав системы доставки контента входят:

* группа компонентов SmartMedia;
* компонент TightVideo Transcoder;
* группа компонентов Verimatrix.

##### SmartMedia Transcoder

SmartMedia Transcoder — компонент системы доставки контента, который осуществляет транскодирование видеоконтента.

SmartMedia Recorder работает на основе набора библиотек FFmpeg.

##### SmartMedia Recorder

SmartMedia Recorder — компонент системы доставки контента, который осуществляет получение, хранение и передачу контента.

SmartMedia Recorder получает контент (CoD-файлы, эфир телеканалов), разделяет его на «сегменты» и сохраняет их в собственном хранилище.

SmartMedia Recorder хранит сегменты файлов либо в локальной файловой системе, либо в распределенной файловой системе GlusterFS. GlusterFS позволяет распределенно хранить данные в нескольких физических машинах. На каждом сервере запущен демон glusterfsd который предоставляет доступ к локальному хранилище в виде тома посредством TCP/IP или InfiniBand протоколов, объединяет все доступные серверные тома в один, используя расширяемые трансляторы (функциональные модули GlusterFS). Получившийся том монтируется на центральном сервере при помощи механизма Filesystem in Userspace (FUSE).

##### SmartMedia Indexer

SmartMedia Indexer — компонент системы доставки контента, который осуществляет создание индексных файлов для сегментов контента, хранящихся в SmartMedia Recorder.

Индексные файлы содержат информацию о доступных сегментах и используются компонентами SmartMedia fastHLS и SmartMedia Streamer RTSP для вещания контента клиентским продуктам.

##### SmartMedia Encryptor

SmartMedia Encryptor — компонент системы доставки контента, который осуществляет DRM-шифрование контента, хранящегося в файловом хранилище компонента SmartMedia Recorder.

Для этого SmartMedia Encryptor получает ключи шифрования от сервера Verimatrix CAS.

##### SmartMedia fastHLS

SmartMedia fastHLS — компонент системы доставки контента, который осуществляет вещание контента пользовательским продуктам посредством протокола HLS, а также передачу контента в CDN.

SmartMedia fastHLS осуществляет вещание IPTV/OTT-приставкам (работающим с HLS-протоколом), OTT-приставкам и клиентским приложениям.

##### SmartMedia Streamer RTSP

SmartMedia Streamer RTSP (он же *rtspServer*) — компонент системы доставки контента, который осуществляет вещание контента пользовательским продуктам посредством протокола RTSP

SmartMedia Streamer RTSP осуществляет вещание IPTV/OTT-приставкам, работающим с RTSP-протоколом.

##### SmartMedia HLS Redirector

SmartMedia HLS Redirector — компонент системы доставки контента, который осуществляет перенаправление запроса контента от пользовательского продукта на сервер SmartMedia fastHLS.

##### SmartMedia RTSP Redirector

SmartMedia RTSP Redirector — компонент системы доставки контента, который осуществляет перенаправление запроса контента от пользовательского продукта на сервер Streamer RTSP.

##### TightVideo Transcoder

TightVideo Transcoder — компонент системы доставки контента, который осуществляет транскодирование видеоконтента.

##### Verimatrix Encoder

Verimatrix Encoder — Real Time Encryption Server — компонент системы доставки контента, который осуществляет DRM-шифрование контента.

Verimatrix Encoder используется только для шифрования эфира федеральных телеканалов в центральном узле, передающихся посредством IPTV-вещания.

Verimatrix Encoder осуществляет в реальном времени шифрование эфира в формате MPEG-TS, получая ключи шифрования от сервера Verimatrix CAS.

##### Verimatrix CAS

Verimatrix CAS — Content Authority System — компонент системы доставки контента, который осуществляет выдачу DRM-ключей шифрования и дешифрования контента.

#### Система управления контентом

Компоненты системы управления контентом отвечают за управление контентом и услугами, предоставляют графический интерфейс администраторам платформы, обеспечивают взаимодействие платформы и пользовательских продуктов.

В состав системы управления контентом входят:

* группа компонентов SmartTube;
* группа компонентов SmartSpy;
* группа компонентов Content Manager.

##### SmartTube HTTP Proxy

SmartTube HTTP Proxy — компонент системы управления контентом, который осуществляет проксирование API-запросов внешним системам и ответов от них.

SmartTube HTTP Proxy проксирует запросы от сервера SmartTube SDP (и SmartTube SDP Cache), а также пользовательских продуктов (за исключением запросов к социальным сетям).

##### SmartTube SDP

SmartTube SDP — основной компонент системы управления контентом, который осуществляет:

* управление услугами;
* управление контентом;
* управление пользовательскими данными;
* хранение данных о работе платформы.

SmartTube SDP выступает в качестве точки доступа к данным, хранящимся в SmartTube SDP Database для остальных компонентов платформы и пользовательских устройств.

##### SmartTube SDP Cache

SmartTube SDP — компонент системы управления контентом, который представляет собой кеш-сервер SmartTube SDP.

SmartTube SDP предназначен для распределения нагрузки, создаваемой запросами клиентских продуктов на сервер SmartTube SDP. Запросы от клиентских продуктов на сервер SDP делятся на три категории:

1. Некешируемые запросы — запросы авторизации, статуса учетной записи (активна/заблокирована), запросы биллинга. Эти запросы направляются в SmartTube SDP.
2. Кешируемые простые запросы — запросы справочников значений: тематик ТВ-каналов, жанров фильмов, уровней доступа, прогноза погоды и т.д. Эти запросы направляются в SmartTube SDP Сache.
3. Кешируемые сложные запросы — запросы программы передач (EPG), списков контента: ТВ-каналов, фильмов, сериалов. Эти запросы направляются в SmartTube SDP Сache.

##### SmartTube SDP Database

SmartTube SDP Database — компонент системы управления контентом, база данных компонента SmartTube SDP.

SmartTube SDP Database — это реляционная база данных, управляемая посредством СУБД Oracle RDBMS.

##### SmartTube AdminUI

SmartTube AdminUI — компонент системы управления контентом, предоставляющий веб-интерфейс администраторам платформы.

SmartTube AdminUI — это статический веб-сайт, содержащий разделы и страницы для разных групп администраторов.

Работа SmartTube AdminUI строится на запросах в БД SmartTube SDP Database через компонент SmartTube SDP и БД SmartSpy Database через компонент SmartSpy.

##### SmartTube Hermes

SmartTube Hermes — компонент системы управления контентом, который отправляет нотификационные сообщения пользовательским продуктам.

##### SmartSpy

SmartSpy — компонент системы управления контентом, который осуществляет сбор логов пользовательского поведения с клиентских продуктов.

SmartSpy регистрирует потери IP-пакетов на пользовательских продуктах, фиксирует случаи переполнения/опустошения видеобуфера, вызванные скачками битрейта потока.

На абонентских устройствах SmartSpy позволяет собирать статистику активности абонентов. Активность абонентов регистрируется в виде событий переключения канала. По данным о переключениях ТВ-каналов составляется статистика длительности просмотра различных каналов, популярности телепередач.

##### SmartSpy Database

SmartSpy Database — компонент системы управления контентом, база данных компонента SmartSpy.

SmartSpy — это реляционная база данных, управляемая посредством СУБД Oracle RDBMS.

##### Content Manager FTP Storage

Content Manager FTP Storage — компонент системы управления контентом, который используется для получения мастер-копий CoD-файлов.

##### Content Manager

Content Manager Server — компонент системы управления контентом, который осуществляет наблюдение за изменениями в файловой системе компонента Content Manager FTP Storage.

Content Manager Server отслеживает состояние директории с мастер-копиями на сервере Content Manager FTP Storage и при появлении нового файла или при изменении существующего файла запрашивает параметры (ассет) соответствующего контента с сервера SmartTube SDP Core. Если ассет с таким именем существует, Content Manager Server передаёт мастер-копии серверам SmartMedia Transcoder по протоколу NFS.

##### Content Manager CMS

Content Manager CMS — компонент системы управления контентом, который предоставляет веб-интерфейс для управления загрузкой CoD-файлов контента. Content Manager CMS — это статический веб-сайт, содержащий:

* пакеты фильмов (категорий);
* статусы CoD-файлов контента;
* справочник регионов.

Content Manager CMS установлен в центральном узле платормы и выполняет функции единого хранилища метаданных контента и центра распределения данных по макрорегионам.

##### Content Manager CMS Cache

Content Manager CMS Cache — компонент системы управления контентом, который представляет собой кеш-сервер компонента Content Manager CMS, расположенный в макрорегионе.

Content Manager CMS Cache установлен в каждом макрорегиональном узле, выполняет функции кеширования и обслуживания запросов метаданных CoD от клиентских продуктов.

#### Общесистемные сервисы

Общесистемные сервисы используются совместно с компонентами других систем или же обеспечивают работу решения в целом.

##### nginx

nginx — веб-сервер и кеширующий прокси-сервер, работающий на Unix-подобных операционных системах. Север nginx осуществляет:

* обслуживание статических запросов, запросов индексных файлов;
* автоматическое создание списка файлов, кеш дескрипторов доступных файлов;
* акселерированное проксирование без кэширования;
* распределение нагрузки;
* кеширование при акселерированном проксировании и FastCGI.

В рамках платформы «Интерактивное телевидение Ростелеком» кеш-серверы nginx используются повсеместно на различных серверах.

##### Puppet

Puppet — кроссплатформенное клиент-серверное решение, которое позволяет централизованно управлять конфигурацией операционных систем и программ, установленных на множестве серверов.

Серверы, управляемые с помощью системы Puppet, периодически опрашивают центральных сервер (компонент Puppet), получают и применяют внесённые администратором изменения в конфигурацию. Конфигурация описывается на специальном декларативном предметно-ориентированном языке.

##### Norm

Norm — сервер распространения обновлений прошивок абонентских устройств.

### Аппаратное обеспечение

#### Состав оборудования

Компоненты платформы типового макрузла размещаются на следующем аппаратном обеспечении:

* SmartTUBE SDP Server — 4 сервера.
* SmartTUBE SDP Database — 2 сервера.
* SmartTUBE SDP Cache — 2 сервера.
* SmartTUBE SDP Cache (OTT) — 2 сервера.
* SmartTUBE Hermes — 2 сервер.
* SmartTUBE Portal Cache — 2 сервера.
* SmartMEDIA — 14 серверов.

Спецификация оборудования платформы типового МРФ представлена в таблице ниже.

Спецификация Оборудование типового МРФ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Компонент* | *Процессор* | *RAM* | *HDD* | *RAID* | *Ethernet* |
| SmartTUBE SDP Cache | 2 x E5-2630 | 32 | 4 x 300 |  | 4 x Gigabit |
| SmartTUBE SDP Server | 2 x E-5620 | 12 | 4 x 300 |  | 2 x Gigabit |
| SmartTUBE SDP Database | 2 x E5-620 | 16 | 4 x 300 |  | 2 x Gigabit |
| SmartTUBE SDP Cache OTT | 2 x E5-2630 | 32 | 4 x 300 |  | 4 x Gigabit |
| SmartTUBE Hermes | E3-1230 | 16 | 2 x 300 |  | 2 x Gigabit |
| SmartTUBE Portal Cache | 2 x E5-2630 | 32 | 4 x 300 |  | 4 x Gigabit |
| SmartTUBE SDP Storage |  |  |  | 12 x 146 | 2 x Gigabit   * 4 x FC |
| Content Manager Endpoint | Quad-Core | 16 | 2 x 146 | 24 x 2000 | 12 x Gigabit |
| SmartMEDIA CoD | E3-1230 | 16 | 2 x 300 | 24 x 2000 | 10 x Gigabit |
| SmartMEDIA CoD EDGE | E3-1230 | 16 | 2 x 300 | 12 x 2000 | 10 x Gigabit |
| SmartMEDIA PVR | E3-1230 | 16 | 2 x 300 | 24 x 2000 | 10 x Gigabit |
| SmartMEDIA PVR-Cache | E-5620 | 144 | 2 x 300 | 12 x 2000 | 10 x Gigabit |
| SmartMEDIA PVR-REC (mstor) |  | 64 | 2 x 300 | 24 x 2000 | 4 x 10Gigabit |
| SmartMEDIA PVR-FE (mfe) | E5-2630 | 256 | 2 x 300 |  | 4 x 10Gigabit |
| SmartSPY Server | 2 x Quad-Core | 8 | 2 x 300 |  | 2 x Gigabit |
| Verimatrix CSM | Quad-Core | 8 | 2 x 146 |  | 6 x Gigabit |
| Verimatrix RTES | 2 x Quad-Core | 6 | 2 x 146 |  | 6 x Gigabit |
| Verimatrix VPP | Quad-Core | 4 | 2 x 500 |  | 2 x Gigabit |
| Verimatrix VARC | 2 x Quad-Core | 4 | 2 x 146 |  | 2 x Gigabit |
| Verimatrix DB | 2 x Quad-Core | 8 | 4 x 146 |  | 4 x Gigabit |
| Verimatrix Storage |  |  |  | 12 x 146 |  |

#### Стендовая инсталяция

*[Настройка тестовой сред с выявленным перечнем элементов ПО SmartMedia для определения полноты переданного ПО]*

*[Оценка производительности текущей системы транскодирования и потенциала развития]*

*[Анализ подсистем записи входящего видеопотока и пакетной загрузки видеофайлов для применения в новой платформе]*

## Описание услуг, предоставляемых решением Ростелеком ИТВ

*[Раздел о текущих услугах, предоставляемых решением]*

*[Референс: SmartTUBE\_SDP\_2.7.5.18.\_Users\_guide, раздел 11]*

### Состав услуг

*[Структура услуг: периодические/разовые, бесплатные/монетизируемые]*

*[Список услуг, доступных пользователю, с кратким описанием]*

*[Матрица услуг, доступных в продуктах]*

### Основные сценарии использования

*[Текущие сценарии испоьзования решения]*

#### Регистрация и авторизация устройства

#### Регистрация пользователя

#### Авторизация пользователя

#### Просмотр пользователем IPTV-контента в прямом эфире

#### Просмотр пользователем OTT-контента в прямом эфире

#### Просмотр пользователем OTT-контента CoD

#### Загрузка нового контента

#### Биллинговое обслуживание

##### Создание нового тарифного плана

##### Создание новой услуги/пакета услуг

##### Создание нового абонента

##### Смена тарифного плана

##### Активация подключений

##### Подключение дополнительной STB

##### Отключение дополнительной STB

##### Отключение услуги IPTV

##### Восстановление активационных данных

##### Блокировка услуги IPTV

##### Снятие блокировки услуги IPTV

##### Заказ/отказ от разовой услуги

##### Подключение периодической услуги

##### Отключение периодической услуги

##### Получение информации о состоянии счета (баланс)

## Описание клиентских продуктов решения Ростелеком ИТВ

Абонентские устройства:

* *IPTV-приставки первого поколения (IPTV/RTSP).*
* *IPTV-приставки второго поколения (IPTV/HLS).*
* *OTT-приставки (HLS).*

Клиентские приложения:

* *Android-приложение Zabava (HLS.)*
* *iOS-приложение Zabava (HLS).*
* *Веб-плеер Zabava (HLS).*

### IPTV/RTSP-приставки

#### Предоставляемые услуги

#### Описание работы

#### Взаимодействие с серверной платформой

#### Структура пользовательского интерфейса

*(Анализ интерфейсных решений мобильных приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

*(Анализ интерфейсных решений SmartTV приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

### IPTV/OTT-приставки

#### Предоставляемые услуги

#### Описание работы

#### Взаимодействие с серверной платформой

#### Структура пользовательского интерфейса

*(Анализ интерфейсных решений мобильных приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

*(Анализ интерфейсных решений SmartTV приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

### OTT-приставки

#### Функции

#### Предоставляемые услуги

#### Описание работы

#### Взаимодействие с серверной платформой

#### Структура пользовательского интерфейса

*(Анализ интерфейсных решений мобильных приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

*(Анализ интерфейсных решений SmartTV приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

#### Архитектура

*Раздел под вопросом*

### Android-приложение Забава

#### Предоставляемые услуги

#### Описание работы

#### Взаимодействие с серверной платформой

#### Структура пользовательского интерфейса

*(Анализ интерфейсных решений мобильных приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

*(Анализ интерфейсных решений SmartTV приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

### iOS-приложение Забава

#### Предоставляемые услуги

#### Описание работы

#### Взаимодействие с серверной платформой

#### Структура пользовательского интерфейса

*(Анализ интерфейсных решений мобильных приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

*(Анализ интерфейсных решений SmartTV приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

### Веб-плеер Забава

#### Предоставляемые услуги

#### Описание работы

#### Взаимодействие с серверной платформой

#### Структура пользовательского интерфейса

*(Анализ интерфейсных решений мобильных приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

*(Анализ интерфейсных решений SmartTV приложений для формирования предложений по предоставлению услуг на платформе SmartMEDIA 4.0)*

## Недостатки текущего решения

### Выявленные недостатки платформы «Интерактивное телевидение Ростелеком»

*(Оценка показателей производительности выявленных элементов платформы и их соответствия заявленных а презентационных материалах и технической документации Smartlabs (fastHLS, HLSRedirector, smartIndex, smartRecord, rtspServer, smartMediaUtils, fastHLS, HLSRedirector, smartIndex, smartRecord, rtspServer, smartMediaUtils и других элементов платформы SmartMedia 3.x))*

*(Оценка производительности аппаратной платформы методом нагрузочного тестирования — в рамках стендовой инсталяции)*

*(Тестирование централизованной архитектуры платформы и определение возможностей для масштабирования системы балансировки запросов на просмотр видеоконтента, поступающих от STB)*

*(Тестирование тестовой среды с использование различных типов нагрузки и угроз)*

*(Определение проблемных архитектурных решений и ограничений для масштабирования программного комплекса компонентов при использовании на платформе SmartMedia 3.x (fastHLS, HLSRedirector, smartIndex, smartRecord, rtspServer, smartMediaUtils, fastHLS, HLSRedirector, smartIndex, smartRecord, rtspServer, smartMediaUtils и других элементов платформы SmartMedia))*

*(Оценка эффективности системы управления услугами доступа к цифровому контенту SDP и формирование требований для SDP 3.0)*

*(Анализ модели данных и структуры БД, оценка эффективности принятых решений)*

*(Тестирование и оценка эффективности принятых технических решений в клиентском интерфейсе программного взаимодействия (API))*

*(Тестирование тестовой среды с использование различных типов нагрузки и угроз)*

#### Общесистемные

1. Отсутствует единый инструмент управления работой платформы.
2. Текущие программные интерфейсы и их методы не позволяют реализовать развитие возможностей мобильных приложений и веб-сайта.
3. Низкое качество оказания услуг.
4. Перенос контента из одного устройства пользователя в другое происходит с существенной задержкой.

#### Управление работой и контентом

1. Управление контентом:
   * Недостаточная функциональность для работы с метаданными контента: справочники не позволяют структурировать данные описаний, инструменты администратора не позволяют поддерживать целостность и корректность данных, нельзя задать необходимые параметры.
   * Низкий уровень автоматизации работы администратора: для выполнения типовых операций необходимо выполнение большого количествка действий оператора, что повышает вероятность возникновения ошибкок.
   * Распространение контента по узлам происходит в течении значительного времени из-за отсутствия централизации. Большое количество ручных операций по согласованию данных между узлами и высокая вероятность несогласованности данных.
   * Не реализованны автоматизированные процедуры загрузки контента и метаданных из внешних источников
2. Управление услугами:
   * Децентрализованнасть управления приводит к необходимости дублировать действия администраторов.
   * Отсутствуют шаблоны создания услуг.
   * Не реализовано управление стоимостью услуг.
3. Управление абонентами:
   * Управление абонентами реализовано в ограниченном объеме.
   * Нет возможности отправлять персонализированные и групповые уведомления абонентам.
   * Нет возможности просмотреть историю изменений тарифных планов, привязок клиентских продуктов, активаций/деактиваций устройств и приложений по отдельному пользователю или группе пользователей.
4. Управление клиентскими продуктами:
   * Отсутствует возможность автоматического обновления прошивок абонентских устройств.
   * Отсутствует возможность автоматического обновления клиентских приложений.
   * В качестве уникальных идентификаторов абонентских устройств используются MAC-адреса.
   * Нет возможности просмотреть историю изменений параметров клиентских продуктов и групп.
5. Статистика и анализ:
   * Отсутствие единого специализированного хранилища статистических данных приводит к противоречиям и несогласованности данных отчетности.
   * Не проводится сбор и анализ данных мониторинга, работы пользовательской поддержки, системы рекомендаций, действий пользователя, качества оказания услуг и т.д.
   * Имеющийся набор отчетов и исходных данных не позволяет проводить полноценный продуктовый анализ, анализ востребованности/реализации продуктов, контента и услуг.
   * Создание новых отчетов требует привлечение разработчиков.
   * Нет визуального представления данных.
   * Нет инструментов для проведения A/B-тестирования.
6. Мониторинг:
   * Ведётся мониторинг только работоспособности сегментов оборудования платформы и отдельных подсистем. Ограниченный мониторинг качества предоставления услуг реализован только в некоторых IPTV/OTT-приставках.
   * Мониторинг каналов связи, работоспособности компонентов платформы и наличие связи с внешними системами не проводится.
   * Нет единого интерфейса мониторинга для администраторов.
   * Нет системы уведомления об инцедентах.
7. Пользовательская поддержка:
   * Нет инструментов получения телеметрических данных с клиентских продуктов и автоматической передачи их сотрудникам техподдержки.
   * Нет единого механизма сбора диагностических данных инцидентов; для сбора данных требуется привлечение дополнительных специалистов.
   * Не реализована связь с системой мониторинга
   * Ограниченная функциональность подачи заявок от пользователей.
8. Администраторский интерфейс управления платформой:
   * Интерфейсы управления платформы не объединяют все инструменты управления, что приводит к увеличению сложности поддержки и обслуживания интерфейсных решений.
   * Слабые возможности персонализации интерфейса.
   * Нет системы уведомлений.
   * Неудовлетворительная реализация отдельных интерфейсных компонентов приводит к необоснованному увеличению времени ввода данных и некорректности их заполнения.
9. Сервис рекомендаций:
   * Сервис рекомендаций по факту отсутствует.
   * Нет инструментов для анализа пользовательских данных и данных контента, что приводит к невозможности персонализировать выдачу под каждого пользователя.
   * Нет инструментов обратной связи от пользователя.
   * Нет возможности настроить выдачу рекомендаций для групп пользователей на основе их предпочтений.

#### Доставка контента

*[Раздел в написании]*

#### База данных SDP

1. Структура хранения данных не нормализована: в столбцах таблиц зачастую хранится не одно значение (например, список актеров фильма/передачи — это строка с перечислением имен через разделитель, причём разделитель может быть любой). Это значительно затрудняет анализ данных, требует усилий при генерации отчетности и приводит к ошибкам в данных БД, не позволяет использовать современные рекомендательные алгоритмы.
2. Связи между сущностями реализованы посредством generic-таблицы object\_link с помощью отношения «родитель»–«потомок» (parent–child), без использования внешних ключей. Некоторые таблицы содержат несколько разных сущностей (например таблица content). Это приводит к различным проблемам:
   * производительность БД снижается при обработке нескольких одновременных запросов, которые запрашивают связанные данные и неизбежно обращаются к таблицам object\_link, dictionary, content (что приводит к состояниям ожидания — «waits» и блокировкам — «latches»);
   * избыточная сложность SQL-запросов к БД как для разработчика/администратора, так и для функций оптимизации БД;
   * отсутствует контроль целостности данных;
   * недостаточно информации для работы функций оптимизации БД;
   * сложность администрирования большой непартиционированной таблицы;
   * сложность анализа данных и расширения функциональности решения на их основе;
   * отсутствие внешних ключей переносит контроль целостности данных на приложение, что увеличивает риск возникновения ошибок.
3. Таблицы в БД не партиционированы и содержат избыточные индексы:
   * отсутствие партиций снижает производительность работы с таблицами и усложняет процесс удаления устаревших данных из оперативной БД;
   * избыточные индексы снижают производительность при вставке данных и потребляют ресурс дисковой подсистемы.
4. В БД не реализован единый программный интерфейс (API) с функциями и процедурами доступа к данным и внесения новых данных, что:
   * значительно усложняет добавление новых подсистем и модулей — эти модули и подсистемы должны обеспечивать правильность внесения данных самостоятельно;
   * увеличивает общую нагрузку на БД, так как одни и те же данные будут выбираться разными SQL-командами

### Выявленные недостатки услуг решения

*[Список пополняется]*

* Набор услуг в продуктах различается.
* Пользователю необходим авторизоваться чтобы получить полный перечень доступных услуг.

### Выявленные недостатки клиентских продуктов

*[Список пополняется]*

* Различается функциональность клиентских продуктов.
* Различаются интерфейсные решения клиентских продуктов.
* Различается логика оказания услуг на клиентских продуктах.
* Отсутствует полноценный поиска по решениям и услугам.
* Деавторизация и повторная авторизация приводят к полной перезагрузке контента.

# Концепция решения Ростелеком ИТВ 2.0

## Предпосылки

На данный момент ПАО «Ростелеком» предоставляет пользователям набор услуг интерактивного телевидения в рамках решения, основанного на технической платформе Smartlabs. Это решение предоставляет пользователям базовые возможности интерактивного телевидения, доступ к телевизионным трансляциям в прямом эфире и видеоконтенту по запросу (VoD) с различными способами оплаты.

Текущее решение интерактивного телевидения ПАО «Ростелеком» обладает рядом существенных недостатков, описанных в соответствующих разделах этого документа:

* недостатки серверной платформы интерактивного телевидения (подробнее в разделе [*«Выявленные недостатки платформы «Интерактивное телевидение Ростелеком»»*](#выявленные-недостатки-платформы-итв));
* недостатки пользовательских продуктов: устройств и приложений (подробнее — [*«Выявленные недостатки клиентских продуктов»*](#выявленные-недостатки-клиентских-продуктов));
* недостатки набора услуг, предоставляемых пользователям (подробнее — [*«Выявленные недостатки услуг решения»*](#выявленные-недостатки-услуг-решения));
* невозможность использования решения в B2B- и B2G-сегментах из-за отсутствия исключительных прав на платформу.

Все эти недостатки в совокупности с качеством реализации текущих клиентских продуктов и платформы интерактивного телевидения не позволяют осуществить:

* предоставление услуг пользователям с необходимым уровнем качества;
* оперативное расширение спектра функциональных возможностей решения;
* оперативное добавление новых услуг;
* быструю интеграцию новых клиентских продуктов.

*[необходимо избежать использования существующих решений SL из-за лицензионных ограничений]*

## Постановка задачи

В рамках развития решения интерактивного телевидения необходимо модернизировать существующую платформу «Интерактивное телевидение Ростелеком» и разработать дополнительные компоненты платформы для повышения управляемости сервиса. Это позволит:

* Обеспечить высокое качество доставки контента пользователям.
* Реализовать механизмы статистики, аналитики и мониторинга использования платформы.
* Обеспечить возможность предоставления единого набора услуг в клиентских продуктах.
* Разработать и внедрить единое интерфейсное решения для всех клиентских продуктов:
  + ТВ-приставки (Set-top-box);
  + Мобильные приложения;
  + SmartTV-приложения;
  + Веб-сайт ИТВ.
* Расширить аудиторию сервиса за счёт разработки приложений для новых мобильных и SmartTV-платформ, игровых консолей и ТВ-приставок независимых производителей.
* Разработать систему релевантных рекомендаций видеоконтента.
* Реализовать функцию дистанционного управления просмотром в рамках услуги «Мультискрин».
* Осуществлять поставку данных и предоставить площадку для интеграции различных услуг и сервисов в рамках B2B-взаимодействия.
* Осуществлять предоставление медиаконтента в рамках B2G- и B2B-взаимодействия.

### Услуги, предоставляемые решением

*[Список не окончательный, уточняется]*

#### Перечень услуг

Решение ИТВ 2.0 предоставляет пользователям следующие виды услуг:

* контентные услуги — услуги, предоставляющие доступ к контенту;
* системные услуги — дополнительные услуги, связанные и несвязанные с просмотром контента.

Контентные услуги:

* Просмотр прямого эфира ТВ-каналов.
* Просмотр архива передач ТВ-каналов.
* Доступ к программе передач ТВ-каналов
* Просмотр VoD-контента (фильмы, сериалы и мультфильмы).
* Просмотр видеотрансляций (лекции, семинары, конференции и т.д.).
* Доступ к информационным сервисам (прогноз погоды, курсы валют, пробки и новости).
* Доступ к сервису облачных игр.

Системные услуги:

* Мультискрин — услуга переноса покупок и личной информации между устройствами одного пользователя.
* Управление просмотром эфира — пауза, перемотка эфира ТВ-каналов, просмотр закончившихся телепередач, запись телепередач (TSTV/nPVR/PauseLive).
* Сервис рекомендаций фильмов и ТВ-передач.
* Просмотр контента с USB-носителя (для ТВ-приставок).
* Просмотр загруженного контента без подключения к интернету (для мобильных устройств).
* Функция родительского контроля.
* Функция второго экрана (Second Screen).
* Функция поиск по контента.
* Использование мобильного устройства как пульта дистанционного управления ТВ-приставкой и SmartTV.
* Просмотр на дополнительных устройствах связанных с основных через DLNA/AirPlay/WiDi.
* Организация ставок на спортивные состязания.
* Оплата внешних услуг (ЖКХ, мобильная связь, штрафы ГИБДД) из личного счёта абонента, привязанной банковской картой, внешней платёжной системой, счёта мобильного телефона.
* Синхронизация покупок с внешними сервисами.
* Географическая сортировка контента.
* Привязка банковской карты к учетной записи абонента.
* Идентификация, регистрация и активация устройства в системе [?].
* Услуги телефонии.
* Услуга «Подарки».
* Услуга «Кинорулетка».
* Файловые квоты. […]

Пользователи могут воспользоваться услугами решения в рамках нескольких моделей доступа:

* бесплатный доступ;
* разовая подписка;
* периодическая подписка;
* предоплата (подписка на ещё недоступный контент)
* аренда VoD-контента;
* предустановленный на устройство доступ.
* промокоды и промоакции;
* подарки;
* синхронизация покупок с внешними сервисами.

[*матрица услуг и их применимость на разных платформах*]

#### Ценообразование и структура услуг

[*TBD*]

### Клиентские продукты решения

Клиентские продукты решения ИТВ 2.0 разделяются на абонентские устройства и клиентские приложения.

#### Абонентские устройства

Абонентское устройство — это аппаратная платформа с установленным на нём специализированным программным обеспечением, предоставляющая доступ к услугам решения ИТВ 2.0. Абонентским устройство может быть ТВ-приставка, PLC-адаптер, HDMI-медиаплееры.

Аппаратная платформа передаётся пользователю в готовом для использования виде. Самостоятельная модификация пользователем ПО или аппаратных компонентов абонентского устройства не допускается.

Типы абонентских устройств.

|  |  |  |  |
| --- | --- | --- | --- |
| Производитель | Модель | Тип | Комментарий |
| AmiNET | AmiNET 110 | IPTV | Устаревшие приставки в МРФ ДВ |
| Yuxing | YX6916A | IPTV |  |
| SmartLabs | SML-282 | IPTV |  |
| SmartLabs | SML-292 | IPTV |  |
| SmartLabs | SML-482 | OTT | c WiFi-модулем |
| Motorola | VIP 1003 | IPTV |  |
| Motorola | VIP 1963 | IPTV |  |
| Промсвязь | IPTV HD | IPTV |  |
| Промсвязь | IPTV HD mini | IPTV |  |
| Инфомир | MAG200 | IPTV |  |
| Инфомир | MAG250 | IPTV |  |
| ZyXEL | STB1001S | IPTV |  |
| ZyXEL | STB1001S2 | IPTV |  |
| ZyXEL | STB1001H | IPTV |  |

#### Клиентские приложения

Решение предоставляет набор клиентских приложений, устанавливаемых на пользовательские устройства.

Клиентские приложения могут быть установлены на устройства (смартфоны, планшеты, телевизоры), работающие под управлением операционных систем:

* iOS (версий 7 и выше);
* Android (версий 4 и выше);
* SmartTV (выпуска 2013 года и позже).

Разработанные API-решения и база данных должны позволять оперативное расширение поддерживаемых платформ и типов устройств.

### Платформа интерактивного телевидения 2.0

Подробное описание проектируемой архитектуры платформы «Интерактивное телевидение Ростелеком» 2.0 приведено в главе [*«Техническое описание проектируемой платформы Интерактивного телевидения 2.0»*](#техническое-описание-проектируемой-платформы-интерактивного-телевидения-2.0) этого документа.

#### Система управления платформой

Система управления контентом существующей архитектуры платформы «Интерактивное телевидение Ростелеком» преобразуется в Систему управления платформой нового решения за счёт модернизации и доработки соответствующих компонентов.

#### Система доставки контента

Система доставки контента модернизируется с целью оптимального использования существующего аппаратных обеспечения и устранения недостатков текущего решения.

## Основные сценарии использования решения

### Пользовательские сценарии

*[Все сценарии для которых не указан тип клиентского устройства по умолчанию должны быть описаны для всех типов]*

*[Сценарии покупки необходимо рассмотреть с точки зрения возможности авансовых/кредитных платежей, приобретения через подарки других пользователей, платежей с ЛС, карточек, внешних платежных систем]*

*[Для сценариев просмотра рассмотреть ситуации покупки на одном устройстве и потребления на другом]*

*[На данный момент нет четкого понимания логики доступности и стоимости услуг при нахождении в группе разнородных устройств (IPTV и OTT) или устройств, активированных в различных МРФ. Предложение: в сценариях описать эти ветки и указать их, как проблемные. UPD: решится в ближайшее время]*

#### Подключение IPTV-приставки

*Название сценария:* Подключение IPTV-приставки.

*Назначение сценария:* Сценарий описывает процесс подключения и первоначальной настройки IPTV-приставки.

*Участники сценария:*

* Абонент;
* Сотрудник Ростелекома Ростелекома.

*Предусловие:* Абонент является пользователем услуги ШПД; подключает услугу ШПД и IPTV. Абонент инициировал процесс подключения.

*Основные шаги:* [Описать процесс установки и подключения IPTV-приставки монтажником или абонентом самостоятельно]

#### Подключение OTT-приставки

*Название сценария:* Подключение OTT-приставки.

*Назначение сценария:* Сценарий описывает процесс подключения и первоначальной настройки OTT-приставки.

*Участники сценария:*

* Абонент.

*Предусловие:* Абонент приобрел/арендовал OTT-приставку. Абонент инициировал процесс подключения.

*Основные шаги:* [Описать процесс установки и подключения OTT-приставки]

#### Установка приложения

*Название сценария:* Установка клиентского приложения.

*Назначение сценария:* Сценарий описывает процесс установки клиентского приложения на пользовательское устройство.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь включил устройство и подключил его к интернету.

*Основные шаги:* [Описать процесс установки приложения на мобильные устройства и SmartTV]

#### Активация устройства

*Название сценария:* Активация устройства.

*Назначение сценария:* Сценарий описывает процесс активации устройства пользователем.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь включил устройство и подключил его к интернету.

*Основные шаги:* [Описать процесс активации устройства/приложения]

#### Деактивация устройства

*Название сценария:* Деактивация устройства.

*Назначение сценария:* Сценарий описывает процесс деактивации устройства пользователем.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь активировал устройство.

*Основные шаги:* [Описать процесс деактивации устройства/приложения]

#### Авторизация пользователя

*Название сценария:* Авторизация пользователя.

*Назначение сценария:* Сценарий описывает процесс авторизации пользователя.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь включил устройство и подключил его к интернету.

*Основные шаги:* [Описать процесс авторизации пользователя]

#### Объединение клиентских продуктов в группы

*Название сценария:* Объединение клиентских продуктов в группу.

*Назначение сценария:* Сценарий описывает процесс объединения клиентских продуктов в группу при помощи услуги «мультискрин».

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах под разными учетными записями. Устройства могут быть активированы в различных МРФ, типы клиентских продуктов – любые.

*Основные шаги:* [Описать процесс объединения клиентских продуктов в группы]

#### Управление профилями

*Название сценария:* Управление профилями.

*Назначение сценария:* Сценарий описывает процесс создания, редактирования, удаления профилей пользователя.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах под разными учетными записями. Устройства могут быть активированы в различных МРФ, типы клиентских продуктов – любые.

*Основные шаги:* [Описать процесс управления профилями: создание, редактирование, удаление, переноса между устройствами]

#### Удаление клиентских продуктов из группы

*Название сценария:* Удаление клиентских продуктов из группы.

*Назначение сценария:* Сценарий описывает процесс удаления клиентских продуктов из группы

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Устройства могут быть активированы в различных МРФ, типы клиентских продуктов – любые.

*Основные шаги:* [Описать процесс удаления клиентских продуктов из группы]

#### Смена тарифного плана

*Название сценария:* Смена тарифного плана.

*Назначение сценария:* Сценарий описывает процесс смены тарифного плана.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет одно или несколько активированных клиентских продуктов, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс смены тарифного плана.]

#### Приобретение подписки на ТВ-пакет

*Название сценария:* Приобретение подписки на ТВ-пакет.

*Назначение сценария:* Сценарий описывает процесс приобретения подписки на ТВ-пакет.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс подписки]

#### Просмотр ТВ-канала

*Название сценария:* Просмотр ТВ-канала.

*Назначение сценария:* Сценарий описывает процесс просмотра ТВ-канала.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Пользователь приобрел подписку на пакет ТВ-каналов.

*Основные шаги:* [Описать процесс просмотра ТВ-канала. Описать доступность ТВ-каналов между устройствами в группе, использование EPG, услуг управления просмотром TS/PL/nPVR]

#### Приобретение/аренда VoD, VoD-пакета

*Название сценария:* Приобретение/аренда VoD, VoD-пакета.

*Назначение сценария:* Сценарий описывает процесс приобретения VoD, VoD-пакета.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс покупки VoD, VoD-пакета, сериала. При схожести с предыдущим сценарием – объединить в один]

#### Просмотр VoD контента

*Название сценария:* Просмотр контента.

*Назначение сценария:* Сценарий описывает процесс просмотра приобретенного/арендованного VoD.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Пользователь приобрел/арендовал контентVoD/VoD-пакет/сериал.

*Основные шаги:* [Описать процесс просмотра контента]

#### Загрузка контента

*Название сценария:* Загрузка контента

*Назначение сценария:* Сценарий описывает процесс загрузки VoD

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Пользователь приобрел/арендовал VoD-контент, VoD-пакет.

*Основные шаги:* [Описать процесс загрузки VoD.]

#### Просмотр контента без подключения к интернету

*Название сценария:* Просмотр контента без подключения к интернету

*Назначение сценария:* Сценарий описывает процесс просмотр контента без подключения к интернету.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Пользователь приобрел/арендовал контент или VoD-пакет и загрузил контент.

*Основные шаги:* [Описать процесс просмотра VoD без сети.]

#### Поиск

*Название сценария:* Поиск контента.

*Назначение сценария:* Сценарий описывает процесс поиска контента.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Устройства могут быть активированы в различных МРФ, типы клиентских продуктов – любые.

*Основные шаги:* [Описать процесс поиска. Поиск проходит по всему доступному контенту, EPG и услугам.]

#### Кинорулетка

*Название сценария:* Использование услуги кинорулетка.

*Назначение сценария:* Сценарий описывает процесс использования услуги кинорулетка.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Устройства могут быть активированы в различных МРФ, типы клиентских продуктов – любые. *Основные шаги:* [Описать процесс использования услуги кинорулетка - случайный выбор контента из всего доступного контента.]

#### Родительский контроль

*Название сценария:* Использование функции родительского контроля

*Назначение сценария:* Сценарий описывает процесс установки ограничений на просмотр контента и использования сервисов решения при помощи функции Родительский контроль на основании профилей

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс работы родительского контроля.]

#### SecondScreen

*Название сценария:* Использование функции SecondScreen.

*Назначение сценария:* Сценарий описывает процесс использования функции SecondScreen.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Пользователь подключил услугу SecondScreen.

*Основные шаги:* [Описать процесс работы услуги SecondScreen.]

#### Детский пульт

*Название сценария:* Использование функции Детский пульт.

*Назначение сценария:* Сценарий описывает процесс использования функции Детский пульт.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин». Пользователь подключил услугу Детский пульт.

*Основные шаги:* [Описать процесс работы услуги Детский пульт. Есть документация со сценариями, использовать]

#### Подключение дополнительного IPTV-приставки

*Название сценария:* Подключение дополнительного IPTV-приставки

*Назначение сценария:* Сценарий описывает процесс подключения и настройки дополнительного IPTV-приставки.

*Участники сценария:*

* Абонент
* Сотрудник Ростелекома

*Предусловие:* Абонент является пользователем услуги IPTV-приставки. Абонент инициировал процесс подключения дополнительного IPTV-приставки.

*Основные шаги:* [Описать процесс установки и подключения IPTV-приставки монтажником или абонентом самостоятельно. Может использоваться как дополнительная абонентская линия, так и существующая]

#### Дополнительная файловая квота

*Название сценария:* Предоставление дополнительной файловой квоты.

*Назначение сценария:* Сценарий описывает процесс подключения дополнительной файловой квоты.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс подключения и использования дополнительной файловой квоты]

#### Просмотр файлов c USB-носителя

*Название сценария:* Просмотр файлов с USB-носителя.

*Назначение сценария:* Сценарий описывает процесс переноса в облако персональных медиа файлов (видео, фото) и управление ими между устройствами.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:*

1. Запись файлов в облаке.
2. Синхронизация файлов между устройствами пользователя.
3. Демонстрация файлов. [Описать процесс подключения и использования дополнительной файловой квоты.]

#### Добровольная блокировка

*Название сценария:* Предоставление услуги добровольной блокировки.

*Назначение сценария:* Сценарий описывает процесс предоставления услуги добровольной блокировки.

*Участники сценария:*

* Пользователь.

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс подключения и использования услуги добровольной блокировки.]

### Сторонние сервисы

[ЖКХ, мобильная связь, штрафы, ставки на спорт]

[доработать]

### Обращение в пользовательскую поддержку

*Название сценария:* Обращение в пользовательскую поддержку

*Назначение сценария:* Сценарий описывает процесс обращения в пользовательскую поддержку

*Участники сценария:*

* Пользователь.
* Сотрудник технической поддержки

*Предусловие:* Пользователь имеет несколько активированных клиентских продуктов. Пользователь авторизован в клиентских продуктах, объединенных в группу при помощи услуги «мультискрин».

*Основные шаги:* [Описать процесс обращения пользователя в поддержку через клиентское приложение/устройство, по телефону, e-mail и пр.]

### Административные сценарии

*[Многие сценарии можно взять из руководства пользователя существующей системы от СЛ и откорректировать в соответствии с новыми требованиями]*

#### Регистрация администратора

*Название сценария:* Регистрация администратора платформы

*Назначение сценария:* Сценарий описывает процесс регистрации нового администратора платформы

*Участники сценария:*

* Новый администратор решения
* Администратор решения

*Предусловие:* Новый администратор решения со своего рабочего места имеет доступ к web-интерфейсу панели управления платформой. Права доступа нового администратора платформы в соответствии с внутренними документами Ростелекома включают в себя доступ к функциональности панели управления платформой. Администратору регистрирующему нового администратора доступна соответствующая функциональность.

*Основные шаги:* [Описать процесс регистрации нового администратора в системе.]

#### Авторизация администратора

*Название сценария:* Авторизация администратора платформы

*Назначение сценария:* Сценарий описывает процесс авторизации администратора платформы

*Участники сценария:*

* Администратор решения

*Предусловие:*

* Администратор решения со своего рабочего места имеет доступ к web-интерфейсу панели управления платформой и зарегистрирован в системе.

*Основные шаги:* [Описать процесс авторизации администратора в системе.]

#### Загрузка и настройка VoD

*Название сценария:* Загрузка и настройка VoD

*Назначение сценария:* Сценарий описывает процесс загрузки VoD из внешних источников, настройки параметров и распространения его по узлам платформы

*Участники сценария:*

* Администратор.

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Перед администратором стоит задача загрузки VoD в платформу. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Описать процесс загрузки VoD в ручном и полуавтоматическом режиме с подтверждением от администратора, задание параметров контента, его метаданных и управление его распространением по узлам платформы. Привязка контента к категориям.]

#### Создание и настройка ТВ-канала

*Название сценария:* Создание и настройка ТВ-канала

*Назначение сценария:* Сценарий описывает процесс создания, изменения и удаления ТВ-канала, настройки EPG, задание параметров ТВ-канала.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Описать процесс создания ТВ-канала, задания параметров вещания, ограничений вещания, при условии, что канал является локализацией федерального канала привязки к нему, настройка EPG. Описать процессы изменения параметров канала и его удаления]

#### Создание сериалов

*Название сценария:* Объединение загруженного VoD в сериалы и сезоны сериалов

*Назначение сценария:* Сценарий описывает процессы создания, изменения и удаления сериалов и сезонов сериалов на базе загруженного VoD.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. VoD-контент (серии) загружен и в параметрах загрузки есть указания на то, что он является серией сериала. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Описать процесс создания сезона сериала и сериала, процесс добавления новых серий, сортировки серий в сезоне, сериале, добавления метаданных сериала, изменения и удаления вхождения VoD (серии) в сериал, сезон]

#### Объединение VoD в пакеты

*Название сценария:* Объединение VoD в пакеты

*Назначение сценария:* Сценарий описывает процесс создания, изменения и удаления пакета VoD.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Единицы VoD, для объединения в пакет загружены и настроены, сериалы созданы. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Описать процесс создания, изменения и удаления пакета VoD на базе загруженных единиц контента]

#### Объединение ТВ-каналов в пакеты

*Название сценария:* Объединение ТВ-каналов в пакеты

*Назначение сценария:* Сценарий описывает процесс создания, изменения и удаления пакета ТВ-каналов.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. ТВ-каналы для объединения в пакет созданы и настроены. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Описать процесс создания, изменения и удаления пакетов ТВ-каналов на базе существующих и настроенных ТВ-каналов]

#### Ведение соглашений с правообладателями

*Название сценария:* Ведение соглашений с правообладателями

*Назначение сценария:* Сценарий описывает процесс создания, изменения и удаления соглашений с правообладателями, привязка контента к соглашениям, уведомление администраторов о приближении момента истечения соглашения и окончании прав на вещание контента.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. VoD-контент для существующего соглашения загружен и настроен. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, изменения и удаления соглашений с правообладателями, привязка контента к соглашениям, уведомление администраторов о приближении момента истечения соглашения и окончании прав на вещание контента.]

#### Ведение справочников

*Название сценария:* Ведение контентных справочников

*Назначение сценария:* Сценарий описывает процесс создания, изменения и удаления записей в контентные справочники – жанров, стран производстваактеров, режисеров и пр.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, изменения и удаления записей в контентных справочниках]

### Управление услугами

#### Создание и настройка услуги

*Название сценария:* Создание и настройка услуги

*Назначение сценария:* Сценарий описывает процесс создания, настройки, изменения и удаления услуг.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, настройки, изменения и удаления услуги. Привязка к правилам тарификации. пакетам и категориям контента.]

#### Создание и настройка правил тарификации

*Название сценария:* Создание и настройка правил тарификации

*Назначение сценария:* Сценарий описывает процесс создания, настройки, изменения и удаления правил тарификации.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, настройки, изменения и удаления правил тарификации.]

#### Создание и настройка тарифного плана

*Название сценария:* Создание и настройка тарифного плана

*Назначение сценария:* Сценарий описывает процесс создания, настройки, изменения и удаления тарифного плана.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, настройки, изменения и удаления тарифного плана. Процесс привязки услуг к тарифному плану, соответствия МРФ, проверки совместимости услуг.]

#### Управление скидками и промоакциями

*Название сценария:* Управление скидками и промоакциями

*Назначение сценария:* Сценарий описывает процесс создания, настройки, изменения и удаления промоакций.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, настройки, изменения и удаления промоакций. Процесс привязки контента к акции и проверки совместимости действий.]

### Управление клиентскими продуктами

#### Создание групп пользовательских продуктов

*Название сценария:* Создание группы пользовательских продуктов

*Назначение сценария:* Сценарий описывает процесс создания группы пользовательских продуктов.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, изменения и удаления группы пользовательских продуктов на основе фильтрации по МРФ, типу клиентского продукта, конкретной реализации аппаратного обеспечения, версии прошивки или приложения, группе абонентов и пр.]

#### Обновление прошивок устройств/версий приложений

*Название сценария:* Обновление прошивок, версий приложений

*Назначение сценария:* Сценарий описывает процесс обновления прошивок устройств и обновление версий приложений.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс обновления прошивок и версий приложения группы пользовательских продуктов или отдельных пользовательских продуктов.] Управление пользователями

*Название сценария:* Управление пользоавателями

*Назначение сценария:* Сценарий описывает процесс сбора взаимодейтсвия с интерфейсом для получения данных о состоянии подключенных услуг, тарифов и группе устройств.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:*

[Сценарий описывает процесс запроса информации о потреблении медиа улсуг пользователям и диагностической информации для целей службы поддержки и мониторинга]

#### Создание групп пользователей

*Название сценария:* Создание группы пользователей

*Назначение сценария:* Сценарий описывает процесс создания группы пользовательских продуктов.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:*

[Сценарий описывает процесс создания, изменения и удаления группы пользователей на основе фильтрации по тарифному плану, МРФ, типу клиентского продукта, конкретной реализации аппаратного обеспечения, версии прошивки или приложения, группе абонентов и пр.]

### Сбор статистики её аналитика

#### Создание периодических отчетов

*Название сценария:* Создание периодического отчета

*Назначение сценария:* Сценарий описывает процесс создания периодического отчета.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создания, изменения и удаления периодического отчета. Отчет создается на базе существующего отчета или строится на базе существующих созданных разработчиками представлений с наложением необходимых фильтров, группировки и сортировки. Для отчета задается расписание построения, ответственный администратор, параметры уведомления.]

#### Формирование периодических отчетов

*Название сценария:* Формирование периодических отчетов.

*Назначение сценария:* Сценарий описывает процесс формирования периодических отчетов.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс формирования периодических отчетов по расписанию, уведомление пользователей о необходимости сформировать отчет для неавтоматизированных отчетов, уведомление о завершении формирования отчета и готовности его к выгрузке.]

#### Выгрузка данных

*Название сценария:* Выгрузка данных из системы статистики и аналитики

*Назначение сценария:* Сценарий описывает процесс выгрузка данных из системы статистики и аналитики.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс формирования периодических отчетов по расписанию, уведомление пользователей о необходимости сформировать отчет для неавтоматизированных отчетов, уведомление о завершении формирования отчета и готовности его к выгрузке.]

#### Формирование аналитических отчетов по запросу

*Название сценария:* Формирование аналитических отчетов по запросу.

*Назначение сценария:* Сценарий описывает процесс формирования аналитических отчетов по запросу для задач продуктовой аналитики.

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс формирования аналитических отчетов по запросу. ]

### Проведение A/B-тестирования

[раздел в работе]

### Мониторинг

#### Мониторинг работоспособности аппаратного обеспечения

*Название сценария:* Мониторинг работоспособности аппаратного обеспечения

*Назначение сценария:* Сценарий описывает процесс мониторинга работоспособности аппаратного обеспечения

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс мониторинга работоспособности аппаратного обеспечения. ]

#### Мониторинг доступности внешних систем

*Название сценария:*

Мониторинг доступности внешних систем *Назначение сценария:*

Сценарий описывает процесс мониторинга доступности внешних систем

*Участники сценария:*

* Администратор

*Предусловие:*

Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:*

[Сценарий описывает процесс мониторинга доступности внешних систем. ]

#### Мониторинг работоспособности компонентов платформы

*Название сценария:*

Мониторинг работоспособности компонентов платформы

*Назначение сценария:*

Сценарий описывает процесс мониторинга работоспособности компонентов платформы

*Участники сценария:*

* Администратор

*Предусловие:*

Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:*

[Сценарий описывает процесс мониторинга работоспособности компонентов платформы. ]

#### Мониторинг качества предоставления услуг

*Название сценария:*

* Мониторинг качества предоставления услуг

*Назначение сценария:*

Сценарий описывает процесс мониторинга качества предоставления услуг

*Участники сценария:*

* Администратор

*Предусловие:*

Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:*

[Сценарий описывает процесс мониторинга качества предоставления услуг. ]

#### Задание параметров мониторинга

*Название сценария:* Задание параметров мониторинга

*Назначение сценария:* Сценарий описывает процесс задания параметров мониторинга

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс задания параметров мониторинга: метрики, пределы допустимых значений, периоды обновления, способ хранения, интервал сбора]

### Поддержка пользователей

#### Создание заявки

*Название сценария:* Создание заявки по инциденту

*Назначение сценария:* Сценарий описывает процесс создание заявки по инциденту

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс создание заявки по инциденту администратором платформы через панель управления, по эл.почте, по телефону, на основе поступившей от клиентского продукта.]

#### Обработка заявки

*Название сценария:* Обработки заявки по инциденту

*Назначение сценария:* Сценарий описывает процесс обработки заявки по инциденту

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс обработки заявки: сбор диагностических данных, категоризация, приоритезация, назначение ответственных, отслеживание статуса, контроль сроков, информирование заявителей, закрытие и подтверждения устранения инцидента .]

#### Формирование отчетности

*Название сценария:* Формирование отчетности по пользовательской поддержке

*Назначение сценария:* Сценарий описывает процесс формирования отчетности по пользовательской поддержке

*Участники сценария:*

* Администратор

*Предусловие:* Администратор зарегистрирован и авторизован через интерфейс панели управления платформой. Администратору доступна соответствующая функциональность.

*Основные шаги:* [Сценарий описывает процесс формирования отчетности по пользовательской поддержке по расписанию, уведомление пользователей о необходимости сформировать отчет для неавтоматизированных отчетов, уведомление о завершении формирования отчета и готовности его к выгрузке.]

### Управление системой рекомендаций

[В процессе]

## Основные требования к развитию решения ИТВ

### Бизнес-требования

1. Решение ИТВ 2.0 должно предоставлять пользователям контентные системные и коммуникационные услуги в рамках B2C-взаимодействия.
2. Решение ИТВ 2.0 должно выступать поставщиком данных и предоставлять площадку для интеграции различных услуг и сервисов в рамках B2B-взаимодействия.
3. Решение ИТВ 2.0 должно выступать в качестве технического средства для построения системы предоставления видеоконтента в рамках B2G- и B2C-взаимодействия.
4. Решение ИТВ 2.0 должно предоставлять услуги пользователям на территории РФ с возможностью расширения географии присутствия за счёт дополнительных услуг и расширения территории действия лицензионных прав.
5. Решение ИТВ 2.0 должно обеспечивать лицензионную чистоту в отношении действующих прав на интеллектуальную собственность.

### Финансово-юридические требования

[TBD: Процедуры оплаты и требования к оферте]

[Соответствие формы оферты ст. 432 ГК РФ.]

[Надо проговорить с РТК - зависит от того, будет ли распространяться на пользователя ИТВ уже существующих SLA, TS и PP РТК.]

[Договор РТК-Смартлабс предполагает передачу неисключительных прав на платформу. Соответственно платформу в исходном виде и ее модификации невозможно использовать в качестве продукта для 3х лиц, а такое пожелание есть. ПИТВ 2.0 в идеале не должна содержать в себе элементы платформы Смартлабс.]

[проработать вопрос встраивания нашего плеера в Спутник]

### Требования к процессу разработки

1. Разработка отдельных компонентов решения и соответствующих модулей должна производиться параллельно. Принятые технические решения и особенности реализации независимых компонентов не должны влиять на остальные компоненты.
2. Разработка программных компонентов решения должна производится в рамках итеративного подхода к разработке программного обеспечения: работы должны выполнятся поэтапно и параллельно в рамках постепенно расширения функциональности и с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы.
3. В процессе разработки должен осуществляться контроль версий разрабатываемого программного обеспечения, с использованием систем контроля версий и принятых в их рамках методологий.

### Требования к процессу тестирования

### Требования к информационной безопасности

[*Требования к безопасности (защита контента, платформы, разделение доступа и пр.*]

[*Модель разграничения доступа*]

[*Логика доступности услуг в контексте мультискрина на различных продуктах и различных МРФ*]

### Требования к документированию

1. Основные технические решения и их реализация в рамках разработки решения должны быть задокументирована.
2. Документация решения должна включать следующие виды документов:
   * документация требований: функциональные требования, требования к возможностями и характеристикам решения и его компонентов; требования к качеству.
   * описания архитектуры решения и технических средств, взаимодействия компонентов и связей с внешними системами;
   * техническая документация: документация в составе исходных кодов, документация алгоритмов, программных и пользовательских интерфейсов;
   * пользовательская документация: руководства для конечных пользователей, системных администраторов, сотрудников техподдержки и т.д.
3. Процесс разработки документации решения должен учитывать модель и особенности разработки соответствующих программных компонентов, должен быть интегрирован в процесс разработки программных средств.
4. Документация решения должна разрабатываться по принципу «единого источника» (single sourcing).
5. Документация решения должна поддерживаться в актуальном состоянии.

### Требования к процессу поддержки

Техническая поддержка должна обеспечивать реакцию на инциденты в рамках целевых значений, указанных в таблице ниже.

Таблица реакций на инциденты.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип** | **Приоритет** | **Действия исполнителя** | **Сроки** |
| Аварийная | Приоритет 1 | Время реагирования | 15 минут |
| техподдержка |  | Время восстановления | 2 часа |
|  |  | Время решения | 7 дней |
| Поддержка по запросу | Приоритет 2 | Время реагирования | 30 минут |
|  |  | Время восстановления | 4 часа |
|  |  | Время решения | 14 дней |
|  | Приоритет 3 | Время реагирования | 1 час |
|  |  | Время восстановления | 24 дня |
|  |  | Время решения | 30 дней |
| Информационные запросы | Приоритет 4 | Время реагирования | 1 час |
|  |  | Время решения | 24 часа |

Время реагирования определяется как временной интервал от момента направления запроса технической поддержке до момента, когда специалист поддержки осуществит первичный контакт с инициатором запроса или начнет обработку запроса.

Время восстановления определяется как временной интервал, необходимый специалистам поддержки для решения проблемы или устранения аварийной ситуации. При этом допускается реализовывать временное обходное решение, позволяющее скомпенсировать негативный эффект при предоставлении сервиса пользователя, устранить проблемы безопасности, восстановить контроль и управление оборудованием. Отсчёт времени начинается с момента направления запроса.

Время решения определяется как временной интервал, необходимый специалистам поддержки для окончательного решения запроса. Отсчёт времени начинается с момента направления запроса.

**Приоритет 1:** под аварийной ситуацией понимается неотложная и серьёзная проблема, которая вызывает значительные ограничения в эксплуатации и обслуживании, требующие скорейшего устранения, в том числе с использованием значительных ресурсов круглосуточно для того, чтобы ликвидировать проблему.

Некоторые примеры неисправностей вида Приоритет 1 (аварийная ситуация):

* отказы оборудования, его блоков (модулей) и/или программного обеспечения, влияющие на предоставление сервиса пользователям, в случае, если уровень сервиса нельзя восстановить стандартными процедурами техобслуживания, предусмотренными для данного оборудования;
* оборудование находится в нерабочем состоянии, что в свою очередь негативно сказывается на предоставлении услуг пользователям;
* многочисленный выход из строя однотипных блоков (модулей) оборудования или нарушения их основных функций;
* отказы оборудования, влияющие на вещание телеканалов, имеющих социальное и политическое значение;
* проблемы в информационной безопасности оборудования.

**Приоритет 2:** оборудование в значительной степени не работоспособно, что оказывает серьёзное воздействие на предоставление сервиса; проблема требует скорейшего разрешения в том числе с использованием значительных ресурсов в течение полного рабочего дня.

К Приоритету 2 относятся проблемы оборудования, вызывающие ограничения при эксплуатации и обслуживании, однако не затрагивающие основные функции системы.

Некоторые примеры неисправностей вида Приоритет 2:

* потеря некоторых не основных функций оборудования;
* нарушения в работе систем управления и мониторинга, не позволяющие осуществлять контроль и управление оборудованием посредством их использования;
* проблемы, приводящие к снижению качественных показателей предоставляемых услуг, а также надёжности и устойчивости функционирования Оборудования.

**Приоритет 3:** Работоспособность оборудования значительно уменьшилась, но большинство функций сохранено.

Некоторые примеры неисправностей вида Приоритет 3:

* проблемы, влияющие на администрирование оборудования, стандартное обслуживание и диагностику, но не влияющие на предоставление сервиса пользователям;
* проблемы при выполнении документированных процедур;
* ошибки в документации, приводящие к проблемам в эксплуатации;
* невозможность производить некоторые операции по эксплуатации;
* отказы отдельных, блоков (модулей) оборудования, не влияющие на предоставление сервиса пользователям;
* эпизодические нерегулярные сбои в работе платформы, не приводящие к прерыванию трафика;
* сложности, связанные с неправильными измерительными данными, полученными с использованием внутренних функций оборудования диагностики;
* незначительные неудобства в работе с системой управления, или невозможность выполнить некоторые операции без внедрения обходного решения.

**Приоритет 4:** реагирование на поступающие информационные запросы.

[TBD: Схема службы поддержки 3 линии]

[Описания отчётов ]

### Требования к услугам

#### Контентные услуги

1. Контентные услуги должны включать в себя:
   * Просмотр трансляции эфира ТВ-каналов.
   * Просмотр архивных передач ТВ-каналов.
   * Просмотр CoD-контента (фильмы, сериалы, мультфильмы и клипы).
   * Прослушивание аудиокниг, радиоэфира, музыкальных произведений.
   * Доступ к услуге «Караоке».
   * Доступ к видеотрансляциям (лекции, семинары, конференции и пр.)
   * Доступ к информационным сервисам (прогноз погоды, курсы валют, пробки, новости и т.д.).
   * Доступ к сервису облачных игр.
2. Доступ к контентным услугам должен предоставляться пользователю в виде:
   * бесплатный доступ;
   * разовая подписка;
   * периодическая подписка;
   * предоплата (подписка на ещё недоступный контент)
   * аренда CoD-контента;
   * предустановленный на устройство доступ;
   * промокоды и промоакции;
   * подарки;
   * синхронизация покупок с внешними сервисами. 3.Цена услуги должна демонстрироваться пользователю перед покупкой.
3. Пользователь должен иметь возможность оплатить услугу при помощи привязанный к его профилю банковской карты и/или дополнительных внешних сервисов оплаты.
4. Пользователь должен иметь возможность оплатить услугу посредством списания средств с личного счёта, привязанного к его профилю.

#### Сервисные услуги

##### Управление просмотром телеэфира

1. Пользователю должна быть доступна функция паузы эфира (PauseLive).
2. Пользователю должна быть доступна функция перемотки эфира (TSTV).
3. Пользователю должна быть доступна функция просмотра закончившихся телепередач, записи телепередач (nPVR).

##### Услуга «Мультискрин»

1. Услуга «Мультискрин» должна предоставлять пользователю возможность переносить данные профиля, настройки между своими устройствами и состояние использования.
2. Услуга должна предоставлять пользователю возможность переносить купленный контент и услуги между своими устройствами.
3. Услуга должна предоставлять пользователю возможность переносить информацию о сохраненных позициях просмотра контента между своими устройствами.
4. Услуга должна накладывать ограничения на подключаемые устройства. [какие?]

[Логика доступности услуг в контексте мультискрина на различных продуктах и различных МРФ]

##### Услуга «Second screen»

1. Услуга «Second screen» должна предоставлять пользователю возможность без прерывания просмотра контента:
   * получать дополнительную информацию о фильме;
   * общаться с другими пользователями при помощи комментариев;
   * получать рекомендации.
2. Услуга должна предоставляться посредством технологий Airplay, WiDi, Chromecast, и Second screen.
3. Услуга должна предоставлять интерфейс управления просмотром на подключенном устройстве (ТВ-приставке или SmartTV)

##### Сервис рекомендаций

[В написании]

##### Услуга просмотра контента без подключения к интернету

1. Услуга просмотра контента без подключения к интернету должна быть доступна на всех мобильных клиентских приложениях.
2. Услуга должна быть доступна всем пользователям мобильных клиентских приложениях вне зависимости от их географического местоположения.

##### Услуга просмотра контента с USB-носителя

[В написании]

##### Родительский контроль

1. Услуга родительского контроля должна предоставлять пользователю возможность устанавливать настройки профиля родительского контроля, накладывающие следующие ограничения на контент:
   * демонстрация в клиентских продуктах только контента, удовлетворяющего настройкам профиля;
   * «безопасная» выдача контента в результатах поиска и рекомендациях;
   * ограничения на совершение покупок.
2. Услуга родительского контроля должна быть привязана к профилю пользователя.
3. Сценарии ограничения доступа должна основываться на механизме защиты доступа через PIN-коды. По умолчанию активация механизма защиты осуществляется в момент смены PIN-кода.

##### Поиск по контенту

##### Просмотр на связанных устройствах

[DLNA/AirPlay/WiDi]

##### Услуги телефонии

##### Услуга «Подарки»

##### Услуга «Кинорулетка»

##### Файловые квоты

##### Услуга «Пульт ДУ»

1. Услуга «Пульт ДУ» должна предоставлять пользователю возможность использования мобильного устройства в качества пульта дистанционного управления ТВ-приставкой и SmartTV.

##### Услуги внешних сервисов

[оплата ЖКХ, мобильной связи, штрафов ГИБДД, ставок на спортивные состязания и пр.]

#### Оплата услуг

1. Пользователь должен иметь возможность оплатить услугу/пакет услуг посредством списания с личного счёта абонента.
2. Пользователь должен иметь возможность оплатить услугу/пакет услуг посредством привязанной банковской карты.
3. Пользователь должен иметь возможность оплатить услугу/пакет услуг посредством внешней платёжной системы.
4. Пользователь должен иметь возможность оплатить услугу/пакет услуг посредством платежной системы, привязанной к операционной системе устройства.
5. Пользователь должен иметь возможность оплатить услугу/пакет услуг посредством списания со счёта мобильного телефона.

#### Коммуникационные сервисы

[раздел в написании]

### Требования к клиентским продуктам

[ТВD: часть из ТЗ для мобильных приложений]

[кастомизация отображения пользовательского интерфейса на клиентских продуктах по желанию абонента]

[сохранение состояния клиентского продукта и перенос состояний между клиентскими продуктами абонента]

[персонализация порядка отображения контента, ТВ-каналов на основе явных инструкций пользователя (помеченный “избранное”, “нравиться” и пр. контент должен отображаться в первую очередь]

[мгновенный перенос покупок между клиентскими продуктами пользователя]

[перенос профилей между устройствами]

[блоки сервиса рекомендаций]

[Должны сохранятся состояния клиентских продуктов и переностся данные между различными клиентскими продуктами одного абонента: личные настройки пользовательского интерфейса клиентских продуктов по желанию абонента; личные настройки порядка отображения контента и ТВ-каналов на основе команд пользователя или алгоритмов платформы (контент, помеченный как «избранное», «нравится» и т.д., часто просматриваемые ТВ-каналы, предложения системы рекомендаций); купленный контент (в режиме реального времени); пользовательские профили.]

### Требования к клиентскому оборудованию

[Аппаратные требования для мобильных платформ, smartTV, веб-плеера (браузеры): ios 7+ Android 4.2+ SmartTV 2013 + ]

### Требования к Платформе интерактивного телевидения 2.0

#### Основные нефункциональные требования

1. Платформа 2.0 должна состоять из двух групп компонентов:
   * Система управления платформой (СУП) — отвечает за управлением продуктами, контентом, оплатой, абонентами, системой рекомендаций, сбор статистики и аналитики, мониторинг и взаимодействие с технической поддержкой.
   * Система доставки контента (СДК) – отвечает за доставку контента пользователям посредством различных механизмов передачи данных.
2. Логическая модель данных должна соответствовать требованиям, предъявляемым к услугам и платформы «Интерактивное телевидение Ростелеком» 2.0.

##### Требования к нагрузке

[Объём аудитории сервиса и нагрузки платформы в целом]

[Сейчас 3 000 000 абонентов на приставках. Плюс таблица планов для разных продуктов. Запрошено у РТК 13.11]

[Рост до 4500000 приставок и можно заложить бурный рост устройств]

##### Требования к надежности платформы

1. Платформа 2.0 должна функционировать в круглосуточном режиме 24x7.
2. Архитектура Платформы 2.0 должна реализовывать выполнение требований к восстановлению, указанных в разделе [*«Требования к процессу поддержки»*](#требования-к-процессу-поддержки).

#### Основные функциональные требования

Функциональные требования к решению разделяются на функциональные блоки — наборы функций со общей задачей.

За реализацию функциональных блоков отвечают компоненты систем управления платформой и доставки контента:

* Система управления платформой:
  + Управление контентом.
  + Управление услугами.
  + Управление стоимостью услуг.
  + Управление абонентами.
  + Управление клиентскими продуктами.
  + Сбор и анализ статистики.
  + Мониторинг работы платформы.
  + Пользовательская поддержка.
  + Панель управления платформой.
  + Сервис рекомендаций.
  + Взаимодействие с внешними системами.
* Система доставки контента:
  + Требования к защите контента.
  + Требования к доставке контента.

##### Управление контентом

**Загрузка контента и метаданных из внешних источников:**

1. Система управления платформой должна реализовывать получения контента и связанных с ним метаданных из внешних источников (поставщики контента правообладателей и поставщики метаданных) в автоматическом и ручном режимам.
2. Система управления платформой должна предоставлять средства управления и настройки автоматической загрузки контента из внешних систем.
3. Система управления платформой должна предоставлять администратору инструменты для создания и настройки инструкций и шаблонов загрузки контента в автоматическом режиме.
4. Система управления платформой должна отправлять администратору уведомления об успешном/неудачном завершении процесса загрузки контента.
5. Система управления платформой должна предоставлять интерфейс для управления загрузкой контента и метаданных в ручном режиме.

**Управление метаданными и доставкой контента:**

1. Система управления платформой должна предоставлять инструменты управления загруженных метаданных и доставкой загруженного контента пользователям.
2. Система управления платформой должна предоставлять администраторам графический интерфейс для выставления и изменения следующих параметров:
   * формат файлов мастер-копий контента;
   * правила распространения контента, такие как:
     + территориальные ограничения;
     + возрастные ограничения;
     + ограничения срока доступа;
   * привязка метаданных к контенту;
   * принадлежность определенного контента к контентной группе;
   * метаданных;
   * объединение серий в сезоны и сериалы.
3. Система управления платформой должна осуществлять управление ТВ-каналами, доступными пользователям:
   * создание;
   * задание и изменение параметров;:
     + определение источника вещания;
     + настройки записи;
     + привязка к EPG;
   * задание параметров доступности;
   * привязка к пакетам ТВ-каналов;
   * удаление.
4. Система управления платформой должна предоставлять администраторам инструменты для привязки контента к определенному соглашению с правообладателем.

**Взаимодействие с правообладателями:**

1. Система управления платформой должна отслеживать наличие разрешения правообладателя на распространение контента на основании лицензионного соглашения.
2. Система управления платформой должна предоставлять администраторам инструменты управления документами-соглашениями с правообладателями:
   * создание соглашений и задание параметров;
   * отслеживание сроков действия и изменений в лицензионных соглашениях, которые могут повлиять на территориальную доступность контента, способы его распространения и другие параметры;
   * уведомление администраторов в случае изменения условий и прав распространения контента.
   * автоматическое приостановление распространения контента, права на распространения которого были отозваны; дальнейшее распространение такого контента должно быть продолжено только после подтверждения администратором.

**Управление контентными справочниками:**

1. Система управления платформой должна хранить и предоставлять администраторам следующие контентные справочники:
   * справочник жанров;
   * справочник стран-производителей контента;
   * справочники съёмочной группы: режиссеров, актеров, сценаристов, продюсеров и других;
   * справочники ограничений показа.
2. Система управления платформой должна предоставлять инструменты добавления, изменения и удаления записей справочников с поддержкой целостности и контролем корректности ввода. Справочники должны вестись централизованно, с логированием всех вносимых изменений.

**Управление EPG:**

1. Система управления платформой должна предоставлять инструменты управления и редактирования EPG-данных, получаемых из внешних источников.
2. Система управления платформой должна позволять администратору редактировать EPG-данные, в соответствии с предопределенными правилами, для вывода на клиентские устройства, в том числе изменять описания программ, графические материалы и т.д.

##### Управление услугами

1. Система управления платформой должна предоставлять администраторам инструменты для создания и редактирования услуг:

* создание услуги;
* задание характеристик услуги:
  + по стоимости: платная/бесплатная; единая стоимость для всех МРФ/разная стоимость для разных МРФ и регионов;
  + по доступность услуги в различных МРФ и регионах;
  + по обязательности: обязательная/необязательная для тарифного плана;
  + по типу оплаты: единовременный платеж/периодический платеж;
  + по возможности зависимости: может/не может быть зависима;
  + по способам оплаты: авансовая/кредитная;
  + по доступности пользователю: явная/неявная.
* задание типа услуги:
  + пакет ТВ-каналов;
  + единица VoD;
  + пакет VoD;
  + управление просмотром;
  + сервисная услуга;
  + услуга внешних сервисов;
  + аренда/покупка музыки, аудиокниг.
* привязка групп контента к услугам;
* указание клиентских продуктов, на которых возможно оказание услуги;
* редактирование и удаление услуги;
* задание правил тарификации и их параметров.

1. Система управления платформой должна предоставлять администраторам инструменты создания услуг на основе шаблонов:
   * создание шаблонов услуг на основе существующих услуг или параметров, заданных администратором;
   * создание услуги на основе существующих шаблонов услуг;
   * задание при создании услуги порядка наследования параметров шаблонов для их редактирования.

**Управление тарифными планами:**

1. Система управления платформой должна предоставлять администраторам инструменты для управления тарифными планами:
   * создание тарифного плана;
   * привязка услуг к тарифному плану;
   * проверка услуг на совместимость;
   * привязка тарифного плана к МРФ и регионам;
   * привязка новых тарифных планов к существующим и определение правил перехода.

**Управление правилами тарификации:**

1. Система управления платформой должна предоставлять администраторам инструменты для создания, изменения и удаления правил тарификации услуг:
   * создание правила тарификации;
   * задание правил через графический интерфейс:
     + задание вводных переменных;
     + задание вводных параметров;
     + задание соотношений;
   * сохранение шаблонов правил на основе существующих;
   * создание правил на основе шаблонов.

**Управление тарификацией услуг:**

1. Система управления платформой должна предоставлять администраторам инструменты для управления тарификацией услуг:
   * привязка правил тарификации к услугам;
   * задание параметров тарификации для услуги.

**Управление интеграции с биллинговыми системами:**

1. Система управления платформой должна предоставлять администраторам инструменты для контроля интеграции с биллинговыми системами: проверки корректности соответствия вновь созданных услуг и тарифных планов в системе управления платформой и биллинговых системах.

**Управление скидками и промоакциями**

1. Система управления платформой должна предоставлять администраторам инструменты для управления скидками и промоакциями.

[TBD]

##### Управление пользователями

1. Система управления платформой должна предоставлять администраторам инструменты для поиска пользователей по параметрам:
   * территориальное расположение;
   * клиентский продукт;
   * тип клиентского устройства;
   * версия прошивки/приложения;
   * тарифный план;
   * подключенные услуги;
   * уникальный идентификатор пользователя;
   * MAC-адрес устройства;
   * идентификаторы внешних сервисов.
2. Система управления платформой должна предоставлять администраторам инструменты для объединения абонентов в группы на основе общих параметров:
   * территориальное расположение;
   * клиентский продукт;
   * тип клиентского устройства;
   * версия прошивки/приложения;
   * тарифный план;
   * подключенные услуги.
3. Система управления платформой должна предоставлять администраторам инструменты для управления абонентами и группами абонентов:
   * подключение и отключение услуг;
   * статистика потребления услуг (просмотренный контент, покупки, и т.д.);
   * отправление сообщений и уведомлений;
   * проведение опросов среди пользователей; опросы должны создаваться администратором через интерфейс системы управления платформой или загружаться в предопределенных форматах.
4. Система управления платформой должна предоставлять администраторам инструменты для отправки сообщений и уведомлений пользователям и их группам по сформированным правилам:

* напоминания, установленные пользователем:
  + напоминания о начале телепередачи;
  + уведомления о доступности предоплаченного контента;
  + уведомления о появлении контента, заинтересовавшего пользователя ранее;
  + сообщения об исчерпании денежных средств на личном счёте;
  + уведомления об окончании срока действия услуг (окончание срока аренды, недоступность контента из-за окончания лицензионных прав на контент и т.д.);
  + уведомления об изменениях в тарифном плане, условиях оказания услуг и т.д.;
  + уведомления о необходимости обновления прошивки/приложения;
  + оповещения о чрезвычайной ситуации.

##### Управление клиентскими продуктами

1. Система управления платформой должна предоставлять администраторам инструменты для управления клиентскими продуктами (абонентскими устройствами и клиентскими приложениями):
   * объединение клиентских продуктов в группы;
   * просмотр зарегистрированных абонентских устройств и клиентских приложений;
   * управление параметрами клиентских продуктов и групп;
   * обновление прошивок абонентских устройств и версий клиентских приложений;
   * сопоставление клиентских продуктов и доступных услуг;
   * проведение мониторинга качества предоставления услуг.
2. Система управления платформой должна предоставлять администраторам инструменты для объединения клиентских продуктов в группы на основе общих параметров:
   * территориальное расположение;
   * клиентское устройство;
   * тип клиентского устройства;
   * версия прошивки/приложения.
3. Система управления платформой должна предоставлять администраторам инструменты для просмотра зарегистрированных клиентских продуктов, истории изменений параметров, типов устройств, истории изменений версий прошивок абонентских устройств и клиентских приложений.
4. Система управления платформой должна предоставлять администраторам инструменты для управления клиентскими продуктами и их группами: изменение параметров, обновление прошивок абонентских устройств и версий клиентских приложений, принудительная перезагрузка, активация/деактивация.
5. Система управления платформой должна предоставлять администраторам инструменты проведения удаленного мониторинга качества связи, работоспособности оборудования и ПО клиентских продуктов.

##### Сбор и анализ статистики

1. Система управления платформой должна предоставлять администратору инструменты для проведения анализа и создания отчётов:
   * по востребованности/продажам продуктов/услуг/контента; [требования будут сформированы рабочей группой РТК, письмо «Требования к системе статистики.» 24.11.2014 от Рубец Е.В.];
   * по использованию продуктов: востребованные функции, отслеживание взаимодействия с пользовательским интерфейсом;
   * по анализу качества (по воспринимаемым абонентами характеристикам).
2. Система управления платформой должна предоставлять возможности для анализа пользовательского поведения, а также:
   * выявления моделей потребления контента и сегментов, демонстрирующих разные модели при помощи алгоритмов машинного обучения;
   * поиска источников и способов получения дополнительной информации о пользователе;
   * сбора данных о потреблении контента на других онлайн-сервисах; анализ просмотренного ТВ-контента;
   * распознавания речи в контенте и дальнейшего анализа полученных данных (технология SpeechKit).

**Сбор и хранение данных:**

1. Система управления платформой должна собирать следующую информацию:
   1. Устройства и приложения, привязанные к пользователям:
      * типы устройств, названия устройств, версии прошивок абонентских устройств и версии клиентских приложений с историей изменений;
      * данные об активациях абонентских устройств и установках приложений.
   2. Подключения устройств/приложений, привязанных к пользователям:
      * данные об авторизациях;
      * географические привязки;
      * операторы доступа в интернет;
      * ширина канала передачи данных;
      * длительность сессий;
      * тип и версия ОС, версии клиентских приложений и прошивок абонентских устройств.
   3. Потребление услуг пользователями:
      * контентных услуг;
      * сервисных услуг;
      * коммуникационных услуг.
   4. Качество предоставления услуг:
      * время отклика на запросы;
      * качество видеоизображения;
      * типы и количество ошибок;
      * время решения запросов поддержкой пользователей (в зависимости от типа запроса);
      * длительность недоступности компонентов платформы.
   5. Статистика действий пользователей при работе с клиентским продуктом.
   6. Действия пользователей при использовании клиентского продукта:
      * статус просмотра;
      * длительность просмотра;
      * время и дата просмотра;
      * взаимодействие с пользовательским интерфейсом;
      * переходы между экранами;
      * просмотры метаданных;
      * поисковые запросы.
   7. Пользователи:
      * типы учетных записей и их статусы,;
      * привязки к регионам;
      * социальные, демографические и географические данные;
      * привязка к учётным записям внешних OSS/BSS-систем;
      * доступные продукты, тарифы и услуги;
      * данные по предпочтениям (любимые телеканалы/ТВ-программы, отметки «Нравиться», «Хочу посмотреть» и пр.).
   8. Данные о монетизации:
      * доходы от продажи контента;
      * оплата различными способами (платежи банковской картой, списания с личного счёта, использование других платежные систем).
2. Система управления платформой должна хранить собранные данные в течении не менее 5 лет, включая историю их изменений.
3. Система управления платформой должна хранить данные с учётом истории их изменения, целостности, согласованности и предметной ориентированности.
4. Система управления платформой должна обеспечивать построение витрин данных и формирование произвольной отчетности на основе собранной информации.
5. Система управления платформой должна собирать и агрегировать данные из внешних систем статистики, таких как: Google Analytics, iTunes Connect, Flurry Analytics, Mix Panel и других.

**Агрегация и создание отчётов:**

1. Система управления платформой должна предоставлять администраторам инструменты для выгрузки исходных и агрегированных данных, включая социальные, демографические и географические данные.
2. Система управления платформой должна хранить агрегированные данные в течении неограниченного срока времени.
3. Система управления платформой должен позволять построение «воронки продаж» всех услуг от активации абонентского устройства/установки клиентского приложения до покупки конкретной услуги с возможностью фильтрации и построения срезов данных и сортировкой по типам устройств/версий/местоположению.
4. Система управления платформой должна предоставлять администратору инструменты для создания и изменения отчётных формы: изменять состав отчетов, применять фильтры и сортировку, изменять выходные форматы данных и т.д.
5. Система управления платформой должна предоставлять администратору инструменты создания статистических отчётов на основе данных отчётных форм.
6. Система управления платформой должна предоставлять администратору инструменты для создания отчётов по временным периодам в соответствии с заданными правилами, с возможностью группировки по тарифным планам, МРФ, регионам, промоакциям, предопределенным группам абонентов и т.д. Отчеты должны включать в себя:
   * количество пользователей установивших приложение/активировавших абонентское устройство;
   * количество пользователей обновивших приложение/прошивку абонентского устройства;
   * количество пользователей запустивших приложение/включивших абонентское устройство;
   * количество заходов в приложение/включений абонентского устройства;
   * количество уникальных пользователей;
   * длительность сессий пользователей;
   * частота использования приложения/устройства;
   * показатели возврата пользователей после установки/активации;
   * количество пользователей авторизовавшихся в приложении;
   * количество пользователей привязавших учетную запись абонента других сервисов ПАО «Ростелком»;
   * данные по производителям, типам и версиям ОС, версиям приложений;
   * распределение контента, разделов, телеканалов по востребованности;
   * количество ошибок и проблемных сценариев в приложении/устройстве;
   * социальные, демографические и географические данные пользователей;
   * количество подписок/покупок/транзакций.
7. Система управления платформой должна выгружать данные в заранее определенных форматах для проведения сверок с данными из внешних систем.
8. Система управления платформой должна поддерживать загрузку/выгрузку данных о доходах от продажи контента в системы бухгалтерского учета при его оплате через личный кабинет, внешние платежные системы и привязанной банковской картой.

**Аналитика:**

1. Система управления платформой должна предоставлять администраторам аналитические инструменты для:
   * визуального представление данных;
   * обнаружение данных (скрещивание/наложение показателей, выявление сегментов);
   * выгрузки детальных данных по выявленным в процессе обнаружения данных сегментам;
   * генерация отчетов за заданные временные периоды;
   * задания уведомлений о событиях и показателях (критически низкий, аномально высокий, достижение уровня);
   * проведения A/B-тестирования (пользовательского интерфейса, настроек пульта дистанционного управления абонентского устройства).
2. Система управления платформой должна предоставлять администраторам инструменты для получения аналитических данных:
   * популярность ТВ-каналов, ТВ-передач, CoD, CoD-пакетов и их групп по жанрам и поджанрам; востребованность определенного качества изображения, режимов линейного/нелинейного просмотра;
   * потребление контента в режиме реального времени;
   * поведение пользователя при просмотре контента: линейный/нелинейный, время между переключения и т.д.;
   * вовлеченность пользователя: частота выставления оценок контенту, активность в соцсетях и т.д.;
   * монетизация;
   * эффективность сервиса рекомендаций: отслеживание конверсии, использования;
   * поведение абонента при переходах (навигация) между экранами пользовательского интерфейса клиентских продуктов;
   * часто используемые способы поиска и поисковые запросы;
   * длительность просмотра контента и соотношение популярности контента и его рейтинга на основе отзывов пользователей, списков рекомендованного контента и промоконтента.
3. Система управления платформой должна предоставлять администраторам инструменты для выявления на основе анализа отчетности:
   * сегментов абонентов, групп VoD и ТВ-каналов, ТВ-передач;
   * предпочтений на основе анализа интересов абонентов и их групп.

##### Мониторинг работы платформы

[Необходимо актуализировать после завершения технической проработки]

1. Система управления платформой должна осуществлять круглосуточный мониторинг и контроль параметров работы программно-технических средств, используемых платформой:
   * доступность;
   * работоспособность;
   * количество соединений;
   * количество запросов и операций.
2. Система управления платформой должна осуществлять мониторинг доступности внешних систем (OSS/BSS, ЕКП и других).
3. Система управления платформой должна осуществлять круглосуточный мониторинг и контроль параметров качества связи:
   * потеря пакетов;
   * сетевая задержка;
   * занимаемая полоса пропускания;
   * максимально доступная полоса пропускания.
4. Система управления платформой должна предоставлять администратору инструменты для управления параметрами мониторинга компонентов/групп компонентов платформы:
   * метрики;
   * пределы допустимых значений;
   * период обновления;
   * способ хранения;
   * интервал сбора.
5. Система управления платформой должна отправлять уведомления администраторам в случае выхода метрики за заданные допустимые значения посредство электронной почты и SMS-сообщений.
6. Система управления платформой должна отслеживать и сохранять действия пользователей при использовании клиентских продуктов и качество предоставления услуг:
   * Действия пользователей:
     + статус просмотра;
     + длительность просмотра;
     + время и дата просмотра;
     + взаимодействие с интерфейсом клиентских продуктов;
     + переходы между экранами интерфейса;
     + просмотр метаданных;
     + поисковые запросы.
   * Качество предоставления услуг:
     + время отклика на запросы;
     + качество видеоизображения;
     + типы и количество ошибок;
     + время решения запросов технической поддержкой (в зависимости от типа запроса);
     + длительность недоступности компонентов платформы.

**Хранение данных и формирование отчётов о результатах мониторинга:**

1. Система управления платформой должна обеспечивать хранение исходных данных мониторинга в течение 3 месяцов.
2. Система управления платформой должна обеспечивать хранение агрегированных данных мониторинга в течении 2 лет.
3. Система управления платформой должна предоставлять администраторам инструменты построения отчетности на основании исходных данных с возможностью детализации до минимального хранимого периода обновления.
4. Система управления платформой должна предоставлять администраторам инструменты просмотра предварительно настроенных наборов данных.

##### Пользовательская поддержка

1. Система управления платформой должна осуществлять сбор и обработку заявок об инцидентах в работе платформы и обращений пользователей.
2. Система управления платформой должна осуществлять систематизацию, анализ и формирование отчётов на основе данных об инцидентах и обращениях.
3. Система управления платформой должна обеспечивать выполнение требований к процессу поддержки пользователей (подробнее — [*«Требования к процессу пользовательской поддержки»*](#требования-к-процессу-пользовательской-поддержки)).

**Регистрация заявок и сбор сведений**

1. Система управления платформой должна обеспечивать создания заявок по инцидентам:
   * администраторами через раздел панели управления СУП, через обращение по электронной почты или звонком на телефон службы поддержки;
   * автоматически на основе уведомлений мониторинга.
2. Система управления платформой должна обеспечивать создания заявок по обращениям пользователей на основе:
   * обращений в колл-центр по телефону, электронной почты или через ЕЛК;
   * обращений через интерфейс клиентских продуктов.
3. Система управления платформой должна добавлять данные автоматически генерируемого отчёта от клиентского продукта к заявке, созданной на основе обращений через интерфейс клиентских продуктов и инцидентов:
   * UID продукта абонента;
   * IP-адрес продукта;
   * местонахождение продукта абонента;
   * тип продукта;
   * серийный номер (при наличии);
   * версия ПО клиентского приложения или абонентского устройства (при наличии);
   * версия и название браузера (при наличии);
   * скорость соединения до сервера, с которого осуществляется трансляция контента;
   * IP-адрес CDN-сервера, с которого осуществляется доставка контента до клиентского продукта;
   * ID в CDN NGenix;
   * дата и время обращения
   * тип ошибки и комментарий пользователя;
   * скорость и тип сетевого подключения;
   * последние 100 API-вызовов клиентского продукта.
4. Система управления платформой должна автоматически или на основании запроса сотрудника пользовательской поддержки добавлять к зарегистрированной заявке дополнительные данные (при их наличии):
   * группы устройств привязанные к учетной записи пользователя (включая историю активации);
   * состав приобретенных пакетов и услуг;
   * история обращений пользовательских устройств к отдельным компонентов платформы.

**Обработка заявок**

1. Система управления платформой должна позволять сотрудникам пользовательской поддержки проводить категоризацию и приоритизацию заявки и назначать ответственного за её обработку.
2. Система управления платформой должна предлагать ответственному за заявку наиболее вероятные пути её решения в соответствии с классификацией заявки.
3. Система управления платформой должна предоставлять администраторам и сотрудникам пользовательской поддержки инструменты автоматизации процессов отслеживания статуса заявки, контроля сроков, информирования заявителей, закрытия заявок и подтверждения устранения инцидентов.
4. Система управления платформой должна отслеживать проводимые работы и регистрировать затраченное время по заявкам, категориям заявок и сотрудникам.

**Отчетность и анализ**

1. Система управления платформой должна должна предоставлять администраторам инструменты для формирования отчетов о работе пользовательской поддержки.

##### Панель управления платформой

1. Система управления платформой должна предоставлять администраторам набор инструментов для управления в виде веб-приложения — панели управления платформой.
2. Панель управления платформой должна предоставлять администраторам инструменты для регистрации новых администраторов, управления правами в рамках модели разграничения доступа, аутентификации и авторизации.
3. Управление доступом к панели управления должно происходить на основе администраторских ролей. Права доступа администраторов платформы группируются с учетом должностных обязанностей, образуя роли. Должны быть реализованы следующие роли:
   * контент-менеджер — управление контентом;
   * продуктолог — управление услугами, тарификацией;
   * сотрудник мониторинга — мониторинг работоспособности платформы и качества предоставления услуг;
   * сотрудник поддержки — техническая поддержка пользователей и администраторов платформы, управление абонентами;
   * аналитик — формирование отчетности, анализ данных, управление системой рекомендаций;
   * системный администратор — управление конфигурациями, управление настройками платформы, управление клиентскими продуктами.
4. Панель управления платформой должна предоставлять администраторам инструменты поиска, фильтрации, сортировки по выбранным параметрам и совершения групповых операций.
5. Панель управления платформой должна предоставлять администраторам инструменты для выгрузки данных в машиночитаемых форматах с учетом их уровня доступа.
6. Панель управления платформой должна осуществлять проверку корректности ввода данных.
7. Панель управления платформой должна предзаполнять справочные значения из справочников.
8. Панель управления платформой должна осуществлять персонификацию интерфейса для каждого администратора в зависимости от его полномочий в рамках модели разграничения доступа.
9. Панель управления платформой должна предоставлять администраторам инструменты для настройки отдельных элементов интерфейсов (изменение порядка показа элементов на странице, сокрытие элементов, сохранение настроек выдачи элементов и т.д.)
10. Панель управления платформой должна уведомлять администраторов о наступивших событиях, поступлении новых задач и сбоях в зависимости от их полномочий в рамках модели разграничения доступа.
11. Панель управления платформой должна предоставлять администраторам инструменты размещения и публикации информационных материалов (инструкций, документации, обучающих материалы и других).

##### Сервис рекомендаций

1. Система управления платформой должна реализовывать сервис рекомендаций, единый для всех клиентских продуктов, позволяющий на основе данных пользовательских профилей, пользовательского поведения, базы данных о контенте и данных внешних источников формировать персонализированные списки рекомендованного контента.

**Сбор данных:**

1. Сервис рекомендаций должен собирать исходные и агрегированные данные на основе данных статистики и мониторинга:
   * действия пользователя при использовании клиентских продуктов:
     + статус просмотра;
     + длительность просмотра;
     + время и дата просмотра;
     + взаимодействие с интерфейсом клиентских продуктов;
     + переходы между экранами интерфейса;
     + просмотр метаданных;
   * данные контекста потребления:
     + IP-адрес и история его изменения;
     + тип пользовательского продукта;
     + характеристики устройства, на котором происходит взаимодействие с пользователем;
     + местоположение клиентского продукта;
   * данные профиля абонента:
     + демографические;
     + социальные;
     + географические.
2. Сервис рекомендаций должен собирать данные введенные администратором: новинки и промоконтент.
3. Сервис рекомендаций должен собирать данные из внешних источников:
   * рейтинги и рекомендации сторонних сервисов;
   * данные о популярности контента;
   * новинки;
   * данные о пользователях и их предпочтениях из социальных сетей и других источников;
   * обезличенные профили поведения.

**Формирование списков рекомендованного контента:**

1. Система управления платформой должна по предопределенным алгоритмам, основываясь на собранных данных:
   * формировать единые для всех клиентских продуктов отсортированные списки рекомендованного контента для каждого абонента;
   * изменять сформированные списки рекомендованного контента под контекст потребления: используемый клиентский продукт, качество связи, местонахождение, тип просмотра и т.д.;
   * обеспечить обновление списков рекомендованного контента при каждой смене экрана интерфейса пользователем;
   * в случае недостатка вычислительных мощностей обновлять списки рекомендованного контента равномерно по абонентам;
   * исключить попадание в списки рекомендованного контента недопустимого к показу абоненту контента исходя из ограничений (возрастных, ограничений на показ для региона и т.д.).
2. Система управления платформой должна осуществлять формирование списков рекомендованного контента:
   * в автоматизированном режиме на основе алгоритмов машинного обучения, коллаборативной фильтрации;
   * с возможностью в ручном/полуавтоматическом режиме контролировать и при необходимости вносить коррективы в списки;
   * с периодическим обновлением списков рекомендованного контента (недопустима ситуация, при которой абонент в течении длительного времени получает статичные списки рекомендованного контента).
   * с учётом данных обратной связи сервиса рекомендаций для каждого абонента:
     + конверсии на основе списков рекомендованного контента;
     + монетизации на основе списков рекомендованного контента.

##### Взаимодействие с внешними системами

1. Платформа 2.0 должна обеспечивать взаимодействие с автоматизированными системами расчётов, используемых сервисами ПАО «Ростелеком»:
   * Старт.IP (на территориях центрального, волжского, дальневосточного и сибирских регионов).
   * Петерсервис (на территориях северо-западного и уральского регионов).
   * Онима (на территориях южных и московского регионов).
2. Платформа 2.0 должна, при необходимости, обеспечивать взаимодействие с промежуточными АСР-системами, такими как ИС «Лира».
3. Платформа 2.0 должна осуществлять взаимодействие с АСР в зависимости от типа абонента:
   * для абонентов услуги ШПД Ростелкома используется АСР соответствующего региона местоположения пользователя.
   * для других абонентов используется АСР уральского региона — Петер-сервис.
   * [АСР BIS — новый биллинг который где-то используется в РТК]

##### Требования к защите контента

[TBD]

[Описание подхода по работе с токенизированными ссылки в перспективе, доступ предоставляется на основе механизма сессий и UID-устройств]

##### Требования к доставке контента

[TBD: раздельное качество]

## Возможные риски

Разработка решения ИТВ 2.0 требует непрерывной идентификации и управления рисками, включая их качественный и количественный анализ и планирование реагирования на риски разных типов.

Выявленные группы рисков

|  |  |  |  |
| --- | --- | --- | --- |
| Риск | Вероятность | Влияние |  |
| 1. | Правовые риски |  |  |
| 1.1. | Обработка персональных данных | Низкая | Среднее |
| 1.2. | Лицензионные риски | Низкая | Среднее |
| 1.3. | Права на контент | Средняя | Высокое |
| 2. | Операционные риски |  |  |
| 2.1. | Выход из строя оборудования | Низкая | Среднее |
| 2.2. | Рост нагрузки на платформу | Низкая | Низкое |
| 2.3. | Увольнение, временная нетрудоспособность | Высокая | Низкое |
|  | ключевых специалистов |  |  |
| 2.4. | Получение несанкционированного доступа |
|  | к персональной информации | Низкая | Среднее |
| 2.5. | Мошенничество | Средняя | Среднее |
| 2.6. | Неработоспособность смежных элементов системы, | Средняя | Низкое |
|  | реализуемых внешними разработчиками |  |  |
| 3. | Технические риски |  |  |
| 3.1. | Смена технических решений | Низкая | Среднее |
| 3.2. | Некорректные, неактуальные и неполные данные | Высокая | Низкое |
| 4. | Организационные риски |  |  |
| 4.1. | Недостаточное и нестабильное финансирование | Средняя | Высокое |
| 4.2. | Риски изменения содержания проекта | Высокая | Среднее |
| 5. | Политические риски |  |  |
| 5.1. | Введение санкций | Низкая | Среднее |
| 6. | Валютный риск | Высокая | Среднее |
| 7. | Отраслевой риск | Низкая | Высокое |

### Правовые риски

Правовые риски — риски связанные с нарушениями или несоответствием внутренним и внешним правовым нормам.

#### Обработка персональных данных

Несоответствие платформы «Интерактивное телевидение Ростелеком» 2.0 требованиям к обработке персональных данных существующих и планируемых нормативно-правовых актов, регулирующих работу с персональными данными.

*Вероятность:* низкая.

*Влияние:* среднее.

*Реакция на риск:* Изменение логики работы с персональными данными или сертификация платформы «Интерактивное телевидение Ростелеком» 2.0 в соответствии с новыми правилами.

*Комментарии:* Существующее законодательство позволяет избежать сертификации платформы «Интерактивное телевидение Ростелеком» 2.0 в качестве информационной системы персональных данных (ИСПД) путем обработки, хранения и выгрузки обезличенных данных в самой платформе и переноса задач по обработке персональных данных в OSS/BSS-системы ПАО «Ростелеком». Ужесточение правил обработки персональных данных может привести к необходимости сертификации платформы «Интерактивное телевидение Ростелеком» 2.0 в качестве ИСПД.

#### Лицензионные риски

[по лицензиям есть новая вводная. Договор РТК-Смартлабс предполагает передачу неисключительных прав на платформу. Соответственно платформу в исходном виде и ее модификации сложно использовать в качестве продукта для B2B и B2G, а такое пожелание есть. ПИТВ 2.0 в идеале не должна содержать в себе элементы платформы Смартлабс]

В платформе «Интерактивное телевидение Ростелеком» 2.0 предполагается использование объектов интеллектуальной собственности принадлежащих третьим лицам, использование которых требует заключения прямых лицензионных соглашений с правообладателями. Основные риски в данной области:

* изменения правил лицензирования;
* значительное изменение стоимости лицензий на ПО и интеллектуальную собственность;
* отказ владельцев прав на объекты интеллектуальной собственности от продажи лицензий.

*Вероятность:* низкая.

*Влияние:* среднее.

*Реакция на риск:* рассмотрение возможности замены компонентов платформы, распространяемых на условиях лицензий GPL и AGPL аналогичным компонентом под более свободной лицензией (например, LGPL, MIT, Apache и т.д.). Также используемой технологии и техническим решениям следует присвоить статус объекта коммерческой тайны с ограниченным доступом в целях исключить обзор входящих в её состав компонентов для третьих лиц.

*Комментарии:* Используемые технические решения использует ряд программных библиотек, распространяемых на условиях свободных лицензий.

Большинство библиотек, используемых в платформе распространяются на условиях лицензий MIT, LGPL и Appache v2.Также используется ряд библиотек под лицензией GNU Free Documentation License (GPL). Это представляет определенный риск коммерциализации технологии платформа «Интерактивное телевидение Ростелеком».

В общем случае платформа «Интерактивное телевидение Ростелеком» 2.0 предоставляет возможность использования собственными сервисами как услугой (SaaS). В таком формате использования ПО не распространяется каким-либо путем и, как следствие, не нарушаются условия указанных лицензий.

Однако, ситуация усложняется в случае, если ПАО «Ростелеком» (оператор решения) не осуществляет непосредственного владения, контроля и регулирования распоряжением технологией платформы «Интерактивное телевидение Ростелеком» 2.0.

Риски высоки особенно в случае, если оператор решения предполагает программу лицензирования технологии, при которой ПО, содержащее «открытые библиотеки» под лицензией GPL, фактически будет распространяться среди её лицензиаров.

В таком случае, согласно условиям лицензии GPL, ПО должно распространяться бесплатно, а код ПО должен быть опубликован. В то же время существуют более уместные версии лиценцзии GNU GPL.

Из всех лицензий наиболее неуместной представляется GPL 3.0 и сравнительно новая лицензий AGPL, которая исключает возможность использования решения по SaaS-модели.

### Операционные риски

Операционные риски — риски связанные с недостатками или ошибками во внутренних процессах в ходе создания и эксплуатации решения.

#### Выход из строя оборудования

Единовременный выход из строя оборудования, используемого для предоставления услуг пользователям платформы «Интерактивное телевидение Ростелеком» 2.0 может привести к временной недоступности сервиса. Наиболее критичными для работоспособности платформы являются ЦОДы Центрального узла, долговременная неработоспособность которых может вызвать остановку работы платформы в целом.

*Вероятность:* низкая.

*Влияние:* среднее.

*Реакция на риск:* на этапе проектирования требуется разработать временные схемы работы, которые позволят проводить замену вышедшего из строя оборудования без потери работоспособности платформы и ее частей. Должно быть проведено тестирование платформы на отказ и восстановление по результатам которого должны быть разработаны рекомендации по обеспечению бесперебойной работы платформы, регламенты обслуживания и плановой замены оборудования, нормы по наличию резервов оборудования и необходимые инструкции по действию в случае отказа оборудования. Также в ходе эксплуатации платформы необходимо проводить постоянный мониторинг соблюдения норм на наличие резервного оборудования; сотрудники должны пройти обучение и иметь необходимый опыт для решения подобных ситуаций.

#### Рост нагрузки на платформу

В случае расширения географии присутствия платформы (выход на зарубежные рынки, расширение присутствия в регионах и т.д.) или резкого роста количества абонентов возможен быстрый рост нагрузки на платформу с превышением технических возможностей платформ ИТВ 2.0 по уровню нагрузки, что приведет к временной недоступности сервиса.

*Вероятность:* низкая.

*Влияние:* низкое.

*Реакция на риск:* на этапе проектирования должно быть предусмотрена возможность масштабирования платформы с целью повышения показателей нагрузки. По результатам нагрузочного тестирования должен быть проведен мониторинг «узких» мест для разработки планов по развитию платформы.

#### Увольнение, временная нетрудоспособность ключевых специалистов

Увольнение или временная нетрудоспособность ключевых специалистов или единовременное массовое увольнение большого числа сотрудников может привести к потере компетенций, необходимых для функционирования платформы или ее элементов. Данная ситуация может значительно повлиять на надежность и потребительские качества предоставления услуг.

*Вероятность:* высокая.

*Влияние:* низкое.

*Реакция на риск:* для уменьшения негативных последствий подобных ситуаций должны быть разработаны обучающие материалы и инструкции для каждой группы сотрудников по работе с платформой.

На платформу и её составные части должна быть разработана документация, описывающая все ключевые аспекты её работы: детальную информацию о требованиях к платформе в целом, показатели назначения и эксплуатационные характеристики, архитектурные и технические решения, исходный код всех элементов платформы с комментариями, руководства пользователей и системных администраторов, руководства по обслуживанию, эксплуатационные документы. Документация должна поддерживаться в актуальном состоянии.

#### Получение несанкционированного доступа к персональной информации

В случае злонамеренных или ошибочных действий сотрудников, эксплуатирующих платформу, возможно получение третьими лицами доступа к персональной информации абонентов и её неправомерного использования.

*Вероятность:* низкая.

*Влияние:* среднее.

*Реакция на риск:* На этапе проектирования необходимо разработать схему разграничения прав доступа к персональным данным, исключающую нарушение прав доступа, провести тестирование платформы.

#### Мошенничество

Ввиду использования при функционировании платформы смежных элементов, реализуемых внешними по отношению к платформе разработчиками, таких как ЕЛК, OSS/BSS-системы ПАО «Ростелеком», внешние платежные системы, возникает риск потери функциональности или полной неработоспособности платформы в случае отказа данных систем. В случае невозможности взаимодействия платформы с OSS/BSS-системами будет полностью утрачена функциональность платформы в части предоставления услуг абонентам.

*Вероятность:* средняя.

*Влияние:* низкое.

*Реакция на риск:* На этапе проектирования необходимо предусмотреть аварийные режимы работы платформы, позволяющие сохранить работоспособность даже с частичной потерей функциональности.

#### Права на контент

Повышение стоимости контента, изменение модели распространения или отказ от продажи контента правообладателем может привести к повышению стоимости и отказу от использования контента.

*Вероятность:* средняя.

*Влияние:* высокое.

*Реакция на риск:* В платформу должна быть заложена функциональность отслеживания статуса и условий лицензионных соглашений с правообладателями с автоматизированным уведомлением администраторов о необходимых действиях. Платформа должна позволять построение бизнес-процессов управления контентом с учетом изменений в соглашениях с правообладателем, в том числе оперативного удаления контента с клиентских продуктов.

### Технические риски

Технические риски — риски связанные с ошибками проектирования, неполнотой наследуемых данных, несоответствием требованиям выбранных решений.

#### Смена технических решений

Вследствие различных факторов (санкции, изменение стоимости выбранных решений, несоответствие требованиям, изменения в смежных системах и пр.) в ходе создания и эксплуатации платформы может потребоваться смена технических решений. Изменения, вносимые на поздних этапах создания и на этапе эксплуатации, могут привести к значительным финансовым потерям и срывам сроков разработки.

*Вероятность:* низкая.

*Влияние:* среднее.

*Реакция на риск:* Проектируемая платформа должна основываться на принципах модульности; программные интерфейсы взаимодействия компонентов платформы должны быть задокументированы и поддерживаться в актуальном состоянии.

#### Некорректные, неактуальные и неполные данные

В ходе анализа текущего решения ИТВ были выявлены значительное количество данных, требующих переноса в решение ИТВ 2.0, которые были введенны некорректно (включая орфографические, грамматические, фактические ошибки и прочие), неактуальны и содержат неполную информацию. Также были выявлены значительные недостатки в модели данных, которые могут повлечь за собой необходимость корректировки данных в ручном режиме.

*Вероятность:* высокая.

*Влияние:* низкое.

*Реакция на риск:* разработка автоматизированных ETL-процедур (извлечение, преобразование и загрузка) для упрощения процессов переноса данных. Панель управления платформой должна осуществлять проверку входных данных на предмет неверных форматов и недопустимых значений. Ввод данных должен осуществляться на основе актуальных справочников.

### Организационные риски

Организационные риски — риски, связанные с организацией работы компании.

#### Недостаточное и нестабильное финансирование

Недостаточное и нестабильное финансирование может привести к срыву сроков создания платформы «Интерактивное телевидение Ростелеком» 2.0, превышению бюджета и снижению качества конечного продукта.

*Вероятность:* средняя.

*Влияние:* высокое.

*Реакция на риск:* Согласование бюджета проекта и графика платежей со всеми заинтересованными сторонами на ранних стадиях реализации. Своевременное заключение договоров для обеспечения соблюдения сроков разработки.

#### Риски изменения содержания проекта

Внесение изменений в содержание проекта в ходе создания платформы «Интерактивное телевидение Ростелеком» 2.0 может привести к увеличению сроков разработки и изменению бюджета проекта.

*Вероятность:* высокая.

*Влияние:* среднее.

*Реакция на риск:* В ходе создания платформы «Интерактивное телевидение Ростелеком» 2.0 необходимо проводить работы по управлению проектными требованиями и содержанием проекта. Все изменения должны документироваться и проходить процедуру согласования.

### Политические риски

Политические риски — это риски, являющиеся следствием политики государств.

#### Введение санкций

Введение санкций и как следствие отказ от продажи и поддержки лицензионного ПО, оборудования и контента зарубежными производителями, отказ от предоставления услуг зарубежными компаниями.

*Вероятность:* низкая.

*Влияние:* среднее.

*Реакция на риск:* для лицензионного ПО компании Oracle требуется разработка логической и физической моделей данных сервиса с учётом современных требований к проектированию баз данных, обеспечивающих процедуры миграции на решения сравнимого класса (например: PostgreSQL).

Для лицензионного ПО Verimatrix и Adobe Flash Access переход на аналогичное ПО других производителей не представляется возможным. В случае невозможности использовать данные системы защиты контента от несанкционированного использования возможно использование программных или аппаратных токенов и токенизированных ссылок на основании этих токенов и сессий подключения. Данное решение может потребовать дополнительного согласования с правообладателями контента.

Для лицензируемых стандартов видеокодирования (AVC и прочие) возможен переход на аналогичные лицензируемые или открытые стандарты.

Для платформ App Store, Google Play, LG Smart World, Samsung Apps ввиду привязки устройств к платформам реакция на риск не предусматривается.

Для оборудования, используемого платформой ИТВ 2.0, ввиду наличия большого количества совместимого оборудования зарубежных производителей в т.ч. в странах введение санкций которыми представляется маловероятным реакция на риск не предусматривается.

### Валютный риск

Связан со значительным изменением стоимости предполагаемого к закупке оборудования и лицензионного ПО вследствие неблагоприятного изменения курсов иностранных валют.

*Вероятность:* высокая.

*Влияние:* среднее.

*Реакция на риск:* Проведение закупок на возможно более ранних этапах создания платформы «Интерактивное телевидение Ростелеком» 2.0, заключение долговременных соглашений с фиксированной стоимостью. с правообладателями интеллектуальной собственности.

### Отраслевой риск

Неудовлетворительное качество обслуживание пользователей ПИТВ в сетях других интернет-провайдеров из-за политик управления межсетевым трафиком.

*Вероятность:* низкая.

*Влияние:* высокое.

*Реакция на риск:* Разработка и внедрение процесса мониторинга качества оказания услуг на основании отчётов пользователей. Формирование архитектуры сети доставки контента с учётом доступности сегментов сетей сторонних операторов.

# Техническое описание Платформы интерактивного телевидения 2.0

## Концепция архитектуры платформы

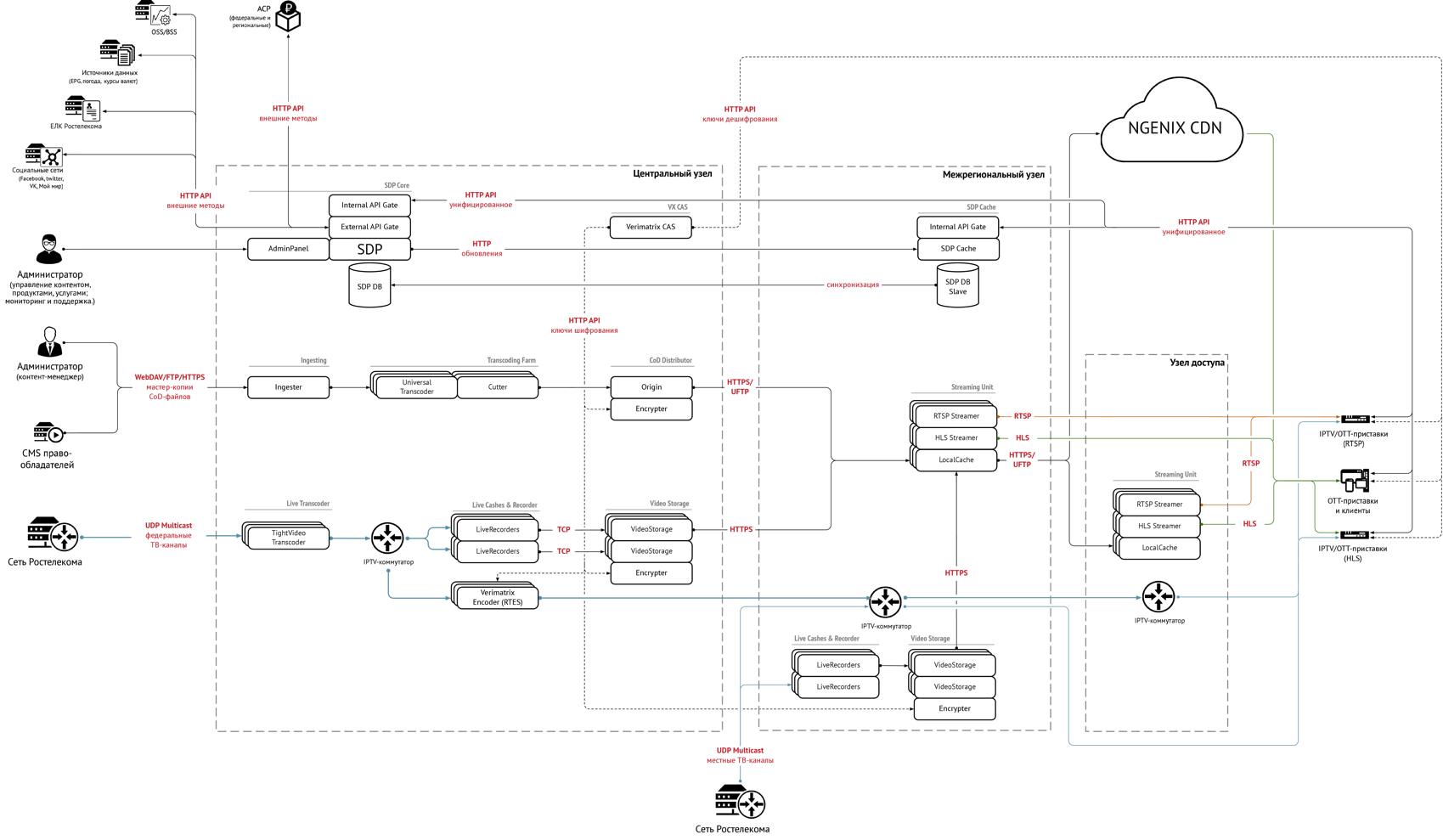
[*Общая архитектурная схема, dataflow, описание основных компонентов, отличия от текущей архитектуры*]

Проектируемая архитектура платформы «Интерактивное телевидение Ростелеком» 2.0 является развитием существующей архитектуры платформы «Интерактивное телевидение Ростелеком». Используя существующую инфраструктуру, платформа «Интерактивное телевидение Ростелеком» 2.0 призвана:

* обеспечить высокую управляемость и модифицируемость решения;
* улучшить качество предоставляемых пользователям услуг;
* повысить эффективность работы решения;
* устранить недостатки существующего решения.

Архитектура платформы «Интерактивное телевидение Ростелеком» 2.0 строится на наборе компонентов, собранных в две группы:

* Система управления платформой — отвечает за управление продуктами, контентом, оплатой, абонентами, системой рекомендаций; сбор статистики и аналитики; мониторинг и взаимодействие с системами технической поддержки; взаимодействие с внешними сервисами.
* Система доставки контента — отвечает за управление контентом, доставку контента пользователям.

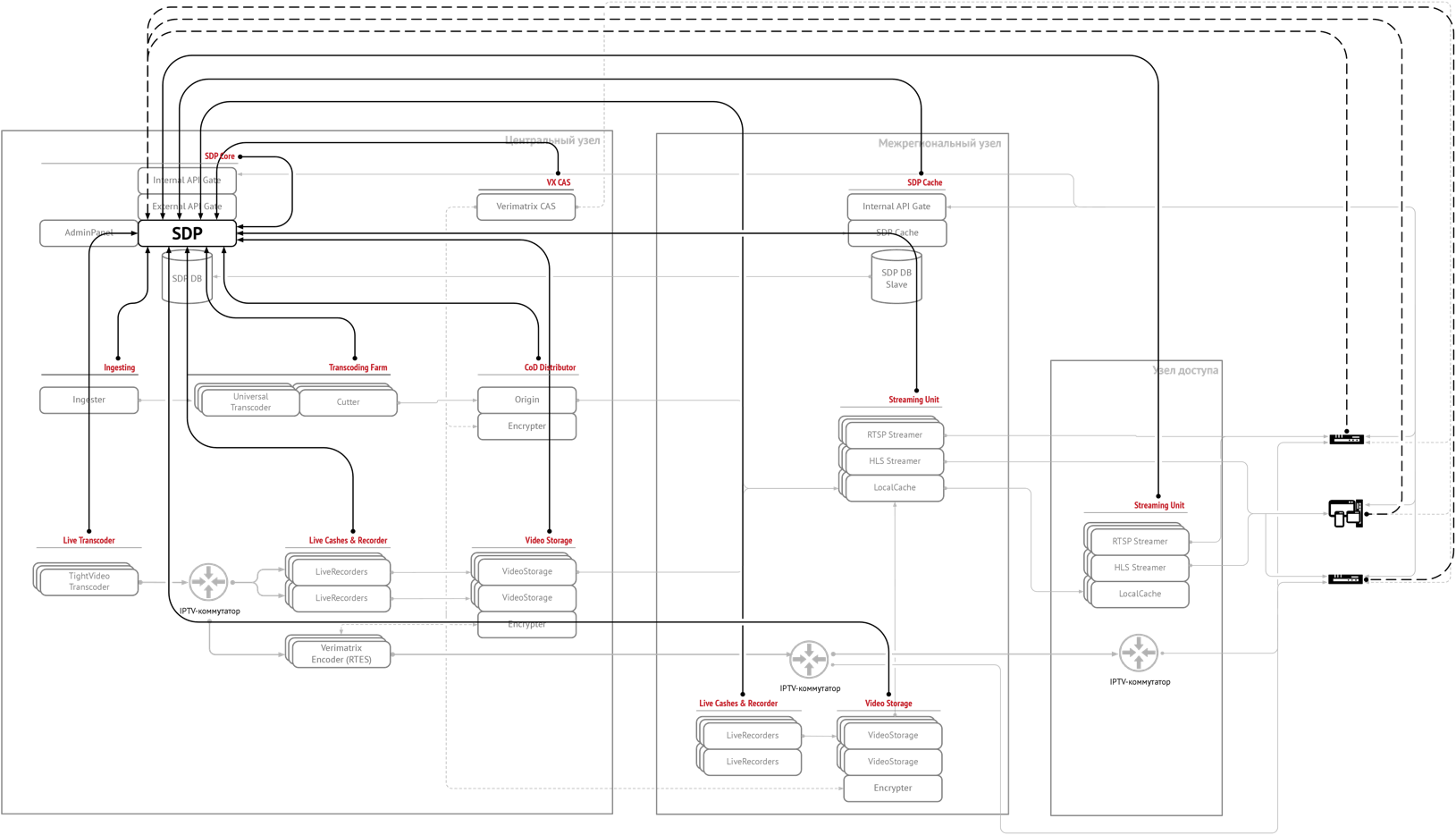


*Логическая архитектура проектируемой платформы «Интерактивное телевидение Ростелеком» 2.0*

Система управления платформой хранит данные, обрабатывает запросы клиентских продуктов и внешних систем, предоставляет администраторам инструменты управления решением (Панель администрирования). Система управления платформой выступает связующим программным обеспечением (middleware), реализующим взаимодействие между различными приложениями, системами и компонентами, входящими в состав платформы, предоставляющим доступ к данным.

Система доставки контента осуществляет доставку контента (CoD, ТВ-эфира) пользователям напрямую посредством протоколов HLS и RTSP (для прошлого поколения абонентских устройств) и через CDN.

Системы управления платформой и доставки контента реализовывают комплексное управление цифровыми активами (Digital asset management) — совокупность мероприятий по внесению, аннотированию, каталогизации, хранению, извлечению и распространению цифрового контента.

*Принципиальная схема мониторинга компонентов и пользовательских продуктов проектируемой платформы «Интерактивное телевидение Ростелеком» 2.0*

Система управления платформой осуществляет постоянный мониторинг состояния и работы программных компонентов и аппаратного обеспечения, входящих в состав решения, качества предоставляемых услуг и действий пользователей. Это позволяет осуществлять менеджмент уровня услуг (service-level management) в режиме реального времени: отслеживать качество предоставляемых услуг (QoS) и ключевые показатели производительности решения (KPI).

Данные мониторинга и статистики используются для контроля и управления работы системы, вывода отчётности, работы рекомендационных систем, настройки контентных политик.

## Описание API пользовательских устройств

*[TBD: раздел в написании]*

*Основные группы ресурсов и методов; базовый референс, без детального описания*

### Цель создания

Основной целью создания API ИТВ является повышение качества услуг сервиса Интерактивного телевидения Ростелекома для абонентов различных МРФ ПАО Ростелеком и пользователей сети интернет.

В качестве дополнительных целей выделяются:

Упрощение процедуры управления сервисом.

Сокращение затрат на масштабирование сервиса и запуск новых продуктов.

Расширение функциональности клиентских продуктов Повышение стабильности работы сервиса

API обеспечивает взаимодействие с различными информационными подсистемами сервиса ИТВ. Такими как:

#### Описание

API обеспечивает взаимодействие с различными информационными подсистемами сервиса ИТВ. Такими как:

* Система хранения данных ИТВ
* Сеть доставки контента CDN и внутренними видео системами
* Системой управления контентом
* Магазинами приложений (App Store, Google Play)
* Системами контроля оплаты

API обеспечивает реализацию следующих сценариев взаимодействия с клиентскими устройствами:

* Управляет личной записью пользователя сервиса
* Регистрирует клиентские устройства в системе
* Получает информацию о контенте и промо материалах
* Получает информацию о совершенных покупках пользователя
* Обеспечивает покупку периодические услуги и производит разовые покупки контента
* Управляет профилями клиентских приложений
* Синхронизирует информацию между различными внешними подсистемами (перенос и синхронизация покупок)

Формат выдачи методов должен удовлетворять широкому спектру устройств, в качестве базовых формат обмена данными используются стандарты XML, NCDXML, CSV и JSON с возможностью расширения форматов выдачи.

Логика работы API реализуется следующих базовых состояний авторизации пользователей:

* Обезличенная авторизация
* Абонент интерактивного ТВ (пользователь услуги ШПД и ИТВ)
* Пользователь сервиса Zabava.ru
* Пользователь интернета (социальные сети, email авторизация)

Логика API обеспечивает регистрацию платного доступа к контенту в формате обезличенной авторизации, с возможностью переноса покупок после прохождения процедуры регистрации и авторизации. Для каждого метода предусматриваются обязательные входные и опциональные параметры. В качестве неотъемлемой части системы API необходимо реализовать механизм кэширования высокочастотных запросов.

В качестве дополнительной функциональность API обеспечивает взаимодействие платформы с партнерскими площадками и сервисами. Для реализации подобного взаимодействия в логику решения API необходимо включить механизм выгрузки данных о доступном контенте в формате единого обновляемого архива XML, JSON.

## Описание информационной структуры

*Основные сущности; общая ERD-схема БД; описание таблиц; связь с бизнес- и функциональными требованиями*

# Приложение. Справочник API клиентских продуктов платформы «Интерактивное телевидение Ростелеком»

## Порядок работы c API клиентских продуктов

С помощью методов API клиентский продукт может выполнить следующие действия в SmartTube SDP:

* Авторизовать клиентские устройства в системе.
* Получать информацию о контенте.
* Получать информацию о кросс-покупках.
* Подписывать периодические услуги и осуществлять разовые покупки.
* Управлять профилями клиентских приложений.
* Создать нового абонента в АСР.
* Передать информацию о покупках с помощью Apple ID в АСР.

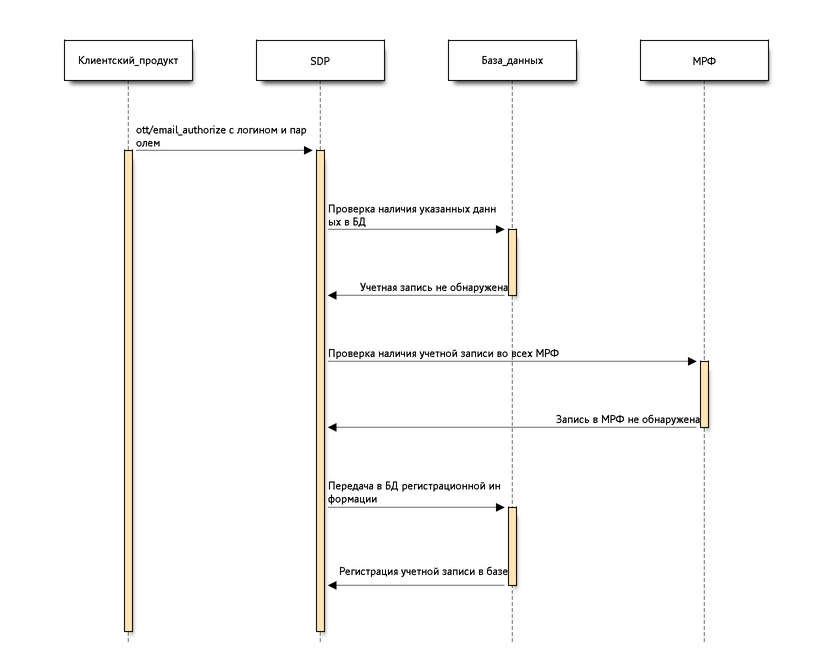
Клиентские продукты используют следующий порядок работы с серверами SmartTube SDP:

* Клиентский продукт устанавливает соединение с SDP-сервером определенного макрорегиона:
  + для пользователей, созданных через ЕКП Zabava и не имеющих подключения к услугам ШПД или IPTV Ростелекома — в SDP-сервере МРФ «Центр»;
  + для абонентов, подключенных к услугам ШПД или IPTV Ростелеком — в SDP-сервере соответствующего МРФ; МРФ определяется по первым двум цифрам лицевого счета абонента.
* Клиентский продукт вызывает методы /list для загрузки справочников, использующихся в описании контента — уровни доступа, списки жанров и групп CoD, тематики ТВ-каналов и т.д.
* Клиентский продукт авторизует абонента в SDP макрорегиона, в котором он зарегистрирован, вызовом метода ServiceAccount/web\_authorize или ott/email\_authorize. Метод возвращает идентификатор тарифного плана, который используется далее для получения списка услуг.
* Клиентский продукт получает полный список услуг в тарифном плане вызовом метода ProductOffer/list\_services\_terminal.
* Клиентский продукт получает список опциональных услуг, доступных абоненту, вызовом метода ServiceAccount/list\_services.
* Все подключенные на учетную запись пользователя услуги — это объединение услуг из ответа метода ProductOffer/list\_services\_terminal, имеющих признак <mandatory>true</mandatory> (все обязательные, подписанные по умолчанию), и услуг из ServiceAccount/list\_services (все опциональные подписанные дополнительно с serviceState=1).
* Используя полученные идентификаторы услуг, клиентский продукт вызывает методы ...Package/list\_... для загрузки соответствующего контента (списков ТВ-каналов, фильмов и т.д.).
* Клиентский продукт получает матрицу цен для каждой разовой услуги вызовом метода /list\_rules... .
* Для подписки на периодическую услугу (в том числе абонементы) клиентский продукт вызывает метод .../web\_subscribe.
* Для разовой покупки контента клиентский продукт вызывает метод /web\_purchase. Стоимость разовой покупки может быть различной в зависимости от правил использования (согласно требованиям к тарификации). Матрица стоимостей разовой покупки из определенного CoD-пакета возвращается в методе /list\_rules... . При этом параметр price CoD-пакетов игнорируется. Выбранная при покупке стоимость будет однократно списана со счета абонента в АСР.
* Для покупок по абонементу клиентский продукт вызывает методы .../web\_purchase\_by\_abonement.
* Для получения списка сохраненных позиций просмотра клиентский продукт периодически вызывает метод Content/mediaPositionList. Для добавления новой или обновления существующей позиции использует метод Content/addMediaPosition.
* Для обновления данных о контенте и состава услуг клиентский продукт вызывает методы /get\_delta.
* Для загрузки программы передач для ТВ-каналов клиентский продукт вызывает методы EPG/get\_by\_Chnnl, EPG/get\_by\_pkg.
* Для работы с профилями клиентских приложений клиентский продукт использует методы ServiceAccount/list\_avalable\_profiles, Profile/update\_new, Profile/insert\_new.

### Авторизация пользователя

Для авторизации используются методы ServiceAccount/web\_authorize и ott/email\_authorize.

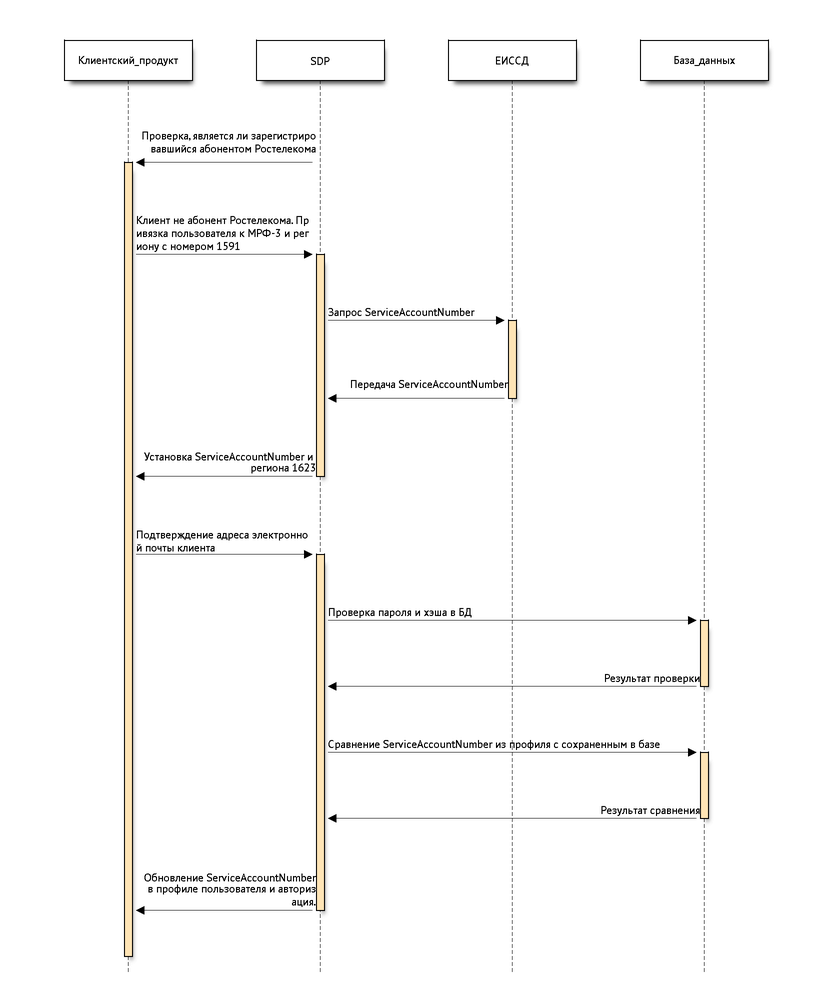
* Пользователь вводит логин и пароль.
* Проверяется наличие авторизируемой учётной записи в базе данных сайта.
* Если учётная запись не обнаружена, то:
  + проводится валидация длины пароля (она должны быть не меньше 6 символов);
  + проверяется наличие учётной записи во всех МРФ. Если учётная запись найдена в каком-либо МРФ, выдаётся ошибка «E-mail занят»;
  + если учетная запись не найдена, то она регистрируется в базе.
* Проводится проверка, является ли пользователь абонентом Ростелекома.



Порядок получения адресов вещания ТВ-каналов

**Если пользователь не абонент Ростелекома**

* Пользовательский профиль привязывается к МРФ-3 и региону 1591. Для пользователя регистрируется учетная запись в Единой интернет системе сервиса дилера (ЕИССД).
* Из ЕИССД передается параметр ServiceAccountNumber.
* В пользовательские данные добавляется полученный ServiceAccountNumber и устанавливается город 1623.



Порядок получения адресов вещания ТВ-каналов

Если учетная запись существует в базе сайта, то:

* Валидируется подтверждение почты пользователем.
* Авторизируемый пароль и хэш в проверяются в базе данных.
* Параметр ServiceAccountNumber пользователя проверяется в базе данных и в МРФ. Если значения не совпадают, то ServiceAccountNumber обновляется. Если у пользователя не задан ServiceAccountNumber, но он есть в МРФ, то передается ошибка «Email занят».
* Пользователь авторизуется.

**Если пользователь абонент Ростелекома:**

SDP выполняет проверку, является ли пользователь абонентом Ростелекома, предлагая выбрать один из трех вариантов:

1. Интерактивное ТВ (интернет Ростелекома).
2. Интерактивное ТВ 2.0 (интернет другого провайдера).
3. Домашний интернет.

Вариант “Интерактивное ТВ”:

* Пользователю предлагается выбрать регион.
* После выбора региона предлагается ввести номер и код активации, указанные на обратной стороне ТВ-приставки или в договоре.

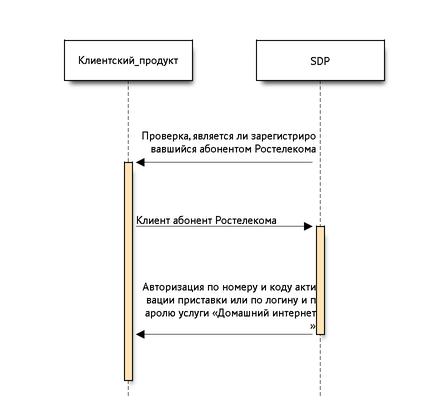
Вариант “Интерактивное ТВ 2.0”:

Выводится сообщение: “Являетесь ли вы абонентом услуги Домашний интернет от компании Ростелеком?”

* Если пользователь — абонент Ростелекома, то пользователю предлагается выбор региона, ввод логина и пароля услуги “Домашний интернет”.
* Если пользователь — не абонент Ростелекома, то пользователю предлагается ввести номер и код активации с приставки.

Вариант “Домашний интернет”:

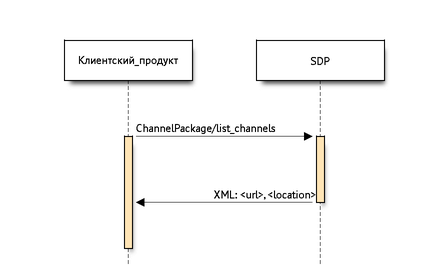
* Пользователю предлагается выбрать регион.
* После выбора региона пользователь должен ввести логин и пароль услуги “Домашний Интернет”.



Порядок получения адресов вещания ТВ-каналов

### Получение источников вещания ТВ-каналов

Получить описания ТВ-каналов можно вызовом ChannelPackage/list\_channels. Источник вещания ТВ-канала указывается в параметре <urls><location> для соответствующего региона.



Порядок получения адресов вещания ТВ-каналов

Пример:

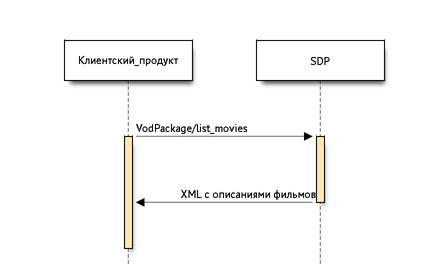
<urls><!--переопределенные по регионам URL источников вещания ТВ-канала-->  
 <location id="1000001"><!--регион Bradbury-->  
 <url>http-streaming/1tv.m3u8</url>  
 </location>  
 <location id="1000002"><!--регион Портал-->  
 <url>http-streaming/1tv.m3u8</url>  
 </location>  
</urls>

Если значения в <urls> для нужного региона отсутствуют, следует использовать источник вещания по умолчанию, указанный в параметре <url>:

<url>igmp://239.195.8.4:30120</url><!--адрес вещания ТВ-канала по умолчанию -->  
<pipUrl>igmp://229.255.0.100:5600</pipUrl><!--адрес вещания ТВ-канала для PIP по умолчанию -->

### Получение описаний фильмов

Описания фильмов запрашиваются вызовом VodPackage/list\_movies. Имя файла с фильмом на видеосервере указывается в параметре <assets><content\_asset><ifn>:



Порядок получения описаний фильмов

Пример:

<assets><!-- физические источники-->  
 <content\_asset><!--описание физического источника-->  
 <id>1029753</id><!--идентификатор источника-->  
 <ifn>http-streaming/GreenZone.m3u8</ifn><!--имя файла на видеосервере или URL-->  
 <duration>300</duration><!--длительность в секундах-->  
 <asset\_type>378264</asset\_type><!--идентификатор типа источника (типа ассета), используется в правилах использования VodPackage/list\_rules-->  
 <is\_crypted>0</is\_crypted><!--флаг шифрования. 0 — не шифрован-->  
 <transport>http</transport><!--протокол вещания-->  
 <loc><!--идентификатор регионов, в которых доступен ассет-->  
 <id>1013628</id>  
 <id>1021478</id>  
 </loc>  
 </content\_asset>  
</assets>

Дополнительно в секции <assets><preview\_asset> может быть указан трейлер фильма.

### Реализация логики работы с правами доступа

У любого типа контента (фильм, канал, серия) есть ссылка на значение из справочника уровней доступа. По этому полю из справочника уровней доступа можно получить значение (например, фильм с пометкой “18+” или “3+”).

В профиле пользователя есть две ссылки на значение из справочника.

* lastAccessLevelId — текущий уровень доступа к контенту. Возможен просмотр контента, требующего более низкого уровня доступа. Для просмотра контента, требующего более высокого уровня доступа, необходимо ввести pin-код. Также это последний максимальный уровень контента, к которому пользователь получал доступ по вводу pin-кода, если была активна галочка “Запомнить уровень доступа” (значение isAccessLevelPersistent).
* maxAllow — максимальный уровень доступа к контенту. Контент, требующий уровень доступа выше maxAllow недоступен для просмотра.

Чтобы изменить максимальный уровень доступа профиля, необходимо выполнить запрос Profile/update\_new, указав идентификатор нужного уровня по справочнику SDP:

http://<host>:<port>/smarttube/master/spine/services/xml/Profile/update\_new?ID=15472491&maxAccessLevel=127100&mac-address=1751482113&sessionId=104042

Чтобы изменить текущий уровень доступа профиля, необходимо выполнить запрос Profile/update\_new, указав идентификатор нужного уровня по справочнику SDP:

http://<host>:<port>/smarttube/master/spine/services/xml/Profile/update\_new?ID=15472491&lastAccessLevel=127100&mac-address=1751482113&sessionId=104042

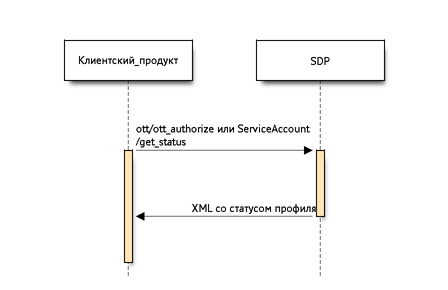
### Обработка состояний учётной записи

Поддерживаются следующие виды статусов:

* ACTIVE;
* BLOCKED;
* SELFBLOCKED;
* FULLBLOCKED;
* CLOSED;
* NOTACTIVE;
* DELETED.

Текущее значение статуса учётной записи можно получить следующими способами:

1. Из уведомления “Изменение статуса учётной записи”.
2. В поле <state> ответа на запрос ott/ott\_authorize (например: <state>ACTIVE</state>).
3. Методом ServiceAccount/get\_status.



Порядок удаления устройств из группы

Далее перечислены доступные действия с приложением в различный статусах:

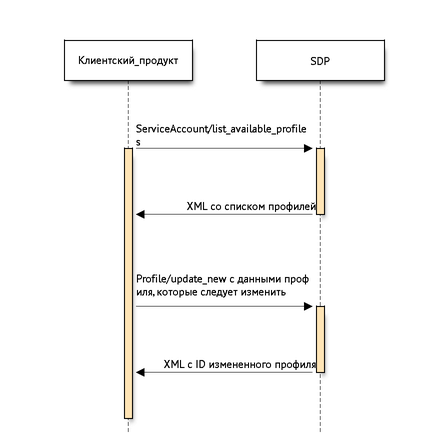
* ACTIVE (аккаунт активен) — в приложении доступны все функции.
* BLOCKED/SELFBLOCKED (аккаунт частично заблокирован):
  + доступен просмотр купленных фильмов;
  + возможен просмотр каналов из пакетов, доступных заблокированным учетным записям (с параметром <isAccessByLock>true</isAccessByLock>);
  + покупка нового контента запрещена, выводится сообщение с отображением текста причины блокировки;
  + просмотр каналов недоступных в блокировке запрещен, выводится сообщение с отображением текста причины блокировки.
* FULLBLOCKED (аккаунт заблокирован) — запрещен просмотр любого контента. Приложение запускается и выводится окно с сообщением о причине блокировки. При попытке просмотра любого контента выводится сообщение “Аккаунт полностью заблокирован”.
* CLOSED — при запуске выводится сообщение “Договор на оказание услуги расторгнут” и приложение не запускается.
* NOTACTIVE/DELETED и прочих неизвестных — весь контент недоступен. При запуске показывается сообщение «Ошибка загрузки приложения, обратитесь в техническую поддержку».

### Управление профилем

Для получения списка профилей необходимо выполнить запрос ServiceAccount/list\_available\_profiles, указав serviceAccountNumber и mac-address.

Из приложения можно работать только с профилем master. Имя профиля может отличаться, но в информации о профиле приходит параметр <isMaster>1</isMaster>. Переключаться между профилями, изменять имя профиля и создавать новые профили из приложения нельзя.

Для изменения профиля используется метод Profile/update\_new.



Порядок изменения данных профиля

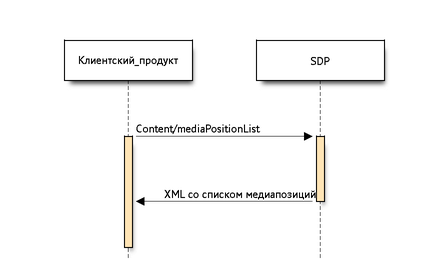
Используются следующие поля профиля:

* ID — идентификатор — обязательное поле, передается во всех запросах Profile/update\_new.
* name — имя профиля, изменение этого поля запрещено.
* reminders — список напоминаний о передачах. Напоминания можно добавлять и удалять. Для добавления нового напоминания необходимо к уже созданным напоминаниям добавить новое и отправить на сервер все записи. Если отправлять информацию только о добавляемом новом, то остальные напоминания будут удалены на сервере. Для удаления передачи на сервере необходимо отправить весь список напоминаний без удаляемого напоминания. Напоминания состоят из трех полей:
  + program id — идентификатор передачи.
  + cid — идентификатор ТВ-канала.
  + sdate — дата и время начала передачи (“08/04/2011 14:44:31”).
* maxAllow — максимальный уровень доступа профиля.
* lastAccessLevelId — текущий уровень доступа.
* purchaseVodAllow — Разрешение покупок видеофильмов. Значение можно изменять, пользователь при этом должен ввести PIN-код. Параметр принимает три значения:
  + 2 — покупки разрешены.
  + 1 — покупки разрешены после ввода PIN.
  + 0 — покупки запрещены.
* pin — PIN-код профиля, состоит из четырех цифр. Пользователь может изменить PIN-код, для этого необходимо сначала ввести старый код, а затем новый.
* isAccessLevelPersistent — флаг сохранения уровень доступа (для которого ввели пин). Определяет возможность доступа к контенту, который выше уровня lastAccessLevelId и одновременно ниже или равен maxAllow. Если функция выключена, то при открытии такого контента и вводе верного PIN-кода пользователю предоставляется однократный доступ к контенту (уровень контента выше, чем текущий, но ниже максимального). Текущий уровень доступа при этом не меняется. Если функция включена, то при открытии такого контента, пользователю будет предложено ввести пин-код один раз и последующий доступ к контенту такого же или более низкого уровня предоставляется без ввода пин-кода. Текущий уровень доступа при этом повышается до уровня доступа контента.

### Медиапозиции

Медиапозиции — это метки для восстановления проигрывания каналов и фильмов с последней проигрываемой позиции.

Для получения списка медиапозиций необходимо выполнить запрос Content/mediaPositionList.



Порядок получения медиапозиций

Пример:

http://fe-ural.svc.iptv.rt.ru/mrf-ural-1000181-1000181/spine/services/ncdxml/Content/mediaPositionList?serviceAccountNumber=77099077454&sessionId=&mac-address=8E73A66B7CF58C95&deviceType=ANDROID&profileName=MASTER

Параметры:

* serviceAccountNumber — <serviceAccountNumber или <secondaryServiceAccountNumber> из метода ott\_authorize.
* mac-address — поле resourceId из метода ott\_authorize.
* deviceType — тип устройства, ANDROID или IPAD.
* profileName — имя текущего профиля.

Для получения полного списка медиапозиций необходимо выполнить запрос Content/mediaPositionList с параметром ServiceAccountNumber, а затем с передачей в параметре ServiceAccountNumber значения <secondaryServiceAccountNumber> если оно было возвращено методом ott\_authorize.

Ответы объединяются, если в общем списке на один и тот же фильм или канал есть две медиапозиции, выбирается та, которая была сделана позже.

Медиапозиция содержит следующие поля:

* key — ID контента.
* type — тип позиции (VIDEOMOVIE/CHANNEL).
* value — значение позиции в секундах. Для фильма — смещение относительно начала, для канала — время сохранения позиции просмотра канала в формате unixtime.
* mac — ID ресурса (значение resourceId из метода ott\_authorize).
* action — тип действия (PAUSE/STOP/PLAY).
* profileName — имя профиля.
* dt — дата-время когда сделана позиция в строковом формате, MM/DD/YYYY HH:MM:SS.

Для добавления медиапозиций используется метод Content/addMediaPosition.

* serviceAccountNumber — параметр <serviceAccountNumber> или <secondaryServiceAccountNumber> из метода ott\_authorize.
* mac-address — поле resourceId из метода ott\_authorize.
* deviceType — тип устройства.
* type — тип позиции.
* value — значение позиции в секундах.
* action — тип действия.
* profileName — имя текущего профиля.

В случае, если метод ott\_authorize вернул непустой параметр secondaryServiceAccountNumber, с его значением также необходимо выполнить запрос, передав его в поле параметра serviceAccountNumber.

Медиапозиции также делятся на 2 типа по месту сохранения:

1. Серверные — позиции, которые отправляются запросом Content/addMediaPosition.
2. Локальные — позиции, хранящиеся только локально (предназначаются только для одного пользователя). Такие позиции сохраняются периодически. Период меньше 5 минут.

При наличии двух типов медиапозиций на канал (серверной и локальной), необходимо выбирать последнюю добавленную по времени. Серверная команда PLAY для канала не учитывается, если она была отправлена с другого устройства, вместо неё необходимо брать последнюю локальную медиапозицию.

Для удаления медиапозиции используется метод Content/deleteMediaPosition.

Особенности отправки позиций просмотра с клиентских устройств:

* При включении потока ТВ-канала с любой позиции кроме прямого эфира, на сервер отправляется событие типа PLAY.
* При воспроизведении контента следует периодически сохранять в памяти устройства позицию типа PLAY для данного контента. Отправлять позиции PLAY, сохраненные таким способом, на сервер не нужно.
* При постановке потока канала на паузу всегда следует передавать событие типа PAUSE на сервер.
* В случае перехода с сохраненной позиции на прямой эфир необходимо отослать на сервер тип действия STOP для данного канала.

Интерпретация позиций на клиентских устройствах:

* Если при включении ТВ-канала устройство видит сохраненную позицию PAUSE, то предлагает начать воспроизведение с сохраненной позиции.
* Если при включении ТВ-канала устройство видит сохраненную позицию PLAY (из памяти устройства), то предлагает начать воспроизведение с сохраненной позиции.

При переходе в прямой эфир на сервер следует передавать тип действия STOP для канала, чтобы стереть все другие типы действий.

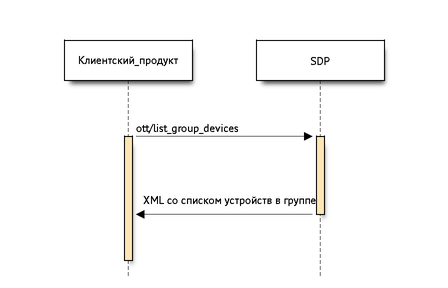
Работать локально с паузой нельзя, при снятии с паузы нужно отправить серверу тип действия PLAY. Это необходимо для того, чтобы убрать с сервера тип действия PAUSE.

Если позиция с переданным значением key уже существует, будет обновлено значение value позиции. Если позиции с таким key не существует (новая), она будет добавлена. Максимальное число позиций — не более 30 (настраивается в конфигурации SmartTUBE SDP). При добавлении большего числа позиций самые старые удаляются по принципу FIFO.

### Работа с устройствами в группе

Устройства, входящие в группу активной учетной записи отображаются в пункте “Мультискрин” раздела “Моё”.

Для получения списка устройств следует выполнить запрос ott/list\_group\_devices.



Порядок получения списка устройств в группе

Пример:

http://fe-ct.svc.iptv.rt.ru/mrf-ct-100001-100001/spine/services/ncdxml/ott/list\_group\_devices?serviceAccountNumber=9154720563

В параметрах передаётся только serviceAccountNumber — значение из поля serviceAccountNumber или secondaryServiceAccountNumber, полученное методом ott/ott\_authorize.

Для получения полного списка устройств необходимо выполнить запрос ott/list\_group\_devices с параметром serviceAccountNumber. Затем, если secondaryServiceAccountNumber не пустое, повторить запрос ott/list\_group\_devices с параметром secondaryServiceAccountNumber.

Полученные списки устройств объединяются. Повторяющиеся записи из итогового списка удаляются.

В каждой записи об устройстве содержатся следующие поля:

* ID — id устройства.
* UID — id пользователя.
* deviceType — тип устройства.
* terminalName — название устройства.
* isTrustedDevice — не используется.

В интерфейсе клиентского продукта формируется список устройств. У каждого устройства из списка отображается иконка типа устройства и название из поля <terminalName>. О каждом устройстве можно просмотреть подробную информацию: uid, номер и код активации. Номер и код активации получаются методом ott\_authorize, из полей <activationNumber> и <pin>.

При подключении к аккаунту ЕКП Zabava или аккаунту Интерактивного ТВ, пользователь попадает в группу этих аккаунтов.

При использовании гостевого доступа пользователь также находится в собственной группе. При этом другие устройства могут подключаться к его группе c логином “Интерактивное ТВ”. Для этого используются номер и код активации устройства, к которому следует подключиться.

Поле <multiroom> из метода ott\_authorize отвечает за нахождение в группе более чем одного устройства. Поле <isHomeGroup> из метода ott\_authorize определяет, служит ли устройство автором группы. Возможные варианты нахождения в группах описаны ниже.

Устройство находится в своей группе с другими устройствами:

<multiroom>true</multiroom>  
<isHomeGroup>1</isHomeGroup>

Устройство находится в чужой группе с другими устройствами:

<multiroom>true</multiroom>  
<isHomeGroup>0</isHomeGroup>

Устройство находится в своей группе одно:

<multiroom>false</multiroom>  
<isHomeGroup>1</isHomeGroup>

Устройство находится в чужой группе одно:

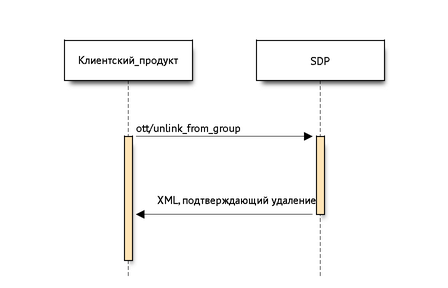
<multiroom>false</multiroom>  
<isHomeGroup>0</isHomeGroup>

Если пользователь не авторизован и находится в своей группе, в интерфейсе кнопка “Выйти из группы” не отображается.

Если пользователь авторизован и находится в чужой группе, на устройстве отображается кнопка “Выйти из группы”.

Для других устройств всегда отображается кнопка “Удалить из группы”, кроме устройств с типом STB.

Для удаления используется запрос ott/unlink\_from\_group.



Порядок удаления устройств из группы

Пример:

http://fe-ct.svc.iptv.rt.ru:80/mrf-ct-100001-100001/spine/services/ncdxml/ott/unlink\_from\_group?ID=69968942&serviceAccountNumber=9182783132

Параметры:

* ID — идентификатор удаляемого устройства;
* serviceAccountNumber — значение из ott\_authorize.

Если устройство вошло в группу другого устройства (STB или не STB) и авторизация произошла через Интерактивное ТВ, то имя аккаунта для отображения необходимо получать из поля <serviceAccountNumber> в запросе ott\_authorize.

### Нотификации сервера

В основе механизма нотификаций лежит технология “long polling”.

Порядок действий для получения нотификаций:

1. Из клиентского приложения выполняется запрос на адрес следующего формата: frontendUrl + "/subscribe/" + uid.toUpperCase() + "/" + locationId + "/" + productOfferId + "/all". Если пользователь авторизован в ЕКП Zabava или Интерактивном ТВ, значит в конец запроса требуется дописать параметр :login

Пример:

http://fe-ct.svc.iptv.rt.ru:80/mrf-ct-100001-100001/subscribe/BCFB762942EC52EF/100902/1716852/all/133113

2. Ожидается ответ от сервера. 3. Если сервер ничего отвечает в течении некоторого времени — произойдёт событие SocketTimeoutException, после чего запрос повторяется на тот же адрес. 4. В случае появления информации для пользователя сервер возвращает данные. В ответе содержится строка приблизительно следующего вида:

"command?commandType=Control&commandName=profileSync"

По полю commandType определяется тип нотификации:

* Control:
  + purchasedVodSync — нотификация об изменении списка купленных фильмов.
  + profileSync — нотификация об изменении сохраненных позиций просмотра.
  + reloadNPvrList — нотификация об изменении запланированных/записанных ТВ-передач (nPVR).
  + reloadServices — нотификация об изменении подключенных услуг.
  + recomPkg — передача рекомендованного к покупке пакета ТВ-каналов. Дополнительный параметр — id — идентификатор пакета ТВ-каналов.
  + reloadDeviceList — нотификация о перезагрузке списка устройств. Приходит на все устройства, подключенные к учетной записи.
  + deletedFromGroup — оповещение об удалении устройства из группы. Приходит на удаляемое из группы устройство.
  + updateStatus — оповещение об изменении статуса. Дополнительный параметр — state. Иначе — неподдерживаемая нотификация.
* Message
  + msgDisplay — сообщения.
    - Дополнительный параметр message — текст сообщения.
    - Иначе, неподдерживаемая нотификация.
* EmergencyMessage
  + msgDisplay — сообщения.
    - Дополнительный параметр — “message” — текст сообщения.
    - Иначе — неподдерживаемая нотификация.

### Получение списка услуг

Получение списка услуг осуществляется методом ProductOffer/list\_services\_terminal. Метод возвращает список услуг по типу устройства, тарифному плану и региону.

Атрибут version="" элемента <services> — дата обновления данного тарифного плана в SDP в формате Unix time. Каждый элемент <service> соответствует отдельной услуге в тарифном плане. Запросы подписки услуг с terminal\_type=0 нужно направлять в личный кабинет, подписки услуг с terminal\_type=<terminal\_type> — в SDP. Параметры услуги (экземпляр <service>):

* <isInvisible> — флаг невидимости услуги, задаваемый в Admin UI. Значения: True — услуга невидимая; False — услуга видимая.
* <showIfInvisibleAndDepends> — флаг, по активации которого невидимая зависимая услуга отображается в интерфейсе пользователя
* <oneTime> — тип тарификации услуги. Значения: True — разовая; False — периодическая.
* <plType> — тип подписки прайс-листа услуги: Purchase — разовая; Subscribe — абонентская плата; Daily — ежедневная.
* <mandatory> — обязательность услуги в тарифном плане. Значения: True — обязательная, подписана по умолчанию, не может быть отменена; False — опциональная, может быть подписана/отменена абонентом. Логика подписки/отмены зависит от параметра <terminalType>. Если <terminalType> = 0, то на web-портале управление услугой выполняется в личном кабинете, в клиентском интерфейсе на устройствах это отображается как статус «услуга подписана/не подписана», без возможности подписки. Если <terminalType> ≠ 0, управление услугой выполняется в зависимости от параметра SubscribeMode, возвращаемого в методе ProductOffer/list\_services.
* <terminalType> — внешний идентификатор типа терминала, для которого предназначена услуга (по настраиваемому справочнику SDP)
* <packageId> — используется для абонементов. Для услуг типа VODABONEMENT, KARAOKEABONEMENT — идентификатор пакета контента
* <allowedPurchases> — используется для абонементов. Максимально доступное число покупок по абонементу.
* <unlimited> — используется для абонементов. Признак безлимитного срока действия абонемента. Значения: true — Безлимитный абонемент. Значение <allowedPurchases> игнорируется. false — Абонемент с ограничением числа покупок, указанным в <allowedPurchases>.
* <isDaily> — признак периодической услуги с ежедневной абонентской платой. Значения: True — ежедневная абонентская плата; False — ежемесячная абонентская плата.
* <endDate> — дата и время окончания предоставления услуги в формате Unix time.
* <type> — тип услуги. Возможные значения:
  + CHANNELPACKAGE — пакет ТВ-каналов;
  + VODPACKAGE — VoD-пакет для разовой покупки (видеофильмы);
  + VODABONEMENT — абонемент на пакет видеофильмов;
  + RADIOCHANNELPACKAGE — пакет радиоканалов;
  + KARAOKEPACKAGE — пакет караоке;
  + KARAOKEABONEMENT — абонемент караоке;
* <externalId> — идентификатор услуги для внешних систем.
* <id> — внутренний идентификатор услуги в SDP.
* <description> — описание услуги.
* <name> — название услуги.
* <price> — используется для периодических услуг, если пусто <bundledGrp>. Стоимость услуги за период использования, значение в копейках.
* <isAccessByLock> — флаг доступности услуги для заблокированных учетных записей (у которых statusExternalId=BLOCKED). Значения: false — услуга не должна быть доступна, если учетная запись абонента заблокирована. true — услуга доступна для заблокированных абонентов.
* <changeDate> — дата изменения тарифного плана.
* <plDiscriminator> — тип прайс-листа:
  + SUBSCRIPTION…PRICELIST или …S\_PRICELIST — периодическая услуга.
  + …OT\_PRICELIST — услуга с разовой покупкой.
* <plId> — идентификатор прайс-листа.
* <unsubscribeByLimit> — используется для абонементов. Флаг отписки абонемента по исчерпании покупок: true — отписывать абонемент по исчерпанию разрешенного числа покупок; false — не отписывать абонемент
* <bundledGrp> — используется для периодических услуг. Имя ценовой группы. Цены из <bundle\_prices> применяются для услуг с одинаковым именем ценовой группы. Если имя ценовой группы не задано, то стоимость услуги за период использования равна <price>.
* <bundle\_prices> — используется для периодических услуг. Список цен, которые распространяются на данную ценовую группу. Значение цены в копейках. В зависимости от количества уже подписанных услуг одной группы очередная услуга подписывается по разной цене. Если подписывается первой (seq\_num=1), то применяется соответствующая цена (price), если второй (seq\_num=2), то другая цена, и т.д. Пример: услуга относится к ценовой группе BASEGROUP, ко всем услугам данной группы применяется цена 22000 копеек, если услуга подписывается первой, цена 10000 копеек, если услуга подписывается второй.

<bundledGrp>BASECPGROUP</bundledGrp>  
<bundle\_prices>  
 <bundle\_price>  
 <price>22000</price>  
 <seq\_num>1</seq\_num>  
 </bundle\_price>  
 <bundle\_price>  
 <price>10000 </price>  
 <seq\_num>2 </seq\_num>  
 </bundle\_price>  
</bundle\_prices>

* <subscribeMode> — доступность функций управления услугой в интерфейсе для абонента:
  + 1 — Можно подписать/отписать
  + 0 — Нельзя подписать/отписать
  + 2 — Можно подписать, нельзя отписать.
* <appleId> — идентификатор услуги в App Store.
* <androidId> — служебный идентификатор.
* <smarttvId> — служебный идентификатор.
* <depends> — используется для отображения нескольких услуг как одной в интерфейсе клиентского приложения. Если <depends> пусто, то для услуги нет зависящих от нее услуг. Если в <depends> есть список идентификаторов (<id>), то перечисленные услуги не должны быть видимы в интерфейсе (зависящие услуги). Команды на подписку/отписку зависящих услуг посылать не нужно, зависимости контролирует SDP и система биллинга. Если <parent><id> пусто, то вместо них должна отображаться одна данная услуга, которая является объединяющей. Если <parent><id> не пусто, то данная услуга тоже является зависящей и не должна отображаться в интерфейсе. Зависимость услуг может быть любой вложенности, на текущий момент используется 1 уровень для услуги «Управление просмотром». В SDP она представляется как набор из трех услуг — пакета TSTV, пакета nPVR и пакета Pause Live, которые не должны быть видимы в интерфейсе.
* <parent> — используется для груповых услуг У услуг, которые должны отображаться в интерфейсе, <parent> пусто. Для управления подпиской нужно посылать команду только на подписку/отписку услуг с пустым <parent>, команды на подписку/отписку зависящих услуг (из списка <depends>) посылать не надо, зависимости контролирует SDP и система биллинга. Если <parent> не пусто (есть вложенный <id>), то услуга не должна отображаться в интерфейсе. Такая услуга является зависящей, и в <id> указан идентификатор объединяющей услуги.
* <portalId> — идентификатор услуги портала в SDP.
* <sortOrder> — индекс сортировки услуг в ТП (поле «Сортировка» в приложении AdminUI, раздел “Тарифные планы” — “Тарифные планы” — “Услуги”). Поле позволяет задавать настраиваемый порядок сортировки услуг при их представлении в клиентском приложении.
* <textOn> — текст, который будет отображаться при подключении услуги.
* <textOff> — текст, который будет отображаться при отключении услуги.
* <recurrentSubscribePeriod> — период рекуррентной подписки — количество дней, через которое будет осуществлено автоматическое продление подписки и списание денег с привязанной карты.
* <isBasic> — признак необходимости отображения услуги в разделе основных пакетов клиентского приложения: 1 — услуга будет отображаться в разделе основных пакетов. Прочие значения — услуга не будет отображаться в разделе основных пакетов. Важно: раздел основных пакетов реализован не во всех UI.

### Покупка услуг

#### Абонементы

Абонемент — это услуга, покупка которой позволяет пользователю просмотреть несколько фильмов из определенного пакета, вместо покупки этих фильмов по отдельности. Условия действия абонемента и контент, на который он распространяется, задаются в SDP. Существует два типа абонементов:

1. С ограничением по числу покупок;
2. Безлимитные по числу покупок.

Абонементы возвращаются как услуги вида:

<service>...<type>xxxABONEMENT</type></service>

по запросу списков услуг ServiceAccount/list\_services и ProductOffer/list\_services\_terminal. Абонемент является периодической услугой в SDP. Стоимость абонемента за период использования указана в параметре price. Абонемент имеет специальные параметры packageId, allowedPurchases и unlimited:

<service>  
...  
<type>xxxABONEMENT</type>  
<packageId>14077421</packageId>  
<allowedPurchases>2</allowedPurchases>  
<unlimited>false</unlimited>  
<endDate>1513430147</endDate>  
<price>1000</price>~~~  
<isDaily>false</isDaily>  
...  
</service>

В параметре packageId указан идентификатор пакета контента, на который распространяется абонемент.

Предусмотрены абонементы на наборы видеофильмов — тип услуги:

<type>VODABONEMENT</type>

Список фильмов, входящих в пакет, можно получить вызовом VodPackage/list\_movies с параметром:

vodPackageId=<packageId>

Если число покупок в абонементе ограничено, то устанавливается параметр:

<unlimited>false</unlimited>

и значение параметра allowedPurchases равно максимально возможному числу покупок по абонементу.

Если абонемент безлимитный, то устанавливается параметр:

`<unlimited>true<unlimited>

и значение параметра allowedPurchases игнорируется.

Логика работы с абонементом:

1. Подписанные для учетной записи абонементы возвращаются в списке услуг ProductOffer/list\_services\_terminal c параметром:

`<mandatory>true</mandatory>

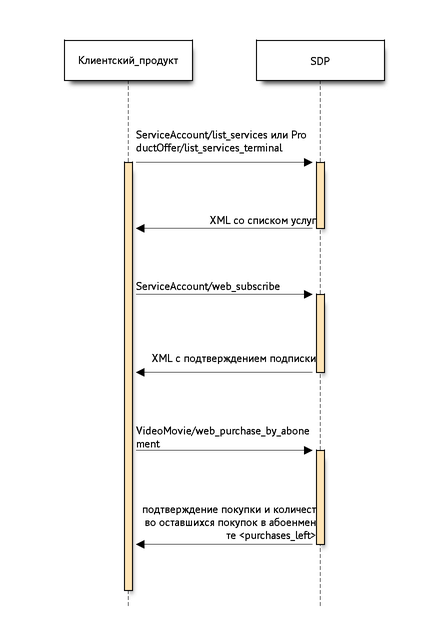
или в списке ServiceAccount/list\_services.

1. Если абонемент является опциональным (абонемент из списка ProductOffer/list\_services\_terminal c параметром:

`<mandatory>false</mandatory>

и еще не подписан для учетной записи (его нет в списке ServiceAccount/list\_services), то этот абонемент можно подписать вызовом метода ServiceAccount/web\_subscribe.

1. Покупки видеофильмов по абонементу выполняются с помощью метода VideoMovie/web\_purchase\_by\_abonement.
2. Опциональный безлимитный абонемент можно отписать от учетной записи вызовом ServiceAccount/web\_subscribe. Абонемент с ограничением числа покупок заканчивается по исчерпании покупок в его учет.



Порядок покупки абонемента

#### Разовые услуги

Разовые услуги — это разовые покупки контента из CoD-пакета (видеофильм, музыкальный альбом и др.). Стоимость разовой услуги списывается со счета абонента в АСР однократно и каждый раз при заказе услуги. Разовые услуги в SDP (CoD-пакеты) — это услуги типа:

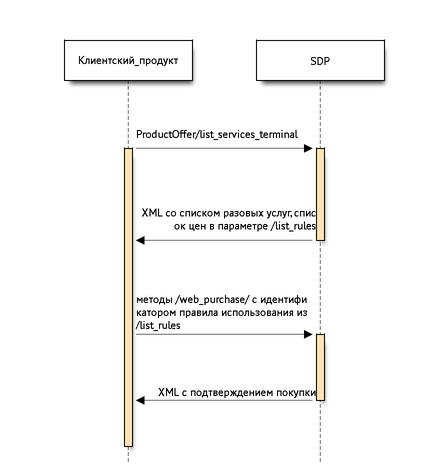
1. VODPACKAGE — пакет видеофильмов для разовой покупки;
2. CONTENTBUNDLEPACKAGE — пакет сезонов сериалов для разовой покупки;
3. SERIESPACKAGE — пакет сериалов.

Логика работы с разовой услугой:

1. Доступные для учетной записи CoD-пакеты возвращаются в списке услуг ProductOffer/list\_services\_terminal c параметром:

<mandatory>true</mandatory>

1. Матрица стоимостей разовой услуги в зависимости от варианта использования для каждого CoD-пакета возвращается в методе ServiceSpec/list\_terminal\_type\_rules.
2. Разовые покупки из CoD-пакета выполняются с помощью метода /web\_purchase. Для определения стоимости контента в параметрах вызова необходимо передать идентификатор правила использования, полученный из ServiceSpec/list\_terminal\_type\_rules. Определенная стоимость будет списана со счета абонента в АСР, параметр price CoD-пакета игнорируется.
3. Чтобы получить список купленного контента, следует выполнить метод ...Package/list\_purchased\_<content> для каждого CoD-пакета в тарифном плане или выполнить метод ...Package/list\_purchased\_by\_acc по каждому типу CoD.



Порядок покупки абонемента

#### Сезоны и сериалы

Отдельные серии в SDP объединены в сезоны, сезоны объединены в сериалы. В SDP предусмотрены пакеты сериалов, пакеты сезонов и пакеты отдельных серий. Абонент может купить отдельный сериал, сезон или серию. Пакеты серий возвращаются в тарифном плане как услуги типа VODPACKAGE. Пакеты сезонов — CONTENTBUNDLEPACKAGE. Пакеты сериалов — SERIESPACKAGE.

**Логика получения всех серий сезона сериала:**

Загрузить список фильмов из каждого пакета VODPACKAGE, полученного в списке услуг в тарифном плане методом ProductOffer/list\_services\_terminal. Чтобы загрузить список фильмов, входящих в пакет, используется метод VodPackage/list\_movies, с id VoD-пакета. Если серии не предназначены для поштучной продажи, список правил покупки ServiceSpec/list\_terminal\_type\_rules будет пустым.

Фильмы, у которых указан <bundles><id>, являются эпизодами сезона с данным id. Выбор всех фильмов с одинаковым id возвращает список всех серий сезона.

Если в тарифном плане нет пакета CONTENTBUNDLEPACKAGE, то можно загрузить описание сезона вызовом ContentBundle/list?ID=id сезона. Если в тарифном плане кроме VODPACKAGE серий есть пакеты сезонов CONTENTBUNDLEPACKAGE, то серии могут продаваться целыми сезонами.

**Логика получения списка сезонов сериала:**

Чтобы получить список сезонов в пакете, следует вызвать метод ContentBundlePackage/list\_bundles. Если сезоны не предназначены для поштучной продажи, список правил покупки ServiceSpec/list\_terminal\_type\_rules будет пустым. Сезоны, у которых указан <series><id>, являются сезонами сериала с данным id. Выбор всех сезонов с одинаковым id — вернет список сезонов сериала. Если в тарифном плане нет пакета SERIESPACKAGE, то можно загрузить описание сериала вызовом Series/list?ID=id сериала.

**Логика получения списка сериалов:**s

Если в тарифном плане есть пакеты SERIESPACKAGE, то серии могут продаваться целыми сериалами. Для получения списка сериалов в пакете используется метод SeriesPackage/list\_series. Для загрузки условий покупки сериалов из пакета используется метод ServiceSpec/list\_terminal\_type\_rules, с id пакета в качестве параметра.

Жанры сезона — значения из справочника ContentBundleGenre/list.

* Чтобы получить список купленных серий для учетной записи, следует вызвать метод VodPackage/list\_purchased\_movies.
* Чтобы получить список купленных сезонов для учетной записи, следует вызвать метод ContentBundlePackage/list\_purchased\_bundles.
* Чтобы получить список купленных сериалов для учетной записи, следует вызвать метод SeriesPackage/list\_purchased\_series.

**Серии:** Пакет серий (эпизодов сезона) для поштучной продажи в SDP возвращается в тарифном плане как услуга типа VODPACKAGE. Принцип работы с сериями аналогичен работе с фильмами.

Для получения списка серий используется метод VodPackage/list\_movies, с идентификатором пакета серий. Для каждой серии в возвращаемом значении VodPackage/list\_movies будет указан идентификатор сезона, в который она входит — <bundles><id>:

<film> ... <bundles> <id>15470610</id> </bundles> ... </film>

Для получения описания сезона используется метод ContentBundle/list?ID=\<id сезона>. Для получения списка правил покупки отдельной серии в рамках пакета серий, следует вызвать VodPackage/list\_movies с идентификатором пакета. Купленные для учетной записи серии возвращаются вместе с купленными фильмами в одном списке в результате вызова ServiceAccount/list\_purchased.

### Разделение услуг по типам устройств

Услуги в тарифном плане и абонентское устройство имеют параметр terminal\_type — идентификатор типа устройства. Список всех услуг, которые должны быть доступны на устройстве состоит из списка услуг для всех устройств и списка услуг для устройств конкретного типа. Значение terminal\_type устройства возвращается в методах ServiceAccount/web\_authorize? и ServiceAccount/bind?. Значение terminal\_type услуги указано в описании каждой услуги, возвращаемом в методе ProductOffer/list\_services (в методе ServiceAccount/list\_services сразу возвращаются только доступные для данного устройства услуги).

Параметр terminal\_type может принимать следующие значения:

* 0 — Услуга для всех устройств.
* 1 — Для основных STB.
* 2 — Для дополнительных STB.
* 3 — Для iPAD
* 4 — Для WEB (приложений на портале).

Пример: чтобы получить получить список всех подключенных услуг для iPAD-приложения следует выполнить следующие шаги:

1. В методе ServiceAccount/web\_authorize? получить тип устройства — значение terminal\_type. Для iPAD-устройства значение будет <terminal\_type>3</terminal\_type>.
2. Выполнить ProductOffer/list\_services.
3. Выбрать услуги с <mandatory>true</mandatory> и <terminal\_type>true</terminal\_type> (для всех устройств). Пусть это будут два пакета каналов — Пакет 1 и Пакет 2.
4. Выбрать все услуги с с true и <terminal\_type>3</terminal\_type> (для iPAD). Пусть это будет пакет каналов Пакет 4.
5. Выполнить ServiceAccount/list\_services.
6. Выбрать все услуги с <terminal\_type>1</terminal\_type>. Пусть это будет пакет каналов Пакет 5.
7. Результат: полный список подключенных услуг для iPAD-приложения равен: Пакет 1, Пакет 2, Пакет 4, Пакет 5.

### Логика видимости услуг в интерфейсе устройств

Видимость услуги в интерфейсе для абонента определяется совокупностью нескольких параметров. Если у услуги установлен флаг <isInvisible>false</isInvisible> из метода ProductOffer/list\_services\_termina, то услуга отображается в клиентском интерфейсе в следующих случаях:

* Услуга приходит в ProductOffer/list\_services\_terminal, но ее нет в ServiceAccount/list\_services абонента;
* Услуга приходит в ServiceAccount/list\_services с <ServiceState>0</isInvisible>ServiceState> или <ServiceState>1</isInvisible>ServiceState>.

В остальных случаях при <isInvisible>false</isInvisible> услуга не отображается. Если у услуги флаг <isInvisible>true</isInvisible>, то услуга не отображается, кроме случая, когда она приходит в ServiceAccount/list\_services с <ServiceState>0</ServiceState> или <ServiceState>1</ServiceState> — тогда услуга отображается. При этом существуют исключения, когда невидимая услуга не отображается со <ServiceState>0</ServiceState>:

1. Выполняются одновременно 3 условия:
   * serviceState = 0;
   * isInvisible=true;
   * subscribeMode = 0 (запрещено подписывать/запрещено отписывать).
2. Выполняются одновременно 3 условия:
   * serviceState = 0;
   * isInvisible=true;
   * услуга связана с родительским пакетом (зависима от другой услуги).

Однако исключение 2 не будет выполняться, если услуга приходит в ProductOffer/list\_services\_terminal с флагом <showIfInvisibleAndDepends>.

### Получение описаний ТВ-каналов

Описания ТВ-каналов можно получить вызовом метода ChannelPackage/list\_channels. Источник вещания ТВ-канала указывается в параметре <urls><location> для соответствующего региона.

Пример:

<urls><!--переопределенные по регионам URL источников вещания ТВ-канала-->  
 <location id="1000001">  
 <url>http-streaming/1tv.m3u8</url>  
 </location>  
 <location id="1000002">  
 <url>http-streaming/1tv.m3u8</url>  
 </location>  
 </urls>

Если значения в <urls> для нужного региона отсутствуют, используется источник вещания по умолчанию, указанный в параметре <url>:

<url>igmp://239.195.8.4:30120</url><!--адрес вещания ТВ-канала по умолчанию -->  
<pipUrl>igmp://229.255.0.100:5600</pipUrl><!--адрес вещания ТВ-канала для PIP по умолчанию -->

### Получение описаний фильмов

Описания фильмов можно получить вызовом VodPackage/list\_movies. Имя файла с фильмом на видеосервере указывается в параметре <assets><content\_asset><ifn>:

<assets><!--физические источники-->  
 <content\_asset><!--описание физического источника-->  
 <id>1029753</id><!--идентификатор источника-->  
 <ifn>http-streaming/GreenZone.m3u8</ifn><!--имя файла на видеосервере или URL-->  
 <duration>300</duration><!--длительность в секундах-->  
 <asset\_type>378264</asset\_type><!--идентификатор типа источника (=типа ассета), используется в правилах использования VodPackage/list\_rules-->  
 <is\_crypted>0</is\_crypted><!--флаг шифрования. 0 — не шифрован-->  
 <transport>http</transport><!--протокол вещания-->  
 <loc><!--идентификатор регионов, в которых доступен ассет-->  
 <id>1013628</id>  
 <id>1021478</id>  
 </loc>  
 </content\_asset>  
</assets>

Дополнительно в секции <assets><preview\_asset> может быть указан трейлер фильма.

### Периодические услуги в SDP

Периодические услуги — это услуги с абонентской платой за определенный период использования, 1 месяц или 1 день. Тип услуги передается в параметре <type>.

Периодические услуги в SDP — это услуги типа:

CHANNELPACKAGE — пакет ТВ-каналов. RADIOCHANNELPACKAGE — пакет радиоканалов. xxxABONEMENT — абонемент на пакет контента. BUNDLEPACKAGE — групповые услуги.

Периодическая услуга имеет специальные параметры price и isDaily:

<service>  
...  
<type>CHANNELPACKAGE</type>  
<endDate>1513430147</endDate>  
<isDaily>false</isDaily>  
<price>1000</price>  
...  
</service>

За период пользования услугой со счета абонента АСР списывает стоимость, указанную в параметре price (в копейках). Период снятия абонентской платы определяется по isDaily: <isDaily>true</isDaily> — 1 день (ежедневная абонентская плата), <isDaily>false</isDaily> — 1 месяц (ежемесячная абонентская плата). В примере выше ежемесячная абонентская плата за пакет ТВ-каналов составляет 10 рублей (10,00). «Бесплатная» периодическая услуга имеет <price>0</price>. Периодические услуги действуют до момента, указанного в параметре endDate.

Логика работы с периодической услугой:

1. Подписанные для учетной записи периодические услуги возвращаются в списке услуг ProductOffer/list\_services\_terminal c параметром <mandatory>true</mandatory>, или в списке ServiceAccount/list\_services.
2. Если услуга является опциональной (услуга из списка ProductOffer/list\_services\_terminal c параметром <mandatory>false</mandatory>) и еще не подписана для учетной записи (ее нет в списке ServiceAccount/list\_services), то эту услугу можно подписать вызовом метода ServiceAccount/web\_subscribe.
3. Опциональную услугу можно отписать от учетной записи вызовом метода ServiceAccount/web\_subscribe.

### Групповые услуги

В SDP поддерживается понятие групповых услуг (пакетов услуг). Данный тип услуг используется для управления прочими оказываемыми услугами в рамках выполнения одного запроса на подписку/отписку услуг. Сама групповая услуга как такового сервиса не оказывает, однако может быть тарифицируемой. Групповые услуги также используются для подключения комплексных тарифов. В случае если услуга, входящая в комплекс, включена на учетной записи вместе с групповой — значит она была подключена со стороны АСР в рамках комплексного тарифа, иначе — самим абонентом. Логика работы с групповыми услугами аналогична обычным периодическим услугам.

### Закачка купленных фильмов (оффлайн-просмотр)

Некоторые купленные фильмы и серии можно закачивать на устройство и просматривать в оффлайн режиме.

Определяет возможность загрузки фильма поле <upload>. Для большинства фильмов этот параметр не приходит вообще, но если он есть и равен “1” (<upload>1</upload>), значит фильм можно скачивать на устройство после покупки.

#### Алгоритм загрузки фильма:

* Проводится проверка, куплен ли фильм, и установлен ли флагу <upload> значение 1.
  + Если фильм куплен и его можно скачивать, в интерфейсе отображается кнопка “Скачать”.
  + Если фильм не куплен или его нельзя скачивать, то кнопка “Скачать” не отображается и загрузка невозможна.
* Проверяется размер фильма:
  + Сформировать URL для получения плейлиста фильма (этот же URL используется для потокового воспроизведения фильма). URL плейлиста формируется так: http://vod-ott.svc.iptv.rt.ru/hls + ifn. ifn — значение поля <ifn> из секции <ott\_content\_asset> из секции <assets> в информации о фильме. При этом ПУТЬ\_К\_ФИЛЬМУ всегда оканчивается на “.wvm”. Для воспроизведения файла “.m3u8” необходимо в пути заменить “.wvm” на “/variant.m3u8” Пример сформированного URL: http://vod-ott.svc.iptv.rt.ru/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\\_\_q0w2\_ar2e2\_film/variant.m3u8
  + Скачать файл плейлиста по сформированному URL.
  + Из полученного файла необходимо выбрать максимальный битрейт. Для этого в файле ищутся строки с параметром BANDWITH=, из которых выбираются все значения, идущие после BANDWIDTH=. Пример: #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1500000. Максимальное из значений сохраняется.
  + Формируется URL для получения длины фильма в максимальном качестве:
    - Из frontendUrl (http://fe-ct.svc.iptv.rt.ru:80/mrf-ct-100001-100001) вырезается адрес сервера (http://fe-ct.svc.iptv.rt.ru/).
    - В URL плейлиста (http://vod-ott.svc.iptv.rt.ru/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\\_\_q0w2\_ar2e2\_film/variant.m3u8) адрес сервера (http://vod-ott.svc.iptv.rt.ru/) заменяется на адрес сервера из URL фильма в максимальном качестве (http://fe-ct.svc.iptv.rt.ru/).
    - hls/ заменяется на hls/size/.
    - /variant.m3u8 заменяется на /bw" + найденное значение битрейта.
    - Результат: http://fe-ct.svc.iptv.rt.ru/hls/size/sd\_2014\_Vozdushnyy\_marshal\\_\_q0w2\_ar2e2\_film/bw1500000
  + Осуществляется запрос на полученный адрес, в ответе из заголовка выбирается значение “Content-Length” — размер файла в байтах.
* Проверяется количество свободного места на устройстве. Если места недостаточно, выводится сообщение об ошибке. Если достаточно — продолжается загрузка файла.
* Определяется путь для сохранения фильма.
* Формируется URL для получения фильма в максимальном качестве:
  + Скачивается файл плейлиста по сформированному ранее URL (http://vod-ott.svc.iptv.rt.ru/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\\_\_q0w2\_ar2e2\_film/variant.m3u8).
  + В файле находится строка, содержащая EXT-X-STREAM-INF BANDWIDTH и где BANDWIDTH равно максимальному битрейту.
  + Если строка нашлась, значит в следующей строке будет содержаться необходимый URL на плейлист с фильмом нужного качества, который следует сохранить. Пример: #EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1500000, http://212.193.147.129/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film/bw1500000/video.ets/playlist.m3u8
* По сформированному адресу скачивается файл плейлиста. Файл сохраняется в той же папке, в которую будет скачиваться фильм. Этот файл будет использоваться для проигрывания плеером фильма оффлайн. Файл содержит окончания ссылок на ts-фрагменты фильмов, и ключи для проигрывания.
* Извлекаются ключи:
  + В файле плейлиста необходимо найти строки с EXT-X-KEY, в которых находятся ссылки на ключи в значениях полей URI. Таких строк в плеере может быть несколько Пример: #EXT-X-KEY: METHOD=AES-128, IV=0x00000000000000000000000000000000, URI="https://vmxott.svc.iptv.rt.ru/CAB/keyfile?r=sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film&t=VOD&p=0" Результат: https://vmxott.svc.iptv.rt.ru/CAB/keyfile?r=sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film&t=VOD&p=0
  + Список ссылок на ключи необходимо передать в плеер и сохранить, чтобы использовать при воспроизведении фильма в офлайновом режиме.
* Фрагменты фильма скачиваются и сохраняются в целевой папке по следующему алгоритму:
  + Все строки файла плейлиста, в которых есть подстрока #EXT, игнорируются.
  + Непроигнорированные строки должны содержать окончание ссылки на фрагмент фильма (пример: 0-488432.ts).
  + В URL фильма в выбранном качестве http://212.193.147.129/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film/bw1500000/video.ets/playlist.m3u8 заменяется playlist.m3u8 на название фрагмента (пример: 0-488432.ts). Пример результата: http://212.193.147.129/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film/bw1500000/video.ets/0-488432.ts
  + По сформированному адресу скачивается фрагмент фильма.
  + По такому же принципу скачиваются остальные фрагменты.

Пример консольной команды для скачивания в текущую директорию всех TS-файлов:

for i in `wget -O — "http://212.193.147.129/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film/bw1500000/video.ets/playlist.m3u8" | grep -v "^#` ; do wget http://212.193.147.129/hls/00/72/sd\_2014\_Vozdushnyy\_marshal\_\_q0w2\_ar2e2\_film/bw1500000/video.ets/$  
{i}  
; done

### UID OTT-устройств

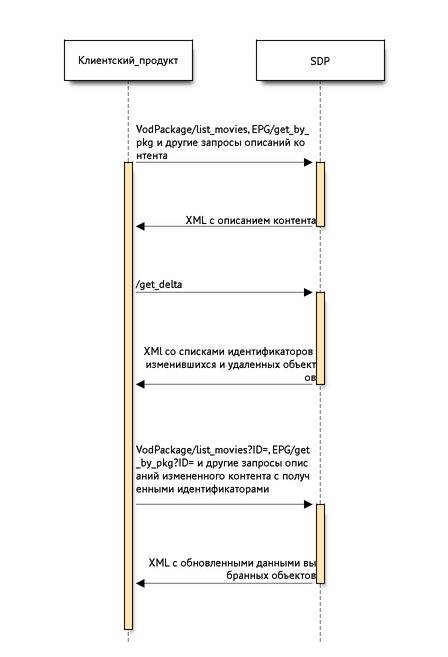
Идентификатор устройства пользователя — UID — формируется для каждого типа устройства следующим образом: для устройств с ОС Android это DeviceID; для SmartTV телевизоров это серийный номер; для устройств с iOS UID генерируется при инсталляции приложения.

### Работа с дельтами

Методы get\_delta обеспечивают постоянное обновление информации о контенте и услугах с минимальным трафиком. Дельта — это список изменений в пакете относительно предыдущего состояния (например, список добавленных в пакет фильмов). Приложение загружает дельту по пакету услуг или программе передач без необходимости полной перечитки всех данных.

Принцип работы с дельтами заключается в следующем:

* Получить описание контента с помощью методов VodPackage/list\_movies, EPG/get\_by\_pkg и др.
* Получить версию пакета — значение атрибута version.
* Через определенный период времени (например, 5 минут) выполнить соответствующий запрос /get\_delta в котором передать полученную ранее версию пакета.
* Сервер SDP возвращает список изменившихся объектов относительно переданной версии:
  + В элементе <add> — список идентификаторов добавленных объектов, относительно предыдущей версии пакета. Для дельты версии 0 сервер возвращает список идентификаторов всех объектов в пакете.
  + В элементе <del> — список идентификаторов удаленных объектов относительно предыдущей версии пакета.



Порядок удаления устройств из группы

Пример возвращаемого значения get\_delta:

<delta code="0">  
 <add>  
 8754940,8758942,8764991  
 </add>  
 <del>  
 8761663,8761662,8761661  
 </del>  
</delta>

* После получения дельты необходимо убрать из интерфейса все объекты, идентификаторы которых получены в секциях <add> и <del>.
* Для каждого объекта из секции add и del по Id следует загрузить изменившееся описание. При этом часть объектов может отсутствовать. Для загрузки объектов следует использовать методы VodPackage/list\_movies?ID=, EPG/get\_by\_pkg?ID= с соответствующим списком идентификаторов.

Пример:

.../VodPackage/list\_movies?ID=8754940,8758942,8764991,8761663,8761662,8761661

* Выбрать последнюю версию загруженных объектов — значение <version>.
* Через некоторое время снова выполнить get\_delta, передав в него последнюю версию, и т.д.

Входные параметры, передаваемые в методах get\_delta, должны быть упорядочены по алфавиту, но входной параметр version должен быть последним.

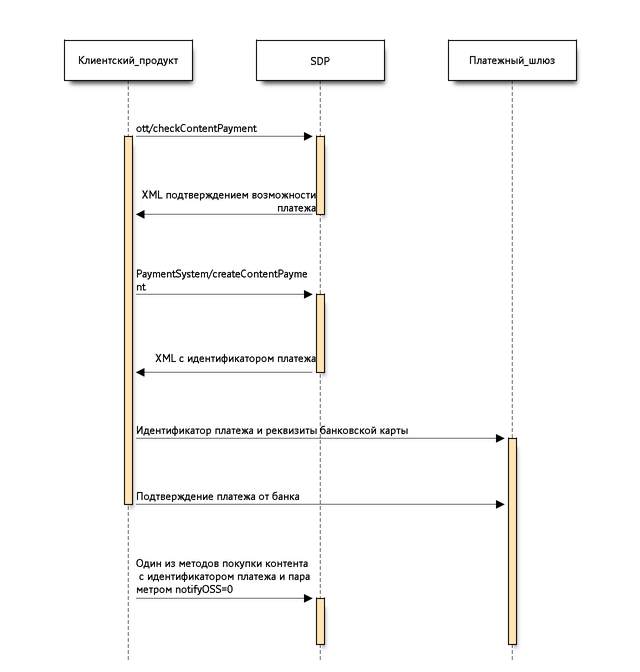
Пример:

.../VodPackage/get\_delta?vodPackageId=57156&version=1321488000

### Работа с платежными системами

Оплата услуг с помощью банковских карт, привязанных к учетной записи, описывается следующим алгоритмом:

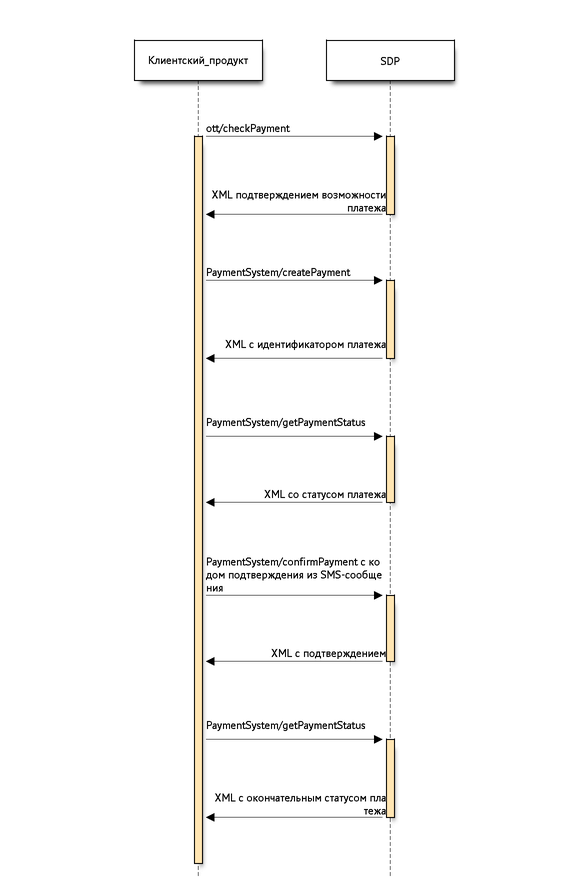
1. При входе абонента в меню оплаты c помощью банковских карт необходимо вызвать метод проверки возможности оплаты ott/checkPayment. Если ошибки нет, процесс оплаты можно продолжить.
2. Выполнить регистрацию платежа с помощью метода PaymentSystem/createPayment.
3. Получить статус платежа с помощью метода PaymentSystem/getPaymentStatus. Если статус платежа равен 101 (PAY\_STATUS\_CODE, “Ожидается получение кода активации из системы-источника”), то следует вывести поле для ввода абонентом кода, полученного им в SMS-сообщении (список кодов статуса см. в описании метода PaymentSystem/getPaymentStatus)
4. Подтвердить платеж с помощью кода, введенного абонентом, выполнив метод PaymentSystem/confirmPayment
5. Получить финальный статус платежа (выполнение/отмена/отклонение) или код ошибки с помощью метода PaymentSystem/getPaymentStatus.



Порядок оплаты с помощью привязанной банковской карты

Оплата услуг с помощью произвольных банковских карт описывается следующим алгоритмом:

1. При входе абонента в меню оплаты c помощью банковских карт необходимо вызвать метод проверки возможности оплаты ott/checkContentPayment. Если ошибки нет, процесс оплаты можно продолжить. Цель использования данного метода — проверить, включен ли сервис оплаты произвольными банковскими картами.
2. Выполнить регистрацию платежа с помощью метода PaymentSystem/createContentPayment. Данный метод вернет уникальный идентификатор платежа на стороне ЕСПП — orderId. Этот идентификатор далее используется приложением для взаимодействия напрямую с платежным шлюзом Сбербанка.
3. Приложение открывает окно ввода реквизитов карты, и далее взаимодействует с платежным шлюзом Сбербанка, используя реквизиты и orderId.
4. После успешного ответа от Сбербанка приложение отправляет в SDP стандартную команду покупки (например, VideoMovie/purchase\_no\_content), передавая параметры notifyOSS=0 и orderId. Покупка фиксируется в SDP — с этого момента контент доступен абоненту.



Порядок оплаты с помощью привязанной банковской карты

Алгоритм для пополнения счета произвольной банковской картой:

1. Вызов метода ott/checkContentPayment. Если ошибки нет, процесс оплаты можно продолжить.
2. Создать ордер для пополнения счета при помощи метода PaymentSystem/createTopUpPayment.

## Описние методов API клиентских приложений платформы «Интерактивное телевидение Ростелеком»

API пользовательских продуктов решения ИТВ доступно по адресу:

<protocol>://{sdp\_ip}:{port}

API пользовательских продуктов решения ИТВ использует сетевые протоколы:

* HTTP;
* HTTPS.

### /ServiceAccount

Методы для работы с учетными записями.

#### /ServiceAccount/web\_authorize

**GET-запрос**

Метод авторизует устройство в SDP.

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).
* appVersion — *number* — Версия клиентского приложения. .
* ipAddress — *string* — IP-адрес устрйоства, используемого клиентским приложением. .

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="row">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:int" name="serviceAccountNumber"/>  
 <xs:element type="xs:string" name="datetime"/>  
 <xs:element type="xs:string" name="state"/>  
 <xs:element type="xs:int" name="locationId"/>  
 <xs:element type="xs:int" name="activationNumber"/>  
 <xs:element type="xs:string" name="backend\_version"/>  
 <xs:element type="xs:int" name="sessionId"/>  
 <xs:element type="xs:int" name="resourceId"/>  
 <xs:element type="xs:int" name="productOfferId"/>  
 <xs:element type="xs:byte" name="terminalType"/>  
 <xs:element type="xs:string" name="multiroom"/>  
 <xs:element type="xs:int" name="notificationLocationId"/>  
 <xs:element type="xs:string" name="reason"/>  
 <xs:element type="xs:string" name="bonusType"/>  
 <xs:element type="xs:string" name="netLogIP"/>  
 <xs:element type="xs:short" name="netLogPort"/>  
 <xs:element type="xs:string" name="networkTypeId"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 102102 </serviceAccountNumber>  
 <datetime> 09/23/2011 12:53:46 </datetime>  
 <state> ACTIVE </state>  
 <locationId> 145462 </locationId>  
 <activationNumber> 111111 </activationNumber>  
 <backend\_version> N/A </backend\_version>  
 <sessionId> 104042 </sessionId>  
 <resourceId> 1751482113 </resourceId>  
 <productOfferId> 2455498 </productOfferId>  
 <terminalType> 0 </terminalType>  
 <multiroom> true </multiroom>  
 <notificationLocationId> 1000050 </notificationLocationId>  
 <reason> FINBLOCK </reason>  
 <bonusType> Standard </bonusType>  
 <netLogIP> 85.94.1.15 </netLogIP>  
 <netLogPort> 5040 </netLogPort>  
 <networkTypeId/>  
 </row>  
</rows>

#### /ServiceAccount/list\_services

**GET-запрос**

Метод возвращает список опциональных услуг в рамках тарифного плана и состояние каждой услуги для учетной записи. Возможность подписки на услугу определяется параметром subscribeMode, возвращаемым методом ProductOffer/list\_services.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="services">  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element name="service" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element type="xs:int" name="id" minOccurs="0"/>  
 <xs:element type="xs:string" name="type" minOccurs="0"/>  
 <xs:element type="xs:string" name="name" minOccurs="0"/>  
 <xs:element type="xs:string" name="description" minOccurs="0"/>  
 <xs:element type="xs:string" name="mandatory" minOccurs="0"/>  
 <xs:element type="xs:string" name="externalId" minOccurs="0"/>  
 <xs:element type="xs:int" name="packageId" minOccurs="0"/>  
 <xs:element type="xs:int" name="endDate" minOccurs="0"/>  
 <xs:element type="xs:byte" name="allowedPurchases" minOccurs="0"/>  
 <xs:element type="xs:string" name="unlimited" minOccurs="0"/>  
 <xs:element type="xs:int" name="startDate" minOccurs="0"/>  
 <xs:element type="xs:string" name="oneTime" minOccurs="0"/>  
 <xs:element type="xs:byte" name="price" minOccurs="0"/>  
 <xs:element type="xs:byte" name="serviceState" minOccurs="0"/>  
 <xs:element type="xs:byte" name="notConfirmed" minOccurs="0"/>  
 <xs:element type="xs:int" name="portalId" minOccurs="0"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<services code="0">  
 <service>  
 <id> 11081787 </id>  
 <type> CHANNELPACKAGE </type>  
 <name> Популярный </name>  
 <description> </description>  
 <mandatory> false </mandatory>  
 <externalId> greg </externalId>  
 <packageId> 14077421 </packageId>  
 <endDate> 1331985600 </endDate>  
 <allowedPurchases> 2 </allowedPurchases>  
 <unlimited> false </unlimited>  
 <startDate> 1331899200 </startDate>  
 <oneTime> false </oneTime>  
 <price> 0 </price>  
 <serviceState> 0 </serviceState>  
 <notConfirmed> 0 </notConfirmed>  
 <portalId> 974305710 </portalId>  
 </service>  
 <service>  
 ...  
 </service>  
 ...  
</services>

#### /ServiceAccount/checkSession

**GET-запрос**

Метод проверяет, не превышает ли количество одновременно запущенных сессий на одной учетной записи установленное значение.# Возвращаемое значение может приходить в формате XML или JSON.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:byte" name="reload"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<rows code="0">  
 <reload> 4 </reload>  
</rows>

#### /ServiceAccount/list\_purchased

**GET-запрос**

Метод возвращает список купленных франшиз и фильмов, купленных по франшизе. Что именно вернулось - франшиза или фильм, определяется полем .

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* secondaryServiceAccountNumber — *string* — Номер личного счета абонента. .

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="row">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:int" name="id"/>  
 <xs:element type="xs:int" name="cur\_id"/>  
 <xs:element type="xs:string" name="su\_discr"/>  
 <xs:element type="xs:int" name="usage\_date"/>  
 <xs:element type="xs:int" name="start\_date"/>  
 <xs:element type="xs:int" name="stop\_date"/>  
 <xs:element type="xs:int" name="packageId"/>  
 <xs:element type="xs:byte" name="forever"/>  
 <xs:element type="xs:byte" name="isProlongation"/>  
 <xs:element type="xs:int" name="profile\_id"/>  
 <xs:element type="xs:string" name="deviceType"/>  
 <xs:element type="xs:string" name="terminalName"/>  
 <xs:element type="xs:string" name="ss\_discr"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="number"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<row number="0">  
 <id> 18770089 </id>  
 <cur\_id> 45316844 </cur\_id>  
 <su\_discr> FRANCHISEPURCHASE </su\_discr>  
 <usage\_date> 1436460462 </usage\_date>  
 <start\_date> 1436460462 </start\_date>  
 <stop\_date> 2145916800 </stop\_date>  
 <packageId> 18770188 </packageId>  
 <forever> 1 </forever>  
 <isProlongation> 0 </isProlongation>  
 <profile\_id> 1104061 </profile\_id>  
 <deviceType> OTTSTB </deviceType>  
 <terminalName> Linux </terminalName>  
 <ss\_discr> FRANCHISEPACKAGE </ss\_discr>  
</row>

#### /ServiceAccount/get\_status

**GET-запрос**

Метод возвращает состояние учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="row">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="id"/>  
 <xs:element type="xs:string" name="statusId"/>  
 <xs:element type="xs:string" name="statusName"/>  
 <xs:element type="xs:string" name="statusExternalId"/>  
 <xs:element type="xs:string" name="reason"/>  
 <xs:element type="xs:string" name="po\_changed"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

#### /ServiceAccount/subscribe\_off

**GET-запрос**

Метод отключает услугу на учетной записи. В случае успешного отключения услуги SDP возвращает code=“0”, в случае успешного принятия заявки на отключение услуги - code=“2”. В случае ошибки – код ошибки и соответствующее сообщение.

**URL-параметры запроса:**

* serviceAccountNumber — *number* — Номер учетной записи. (Обязательный параметр.).
* serviceSpecId — *number* — Идентификатор услуги в SDP. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /ServiceAccount/subscribe\_on

**GET-запрос**

Метод подключает услугу на учетной записи. При покупке через банковскую карту или встроенный магазин метод должен быть оправлен два раза – перед и после списания средств со счета абонента; метод ServiceAccount/subscribe\_on также должен выполняться при получении услуги из ServiceAccount/list\_services с параметром <notConfirmed>=1:

* Устройство должно отправлять запрос ServiceAccount/subscribe\_on с флагом <confirmOption>=prepare перед списанием средств за услугу.
* Если средства списаны, устройство должно посылать второй раз запрос ServiceAccount/subscribe\_on, но с параметром <confirmOption>=finish.
* Если покупка не прошла, устройство должно посылать второй раз запрос ServiceAccount/subscribe\_on, но с параметром <confirmOption>=cancel.
* Также при получении параметра <notConfirmed> =1 в запросе ServiceAccount/list\_services устройство должно проверить оплату за услугу и в случае подтверждения отправить запрос ServiceAccount/subscribe\_on с параметром <confirmOption>=finish,иначе — \ <confirmOption>=cancel.

**URL-параметры запроса:**

* serviceAccountNumber — *number* — Номер учетной записи. (Обязательный параметр.).
* serviceSpecId — *number* — Идентификатор услуги в SDP. (Обязательный параметр.).
* notifyOSS — *number* — Флаг необходимости проведения покупки в АСР. 1 — провести покупку через АСР, выполнить запрос списания стоимости покупки со счета абонента в АСР. 0 — не проводить покупку через АСР, фиксировать в SDP факт оплаты покупки в сторонней системе. .
* orderID — *number* — Идентификатор заказа услуги. .
* dayCount — *number* — Продолжительность подписки на услугу в днях. .
* confirmOption — *string* — Параметр, посылаемый при оплате через банковкую карту или встроенный магазин. Значения: prepare — заявка на подключение услуги; cancel — оплата услуги не прошла; finish — подтверждение оплаты услуги. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /ServiceAccount/web\_subscribe

**GET-запрос**

Этот метод посылает запрос подписки учетной записи на периодическую услугу (в том числе может использоваться для подписки на абонемент). То, в какое состояние нужно перевести подписку на услугу (подписана/отписана) передается с помощью опционального входного параметра isSubscribe.

Если входной параметр isSubscribe не передается, то каждый последующий вызов метода ServiceAccount/web\_subscribe изменяет состояние услуги для учетной записи на противоположное. Т.е. при первом вызове метода услуга подписывается, если она не была подписана; при втором – подписка отменяется, при третьем – снова подписывается и т.д. При покупке через банковскую карту или встроенный магазин метод должен быть оправлен два раза – перед и после списания средств со счета абонента; метод ServiceAccount/web\_subscribe также должен выполняться при получении услуги из ServiceAccount/list\_services с параметром \<notConfirmed>=1:

* Устройство должно отправлять запрос ServiceAccount/web\_subscribe с параметром <confirmOption>=prepare перед списанием средств за услугу.
* Если средства списаны, устройство должно посылать второй раз запрос ServiceAccount/web\_subscribe, с параметром \<confirmOption>=finish.
* Если покупка не прошла, устройство должно посылать второй раз запрос ServiceAccount/web\_subscribe, но с параметром \<confirmOption>=cancel.
* Также при получении параметра \<notConfirmed> =1 в запросе ServiceAccount/list\_services устройство должно проверить оплату за услугу и в случае подтверждения отправить запрос ServiceAccount/web\_subscribe с параметром \<confirmOption>=finish, иначе - \ <confirmOption>=cancel.

При использовании метода могут возникать следующие ошибки:

1. В случае если услуга уже подписана и передается параметр isSubscribe=true, SDP возвращает code="16" и структуру:

<rows code="16">  
 <message> already subscribed </message>  
</rows>

2. В случае если услуга не подписана и передается параметр isSubscribe=false, SDP возвращает code="15" и структуру:

<rows code="15">  
 <message> already unsubscribed </message>  
</rows>

3. При отказе от подписки на услугу, в случае если отписка является отсроченной (возможна только через определенный срок), SDP возвращает структуру:

<rows code="17">  
 <params>  
 <day/> <!-- дата отсроченной отписки от услуги в формате UnixTime-->  
 </params>   
 <message> unsubscription request received </message>  
</rows>

Метод может возвращать пустое значение параметра <day>.

4. При подписке на услугу, в случае если подписка возможна только через определенный срок, SDP возвращает структуру:

<rows code="18">  
 <params>  
 <day/> <!-- дата отсроченной подписки на услугу в формате UnixTime-->  
 </params>  
 <message> subscription request received </message>  
</rows>

Метод может возвращать пустое значение параметра <day>.

5. Если SDP возвращает code="0", это означает, что услуга успешно подписана в онлайн-режиме и уже должна быть доступна. 6. Если SDP возвращает code="2", это означает, что заявка на подписку услуги успешно принята в системе биллинга. Услуга будет доступна в SDP, как только она будет подключена из системы биллинга. Чтобы получить уведомление об изменении списка подключенных услуг сразу после подключения новой услуги в SDP, подпишитесь на нотификацию reloadServices, подробнее см. в разделе 11. Или переодически перечитывайте список услуг вызовом ServiceAccount/list\_services. 7. В случае не специфицированной в SDP ошибки (особое сообщение об ошибке, генерируемое во внешней OSS-системе, например, “Недостаточно средств на счету абонента”), метод возвращает code="1", и соответствующее сообщение об ошибке:

<rows code="1">  
 <message> Сообщение об ошибке </message>  
</rows>

#### /ServiceAccount/get\_billing\_info

**GET-запрос**

Метод возвращает состояние лицевого счета в системе биллинга оператора.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <balance> value1 </balance>  
 <param2> value2 </param2>  
 ...  
 </row>  
</rows>  
  
В случае ошибки – код ошибки и текст сообщения (коды и тексты ошибок зависят от системы биллинга):  
<rows code="100">  
 <message> Биллинг не доступен. (log: QUERY-TO-BILLING:2274518: )  
 </message>  
</rows>

#### /ServiceAccount/list\_available\_profiles

**GET-запрос**

Метод запрашивает список профилей клиентского приложения на устройстве, подключенном к учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <id> 11246124 </id>  
 <name> master </name>  
 <isMaster> 1 </isMaster>  
 <isPublic> 0 </isPublic>  
 <isCurrent> 1 </isCurrent>  
 <favorite> &lt;items&gt; &lt;item&gt;213580&lt;/item&gt; &lt;/items&gt; </favorite>  
 <forbidden> &lt;items&gt; &lt;item&gt;422485&lt;/item&gt; &lt;/items&gt; </forbidden>  
 <sort> &lt;items&gt; &lt;item&gt;11044931&lt;/item&gt; … &lt;/items&gt; </sort>  
 <lastChannelId> </lastChannelId>  
 <specChannelPIDs> &lt;items&gt; &lt;item&gt; &lt;key&gt;12952401&lt;/key&gt; &lt;value&gt;2060&lt;/value&gt; &lt;/item&gt; &lt;/items&gt; </specChannelPIDs>  
 <reminders> &lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt; &lt;remindlist&gt; &lt;program id=&quot;16897255&quot; cid=&quot;2843511&quot;sdate=&quot;08/04/2011 14:44:31&quot;/&gt; &lt;/remindlist&gt; </reminders>  
 <maxAllow> 249564 </maxAllow>  
 <lastAccessLevelId> 249564 </lastAccessLevelId>  
 <isAccessLevelPersistent> 0 </isAccessLevelPersistent>  
 <customProperties> </customProperties>  
 <purchaseVodAllow> 1 </purchaseVodAllow>  
 <pinEnable> 1 </pinEnable>  
 <pin> 0000 </pin>  
 <lang> ru </lang>  
 <viewLimFrom> </viewLimFrom>  
 <viewLimTo> </viewLimTo>  
 <viewDurationHour> </viewDurationHour>  
 <forbidenContent> </forbidenContent>  
 </row>  
 <row number="1">  
 …   
 </row>  
</rows>

#### /ServiceAccount/list\_all\_profiles

**GET-запрос**

Метод возвращает список профилей на всех устройствах, подключенных к учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number=="0">  
 <id> 11246110 </id>  
 <name> Мастер </name>  
 <isMaster> 1 </isMaster>  
 <device> e4:27:71:00:00:d7 </device>  
 </row>  
 <row number=="1">  
 <id> 11246112 </id>  
 <name> Семья </name>  
 <isMaster> 0 </isMaster>  
 <device> логинIPad </device>  
 </row>  
 ...  
</rows>

### /VodPackage

Методы для работы с пакетами фильмов.

#### /VodPackage/list\_purchased\_movies

**GET-запрос**

Метод возвращает список купленных фильмов для учетной записи. Метод необходимо вызывать для каждого VoD-пакета в тарифном плане.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* vodPackageId — *number* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="films">  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element name="film" maxOccurs="unbounded" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор профиля для которого покупка должна быть доступна. Если указано значение и параметр master-профиля isPublic=0, то купленный контент должен быть доступен только для данного профиля. Иначе покупка должна быть доступна для всех профилей. </xs:documentation>  
 </xs:annotation>  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element type="xs:int" name="id" minOccurs="0"/>  
 <xs:element type="xs:int" name="cur\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор фильма </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="usage\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор правила использования, content\_usage\_rule\_id </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="start\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="stop\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время начала срока покупки фильма(unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="profile\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время окончания срока покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<films code="0">  
 <film>  
 <id> 2269681 </id>  
 <cur\_id> 57194 </cur\_id>  
 <usage\_date> 1260271353 </usage\_date>  
 <start\_date> 1260271353 </start\_date>  
 <stop\_date> 1260357753 </stop\_date>  
 <profile\_id/>  
 </film>  
 <film>  
 …  
 </film>  
...  
</films>

#### /VodPackage/list\_purchased\_by\_acc

**GET-запрос**

Метод возвращает список купленных фильмов для учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* secondaryServiceAccountNumber — *string* — Номер личного счета абонента. .
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element name="row" maxOccurs="unbounded" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> Имя устройства </xs:documentation>  
 </xs:annotation>  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element type="xs:int" name="id" minOccurs="0"/>  
 <xs:element type="xs:int" name="cur\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор фильма </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="usage\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор правила использования, content\_usage\_rule\_id </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="start\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="stop\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время начала срока покупки фильма(unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="profile\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время окончания срока покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="packageId" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор профиля, для которого покупка должна быть доступна. Если указано значение и параметр master-профиля isPublic=0, то купленный контент должен быть доступен только для данного профиля. Иначе покупка должна быть доступна для всех профилей. </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="isProlongation" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор пакета, из которого куплен фильм </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="deviceType" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> признак покупки без ограничения срока действия: 0 – купленный контент действителен в течении N часов 1 – нет ограничения по времени использования покупки (автоматическая пролонгация) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="terminalName" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> Тип устройства </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="number" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <id> 2269681 </id>  
 <cur\_id> 57194 </cur\_id>  
 <usage\_date> 1260271353 </usage\_date>  
 <start\_date> 1260271353 </start\_date>  
 <stop\_date> 1260357753 </stop\_date>  
 <profile\_id/>  
 <packageId/>  
 <isProlongation/>  
 <deviceType> STB </deviceType>  
 <terminalName> Motorola\_VIP1963 </terminalName>  
 </row>  
 <row>  
 …  
 </row>  
...  
</rows>

#### /VodPackage/list\_movies

**GET-запрос**

Метод возвращает список и описания видеофильмов, удовлетворяющих параметрам вызова.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. .
* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* ottLocationId — *number* — Идентификатор региона. При передаче ottLocationId, то метод возвращает список фильмов с доступными для просмотра OTT-ассетами. (Обязательный параметр.).
* genreId — *number* — Идентификатор жанра фильма, получаемый методом GenreDictionary/list. .
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. .

**Параметры ответа:**

Пример XML-данных ответа:

<movie\_list code="0">  
 <films packageId="22874845" locationId="100902" groupId="-1" genreId="-1" deviceType="STB" version="1411573486">  
 <film>  
 <id> 12301035 </id>  
 <name> Pan Am. Сезон 1. Пилотная серия (Бесплатная серия) </name>  
 <year> 2011 </year>  
 <actors> Кристина Риччи, Марго Робби, Майкл Мосли, Карин Ванасс </actors>  
 <director> Эндрю Бернштейн, Джон Фортенберри, Томас Шламми </director>  
 <ageLevel> 52653 </ageLevel>  
 <country> США </country>  
 <logo> vod-p1389768360961.jpg </logo>  
 <logo2> vod-a1389768380607.jpg </logo2>  
 <logo3> poster\_portal1353499716031.jpg </logo3>  
 <al> 3 </al>  
 <ir> 0 </ir>  
 <delDate> 1416639472 </delDate>  
 <sDate> 1352923199 </sDate>  
 <eDate> 1414267198 </eDate>  
 <rating> 6.3 </rating>  
 <orderNumber> 1 </orderNumber>  
 <version> 1411004724 </version>  
 <skoro> 0 </skoro>  
 <is3d> false </is3d>  
 <showAsVod> 0 </showAsVod>  
 <isErotic> 0 </isErotic>  
 <isLastEpisode> 1 </isLastEpisode>  
 <packages>  
 <id> 1000 </id>  
 <id> 1100 </id>  
 <id> 22874845 </id>  
 </packages>  
 <genres>  
 <id> 1744094 </id>  
 <id> 22608975 </id>  
 </genres>  
 <bundles>  
 <id> 12547390 </id>  
 </bundles>  
 <loc/>  
 <assets>  
 <preview\_asset>  
 <id> 12301038 </id>  
 <ifn> sd\_2011\_Pan\_am\_s01\_01\_\_trailer.ts </ifn>  
 <duration> 115 </duration>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 1 </is\_crypted>  
 <transport> rtsp </transport>  
 <loc/>  
 </preview\_asset>  
 <content\_asset>  
 <id> 12301135 </id>  
 <ifn> sd\_2011\_Pan\_am\_s01\_01\_\_film.ts </ifn>  
 <duration> 2535 </duration>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 1 </is\_crypted>  
 <transport> rtsp </transport>  
 <loc/>  
 </content\_asset>  
 <ott\_preview\_asset>  
 <id> 12301037 </id>  
 <ifn> trailers/\_definst\_/00/62/sd\_2011\_Pan\_am\_s01\_01\_\_trailer/smil:sd\_2011\_Pan\_am\_s01\_01\_\_trailer/playlist.m3u8 </ifn>  
 <duration> 115 </duration>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 1 </is\_crypted>  
 <transport> http </transport>  
 </ott\_preview\_asset>  
 <ott\_content\_asset>  
 <id> 12301036 </id>  
 <ifn> 00/61/sd\_2011\_Pan\_am\_s01\_01\_\_film.wvm </ifn>  
 <duration> 2535 </duration>  
 <asset\_type> 2377438 </asset\_type>  
 <is\_crypted> 1 </is\_crypted>  
 <transport> http </transport>  
 <loc/>  
 </ott\_content\_asset>  
 </assets>  
 </film>  
 <film>  
 <id> 22033152 </id>  
 <name> Во все тяжкие. Сезон 1. Серия 1 (Бесплатная серия) </name>  
 <year> 2008 </year>  
 <actors> Брайан Крэнстон, Анна Ганн, Аарон Пол, Дин Норрис, Бетси Брандт </actors>  
 <director> Мишель Максвелл МакЛарен, Адам Бернштейн, Винс Джиллигэн </director>  
 <ageLevel> 1000297 </ageLevel>  
 <newDays> 1356983999 </newDays>  
 <country> США </country>  
 <logo> vod-p1389769100606.jpg </logo>  
 <logo2> vod-a1389769116108.jpg </logo2>  
 <logo3> poster\_portal1356607620451.jpg </logo3>  
 <al> 3 </al>  
 <ir> 0 </ir>  
 <isPopular> 1 </isPopular>  
 <upload> 1 </upload>  
 <delDate> 1416824680 </delDate>  
 <sDate> 1356379199 </sDate>  
 <eDate> 1414267198 </eDate>  
 <rating> 8.1 </rating>  
 <imdbR> 9.6 </imdbR>  
 <kinopR> 8.947 </kinopR>  
 <kinopId> 404900 </kinopId>  
 <orderNumber> 1 </orderNumber>  
 <version> 1411004724 </version>  
 <skoro> 0 </skoro>  
 <is3d> false </is3d>  
 <showAsVod> 0 </showAsVod>  
 <isErotic> 0 </isErotic>  
 <packages>  
 <id> 1000 </id>  
 <id> 1100 </id>  
 <id> 22874845 </id>  
 </packages>  
 <genres>  
 <id> 1744094 </id>  
 <id> 22608975 </id>  
 </genres>  
 <bundles>  
 <id> 13811260 </id>  
 </bundles>  
 <loc/>  
 <assets>  
 <ott\_content\_asset>  
 <id> 22033354 </id>  
 <ifn> 00/38/sd\_2008\_Vo\_vse\_tyazhkie\_s01\_01\_\_film.wvm </ifn>  
 <duration> 3307 </duration>  
 <asset\_type> 2377438 </asset\_type>  
 <is\_crypted> 1 </is\_crypted>  
 <transport> http </transport>  
 <loc/>  
 </ott\_content\_asset>  
 <content\_asset>  
 <id> 22033258 </id>  
 <ifn> sd\_2008\_Vo\_vse\_tyazhkie\_s01\_01\_\_film.ts </ifn>  
 <duration> 3307 </duration>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 1 </is\_crypted>  
 <transport> rtsp </transport>  
 <loc/>  
 </content\_asset>  
 <preview\_asset>  
 <id> 22033364 </id>  
 <ifn> sd\_2008\_Vo\_vse\_tyazhkie\_s01\_\_trailer.ts </ifn>  
 <duration> 145 </duration>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 0 </is\_crypted>  
 <transport> rtsp </transport>  
 <loc/>  
 </preview\_asset>  
 <ott\_preview\_asset>  
 <id> 22033359 </id>  
 <ifn> trailers/\_definst\_/00/59/sd\_2008\_Vo\_vse\_tyazhkie\_s01\_\_trailer/smil:sd\_2008\_Vo\_vse\_tyazhkie\_s01\_\_trailer/playlist.m3u8 </ifn>  
 <duration> 145 </duration>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 0 </is\_crypted>  
 <transport> http </transport>  
 </ott\_preview\_asset>  
 </assets>  
 </film>

#### /VodPackage/get\_delta

**GET-запрос**

Метод для получения дельты.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* version — *string* — Версия пакета видеофильмов, полученная последним выводом метода VodPackage/list\_movies. Атрибут version элемента <films>. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. .

**Параметры ответа:**

Пример XML-данных ответа:

<delta code="0">  
 <add> 8754940,8758942,8764991 </add>  
 <del> 8761663,8761662,8761661 </del>  
</delta>

#### /VodPackage/list\_purchase\_period

**GET-запрос**

Метод возвращает список окон проката (если они определены) для каждого фильма из VoD-пакета (или EST VoD-пакета). Идентификатор пакета задается во входном параметре. Фильмы, у которых окна проката для данного пакета не указаны, доступны для продажи/аренды в соответствии с обычными правилами (определяются параметрами Начало и Окончание (проката) — <sDate>, <eDate> — из метода VodPackage/list\_movies).

Клиентское приложение на STB позволяет осуществлять просмотр фильма путем его аренды или покупки (схема EST-продажи), при условии включения фильма в соответствующий пакет (в «VoD-пакет» — для аренды, и в «EST VoD-пакет» — для EST-продажи). Для того чтобы определить доступность фильма в текущий момент как по схеме аренды, так и по схеме EST-продажи, необходимо выполнить следующие запросы:

1. Получить список фильмов.
2. Получить окна проката для соответствующих пакетов (“VoD-пакетов” и “EST VoD-пакетов”) — см. описание запроса VodPackage/list\_purchase\_period.

Для фильма с идентификатором <movie> <id> следует определить доступные окна проката (<row><sDate> <eDate>), после чего принять решение, доступен ли данный фильм в текущий момент по соответствующей схеме приобретения.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. .

**Параметры ответа:**

Пример XML-данных ответа:

<contents code="0">  
 <content>  
 <id> 1719423 </id>  
 <periods>  
 <row>  
 <sDate> 1327525200 </sDate>  
 <eDate> 1415804985 </eDate>  
 </row>  
 <row>  
 <sDate> 1384237122 </sDate>  
 <eDate> 1415804985 </eDate>  
 </row>  
 </periods>  
 </content>  
</contents>

### /EstVodPackage

Методы для работы с пакетами фильмов с возможностью загрузки.

#### /EstVodPackage/list\_purchased\_by\_acc

**GET-запрос**

Метод возвращает список купленных и доступных для просмотра EST VoD-пакетов на учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* secondaryServiceAccountNumber — *string* — Номер личного счета абонента. .

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="row">  
 <xs:annotation>  
 <xs:documentation> Имя устройства </xs:documentation>  
 </xs:annotation>  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:int" name="id"/>  
 <xs:element type="xs:int" name="cur\_id"/>  
 <xs:element type="xs:string" name="su\_discr"/>  
 <xs:element type="xs:string" name="giverAccount"/>  
 <xs:element type="xs:string" name="giverProfile"/>  
 <xs:element type="xs:string" name="abId"/>  
 <xs:element type="xs:int" name="usage\_date"/>  
 <xs:element type="xs:int" name="start\_date"/>  
 <xs:element type="xs:int" name="stop\_date"/>  
 <xs:element type="xs:int" name="packageId"/>  
 <xs:element type="xs:byte" name="isProlongation"/>  
 <xs:element type="xs:int" name="profile\_id"/>  
 <xs:element type="xs:string" name="deviceType"/>  
 <xs:element type="xs:string" name="terminalName">  
 <xs:annotation>  
 <xs:documentation> Тип устройства </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="number"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <id> 15662331 </id>  
 <cur\_id> 19396248 </cur\_id>  
 <su\_discr> CONTENTSERVICEUSAGE </su\_discr>  
 <giverAccount />  
 <giverProfile />  
 <abId />  
 <usage\_date> 1386572673 </usage\_date>  
 <start\_date> 1386572673 </start\_date>  
 <stop\_date> 1418108673 </stop\_date>  
 <packageId> 19396173 </packageId>  
 <isProlongation> 0 </isProlongation>  
 <profile\_id> 16026558 </profile\_id>  
 <deviceType> STB </deviceType>  
 <terminalName> Motorola\_VIP1963 </terminalName>  
 </row>  
</rows>

### /ContentBundlePackage

Методы для работы с сезонами сериалов.

#### /ContentBundlePackage/list\_purchased\_bundles

**GET-запрос**

Метод возвращает список купленных сезонов для учетной записи. Метод необходимо вызывать для каждого пакета сезонов в тарифном плане.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* cbPackageId — *number* — Идентификаторы пакетов сезонов <service> <id>, полученные методами ServiceAccount/list\_services и ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="bundles">  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element name="bundle" maxOccurs="unbounded" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор профиля для которого покупка должна быть доступна. Если указано значение и параметр master-профиля isPublic=0, то купленный контент должен быть доступен только для данного профиля. Иначе покупка должна быть доступна для всех профилей. </xs:documentation>  
 </xs:annotation>  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element type="xs:int" name="id" minOccurs="0"/>  
 <xs:element type="xs:int" name="cur\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор сезона </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="usage\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор правила использования, content\_usage\_rule\_id </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="start\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="stop\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время начала срока покупки сезона (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="profile\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время окончания срока покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<bundles code="0">  
 <bundle>  
 <id> 2269681 </id>  
 <cur\_id> 57194 </cur\_id>  
 <usage\_date> 1260271353 </usage\_date>  
 <start\_date> 1260271353 </start\_date>  
 <stop\_date> 1260357753 </stop\_date>>  
 <profile\_id/>  
 </bundle>  
 <bundle>  
 …  
 </bundle>  
...  
</bundles>

#### /ContentBundlePackage/list\_purchased\_by\_acc

**GET-запрос**

Метод возвращает список купленных сезонов для учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* secondaryServiceAccountNumber — *string* — Номер личного счета абонента. .
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element name="row" maxOccurs="unbounded" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> Имя устройства </xs:documentation>  
 </xs:annotation>  
 <xs:complexType mixed="true">  
 <xs:sequence>  
 <xs:element type="xs:int" name="id" minOccurs="0"/>  
 <xs:element type="xs:int" name="cur\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор сезона </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="usage\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор правила использования, content\_usage\_rule\_id </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="start\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:int" name="stop\_date" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время начала срока покупки сезона (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="profile\_id" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> дата и время окончания срока покупки (unix) </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="packageId" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор профиля для которого покупка должна быть доступна. Если указано значение и параметр master-профиля isPublic=0, то купленный контент должен быть доступен только для данного профиля. Иначе покупка должна быть доступна для всех профилей. </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="deviceType" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> идентификатор пакета, из которого куплен сезон </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element type="xs:string" name="terminalName" minOccurs="0">  
 <xs:annotation>  
 <xs:documentation> Тип устройства </xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <id> 2269681 </id>  
 <cur\_id> 57194 </cur\_id>  
 <usage\_date> 1260271353 </usage\_date>  
 <start\_date> 1260271353 </start\_date>  
 <stop\_date> 1260357753 </stop\_date>  
 <profile\_id/>  
 <packageId/>  
 <deviceType> STB </deviceType>  
 <terminalName> Motorola\_VIP1963 </terminalName>  
 </row>  
 <row>  
 …  
 </row>  
...  
</rows>

#### /ContentBundlePackage/list\_bundles

**GET-запрос**

Метод возвращает список и описания сезонов, удовлетворяющих параметрам вызова. Получить описания можно по пакету сезонов; по ID сезонов, перечислив идентификаторы в запросе через запятую; по ID региона.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<cb\_list code="0">  
 <cb packageId="1521024" locationId="1081481" genreId="null" version="1332288000">  
 <bundle>  
 <id> 1576211 </id>  
 <externalID/>  
 <name> Бони сезон 1 </name>  
 <year> 2012 </year>  
 <actors/>  
 <director/>  
 <country/>  
 <logo> 1332249110964.jpg </logo>  
 <logo2/>  
 <al> 10758 </al>  
 <ageLevel> 10758 </ageLevel>  
 <ir> 0 </ir>  
 <appleId/>  
 <androidId/>  
 <smarttvId/>  
 <sDate/>  
 <eDate/>  
 <rating/>  
 <version> 1332288000 </version>  
 <packages>  
 <id> 1521024 </id>  
 </packages>  
 <genres/>  
 <series>  
 <id> 1576165 </id>  
 </series>  
 <movies>  
 <id> 1576225 </id>  
 <id> 1576235 </id>  
 </movies>  
 <loc>  
 <id> 1081481 </id>  
 <id> 1557546 </id>  
 </loc>  
 </bundle>  
 ...  
 </cb>  
</cb\_list>

#### /ContentBundlePackage/get\_delta

**GET-запрос**

Метод для получения дельты.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* version — *date* — Версия пакета видеофильмов, полученная последним выводом метода VodPackage/list\_movies. Атрибут version элемента <films>. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<delta code="0">  
 <add> 8754940,8758942,8764991 </add>  
 <del> 8761663,8761662,8761661 </del>  
</delta>

#### /ContentBundlePackage/list\_purchase\_period

**GET-запрос**

Метод показывает периоды действия окна показа для данного пакета.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<contents code="0">  
 <content>  
 <id> 1719423 </id>  
 <periods>  
 <row>  
 <sDate> 1327525200 </sDate>  
 <eDate> 1415804985 </eDate>  
 </row>  
 <row>  
 <sDate> 1384237122 </sDate>  
 <eDate> 1415804985 </eDate>  
 </row>  
 </periods>  
 </content>  
</contents>

### /EstContentBundlePackage

Методы для работы с сезонами сериалов, для которых предусмотрена возможность загрузки.

#### /EstContentBundlePackage/list\_purchased\_by\_acc

**GET-запрос**

Метод возвращает список купленных EST сезонов для учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* secondaryServiceAccountNumber — *string* — Номер личного счета абонента. .
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="row">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="id"/>  
 <xs:element type="xs:string" name="cur\_id"/>  
 <xs:element type="xs:string" name="usage\_date"/>  
 <xs:element type="xs:string" name="start\_date"/>  
 <xs:element type="xs:string" name="stop\_date"/>  
 <xs:element type="xs:string" name="profile\_id"/>  
 <xs:element type="xs:string" name="packageId"/>  
 <xs:element type="xs:string" name="deviceType"/>  
 <xs:element type="xs:string" name="terminalName"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

### /SeriesPackage

Методы для работы с сериалами.

#### /SeriesPackage/list\_purchased\_series

**GET-запрос**

Метод возвращает список купленных сериалов для учетной записи. Метод необходимо вызывать для каждого пакета сериалов в тарифном плане.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sPackageId — *number* — Идентификаторы пакетов сериалов <service> <id>, полученные в методах ServiceAccount/list\_services и ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="row">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="id"/>  
 <xs:element type="xs:string" name="cur\_id"/>  
 <xs:element type="xs:string" name="usage\_date"/>  
 <xs:element type="xs:string" name="start\_date"/>  
 <xs:element type="xs:string" name="stop\_date"/>  
 <xs:element type="xs:string" name="profile\_id"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

#### /SeriesPackage/list\_purchased\_by\_acc

**GET-запрос**

Метод возвращает список купленных сериалов для учетной записи.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* secondaryServiceAccountNumber — *string* — Номер личного счета абонента. .
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="rows">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="row">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="id"/>  
 <xs:element type="xs:string" name="cur\_id"/>  
 <xs:element type="xs:string" name="usage\_date"/>  
 <xs:element type="xs:string" name="start\_date"/>  
 <xs:element type="xs:string" name="stop\_date"/>  
 <xs:element type="xs:string" name="profile\_id"/>  
 <xs:element type="xs:string" name="packageId"/>  
 <xs:element type="xs:string" name="deviceType"/>  
 <xs:element type="xs:string" name="terminalName"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="code"/>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

#### /SeriesPackage/list\_series

**GET-запрос**

Метод возвращает список сериалов в пакете.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<series\_list code="0">  
 <series packageId="25182943" locationId="145462" genreId="null" version="1332288000">  
 <serie>  
 <id> 25182920 </id>  
 <externalID/>  
 <name> NewSeries </name>  
 <year/>  
 <actors/>  
 <director/>  
 <country/>  
 <logo/>  
 <logo2/>  
 <al/>  
 <ir> 0 </ir>  
 <isPopular> 1</isPopular> <!-—Флаг «Популярные сериалы»-->  
 <appleId/>  
 <androidId/>  
 <smarttvId/>  
 <sDate> 1332187200 </sDate>  
 <eDate> 1584648000 </eDate>  
 <rating/>  
 <version> 1332288000 </version>  
 <packages>  
 <id> 25182943 </id>  
 </packages>  
 <genres>  
 <id> 15470867 </id>  
 <id> 24267132 </id>  
 <id> 15470875 </id>  
 </genres>  
 <bundles>  
 <id> 15470610 </id>  
 <id> 25181389 </id>  
 </bundles>  
 <loc>  
 <id> 145354 </id>  
 <id> 145462 </id>  
 <id> 25022419 </id>  
 </loc>  
 </serie>  
 ...  
 </series>  
</series\_list>

#### /SeriesPackage/get\_delta

**GET-запрос**

Метод для получения дельты.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* version — *date* — Версия пакета видеофильмов, полученная последним выводом метода VodPackage/list\_movies. Атрибут version элемента <films>. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<delta code="0">  
 <add> 8754940,8758942,8764991 </add>  
 <del> 8761663,8761662,8761661 </del>  
</delta>

### /ProductOffer

Методы для работы с услугами в тарифном плане.

#### /ProductOffer/list\_services\_terminal

**GET-запрос**

Метод возвращает список услуг по типу устройства, тарифному плану и региону.

**URL-параметры запроса:**

* productOfferId — *string* — Идентификатор тарифного плана. (Обязательный параметр.).
* locationId — *string* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* terminalType — *string* — Внешний идентификатор типа устройства, для которого предназначена услуга. (Обязательный параметр.).

**Параметры ответа:**

XML-схема ответа:

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="services">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="service">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="isInvisible"/>  
 <xs:element type="xs:string" name="showIfInvisibleAndDepends"/>  
 <xs:element type="xs:string" name="oneTime"/>  
 <xs:element type="xs:string" name="plType"/>  
 <xs:element type="xs:string" name="mandatory"/>  
 <xs:element type="xs:string" name="terminalType"/>  
 <xs:element type="xs:string" name="packageId"/>  
 <xs:element type="xs:string" name="allowedPurchases"/>  
 <xs:element type="xs:string" name="unlimited"/>  
 <xs:element type="xs:string" name="isDaily"/>  
 <xs:element type="xs:string" name="endDate"/>  
 <xs:element type="xs:string" name="type"/>  
 <xs:element type="xs:string" name="externalId"/>  
 <xs:element type="xs:string" name="id"/>  
 <xs:element type="xs:string" name="description"/>  
 <xs:element type="xs:string" name="name"/>  
 <xs:element type="xs:string" name="price"/>  
 <xs:element type="xs:string" name="isAccessByLock"/>  
 <xs:element type="xs:string" name="changeDate"/>  
 <xs:element type="xs:string" name="plDiscriminator"/>  
 <xs:element type="xs:string" name="plId"/>  
 <xs:element type="xs:string" name="unsubscribeByLimit"/>  
 <xs:element type="xs:string" name="bundledGrp"/>  
 <xs:element type="xs:string" name="bundle\_prices"/>  
 <xs:element type="xs:string" name="subscribeMode"/>  
 <xs:element type="xs:string" name="appleId"/>  
 <xs:element type="xs:string" name="androidId"/>  
 <xs:element type="xs:string" name="smarttvId"/>  
 <xs:element type="xs:string" name="depends"/>  
 <xs:element type="xs:string" name="parent"/>  
 <xs:element type="xs:string" name="portalId"/>  
 <xs:element type="xs:string" name="sortOrder"/>  
 <xs:element type="xs:string" name="textOn"/>  
 <xs:element type="xs:string" name="textOff"/>  
 <xs:element type="xs:string" name="recurrentSubscribePeriod"/>  
 <xs:element type="xs:string" name="isBasic"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

### /WebPortal

Методы для взаимодействия ЕКП Zabava и SDP.

#### /WebPortal/authorize\_san

**GET-запрос**

Метод возвращает авторизационные данные первого найденного на учетной записи веб-подключения (устройства типа WEB в SDP). В результате выполнения метода SDP создает сессию и возвращает ее идентификатор («авторизует» устройство на WEB-подключение). По этой сессии c SDP можно работать далее.

**URL-параметры запроса:**

* serviceAccount — *string* — Номер учетной записи, на которой нужно найти веб-подключение. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 102102 </serviceAccountNumber>  
 <datetime> 09/23/2011 12:53:46 </datetime>  
 <state> ACTIVE </state>  
 <locationId> 145462 </locationId>  
 <activationNumber> 111111 </activationNumber>  
 <backend\_version> N/A </backend\_version>  
 <sessionId> 104042 </sessionId>  
 <resourceId> 1751482113 </resourceId>  
 <productOfferId> 2455498 </productOfferId>  
 <bonusType> Standard </bonusType>  
 <terminalType> 0 </terminalType>  
 <reason> FINBLOCK </reason>  
 <networkTypeId/>  
 </row>  
</rows>

#### /WebPortal/bind

**GET-запрос**

Метод активирует iPad / Web APP в SDP.

Активация требуется для всех новых пользователей iPad / Web APP. Активация привязывает логин/пароль входа в систему к учетной записи абонента в SDP. В случае ошибки (см. Обработка ошибок, ошибки 1-2) логин/пароль iPad / Web APP активирован не будет. В случае если учетная запись заблокирована, активация проходит успешно, но статус у учетной записи остается BLOCKED. Алгоритм активации устройства с подтверждением через e-mail Алгоритм подразумевает, что в качестве параметра login используется адрес электронной почты абонента.

1. Для активации устройства с подтверждением через e-mail в метод необходимо передать параметр useMailConfirmation=1. При удачной отправке сообщения абоненту и после подтверждения абонентом активации произойдет активация устройства. Активация устройства не будет завершена, пока абонент не подтвердит активацию. Авторизация устройства в SDP (метод ServiceAccount/web\_authorize) также не сможет быть выполнена до тех пор, пока абонент не подтвердит активацию.
2. В случае возникновения ошибки при отправке сообщения абоненту по e-mail, метод вернет ошибку с кодом code=1. Для активации устройства потребуется повторный вызов метода WebPortal/bind или WebPortal/bind\_san.

В случае указания абонентом ошибочного адреса e-mail в качестве логина, остается возможность завершить активацию с подтвержением по e-mail с помощью метода Terminal/send\_confirmation\_mail (см. описание метода). В этом случае в параметре login метода Terminal/send\_confirmation\_mail указывается первоначально указанный адрес e-mail (т.е. уже занесенный в БД логин), а в параметре sentToMail настоящий адрес e-mail. Решение о том, что адрес указан ошибочно, принимается в клиентском приложении.

**URL-параметры запроса:**

* activationNumber — *string* — Номер активации. (Обязательный параметр.).
* activationCode — *string* — Код активации. (Обязательный параметр.).
* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).
* tmp\_login — *string* — Признак того, что на учетной записи надо установить признак временного логина и пароля. Допустимые значения: true — установить признак временного логина и пароля; false — не устанавливать. .
* useMailConfirmation — *string* — Признак того, что требуется выполнить подтверждение активации через e-mail. Допустимые значения: 1 — установить признак необходимости подтверждения через e-mail; 0 — не устанавливать. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 102102 </serviceAccountNumber>   
 <state> ACTIVE </state>  
 <locationId> 145462 </locationId>  
 <productOfferId> 2455498 </productOfferId>  
 <terminalType> 0 </terminalType>  
 <bonusType> Standard </bonusType>  
 </row>  
</rows>

#### /WebPortal/bind\_san

**GET-запрос**

Метод используется для «доверительной регистрации абонентов ШПД», т.е. привязки к существующей учетной записи логина/пароля пользователя без запроса кода активации.

Метод активирует все имеющиеся неактивированные веб-подключения на учетной записи в SDP (тем не менее предполагается, что у учетной записи имеется лишь одно неактивированное веб-подключение).

В случае ошибки веб-подключение активировано не будет. В случае если учетная запись заблокирована, активация проходит успешно, но статус у учетной записи остается BLOCKED.

**URL-параметры запроса:**

* serviceAccount — *string* — Номер учетной записи, на которой нужно найти веб-подключение. (Обязательный параметр.).
* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. .
* tmp\_login — *string* — Признак того, что на учетной записи надо установить признак временного логина и пароля. Допустимые значения: true — установить признак временного логина и пароля; false — не устанавливать. .
* useMailConfirmation — *string* — Признак того, что требуется выполнить подтверждение активации через e-mail. Допустимые значения: 1 — установить признак необходимости подтверждения через e-mail; 0 — не устанавливать. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 11122 </serviceAccountNumber>   
 <state> ACTIVE </state>  
 <locationId> 145354 </locationId>  
 <productOfferId> 1716852 </productOfferId>  
 <terminalType> 4 </terminalType>  
 <bonusType> Standard </bonusType>  
 </row>  
</rows>

#### /WebPortal/login\_exists

**GET-запрос**

Метод проверяет существование указанного логина в SDP.

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

#Если логин есть:  
<rows code="0"/>  
  
#Если логина нет:  
<rows code="1"> <message>Login-password not found</message> </rows>

#### /WebPortal/change\_password

**GET-запрос**

Метод изменяет пароль веб-подключения в SDP (т.е. пароль авторизации iPad / Web-клиента в SDP).

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* oldPassword — *string* — Старый пароль для авторизации. .
* newPassword — *string* — Новый пароль для авторизации. (Обязательный параметр.).
* noCheck — *string* — Флаг проверки старого пароля: 0 — проверять, что пароль, переданный в oldPassword, соответствует старому паролю; 1 — не проверять старый пароль. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /WebPortal/service\_account\_add\_request

**GET-запрос**

Метод авторизует устройство в SDP через АСР. Метод используется для всех МРФ, в которых установлена АСР StartIP (Центр, Волга, часть регионов Дальнего Востока).

**URL-параметры запроса:**

* external\_login — *string* — Логин. (Обязательный параметр.).
* external\_password — *string* — Пароль. (Обязательный параметр.).
* serviceProvider — *string* — Внешний идентификатор оператора. Если оператор не задан, запрос на авторизацию передается в АСР, указанную в параметре wsdl.start2com.url конфигурации SDP. .
* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 102102 </serviceAccountNumber>  
 <datetime> 09/23/2011 12:53:46 </datetime>  
 <state> ACTIVE </state>  
 <locationId> 145462 </locationId>  
 <activationNumber> 111111 </activationNumber>  
 <backend\_version> N/A </backend\_version>  
 <sessionId> 104042 </sessionId>  
 <resourceId> 1751482113 </resourceId>  
 <productOfferId> 2455498 </productOfferId>  
 <terminalType> 0 </terminalType>  
 </row>  
</rows>

#### /WebPortal/add\_service\_account

**GET-запрос**

Метод создает учетную запись в SDP, подключает на указанный тарифный план, добавляет на нее 2 подключения для iPad и Web с возможностью авторизации по заданному логину и паролю. Номер активации создаваемых подключений = номеру учетной записи. Код активации = 111111. Учетная запись создается в регионе, заданном в параметре web.default.location конфигурации SDP, если параметр не определен – в регионе default.location. Метод используется для всех остальных МРФ, кроме тех, где АСР не StartIP.

**URL-параметры запроса:**

* service\_account\_number — *string* — Номер учетной записи. (Обязательный параметр.).
* product\_offer\_external\_id — *string* — Внешний идентификатор тарифного плана. (Обязательный параметр.).
* web\_login — *string* — Логин. (Обязательный параметр.).
* web\_password — *string* — Пароль. При передаче необязательного параметра web\_password\_phrase значение пароля принимается как уже зашифрованное по схеме md5 (<пароль> + <секретная\_фраза>) и заносится в БД без шифрования. (Обязательный параметр.).
* web\_password\_phrase — *string* — Секретная фраза. Используется при авторизации для вычисления хэш-функции md5 (<пароль> + <секретная\_фраза>). .
* service\_provider\_external\_id — *string* — Внешний идентификатор оператора. Если значение не указано, учетная запись создается на оператора, заданного в параметре default\_provider\_id конфигурации SDP. .
* tmp\_login — *string* — Признак того, что на учетной записи надо установить признак временного логина и пароля. Допустимые значения: true — установить признак временного логина и пароля; false — не устанавливать. .
* region\_external\_id — *string* — Внешний ID региона или подрегиона. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /WebPortal/set\_login\_and\_pwd

**GET-запрос**

Метод позволяет заменить логин и пароль на учетной записи, которая изначально была создана с временным логином и паролем.

**URL-параметры запроса:**

* service\_account\_number — *string* — Номер учетной записи. (Обязательный параметр.).
* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /WebPortal/service\_account\_phone\_get

**GET-запрос**

Метод используется для получения номера телефона для данной учетной записи.

**URL-параметры запроса:**

* service\_account\_number — *string* — Номер учетной записи. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <FIO> ФИО </FIO>  
 <phone> Контактный телефон </phone>  
 </row>  
</rows>

#### /WebPortal/npvr\_create

**GET-запрос**

Метод ставит ТВ-передачу на запись.

**URL-параметры запроса:**

* service\_account\_number — *string* — Номер учетной записи. (Обязательный параметр.).
* scheduleID — *string* — Идентификатор ТВ-передачи. Значение поля ``` передачи, возвращаемой в методах EPG/get\_by\_Chnnl или EPG/get\_by\_pkg. (Обязательный параметр.).
* location — *number* — Идентификатор региона. (Обязательный параметр.).
* ottAssetName — *string* — Сгенерированное имя файла записанной передачи. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <quotaUsed> 115300000 </quotaUsed>  
 <channelId> 25315013 </channelId>  
 <size\_quota> true </size\_quota>  
 <assetDuration> 600 </assetDuration>  
 <videoserverIpAddress> 10.65.2.202 </videoserverIpAddress>  
 <quotaTotal> 115343360 </quotaTotal>  
 <assetIsCrypted> 0 </assetIsCrypted>  
 <recordedProgramId> 25323052 </recordedProgramId>  
 <quotaRequired> 115388889 </quotaRequired>  
 <assetId> 25323053 </assetId>  
 <assetInternalFileName> NPVR\_RECORD\_25323052.ts </assetInternalFileName>  
 <videoserverPort> 554 </videoserverPort>  
 <assetType> 1 </assetType>  
 <validDate> 1335276000 </validDate>  
 <recordedProgramStatus> 1 </recordedProgramStatus>  
 <accessLevelId> 2 </accessLevelId>  
 <ott\_contentasset> xxx </ott\_contentasset>  
</rows>

### /elk

#### /elk/authorize

**GET-запрос**

Метод возвращает авторизационные данные, необходимые для вызова ProductOffer/list\_services\_terminal и ServiceAccount/list\_services.

**URL-параметры запроса:**

* service\_account\_number — *string* — Номер учетной записи. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 11122 </serviceAccountNumber>  
 <state> ACTIVE </state>  
 <id> 57214 </id>  
 <locationId> 145462 </locationId>  
 <productOfferId> 24871598 </productOfferId>  
 <reason/>  
 <description> test wsdl </description>  
 <terminals>  
 <terminal>  
 <mac> 61:61:61:61:61:61 </mac>  
 <activationNumber> 123666 </activationNumber>  
 <terminalType> 3 </terminalType>  
 <deviceType> STB </deviceType>  
 </terminal>  
 <terminal>  
 <mac> 34:23:32:41:23:12 </mac>  
 <activationNumber> ergergergeg </activationNumber>  
 <terminalType> 1 </terminalType>  
 <deviceType> STB </deviceType>  
 </terminal>  
 …  
 </row>

### /Terminal

#### /Terminal/send\_confirmation\_mail

**GET-запрос**

Метод для отправки абоненту сообщения e-mail с запросом подтверждения активации устройства. Используется при ошибочном указании адреса e-mail в методах WebPortal/bind и WebPortal/bind\_san.

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* sentToMail — *string* — Адрес электронной почты, на который необходимо послать запрос подтверждения активации. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /VideoMovie

Методы для работы c видеофильмами.

#### /VideoMovie/web\_purchase

**GET-запрос**

Метод посылает запрос на разовую покупку видеофильма из библиотеки SmartTUBE SDP. Для прочего внешнего контента (ПО, книги, и т.п.) используется метод Content/web\_purchase.

**URL-параметры запроса:**

* contentId — *number* — Идентификатор видеофильма. (Обязательный параметр.).
* contentPackageId — *number* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* contentUsageRuleId — *number* — Идентификатор правила покупки, по которому приобретается видеофильм полученный с помощью метода ServiceSpec/list\_terminal\_type\_rules, выходной параметр <cur\_id>. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* assetName — *string* — Название фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* notifyOSS — *number* — Флаг необходимости проведения покупки в АСР. 1 — провести покупку через АСР, выполнить запрос списания стоимости покупки со счета абонента в АСР. 0 — не проводить покупку через АСР, фиксировать в SDP факт оплаты покупки в сторонней системе. .
* transactionId — *integer* — Идентификатор транзакции. .
* profileId — *number* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /VideoMovie/web\_purchase\_by\_abonement

**GET-запрос**

Метод посылает запрос на покупку видеофильма для учетной записи по абонементу.

**URL-параметры запроса:**

* abonementId — *string* — Идентификатор абонемента видеофильмов. (Обязательный параметр.).
* contentId — *number* — Идентификатор видеофильма. (Обязательный параметр.).
* contentPackageId — *number* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

#### /VideoMovie/purchase\_by\_franchise

**GET-запрос**

Покупка видеофильма по франшизе. Цена контента в данном случае равна нулю. Покупка доступна только при наличии приобретенной франшизы, т.е. сначала должен быть выполнен метод Franchise/purchase\_no\_content. Если правило использования пакета франшиз - бессрочная покупка (EST), то фильм приобретается бессрочно. В иных случаях по указанному в параметрах правилу.

**URL-параметры запроса:**

* franchisePackageId — *string* — Идентификатор пакета франшиз. (Обязательный параметр.).
* franchiseId — *string* — Идентификатор франшизы. (Обязательный параметр.).
* usageRuleId — *string* — Идентификатор правила покупки фильма. (Обязательный параметр.).
* contentId — *string* — Идентификатор видеофильма. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. .
* profileId — *string* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .
* isRecomended — *string* — Флаг, определяющий куплен ли фильм по рекомендации. .
* dsc — *string* — Имя франшизы. .
* duration — *string* — Продолжительность купленного контента в секундах. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /ContentBundle

#### /ContentBundle/web\_purchase

**GET-запрос**

Метод посылает запрос на разовую покупку сезона сериала из библиотеки SmartTUBE SDP для учетной записи. Для прочего внешнего контента (софт, книги, и т.п.) используйте метод Content/web\_purchase.

**URL-параметры запроса:**

* contentId — *number* — Идентификатор видеофильма. (Обязательный параметр.).
* contentPackageId — *number* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* contentUsageRuleId — *number* — Идентификатор правила покупки, по которому приобретается видеофильм полученный с помощью метода ServiceSpec/list\_terminal\_type\_rules, выходной параметр <cur\_id>. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* assetName — *string* — Название фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* notifyOSS — *number* — Флаг необходимости проведения покупки в АСР. 1 — провести покупку через АСР, выполнить запрос списания стоимости покупки со счета абонента в АСР. 0 — не проводить покупку через АСР, фиксировать в SDP факт оплаты покупки в сторонней системе. .
* transactionId — *integer* — Идентификатор транзакции. .
* profileId — *number* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /Series

#### /Series/web\_purchase

**GET-запрос**

Метод запрашивает разовую покупку серии сериала из библиотеки SmartTUBE SDP. Для прочего внешнего контента (софт, книги, и т.п.) используется метод Content/web\_purchase.

**URL-параметры запроса:**

* contentId — *number* — Идентификатор видеофильма. (Обязательный параметр.).
* contentPackageId — *number* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* contentUsageRuleId — *number* — Идентификатор правила покупки, по которому приобретается видеофильм полученный с помощью метода ServiceSpec/list\_terminal\_type\_rules, выходной параметр <cur\_id>. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* assetName — *string* — Название фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* notifyOSS — *number* — Флаг необходимости проведения покупки в АСР. 1 — провести покупку через АСР, выполнить запрос списания стоимости покупки со счета абонента в АСР. 0 — не проводить покупку через АСР, фиксировать в SDP факт оплаты покупки в сторонней системе. .
* transactionId — *integer* — Идентификатор транзакции. .
* profileId — *number* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /Content

Методы для приобретения контента и управления медиапозициями.

#### /Content/web\_purchase

**GET-запрос**

Метод запрашивает разовую покупку внешнего контента (софт, книги, и т.п.). Для покупок фильмов из библиотеки на платформе SmartTUBE SDP используется метод VideoMovie/web\_purchase.

**URL-параметры запроса:**

* contentId — *integer* — Идентификатор видеофильма. (Обязательный параметр.).
* name — *string* — Имя контента во внешней системе. (Обязательный параметр.).
* portalId — *number* — Идентификатор услуги портала в SDP, полученный методом ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* price — *integer* — Цена в копейках без разделителя. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* notifyOSS — *number* — Флаг необходимости проведения покупки в АСР. 1 — провести покупку через АСР, выполнить запрос списания стоимости покупки со счета абонента в АСР. 0 — не проводить покупку через АСР, фиксировать в SDP факт оплаты покупки в сторонней системе. .
* transactionId — *integer* — Идентификатор транзакции. .
* service\_id — *string* — Идентификатор услуги во внешней системе. .
* price\_category — *string* — Идентификатор тарифа. .
* vat — *string* — Признак НДС: 1 — Цена в параметре price указана с НДС; 2 — Цена в параметре price указана без НДС; 3 — Невозможно определить для данной транзакции. .
* product\_type — *string* — Идентификатор типа покупаемого продукта. .
* profileId — *number* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /Content/portal\_purchase

**GET-запрос**

Специальный метод для веб-портала для выполнения регулярных запросов списаний средств через SDP в АСР за пользование услугами. Запросы списаний фиксируются в SDP на веб-подключение указанной учетной записи (deviceType=WEB). Передавать ID сессии и ID ресурса в параметрах этого метода не нужно. Если покупка прошла успешно, то возвращается код 0. Если покупку выполнить не удалось, возвращаются ошибки с кодом 1 или 15.

**URL-параметры запроса:**

* contentId — *integer* — Идентификатор видеофильма. (Обязательный параметр.).
* name — *string* — Имя контента во внешней системе. (Обязательный параметр.).
* portalId — *number* — Идентификатор услуги портала в SDP, полученный методом ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* price — *integer* — Цена в копейках без разделителя. (Обязательный параметр.).
* notifyOSS — *number* — Флаг необходимости проведения покупки в АСР. 1 — провести покупку через АСР, выполнить запрос списания стоимости покупки со счета абонента в АСР. 0 — не проводить покупку через АСР, фиксировать в SDP факт оплаты покупки в сторонней системе. .
* transactionId — *integer* — Идентификатор транзакции. .
* service\_id — *string* — Идентификатор услуги во внешней системе. .
* price\_category — *string* — Идентификатор тарифа. .
* vat — *string* — Признак НДС: 1 — Цена в параметре price указана с НДС; 2 — Цена в параметре price указана без НДС; 3 — Невозможно определить для данной транзакции. .
* product\_type — *string* — Идентификатор типа покупаемого продукта. .
* device\_type — *string* — (Обязательный параметр.).
* profileId — *number* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /Content/mediaPositionList

**GET-запрос**

Метод возвращает список сохраненных позиций просмотра фильмов, роликов UGC, ТВ-каналов.

В результате выполнения запроса сервер возвращает список сохраненных позиций просмотра. Позиция имеет следующие параметры:

* key — ID контента
* type — тип позиции
* value — значение позиции
* mac — ID ресурса (значение resourceId)
* action — тип действия
* dt — дата-время в строковом формате, MM/DD/YYYY HH:MM:SS
* profileName — имя профиля

Типы позиций:

* VIDEOMOVIE — фильм, value (секунды) — смещение относительно начала файла.
* UGC — ролик UGC, value (секунды) — смещение относительно начала файла.
* CHANNEL — ТВ-канал, value (секунды) — время сохранения позиции просмотра канала (в рамках услуг Pause Live, TSTV, nPVR) в формате unixtime.

Типы действий:

* PAUSE — Пауза. Для ТВ-канала и фильма момент времени, в который пользователь нажал паузу. При наличии такой позиции в клиентском приложении нужно начать воспроизведение канала/фильма с этого момента.
* STOP — Стоп. Для ТВ-канала означает, что в клиентском приложении нужно начать воспроизведение «живой» трансляции (отменяет сохраненную позицию). Для фильма означает, что нужно начать воспроизведение с этого момента.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* profileName — *string* — Имя профиля. Если параметр не указан, позиция добавляется на текущий профиль. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <items>  
 <item>  
 <key> 123321 </key>  
 <type> VIDEOMOVIE </type>  
 <value> 20 </value>  
 <mac> 1751482113 </mac>  
 <action> PAUSE <action>  
 <dt> 04/26/2012 18:11:17 </dt>  
 <profileName> master </profileName>  
 </item>  
 <item>  
 <key> 123300 </key>  
 <type> CHANNEL </type>  
 <value> 1335370552 </value>  
 <mac> 1751482113 </mac>  
 <action> PAUSE <action>  
 <dt> 04/26/2012 18:20:17 </dt>  
 <profileName> master </profileName>  
 </item>  
 </items>  
</rows>

#### /Content/addMediaPosition

**GET-запрос**

Метод создает или обновляет медиапозицию.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* key — *string* — Идентификатор контента. (Обязательный параметр.).
* type — *string* — Тип медиапозиции. Возможные значения: VIDEOMOVIE, UGC, CHANNEL. (Обязательный параметр.).
* value — *string* — Значение в зависимости от типа позиции (см. метод Content/mediaPositionList). (Обязательный параметр.).
* action — *string* — Тип действия. Возможные значения: PAUSE, STOP, PLAY. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* profileName — *string* — Имя профиля. Если параметр не указан, позиция добавляется на текущий профиль. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

#### /Content/deleteMediaPosition

**GET-запрос**

Метод удаляет сохраненную позицию просмотра. Если из параметров key или type не задан хотя бы один или не заданы оба - удаляется вся история указанного профиля, если заданы оба - удаляются позиции по этому контенту.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* profileName — *string* — Имя профиля. Если параметр не указан, позиция добавляется на текущий профиль. .
* key — *string* — Идентификатор контента. (Обязательный параметр.).
* type — *string* — Тип медиапозиции. Возможные значения: VIDEOMOVIE, UGC, CHANNEL. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /Franchise

#### /Franchise/purchase\_no\_content

**GET-запрос**

Метод для покупки франшизы.

**URL-параметры запроса:**

* contentPackageId — *string* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* contentId — *string* — Идентификатор видеофильма. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* contentUsageRuleId — *string* — Идентификатор правила покупки, по которому приобретается видеофильм полученный с помощью метода ServiceSpec/list\_terminal\_type\_rules, выходной параметр <cur\_id>. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /ServiceUsage

Методы для получения информации о покупках.

#### /ServiceUsage/listByTransaction

**GET-запрос**

Метод проверяет статус покупок в АСР по идентификатору транзакции. Если транзакция с заданным transactionId не найдена в АСР, то сервер SDP возвращает пустую структуру .

**URL-параметры запроса:**

* transactionId — *integer* — Идентификатор транзакции. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <date> 06/18/2009 14:42:30 </date>  
 <result> 1 </result>  
 <comment> OK </comment>  
 <content\_id> 24141088 </content\_id>  
 </row> <row number="1">  
 <date> 06/18/2009 16:22:14 </date>  
 <result> -3 </result>  
 <comment> error in services </comment>  
 <content\_id> 535767 </content\_id>  
</row>  
</rows>

#### /ServiceUsage/list\_purchase

**GET-запрос**

Метод возвращает список покупок пользователя.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<date> Дата покупки в формате ММ/ДД/ГГГГ ЧЧ:ММ:СС </date>  
<unixDate> Дата покупки в Unix-time формате </unixDate>  
<su\_id> Идентификатор покупки </su\_id>  
<c\_id> Идентификатор контента <c\_id>  
<dsc> Название контента </dsc>  
<su\_discr>  
Тип покупки. Возможные значения: CONTENTSERVICEUSAGE BONUSSUBSCRIPTIONSERVICEUSAGE CREATECONTENTPAYMENT BINDSERVICEUSAGE BONUSSERVICEUSAGE NPVRRECORD FRANCHISEVODPURCHASE GIFTSERVICEUSAGE SUBSCRIPTIONSERVICEUSAGE ABONEMENTEDSERVICEUSAGE CREATEPAYMENT PROMOSERVICEUSAGE FRANCHISEPURCHASE UNSUBSCRIPTIONREQUEST  
</su\_discr>  
<doneeAccount> Номер учетной записи, на которую был отправлен подарок </doneeAccount>  
<doneeProfile> Идентификатор профиля учетной записи, на которую был отправлен подарок </doneeProfile>  
<s\_id> Идентификатор типа контента </s\_id>  
<s\_discr>  
Тип контента. Возможные значения: KARAOKEPACKAGE PPVCHANNELPACKAGEFIXEDPRICE FRANCHISEPACKAGE INTERNETRADIOCHANNEL CONTENTBUNDLEPACKAGE PPVCHANPACKFIXPRICEFORSUBSC RADIOCHANNEL DLNASERVICE CHANNELPACKAGE NPVRUNLIMCHANNELPACKAGE VODPACKAGE NPVRCHANNELPACKAGE PPVCHANNELPACKAGE ESTVODPACKAGE BUNDLEPACKAGE NVODCHANNEL PVRCHANNELPACKAGE RADIOCHANNELPACKAGE SERVICESPEC TSTVCHANNELPACKAGE INTERNETRADIOCHANNELPACKAGE KARAOKEABONEMENT RTUGCQUOTA SERIESPACKAGE ESTCONTENTBUNDLEPACKAGE ISSASERVICE NPVRADDLIMITPACKAGE DISCOUNTPACKAGE ACTIVATIONSERVICE BROADCASTCHANNEL PAUSELIFECHANNELPACKAGE VODABONEMENT EXTERNALSERVICE  
</s\_discr>  
<price> Цена в копейках </price>  
<paymentId> Идентификатор платежа по карте. Если поле пустое, покупка совершалась не по карте </paymentId>

### /ServiceSpec

#### /ServiceSpec/list\_terminal\_type\_rules

**GET-запрос**

Метод возвращает правила покупки для CoD-контента с учетом типа терминала. Если возвращаемый параметр cur\_vtl\_forever=1 — это значит, что срок действия купленного контента будет не ограничен, т.е. навсегда. Для определения цены контента следует выполнить следующее:

1. Получить список контента, например, VodPackage/list\_movies.
2. Получить значение <asset\_type> ассета фильма.
3. Получить список правил использования ServiceSpec/list\_terminal\_type\_rules для данного пакета.
4. Выбрать правило использования с <cur\_assetTypeExt> =<asset\_type> и нужным <cur\_externalId>=OTT\_WEB для портала или <cur\_externalId>=OTT\_IPAD.

Значение в поле <cur\_price> — стоимость контента для портала в копейках, для iPad — в центах.

**URL-параметры запроса:**

* packageId — *string* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. .
* locationId — *string* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* poId — *string* — Идентификатор тарифного плана. (Обязательный параметр.).
* terminalType — *string* — Внешний идентификатор типа устройства, для которого предназначена услуга. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows packageId="-1" code="0">  
 <row>  
 <id> 43785367 </id>  
 <cur\_id> 43249393 </cur\_id>  
 <cur\_name> EST-покупка на OTT-устройствах </cur\_name>  
 <cur\_externalId> OTT\_WEB </cur\_externalId>  
 <cur\_descr> покупка OTT-фильмов со всех OTT-устройств, использующих ServiceSpec/list\_terminal\_type\_rules </cur\_descr>  
 <cur\_assetType> 1722808 </cur\_assetType>  
 <cur\_assetTypeExt> 2377438 </cur\_assetTypeExt>  
 <cur\_default> 1 </cur\_default>  
 <cur\_vtl\_name> Без ограничений времени просмотра </cur\_vtl\_name>  
 <cur\_duration> 1080000000 </cur\_duration>  
 <cur\_vtl\_forever> 1 </cur\_vtl\_forever>  
 <cur\_price> 55000 </cur\_price>  
 <packageId> 21713916 </packageId>  
 </row>  
 <row>  
 <id> 42912663 </id>  
 <cur\_id> 1000067 </cur\_id>  
 <cur\_name> Покупка SD на 48 часов </cur\_name>  
 <cur\_externalId> SD48 </cur\_externalId>  
 <cur\_descr />  
 <cur\_assetType> 1000059 </cur\_assetType>  
 <cur\_assetTypeExt> 1000059 </cur\_assetTypeExt>  
 <cur\_default> 1 </cur\_default>  
 <cur\_vtl\_name> Покупка на 48 часов </cur\_vtl\_name>  
 <cur\_duration> 172800 </cur\_duration>  
 <cur\_vtl\_forever> 0 </cur\_vtl\_forever>   
 <cur\_price> 10077 </cur\_price>  
 <packageId> 1000 </packageId>  
 </row>  
</rows>

### /ott

Методы взаимодействия ОТТ-устройств с SDP.

#### /ott/get\_network\_config

**GET-запрос**

Запрос на получение данных о конфигурации сети перед авторизацией. Если по методу ott/get\_cur\_geolocation не найден текущий регион и нет домашнего региона, приложение должно спросить у пользователя регион и далее повторно отправить запрос ott/get\_network\_config с параметром curLocation. В случае ошибки – код ошибки и сообщение.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).
* curLocation — *string* — Идентификатор локации. Параметр отправляется в случае, если не определен текущий регион и нет домашнего региона. .

**Параметры ответа:**

Пример XML-данных ответа:

<data code="0">  
 <network>  
 <frontend>  
 <url> ${schema}://${curHost}/mrf-${homeMrf}/${prefix} </url>  
 </frontend>  
 <proxy>  
 <url> ${header\_PROXY} </url>  
 </proxy>  
 <homeMrf> ct </homeMrf>  
 <curMrf> ct </curMrf>  
 <curLocation> 100001 </curLocation>  
 <curSubLocation> 15266876 </curSubLocation>  
 <deviceType> OTTSTB </deviceType>  
 </network>  
</data>

#### /ott/ott\_authorize

**GET-запрос**

Метод производит обезличенную авторизацию по идентификатору устройства (uid), mac-адресу или идентификатору социальной сети.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).
* hash — *string* — Проверочный код при авторизации по uid. Формируется следующим образом: salt = skgwo2h8; hash = md5(salt + mac\_address). (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. .
* terminalName — *string* — Имя устройства. Сохраняется при каждой авторизации, если приложение его передает. Если не передает, поле будет заполняться значением из deviceType. .
* geoLocId — *string* — Геоидентификаторы. Сервер nginx использует данное значение при формировании запросов. .
* curLocation — *string* — Идентификатор локации. Параметр отправляется в случае, если не определен текущий регион и нет домашнего региона. .
* curSubLocation — *string* — Параметр, получаемый в методе ott/get\_network\_config. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 0928507149 </serviceAccountNumber>  
 <datetime> 08/25/2014 18:23:31 </datetime>  
 <state> ACTIVE </state>  
 <locationId> 100001 </locationId>  
 <subLocationId/>  
 <backend\_version> 2.7.4.11 </backend\_version>  
 <sessionId/>  
 <productOfferId> 2176407 </productOfferId>  
 <casId/>  
 <shouldUseRaptor> 0 </shouldUseRaptor>  
 <netLogIP> 85.94.1.25 </netLogIP>  
 <netLogPort> 5040 </netLogPort>  
 <providerId> 7379715 </providerId>  
 <terminalType> 11 </terminalType>  
 <time\_zone/>  
 <networkBlock> 1 </networkBlock>  
 <networkTypeId/>  
 <helpId/>  
 <smLogin> 1231233211 </smLogin>  
 <smPassword> 1231233211 </smPassword>  
 <esamservice/>  
 <randomservice/>  
 <accountType> unknown </accountType>  
 <offerId/>  
 <isOfferConfirmed> 0 </isOfferConfirmed>  
 <defaultUrlSource/>  
 <reason/>  
 <multiroom> true </multiroom>  
 <unixtime> 1408980402 </unixtime>  
 <resourceId> 1231233211 </resourceId>  
 <bonusType/>  
 <isTrustedDevice> 0 </isTrustedDevice>  
 <isOss> 0 </isOss>  
 <isHomeGroup> 1 </isHomeGroup>  
 <zabavaLogin/>>  
 <IP> 0:0:0:0:0:0:0:1 </IP>  
 <stbFunctions>  
 <id> 8000948 </id>  
 </stbFunctions>  
 <activationNumber> 0928507149 </activationNumber>  
 <pin> 8730 </pin>  
 <curLocation> 100100 </curLocation>  
 <curSubLocation> 100100 </curSubLocation>  
 </row>  
</rows>

#### /ott/sn\_authorize

**GET-запрос**

Обезличенная авторизация подключения в социальной сети. Если идентификатор в социальной сети найти не удается, метод возвращает ошибку OTT\_UID\_NOT\_FOUND.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).
* hash — *string* — Проверочный код при авторизации по uid. Формируется следующим образом: salt = skgwo2h8; hash = md5(salt + mac\_address). (Обязательный параметр.).
* deviceType — *string* — Тип пользовательского устройства. .
* terminalName — *string* — Имя устройства. Сохраняется при каждой авторизации, если приложение его передает. Если не передает, поле будет заполняться значением из deviceType. .
* geoLocId — *string* — Геоидентификаторы. Сервер nginx использует данное значение при формировании запросов. .
* curLocation — *string* — Идентификатор локации. Параметр отправляется в случае, если не определен текущий регион и нет домашнего региона. .
* curSubLocation — *string* — Параметр, получаемый в методе ott/get\_network\_config. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 2032540286 </serviceAccountNumber>  
 <datetime> 07/07/2014 15:43:23 </datetime>  
 <state> ACTIVE </state>  
 <locationId> 15266876 </locationId>  
 <subLocationId />  
 <backend\_version> 2.7.4.11 </backend\_version>  
 <sessionId />  
 <productOfferId> 1000073 </productOfferId>  
 <casId />  
 <shouldUseRaptor> 0 </shouldUseRaptor>  
 <netLogIP />  
 <netLogPort />  
 <providerId> 1000026 </providerId>  
 <terminalType> 0 </terminalType>  
 <time\_zone> +4 </time\_zone>  
 <networkBlock> 1 </networkBlock>  
 <networkTypeId />  
 <helpId> 19911944 </helpId>  
 <smLogin> 77457-45345-5577-384875 </smLogin>  
 <smPassword> 77457-45345-5577-384875 </smPassword>  
 <esamservice />  
 <randomservice />  
 <accountType> unknown </accountType>  
 <offerId />  
 <isOfferConfirmed> 0 </isOfferConfirmed>  
 <defaultUrlSource />  
 <reason />  
 <multiroom> false </multiroom>  
 <unixtime> 1404733399 </unixtime>  
 <resourceId> 77457-45345-5577-384875 </resourceId>  
 <bonusType />  
 <isTrustedDevice> 0 </isTrustedDevice>  
 <isOss> 0 </isOss>  
 <IP> 127.0.0.1 </IP>  
 <stbFunctions />  
 <activationNumber> 2032540286 </activationNumber>  
 <pin> 2988 </pin>  
 <curLocation> 100001 </curLocation>  
 <curSubLocation> 15266876 </curSubLocation>  
 </row>  
 </rows>

#### /ott/email\_authorize

**GET-запрос**

Авторизация по логину (адресу электронной почты) и паролю. В случае неудачи метод возвращает ошибки OTT\_EMAIL\_NOT\_FOUND или BIND\_SERVICE\_ACCOUNT\_WRONG\_PASSWORD.

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. (Обязательный параметр.).
* uId — *string* — Идентификатор устройства. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 93396470609 </serviceAccountNumber>  
 <secondaryServiceAccountNumber> 3396487754433 </secondaryServiceAccountNumber>  
 <datetime> 06/09/2014 15:06:40 </datetime>  
 <state> ACTIVE </state>  
 <locationId> 100007 </locationId>  
 <subLocationId> 100007 </subLocationId>  
 <activationNumber> 2019193522@IPAD </activationNumber>  
 <backend\_version> 2.7.4.11 </backend\_version>  
 <sessionId/>  
 <productOfferId> 1000073 </productOfferId>  
 <casId/>  
 <shouldUseRaptor> 0 </shouldUseRaptor>  
 <netLogIP> sdp.svc.iptv.rt.ru </netLogIP>  
 <netLogPort> 5040 </netLogPort>  
 <providerId> 1000026 </providerId>  
 <terminalType> 3 </terminalType>  
 <time\_zone/>  
 <networkBlock> 1 </networkBlock>  
 <networkTypeId/>  
 <helpId/>  
 <smLogin> 333444553335555 </smLogin>  
 <smPassword> 333444553335555 </smPassword>  
 <esamservice/>  
 <randomservice/>  
 <accountType> unknown </accountType>  
 <offerId/>  
 <isOfferConfirmed> 0 </isOfferConfirmed>  
 <defaultUrlSource/>  
 <reason/>  
 <multiroom> false </multiroom>  
 <unixtime> 1402315628 </unixtime>  
 <resourceId> 333444553335555 </resourceId>  
 <bonusType/>  
 <IP> 0:0:0:0:0:0:0:1 </IP>  
 <stbFunctions>  
 <id> 10199387 </id>  
 </stbFunctions>  
 <activationNumber/>  
 <pin> 000000 </pin>  
 <curLocation> 123 </curLocation>  
 <curSubLocation> 321 </curSubLocation>  
 <isTrustedDevice> 1 <isTrustedDevice>  
 </row>  
 </rows>

#### /ott/link\_email\_stb

**GET-запрос**

Метод отправляет номер и код активации, чтобы привязать к учетной записи аккаунт пользователя на сайте Zabava.ru. В случае неудачи метод возвращает следующие ошибки — OTT\_MRF\_NOT\_FOUND, OTT\_EMAIL\_NOT\_FOUND, OTT\_ACNUM\_NOT\_FOUND, OTT\_EMAIL\_REGISTERED.

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* activationNumber — *string* — Номер активации. (Обязательный параметр.).
* pin — *string* — Код активации. (Обязательный параметр.).
* uId — *string* — Идентификатор устройства. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/billing\_check\_number

**GET-запрос**

Метод отправляет запрос на подтверждение паспорта и ЛС пользователя.

**URL-параметры запроса:**

* account — *string* — Номер личного счета в АСР. (Обязательный параметр.).
* verifyCode — *string* — Проверочный код (последние 4 цифры паспорта). (Обязательный параметр.).
* login — *string* — Логин пользователя. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/link\_to\_group

**GET-запрос**

Метод привязывает устройство к группе. После успешной привязки на все устройства, подключенные к учетной записи, приходит control-сообщение о перезагрузке списка устройств — command?commandType=Control&commandName=reloadDeviceList получив его, устройство должно заново авторизоваться и проверить список услуг и покупок, так как они могу измениться. В случае неудачи метод возвращает ошибки “10200 — Группа не найдена или неверный пароль” или “10201 — Невозможно подключить больше устройств этого типа”.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).
* activationNumber — *string* — Номер активации. (Обязательный параметр.).
* pin — *string* — Код активации. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/link\_to\_group\_by\_email

**GET-запрос**

Метод привязывает устройство к группе по логину и паролю группы. В случае неудачи метод возвращает ошибки OTT\_EMAIL\_NOT\_FOUND (email не найден), BIND\_SERVICE\_ACCOUNT\_WRONG\_PASSWORD (неверный пароль), OTT\_NO\_GROUP\_FOUND (группа не найдена или неверный пароль), OTT\_DEVICE\_COUNT\_CONSTRAINT (невозможно подключить больше устройств указанного типа). После успешной привязки на все устройства, подключенные к учетной записи, приходит control-сообщение о перезагрузке списка устройств command?commandType=Control&commandName=reloadDeviceList получив его, устройство должно заново авторизоваться и проверить список услуг и покупок, так как они могу измениться.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).
* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/list\_group\_devices

**GET-запрос**

Метод возвращает список устройств группы.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <ID> 13141931 </ID>  
 <UID> 54:04:AE:FA:91:08 </UID>  
 <deviceType> STB </deviceType>  
 <terminalName/>  
 <isTrustedDevice> 1 </isTrustedDevice>  
</row>  
<row number="1">  
 <ID> 13141947 </ID>  
 <UID> 54:04:A6:F1:01:00 </UID>  
 <deviceType> OTTSTB </deviceType>  
 <terminalName />  
 <isTrustedDevice> 1 </isTrustedDevice>  
</row>  
<row number="2">  
 <ID> 13141939 </ID>  
 <UID> 83847-48834-65887-345877-45885 </UID>  
 <deviceType> ANDROID </deviceType>  
 <terminalName> Nexus X30 <terminalName/>  
 <isTrustedDevice> 0 </isTrustedDevice>  
</row>  
<row number="3">  
 <ID> 13141935 </ID>  
 <UID> VK\_123456789 </UID>  
 <deviceType> SNETWORK </deviceType>  
 <terminalName />  
 <isTrustedDevice> 0 </isTrustedDevice>  
</row>  
</rows>

#### /ott/get\_max\_device\_type\_count

**GET-запрос**

Возвращает максимальное количество устройств данного типа, которые можно добавить в одну группу.

**URL-параметры запроса:**

* deviceType — *string* — Тип пользовательского устройства. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <count> 4 </count>   
</rows>

#### /ott/unlink\_from\_group

**GET-запрос**

Метод удаляет устройство из группы.

При неудаче метод возвращает следующие ошибки:

<rows code="10202">  
 <message> Устройство не принадлежит группе </message>  
</rows>

<rows code="10203">  
 <message> Устройство нельзя удалить из этой группы, так как она является базовой для этого устройства </message>  
</rows>

В случае ошибки 10203 появляется сообщение «Приложение информирует Пользователя о том, что при удалении Устройства из Группы произойдет деактивация Устройства и для него будет недоступен приобретенный ранее Медийный контент».

Если пользователь подтверждает удаление, выполняется повторный вызов метода с добавлением параметра delete=true, после чего устройство удаляется с учетной записи. После удаления из группы на все устройства, подключенные к учетной записи, приходит control-сообщение о перезагрузке списка устройств:

command?commandType=Control&commandName=reloadDeviceList

Получив его, устройство должно перечитать список услуг и покупок, так как они могут измениться.

На удаляемое из группы устройство приходит сообщение:

command?commandType=Control&commandName=deletedFromGroup

**URL-параметры запроса:**

* ID — *string* — Идентификатор фильма. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* delete — *boolean* — .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/set\_trusted

**GET-запрос**

Установка или снятие признака доверенного устройства. Доверенные устройства могут совершать покупки через личный счет абонента, а недоверенные не могут. В случае неудачи метод возвращает ошибки OTT\_EMAIL\_NOT\_FOUND (email не найден), BIND\_SERVICE\_ACCOUNT\_WRONG\_PASSWORD (неверный пароль).

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* login — *string* — Логин пользователя. .
* password — *string* — Пароль для учетной записи пользователя. .
* isTrustedDevice — *number* — Признак того, что с устройства можно оплачивать покупки с личного счета, ассоциированного с учетной записью, без ввода пароля. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/link\_snetwork

**GET-запрос**

Привязка аккаунта социальной сети к учетной записи. После привязки соцсети на все устройства, подключенные к учетной записи, приходит control-сообщение о перезагрузке списка устройств command?commandType=Control&commandName=reloadDeviceList получив его, устройство должно проверить список услуг и покупок, так как они могу измениться. В случае неудачи метод возвращает ошибку OTT\_UID\_EXISTS\_ANOTHER\_MRF (Такой идентификатор уже зарегистрирован).

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* uId — *string* — Идентификатор устройства. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /ott/preregister\_email

**GET-запрос**

Метод предварительной регистрации почты. Метод проверяет email на уникальность, генерирует токен и отправляет его на указанный email. Шаблон письма с идентификатором “CREATE\_EMAIL\_TEMPLATE” создается в AdminUI. В случае неудачи метод возвращает ошибку OTT\_EMAIL\_ALREADY\_EXISTS (Данный email уже занят).

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/email\_check\_token

**GET-запрос**

Метод проверяет соответствие email и token. В случае неудачи метод возвращает ошибки OTT\_EMAIL\_NOT\_FOUND (email не найден), OTT\_EMAIL\_TOKEN\_ERROR (неверный token).

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* token — *string* — Токен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /ott/email\_set\_password

**GET-запрос**

Метод проверки пароля на соответствие заданным требованиям. В случае неудачи метод возвращает ошибку OTT\_EMAIL\_NOT\_FOUND (email не найден).

**URL-параметры запроса:**

* login — *string* — Логин пользователя. (Обязательный параметр.).
* password — *string* — Пароль для учетной записи пользователя. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /ott/get\_cur\_geolocation

**GET-запрос**

Метод получения текущей локации по геоидентификатору.

**URL-параметры запроса:**

* geoLocId — *string* — Геоидентификаторы. Сервер nginx использует данное значение при формировании запросов. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <curMrf> ct </curMrf>  
 <curLocation> 100001 </curLocation>  
 <curSubLocation> 100001 </curSubLocation>  
 </row>  
</rows>

#### /ott/update\_uid

**GET-запрос**

Метод для обновления uId устройства.

**URL-параметры запроса:**

* oldUID — *string* — Старый идентификатор устройства. (Обязательный параметр.).
* hash — *string* — Проверочный код при авторизации по uid. Формируется следующим образом: salt = skgwo2h8; hash = md5(salt + mac\_address). (Обязательный параметр.).
* newUID — *string* — Новый идентификатор устройства. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/get\_account\_additional\_billing\_info

**GET-запрос**

Метод возвращает персональные данные пользователя.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <surname> Иванов </surname>  
 <name> Иван </name>  
 <middleName> Иванович </middleName>  
 <sex> male </sex>  
 <birthday> 05/29/2005 </birthday>  
 <phoneNumber> 89053745638 </phoneNumber>  
 <address> г.Москва, улица Мира, дом 5 </address>  
 </row>  
</rows>

#### /ott/set\_account\_additional\_billing\_info

**GET-запрос**

Метод устанавливает персональные данные пользователя в биллинге.

**URL-параметры запроса:**

* uId — *string* — Идентификатор устройства. (Обязательный параметр.).
* surname — *string* — Фамилия. .
* name — *string* — Имя. .
* middleName — *string* — Отчество. .
* sex — *string* — .
* birthday — *string* — .
* phoneNumber — *string* — .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0" />

#### /ott/checkPayment

**GET-запрос**

Метод выполняет проверку возможности оплаты по банковской карте. В случае положительного ответа в клиентском приложении можно показывать экран оплаты.

Если оплата возможна, то сервер возвращает XML-структуру следующего вида:

<rows code="0">  
</rows>

В случае ошибки SDP метод возвращает следующие структуры:

Учетная запись не найдена:

<rows code="1006">  
 <message> Service account not found </message>  
</rows>

В справочнике регионов не определен параметр “Область именования для платежной системы”:

<rows code="10003">  
 <message> payTypeId on location is empty </message>  
</rows>

Если код ошибки равен единице (<rows code="1">) или принимает отрицательные значения, то произошла ошибка либо произошла в платежной системе, либо оплата по банковской карте невозможна. В данном случае в параметре <message> повторяется содержимое сообщения об ошибке, возвращаемого платежной системой.

Полный список кодов ошибок, возвращаемых платежной системой, см. в описания метода PaymentSystem/createPayment.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* svcTypeId — *string* — Область именования для платежной системы. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">

#### /ott/checkContentPayment

**GET-запрос**

Метод выполняет проверку возможности оплаты по банковской карте. В случае положительного ответа в клиентском приложении можно показывать экран оплаты. Цель использования данного метода – проверить, включен ли сервис оплаты произвольными банковскими картами.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /DeviceType

#### /DeviceType/list

**GET-запрос**

Метод возвращает список устройств, зарегистрированных на учетной записи.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <DeviceType\_externalId> STB </DeviceType\_externalId>   
 <DeviceType\_mrf> ct </DeviceType\_mrf>  
 <DeviceType\_ratingGroup> 0 </DeviceType\_ratingGroup>  
 <DeviceType\_isTrustedDevice />  
 <DeviceType\_ID> 12424620 </DeviceType\_ID>  
 <DeviceType\_DISCRIMINATOR> DEVICETYPE </DeviceType\_DISCRIMINATOR>  
 <DeviceType\_name> STB </DeviceType\_name>  
 <DeviceType\_description />  
 <DeviceType\_sortOrder> 1 </DeviceType\_sortOrder>  
 <DeviceType\_logo />  
 <DeviceType\_ownerUser />  
 <ownerUser\_login />  
 <ownerUser\_ID />  
 <ownerUser\_DISCRIMINATOR />  
 <DeviceType\_internalId />  
 <DeviceType\_langId> 8349690 </DeviceType\_langId>  
 <langId\_name> Русский </langId\_name>  
 <langId\_ID> 8349690 </langId\_ID>  
 <langId\_DISCRIMINATOR> LANGUAGE </langId\_DISCRIMINATOR>  
 </row>  
 <row number="13">  
 <DeviceType\_externalId> ANDROID </DeviceType\_externalId>  
 <DeviceType\_mrf> ct </DeviceType\_mrf>  
 <DeviceType\_ratingGroup />  
 <DeviceType\_isTrustedDevice />  
 <DeviceType\_ID> 28102331 </DeviceType\_ID>  
 <DeviceType\_DISCRIMINATOR> DEVICETYPE </DeviceType\_DISCRIMINATOR>  
 <DeviceType\_name> ANDROID </DeviceType\_name>  
 <DeviceType\_description />  
 <DeviceType\_sortOrder />  
 <DeviceType\_logo />  
 <DeviceType\_ownerUser />  
 <ownerUser\_login />  
 <ownerUser\_ID />  
 <ownerUser\_DISCRIMINATOR />  
 <DeviceType\_internalId />  
 <DeviceType\_langId />  
 <langId\_name />  
 <langId\_ID />  
 <langId\_DISCRIMINATOR />  
 </row>  
</rows>

### /OttSNetwork

#### /OttSNetwork/list

**GET-запрос**

Возвращает список идентификаторов соцсетей для OTT.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <OttSNetwork\_name> Facebook </OttSNetwork\_name>  
 <OttSNetwork\_externalId> FBK </OttSNetwork\_externalId>  
 <OttSNetwork\_ID> 28102322 </OttSNetwork\_ID>  
 <OttSNetwork\_DISCRIMINATOR> OTTSNETWORK </OttSNetwork\_DISCRIMINATOR>  
 </row>  
 <row number="1">  
 <OttSNetwork\_name> ВКонтакте </OttSNetwork\_name>  
 <OttSNetwork\_externalId> VK </OttSNetwork\_externalId>  
 <OttSNetwork\_ID> 28102323 </OttSNetwork\_ID>  
 <OttSNetwork\_DISCRIMINATOR> OTTSNETWORK </OttSNetwork\_DISCRIMINATOR>  
 </row>  
 <row number="2">  
 <OttSNetwork\_name> Одноклассники </OttSNetwork\_name>  
 <OttSNetwork\_externalId> ODN </OttSNetwork\_externalId>  
 <OttSNetwork\_ID> 28102324 </OttSNetwork\_ID>  
 <OttSNetwork\_DISCRIMINATOR> OTTSNETWORK </OttSNetwork\_DISCRIMINATOR>  
 </row>  
 <row number="3">  
 <OttSNetwork\_name> МойМир@mail.ru </OttSNetwork\_name>  
 <OttSNetwork\_externalId> MAIL </OttSNetwork\_externalId>  
 <OttSNetwork\_ID> 28102325 </OttSNetwork\_ID>  
 <OttSNetwork\_DISCRIMINATOR> OTTSNETWORK </OttSNetwork\_DISCRIMINATOR>  
 </row>  
 <row number="4">  
 <OttSNetwork\_name> Twitter </OttSNetwork\_name>  
 <OttSNetwork\_externalId> TWT </OttSNetwork\_externalId>  
 <OttSNetwork\_ID> 28102326 </OttSNetwork\_ID>  
 <OttSNetwork\_DISCRIMINATOR> OTTSNETWORK </OttSNetwork\_DISCRIMINATOR>  
 </row>  
 <row number="5">  
 <OttSNetwork\_name> Google+ </OttSNetwork\_name>  
 <OttSNetwork\_externalId> PLUS </OttSNetwork\_externalId>  
 <OttSNetwork\_ID> 28102327 </OttSNetwork\_ID>  
 <OttSNetwork\_DISCRIMINATOR> OTTSNETWORK </OttSNetwork\_DISCRIMINATOR>  
 </row>  
</rows>

### /GenreDictionary

#### /GenreDictionary/list

**GET-запрос**

Метод запрашивает список всех жанров видеофильмов, определенных в базе данных платформы. Для удобства пользователя в меню приложения все видеофильмы сгруппированы по жанрам.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows version="0" code="0">  
 <row number="0">  
 <GenreDictionary\_externalId> Фильмы </GenreDictionary\_externalId>  
 <GenreDictionary\_isVirtualCity> 0 </GenreDictionary\_isVirtualCity>  
 <GenreDictionary\_parent/>  
 <parent\_name/>  
 <parent\_ID/>  
 <parent\_DISCRIMINATOR/>  
 <GenreDictionary\_ID> 11646418 </GenreDictionary\_ID>  
 <GenreDictionary\_DISCRIMINATOR> GENREDICTIONARY </GenreDictionary\_DISCRIMINATOR>  
 <GenreDictionary\_name> Комедия </GenreDictionary\_name>  
 <GenreDictionary\_description/>  
 <GenreDictionary\_sortOrder> 1 </GenreDictionary\_sortOrder>  
 <GenreDictionary\_logo/>  
 <GenreDictionary\_isFinal> 1 </GenreDictionary\_isFinal>  
 </row>  
 <row number="1">  
 ...  
 </row>  
</rows>

### /KaraokePackage

Методы для работы с караоке.

#### /KaraokePackage/search

**GET-запрос**

Метод выполняет поиск караоке-роликов по первым буквам названия песни или исполнителя.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* text — *string* — Текст для поиска караоке-роликов по первым буквам названия песни или исполнителя. Текст для поиска должен содержать не менее 3-х символов и не иметь пробелов. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<karaoke\_list code="0">  
 <karaokes packageId="9966,9977" locationId="100001" version="1383821023">  
 <k>  
 <id> 15353622 </id>  
 <name> BUONASERA </name>  
 <sDate> 1378976413 </sDate>  
 <eDate> 1410771614 </eDate>  
 <version> 1383135065 </version>  
 <performerId> 15353617 </performerId>  
 <performer> ADRIANO CELENTANO </performer>  
 <packages>  
 <id> 32112122 </id>  
 <id> 9977 </id>  
 </packages>  
 <genres>  
 <id> 15353620 </id>  
 <id> 15389507 </id>  
 </genres>  
 <assets>  
 <content\_asset>  
 <id> 15379756 </id>  
 <ifn> bounasera.ts </ifn>  
 <duration/>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 0 </is\_crypted>  
 <sm\_externalid/>  
 <transport> rtsp </transport>  
 <audioPIDs/>  
 </content\_asset>  
 </assets>  
 </k>

#### /KaraokePackage/count

**GET-запрос**

Метод возвращает количество караоке-роликов, принадлежащих какому-либо исполнителю или жанру.

В случае подсчета количества караоке-роликов, принадлежащих какому-либо исполнителю, в нормальном режиме сервер возвращает XML-структуру вида:

Метод возвращает количество караоке-роликов, принадлежащих какому-либо исполнителю или жанру.

В случае подсчета количества караоке-роликов, принадлежащих какому-либо исполнителю, в нормальном режиме сервер возвращает XML-структуру вида:

<count code="0" packageId="18736162,23102013" locationId="100001" version="1383840837">  
<performers>  
 <p>  
 <id> 15401036</id> <!—идентификатор исполнителя караоке-ролика-->  
 <cnt> 1 </cnt>  
 </p>

В случае подсчета количества караоке-роликов, принадлежащих какому-либо жанру, в нормальном режиме сервер возвращает XML-структуру вида:

<count code="0" packageId="18736162" locationId="100001" version="1383840837">  
 <genres>  
 <g>  
 <id> -1</id> <!—выводится первым, обозначает общее количество разных караоке-роликов в пакетах-->  
 <cnt> 2058 </cnt>  
 </g>  
 <g>  
 <id> 15398950</id> <!-—идентификатор жанра-->  
 <cnt> 2</cnt> <!-—количество караоке-роликов, принадлежащих данному жанру-->  
 </g>  
 <g>  
 <id> 15396784 </id>  
 <cnt> 109 </cnt>  
 </g>

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* parameter — *string* — Параметр может принимать следующие значения: performen — в этом случае метод возвращает количество караоке-роликов, принадлежащих какому-либо исполнителю; genre — в этом случае метод возвращает количество караоке-роликов, принадлежащих какому-либо жанру. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<count code="0" packageId="18736162,23102013" locationId="100001" version="1383840837">  
 <performers>  
 <p>  
 <id> 15401036</id> <!—-идентификатор исполнителя караоке-ролика-->  
 <cnt> 1 </cnt>  
 </p>

В случае подсчета количества караоке-роликов, принадлежащих какому-либо жанру, в нормальном режиме сервер возвращает XML-структуру вида:

<count code="0" packageId="18736162" locationId="100001" version="1383840837">  
 <genres>  
 <g>  
 <id> -1 </id>  
 <cnt> 2058 </cnt>  
 </g>  
 <g>  
 <id> 15398950</id> <!—идентификатор жанра-->  
 <cnt> 2</cnt> <!—количество караоке-роликов, принадлежащих данному жанру-->  
 </g>  
 <g>  
 <id> 15396784 </id>  
 <cnt> 109 </cnt>  
 </g>

#### /KaraokePackage/list\_karaoke

**GET-запрос**

Метод запрашивает список караоке-роликов в пакете, принадлежащих какому-либо исполнителю или жанру.

**URL-параметры запроса:**

* packageId — *number* — Идентификатор пакета видеофильмов, полученный методами ServiceAccount/list\_services или ProductOffer/list\_services\_terminal. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* performerId — *number* — Идентификатор исполнителя караоке-ролика. (Обязательный параметр.).
* genreId — *number* — Идентификатор жанра фильма, получаемый методом GenreDictionary/list. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<karaoke\_list code="0">  
 <karaokes packageId="18736162" locationId="100001" performerId="-1" genreId="15353581" deviceType="" version="1383844803">  
 <k>  
 <id> 15369396 </id>  
 <name> Megamix 2001 (радио версия) </name>  
 <sDate/>  
 <eDate/>  
 <version> 1382988488 </version>  
 <performerId> 15369107 </performerId>  
 <performer> Вирус </performer>  
 <packages>  
 <id> 18736162 </id>  
 </packages>  
 <genres>  
 <id> 15353581 </id>  
 </genres>  
 <assets>  
 <content\_asset>  
 <id> 15369399 </id>  
 <ifn> Megamix2001.ts </ifn>  
 <duration/>  
 <asset\_type> 1000059 </asset\_type>  
 <is\_crypted> 0 </is\_crypted>  
 <sm\_externalid/>  
 <transport> rtsp </transport>  
 <audioPIDs/>  
 </content\_asset>  
 </assets>  
 </k>

### /FranchisePackage

#### /FranchisePackage/list\_package

**GET-запрос**

Метод запрашивает список франшиз, входящих в пакет.

**URL-параметры запроса:**

* franchisePackageId — *number* — Идентификатор пакета франшиз. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<franchise\_list code="0">  
 <franchises count="1">  
 <franchise>  
 <id> 44487937 </id>  
 <name> test\_fr </name>  
 <usageRule> 1000236 </usageRule>  
 <duration> 172800 </duration>  
 <forever> 0 </forever>  
 <dsc/>  
 <movies>  
 <id> 43510930 </id>  
 <id> 36196750 </id>  
 </movies>  
 <packages>  
 <id> 45299314 </id>  
 <id> 44691681 </id>  
 </packages>  
 </franchise>  
 </franchises>  
</franchise\_list>

### /PerformerDictionary

#### /PerformerDictionary/list

**GET-запрос**

Метод запрашивает список исполнителей караоке-роликов, существующих в SDP.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <PerformerDictionary\_ID> 15369107 </PerformerDictionary\_ID>  
 <PerformerDictionary\_DISCRIMINATOR> PERFORMERDICTIONARY </PerformerDictionary\_DISCRIMINATOR>  
 <PerformerDictionary\_name> Вирус </PerformerDictionary\_name>  
 <PerformerDictionary\_description/>  
 <PerformerDictionary\_sortOrder> 1 </PerformerDictionary\_sortOrder>  
 <PerformerDictionary\_logo/>  
 <PerformerDictionary\_ownerUser/>  
 <ownerUser\_login/>  
 <ownerUser\_ID/>  
 <ownerUser\_DISCRIMINATOR/>  
 <PerformerDictionary\_internalId/>  
 <PerformerDictionary\_externalId/>  
 </row>

### /KaraokeGenreDictionary

#### /KaraokeGenreDictionary/list

**GET-запрос**

Метод возвращает список жанров караоке-роликов, существующих в SDP.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows version="0" code="0">  
 <row number="0">  
 <KaraokeGenreDictionary\_externalId> Техно </KaraokeGenreDictionary\_externalId>  
 <KaraokeGenreDictionary\_ID> 15353581 </KaraokeGenreDictionary\_ID>  
 <KaraokeGenreDictionary\_DISCRIMINATOR> KARAOKEGENREDICTIONARY </KaraokeGenreDictionary\_DISCRIMINATOR>  
 <KaraokeGenreDictionary\_name> Техно </KaraokeGenreDictionary\_name>  
 <KaraokeGenreDictionary\_description/>  
 <KaraokeGenreDictionary\_sortOrder> 2 </KaraokeGenreDictionary\_sortOrder>  
 <KaraokeGenreDictionary\_logo/>  
 <KaraokeGenreDictionary\_ownerUser/>  
 <ownerUser\_login/>  
 <ownerUser\_ID/>  
 <ownerUser\_DISCRIMINATOR/>  
 <KaraokeGenreDictionary\_internalId/>  
 </row>

### /MovieFolder

Методы для получения информации о группировке фильмов.

#### /MovieFolder/list\_movie\_folders

**GET-запрос**

Метод возвращает список папок для группировки фильмов в том случае, если они находятся в разделах Видеопрокат – Подписки, Мое - Подписки или Видеопрокат - Мои покупки - Подписки.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<folders\_list code="0">  
 <folders count="1">  
 <folder>  
 <id> 46959338 </id>  
 <name> oii6799568 </name>  
 <logo> vod1434627544960.png </logo>  
 <year />  
 <mainGenre />  
 <ageLevel> 11154704 </ageLevel>  
 <sDate> 1434574800 </sDate>  
 <eDate> 1434661199 </eDate>  
 <rating> 0 </rating>  
 <al> 3 </al>  
 <country> Италия,Франция </country>  
 <genres>  
 <id> 10580 </id>  
 <id> 42155498 </id>  
 </genres>  
 <childMovies>  
 <id> 1000115 </id>  
 <id> 44704636 </id>  
 <id> 46694764 </id>  
 </childMovies>  
 </folder>  
 </folders>  
</folders\_list>

#### /MovieFolder/get\_dsc

**GET-запрос**

Метод возвращает описание папки для группировки фильмов в том случае, если они находятся в разделах Видеопрокат – Подписки, Мое - Подписки или Видеопрокат - Мои покупки - Подписки.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <content>  
 <id> 47164336 </id>  
 <description> text </description>  
 </content>  
</rows>

### /ChannelSubject

#### /ChannelSubject/list

**GET-запрос**

Метод запрашивает список тематик ТВ-каналов, определенных в базе данных платформы.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows version="0" code="0">  
 <row number="0">  
 <ChannelSubject\_ID> 47702 </ChannelSubject\_ID>  
 <ChannelSubject\_DISCRIMINATOR> CHANNELSUBJECT </ChannelSubject\_DISCRIMINATOR>  
 <ChannelSubject\_name> Movie </ChannelSubject\_name>  
 <ChannelSubject\_description/>  
 <ChannelSubject\_sortOrder> 1 </ChannelSubject\_sortOrder>  
 <ChannelSubject\_logo/>  
 </row>  
 <row number="1">  
 ...  
 </row>  
 </row>  
</rows>

### /ChannelPackage

Методы для работы с пакетами ТВ-каналов.

#### /ChannelPackage/list\_channels\_ott

**GET-запрос**

Метод запрашивает список ТВ-каналов с соответствующей программой передач по идентификатору пакета ТВ-каналов.

**URL-параметры запроса:**

* channelPackageId — *number* — Идентификатор пакета ТВ-каналов. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<channels\_list code="0">  
 <channels channelPackageId="1716760" locationId="100001" subLocationId="-1" version="1411390839">  
 <channel>  
 <bcid> 21458137 </bcid>  
 <bc\_r\_id> 21458137 </bc\_r\_id>  
 <bcname> AMEDIA 1 HD </bcname>  
 <bcdesc> «AMEDIA 1 HD» - это новый канал российской кинокомпании Amedia. Смотрите самые рейтинговые мировые сериалы – от полюбившейся классики – «Секс в большом городе», «Мост», до новых премьер в России - «Красавица и чудовище», «Вегас», а также самые интересные фильмы круглосуточно в качестве высокой четкости. </bcdesc>  
 <url> igmp://233.3.4.234:5000 </url>  
 <hqUrl/>  
 <pipUrl/>  
 <plcUrl/>  
 <backupUrl1/>  
 <backupUrl2/>  
 <logo> broadcastChannel1377691408478.png </logo>  
 <logo2> broadcastChannel1377691414363.png </logo2>  
 <bcal> 3 </bcal>  
 <num> 930 </num>  
 <bceid> 2049 </bceid>  
 <is\_crypted> 1 </is\_crypted>  
 <isOttEncrypted> 0 </isOttEncrypted>  
 <isDvrCrypted/>  
 <soundVolume> 100 </soundVolume>  
 <raptorPort> 0 </raptorPort>  
 <isErotic/>  
 <streamAspectRatio> 0 </streamAspectRatio>  
 <version> 1404442937 </version>  
 <zoomRatio> .5 </zoomRatio>  
 <ottURL/>  
 <smlOttURL/>  
 <ottDvr> 0 </ottDvr>  
 <tstvOttURL/>  
 <plOttURL/>  
 <dvbtChannelName/>  
 <poster> broadcastChannelPoster1377691418118.png </poster>  
 <nPVRChannelID> CH\_C01\_HDAMEDIA </nPVRChannelID>  
 <isQualityMonitoring> 1 </isQualityMonitoring>  
 <isTestStreamQuality> 0 </isTestStreamQuality>  
 <isBarker> 0 </isBarker>  
 <promo\_url/>  
 <videoServerProtocol/>  
 <subjects>  
 <id> 2331473 </id>  
 </subjects>  
 <packages>  
 <id> 37494339 </id>  
 <id> 11818722 </id>  
 <id> 1716760 </id>  
 <id> 1529084 </id>  
 </packages>  
 <loc>  
 <id> 100904 </id>  
 <id> 100001 </id>  
 <id> 100002 </id>  
 <id> 100009 </id>  
 <id> 100003 </id>  
 <id> 100004 </id>  
 <id> 100010 </id>  
 <id> 100011 </id>  
 <id> 100012 </id>  
 <id> 100013 </id>  
 <id> 100014 </id>  
 <id> 100015 </id>  
 <id> 100016 </id>  
 <id> 100017 </id>  
 <id> 100005 </id>  
 <id> 100006 </id>  
 <id> 100007 </id>  
 <id> 100008 </id>  
 <id> 100902 </id>  
 </loc>  
 <excl/>  
 <stbFunctions/>  
 <networkTypes/>  
 <audioPIDs/>  
 <subtitlePIDs/>  
 <urls>  
 <location>  
 <id> 100001 </id>  
 <url> igmp://233.3.4.234:5000 </url>  
 <hqUrl> igmp://233.3.3.149:5000 </hqUrl>  
 <nPVRChannelID/>  
 <pipUrl/>  
 <plcUrl/>  
 <backupUrl1/>  
 <backupUrl2/>  
 </location>  
 <location>  
 <id> 100902 </id>  
 <url> rtsp://85.94.1.15:50000/udp://239.1.24.8:5000 </url>  
 <hqUrl/>  
 <nPVRChannelID/>  
 <pipUrl/>  
 <plcUrl/>  
 <backupUrl1/>  
 <backupUrl2/>  
 </location>  
 </urls>  
 <ott\_urls/>  
 <dvbUrls/>  
 </channel>  
 <channel>

#### /ChannelPackage/list\_incompatible\_packages

**GET-запрос**

Метод запрашивает список несовместимых друг с другом пакетов ТВ-каналов. Сервер SDP возвращает матрицу несовместимости, логика работы с которой инкапсулирована в клиентское приложение. Например, в клиентском приложении может быть запрещена одновременная подписка на два несовместимых друг с другом пакета. Несовместимость используется для управления подпиской на похожие по контенту пакеты услуг.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<packages code="0">  
 <package>  
 <id> 57065 </id>  
 <incompatible\_packages>  
 <id> 482263 </id>  
 <id> 482294 </id>  
 ...  
 <id> 1663073 </id>  
 </incompatible\_packages>  
 </package>  
 <package>  
 …  
 </package>  
</packages>

#### /ChannelPackage/simple\_list?field=internalId

**GET-запрос**

Метод возвращает внутренний Id пакета ТВ-каналов из SDP МРФЦ по единому для всех МРФ значению portalId.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <ChannelPackage\_internalId> 1716757 </ChannelPackage\_internalId>  
 </row>  
</rows>

### /ProgramGenre

#### /ProgramGenre/list

**GET-запрос**

Метод возвращает список жанров ТВ-передач, существующих в SDP.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <ProgramGenre\_externalId> </ProgramGenre\_externalId>  
 <ProgramGenre\_ID> 24993802 </ProgramGenre\_ID>  
 <ProgramGenre\_DISCRIMINATOR> PROGRAMGENRE </ProgramGenre\_DISCRIMINATOR>  
 <ProgramGenre\_name> Концерт </ProgramGenre\_name>  
 <ProgramGenre\_description> </ProgramGenre\_description>  
 <ProgramGenre\_sortOrder> 16 </ProgramGenre\_sortOrder>  
 <ProgramGenre\_logo> </ProgramGenre\_logo>  
 </row>  
 <row number="1">  
 …  
 </row>  
</rows>

### /EPG

Методы для работы с электронной программой передач.

#### /EPG/get\_by\_Chnnl

**GET-запрос**

Метод возвращает список программ передач на дату для заданного ТВ-канала. Обязательно передавать один из параметров – startUnixDt или day.

**URL-параметры запроса:**

* channelId — *number* — Идентификатор пакета ТВ-каналов. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* day — *date* — Дата, на которую требуется программа передач. Если передать текущую дату и время, то в ответе придет текущая передача плюс указанное в параметре to количество последующих. (Обязательный параметр.).
* startUnixDt — *date* — Дата, на которую требуется программа передач. Если передать текущую дату и время, то в ответе придет текущая передача плюс указанное в параметре to количество последующих. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<epg\_list code="0">  
 <epg>  
 <p>  
 <sdate> 1321477200 </sdate>  
 <fdate> 1321477200 </fdate>  
 <c\_id> 1000530 </c\_id>  
 <id> 8755794 </id>  
 <name> Колыбельная для мужчин </name>  
 <discr> PROGRAM </discr>  
 <desc>  
 Мелодрама по рассказу Ю. Яковлева "Девушка из Бреста".Подобно многим женщинам послевоенного времени, Клавдия одна вырастила сына. Они были в </desc>  
 <clogo/>  
 <al> 2 </al>  
 <ir> 0 </ir>  
 <recomW> 0 </recomW>  
 <s\_id> 8755921 </s\_id>  
 <eid> 86225641 </eid>  
 <bundle> Для мужчин </bundle>  
 <genre> 10920257 </genre>  
 <dkey> 11/17/2011 </dkey>  
 <loc> -1 </loc>  
 <version> 1321401600 </version>  
 <imdbR> 8.5 </imdbR>  
 <kinopR> 8.2 </kinopR>  
 <pkg>  
 <id> 1000025 </id>  
 </pkg>  
 </p>  
 …  
 </epg>  
 </epg\_list>

#### /EPG/get\_by\_pkg

**GET-запрос**

Метод возвращает программу передач на ТВ-каналах на дату для заданного тарифного плана.

**URL-параметры запроса:**

* poId — *number* — Идентификатор тарифного плана. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* day — *date* — Дата, на которую требуется программа передач. Если передать текущую дату и время, то в ответе придет текущая передача плюс указанное в параметре to количество последующих. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. (Обязательный параметр.).
* To — *number* — Номер последней страницы списка контента. (Обязательный параметр.).
* ID — *string* — Идентификатор фильма. .

**Параметры ответа:**

Пример XML-данных ответа:

<epg\_list code="0">  
 <epg poId="1000025" locationId="110100" day="11/17/2011" version="1335916800">  
 <p>  
 ...  
 </p>  
 ...  
 </epg>  
</epg\_list>

#### /EPG/get\_delta

**GET-запрос**

Метод для получения дельты.

**URL-параметры запроса:**

* poId — *number* — Идентификатор тарифного плана. (Обязательный параметр.).
* version — *date* — Версия пакета видеофильмов, полученная последним выводом метода VodPackage/list\_movies. Атрибут version элемента <films>. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<delta code="0">  
 <add> 8754940,8758942,8764991 </add>  
 <del> 8761663,8761662,8761661 </del>  
</delta>

#### /EPG/search

**GET-запрос**

Метод выполняет поиск программы передач по названию или идентификатору сериала.

**URL-параметры запроса:**

* text — *string* — Текст для поиска караоке-роликов по первым буквам названия песни или исполнителя. Текст для поиска должен содержать не менее 3-х символов и не иметь пробелов. (Обязательный параметр.).
* series — *number* — Идентификатор сериала, если для данного сезона в SDP указано, к какому сериалу он относится, для поиска программы передач по идентификатору сериала. (Обязательный параметр.).
* channelId — *number* — Идентификатор пакета ТВ-каналов. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* From — *number* — Номер начальной страницы списка контента, начиная с нуля. .
* To — *number* — Номер последней страницы списка контента. .

**Параметры ответа:**

Пример XML-данных ответа:

<epg\_list code="0">  
 <epg poId="1000260" locationId="-1" text="ПУАРО А" series="null" version="1417713306">  
 <p>  
 <sdate> 1417374600 </sdate>  
 <fdate> 1417381800 </fdate>  
 <c\_id> 4739225 </c\_id>  
 <id> 30247210 </id>  
 <name> Т/С. «Пуаро Агаты Кристи». </name>  
 <discr> EPG </discr>  
 <desc> На художественном аукционе Пуаро решает приобрести старинное зеркало. Однако его цену перебивает антиквар Жерваз Шевени, который и становится владельцем вещи. После аукциона Шевени предлагает сыщику зеркало в обмен на расследование одного дела. </desc>  
 <clogo> xls\_imp\_stv\_epggenid\_4705ebc6-1835-4f2c-90.jpg </clogo>  
 <al> 52653 </al>  
 <s\_id> 30247211 </s\_id>  
 <bundle> Agatha Christie's Poirot </bundle>  
 <genre> 1000466 </genre>  
 <imdbR> 8.6 </imdbR>  
 <kinopR> 8.206 </kinopR>  
 <tstvAllowed> 1 </tstvAllowed>  
 <plAllowed> 1 </plAllowed>  
 <seriesNumber> 19734878 </seriesNumber>  
 <year> 1989 </year>  
 <country> 1003442 </country>  
 <category> 22754734 </category>  
 <genreList> 1000489,1000472 </genreList>  
 <live> 0 </live>  
 <repeat> 0 </repeat>  
 <fs> 0 </fs>  
 <ls> 0 </ls>  
 <free> 0 </free>  
 <oName> Agatha Christie's Poirot </oName>  
 <eName> «Коробка шоколадных конфет,Зеркало мертвеца». </eName>  
 <actors> Дэвид Суше, Хью Фрейзер, Филип Джексон, Полин Моран, Дэвид Йелленд, Ричард Бебб, Зоэ Уонамейкер, Джордж Литтл, Тони Ред Ричардс, Лоуренс Ричардсон </actors>  
 <director> Эдвард Беннет, Ренни Рай, Эндрю Грив </director>  
 <screenplay> Агата Кристи, Клайв Экстон, Энтони Хоровиц </screenplay>  
 <operator> Крис О'Делл, Айвен Страсберг, Норман Дж. Лэнгли </operator>  
 <composer> Кристофер Ганнинг, Стефен МакКеон, Кристиан Хенсон </composer>  
 <producer> Брайан Истмэн, Ник Эллиотт, Мишель Бак </producer>  
 </p>  
 </epg>  
</epg\_list>

### /ContentBundleGenre

#### /ContentBundleGenre/list

**GET-запрос**

Метод возвращает список жанров сезонов и сериалов.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <ContentBundleGenre\_ID> 15470871 </ContentBundleGenre\_ID>  
 <ContentBundleGenre\_name> Все </ContentBundleGenre\_name>  
 <ContentBundleGenre\_description> </ContentBundleGenre\_description>  
 </row>  
 ...  
</rows>

### /AccessLevelDictionary

#### /AccessLevelDictionary/list

**GET-запрос**

Метод запрашивает список всех уровней доступа, определенных в базе данных платформы.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <AccessLevelDictionary\_ID> 2 </AccessLevelDictionary\_ID>  
 <AccessLevelDictionary\_DISCRIMINATOR> ACCESSLEVELDICTIONARY </AccessLevelDictionary\_DISCRIMINATOR>  
 <AccessLevelDictionary\_name> +3 </AccessLevelDictionary\_name>  
 <AccessLevelDictionary\_description/>  
 <AccessLevelDictionary\_sortOrder> 1 </AccessLevelDictionary\_sortOrder>  
 <AccessLevelDictionary\_logo/>  
 </row>  
 <row number="1">  
 …  
 </row>  
</rows>

### /AudioPID

#### /AudioPID/list

**GET-запрос**

Метод запрашивает список аудио PID, существующих в системе.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <AudioPID\_name> rus </AudioPID\_name>  
 <AudioPID\_PID> 2061 </AudioPID\_PID>  
 <AudioPID\_broadcastChannelID> 102148 </AudioPID\_broadcastChannelID>  
 <broadcastChannelID\_name> Euronews </broadcastChannelID\_name>  
 <broadcastChannelID\_ID> 102148 </broadcastChannelID\_ID>  
 <broadcastChannelID\_DISCRIMINATOR> BROADCASTCHANNEL </broadcastChannelID\_DISCRIMINATOR>  
 <AudioPID\_ID> 11540638 </AudioPID\_ID>  
 <AudioPID\_DISCRIMINATOR> AUDIOPID </AudioPID\_DISCRIMINATOR>  
 <AudioPID\_audioPIDType> </AudioPID\_audioPIDType>  
 <audioPIDType\_name> </audioPIDType\_name>  
 <audioPIDType\_externalId> </audioPIDType\_externalId>  
 <audioPIDType\_ID> </audioPIDType\_ID>  
 <audioPIDType\_DISCRIMINATOR> </audioPIDType\_DISCRIMINATOR>  
 </row>  
 <row number="1">  
 ...  
 </row>  
</rows>

### /AssetQuality

#### /AssetQuality/list

**GET-запрос**

Метод запрашивает справочник причин плохого качества ассетов.

**URL-параметры запроса:**

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <AssetQuality\_externalId> BADSOUND </AssetQuality\_externalId>  
 <AssetQuality\_ID> 31618831 </AssetQuality\_ID>  
 <AssetQuality\_DISCRIMINATOR> ASSETQUALITY </AssetQuality\_DISCRIMINATOR>  
 <AssetQuality\_name> Плохой звук / Нет звука </AssetQuality\_name>  
 <AssetQuality\_description />  
 <AssetQuality\_sortOrder />  
 <AssetQuality\_logo />  
 <AssetQuality\_ownerUser />  
 <ownerUser\_login />  
 <ownerUser\_ID />  
 <ownerUser\_DISCRIMINATOR />  
 <AssetQuality\_internalId />  
 <AssetQuality\_langId />  
 <langId\_name />  
 <langId\_ID />  
 <langId\_DISCRIMINATOR />  
 </row>  
 <row number="1">  
 <AssetQuality\_externalId> BADPICTURE </AssetQuality\_externalId>  
 <AssetQuality\_ID> 31618832 </AssetQuality\_ID>  
 <AssetQuality\_DISCRIMINATOR> ASSETQUALITY </AssetQuality\_DISCRIMINATOR>  
 <AssetQuality\_name> Плохое качество изображения </AssetQuality\_name>  
 <AssetQuality\_description />  
 <AssetQuality\_sortOrder />  
 <AssetQuality\_logo />  
 <AssetQuality\_ownerUser />  
 <ownerUser\_login />  
 <ownerUser\_ID />  
 <ownerUser\_DISCRIMINATOR />  
 <AssetQuality\_internalId />  
 <AssetQuality\_langId />  
 <langId\_name />  
 <langId\_ID />  
 <langId\_DISCRIMINATOR />  
 </row>  
 <row number="2">  
 <AssetQuality\_externalId> BADASSET </AssetQuality\_externalId>  
 <AssetQuality\_ID> 31618833 </AssetQuality\_ID>  
 <AssetQuality\_DISCRIMINATOR> ASSETQUALITY </AssetQuality\_DISCRIMINATOR>  
 <AssetQuality\_name> Не загружается / фильм неполный </AssetQuality\_name>  
 <AssetQuality\_description />  
 <AssetQuality\_sortOrder />  
 <AssetQuality\_logo />  
 <AssetQuality\_ownerUser />  
 <ownerUser\_login />  
 <ownerUser\_ID />  
 <ownerUser\_DISCRIMINATOR />  
 <AssetQuality\_internalId />  
 <AssetQuality\_langId />  
 <langId\_name />  
 <langId\_ID />  
 <langId\_DISCRIMINATOR />  
 </row>  
</rows>

### /Profile

Методы для работы с пользовательским профилем.

#### /Profile/setFavouriteChannel

**GET-запрос**

Метод добавляет ТВ-канал в список любимых на всех одноименных профилях. Максимальное число избранных каналов на профиле = 30 (настраивается в конфигурации SDP). Список избранных обновляется в порядке очереди. При добавлении 31-го канала 1-й канал будет удален, и новый канал добавится в конец списка.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* channelId — *number* — Идентификатор пакета ТВ-каналов. (Обязательный параметр.).
* profileName — *string* — Имя профиля. Если параметр не указан, позиция добавляется на текущий профиль. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /Profile/setFavouriteChannelsList

**GET-запрос**

Метод перезаписывает список избранных каналов на всех одноименных профилях. При выполнении метода старый список избранных каналов полностью очищается и в него записываются новые каналы. Максимальное число избранных каналов на профиле = 30 (настраивается в конфигурации SDP). При попытке добавить больше каналов, в список избранных сохранятся только первые 30.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* channelIdList — *string* — Список идентификаторов избранных ТВ-каналов через запятую. (Обязательный параметр.).
* profileName — *string* — Имя профиля. Если параметр не указан, позиция добавляется на текущий профиль. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /Profile/getFavouriteChannel

**GET-запрос**

Метод возвращает список любимых каналов для профиля. Список любимых каналов одинаков для одноименных профилей на разных устройствах.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* profileName — *string* — Имя профиля. Если параметр не указан, позиция добавляется на текущий профиль. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <items>  
 <item>  
 <channelId> 3411379 </channelId>  
 <profileName> Name1 </profileName>  
 <dt> 1340101959 </dt>  
 </item>  
 <item>  
 …  
 </item>  
 </items>  
</rows>

#### /Profile/update\_new

**GET-запрос**

Метод обновляет параметры профиля клиентского приложения и синхронизирует имена (name) и список избранных ТВ-каналов (favoriteChannels) на профилях с одинаковым именем. При вызове метода указываются обязательные параметры и список изменяемых параметров профиля.

На веб-сервере может быть ограничение на максимальную длину GET-запроса, а тело запроса Profile/update\_new может быть достаточно длинным. Поэтому рекомендуется использовать такое правило — запросы до 200 символов вызывать методом GET, 200 символов и более — методом POST.

**Добавление напоминания**

Символы <> внутри тегов-параметров профиля (<reminders>,\<forbidden>, <favorite> и др.) должны быть закодированы в HTML (&lt;, &gt;), в примерах ниже они используются только для удобства чтения.

Из возвращаемого значения ServiceAccount/list\_available\_profiles требуется текущий список напоминаний <reminders> нужного профиля, например, профиля master:

<rows code="0">  
 <row number="0">  
 <id> 11246124 </id>  
 <name> master </name>  
 …  
 <reminders>  
 <?xml version="1.0" encoding="UTF-8"?>  
 <remindlist>  
 <program id="16897255" cid="2843511" sdate="08/04/2011 14:44:31"/>  
 </remindlist>  
 </reminders>

Добавить в него новое значение, например, напоминание на канале с id=100 о передаче с id=200, дата и время начала передачи 20 апреля 2012 года:

<reminders>  
 <?xml version="1.0" encoding="UTF-8"?>  
 <remindlist>  
 <program id="16897255" cid="2843511" sdate="08/04/2011 14:44:31"/>  
 <program id="200" cid="100" sdate="04/20/2011 14:44:31"/>  
 </remindlist>  
 </reminders>

Выполнить запрос Profile/update\_new методом POST на сервер SDP, передав в теле запроса параметры:

1. ID профиля;
2. parameters — новый список напоминаний <reminders> с XML-форматированием.

Синтаксис GET-запроса (text/xml; charset=utf-8) будет иметь вид:

http://<host>:8080/smarttube/master/spine/services/xml/Profile/update\_new?ID=15472491&reminders=%26lt%3B%3Fxml%20version%3D%26quot%3B1.0%26quot%3B%20encoding%3D%26quot%3BUTF-8%26quot%3B%3F%26gt%3B%0A%26lt%3Bremindlist%26gt%3B%26lt%3Bprogram%20id%3D%26quot%3B16897255%26quot%3B%20cid%3D%26quot%3B2843511%26quot%3B%20sdate%3D%26quot%3B08%2F04%2F2011%2014%3A44%3A31%26quot%3B%2F%26gt%3B%26lt%3Bprogram%20id%3D%26quot%3B200%26quot%3B%20cid%3D%26quot%3B100%26quot%3B%20sdate%3D%26quot%3B04%2F20%2F2011%2014%3A44%3A31%26quot%3B%2F%26gt%3B&mac-address=1751482113&sessionId=104042

В результате выполнения метод возвращает ID измененного профиля.

**Изменение пин-кода**

Чтобы изменить PIN профиля, выполните Profile/update\_new и укажите в параметрах ID профиля и новый PIN-код:

http://<host> : <port>/smarttube/master/spine/services/xml/Profile/update\_new?ID=15472491&pin=101010&mac-address=1751482113&sessionId=104042

**Cохранение позиции просмотра в видеофильме**

Клиентское приложение может периодически обновлять позицию просмотра в фильме, например, каждые 5 минут. Чтобы обновить позицию просмотра для заданного фильма, нужно вызвать метод Content/addMediaPosition? и передать в него список позиций в видеофильмах. Чтобы получить список позиций просмотра фильмов, используйте Content/mediaPositionList?.

**Изменение уровня доступа профиля**

В профиле предусмотрено 2 уровня доступа: Максимальный уровень доступа профиля – максимально возможный уровень контента, который должен быть доступен. Параметр maxAccessLevel. Текущий уровень доступа профиля — значение от минимального до максимального уровня доступа профиля, которое позволяет быстро изменить уровень доступа без необходимости переключать/перенастраивать профиль. Параметр lastAccessLevel.

В БД хранятся оба значения, можно использовать их согласно логике приложения. Дополнительно в профиле предусмотрен флаг сохранения текущего уровня доступа isAccessLevelPersistent. Вы можете использовать его согласно логике приложения. Чтобы изменить максимальный уровень доступа профиля, выполните запрос, указав идентификатор нужного уровня:

http://<host> : <port>/smarttube/master/spine/services/xml/Profile/update\_new?ID=15472491&maxAccessLevel=127100&mac-address=1751482113&sessionId=104042

Чтобы изменить текущий уровень доступа профиля, выполните запрос, указав идентификатор нужного уровня:

http://<host> : <port>/smarttube/master/spine/services/xml/Profile/update\_new?ID=15472491&lastAccessLevel=127100&mac-address=1751482113&sessionId=104042

**Добавление специальных параметров**

Если используются какие-то собственные параметры профиля, например, плейлисты iPad, храните их в поле customProperties, не удаляя имеющихся в нем значений.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <ENTITY\_NAME> Profile </ENTITY\_NAME>  
 <IDS> 11246109 </IDS>  
 </row>  
 <row>  
 …  
 </row>  
 …  
</rows>

#### /Profile/insert\_new

**GET-запрос**

Метод добавляет профиль клиентского приложения с заданным именем на каждое устройство, привязанное к учетной записи (STB, веб-подключение, подключение iPad). Если на каких-то устройствах уже есть профиль с таким именем, он остается без изменения. Профиль добавляется только на те устройства, на которых его не было. После добавления профиля для настройки его параметров используется метод Profile/update\_new.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* name — *string* — Имя контента во внешней системе. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <ENTITY\_NAME> Profile </ENTITY\_NAME>  
 <IDS> 11246109 </IDS>  
 </row>  
 <row>  
 …  
 </row>  
 …  
</rows>

#### /Profile/delete\_new

**GET-запрос**

Метод удаляет профили клиентского приложения по имени. Профили с одинаковым именем удаляются со всех устройств на учетной записи (STB, веб-подключение, подключение iPad). При удалении сначала идет поиск профиля с указанным ID, затем идет поиск профилей с таким же именем, как у найденного и принадлежащих той же учетной записи, что и найденный профиль. Далее одновременно все нужные профили удаляются. Но в возвращаемом значении в случае успешного выполнения метода возвращается только ID того профиля, который нужно было удалить изначально. Остальные профили с таким же именем на других устройствах на данной учетной записи удаляются без сообщения об удалении.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <ENTITY\_NAME> Profile </ENTITY\_NAME>  
 <IDS> 11246109 </IDS>  
 </row>  
 <row>  
 …  
 </row>  
 …  
</rows>

#### /Profile/copy

**GET-запрос**

Метод создает новые профили и копирует в них все настройки из указанного профиля.

**URL-параметры запроса:**

* name — *string* — Имя контента во внешней системе. (Обязательный параметр.).
* copiedprofileID — *number* — Идентификатор копируемого профиля. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /Recommender

#### /Recommender/get\_vod\_by\_profile

**GET-запрос**

Метод возвращает список рекомендованных фильмов для профиля по его идентификатору. В ответе в поле указан вес фильма. Наибольший приоритет должны иметь фильмы с большим значением веса.

**URL-параметры запроса:**

* profileId — *number* — Идентификатор профиля. Необходим для корректной работы рекомендатора. (Обязательный параметр.).
* serviceAccount — *string* — Номер учетной записи, на которой нужно найти веб-подключение. .

**Параметры ответа:**

Пример XML-данных ответа:

<recomendedContent code="0">  
 <content>  
 <id> 17735572 </id>  
 <weight> 3.0 </weight>  
 </content>  
 <content>  
 <id> 15502751 </id>  
 <weight> 3.0 </weight>  
 </content>  
</recomendedContent>

### /City

#### /City/list\_stb

**GET-запрос**

Метод возвращает список городов, отфильтрованных по идентификатору региона.

**URL-параметры запроса:**

* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <City\_externalId> 100452949 </City\_externalId>  
 <City\_ID> 25152823 </City\_ID>  
 <City\_name> Москва </City\_name>  
 </row>  
 ...  
</rows>

### /ForecaWeather

#### /ForecaWeather/get\_foreca\_forecast

**GET-запрос**

Метод возвращает прогноз погоды в заданном городе.

**URL-параметры запроса:**

* cityId — *number* — (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<weatherlist cityId="25152823" version="-1">  
 <weather>  
 <cityId> 25152823 </cityId>  
 <temperature> -13 </temperature>  
 <forecastDate> 03/09/2012 01:00:00 </forecastDate>  
 <forecastDateUnix> 1331240400 </forecastDateUnix>  
 <windSpeed> 2 </windSpeed>  
 <windType> NE </windType>  
 <precipType> 10 </precipType>  
 <precipVolume> 0 </precipVolume>  
 <cloud> 3 </cloud>  
 <humidity> 81 </humidity>  
 <pressure> 756 </pressure>  
 </weather>  
 ...  
</weatherlist>

### /PromoSchedule

#### /PromoSchedule/list\_for\_interface

**GET-запрос**

Метод возвращает экземпляр расписания - список баннеров и условия их показа. Если какой-либо из параметров , , пуст, то баннеры применяются для всех тарифных планов, регионов и типов устройств соответственно. Баннеры из списка показываются последовательно в цикле, длительно показа одного баннера равна . Если требуется показывать несколько групп баннеров (несколько списков ), то сначала показываются баннеры из первого списка, потом баннеры из второго и т.д. в цикле.

**URL-параметры запроса:**

* interfaceId — *number* — (Обязательный параметр.).
* stbType — *string* — .
* poId — *number* — Идентификатор тарифного плана. .
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. .

**Параметры ответа:**

Пример XML-данных ответа:

<promo\_schedules code="0">  
 <promo\_schedule>  
 <id> 24168713 </id>  
 <name> Test1 </name>  
 <startDate> 1331899200 </startDate>  
 <endDate> 1331985600 </endDate>  
 <duration/>  
 <from\_sec/>  
 <to\_sec/>  
 <promos>  
 <id> 24425354 </id>  
 </promos>  
 <p\_offers>  
 <id> 2455498 </id>  
 </p\_offers>  
 <locations>  
 <id> 145462 </id>  
 <locations>  
 <res\_specs>  
 <name> IPAD </name>  
 </res\_specs>  
 </promo\_schedule>  
 </promo\_schedules>

### /Promo

#### /Promo/get

**GET-запрос**

Метод возвращает баннер фильма. Если в ответе какой-либо параметр logo пуст, это означает, что картинки такого разрешения нет в SDP.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <promos version="-1">  
 <promo>  
 <id> 24425354 </id>  
 <name> test </name>  
 <logo> 1323698596160.jpg </logo>  
 <logo2/>  
 <logo3/>  
 <logo4/>  
 <logo5/>  
 <logo6/>  
 <c\_id/>  
 </promo>  
 ...  
 </promos>  
</rows>

### /PromoCodeGroup

Методы для работы с промокодами.

#### /PromoCodeGroup/checkCode

**GET-запрос**

Метод для проверки промокода. В случае неудачи метод возвращает одну из следующих ошибок PROMOCODE\_NOT\_FOUND (промокод не найден), PROMOCODE\_NOT\_ACTIVE (промокод просрочен), PROMOCODE\_ALREADY\_IN\_USE (промокод уже использован), PROMOCODE\_VODPACK\_NOT\_FOUND (промокод невалиден), PROMOCODE\_SERVICE\_NOT\_FOUND (промокод не поддерживается в тарифе).

**URL-параметры запроса:**

* promocode — *string* — Промокод. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <status> 1 </status>  
 <isActive> 1 </isActive>  
 <promoPeriodDay> 5 </promoPeriodDay>  
 <discriminator> VODABONEMENT </discriminator>  
 <objectId> 37633982 </objectId>  
 <objectName> AMEDIA Premium </objectName>  
 </row>

#### /PromoCodeGroup/useCode

**GET-запрос**

Метод для активции промокода. В случае неудачи метод возвращает одну из следующих ошибок PROMOCODE\_NOT\_FOUND (промокод не найден), PROMOCODE\_NOT\_ACTIVE (промокод просрочен), PROMOCODE\_ALREADY\_IN\_USE (промокод уже использован), PROMOCODE\_VODPACK\_NOT\_FOUND (промокод невалиден), PROMOCODE\_SERVICE\_NOT\_FOUND (промокод не поддерживается в тарифе), PROMOCODE\_UNKNOWN\_ERROR (ошибка активации промокода).

**URL-параметры запроса:**

* promocode — *string* — Промокод. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* profileId — *string* — Идентификатор профиля. Необходим для корректной работы рекомендатора. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> <row> </row>

### /RTUGC

Методы для работы с пользовательским контентом.

#### /RTUGC/create

**GET-запрос**

Метод добавляет ролик UGC.

**URL-параметры запроса:**

* name — *string* — Имя контента во внешней системе. (Обязательный параметр.).
* size — *string* — Размер файла в байтах. (Обязательный параметр.).
* locationId — *number* — Идентификатор региона. При передаче метод возвращает список фильмов с доступными для просмотра IPTV-ассетами. (Обязательный параметр.).
* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* dsc — *string* — Имя франшизы. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <videoServer> 10.65.50.151 </videoServer>  
 <internalFileName> ugc\_center\_25133663.avi </internalFileName>  
 <ugcId> 25133663 </ugcId>  
</rows>

#### /RTUGC/setStatus

**GET-запрос**

Метод изменяет состояние пользовательского ролика.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* status — *number* — Идентификатор нового состояния ролика. Возможные значения: 0 — новый; 1 — загружается; 2 — траскодируется; 3 — ошибка загрузки; 4 — ожидание доступной квоты; 5 — загружен (доступен); 6 — удален. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

#### /RTUGC/listByAcc

**GET-запрос**

Метод возвращает список роликов, загруженных на учетную запись.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<ugc\_list code="0">  
 <ugc\_films>  
 <ugc\_film>  
 <id> 25133663 </id>  
 <name> test </name>  
 <sDate> 1330948793 </sDate>  
 <quota> 98 </quota>  
 <status> 1 </status>  
 <description> testdsc </description>  
 <assets>  
 <content\_asset>  
 <id> 25133664 </id>  
 <ifn> ugc\_center\_25133663.ts </ifn>  
 <name> mymovie.avi </name>  
 <vs\_ip\_address> 10.240.65.210 </vs\_ip\_address>  
 <vs\_port> 5000 </vs\_port>  
 <duration> -1 </duration>  
 <asset\_type> 57186 </asset\_type>  
 <is\_crypted> 0 </is\_crypted>  
 <transport> </transport>  
 </content\_asset>  
 </assets>  
 </ugc\_film>  
 ...  
 </ugc\_films>  
<ugc\_list/>

#### /RTUGC/delete

**GET-запрос**

Метод посылает команду на удаление ролика серверу SmartTUBE SDP. Физическое удаление ролика из дискового хранилища выполняет SDP.

**URL-параметры запроса:**

* ID — *number* — Идентификатор фильма. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /RTUGCQuota

#### /RTUGCQuota/get\_service\_account\_quotas

**GET-запрос**

Метод возвращает размер доступной и израсходованной квоты в рамках услуг UGC и nPVR.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <serviceAccountNumber> 102102 </serviceAccountNumber>  
 <allQuota> 1048576 </allQuota>  
 <usedQuota> 53577 </usedQuota>  
 </row>  
</rows>

### /RecordedProgram

Методы для работы с записанными телепередачами.

#### /RecordedProgram/list\_by\_acc

**GET-запрос**

Метод возвращает список записанных ТВ-передач.

Справочник состояний записанной передачи <npvrCommandsStatus>:

* 1 — не подтверждена видеосервером (дана команда на запись).
* 2 — видеосервер подтвердил начало записи.
* 3 — видеосервер подтвердил окончание записи.
* 4 — отменена; количество запросов на эту передачу равно 0.
* 5 — не записана (не было подтверждения записи – ошибка).
* 6 — временно не доступна.
* 7 — удалена по истечению времени хранения.
* 8 — сохранена (не доступна по времени, но не удалена с видеосервера).

Справочник состояний записанной передачи в SDP <recordedProgramStatus>:

* 1 — дана команда на запись.
* 2 — запись успешно выполнена.
* 3 — запись отменена.
* 4 — запись недоступна из-за огрничения время хранения.
* 5 — заблокировано из-за отсутствия квоты (UGC).

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <ID> 5111015 </ID>  
 <scheduleId> 5109793 </scheduleId>  
 <channelId> 5009769 </channelId>  
 <channelName> Усадьба </channelName>  
 <recordedProgramId> 5111015 </recordedProgramId>  
 <recordedProgramName> Осторожно - злая собака </recordedProgramName>  
 <recordedProgramDescription> Программа о содержании среднеазиатской овчарки (алабай), в условиях загородного дома, особенности воспитания, дрессировки, экстерьер пароды, пожелания хозяевам в уходе за собакой. </recordedProgramDescription>  
 <startDate> 1419496500 </startDate>  
 <endDate> 1419498300 </endDate>  
 <programId> 5109792 </programId>  
 <programName> Осторожно - злая собака </programName>  
 <logo />  
 <programDescription> Программа о содержании среднеазиатской овчарки (алабай), в условиях загородного дома, особенности воспитания, дрессировки, экстерьер пароды, пожелания хозяевам в уходе за собакой. Особенности питания, выбор щенка, плюсы и минусы характера. </programDescription>  
 <assetId> 5111016 </assetId>  
 <assetInternalFileName> NPVR\_RECORD\_24594967.ts </assetInternalFileName>  
 <assetDuration> 300 </assetDuration>  
 <assetType> 1 </assetType>  
 <assetIsCrypted> 0 </assetIsCrypted>  
 <recordedProgramStatus> 1 </recordedProgramStatus>  
 <videoserverIpAddress> 10.65.50.120 </videoserverIpAddress>  
 <videoserverPort> 554 </videoserverPort>  
 <npvrCommandsStatus> 1 </npvrCommandsStatus>  
 <filesize> 0 </filesize>  
 <accessLevelId> 14712604 </accessLevelId>  
 <validDate> 1324978500 </validDate>  
 <is\_ott> true </is\_ott>  
 <isPermanent> 0 </isPermanent>  
 <channelName\_RUS> Усадьба </channelName\_RUS>  
 <channelName\_ENG> Homestead </channelName\_ENG>  
 <channelName\_BEL> Усадьба </channelName\_BEL>  
 </row>  
</rows>

#### /RecordedProgram/cancel

**GET-запрос**

Метод изменяет статус ТВ-передачи для учетной записи. Его можно использовать для отмены запланированной записи или «удаления» уже записанной передачи. После выполнения метода RecordedProgram/cancel она переходит в состояние «Удалена» для данной учетной записи. При этом автоматически освобождается дисковое пространство в рамках файловой квоты, которое было зарезервировано под передачу. При выполнении метода для уже записанной передачи физического удаления файла не происходит – сервер удаляет ассет с видеосервера только тогда, когда не остается учетных записей, которые записали эту программу в рамках nPVR, истек период возможного использования этой записи в рамках Pause Live и TSTV.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* recProgId — *string* — Идентификатор записанной ТВ-передачи. Значение поля <recordedProgramId> передачи, возвращаемой методом RecordedProgram/list\_by\_acc. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /OttContentAsset

#### /OttContentAsset/updateName

**GET-запрос**

Метод изменяет имя основного OTT ассета в SDP.

**URL-параметры запроса:**

* oldName — *string* — Существующее имя OTT-ассета. (Обязательный параметр.).
* newName — *string* — Новое имя OTT-ассета. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /OttPreviewAsset

#### /OttPreviewAsset/updateName

**GET-запрос**

Метод изменяет имя OTT ассета трейлера в SDP.

**URL-параметры запроса:**

* oldName — *string* — Существующее имя OTT-ассета. (Обязательный параметр.).
* newName — *string* — Новое имя OTT-ассета. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /SocialNetworks

#### /SocialNetworks/add\_content\_feedback

**GET-запрос**

Метод устанавливает отметку Нравится/Не нравится контенту (фильму) в базе данных SmartTUBE SDP для заданного профиля.

**URL-параметры запроса:**

* profileId — *string* — Идентификатор профиля. Необходим для корректной работы рекомендатора. (Обязательный параметр.).
* contentId — *string* — Идентификатор видеофильма. (Обязательный параметр.).
* feedback — *string* — Оценка: -1 — Не нравится; 1 — Нравится; 0 — Снять отметку. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. (Обязательный параметр.).
* sessionId — *string* — ID сессии: значение sessionId, возвращаемое в методе ServiceAccount/web\_authorize, либо значение noSession в случае, если идентификатор сессии неизвестен. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"/>

### /ottSpy

#### /ottSpy

**GET-запрос**

Метод пересылает на сервер сообщения о событиях в клиентском приложении, связанных с просмотром контента. Для использования метода необходимо получить IP-адрес и порт сервера статистики из запроса авторизации (параметры и соответственно). События типа keep alive (LKA, DVRKA, VODKA) при непрерывном просмотре выбранной единицы контента должны посылаться каждые 5 мин.

**URL-параметры запроса:**

* san — *string* — Номер учетной записи. (Обязательный параметр.).
* idc — *string* — ID канала илифильма в SDP. (Обязательный параметр.).
* dt — *string* — Тип устройства (deviceType). (Обязательный параметр.).
* e — *string* — Тип события: LS — live start, запуск просмотра ТВ-канала; LKA — live keep alive, продоложение просмотра ТВ-канала; DVRS — dvr start, запуск просмотра записи; DVRKA — dvr keep alive, продолжение просмотра записанной передачи; VODS — vod start, запуск просморта видеофильма; VODKA — vod keep alive, продолжение просмотра видеофильма; LE — событие ошибки (отсутствие контента — ТВ-канала); DVRE — событие ошибки (отсутствие контента — записанной ТВ-передачи); VODE — событие ошибки (отсутствие контента — видеофильма). (Обязательный параметр.).
* rid — *string* — ResourceId из запроса авторизации (идентификатор абонентского устройства). (Обязательный параметр.).

**Параметры ответа:**

### /asset

#### /asset/setQuality

**GET-запрос**

Метод отправляет в SDP данные о плохом качестве ассета – его идентификатор и код проблемы.

**URL-параметры запроса:**

* assetId — *string* — Идентификатор ассета. (Обязательный параметр.).
* assetQualityId — *string* — Код проблемы. Значение из метода AssetQuality/list. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /PaymentSystem

Методы для работы с платежами и платежными системами.

#### /PaymentSystem/createPayment

**GET-запрос**

Метод посылает запрос «создание платежа» в платежную систему ЕСПП.

В случае ошибки SDP метод возвращает следующие структуры: Учетная запись не найдена:

<rows code="1006">  
 <message> Service account not found </message>  
</rows>

В справочнике регионов не определен параметр “Область именования для платежной системы”:

<rows code="10003">  
 <message> payTypeId on location is empty </message>  
</rows>

Если код ошибки равен единице (<rows code="1">) или принимает отрицательные значения, то ошибка произошла в платежной системе. В данном случае в параметре (<message>) повторяется содержимое поля «Сообщение, предназначенное для вывода непосредственно плательщику в случае неуспешного выполнения транзакции», которое возвращается платежной системой. Если данное поле пустое, метод возвращает сообщение «Произошла ошибка во время проведения операции».

* -1 ERROR\_BUSY: Сервер временно недоступен. По внутренним техническим причинам сервер не обрабатывает запросы.
* -2 ERROR\_ACCESS\_DENIED: Доступ к сервису ЕСПП запрещен. Необходимо проанализировать комментарий поля reqNote: возможно запрос делается с недопустимого IP, с некорректным сертификатом, не включен в боевой режим.
* -3 ERROR\_BAD\_REQ: Неизвестный или неподдерживаемый запрос.
* -4 ERROR\_BAD\_FORMAT: Неверный формат запроса. Указаны не все обязательные поля, либо значение полей имеют неправильный формат.
* -12 ERROR\_ACCOUNT\_NOT\_FOUND: Не найдена учетная запись ЕЛК с присоединенной услугой, идентификатор которой передан в запросе на проведение платежа.
* -15 ERROR\_REQ\_DENIED: Запрос отклонен. Причина отклонения указывается в поле reqNote. Сообщение, выводимое непосредственно плательщику, передается в поле errUsrMsg.
* -17 ERROR\_BAD\_SVC\_TYPE: Недопустимое пространство имен.
* -21 ERROR\_PAY\_LIMIT\_EXCEEDED: Превышена сумма допустимых операций по счету за период. Ошибка предназначена для таких видов контроля операций по счету, как контроль максимального числа и максимальной суммы операций за период.
* -26 ERROR\_CONFIRM\_CODE\_NOT\_FOUND: Код подтверждения трансфера не найден.
* -31 ERROR\_PHONE\_NOT\_FOUND: Отсутствует (или некорректный) номер телефона для отправки кода подтверждения.
* 1 ERROR\_PAY\_NOT\_FOUND: Платеж с таким идентификатором отсутствует.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* amount — *string* — Сумма платежа в копейках. (Обязательный параметр.).
* svcTypeId — *string* — Область именования для платежной системы. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <srcPayId> 2066893 </srcPayId>  
 </row>  
</rows>

#### /PaymentSystem/createContentPayment

**GET-запрос**

Метод выполняет регистрацию платежа в ЕСПП. В ответе приходит orderId - уникальный идентификатор платежа в ЕСПП.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* contentId — *number* — Идентификатор видеофильма. (Обязательный параметр.).
* contentPackageId — *number* — Идентификатор пакета видеофильмов. (Обязательный параметр.).
* contentUsageRuleId — *number* — Идентификатор правила покупки, по которому приобретается видеофильм полученный с помощью метода ServiceSpec/list\_terminal\_type\_rules, выходной параметр <cur\_id>. (Обязательный параметр.).
* mac-address — *string* — MAC-адрес абонентского устройства. Параметр resourceId, полученный методом ServiceAccount/web\_authorize, либо параметр <mac>, полученный методом elk/authorize. .
* assetName — *string* — Название фильма. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <orderId> 774589 </orderId>  
 </row>  
</rows>

#### /PaymentSystem/confirmPayment

**GET-запрос**

Метод служит для подтверждения абонентом платежа с помощью кода, полученного в SMS-сообщении.

**URL-параметры запроса:**

* scrPayId — *string* — Идентификатор платежа, полученный методом createPayment. (Обязательный параметр.).
* code — *string* — Код, полученный абонентом в SMS-сообщении. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

#### /PaymentSystem/getPaymentStatus

**GET-запрос**

Метод выполняет проверку статуса платежа в платежной системе ЕСПП.

**URL-параметры запроса:**

* scrPayId — *string* — Идентификатор платежа, полученный методом createPayment. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row>  
 <status> 102 </status>   
 </row>  
</rows>

#### /PaymentSystem/listPayments

**GET-запрос**

Метод возвращает список платежей, выполненных с учетной записи за последние неделю, месяц или за все время. В случае если возвращает пустое значение, или значения 102, 103, значит ни один финальный статус платежа не получен. Необходимо дождаться получения финального статуса.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* Period — *string* — Период, за который необходимо выдать список платежей. Допустимые значения: week — за неделю; month — за месяц. По умолчанию — за все время. .

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0">  
 <row number="0">  
 <payDate> 1372938664 </payDate>  
 <quantity> 500 </quantity>  
 <status/>  
 <confirmDate/>  
 <srcPayId> 2066912 </srcPayId>  
 </row>  
 …  
 <row number="5">  
 …  
 </row>  
</rows>

#### /PaymentSystem/createTopUpPayment

**GET-запрос**

Метод создает запрос для пополнения ЛС через произвольную банковскую карту.

**URL-параметры запроса:**

* serviceAccountNumber — *string* — Номер учетной записи. (Обязательный параметр.).
* amount — *number* — Сумма платежа в копейках. (Обязательный параметр.).

**Параметры ответа:**

Пример XML-данных ответа:

<rows code="0"> </rows>

### /change\_service\_account

#### /change\_service\_account

**GET-запрос**

При вызове метода /mwapi/service\_account\_change\_number, и смене номера учетной записи, SDP вызывает данный метод в портале, для нотификации портала о том что изменился номер учетной записи.

**URL-параметры запроса:**

* old\_service\_account — *string* — Номер учетной записи, который нужно изменить. (Обязательный параметр.).
* new\_service\_account — *string* — Новый номер учетной записи. (Обязательный параметр.).
* mrf — *string* — Идентификатор МРФ. (Обязательный параметр.).
* hash — *string* — Конкатенация параметров old\_service\_account, new\_service\_account, mrf и секретного ключа. (Обязательный параметр.).

**Параметры ответа:**

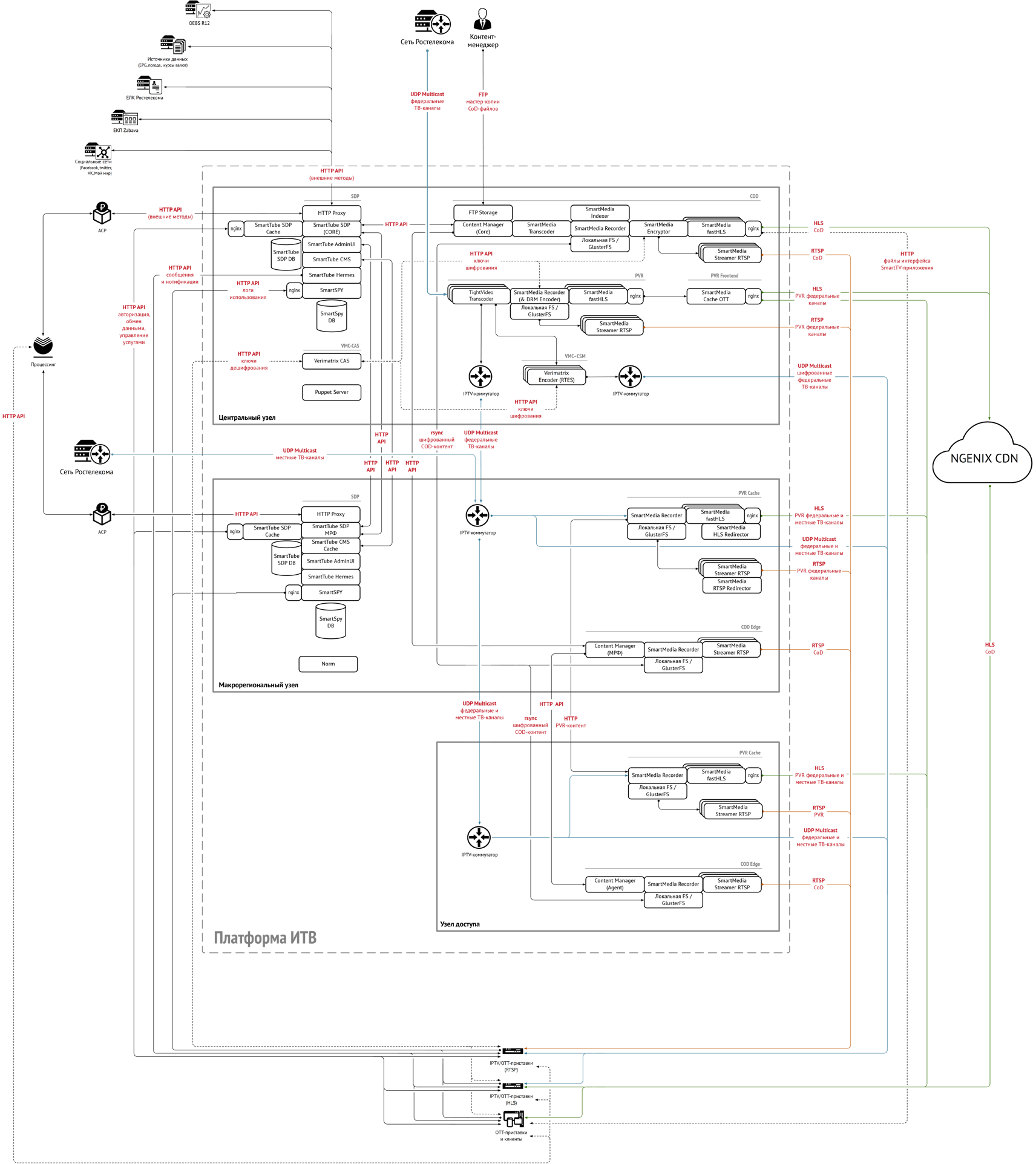
Пример XML-данных ответа:

<rows code="0">

## Параметры возвращаемых XML-данных

* accessLevelId — идентификатор уровня доступа записанной передачи.
* activationNumber — номер активации группы устройств.
* activationNumber — номер активации.
* actors — список актеров через запятую.
* ageLevel — идентификатор Возраста, значение для отражения в клиентском приложении (справочник – уровень доступа).
* al — идентификатор уровня доступа.
* allQuota — полный размер квоты в килобайтах.
* asset\_type — идентификатор типа ассета по справочнику, используется в правилах использования VodPackage/list\_rules.
* assetDuration — длительноость записи в секундах.
* assetDuration — длительность ТВ-передачи.
* assetId — идентификатор ассета.
* assetInternalFileName — имя файла с записанной передачей.
* assetInternalFileName — имя файла с записанной передачей для SDP.
* assetIsCrypted — флаг шифрования ассета или передачи, 0 - не шифрован.
* assetType — тип ассета.
* AudioPID\_audioPIDType — тип аудио PID.
* AudioPID\_broadcastChannelID — идентификаторТВ-канала.
* AudioPID\_ID — идентификатор аудио PID.
* AudioPID\_name — название аудио PID, отображаемое на экране.
* AudioPID\_PID — значение PID.
* backend\_version — версия сервера SDP.
* balance — баланс лицевого счета в копейках без разделителя, со знаком минус – если отрицательный.
* bonusType — идентификатор бонусного типа абонента. Идентификатор используется для загрузки бонусных вариантов покупки с помощью метода Content/list\_bonus\_price.
* broadcastChannelID\_DISCRIMINATOR — следующие значения берутся из таблицы-словаря аудио PID.
* broadcastChannelID\_ID — идентификатор ТВ-канала.
* broadcastChannelID\_name — название ТВ-канала.
* bundle — название сериала, если передача является эпизодом какого-то сериала.
* bundle — описание сезона.
* bundles — список сезонов, которые входят в сериал. Описание сезона можно получить вызовом ContentBundlePackage/list\_bundles?ID=.
* c\_id — идентификатор ТВ-канала.
* c\_id — идентификатор фильма, к которому относится баннер. Описание фильма можно получить вызовом VodPackage/list\_movies?ID=\* c\_id.
* channelId — идентификатор ТВ-канала.
* channelName — название ТВ-канала.
* City\_externalId — внешний идентификатор города.
* City\_ID — идентификатор города для загрузки прогноза погоды вызовом ForecaWeather/get\_foreca\_forecast.
* City\_name — название города.
* cityId — идентификатор города.
* clogo — имя файла со скриншотом к ТВ-передаче.
* cloud — тип облачности.
* cnt — количество караоке-роликов, принадлежащих данному жанру.
* comment — сообщение об ошибке или статус покупки.
* confirmDate — дата и время подтверждения платежа в формате Unix time.
* content\_id — идентификатор контента.
* ContentBundleGenre\_description — описание жанра.
* ContentBundleGenre\_ID — идентификатор жанра.
* ContentBundleGenre\_name — название жанра.
* country — страна.
* count — максимальное количество устройств данного типа.
* cur\_id — идентификатор правила использования, content\_usage\_rule\_id
* date — дата покупки.
* datetime — дата и время авторизации в формате MM/DD/YYYY HH:MM:SS.
* desc — подробное описание.
* deviceType — тип устройства: OTTSTB – ОТТ приставка, другое значение - IPTV (примечание: в методе получения списка типов устройств /DeviceType/list тип устройства приходит в поле externalId).
* DeviceType\_externalId – идентификатор типа устройства. Именно это значение необходимо использовать в приложении
* director — режиссер.
* discr — идентификатор сущности «ТВ-передача», равен PROGRAM.
* dkey — дата передачи.
* duration — длительность показа каждого баннера из списка, значение в секундах или длительность файла в секундах, -1 означает, что значение не задано.
* eDate — дата окончания проката, unixtime.
* eid — внешний идентификатор передачи
* endDate — дата-время окончания передачи или баннера, unixtime
* externalID — внешний идентификатор сезона или сериала.
* fdate — дата и время окончания передачи, временная зона UTC.
* filesize — размер записанного файла.
* forecastDate — время прогноза по местному времени в строковом формате, 8 возможных значений: 01:00, 04:00, 07:00, 10:00, 13:00, 16:00, 19:00, 22:00
* forecastDateUnix — время прогноза по местному времени в формате Unix time
* from\_sec — время начала показа баннеров в секундах относительно суток, значение от 0 (соответствует 00:00:00 часов), до 86399 (соответствует 23:59:59.
* genre — идентификатор жанра передачи, значение ProgramGenre\_ID жанра, возвращаемое в методе ProgramGenre/list
* GenreDictionary\_isFinal — признак, могут ли быть у данного жанра вложенные жанры. 0 - могут быть, 1 – не могут быть.
* genres — жанры, к которым относится сезон или сериал (список идентификаторов). Список жанров можно получить вызовом ContentBundleGenre/list.
* humidity — относительная влажность в %, точность 0,5%.
* id — идентификатор ассета/баннера/жанра/пакета/пакета сезонов/пакета ТВ-каналов/расписания/региона/ролика/сезона/сериала/ТВ-передачи/устройства/фильма. В отдельном случае общее количество разных караоке-роликов в пакетах.
* ifn — внутреннее имя файла на диске видеосервера
* imdbR — рейтинг IMDB
* incompatible\_packages — список идентификаторов пакетов, которые являются несовместимыми для указанного выше пакета.
* internalFileName — Внутреннее имя файла, сгенерированное сервером SmartTUBE.
* ir — Флаг, является ли сериал рекомендуемым (продвигаемым). 1 - рекомендуемый, 0 - нет.
* is\_crypted — флаг шифрования. 0 – не зашифрован, 1 — зашифрован.
* is\_ott — флаг записи через Bradbury. Значения: true – передача записана через Bradbury, false – через SmartMEDIA
* isPermanent — флаг размещения записи в архиве
* isPopular — флаг «Популярные сериалы».
* isProlongation — признак покупки без ограничения срока действия: 0 – купленный контент действителен в течении N часов 1 – нет ограничения по времени использования покупки (автоматическая пролонгация).
* isTrustedDevice — Признак того, что с устройства можно оплачивать покупки с личного счета, ассоциированного с учетной записью, без ввода пароля.
* kinopR — рейтинг на Кинопоиске.
* loc — Регионы, в которых доступен сезон или сериал.
* locationId — Идентификатор региона.
* locations — Список идентификаторов регионов. Регионы показа баннеров. На iPad и в веб-приложении нужно показывать только те баннеры, которые предназначены для регионов iPad и Web.
* logo — имя файла с картинкой для портала или постера.
* logo2 — имя файла с картинкой для портала/к сезону/к сериалу.
* logo3 — имя файла с картинкой для iPad.
* logo4 — имя файла с картинкой для iPad.
* logo5 — имя файла с картинкой для iPad (витрина “Обезьяна”).
* logo6 — имя файла с картинкой для STB.
* movies — список фильмов-серий, которые входят в сезон. Описание серии можно получить вызовом VodPackage/list\_movies?ID=
* multiroom — флаг того, что данное подключение функционирует в режиме multiroom.
* name — имя или название баннера/файла/экземпляра расписания/передачи/пользовательского ролика/сезона/сериала.
* netLogIP — IP-адрес и порт сервера статистики.
* networkTypeId — внешний идентификатор типа сетевого соединения из справочника «Типы сетевого подключения». Возвращает значение по следующему правилу: если параметр network\_type установлен на терминале, то возвращает значение с терминала; если параметр network\_type не установлен на терминале, то возвращает значение с учетной записи (в том случае, если установлен на ней).
* notificationLocationId — Идентификатор региона учетной записи, который портал должен передать в метод синхронизации позиций (позиция остановки просмотра медиаконтента на STB).
* npvrCommandsStatus — Физическое состояние записанной передачи (см. значения в «Примечаниях»)
* orderId — уникальный идентификатор платежа со стороны ЕСПП.
* ott\_contentasset — имя файла, переданное в параметре ottAssetName. Имя файла можно изменить вызовом OttContentAsset/updateName?
* OttSNetwork\_externalId — Идентификатор соцсети. Именно это значение необходимо использовать в приложении
* p — описание передачи.
* p\_offers — список идентификаторов тарифных планов. Тарифные планы, абонентам которых следует показывать баннеры.
* packageId — идентификатор пакета, из которого куплен сезон или фильм.
* packages — пакеты, в которые входит сезон или сериал.
* payDate — дата платежа в формате Unix time
* pin — pin группы устройств
* precipType — тип осадков.
* precipVolume — суммарный объем осадков за 3 часа в мм, точность 0,5 мм
* pressure — атмосферное давление в мм.рт.ст., точность 0,5 мм.рт.ст.
* productOfferId — внутренний идентификатор тарифного плана в SDP.
* profile\_id — идентификатор профиля для которого покупка должна быть доступна. Если указано значение и параметр master-профиля isPublic=0, то купленный контент должен быть доступен только для данного профиля. Иначе покупка должна быть доступна для всех профилей.
* programDescription — описание передачи на данный момент.
* ProgramGenre\_ID — идентификатор жанра передач.
* ProgramGenre\_name — название жанра передач.
* programId — идентификатор ТВ-передачи на данный момент.
* programName — название ТВ-передачи на данный момент.
* promos — список баннеров, на которые распространяется данное расписание. Для загрузки баннеров из списка используйте Promo/get?ID=.
* quantity — величина платежа в копейках
* quota — размер занятой файлом квоты в килобайтах
* quotaRequired — полный размер требуемой файловой квоты, равен quotaUsed+ оценка дискового пространства, требуемого для записи передачи, в килобайтах. Значение вычисляется по битрейту, заданного в параметрах канала в SDP
* quotaTotal — полный размер доступной файловой квоты в килобайтах
* quotaUsed — текущий расход файловой квоты в килобайтах (без учета запланированной на запись передачи)
* rating — рейтинг.
* reason — внешний идентификатор причины блокировки (игнорируется, если состояние учетной записи != BLOCKED). Справочник причин блокировок можно загрузить вызовом Reason/list к simple кеш-клиенту SDP.
* recomW — вес передачи в рекомендации.
* recordedProgramDescription — описание передачи на момент записи
* recordedProgramId — идентификатор записанной передачи.
* recordedProgramName — имя передачи на момент записи.
* recordedProgramStatus — состояние записанной передачи.
* reload — количество минут, через которое метод будет выполнен повторно, если 0 – повторный запрос не будет осуществлен
* res\_specs — список типов устройств. Типы устройств, на которых следует показывать баннеры.
* resourceId — строка, идентификатор ресурса. Этот идентификатор необходимо передавать от iPad и Web-приложения во всех последующих запросах к серверу SDP, и не передавать в запросах к кеш-клиенту.
* result — результат выполнения метода.
* s\_id — идентификатор ТВ-передачи sheduleId, используемый для записи передач.
* scheduleId — идентификатор программы передач.
* sdate — дата и время начала, временная зона UTC. Время окончания — это время начала следующей передачи.
* sDate — дата начала проката, unixtime или загрузка ролика.
* serie — описание сериала.
* series — идентификатор сериала, если для данного сезона в SDP указано, к какому сериалу он относится.
* serviceAccountNumber — номер учетной записи.
* sessionId — Идентификатор сессии. Этот идентификатор необходимо передавать от iPad и Web-приложения во всех последующих запросах к серверу SDP, и не передавать в запросах к кеш-клиенту.
* size\_quota — идентификатор использования файловой квоты для хранения записей nPVR.
* srcPayId — Идентификатор платежа.
* start\_date — дата и время начала срока покупки сезона (unix).
* startDate — Дата-время начала передачи, unixtime.
* startDate — Дата-время начала показа баннера в формате Unix time.
* state — состояние учетной записи.
* status — код статуса платежа.
* status — состояние ролика, значение соответствует параметру status метода RTUGC/setStatus.
* stop\_date — дата и время окончания срока покупки (unix).
* temperature — температура воздуха в градусах Цельсия с точностью до 1 градуса
* terminalName — имя устройства.
* terminalType — Внешний идентификатор типа устройства (ExternalId) по справочнику SDP «Типы STB».
* to\_sec — Время окончания показа баннеров в секундах относительно суток, значение аналогично from\_sec.
* transport — протокол вещания.
* ugc\_film — описание ролика.
* ugc\_films — список роликов.
* ugcId — идентификатор ролика.
* UID — идентификатор устройства.
* usage\_date — дата и время покупки (unix).
* usedQuota — истрачено на момент запроса, в килобайтах.
* validDate — дата, до которой записанная передача будет доступна для просмотра для данной учетной записи.
* validDate — Дата-время окончания доступности передачи, после которой она будет удалена из SDP, unixtime.
* version — Версия описания сезона или сериала в кеше.
* version — Дата последнего изменения объекта в кеше. Может использоваться как версия описания ТВ-передачи для отслеживания изменений EPG на клиентском устройстве.
* videoServer — видеосервер, на который загружен ролик.
* videoserverIpAddress — IP-адрес видеосервера.
* videoserverPort — порт видеосервера.
* vs\_ip\_address — IP-адрес сервера для просмотра (редиректора).
* vs\_port — порт видеосервера.
* windSpeed — скорость ветра в м/с, точность 0,5 м/с.
* windType — направление ветра, 8 возможных значений: N, NE, E, SE, S, SW, W, NW.
* year — год.
* zabavaLogin — логин ZABAVA из учетной записи.

# Приложение. Схема архитектуры платформы «Интерактивное телевидение Ростелеком»



# Приложение. Схема проектируемой архитектуры платформы «Интерактивное телевидение Ростелеком» 2.0