2017

Microsoft TV White Spaces

DESIGN DOCUMENT

VERSION 1.0

Table of Contents

1. Overview and Description 13
2. FCC Rules and Regulations 13
3. OFCOM Rules and Regulations 13
4. Acronyms and Terms 14
5. Architecture 15
5.1 Configuration 20
5.1.1 Build Dependencies 20
5.1.1.1 Unity Application Block 21
5.1.1.2 Azure SDK for .NET 22
5.1.1.3 Configuration Transform 22
5.1.1.4 Security.Cryptography 22
5.1.1.5 Renci.SSHNet 22
5.1.1.6 Newtonsoft.Json 22
5.1.2 Static Runtime Settings 22
5.1.3 Dynamic Runtime Settings 23
5.2 Microsoft.Whitespace.PAWS 30
5.3 Microsoft.Whitespace.Sync 31
5.3.1 Database 31
5.3.1.1 DBSync File worker 31
Use of NextTransactionID in DB Sync File Worker 33
5.3.1.2 DBSync Manager 34
5.3.2 RGN1DBAdminInfo Azure Table 35
5.3.2.1 DBSync Poller 36
5.3.3 FCCDBSyncPoller 37
5.3.4 Region 39
5.3.4.1 CDBSFileType 40
5.3.4.2 FccRegionSync 40
5.3.4.3 RegionSyncConstants 40
5.3.4.4 ULSFileType 40
5.3.4.5 ULSMapping 40

5.3.4.6 ULSRecordType 40
5.4 Microsoft.Whitespace.Common 40
5.4.1 CacheHelper 41
5.4.1.1 FCCServiceCacheHelper 41
5.4.1.2 OFCOMServiceCacheHelper 41
5.4.2 Utilities 41
5.4.2.1 Conversion 41
5.4.2.2 GeoCalculations 41
5.4.2.3 MiscExtensions 41
5.4.2.4 Transform Coordinate 41
5.4.2.5 UnityExtensions Provides utility methods for the type IUnityContainer. 41
5.4.3 Validators 41
5.4.3.1 BasePawsValidator 41
5.4.3.2 BaseRegionManagementValidator 41
5.4.3.3 FccPawsValidator 41
5.4.3.4 FCCRegionManagementValidator 42
5.4.3.5 OfcomPawsValidator 42
5.4.3.6 OFCOMRegionManagementValidator 42
5.4.4 Value Provider 42
5.4.4.1 FccPawsValueProvider 42
5.4.4.2 OfcomPawsValueProvider 42
5.4.4.3 PawsRegionalValueProvider 42
5.4.5 Web API Helper 42
5.4.5.1 ApiDependencyResolver 42
5.4.5.2 UnityDependencyScope 42
5.4.6 Auditor 42
5.4.6.1 AzureAuditor 42
5.4.6.2 AzureConfig 43
5.4.6.3 AzureLogger 44
5.4.6.4 DatabaseCache 44
5.4.6.5 ErrorHelper 44
5.4.6.6 IAuditor 44

5.4.6.7 ICache 44
5.4.6.8 IConfiguration 44
5.4.7 Dalc Interfaces 44
5.4.7.1 IDalcAuditor 44
5.4.7.2 IDalcCommon 44
5.4.7.3 IDalcConfig 45
5.4.7.4 IDalcDBSync 45
5.4.7.5 IDalcDTTAvailabilitySync 45
5.4.7.6 IDalcIncumbent 45
5.4.7.7 IDalcLogger 45
5.4.7.8 IDalcPaws 45
5.4.7.9 IDalcPMSE 45
5.4.7.10 IDalcPmseSync 45
5.4.7.11 IDalcRegionSync 45
5.4.7.12 IDalcTerrain 45
5.4.7.13 IDalcUserManagement 45
5.4.8 DB Sync Interfaces 45
5.4.8.1 IDBSyncFile 45
5.4.8.2 IDBSyncFileIncr 46
5.4.8.3 IDBSyncManager 46
5.4.8.4 IDBSyncPoller 46
5.4.9 Whitespace Driver Interfaces 46
5.4.9.1 IDriverPaws 46
5.4.9.2 IDriverPMSE 46
5.4.9.3 IDriverRegionManagement 46
5.4.9.4 IDTTAvailabilitySync 47
5.4.10 Logger 47
5.4.10.1 ILogger 47
5.4.10.2 INativeMethodService 47
5.4.10.3 INetFileStream 47
5.4.10.4 IPawsBL 48
5.4.10.5 IPawsValidator 48

5.4.10.6 IPmseAssignment 48
5.4.10.7 IPMSEValidator 48
5.4.10.8 IRegionCalculation 48
5.4.10.9 IRegionManagementValidator 48
5.4.10.10 IRegionSync 48
5.4.10.11 IServiceCacheHelper 48
5.4.10.12 ITerrainElevation 48
5.4.10.13 IUserManager 48
5.4.10.14 Logger 48
5.4.10.15 NativeMethods 48
5.4.10.16 NativeMethodServiceClient 48
5.4.10.17 NetFileStream 48
5.4.11 Static Utils Class 49
5.4.12 XmlUnityConfigurationSection 49
5.5 Microsoft.Whitespace.Dalc 49
5.5.1 Microsoft WhiteSpce Dalc 49
5.5.1.1 AuditTableEntity 50
5.5.1.2 AzureDalc 50
5.5.1.3 AzureStorageHelper 50
5.5.1.4 LogTableEntity 50
5.6 Microsoft.Whitespace.Driver 50
5.6.1 Microsoft WhiteSpce Driver Interface 50
5.6.1.1 IDriverPaws 50
5.6.1.2 IDriverPMSE 50
5.6.1.3 IDriverRegionManagement 50
5.6.2 Microsoft.Whitespace.Driver 50
5.6.2.1 PawsDriver 50
5.6.2.2 RegionManagementDriver 50
5.7 Microsoft.Whitespace.DTTSync 50
5.7.1 DTTSync 50
5.7.1.1 DttDataAvailabilitySync 50
5.7.1.2 WorkerRole 50

5.8 Microsoft.Whitespace.Entities 51
5.8.1 Microsoft.Whitespaces.Entities.Versitcard 51
5.8.1.1 Address 51
5.8.1.2 EMail 51
5.8.1.3 FormattedName 51
5.8.1.4 Name 51
5.8.1.5 Organization 51
5.8.1.6 Telephone 51
5.8.1.7 Title 51
5.8.1.8 VCard 51
5.8.2 .Entities 51
5.8.2.1 Address 51
5.8.2.2 AntennaCharacteristics 51
5.8.2.3 AntennaPatternItem 51
5.8.2.4 Area 52
5.8.2.5 AuthorizedDeviceRecord 52
5.8.2.6 AvailableSpectrumConverter 52
5.8.2.7 AvailableSpectrumRequest 52
5.8.2.8 AvailableSpectrumRequestBase 52
5.8.2.9 BaseJsonConverter 52
5.8.2.10 BasePawsClass 52
5.8.2.11 BatchAvailableSpectrumConverter 52
5.8.2.12 BatchAvailableSpectrumRequest 52
5.8.2.13 BatchAvailableSpectrumRequestBase 52
5.8.2.14 BlockedType 52
5.8.2.15 CacheObjectPMSEAssignment 52
5.8.2.16 CacheObjectTvEngdata 52
5.8.2.17 Calendar 52
5.8.2.18 CDBSAntennaPattern 52
5.8.2.19 CDBSApplication 52
5.8.2.20 CDBSAppTracking 52
5.8.2.21 CDBSFacility 53

5.8.2.22 CDBSTableEntity 53
5.8.2.23 CDBSTvEngData 53
5.8.2.24 ChannelInfo 53
5.8.2.25 Check 53
5.8.2.26 CircleArea 53
5.8.2.27 Constants 53
5.8.2.28 Contour 53
5.8.2.29 ContourDetailsInfo 53
5.8.2.30 ContourPointItem 53
5.8.2.31 DatabaseSpec 53
5.8.2.32 DatabaseSpecConverter 53
5.8.2.33 DBAdminInfo 53
5.8.2.34 DbUpdateSpec 53
5.8.2.35 DbUpdateSpecConverter 53
5.8.2.36 DeviceCapabilities 53
5.8.2.37 DeviceDescriptor 54
5.8.2.38 DeviceDescriptorConverter 54
5.8.2.39 DeviceId 54
5.8.2.40 DeviceOwner 54
5.8.2.41 DeviceOwnerConverter 54
5.8.2.42 DeviceRegistrationMap 54
5.8.2.43 DeviceValidity 54
5.8.2.44 DeviceValidityConverter 54
5.8.2.45 DeviceValidityRequest 54
5.8.2.46 DeviceValidityRequestBase 54
5.8.2.47 DistanceUnit 54
5.8.2.1 Distance 54
5.8.2.2 DistinctDTTQueryParams 54
5.8.2.3 DistinctOSGLocations 54
5.8.2.4 DttData 54
5.8.2.5 DTTDataAvailability 54
5.8.2.6 DTTQueryParams 55

5.8.2.7 DTTSquare 55
5.8.2.8 ElevatedAccessRequest 55
5.8.2.9 ElevationCacheEntry 55
5.8.2.10 Ellipse 55
5.8.2.11 EllipseConverter 55
5.8.2.12 Email 55
5.8.2.13 Event 55
5.8.2.14 EventRecurrence 55
5.8.2.15 EventTime 55
5.8.2.16 EventTimeConverter 55
5.8.2.17 FixedTVBDRegistration 55
5.8.2.18 FortranDoubleArray 55
5.8.2.19 FrequencyRange 55
5.8.2.20 FrequencyRangeConverter 55
5.8.2.21 GeoLocation 55
5.8.2.22 GeoLocationConverter 56
5.8.2.23 GeoSpectrumSchedule 56
5.8.2.24 GeoSpectrumScheduleConverter 56
5.8.2.25 GeoSpectrumSpec 56
5.8.2.26 GetChannelsResponse 56
5.8.2.27 IAvailableSpectrumRequest 56
5.8.2.28 IBatchAvailableSpectrumRequest 56
5.8.2.29 IDeviceValidityRequest 56
5.8.2.30 IInitRequest 56
5.8.2.31 IInterferenceQueryRequest 56
5.8.2.32 Incumbent 56
5.8.2.33 IncumbentType 56
5.8.2.34 InitializedDeviceDetails 56
5.8.2.35 InitRequest 56
5.8.2.36 InitRequestBase 56
5.8.2.37 INotifyRequest 56
5.8.2.38 InterferenceQueryConverter 57

5.8.2.39 InterferenceQueryRequest 57
5.8.2.40 InterferenceQueryRequestBase 57
5.8.2.41 IParameters 57
5.8.2.42 IRegisterRequest 57
5.8.2.43 JsonSerialization 57
5.8.2.44 KeyHoleContourItem 57
5.8.2.45 Location 57
5.8.2.46 LpAuxLicenseInfo 57
5.8.2.47 LPAuxRegistration 57
5.8.2.48 MVPDCallSignsInfo 57
5.8.2.49 MVPDRegistration 57
5.8.2.50 NextTransactionID 57
5.8.2.51 NotifyConverter 57
5.8.2.52 NotifyRequest 57
5.8.2.53 NotifyRequestBase 57
5.8.2.54 Organization 58
5.8.2.55 OSGLocation 58
5.8.2.56 ParameterConverter 58
5.8.2.57 ParameterMissingException 58
5.8.2.58 Parameters 58
5.8.2.59 ParametersBase 58
5.8.2.60 ParsePollRequestResult 58
5.8.2.61 PawsResponse 58
5.8.2.62 PawsResponseConverter 58
5.8.2.63 PmseAssignment 58
5.8.2.64 Point 58
5.8.2.65 PointConverter 58
5.8.2.66 Polygon 58
5.8.2.67 PolygonConverter 58
5.8.2.68 Position 58
5.8.2.69 PowerUnit 58
5.8.2.70 Power 59

5.8.2.71 ProtectedDevice 59
5.8.2.72 ProtectedDeviceType 59
5.8.2.73 ProtectedEntity 59
5.8.2.74 ProtectedEntityWithEvent 59
5.8.2.75 QuadrilateralArea 59
5.8.2.76 RadialHAT 59
5.8.2.77 RadiationCenter 59
5.8.2.78 RegionAccess 59
5.8.2.79 RegionManagementResponse 59
5.8.2.80 RegionManagementResponseConverter 59
5.8.2.81 RegisteredDevice 59
5.8.2.82 RegisterRequest 59
5.8.2.83 RegisterRequestBase 59
5.8.2.84 RegistrationDisposition 59
5.8.2.85 Request 59
5.8.2.86 RequestConverter 60
5.8.2.87 Result 60
5.8.2.88 ResultConverter 60
5.8.2.89 RulesetInfo 60
5.8.2.90 RulesetInfoConverter 60
5.8.2.91 RuleSetInfoEntity 60
5.8.2.92 SearchCacheRequestType 60
5.8.2.93 ServiceCacheObjectType 60
5.8.2.94 ServiceCacheRequestParameters 60
5.8.2.95 Settings 60
5.8.2.96 Spectrum 60
5.8.2.97 SpectrumConverter 60
5.8.2.98 SpectrumProfile 60
5.8.2.99 SpectrumSchedule 60
5.8.2.100 SpectrumScheduleConverter 60
5.8.2.101 SpectrumSpec 60
5.8.2.102 SquareArea 60

5.8.2.103 SyncRow 61
5.8.2.104 SyncTable 61
5.8.2.105 Telephone 61
5.8.2.106 TempBASRegistration 61
5.8.2.107 TransactionInfo 61
5.8.2.108 TVReceiveSiteRegistrations 61
5.8.2.109 TvSpectrum 61
5.8.2.110 ULSRecord 61
5.8.2.111 UsedSpectrum 61
5.8.2.112 User 61
5.8.2.113 ValidationHelper 61
5.8.2.114 Vcard 61
5.8.2.115 VcardConverter 61
5.8.2.116 LocationRect 61
5.8.2.117 RegionPolygon 61
5.8.2.118 RegionPolygonsCache 61
5.9 Microsoft.Whitespace.InterpolationLibrary 62
5.9.1.1 BivariateInterpolation 62
5.9.1.2 CurveCalculator 62
5.9.1.3 F2DArray 62
5.9.1.4 FArray 62
5.9.1.5 FCCDataMatrix 62
5.9.1.6 SharedVariables 62
5.9.1.7 ZA 62
5.9.1.8 ZAB 62
5.9.1.9 ZB 62
5.9.1.10 ZX 62
5.9.1.11 ZXY 62
5.9.1.12 ZY 62
5.10 Microsoft.Whitespace.NativeCodeHostService 63
5.10.1 NativeCodeHostServicef 63
5.10.1.1 INativeMethodService 63

5.10.1.2 NativeMethods 63
5.11 Microsoft.Whitespace.PMSESync 63
5.11.1 PMSESync 63
5.11.1.1 ProcessPmseAssignmentFiles 63
5.11.1.2 WorkerRole 63
5.12 Microsoft.Whitespace.RegionCalculation 63
5.12.1 RegionCalculation 63
5.12.1.1 FccCalculation 64
5.12.1.2 Protection of Incumbents 64
5.12.1.3 OfcomCalculation 66
5.12.2 Propagation 66
5.12.2.1 ClutterEnvironment 66
5.12.2.2 OFCOMMatrix 66
5.12.2.3 PMSEEquipmentType 66
5.12.2.4 PMSEUsageNature 66
5.12.2.5 OperationalParameter Enumeration of device operational parameters. 67
5.13 Microsoft.Whitespace.RegionManagement 67
5.13.1 RegionManagement 67
5.14 Microsoft.Whitespace.Terrain 67
5.14.1 Terrain 67
5.14.1.1 ClutterDatasetReader 67
5.14.1.2 ElevationCache 67
5.14.1.3 OrdnanceSurveyOSReader 67
5.14.1.4 SRTMReader 67
5.14.1.5 USGS1ArcReader 67
5.15 Microsoft.Whitespace.Terrain Cache 67
5.15.1.1 Caching 68
5.16 Web Service 69
5.16.1 PAWS Web Service 69
5.16.1.1 Sequence Diagrams 70
5.16.2 Region Management Web Service 76
5.16.2.1 Sequence Diagrams 77

5.16.3 DB Admin Web Service 97
6. Test 97
6.1 Sample Test Configuration Implementation 97
7. Application Tables 99
7.1 Organization of Tables Within the Application 99
7.2 List of Tables Used 99
7.3 Ofcom DTT Availability Data 103

1. Overview and Description

Microsoft TV White Spaces database enables secondary users the ability to use the TV White Spaces spectrum in order to provide for overall spectrum utilization. The design document outlines the technical implementation of the webservices and APIs to support white space access for FCC in the US and Ofcom in the UK.

The primary objectives of this design are as follows:

- Obtain FCC certification as a white space database administrator
- Obtain Ofcom Certification as a white space database administrator.
- Create a design that is flexible to add additional regions beyond the United States and the UK.

2. FCC Rules and Regulations

The FCC's TVWS Rules are documented in the following memorandums:

- FCC 08-260 – November 4, 2008: http://fjallfoss.fcc.gov/edocs_public/attachmatch/FCC-08-260A1.pdf
- FCC 10-174 – September 23, 2010: http://fjallfoss.fcc.gov/edocs_public/attachmatch/FCC-10-174A1.pdf
- FCC 12-36 – April 4, 2012: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-12-36A1.pdf

In order to achieve FCC Certification, there are three Major components of White Space DB design which include:

- Channel Calculations for White Spaces as specified in: http://apps.fcc.gov/ecfs/document/view?id=7022134609
- White space Database-to-Database Synchronization as specified in: http://apps.fcc.gov/ecfs/document/view?id=7520936333(Erratum to document located at http://apps.fcc.gov/ecfs/document/view?id=7022128682).
- Protocol to Access White Space Database: http://tools.ietf.org/html/draft-ietf-paws-protocol-12

3. OFCOM Rules and Regulations

The OFCOM's TVWS Rules are documented in the following:

- http://stakeholders.ofcom.org.uk/binaries/consultations/white-space-coexistence/summary/white-spaces.pdf
- http://stakeholders.ofcom.org.uk/binaries/consultations/white-space-coexistence/annexes/technical-report.pdf
- *Digital dividend review: A statement on our approach to awarding the digital dividend*", 13 December 2007, http://stakeholders.ofcom.org.uk/consultations/ddr/statement/;
- *Digital Dividend: Geolocation for Cognitive Access, a discussion document, 17 November 2009 http://stakeholders.ofcom.org.uk/binaries/consultations/cogaccess/summary/cogaccess.pdf;*
- *Implementing geolocation: consultation*, November 2010, http://stakeholders.ofcom.org.uk/consultations/geolocation/summary;
- *Implementing geolocation: Summary of consultation responses and next step,* September 2011, http://stakeholders.ofcom.org.uk/consultations/geolocation/statement/
- *TV white spaces: A consultation on white space device requirements*, 22 November 2012, http://stakeholders.ofcom.org.uk/consultations/whitespaces/

4. Acronyms and Terms

Acronym/Term	Definition
BAS	Broadcast Auxiliary Service
CDBS	Consolidated Database System
EAS	Equipment Authorization System
FCC	Federal Communications Commission (US Government)
Fixed Devices	A TVBD intended to operate at a fixed location such as an outdoor access point.
HAAT	Height Above Terrain
LM	Land Mobile
MVPD	Multi-channel Video Program Distributor
PAWS	Protocol to Access White Space DB
Personal/Portable Devices	A TVBD that can operate at mobile locations that may change.
PMSE	Program Making and Special Event
Poller	To approach other components with a request for information.
RSSI	Received Signal Strength Indication
TVBD	TV Band Devices
TVWS	TV White Spaces
ULS	Universal License System
WSDBA	White Space Database Administrator

5. Architecture

FCC Implementation

The following diagram outlines the major components that makeup the FCC implementation of the white space database

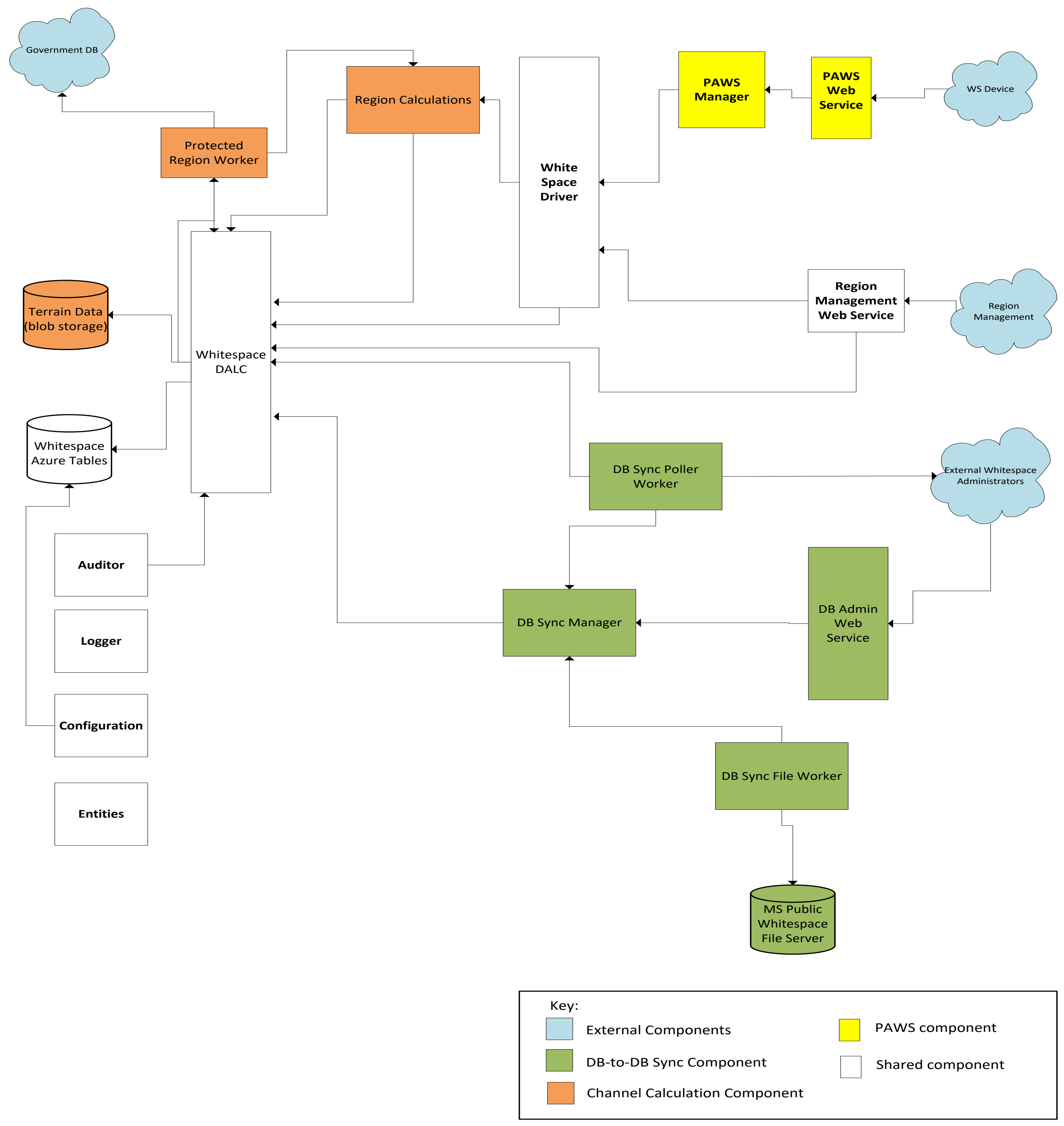


Figure 1 White Space DB Overview – FCC

Ofcom Implementation

The following diagram outlines the major components that makeup the Ofcom implementation of the white space database

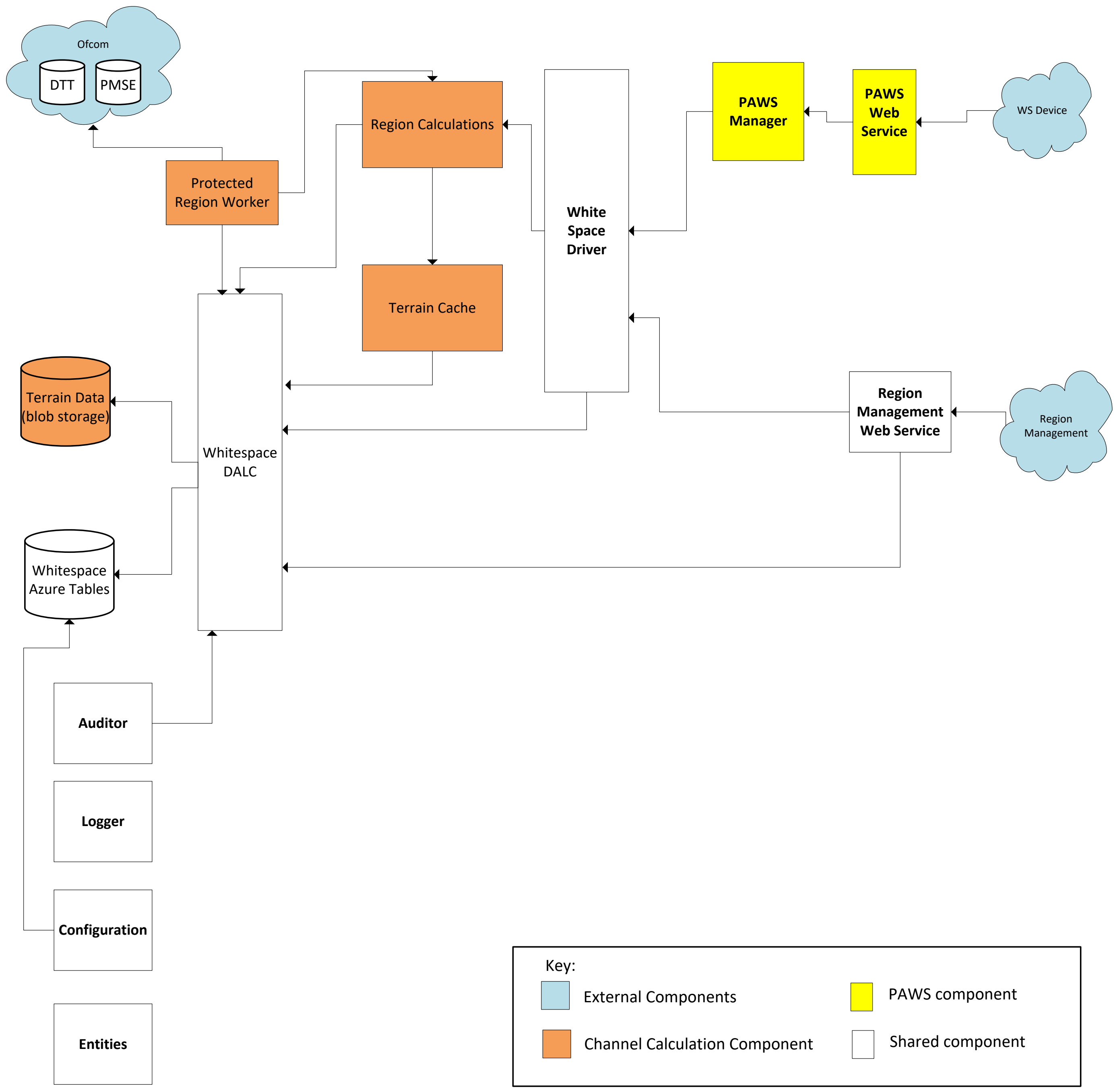


Figure 2 White Space DB Overview – Ofcom

Component	Description
PAWS Web Service	Web service that exposes the Protocol to Access White Space DB (PAWS). PAWS web methods exposed include: • INIT_REQ/INT_RESP • REGISTRATION_REQ/REGISTRATION_RESP • AVAIL_SPECTRUM_REQ/AVAIL_SPECTRUM_RESP • AVAIL_SPECTRUM_BATCH_REQ/AVAIL_SPECTRUM_BATCH_RESP • SPECTRUM_USE_NOTIFY/SPECTRUM_USE_RESP • DEV_VALID_REQ/DEV_VALID_RESP • INTERFERENCE_QUERY (This method applies to the Ofcom PAWS service only)
PAWS Manager	The PAWS Manager is responsible for: • Validating and Parsing PAWS JSON parameters • Validating device Ids • Calling into White Space Driver to get available spectrum • Converting spectrum request responses to JSON.
Region Management Web Service	Web Service provided for region specific administrative management. Functionally exposed includes: • ExcludeIds() – Exclude a list of Ids from reserving white spaces (Ids will be region specific such as FCC Id). • ExcludeChannel() – Exclude white space registration for a given list of channels in a given location (a set of lat/long points will be passed in for the excluded channels). • RegisterDevice() – allows a user to add a new incumbent for protection from interference from White Space devices. The incumbent types that can be registered are MVPD, LP Aux (both licensed and unlicensed)m and Temp BAS. • DeleteIncubment() – Deletes a given incumbent from the registration database. • GetIncumbents() – returns a list of incumbents for the current region (if user is an admin or super admin, then all incumbents are returned – if user is a Licensee, then only incumbents associated with the current user will be returned). • GetChannelList – returns a list of all available “free” channels at a given location. The returned list of channels will indicate the type(s) of device that are permissible at that location (i.e. fixed, mobile, none). • GetDeviceList() – returns a list of all protected devices (incumbents such as TV Towers, wireless mics, …) that occupy channels at a given location. • GetULSCallSigns() – returns a list of ULS call signs that can be used for registration of licensed LPAux devices. • GetULSFileNumbers() - returns a list of valid ULS file numbers that can be used when registering an unlicensed LPAux device. • SearchMVPDCallSigns() – returns a list of valid parent call signs that can be used to register a new MVPD incumbent at a given location • GetPublicData() – returns a list of incumbents (ProtectedEntity objects) that is contained in the whitespace database. Used to provide public visibility to the contents of the whitespace database

- GetPublicDataWithEvents() – returns a list of incumbents (ProtectedEntityWithEvents objects) that is contained in the whitespace database. Used to provide public visibility to the contents of the whitespace database.
- .GetAuthorizedDeviceModels() – returns a list of the wireless device models that have been approved by the FCC
- GetContourData() – returns the set of 360 lat/long pairs that define the protection contours for a particular incumbent.

The following table shows the access level required for each method. Note that these access levels are not enforced by the RegionManagement service; the front end components that invoke RegionManagement services are responsible for ensuring that the user has the required access level to execute a given method of the service.

Certain methods of the service use the user's access level to determine the content that is to be included in the response.

Method	Required Access Level
ExcludeIds	Super Admin, Admin
ExcludeChannel	Super Admin or Admin
ExcludeRegion	Super Admin or Admin
RegisterUser	None
RegisterDevice	Super Admin, Admin or Licensee
DeleteIncubment	Super Admin, Admin or Licensee (note that a licensee will only be able to delete an incumbent which it had previously registered).
GetIncumbents	Super Admin, Admin or Licensee (note that a licensee will only be able to retrieve data for incumbents thye have registered).
GetChannelList	None
GetDeviceList	Super Admin, Admin, Licensee, or Device Vendor
GetULSCallSigns	None
GetULSFileNumbers	None
SearchMVPDCallSigns	None
GetPublicData	None
GetPublicDataWithEvents	None
GetAuthorizedDeviceModels	None
GetContourData	None

User Manager	Component used to validate registered users and their access level (5 different access levels which are: Super Admin, Admin, Licensee, Device Vendor and Portal User). Note: Components related to user management are not currently used within the whitespace database solution. These functions are currently performed by the front end components that use the whitespace database. The UserManager components have been left in the solution in case they are needed in the future and existing references to them have been left in the design document.
White Space Driver	Primary access point for: • Accessing/Updating incumbent data • Instantiating the correct region-specific Channel Calculator. • Processing and culling incumbents. • Updating Device Exclusion List or Exclusion Regions from spectrum usage.
Region Calculations	Responsible for calculating the white space protection contours, RSSI values and distances used for determining channel availability.
Terrain Cache	Cache of the terrain data tiles (if the data is not cached, then the tile will be retrieved from the blob storage via the whitespace DALC).
Whitespace DALC	Data Access Layer Component for accessing the azure tables and blob storage.
Terrain Data	Blob storage that contains all the terrain files.
White Space Tables	Contains all the whitespace azure tables including: • TV_Eng_Data • Ant_Pattern • Application • App_Tracking • Facility • LMbCast • LMComm • LMPriv • Micro • Paging • MVPD_Registration • TV_Receive_Site_Registration • LP_Aux_Registration • Fixed_TVBD_Registration • Temp_BAS_Registration • Audit • DBAdminInfo
Protected Region Worker	Azure worker responsible for: • Download region specific data used for calculating protected entity contours. • Download white list/black list of valid devices (for instance, list of valid serial number from the FCC EAS server). • Calculate contour calculations for new or updated incumbents.

Entities	Contains definition of all the shared objects (Calendar, Position, Registrations, Device Ids, …).
Logger	Helper class used to log data. Four level of logging levels will be supported (verbose, info, warning, error) which willuse the Enterprise Application Logging Framework.
Auditor	Helper class used to create an "Audit" trail of "major" white space events. Audit events will include: • All Web API calls along with success and failure of the call (web calls include PAWS, PMSE, Region Management, and DB Admin). • Success or Failure every time the "Protected Region Sync Worker" is executed. • Success or Failure every time the DB Sync File Worker is launched. • Success or Failure every time the DB Sync Poller Worker is launched.
Configuration	Helper class used for reading and writing application configuration data.
DB Administrator Web Service	Access point for external WSDBA in order to "fast poll" for registration changes within the MS white space database. Note: currently, this component applies only to the FCC implementation of the database. Ofcom does not require a synchronization process between whitespaces DBAs.
DB Sync Manager	Module used to handle the main database-to-database synchronization management. The main responsibilities include: • Generating and singing XML files containing past registrations. • Validating XML signatures from external whitespace administers (spectrum bridge, iconnectiv, …). • Parses external XML files and syncing with MS Whitespace DB.
DB Sync Poller Worker	An azure worker role that will continuously poll external WSDBA for any registration changes.
DB Sync File Worker	An azure worker role that will wake up periodically to generate two kinds of XML files: • The first file will contain all past registration (generated daily). • The 2nd file will be generated hourly and will only contain incremental changes from the previous day. These XML files will then be zip compressed and placed on the web server.
MS Public Whitespace File Server	Public accessible server where external WS Administrators can retrieve zip files generated by the "DB Sync File Worker".

5.1Configuration

5.1.1 Build Dependencies

The whitespace build environment will be dependent on the following components:

- Visual Studio 2013

- Unity Application Block 3.5
- Azure SDK for .NET
- Configuration Transform
- Security.Cryptography
- Renci.SSHNet
- Newtonsoft.Json

5.1.1.1 Unity Application Block

The Unity Application Block is a lightweight, extensible dependency injection container that supports constructor injection, property injection, and method call injection (the whitespace design relies heavily on constructor injection). The following code shows how to use the Unity framework within the whitespace design to dynamically instantiate the default component that implements the IRegionCalculation interface:

```
IUnityContainer container = Whitespace.Common.Utils.Configuration.CreateUnityContainer();

IRegionCalculation calc = container.Resolve<IRegionCalculation>();
```

In order to resolve all of the abstract interfaces to concrete classes, an XML config mapping is used within the CreateUnityContainer method (the config setting is read from IConfiguration[“UnityConfig”] value). The XML schema is defined here. The following sample shows an example how the IAuditor, ILogger, and IRegionCalculation interfaces are mapped:

```
<unity>
<containers>
<container>
<register
name='Auditor'
type='Microsoft.Whitespace.Common.IAuditor, Microsoft.Whitespace.Common'
mapTo='Microsoft.Whitespace.Common.AzureAuditor, Microsoft.Whitespace.Common'>
</register>
<register
name='Log'
type='Microsoft.Whitespace.Common.ILogger, Microsoft.Whitespace.Common'
mapTo='Microsoft.Whitespace.Common.Logger, Microsoft.Whitespace.Common'>
<lifetimetype='ContainerControlledLifetimeManager' />
</register>
<register
name='Calc'
type='Microsoft.Whitespace.Common.IRegionCalculation, Microsoft.Whitespace.Common'
mapTo='Microsoft.Whitespace.RegionCalculation.USCalculations, Microsoft.Whitespace.RegionCalculation'>
</register>
</container>
</containers>
</unity>
```

In the above sample code, note that the ILogger interface has a "lifetime" setting that indicates that the logger is a singleton within each instance of IUnityContainer (the logger has a "Transaction Id" that gets initialized during construction – this then allows for all ILogger instances returned from the same unity container object to have the exact same logged Transaction Id which makes it easy to track calls amongst components).

Note that the IConfiguration is the only component that is not resolved via the Unity Container.

The Unity Application block can be installed via NuGet or from the following web site: http://msdn.microsoft.com/en-us/library/ff647202.aspx.

5.1.1.2 Azure SDK for .NET

The current Azure SDK release for .NET is version 2.3 and can be located here: http://www.microsoft.com/en-us/download/details.aspx?id=42317

5.1.1.3 Configuration Transform

The Configuration Transform allows for custom settings in App.config based on the current build configuration (this is identical to web.config transform built into VS 2012 which is documented here: http://blogs.msdn.com/b/webdev/archive/2009/05/04/web-deployment-web-config-transformation.aspx). The Configuration Transform can be installed from NuGet or downloaded from: http://visualstudiogallery.msdn.microsoft.com/579d3a78-3bdd-497c-bc21-aa6e6abbc859.

5.1.1.4 Security.Cryptography

This package is used to manage xml signatures for the exchange of registration data with other Whitespaces Database Administrators in the database synchronization process. This package can be downloaded from: http://clrsecurity.codeplex.com/wikipage?title=Security.Cryptography.dll

5.1.1.5 Renci.SSHNet

This package provides an implementation of an SFTP client that is used for retrieval of registration information from other Whitespaces Database Administrators when the File Transfer method is required.This package can be downloaded from: https://sshnet.codeplex.com/

5.1.1.6 Newtonsoft.Json

This package provides methods to serialize/deserialize objects to and from json for communication for communication with users of the RegionManagement service methods within the whitespace application.

5.1.2 Static Runtime Settings

The following table describes all of the required settings in the application's config file (web.config or app.config):

Field Name	Default Value	Description
DefaultConfig	"Microsoft. Whitespace.Common.Config"	The fully qualified class name of the class that implements the IConfiguration interface.
RegionId	"1"	Identifies the current region. **Code** **Region**

		1 United States **5** UK
Authority		Identifies the governing authority for the region. Current values are US, GB
DBConnectionString		Connection string for Azure Storage account
UnityContainerNames		Identifies names of containers that are used to resolve unity references. Names are built dynamically based on region id

Note that values in the above table are considered “read-only” and cannot be changed at runtime.

5.1.3 Dynamic Runtime Settings

The following table describes all of the in the configuration table. These values are stored in a separate Azure table for each deployment of the application. These values are retrieved and set via the IConfiguration interface and are described below:

RGN1 Config Settings:

Partition Key	Row Key (Setting Name)	Description	Current Value
Cache	MaxCacheExpiryTime_Hour	Cache expiry time in hours	24
Cache	TimeToEvictCache_Min	Interval, in minutes, to check if items in cache need to be removed if not used for given time	15
Cache	UseAzureCache	Boolean value indicating whether AzureCache is to be used	FALSE
Cache	maxLocalCacheSize_MB	Maximum in memory cache size for single item.	1024
Common	Culture	Used to control date format in request and response objects	en-US
InMemoryCache	RegionPolygons_CacheIntervalMin	Interval between next cache refresh from the table.	12
Configuration	SyncTimer_MilliSec	Updates configuration settings in given time interval	300000
DbSyncService	ClientSettingsProvider.ServiceUri		
DbSyncService	DBSyncVersionNo	Current version of the DB Sync specification that is supported by the application	1.2
DbSyncService	DbSyncFileInterval_MilliSec	Time interval between next file poller call.	5000

DbSyncService	FTPFileDownloadPath	Relative path to location of FTP file downloads for inbound process	FTPDownloads
DbSyncService	LogMessage	Used in logger testing	Unit Testing Azure logger
DbSyncService	OutBoundXMLTemplate	Relative path to location of template file for outbound XML file	XSDFiles\OutBound-Template.xml
DbSyncService	PollInerval	Interval for polling of other WSDBAs for real time DB Synx	900000
DbSyncService	PrivateCertificate	Relative path to location of certificate for DBSync service	Certificates\server.cer
DbSyncService	RealtimePollSchemaFile	Relative path to location of schema file for DB sync real time poller	XSDFiles\RealtimePoll.xsd
DbSyncService	Registrar	Value used to identify our database when communicating with other WSDBAs, and to idetnify our registration records	MSFT1
DbSyncService	RegistrationRecordEnsembleFile	Relative path to location of RegistrationRecordEnsemble XML file.	XSDFiles\RegistrationRecordEnsemble.xml
DbSyncService	RegistrationsStartDate	Starting filter date for retrieval of registration data for DB Sync process	2013-01-01T00:00:00Z
DbSyncService	RegistrationsXSDFilePath	Relative path to location of XSD file for schema validation	XSDFiles\Registrations.xsd
DbSyncService	SignCertificate	Relative path to location of sign certificate for DBSync service	Certificates\Server.pfx
DbSyncService	SignedOutputFilePath	Relative path to location of signed output XML file for outbound process.	OutBoundXMLFiles
DbSyncService	WSDBA	Value used to identify Microsoft when saving registration data on the database	MSFT
FCCChannelCalculations	CMRSAdjacentChannelDistance_KM	Distance in KM for protecting PLCMRS entries on adjacent channel	51
FCCChannelCalculations	CMRSCoChannelDistance_KM	Distance in KM for protecting PLCMRS entries on co channel	54
FCCChannelCalculations	KeyholeInnerAdjChannelDistance_KM	Radius in KM of inner circle of keyhole protection for adjacent channel.	2
FCCChannelCalculations	KeyholeInnerCoChannelDistance_KM	Radius in KM of inner circle of keyhole protection for co channel.	8
FCCChannelCalculations	KeyholeOuterAdjChannelDistance_KM	Adjacent channel protection distance in KM for keyhole arc	20

		extending from inner circle to end point on arc.	
FCCChannelCalculations	KeyholeOuterCoChannelDistance_KM	Co channel protection distance in KM for keyhole arc extending from inner circle to end point on arc.	80
FCCChannelCalculations	LpAuxFixedDeviceDistance_KM	Minimum distance in KM for lpaux device when wsd device type is Fixed.	1
FCCChannelCalculations	LpAuxPortableDeviceDistance_KM	Minimum distance in KM for lpaux device when wsd device type is portable.	0.4
FCCChannelCalculations	MVPDSearchDistance_KM	Search radius in KM for MVPD devices	200
FCCChannelCalculations	RadioAstronomyDistance_KM	Distance for protection of Radio Astronony sites from white space device interference	2.4
FCCChannelCalculations	TBandAdjChannelDistance_KM	Distance for protection of TBand services from white space device interfence on adjacent channels.	131
FCCChannelCalculations	TBandCoChannelDistance_KM	Distance for protection of TBand services from white space device interfence onco-channel.	134
FCCChannelCalculations	TvStationDeviceAdjChannel_HAAT10_KM	Required device separation in kilometers for adjacent channel based on HAAT value	0.7
FCCChannelCalculations	TvStationDeviceAdjChannel_HAAT30_KM	Required device separation in kilometers for adjacent channel based on HAAT value	1.2
FCCChannelCalculations	TvStationDeviceAdjChannel_HAAT3_KM	Required device separation in kilometers for adjacent channel based on HAAT value	0.4
FCCChannelCalculations	TvStationDeviceAdjChannel_HAAT50_KM	Required device separation in kilometers for adjacent channel based on HAAT value	1.8
FCCChannelCalculations	TvStationDeviceCoChannel_HAAT10_KM	Required device separation in kilometers for co channel based on HAAT value	7.3
FCCChannelCalculations	TvStationDeviceCoChannel_HAAT30_KM	Required device separation in kilometers for co channel based on HAAT value	11.1
FCCChannelCalculations	TvStationDeviceCoChannel_HAAT3_KM	Required device separation in kilometers for co channel based on HAAT value	4
FCCChannelCalculations	TvStationDeviceCoChannel_HAAT50_KM	Required device separation in kilometers for co channel based on HAAT value	14.3

FCCChannelCalculations	TvStationSearchDistance_KM	Specifies the size of the square that is to be used to search for TV stations that may be impacted by operation of a white space device at a given lat/long. Used for filtering of TV stations when determining available channels.	310
PAWSApi	DateFormat	Date format used in PAWS requests and responsesl	MM'/'dd'/'yyyy
PAWSApi	PawsApiVersion	Current version of PAWS API supported by the application	1
PAWSApi	PawsDeviceValidationUrl	URL to be contacted to validate PAWS devices	https://apps.fcc.gov/OETLabServices/ getFCCIDList?fccId=
PawsAPI	CDBSUSMexicoTvEngData_CacheInterval	Interval between next cache refresh from table.	3600000
PawsRegistration	LastRegDate	Last registration date of incumbent	7/24/2014
PawsRegistration	LastRegSeq	Last generated sequence number for PAWS registration	8757
RegionManagementApi	MVPDTTvStationQueryDistance_KM	Search radius for MVPD devices	150
RegionManagementApi	RegionManagementApiVersion	Region Management API version	1
RegionSync	AuthorizeDeviceURL	Url used in FCCRegionSync process to retrieve list of authorized devices	https://apps.fcc.gov/OETLabServices/ getWhitespaceAuthorizations?beginDate=01-01-2000&endDate=
RegionSync	IsULSIncremental		FALSE
RegionSync	RegionSyncSynchronizationTimer_Day	Region Sync service timer in days	1
RegionSync	RunCDBSProcess	Boolean flag indicating whether CDBS process is to be executed. Should only be set to 0 for special purposes in test environment.	1
RegionSync	RunContourProcess	Boolean flag indicating whether contour calculation process is to be executed. Should only be set to 0 for special purposes in test environment.	1

RegionSync	RunFCCDeviceAuthorizationProcess	Boolean flag indicating whether FCC device authorization process is to be executed. Should only be set to 0 for special purposes in test environment.	0
RegionSync	RunULSProcess	Boolean flag indicating whether FCC ULS data retrieval process is to be executed. Should only be set to 0 for special purposes in test environment.	0
Terrain	DBConnectionStringTerrain	Connection string to be used for retrieval of terrain data	DefaultEndpointsProtocol=https;AccountName=[MyAccountName];AccountKey=[MyAccountKey]
Terrain	arc1DataDirectory	Azure storage blob container name for USGS1Arc data	usgs1arcsecondgridfloat20130806
Terrain	arc2DataDirectory	Azure storage blob container name for USGS2Arc data	usgs2arcsecondgridfloat20130806
WSDValidatation	ValidateWSDLocation	Boolean flag indicating whether requested WSD location should be validated using Region Polygons cache for FCC.	TRUE

<u>RGN5 Config Settings:</u>

Partition Key	Row Key (Setting Name)	Description	Current Value
Cache	*MaxCacheExpiryTime_Hour*	Cache expiry time in hours	24
Cache	*TimeToEvictCache_Min*	Interval, in minutes, to check if items in cache need to be removed if not used for given time	15
Cache	*UseAzureCache*	Boolean value indicating whether AzureCache is to be used	FALSE
Cache	*maxLocalCacheSize_MB*	Maximum in memory cache size for single item.	1024
InMemoryCache	RegionPolygons_CacheIntervalMin	Interval between next cache refresh from the table.	12
Common	*Culture*	Used to control date format in request and response objects	en-US
Configuration	*SyncTimer_MilliSec*	Updates the configuration settings from config settings table in given time interval	300000
DTTSync	*DttSyncTableSuffix*	Suffix used for tables names to store DTT availability data. This setting has been used during the development	v01112013

		process to enable storing of DTT data in two different formats	
DttSync	*DttSyncSourceContainer*	Azure storage blob container name for DTT data	ofcom-dtt
DttSync	*DttSyncSynchronizationTimer_Day*	DTT sync service timer	1
ExchangeServer	*Domain*	Domain name for Exchange server user account	Redmond
ExchangeServer	*UserID*	Username for Exchange server account	Ukpmse
InterferenceQuery	*Body*	Email body text for interference query response	Interference Query Response CSV attached
InterferenceQuery	*CSVFileName*	Output CSV file name for interference query response	Dataset
InterferenceQuery	*FromAddress*	Email address to be used for sending the interference query response	
InterferenceQuery	*Password*	Password for the email account to be used for sending the interference query response	
InterferenceQuery	*ReferenceFileName*		
InterferenceQuery	*SMTPMail*	SMTP for email account to be used for sending the interference query response	smtp.gmail.com
InterferenceQuery	*SMTPPort*	SMTP Port number for email account to be used for sending the interference query response	587
InterferenceQuery	*Subject*	Email subject line	Interference Query Response
InterferenceQuery	*UserName*	Email address to be used for sending the interference query response	
KillSwitch	*WSDBEnabled*	Used to turn off all white space activity for Ofcom. Set to false to disable all white space access	TRUE
OFCOMChannelCalculations	*AntennaGain_TypeA*	Default Antenna Gain value for Type-A WSDB device	10
OFCOMChannelCalculations	*AntennaGain_TypeB*	Default Antenna Gain value for Type-B WSDB device	0
OFCOMChannelCalculations	*DefaultDeviceEmissionClass_DeviceParam*	Default emission class to be used as per schedule 4	5
OFCOMChannelCalculations	*DefaultHeightUncertinity_DeviceParam*	Default height uncertainty to be used as per schedule 4	0
OFCOMChannelCalculations	*DefaultMaxMasterEIRP*	Default Max Master EIRP value	-19
OFCOMChannelCalculations	*DefaultPREFSENS_DeviceParam*	Default PREFSENS value	-114

OFCOMChannelCalculations	*DefaultTechIdentifier_DeviceParam*	Default technology identifier	Generic
OFCOMChannelCalculations	*DefaultTypeAHeight_DeviceParam*	Default height for Type-A WSDB device as per schedule 4	30
OFCOMChannelCalculations	*DefaultTypeBHeight_DeviceParam*	Default height for Type-B WSDB device as per schedule 4	1.5
PAWSApi	*DateFormat*	Date format used in PAWS requests and responses	MM'/'dd'/'yyyy
PAWSApi	*PawsApiVersion*	Current version of PAWS api supported by the application	1
PMSESync	*ExchangeURI*	Exchange URI for receiving PMSE emails from Ofcom	
PawsRegistration	*LastRegDate*	Last registration date of incumbent	7/18/2014
PawsRegistration	*LastRegSeq*	Last generated sequence number for PAWS registration	6
PmseSync	*CheckSignedEmail*	Indicates whether PMSE assignment emails are to be checked to verify that they are signed. Used for testing only, should always be set to True in production	False
PmseSync	*AcceptableDelayInUnscheduledAdjustmentMin*	Additional acceptable delay time (in mins) between two Unscheduled adjustments emails from Ofcom	30
PmseSync	*Email*	Email address used for retrieval of PMSE assignment data	
PmseSync	*PassCode*	Password used for PMSE assignment email account	
PmseSync	*PmseAssignment_CacheIntervalMin*		60
PmseSync	*PmseSyncSynchronizationTimer_Min*	Interval to be used for checking for PMSE assignment data	10
PmseSync	*UnscheduledAdjustmentTimeMin*	Interval to be used for checking for unscheduled adjustment data	120
PmseSync	*mailBodyAttachmentNotFound*	Body text for email message indicating that no attachment was found for the PMSE data	Attachment Not Found.
PmseSync	*mailBodySubjectNotValid*	Body text for email message indicating that subject is not valid	Subject is not valid
PmseSync	*mailBodyServerError*	Body text for email message indicating that PMSE Sync encountered some server error	Some Error Occured while saving data

PmseSync	*mailBodyFileNotValid*	Body text for email message indicating that received attachment file is not valid	File is not valid
PmseSync	*mailBodyMoreThanOneAttachments*	Body text for email message indicating that multiple attachments are received in the email.	More than one attachments found
PmseSync	*mailBodyUnsignedMail*	Body text for email message indicating that an unsigned email has been received.	Email is not signed
PmseSync	*mailBodyFailToProcessFile*	Body text for email message indicating that PMSE sync data was not processed	Not Able to process file
PmseSync	*mailBodySyncSuccessfully*	Body text for email message indicating that PMSE sync process was completed	Data Sync Successfully.
PmseSync	*mailBodyUnscheduledMailNotFound*	Body text for email indicating that unscheduled adjustment data was not found	We did not receive the unscheduled adjustment file.
PmseSync	*mailSubjectUnscheduledMailNotFound*	Subject text for email indicating that unscheduled adjustment data was not found	Unscheduled Adjustment
PmseSync	*ofcomEmailId*	Email address at Ofcom for receipt of emails	
RegionManagementApi	*RegionManagementApiVersion*	Api version for RegionManagment API	1
Terrain	*ClutterDataFileName*	Blob file name for clutter data	
Terrain	*DBConnectionStringTerrain*	Connection string used for retrieval of terrain data	DefaultEndpointsProtocol=https;AccountName=[MyAccountName];AccountKey=[MyAccountKey]
Terrain	*dataDirectory*	The blob container name to use for lookup of terrain files.	os-terrain-50-ascii-grid-gml-grid
Terrain	*landCoverDataDirectory*	The blob container name used for loading landcover terrain file.	lcm2007-25m-raster-doc
WSDValidatation	*ValidateWSDLocation*	Boolean flag indicating whether requested WSD location should be validated using Region Polygons cache for OFCOM.	FALSE

5.2Microsoft.Whitespace.PAWS

The PAWS Manager will contain the exact same messages as the PAWS web service. The PAWS Manager will be responsible for:

- Validating and Parsing JSON parameters

- Validating device Ids by querying the whitespace azure tables
- Calling into White Space Driver to get available spectrum
- Converting spectrum requests and responses to JSON

Assembly Name	**Microsoft.Whitespace.PAWS.dll**
Source Directory	src\Whitespace\Paws
Root Namespace	Microsoft.Whitespace.Paws
Dependencies	Microsoft.Whitespace.Common.dll, Microsoft.Whitespace.Driver.dll, Microsoft.Whitespace.Entities.dll

5.3Microsoft.Whitespace.Sync

5.3.1 Database

5.3.1.1 DBSync File worker

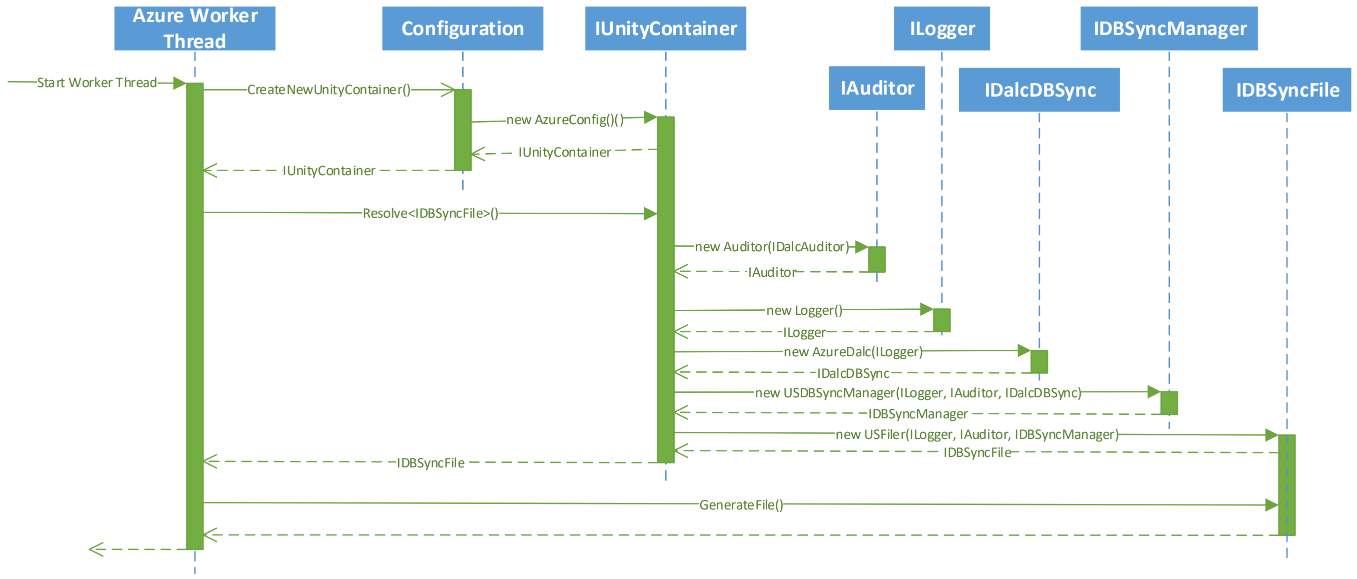
Assembly Name	**Microsoft.Whitespace.Sync.Database.File.dll**
Source Directory	src\Whitespace\Sync\Sync\Database\File
Root Namespace	Microsoft.Whitespace.Sync.Database.File
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common

The DB Sync File Worker is an azure worker role that will wake up periodically to generate two kinds of XML files:

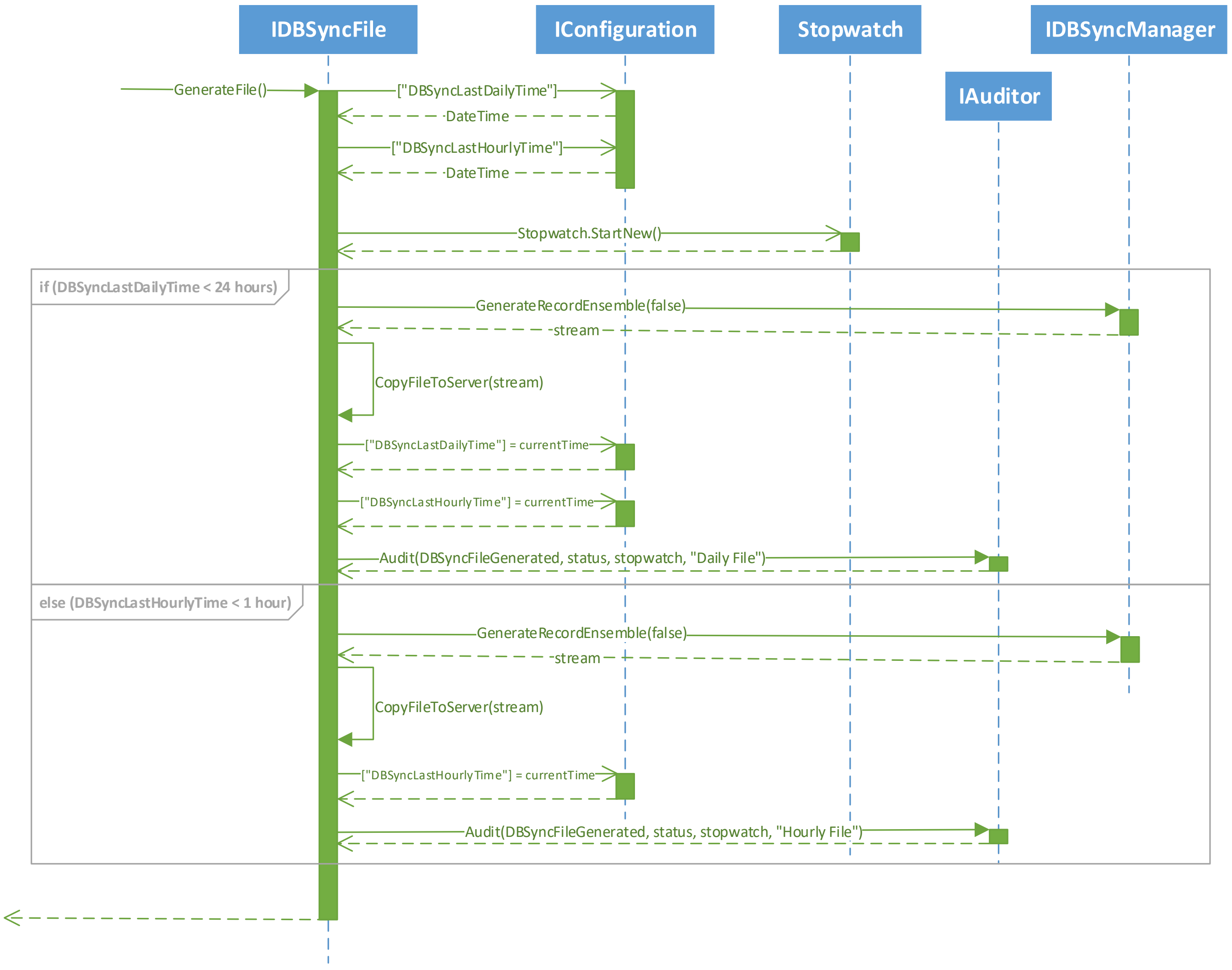
- The first file will contain all past registration (generated daily).
- The 2nd file will be generated hourly and will only contain incremental changes from the previous day.

Note that the actual file generation will be handled by the Registration Sync Manager. Once the files have been generated, the Sync Manager will then copy the generated files to a web server destination as defined in the web.config file.

The following sequence diagram shows the high level view of the DB Sync File Worker:


Azure Worker Thread
Configuration
IUnityContainer
ILogger
IDBSyncManager
IAuditor
IDalcDBSync
IDBSyncFile
Start Worker Thread
CreateNewUnityContainer()
new AzureConfig()()
IUnityContainer
IUnityContainer
Resolve<IDBSyncFile>()
new Auditor(IDalcAuditor)
IAuditor
new Logger()
ILogger
new AzureDalc(ILogger)
IDalcDBSync
new USDBSyncManager(ILogger, IAuditor, IDalcDBSync)
IDBSyncManager
new USFiler(ILogger, IAuditor, IDBSyncManager)
IDBSyncFile
IDBSyncFile
GenerateFile()

The following sequence shows the IDBSyncFile.GenerateFile sequence for the US FCC database:


IDBSyncFile
IConfiguration
Stopwatch
IDBSyncManager
IAuditor
GenerateFile()
["DBSyncLastDailyTime"]
DateTime
["DBSyncLastHourlyTime"]
DateTime
Stopwatch.StartNew()
if (DBSyncLastDailyTime < 24 hours)
GenerateRecordEnsemble(false)
stream
CopyFileToServer(stream)
["DBSyncLastDailyTime"] = currentTime
["DBSyncLastHourlyTime"] = currentTime
Audit(DBSyncFileGenerated, status, stopwatch, "Daily File")
else (DBSyncLastHourlyTime < 1 hour)
GenerateRecordEnsemble(false)
stream
CopyFileToServer(stream)
["DBSyncLastHourlyTime"] = currentTime
Audit(DBSyncFileGenerated, status, stopwatch, "Hourly File")

Use of NextTransactionID in DB Sync File Worker

As indicated in the Whitespace DB Sync specification, the application will use the NextTransactionID value to control the exchange of registration data with the other WSDBAs in the outbound DB Sync process.

The application will use the following approach to manage the NextTransactionID value for synchronization with external WSDBAs:

1. The NextTransactionID will be generated as a guid.
2. Azure table TransactionIDIssued will be used to keep a record of all NextTransactionID values that are used within the system. Each row in the table will contain the generated NextTransactionID and the associated timestamp. We will also include a column that identifies how the NextTransactionID was generated (i.e. by one of the File Workers as the result of a RealTimePollRequest).

3. A new NextTransactionID will be generated and saved in the TransactionIDIssued table when any of the following events occurs:
    a. A RealTimePollRequest is received from another WSDBA. As part of the response, the system generates a new NextTransactionID value and returns it to the requesting WSDBA. The next time the WSDBA sends a request, they must include the generated NextTransactionID value withn the request.
    b. A new AllRegistrations file is created by the DB Sync File Worker. When this file is created, a new NextTransactionID is generated and it is stored within the file.
    c. A new incremental registrations file is created by the DB Sync File Worker. When this file is created, a new NextTransactionID is generated and it is stored within the file.

4. Every time a NextTransactionID value is generated, it will be inserted to the above table.

6. When a RealTimePollRequest is received, the NextTransactionID will be validated as follows:

a. Read the NextTransactionID table to see if the value exists at all. If it does not, return the TransactionIDStale response code.

b. If the NextTransactionID exists, check to see if the associated timestamp is within the last 72 hours. If the timestamp is more than 72 hours old, return the TransactionIDStale response code.

c. If the NextTransactionID is within the last 72 hours, retrieve all registration data on which the create date or the update date is greater than or equal to the associated timestamp for the NextTransactionID.

7. The NextTransactionID table will also be used by the incremental DBSyncFileWorker process. In this case, it will be used as follows:

a. Retrieve all rows from the NextTransactionID table where the generating process is either the daily sync file worker or the incremental sync file worker, ordered by timestamp descending.
b. Use the first row retrieved to control the retrieval of registration records – get all rows where create date or update date is greater than or equal to the retrieved timestamp.

5.3.1.2 DBSync Manager

Assembly Name	**Microsoft.Whitespace.Sync.Database.Manager.dll**
Source Directory	src\Whitespace\Sync\Sync\Database\Manager
Root Namespace	Microsoft.Whitespace.Sync.Database.Manager
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common, Rensi.SSHNet, Security.cryptography

Module used to handle the region-specific database-to-database synchronization management. The main responsibilities include:

- Generating and singing XML files containing past registrations.
- Validating XML signatures from external whitespace administers (spectrum bridge, iconnectiv, …).
- Parses external XML files and syncing with internal MS Whitespace DB.

Each supported region will implement the BaseRegistrationSyncManager class which will have a set of pure virtual methods defined as:

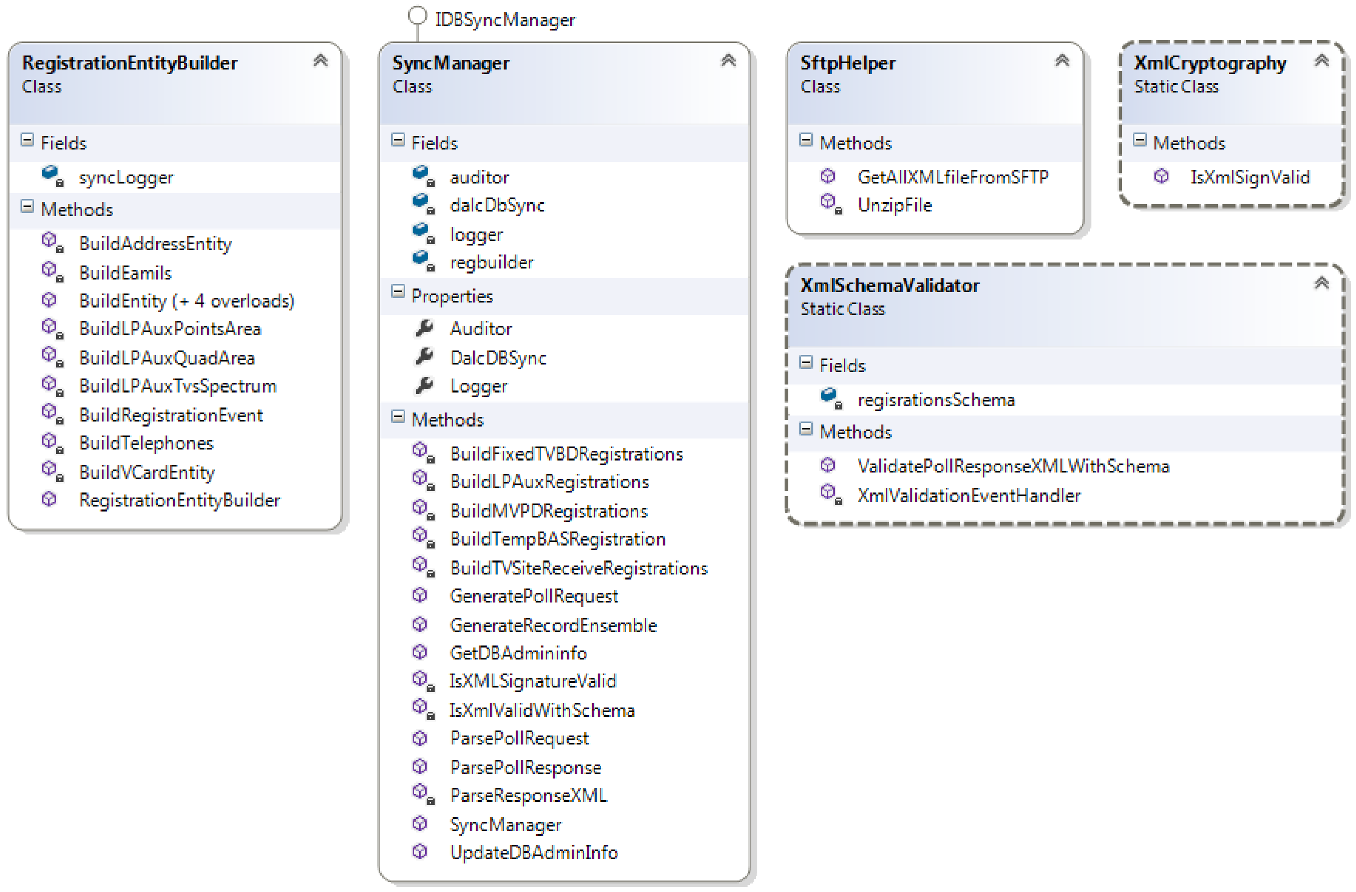
- GeneratePollRequest: This method is called by the DB Sync Poller Worker in order to make a request to other 3rd party white space database administrator.
- ParsePollResponse:This method is called by the DB Sync Poller Worker in order to:
  - validate the real-time poll request result
  - parse the poll request and update the MS white space DB with any changes.
- ParsePollRequest: This method is called by the DB Admin Web Service in response to a web service request from external white space DB Administrators.
- GenerateRecordEnsemble: This method is called by the DB Sync Filer Worker and will either return all of the past MS specific device registrations (for daily posting) or just return the incremental changes from the previous day (for the hourly sync file).

5.3.2 RGN1DBAdminInfo Azure Table

Information regarding other Whitespace DBAs with which the application communicates for the FCC implementation is contained in the RGN1DBAdminInfo table. This table is used to control various aspects of the DBSync process. As new white space database administrators are added, information must be added to this table to provide the necessary values to support the DB Sync process for each DBA. The table below lists the information that is contained within the table.

Column Name	Description
PartitionKey	Contains the name that is used to identify the DBA in the synch process (e.g. KEYB, SPBR, etc)
RowKey	Guid value
Timestamp	Date/time of last update
FTPServerPwd	Password for connecting to the DBA's ftp server when picking up full file
FTPServerUserID	User id for connection to DBAs ftp server
FTPServerURL	URL of the DBA's ftp server
WebServiceURL	End point for the DBA's web service for incremental polling
NextTransactionId	Next transaction id value to be used in next real time poll request for the DBA
PublicKey	Used to validate xml signature
SFTPPath	Path in DBAs sftp server

The following is the class diagram for DBSyncManager:



RegistrationEntityBuilder is used to build the registration entities from the xml which needs to be stored in the Azure able

sftpHelper is used to connect o SFTP and down load the All Registrations File

XMLSchemaValidator is used validate the given xml schema with the given xml.

XMLCryptography is used to verify the xml file received with the signature value.

5.3.2.1 DBSync Poller

The DB Sync Poller Worker is an azure worker role that is responsible for polling external WSDBA for any registration changes. The following figure shows two supported regions (United States and Singapore) that inherit from the BasePoller class:

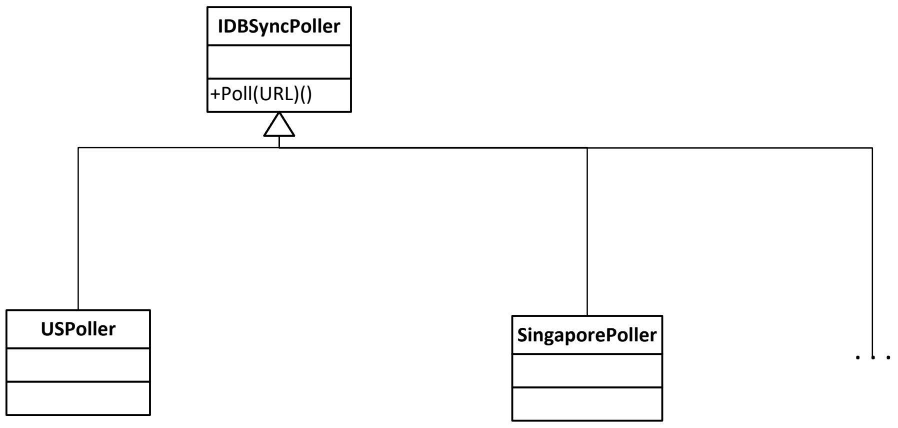


Figure 3 - BasePoller Class Hierarchy

Assembly Name	Microsoft.Whitespace.Sync.Database.Poller.dll
Source Directory	src\Whitespace\Sync\Sync\Database\Poller
Root Namespace	Microsoft.Whitespace.Sync.Database.Poller
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common

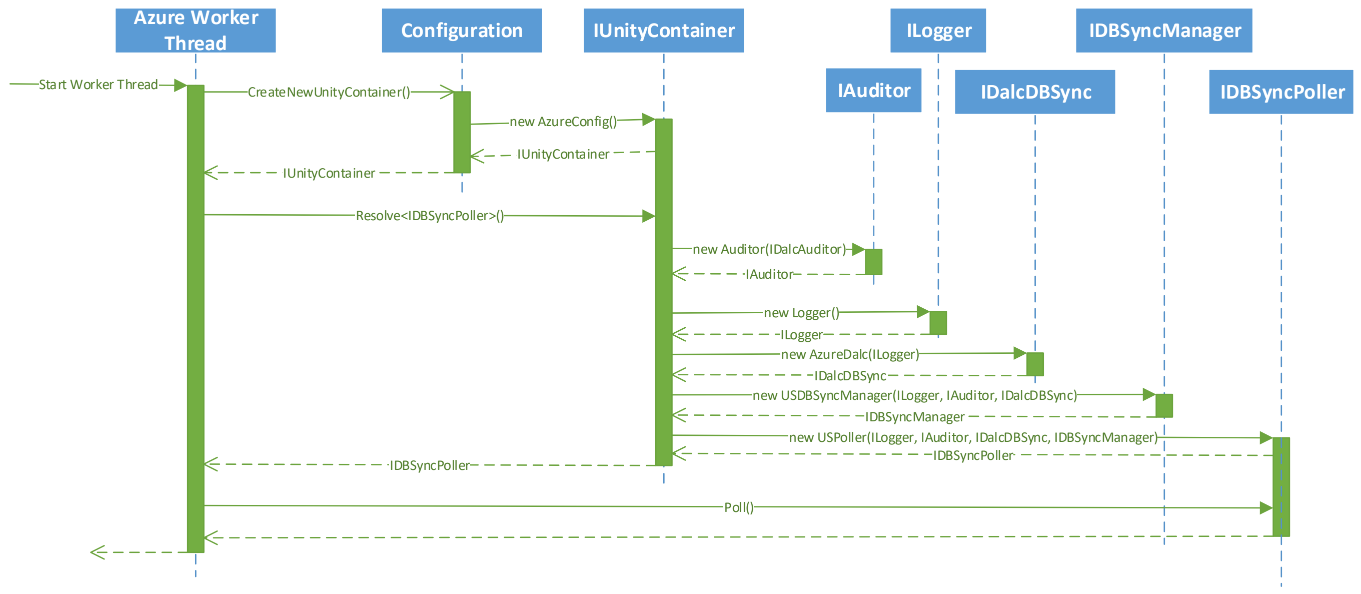
5.3.3 FCCDBSyncPoller

```
public class FCCDBSyncPoller : IDBSyncPoller
{
  public void Poll() { }
}
```

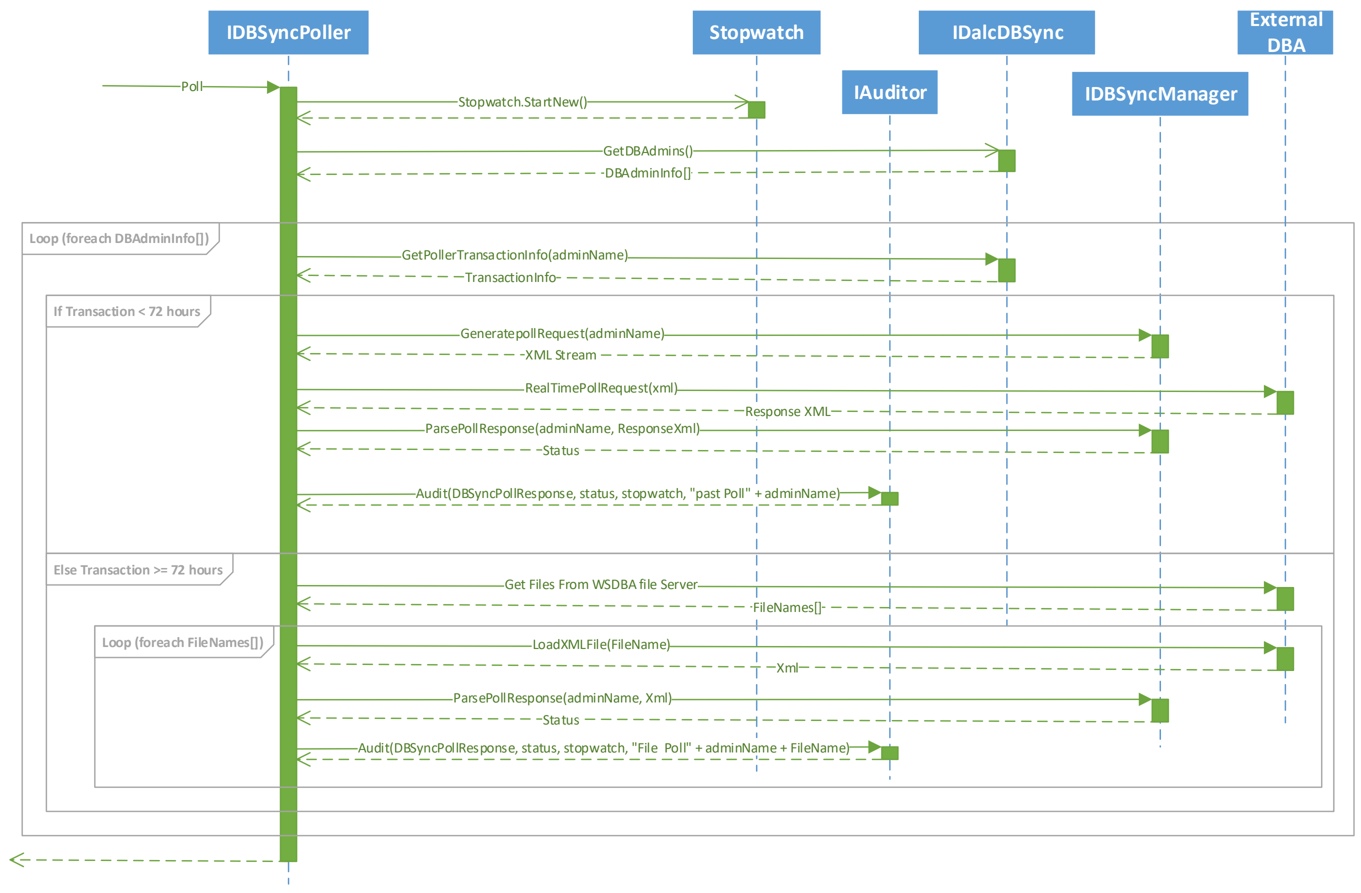
When the DB Sync Poller Worker is initiated, it will perform the following task:

1. Instantiate the appropriate region specific poller class based on the region setting of web.config.
2. The worker role will then loop over all registered WSDBA and then call the “Poll” method on the region specific poller passing in the URL of the WSDBA “fast poll” service.
3. The “Poll” method will then:
    a. Instantiate the region specific “Registration Sync Manager”
    b. Call the GenerateRealTimePollRequest method on the Registration Sync Manager to generate the required parameters for the WSDBA.
    c. Call the WSDBA web service
    d. After the WSDBA method has returned, pass the response string back to the “Registration Sync Manager” in order to update the MS White Space DB with any changes.

The following sequence diagram shows the high-level view of the DB Sync Poll Worker:



The following sequence shows the IDBSyncPoller.Poll sequence for the US FCC database:


IDBSyncPoller
Stopwatch
IDalcDBSync
External DBA
IAuditor
IDBSyncManager
Poll
Stopwatch.StartNew()
GetDBAdmins()
DBAdminInfo[]
Loop (foreach DBAdminInfo[])
GetPollerTransactionInfo(adminName)
TransactionInfo
If Transaction < 72 hours
GeneratepollRequest(adminName)
XML Stream
RealTimePollRequest(xml)
Response XML
ParsePollResponse(adminName, ResponseXml)
Status
Audit(DBSyncPollResponse, status, stopwatch, "past Poll" + adminName)
Else Transaction >= 72 hours
Get Files From WSDBA file Server
FileNames[]
Loop (foreach FileNames[])
LoadXMLFile(FileName)
Xml
ParsePollResponse(adminName, Xml)
Status
Audit(DBSyncPollResponse, status, stopwatch, "File Poll" + adminName + FileName)

5.3.4 Region

The Protected Region Worker is responsible for:

- downloading all the region-specific modules used for calculating protected entity contours
- downloading all the region-specific files for validating device authenticity (i.e. such as validating devices based on serial numbers or device Ids).
- performing and caching region contour calculations

Each supported region will implement the IRegionSyncinterfacethat will have one public method called SyncDB(). The following figure shows two supported regions (United States and Singapore) that inherit from the BaseDataSync class:

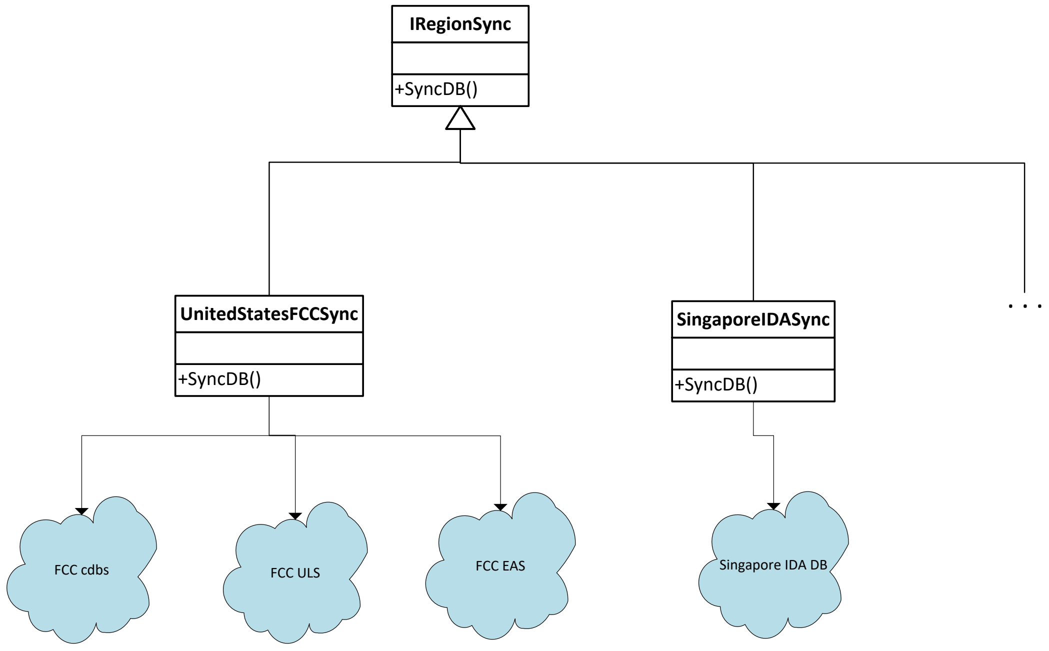


Figure 4 - RegionDataSync Class Hierarchy

UnitedStatesFCCSync

Assembly Name	**Microsoft.Whitespace.Sync.Region.dll**
Source Directory	src\Whitespace\Sync\Region
Root Namespace	Microsoft.Whitespace.Sync.Region
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common

The FCCRegionSync will synchronize the FCC's CDBS, and ULS files daily as documented in the Channel Calculations for White Spaces Guidelines.

Note that the synchronization requirements for Ofcom data are significantly different than for the FCC, so Ofcom does not follow the above class hierarchy.

5.3.4.1 *CDBSFileType*

Enumeration of file types that are retrieved from the FCC Consolidated Database System for the region sync process.

5.3.4.2 *FccRegionSync*

The FCC implementation of the IRegionSync interface. Types of data processed by this class are as follows: methods within this class include the following:

Data Type	Method Name
US and Mexico TV Stations	ProcessCDBSUSMexicoTvEngData
Canadian TV Stations	ProcessCDBSCanadianData
TV Translators	ProcessCDBSTranslatorData
Authorized FCC Devices	ProcessAuthorizedFCCDeviceData
Broadcast Auxiliary Stations	ProcessBroadcastAuxiliaryStations
CMRS/PLCMRS TBand Stations	ProcessPlCmrsStations
Licensed LP Aux Devices	ProcessLicensedLPAuxData
Unlicensed LP Aux Devices	ProcessUnLicensedLPAuxData

5.3.4.3 *RegionSyncConstants*

Contains constant values used in the RegionSync process.

5.3.4.4 *ULSFileType*

Enumeration of file types for the FCC Universal License System synchronization process.

5.3.4.5 *ULSMapping*

Used for mapping field names in data retrieved from ULS.

5.3.4.6 *ULSRecordType*

Enumeration of record types in the FCC Universal License System.

5.4 Microsoft.Whitespace.Common

Contains all of the whitespace's shared components such as:

- Default Auditor Implementation
- Default Logger Implementation
- Default Configuration Implementation
- Default Paws Validator

- XmlUnityConfigurationSection
- Static Utils class implementation (provides easy access to default IConfiguration implementation).
- Common Interface Definitions

Assembly Name	**Microsoft.Whitespace.Common.dll**
Source Directory	src\Whitespace\Common
Root Namespace	Microsoft.Whitespace.Common
Dependencies	Microsoft.Whitespace.Entities.dll

5.4.1 CacheHelper

5.4.1.1 FCCServiceCacheHelper

Helper class to manage cached objects for FCC implementation.

5.4.1.2 OFCOMServiceCacheHelper

Helper class to manage cached objects for Ofcom implementation.

5.4.2 Utilities

5.4.2.1 Conversion

Provides various utility methods for conversion of values used in the application.

5.4.2.2 GeoCalculations

Provides methods to perform various geographical calculations such as distance between two points, bearing of a given azimuth, etc.

5.4.2.3 MiscExtensions

Additional utility methods for conversions and geographic calculations.

5.4.2.4 Transform Coordinate

Provides methods for transformation of coordinates between various coordinate systems.

5.4.2.5 UnityExtensions

Provides utility methods for the type IUnityContainer.

5.4.3 Validators

5.4.3.1 BasePawsValidator

The base class for validation of PAWS requests

5.4.3.2 BaseRegionManagementValidator

The base class for validation of Region Management requests. Validates the specified object and returns the result.

5.4.3.3 FccPawsValidator

The FCC implementation of the PAWS validator class

5.4.3.4 FCCRegionManagementValidator

FCC implementation of the BaseRegionManagement validator class

5.4.3.5 OfcomPawsValidator

Ofcom implementation of the PAWS validator class.

5.4.3.6 OFCOMRegionManagementValidator

Ofcom implementation of the Base Region Management Validator.

5.4.4 Value Provider

5.4.4.1 FccPawsValueProvider

FCC implementation of PawsRegionalValueProvider.

5.4.4.2 OfcomPawsValueProvider

Ofcom implementation of PawsRegionalValueProvider

5.4.4.3 PawsRegionalValueProvider

Abstract class to revolve json properties in PAWS requests and responses. FCC and Ofcom requests for PAWS use the same property names but the content may differ. The value provider classes set the values for each property based on the specific implementation of PAWS.

5.4.5 Web API Helper

5.4.5.1 ApiDependencyResolver

ApiDependency Resolver is used to resolve Unity dependencies in the controller classes for the PAWS and RegionManagement services.

5.4.5.2 UnityDependencyScope

UnityDependencyScope is used for resolving Unity dependencies in the controller classes for the PAWS and RegionManagement services.

5.4.6 Auditor

The Auditor is used to track all major events within the whitespace DB System which include:

- All processed web service calls into the (PAWS, PMSE, Management, DB Admin).
- All region sync with government databases (CDBS, ULS, EAS).
- All DB Sync with external whitespace DB Admins (via the DB Sync Poller Worker).
- All DB Sync File generations.

5.4.6.1 AzureAuditor

Implementation of the Auditor interface that writes audit messages to an Azure table.

The Auditor class construction will perform the following operations:

- Validate that the dalc parameters are not null.
- Set the RegionCode property to the value from the static Utils.Configuration.CurrentRegionId
- Save the passed in dalc parameter to the private member variable.

- Initialize the TransactionId property with a newly generated GUID.

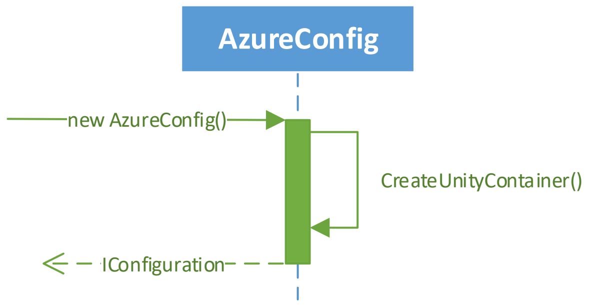
The Audit method will simply forward all of the parameters and properties onto the IDalcAuditor interface in order for the audit message to be written to the DB.

The Azure Audit Table will be named "Audit" and will have the following column definitions:

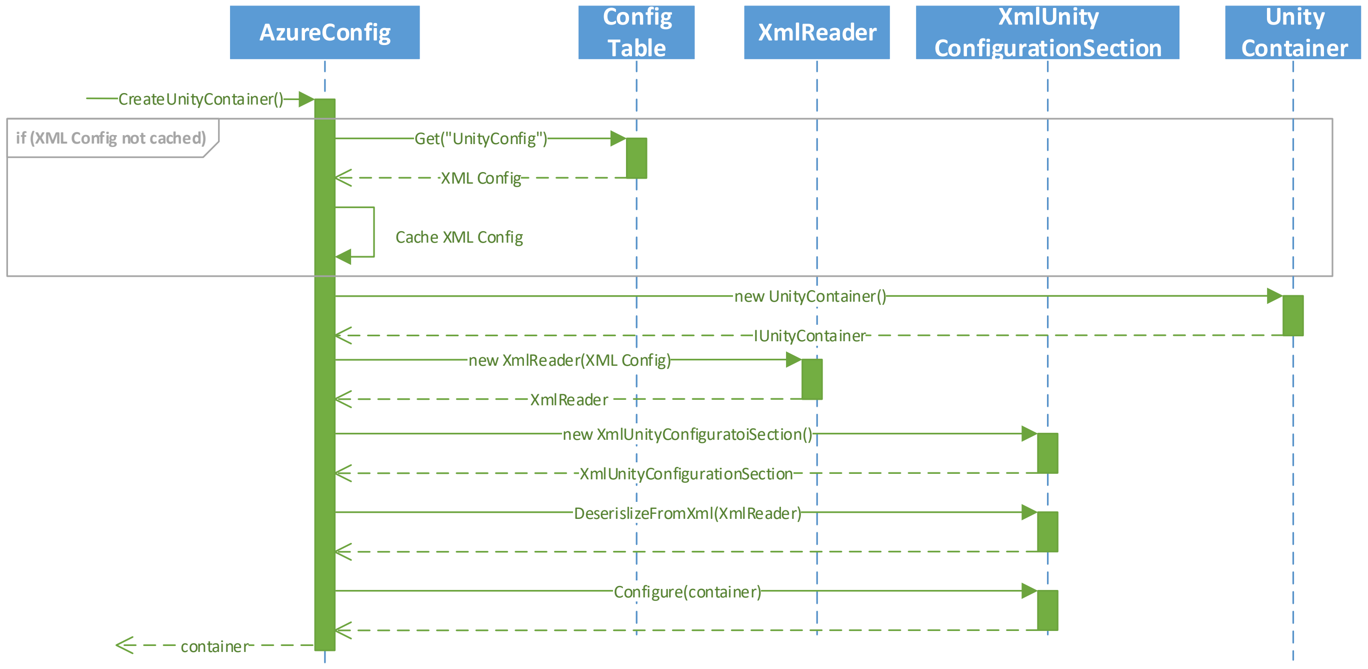
Column Name	Type	Description
PartitionKey	string	RegionId + "-" + YYYYMMDD For example, if the Region Id is "1" (United States) and the current date is September 1, 2013 then the Primary key would be: "1-20130901"
RowKey	string	A unique Guid (i.e. System.NewGuid()).
Timestamp	DateTime	Time the audit operation completed.
AuditId	integer	The type of audit operation that was just completed.
Status	integer	The status of the audit operation (success or failure).
Message	string	A descriptive message that is specific to the type of audit being generated.
ElapsedTime	integer	Time in milliseconds it took for the audit operation to complete.
TransactionId	guid	A unique Id for the transaction.

5.4.6.2 AzureConfig

A default Configuration class will be implemented by the Microsoft.Whitespace.Common assembly and will have the following sequence diagram for the construction:



And the following diagram shows the CreateUnityContainer sequence:



5.4.6.3 AzureLogger

Writes the specified message to the logfile (usesLoggingApplicationBlock).

5.4.6.4 DatabaseCache

Represents class for Database Cache. Service cache helper.

5.4.6.5 ErrorHelper

Helper class to build error messages that are returned by the application.

5.4.6.6 IAuditor

Interface used for recording audit messages used in the application.

5.4.6.7 ICache

The ICache interface is mainly used by TerrainCache, but can also be used by other components to generically cache values.

5.4.6.8 IConfiguration

Defines all of the Whitespace configuration methods.

5.4.7 Dalc Interfaces

The following are all of the interfaces implemented by the data access layer component.

5.4.7.1 IDalcAuditor

Inteface defining all of the audit methods used by the data access layer.

5.4.7.2 IDalcCommon

Interface defining the common DALC methods used throughout the application.

5.4.7.3 IDalcConfig
The IDalcConfig Interface is used by configuration components to get and set config values from the config table.

5.4.7.4 IDalcDBSync
The IDalc DBSync interface is used by the DB Sync Manager component for handling database to database synchronization.

5.4.7.5 IDalcDTTAvailabilitySync
Interface defining DALC methods for management of DTT availability data for Ofcom.

5.4.7.6 IDalcIncumbent
The IDalcIncumbent interface is used by the Whitespace driver component for retrieving, adding, updating and deleting incumbents data.

5.4.7.7 IDalcLogger
Defines all of the required audit methods used by the data access layer component.

5.4.7.8 IDalcPaws
The IDalcPaws interface is used by the Whitespace driver for retrieving device information and reserving spectrum usage.

5.4.7.9 IDalcPMSE
Represents the PMSE interface into the data access layer component.

5.4.7.10 IDalcPmseSync
Interface that defines the methods for the PMSE Sync process for Ofcom.

5.4.7.11 IDalcRegionSync
The IDalcRegionSync interface is used by the “protected region worker” component in order to synchronize database files from potential government agencies. The “protected region worker” will convert a table to a generic SyncTable and SyncRow component which it will then pass into the IDalcRegionSync.UpdatTable method in order to be saved in the underlying database.

5.4.7.12 IDalcTerrain
The IDalc Terrain interface is used for loading terrain files from storage.

5.4.7.13 IDalcUserManagement
The IDalcUserManagement interface is used for managing user access to the system. Note that this component is not currently used in the application.

5.4.8 DB Sync Interfaces

5.4.8.1 IDBSyncFile
The IDBSyncFile interface is used by the DB Sync File worker in order to generate XML files used for DB-to-DB synchronization.

5.4.8.2 *IDBSyncFileIncr*

Represents the DBSync interface (used by azureworkerthread) and generates the DB-to-DB sync file and places it up on the FTP server.

5.4.8.3 *IDBSyncManager*

Represents the DBSync manager interface used by DB Sync Poller worker, DB Sync Filer Worker, and DBAdmin webservice.

5.4.8.4 *IDBSyncPoller*

Represents the IDBSync Poller interface that is used by the DB Sync Poller worker for polling WSDBA for any registration changes.

5.4.9 Whitespace Driver Interfaces

The White Space Driver is the primary access point for:

- Accessing/Updating TVBD
- Instantiating the correct region specific Channel Calculator.
- Processing and culling incumbents.
- Updating Exclusion List or Exclusion Regions

Assembly Name	**Microsoft.Whitespace.Driver.dll**
Source Directory	src\Whitespace\Driver
Root Namespace	Microsoft.Whitespace.Driver
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common, Microsoft.Whitespace.RegionCalculations

The White Space Driver will implement the following interfaces:

- IDriverPaws
- IDriverPMSE
- IDriverRegionManagement

The following interfaces are all implemented by the Whitespace Driver Component.

5.4.9.1 *IDriverPaws*

The IDriver Paws interface is used by the PAWS Manager for handling all PAWS request.

5.4.9.2 *IDriverPMSE*

The IDriver PMSE interface is used by the PMSE web service for handling event registration for LPAux devices.

5.4.9.3 *IDriverRegionManagement*

The IDriver RegionManagement interface is used by the Region Management web service for managing FCC request, incumbents and users.

5.4.9.4 IDTTAvailabilitySync

The IDTTAvailabilitySync interface is used for synchronization of DTT availability data for Ofcom.

5.4.10 Logger

The Logger is a helper classed used to help diagnose and troubleshoot issues with the whitespace system.

5.4.10.1 ILogger

The ILogger interface has the following definition:

The Logger constructor will perform the following operations:

- Validate that the IDalcLogger and IConfiguration interfaces are not null.
- Save the IDalcLogger interface to a private member variable.
- Obtain the RegionCode from the static Utils.Configuration property and save them in the appropriate local attribute.
- Initializes the TransactionId property to a newly generated Guid.

The Log method will perform the following operations:

- Call into the Logging Application Block.

The Azure Log Table will be "WADLogsTable" and will have the following column definitions:

Column Name	Type	Description
PrimaryKey	string	
RowKey	string	
Timestamp	DateTime	Time the log operation completed.
EventTickCount	integer	
DeploymentId	string	
Role	string	
RoleInstance	string	
Level	Integer	
EventId	Integer	
Pid	Integer	
Tid	Integer	
Message	String	

5.4.10.2 INativeMethodService

Interface for calling 32-bit dll for FMTV curves calucation.

5.4.10.3 INetFileStream

Interface used for file downloads in FCC Region Sync process

5.4.10.4 IPawsBL
The IPawsBL contains all of the methods used in the Business layer for processing of PAWS requests.

5.4.10.5 IPawsValidator
Interface that defines the methods for validation of Paws requests.

5.4.10.6 IPmseAssignment
Interface that defines the methods for the synchronization of PMSE assignment data for Ofcom.

5.4.10.7 IPMSEValidator
Interface for validation or LPAux device registration.

5.4.10.8 IRegionCalculation
The IRegion Calculation interface is used by the RegionCalculation component for determining channel availability (contour calculation, RSSI values, …).

5.4.10.9 IRegionManagementValidator
Interface that defines the methods for validation of all RegionManagement requests.

5.4.10.10 IRegionSync
The IRegionSync interface is used by the Protected Region Worker for synchronizing with government mandated database files.

5.4.10.11 IServiceCacheHelper
Interface for management of cached objects within the application.

5.4.10.12 ITerrainElevation
The ITerrain Elevation interface is used by the Region Calculation component for calculating the elevation data for a given position.

5.4.10.13 IUserManager
The IUserManager component is used by the Region Management and PMSE web service for performing user management operations. Note that this interface is not used within the current application.

5.4.10.14 Logger
Writes the specified message to the log file(usesLoggingApplicationBlock).

5.4.10.15 NativeMethods
Represents class NativeMethods.

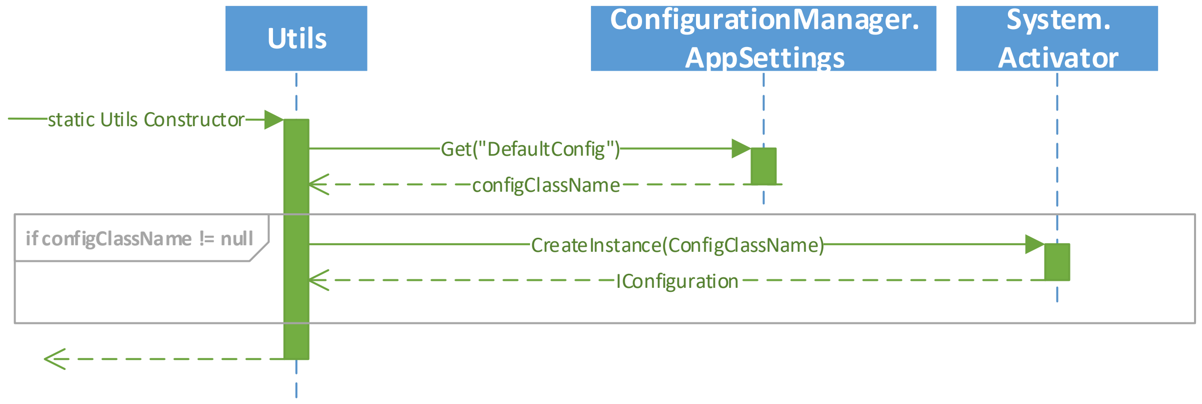
5.4.10.16 NativeMethodServiceClient
Client class for calling interpolation dll to for contour calculations.

5.4.10.17 NetFileStream
Implements the INetFileStream interface for file downloads in the FCC RegionSync process.

5.4.11 Static Utils Class

The static Utils class is used to instantiate the default IConfiguration interface. The IConfiguration interface is the only component that is not instantiated via the UnityContainer (the UnityContainer initialization has a dependency on the IConfiguration interface in order to set all of the interface to class instance mappings). The default IConfiguration is therefore instantiated based on the fully qualified name specified in the application config file under the name "DefaultConfig".

Below is a sequence diagram for the Utils constructor:



Below is an outline of the static Utils class implementation:

5.4.12 XmlUnityConfigurationSection

The Whitespace DB utilizes the Microsoft.Practices.Unity component in order to perform all dependency injection. The default implementation of the UnityContainer expects its XML configuration to be physically present in the app.config or web.config file which limits the dependency injection map from being dynamically modified after installation. To work around this limitation, the Microsoft.Whitespaces.Common assembly will contain an XmlUnityConfigurationSection class that inherits from UnityConfigurationSection and will only implement the DeserializeFromXml (this will then allow for aUnityContainer instance to be initialized from any XML stream instead of the application's config file). The following section shows the implementation of the XmlUnityConfigurationSection class (see the CreateUnityContainer code snippet in the AzureConfig section)

5.5Microsoft.Whitespace.Dalc

5.5.1 Microsoft WhiteSpce Dalc

The White Space DALC is the Data Access Layer for accessing/updating all of the white space tables and blob storage. An Entity Framework Data Model class will exist for each of the Azure tables.

Assembly Name	**Microsoft.Whitespace.Dalc.dll**
Source Directory	src\Whitespace\Dalc
Root Namespace	Microsoft.Whitespace.Dalc
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Common

5.5.1.1 AuditTableEntity

The **Audit** table is an internal MS table that is used to record the success and failure of all web API calls and azure worker operations (DB Sync File Worker, DB Sync Poller Worker, and Protected Region Worker).

5.5.1.2 AzureDalc

The primary Data Access Layer Component for the application. Implements all of the Dalc interfaces and manages storage and retrieval of data from Azure storage.

5.5.1.3 AzureStorageHelper

A helper class that performs the actual storage and retrieval of data in Azure.

5.5.1.4 LogTableEntity

Entity class that defines the contents of the log table. .

5.6Microsoft.Whitespace.Driver

5.6.1 Microsoft WhiteSpce Driver Interface

The following interfaces are all implemented by the Whitespace Driver Component.

5.6.1.1 IDriverPaws

The IDriver Paws interface is used by the PAWS Manager for handling all PAWS request.

5.6.1.2 IDriverPMSE

The IDriver PMSE interface is used by the PMSE web service for handling event registration.

5.6.1.3 IDriverRegionManagement

The IDriver Region Management interface is used by the Region Management web service.

5.6.2 Microsoft.Whitespace.Driver

5.6.2.1 PawsDriver

Implements the IDriverPaws interface to manage the processing of all Paws requests within the application.

5.6.2.2 RegionManagementDriver

Driver class that controls the processing of all of the methods of the RegionManagement service.

5.7Microsoft.Whitespace.DTTSync

5.7.1 DTTSync

This assembly contains classes related to synchronization of DTT availability data for Ofcom.

5.7.1.1 DttDataAvailabilitySync

This class performs the synchronization process for DTT availability data for Ofcom.

5.7.1.2 WorkerRole

A worker role that executes the DTTAvailabilitySync process at specified intervals.

5.8 Microsoft.Whitespace.Entities

The entities assembly contains the definition of all the common schemas.

Assembly Name	**Microsoft.Whitespace.Entities.dll**
Source Directory	src\Whitespace\Entities
Root Namespace	Microsoft.Whitespace.Entities
Dependencies	None

5.8.1 Microsoft.Whitespaces.Entities.Versitcard

All of the following components will reside in the Versitcard namespace and represent personal contact information as documented at: http://tools.ietf.org/html/draft-ietf-vcarddav-vcardxml-10.

5.8.1.1 Address

Represents the address of a registration owner.

5.8.1.2 EMail

Represents an email address of a registration owner.

5.8.1.3 FormattedName

Represents the formatted name of a registration owner.

5.8.1.4 Name

Represents a name of a registration owner.

5.8.1.5 Organization

Represents a VCard organization.

5.8.1.6 Telephone

Represents the VCard owner's telephone information.

5.8.1.7 Title

Represents a owner's VCard Title.

5.8.1.8 VCard

The VCard represents personal contact information.

5.8.2 .Entities

5.8.2.1 Address

Represents a PAWS address.

5.8.2.2 AntennaCharacteristics

Defines the attributes of an antenna for use in channel calculations in FCC implementation.

5.8.2.3 AntennaPatternItem

Entity class that maps to the FCC Antenna Pattern file in the FCC implementation.

5.8.2.4 Area
Defines am area used in geographical calculations.

5.8.2.5 AuthorizedDeviceRecord
Maps to FCC record for authorized wireless devices.

5.8.2.6 AvailableSpectrumConverter
JSON converter for AvailableSpectrum data

5.8.2.7 AvailableSpectrumRequest
Represents a paws available spectrum request:

5.8.2.8 AvailableSpectrumRequestBase
Base class that represents a paws available spectrum request.

5.8.2.9 BaseJsonConverter
Base class for custom serialization and de-serialization of json data

5.8.2.10 BasePawsClass
Base class for Paws request/responses.

5.8.2.11 BatchAvailableSpectrumConverter
Json converter for the Paws Batch Available Spectrum request.

5.8.2.12 BatchAvailableSpectrumRequest
Represents a paws batch available spectrum request.

5.8.2.13 BatchAvailableSpectrumRequestBase
Base class for PAWS Available Spectrum Batch request.

5.8.2.14 BlockedType
Enumeration of values indicating whether channel is blocked for white space usage.

5.8.2.15 CacheObjectPMSEAssignment
Represents a cached PMSE assignment object for Ofcom

5.8.2.16 CacheObjectTvEngdata
Represents a cached entry from the TV Eng Data table for FCC

5.8.2.17 Calendar
The calendar class is used to indicate the time of an event for LPAuxRegistrations and TempBASRegistrations.

5.8.2.18 CDBSAntennaPattern
Maps to an entry on the Antenna Pattern table in the FCC Consolidated Database System (CDBS).

5.8.2.19 CDBSApplication
Maps to an entry on the Application table in the FCC CDBS.

5.8.2.20 CDBSAppTracking
Maps to an entry on the Application Tracking table in the FCC CDBS.

5.8.2.21 CDBSFacility
Maps to an entry on the Application Tracking table in the FCC CDBS.

5.8.2.22 CDBSTableEntity
Base class for all entities used for the FCC CDBS database.

5.8.2.23 CDBSTvEngData
Maps to an entry on the TV Engineering Data table in the FCC CDBS.

5.8.2.24 ChannelInfo
Defines the information for a channel that may be available for use by a white space device.

5.8.2.25 Check
Contains utility methods for checking for null values in entities

5.8.2.26 CircleArea
Defines a circle of a given size. Used for various geographic functions within the application.

5.8.2.27 Constants
A static helper class of constants used with in the Entities components.

5.8.2.28 Contour
Entity class that contains the protection contours of an incumbent

5.8.2.29 ContourDetailsInfo
Contains contour details for an incumbent

5.8.2.30 ContourPointItem
Contains information about a specific contour point for an incumbent, such as the azimuth for the point, the Height Above Average Terrain (HAAT), etc.

5.8.2.31 DatabaseSpec
Contains the Paws Database spec.

5.8.2.32 DatabaseSpecConverter
Json converter Database spec

5.8.2.33 DBAdminInfo
The DB Admin Info is used for identifying all of the registered WSDBAs information.

5.8.2.34 DbUpdateSpec
Represent Class for Paws database update spec.

5.8.2.35 DbUpdateSpecConverter
Represent Json converter for paws device owner information.

5.8.2.36 DeviceCapabilities
Represent the paws device capabilities. Gets or sets the frequency ranges of the paws device.

5.8.2.37 DeviceDescriptor
Represent Paws Device Descriptor.

5.8.2.38 DeviceDescriptorConverter
Json converter for PAWS device descriptor.

5.8.2.39 DeviceId
Identifier for FCC device.

5.8.2.40 DeviceOwner
Contains information about PAWS device owner.

5.8.2.41 DeviceOwnerConverter
A Json converter for paws device owner information.

5.8.2.42 DeviceRegistrationMap
Represent a Device Registration Map. Gets or sets the Device Type and register Type of request.

5.8.2.43 DeviceValidity
Class for determining paws device validity.

5.8.2.44 DeviceValidityConverter
Represent Json converter for device validity.

5.8.2.45 DeviceValidityRequest
Represent a paws device validity request

5.8.2.46 DeviceValidityRequestBase
Represent the base class for a paws device validity request.

5.8.2.47 DistanceUnit
Enumeration for distance units of measure.

5.8.2.1 Distance
Defines a distance that has been calculated between two locations. Used for various geographic calculations within the application.

5.8.2.2 DistinctDTTQueryParams
Equality comparator for DTT availability data query parameters.

5.8.2.3 DistinctOSGLocations
Equality comparator for DTT availability data query parameterEquRepresent the class Distinct OSGLocations.

5.8.2.4 DttData
Struct for the DTT availability data for Ofcom

5.8.2.5 DTTDataAvailability
Represents an entry in the DTTAvailability data for Ofcom

5.8.2.6 DTTQueryParams
Struct that defines the parameters for query of DTT data for OFcom

5.8.2.7 DTTSquare
Defines a block of data from the Ofcom DTT Availability information

5.8.2.8 ElevatedAccessRequest
Defines a request for a given user to provide a higher level of access to the application. Not used in the current implementation.

5.8.2.9 ElevationCacheEntry
A cached elevation entry from the terrain data

5.8.2.10 Ellipse
Represents a Paws Ellipse field. Initializes a new instance of the Ellipse class.

5.8.2.11 EllipseConverter
Represents Ellipse Converter used to convert a Json reader into an Ellipse instance.

5.8.2.12 Email
Represents an email component of contact information.

5.8.2.13 Event
The Event class is used to identify the time for an LP Aux Registration or Temp BAS Registration.

5.8.2.14 EventRecurrence
Provides information about the recurrence of an event for an LP Aux Registration or Temp BAS registration.

5.8.2.15 EventTime
Provides information about the start and stop time of an event.

5.8.2.16 EventTimeConverter
Json converter for EventTime.

5.8.2.17 FixedTVBDRegistration
Contains information about the registration of a Fixed TV Band devi ce.

5.8.2.18 FortranDoubleArray
Represents Fortran Double Array for fortran. Indexing starts with 1 instead of 0.

5.8.2.19 FrequencyRange
Contains the Paws Frequency Range fields.

5.8.2.20 FrequencyRangeConverter
Custom JSON converter for Frequency range.

5.8.2.21 GeoLocation
Represents a Paws Geo Location instance. Initializes a new instance of the GeoLocation.

5.8.2.22 GeoLocationConverter

Json converter for GEOLocation object.

5.8.2.23 GeoSpectrumSchedule

Represents the paws geospectrum schedule.

5.8.2.24 GeoSpectrumScheduleConverter

Represents a Device Descriptor Json Converter.

5.8.2.25 GeoSpectrumSpec

Represents the GeoSpectrum Spec. Gets or sets the spectrum specs.

5.8.2.26 GetChannelsResponse

Defines the response information for a GetFreeChannels request.

5.8.2.27 IAvailableSpectrumRequest

Interface that defines the PAWS Available Spectrum Request. Interfaces are defined for all of the PAWS request types because they have slightly different implementations for FCC and Ofcom.

5.8.2.28 IBatchAvailableSpectrumRequest

Interface that represents a paws batch available spectrum request.

5.8.2.29 IDeviceValidityRequest

Interface that represents a paws device validity request.

5.8.2.30 IInitRequest

Interface that represents a paws init request.

5.8.2.31 IInterferenceQueryRequest

Interface defining the PAWS Interference Query request parameters.

5.8.2.32 Incumbent

Provides information about an existing entity that uses TV spectrum.

5.8.2.33 IncumbentType

Enumeration of values for type of incumbent.

5.8.2.34 InitializedDeviceDetails

Contains details about white space devices for which device initialization request has been processed.

5.8.2.35 InitRequest

Represents an PAW S init request.

5.8.2.36 InitRequestBase

Base class for PAWS init request.

5.8.2.37 INotifyRequest

Interface that defines the spectrum use notify request for PAWS.

5.8.2.38 InterferenceQueryConverter
JSON converter for Interference Query.

5.8.2.39 InterferenceQueryRequest
Contains the details about a PAWS Interference Query Request for Ofcom.

5.8.2.40 InterferenceQueryRequestBase
Base class for PAWS interference query request.

5.8.2.41 IParameters
Defines a generic interface for al l the paws parameters.

5.8.2.42 IRegisterRequest
Interface that defines the PAWS registration request.

5.8.2.43 JsonSerialization
Static class for Json serialization and deserialization

5.8.2.44 KeyHoleContourItem
Contains information about a KeyHole contour used for protection of incumbents that receive signal from a parent station.

5.8.2.45 Location
A lat/long pair that defines a geographic location. .

5.8.2.46 LpAuxLicenseInfo
Contains information about the licensing of an LPAUx device.

5.8.2.47 LPAuxRegistration
Contains registration information for an LPAux device

5.8.2.48 MVPDCallSignsInfo
Contains information about the call signs that are available for registration of an MVPD device.

5.8.2.49 MVPDRegistration
Contains registration information for a Multi-channel Video Program Distributor.

5.8.2.50 NextTransactionID
A value provided to other WSDBAs indicating the transaction id they are to send on their next DB Synchronization request.

5.8.2.51 NotifyConverter
Json converter for PAWS SPECTRUM_USE_NOTIFY request.

5.8.2.52 NotifyRequest
Represents the paws SPECTRUM_USE_NOTIFY request.

5.8.2.53 NotifyRequestBase
Base class for paws SPECTRUM_USE_NOTIFY request.

5.8.2.54 Organization
Element within the VCard entity that defines the organization with which a contact is associated.

5.8.2.55 OSGLocation
Represents a location within the UK Ordinance Survey coordinate system.

5.8.2.56 ParameterConverter
Represents JSon converter for paws parameters.

5.8.2.57 ParameterMissingException
Exception thrown when a parameter is found to be missing from a PAWS request or a RegionManagement requset.

5.8.2.58 Parameters
Represents all paws parameters.

5.8.2.59 ParametersBase
Represents Base class for all paws parameters.

5.8.2.60 ParsePollRequestResult
Response generated when parsing a DB Sync Poll Request from another WS DBA.

5.8.2.61 PawsResponse
Response object for PAWS requests.

5.8.2.62 PawsResponseConverter
Json converter for PAWS response.

5.8.2.63 PmseAssignment
Contains information about an Ofcom PMSE (Program Making and Special Event) assignment that requires protection from White Space devices.

5.8.2.64 Point
Represents a Paws Point instance.

5.8.2.65 PointConverter
The Point Converter is used to convert a Json reader into an Point instance.

5.8.2.66 Polygon
Represents a Polygon instance. Initializes a new instance of the Polygon.

5.8.2.67 PolygonConverter
Polygon Converter is used to convert a Json reader into an Polygon instance.

5.8.2.68 Position
The Position class represent a location on the globe (latitude and longitude).

5.8.2.69 PowerUnit
Enumeration of units of measure for power

5.8.2.70 Power

Class Represents a power value.

5.8.2.71 ProtectedDevice

A device that requires protection from white space devices.

5.8.2.72 ProtectedDeviceType

Enumeration of device types for protected device.

5.8.2.73 ProtectedEntity

An entity that has been registered for protection from white space devices. Returned as part of the public data for the whitespace database.

5.8.2.74 ProtectedEntityWithEvent

An entity that has been registered for protection from white space devices that has event information associated with it.

5.8.2.75 QuadrilateralArea

Area represents a bounding rectangle used for LP Aux Registration.

5.8.2.76 RadialHAT

Contains a collection of Radial HAAT values (Height above average terrain) for an azimuth of a given contour

5.8.2.77 RadiationCenter

Maps to the FCC CDBS radiation center table.

5.8.2.78 RegionAccess

Provides information about a user's level of access to the RegionManagement service.

5.8.2.79 RegionManagementResponse

The response object generated when a region management request is executed.

5.8.2.80 RegionManagementResponseConverter

Json converter for RegionManagement response

5.8.2.81 RegisteredDevice

A device for which a PAWSA registration request is being submitted.

5.8.2.82 RegisterRequest

PAWS device registration request.

5.8.2.83 RegisterRequestBase

Base class for PAWS registration request.

5.8.2.84 RegistrationDisposition

Contains information about the current status of an entity that is registered within the whitespace database.

5.8.2.85 Request

Base class used in PAWS service.

5.8.2.86 RequestConverter
Json converter for Request class.

5.8.2.87 Result
Base class used in PAWS service.

5.8.2.88 ResultConverter
Json converter for paws result.

5.8.2.89 RulesetInfo
Contains information about validation rules for FCC registration requests.

5.8.2.90 RulesetInfoConverter
Json converter for ruleset info.

5.8.2.91 RuleSetInfoEntity
Contains information about validation rules for FCC registration requests.

5.8.2.92 SearchCacheRequestType
Enumeration of types for searching of cache objects.

5.8.2.93 ServiceCacheObjectType
Enumeration of types that identify a cached object

5.8.2.94 ServiceCacheRequestParameters
Contains parameters to be used to search cache for a given object

5.8.2.95 Settings
Represents a configuration value used in the application.

5.8.2.96 Spectrum
Contains information about TV spectrum to be used by a whitespace device.

5.8.2.97 SpectrumConverter
Represents a Spectrum Json Converter.

5.8.2.98 SpectrumProfile
Represents a class Spectrum Profile.

5.8.2.99 SpectrumSchedule
Schedule information regarding the use of TV spectrum by a whitespace device

5.8.2.100 SpectrumScheduleConverter
Json converter for Spectrum schedule.

5.8.2.101 SpectrumSpec
Contains details about a specific entry of TV spectrum to be used by a whitespace device.

5.8.2.102 SquareArea
Used in various geographic calculations within the application.

5.8.2.103 SyncRow
Represents a row that is to be updated from the Protected Region Worker.

5.8.2.104 SyncTable
Represents a table that is to be updated from the Protected Region Worker.

5.8.2.105 Telephone
Telephone information within VCard data.

5.8.2.106 TempBASRegistration
Represents a Temp Broadcast Auxiliary Service Registration.

5.8.2.107 TransactionInfo
Transaction Info class is used to keep track of all WSDBA synchronization transactions.

5.8.2.108 TVReceiveSiteRegistrations
Contains information about registration of a TV Receive site.

5.8.2.109 TvSpectrum
Contains information about the use of TV spectrum by an entity registered within the whitespace database.

5.8.2.110 ULSRecord
Maps to a record in the FCC ULS database.

5.8.2.111 UsedSpectrum
Contains information about TV spectrum that is used by a device.

5.8.2.112 User
Contains information about a user of the application. Not used in the current implementation.

5.8.2.113 ValidationHelper
Helper class used for various validations in the application.

5.8.2.114 Vcard
VCard class represents a personal contact Information from a registration which is outlined below.

5.8.2.115 VcardConverter
Converts a json VCard representation into a VCard instance.

5.8.2.116 LocationRect
Indicates a bounding box having values for North, East, South and West values.

5.8.2.117 RegionPolygon
Indicates a sub region polygon entity of a region, uniquely identified by a Partition Key having region code.

5.8.2.118 RegionPolygonsCache
Contains cached border contours for a sub region.

5.9Microsoft.Whitespace.InterpolationLibrary

The InterpolationLibrary namespace contains a C# implementation of the FCC's FMTVCurves Fortran program that is used in the TV contour calculations for the FCC. This library is deployed as a 32-bit dll to ensure that the precision of the resulting calculations yields the same results as the corresponding calculations in the Fortran program. This namespace contains a set of classes that have been converted from the corresponding Fortran sub-routines. These classes are used to determine field strength along a radial at a given distance from an incumbent. The field strength is used to determine the length of the protection contour along each azimuth around the incumbent.

5.9.1.1 BivariateInterpolation

This is the C# version of the core part of the interpolation routine.

5.9.1.2 CurveCalculator

This class is the public interface for the Interpolation library dll. It is called to calculate the field strength at a given distance along the azimuth for an incumbent.

5.9.1.3 F2DArray

Represents class F2DArray (2Dimensional) for fortran. Indexing starts with 1 instead of 0.

5.9.1.4 FArray

Represents class FArray for fortran. Indexing starts with 1 instead of 0.

5.9.1.5 FCCDataMatrix

This is a static class that defines all of the FCurve Interpolation data points that are described in the Whitespace Channel Calculations guidelines.

5.9.1.6 SharedVariables

Defines the set of shared variables that are used throughout the interpolation routines.

5.9.1.7 ZA

Direct C# conversion of the corresponding Fortran sub-routine.

5.9.1.8 ZAB

Direct C# conversion of the corresponding Fortran sub-routine.

5.9.1.9 ZB

Direct C# conversion of the corresponding Fortran sub-routine.

5.9.1.10 ZX

Direct C# conversion of the corresponding Fortran sub-routine.

5.9.1.11 ZXY

Direct C# conversion of the corresponding Fortran sub-routine.

5.9.1.12 ZY

Direct C# conversion of the corresponding Fortran sub-routine.

5.10 Microsoft.Whitespace.NativeCodeHostService

This assembly is included to enable invocation of a 32-bit dll from the application within Azure. It is required because the C# version of the FCC interploation routine must be run as a 32-bit dll to achieve the correct precision in the results.

5.10.1 NativeCodeHostServicef

5.10.1.1 INativeMethodService

Interface for the GetDistance() method of the CurveCalculator

5.10.1.2 NativeMethods

Implements the INativeMethodService to call the curve calculator

5.11 Microsoft.Whitespace.PMSESync

5.11.1 PMSESync

The PMSESync assembly contains the classes that perform synchronization of PMSE assignment data for Ofcom. .

5.11.1.1 ProcessPmseAssignmentFiles

Performs the actual synchronization process for the PMSE Assignment data.

5.11.1.2 WorkerRole

Worker role that invokes the PMSE Assignment Synchronization process at regular intervals.

5.12 Microsoft.Whitespace.RegionCalculation

5.12.1 RegionCalculation

Each of the supported region will implement the “BaseRegionCalculations” base class as defined in the following figure below.

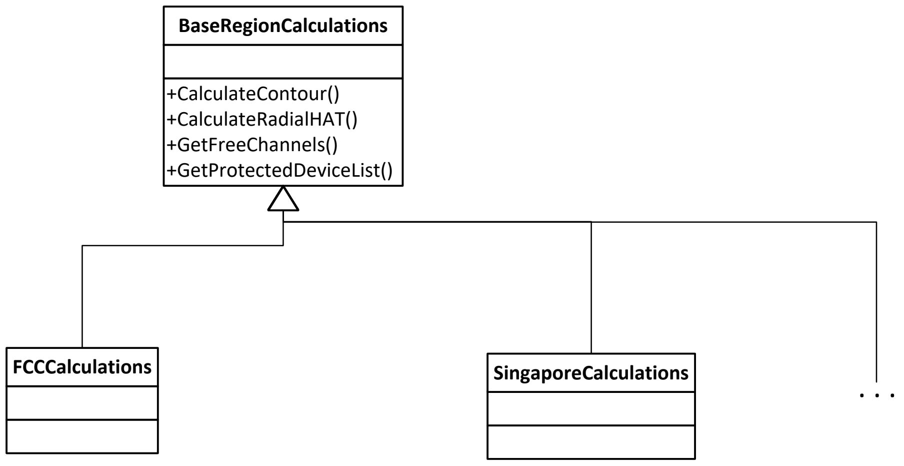


Figure 5 - Region Calculation Hierarchy

Assembly Name	Microsoft.Whitespace.RegionCalculation.dll
Source Directory	src\Whitespace\RegionCalculation
Root Namespace	Microsoft.Whitespace.RegionCalculation
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common

The BaseRegionCalculations will provide a default implementation for the CalculateContour and CalculateRadialHAT methods. But the derived region classes will need to implement the GetFreeChannels and GetProtectedDeviceList methods in order to provide their custom logic (for instance, the US has special rules around the Canadian and Mexican border that would not apply to other regions such as Singapore).

5.12.1.1 FccCalculation

The class FCCCalculation implements the IRegionCalculation interface for the US implementation of the whitespace database. This class has been developed based on the Whitespace Channel Calculations Guidelines document.

The table below provides details about the implementation of selected required methods within the IRegionCalcuation interface,

Method Name	Description	Comment
CalculateContour	Calculates the protection contour for a specific incumbent	This method is called when new incumbent information is added from the FCC. FCC propagation curves methods (see below) are called to calculate the contour points.
CalculateRadialHAT	Calculates the height above average terrain for the incumbent	This is an internal method, that is only called within the FccCalculation class
GetChannelAvailability	Determine the availability of each channel at the specified position	Not implemented in this code drop
GetFreeChannels	Returns a list of channels that are free at the specified location	See "Protection of Incumbents" below

5.12.1.2 Protection of Incumbents

The GetFreeChannels method of the FCCCalculation class implements the algorithms defined in the Channel Calculations For White Spaces Guidelines document to ensure that incumbents are protected from interference by white spaces devices. Channel availability is determined by examining the protection requirements for each of the incumbent types as described in the table below. When a channel is determined to be protected for a given incumbent type, it is removed from the available list, and further checking for that channel is terminated.

Incumbent Type	Protection Approach
Radio Astronomy Site	For the first 15 entries in the list of radio astronomy sites, the point-to-point distance is calculated between the lat/long of the white space device and the lat/long of each of the defined radio astronomy sites. If the white space device is within the specified radius for any of the sites, an empty channel list is returned. This blocks all WSD activity within the specified radius.

	For the last radio astronomy site in the list a quadrant is created using the coordinates provided in the Channel Calculations guide. If the requesting WSD is within the quadrant, en empty channel list is returned.
Offshore Radio Service	A quadrant is created for each of the offshore radio service channels (15, 16, 17, 18), using the locations that are specified in the channel calculations guidelines. If the white space device is within any of the the quadrants, the channel for the associated quadrant is removed from the list of available channels.
T-Band Services	The point-to-point distance is calculated between the lat/long of the white space device and the lat/long of each of the T-band protection sites. For each t-band site, if the white space device is within 134 kilometers of the site, the t-band site's channel is removed from the available channel list. If the white space device is within 131 kilometers of the site, the adjacent channels are also removed from the list of available channels.
TV Stations	1. A square that is 400 km on a side is created based on the latitude and longitude of the requesting white space device. 2. Information is retrieved from the database for all of the incumbent tv stations where the latitude and longitude are within the square 3. For each incumbent that is retrieved in step 2, a polygon is created from the contour points that are associated with the incumbent. If the location of the requesting white space device is within the polygon, the associated channel is removed from the list of available channels. 4. If the white space device is not within the polygon but it is within the specified "buffer" distance from the polygon for co-channel operation, the incumbent's channel is removed from the list of available channels. 5. If the white space device is not within the polygon but it is within the specified "buffer" distance from the polygon for adjacent channel operation, channels that are adjacent to the incumbents channel are removed from the list of available channels. Note that adjacent channels are based on the channel's frequency, not based on the channel number.
Broadcast Auxiliary Stations (includes permanent sites retreieved from ULS and temporary BAS registrations)	1. A square that is 160 km on a side is created based on the latitude and longitude of the requesting white space device. 2. Information is retrieved from the database for all of the broadcast auxiliary sites where the latitude and longitude are within the square. This includes both permanent BAS sites that were retrieved from the ULS and temporary BAS registrations. 3. For each incumbent that is retrieved in step 2, the point-to-point distance between the broadcast auxiliary site and the white space device is calculated. If the distance is less than 8 kilometers, the channel associated with the broadcast auxiliary site will be removed from the list of available channels. If the distance is less than 2 kilometers, the channels adjacent to the broadcast auxiliary site's channel will also be removed from the list. 4. If the channel has n0t been excluded yet, the bearing between the BAS site and its transmitter site is calculated. A 60-degree arc is then calculated, using the calculated bearing +- 30 degrees. If the requesting white space device is within the arc, and the distance between the WSD and the BAS site is 80 km or less, the channel associated with the BAS site is removed from the list of available channels. In addition, if the WSD is withing the arc and the distance between the WSD and the BAS site is 20 km or less, the channels adjacent to the BAS site's channel are removed from the list of available channels.
Wireless Microphones	1. A square is created based on the based on the latitude and longitude of the requesting white space device. If the requesting device is a fixed WSD, the square will be 2 km on a side, else the square will be .8 km on a side. 2. The LP Aux registration tables are queried for registrations on which the latitidue and longitude are within the area of the above square, where the start date of the event is greater than or equal to the current date/time. For each LP Aux registration that is retrieved, the associated channel is removed from the list of available channels.
PLMRS and CMRS Stations	1. A square that is 54 km on a side is created based on the latitude and longitude of the requesting white space device. 2. The ULS data is queried for entities of the specified type (Land Mobile Commercial or Land Mobile Private) that are within the coordinates of the square that was created in step 1 above.

	3. For each entity that is returned from step 2, the distance between the land mobile site and the requesting WSD is calculated. If the distance is 54 km or less, the channel associated with the Land Mobile site is removed from the list of available channels. If the distance is 51 km or less, the adjacent channels are also removed from the list.
MVPD Receive Site	1. A square that is 200 km on a side is created based on the latitude and longitude of the requesting white space device. 2. All registered MVPD sites that are within the square are retrieved. 3. For each MVPD site that is retrieved, the following is performed: a. If the distance from the white space device to the MVPD site is 8 km or less, the MVPD site's receive channel is removed from the available channel list. b. If the distance from the white space device to the MVPD site is 28 km or less, channels on which the frequency is adjacent to the the MVPD site's receive channel are removed from the available channel list. c. The bearing from the MVPD sites to its parent stations is calculated. d. An arc is constructed from +30 degrees to -30 degrees from the bearing. e. A keyhole contour is created using the arc, extending out 80 km from the location of the MVPD site f. If the white space device is within the arc created above, and the distance from the device to the MVPD site is 80 km or less, the MVPD site's receive channel is removed from the list of available channels. g. If the white space device is within the arc created above, and the distance from the device to the MVPD site is 20 km or less, channels on which the frequency is adjacent to the MVPD site's receive channel are removed from the list of available channels.
Reserved Channels For Wireless Microphone	After the available channel list has been adjusted based on protection of all incumbents, 2 channels are reserved for wireless microphone use. The reserved channels are the two closest channels on either side of channel 37 that are not already blocked for some other incumbent.

5.12.1.3 OfcomCalculation

The class OfcomCalculation implements the IRegionCalculation interface for the Ofcom implementation of the whitespace database to perform the necessary protection algorithms for Ofcom.

5.12.2 Propagation

5.12.2.1 ClutterEnvironment

An enumeration of the Clutter Environment values for Ofcom

5.12.2.2 OFCOMMatrix

Provides static values for protection ratios for Ofcom device types.

5.12.2.3 PMSEEquipmentType

Enumeration of PMSEEquipmentType values for Ofcom.

5.12.2.4 PMSEUsageNature

Enumeration of PMSE Usage Nature values for Ofcom.

5.12.2.5 OperationalParameter

Enumeration of device operational parameters.

5.13 Microsoft.Whitespace.RegionManagement

5.13.1 RegionManagement

The RegionManagement contains the controller class that manages the execution of all of the methods of the Region Management service . The control manages the process for:

- Validating and Parsing JSON parameters
- Validating device Ids by querying the whitespace azure tables
- Invoking the appropriate Region Management Driver method to process the request.

5.14 Microsoft.Whitespace.Terrain

5.14.1 Terrain

The Terrain assembly contains the classes that are used to read various terrain files that are used in the application.

5.14.1.1 ClutterDatasetReader

Implements the ITerrainReader interface to read clutter information for Ofcom deployment.

5.14.1.2 ElevationCache

Manages the caching of terrain data.

5.14.1.3 OrdnanceSurveyOSReader

This class reads USGS1ArcReader, 1arc second interval data.

5.14.1.4 SRTMReader

Reads the SRTM terrain data for FCC. This class is currently not used in the application; the SRTM data has been replaced by the USGS 1 Arc second data.

5.14.1.5 USGS1ArcReader

Reads Terrain data from the USGS 1Arc second terrain files.

5.15 Microsoft.Whitespace.Terrain Cache

The Terrain Cache is responsible for returning elevation data for a specified location.

Assembly Name	**Microsoft.Whitespace.Terrain.dll**
Source Directory	src\Whitespace\Terrain
Root Namespace	Microsoft.Whitespace.Terrain
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Dalc, Microsoft.Whitespace.Common

The terrain storage will be populated from files located at the http://seamless.usgs.govfor US data and http://www.geobase.cafor Canadian data.

Each of the original terrain files are between 72 and 123 MB stored. On average, loading one of the 72MB terrain files from an Azure Blob Storage take ~9 seconds (typically, multiple terrain files would need to be loaded for each region resulting in Incumbent calculations taking over 50 seconds in a cold-boot scenario). The following enhancements were made to help improve performance:

- Divided each terrain file into 4MB chunks
- Implemented 3 layers of caching: Memory, Azure, and Blob Storage

5.15.1.1 Caching

The Whitespace DB has three levels of caching:

- **In-Memory Cache**: a dictionary with the index being the "file name" and the value being the "file content" stored as a 4MB byte array. The in-memory cache is only shared within the same session of a web service call.
- The **Azure Cache** is a dedicated worker role that stores the terrain data up to 12 hours and is shared amongst all of the web service sessions. The Azure Cache is also a dictionary of the file content stored as a 4MB byte array (Note that Azure Cache has a maximum size limit of 8MB for each individual object). The Azure Cache worker role is configured as "small" which has a limit of 1.75 GB of memory usage and 224 GB of disk space for the caching session).
- The BLOB Storage is where all of the terrain data are loaded if not found in Memory or Azure Cache.

The following 2 flowcharts shows the general workflow for retrieving an elevation data and saving the elevation file into the in-memory and an azure cache:

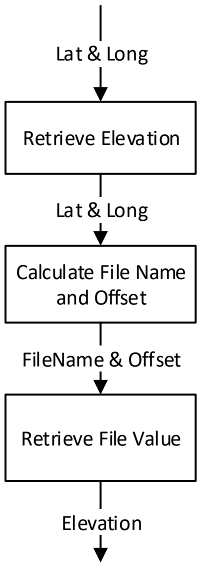


Figure 6 - Flowchart for retrieving elevation data point.

The "Retrieve File Value" block above can then be further broken into the following flowchart:

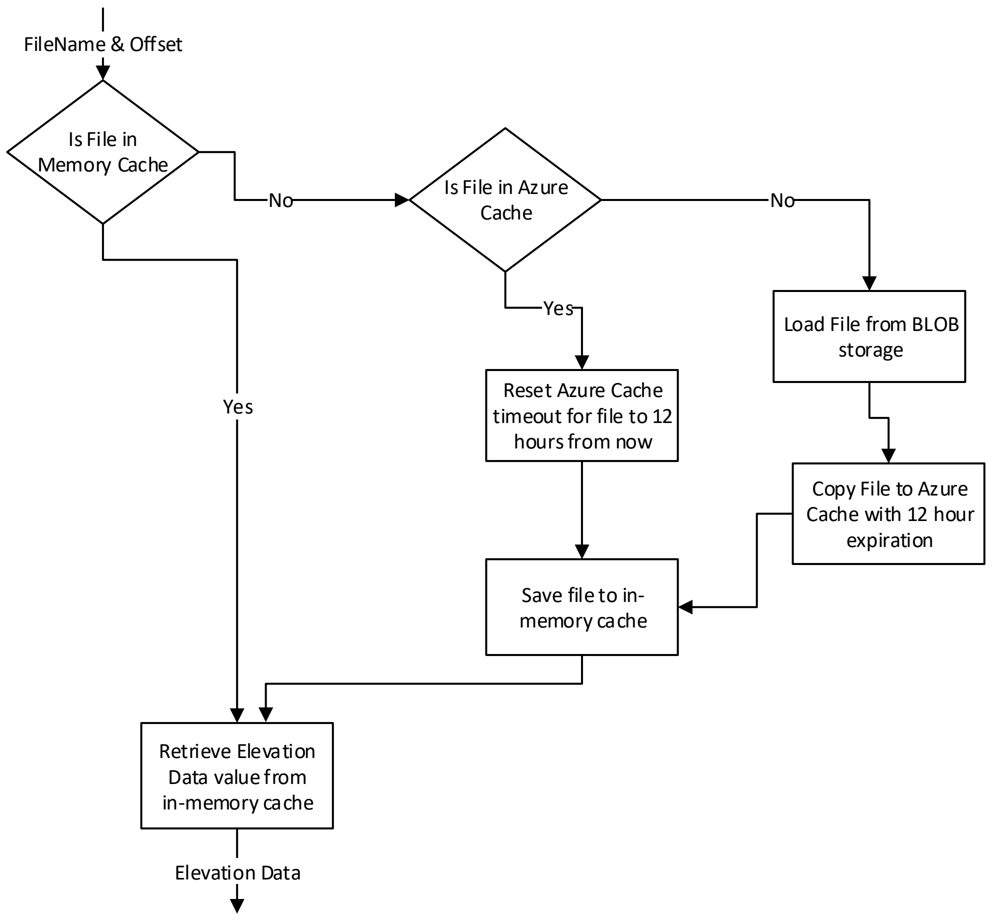

FileName & Offset
Is File in Memory Cache
No
Is File in Azure Cache
No
Yes
Load File from BLOB storage
Reset Azure Cache timeout for file to 12 hours from now
Yes
Copy File to Azure Cache with 12 hour expiration
Save file to in-memory cache
Retrieve Elevation Data value from in-memory cache
Elevation Data

Figure 7 - Retrieve File Value Flowchart

5.16 Web Service

The web.config file for the web services will contain a region identifier code that indicate which country is represented by the hosting service. This region Id code will be used by the internal components (PAWS Manager, White Space Driver, DB Sync Manager) in order to identify which Azure tables should be queried.

5.16.1 PAWS Web Service

The PAWS web service interface is documented at: http://tools.ietf.org/html/draft-ietf-paws-protocol-06. There are currently six methods that makeup the PAWS interface which are:

- **INIT_REQ**: Allows for a device to initiate exchange of capabilities with the white space DB.
- **REGISTRATION_REQ**: Allows for a device to register itself with the white space DB.
- **AVAIL_SPECTRUM_REQ**: Allows for a device to obtain a list of available spectrum at a given location.
- **AVAIL_SPECTRUM_BATCH_REQ**: Allows for a device to obtain a list of available spectrum at multiple locations.
- **SPECTRUM_USE_NOTIFY**: Allows for a device to reserve a spectrum with the white space DB.
- **DEV_VALID_REQ**: Allows for a mater device to validate a slave device.

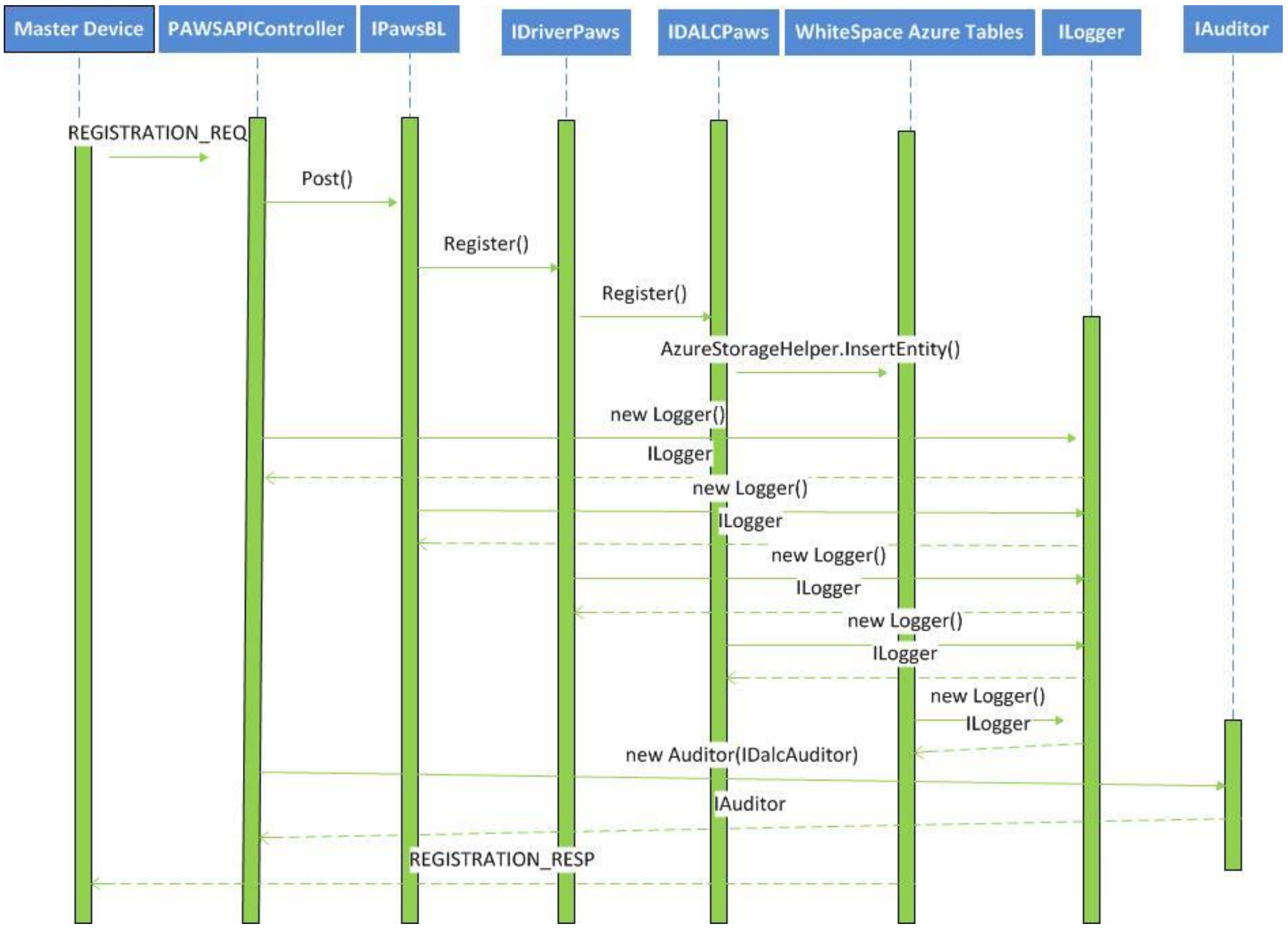
- **INTERFERENCE_QUERY_REQ**: Allows for a master device to send interference query request

Assembly Name	**Microsoft.Whitespace.PAWS.Service.dll**
Source Directory	src\Whitespace\PAWS\Service
Root Namespace	Microsoft.Whitespace.Paws.Service
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Common, Microsoft.Whitespace.PAWs.Manager

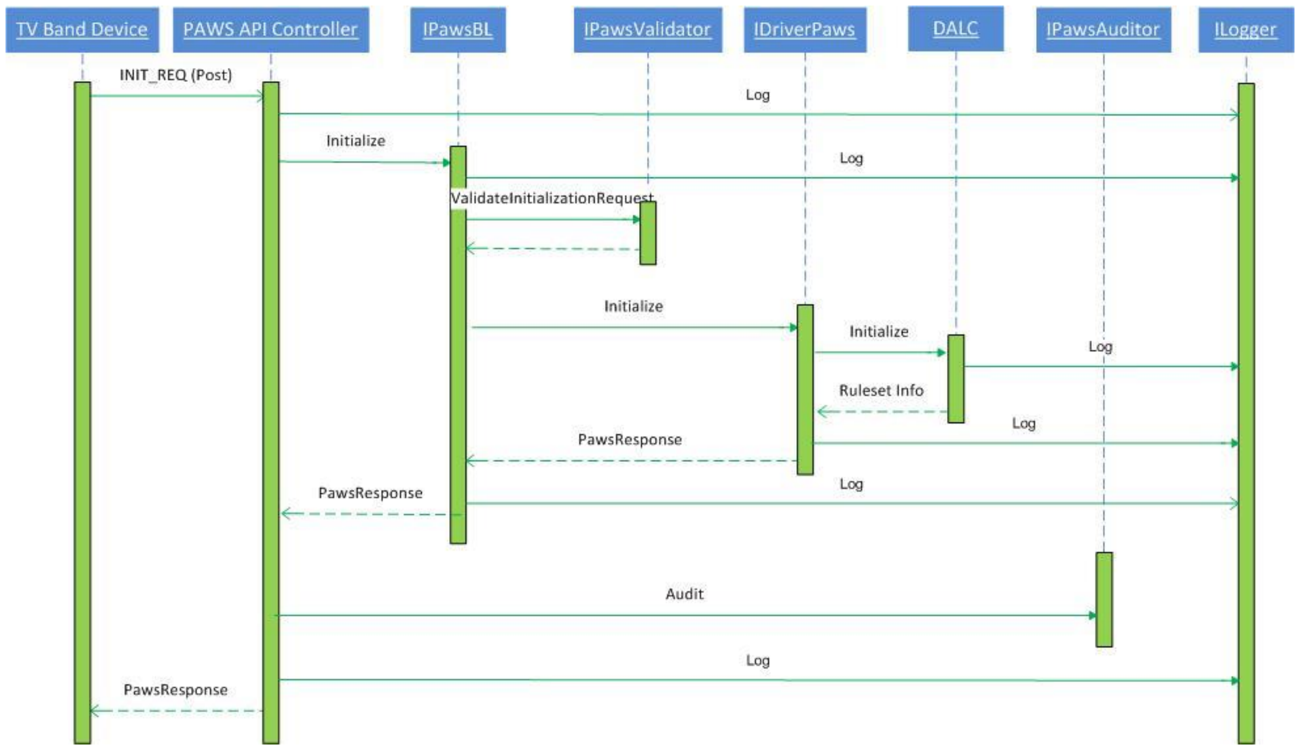
5.16.1.1 Sequence Diagrams

The sequence diagrams below provide a high-level view of each of the methods of the PAWS interface

REGISTRATION_REQ

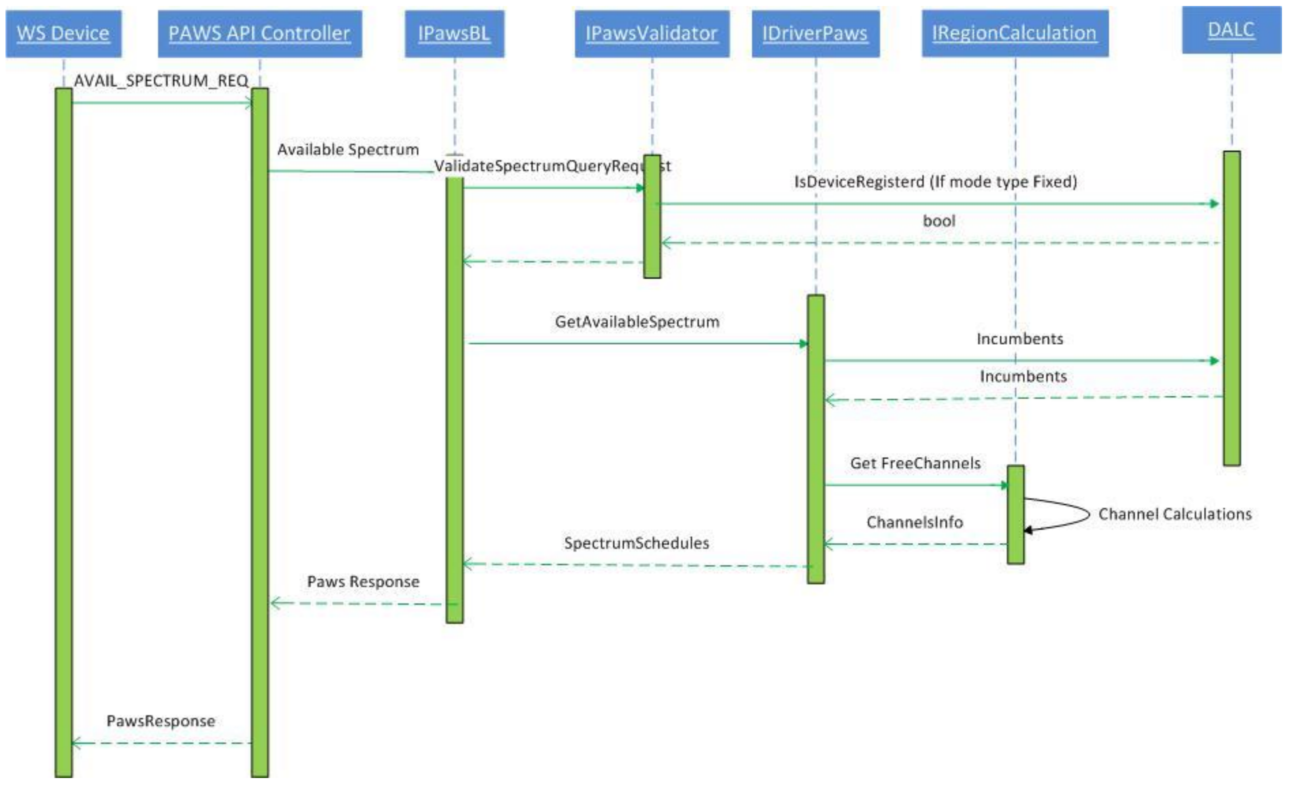


INIT_REQ

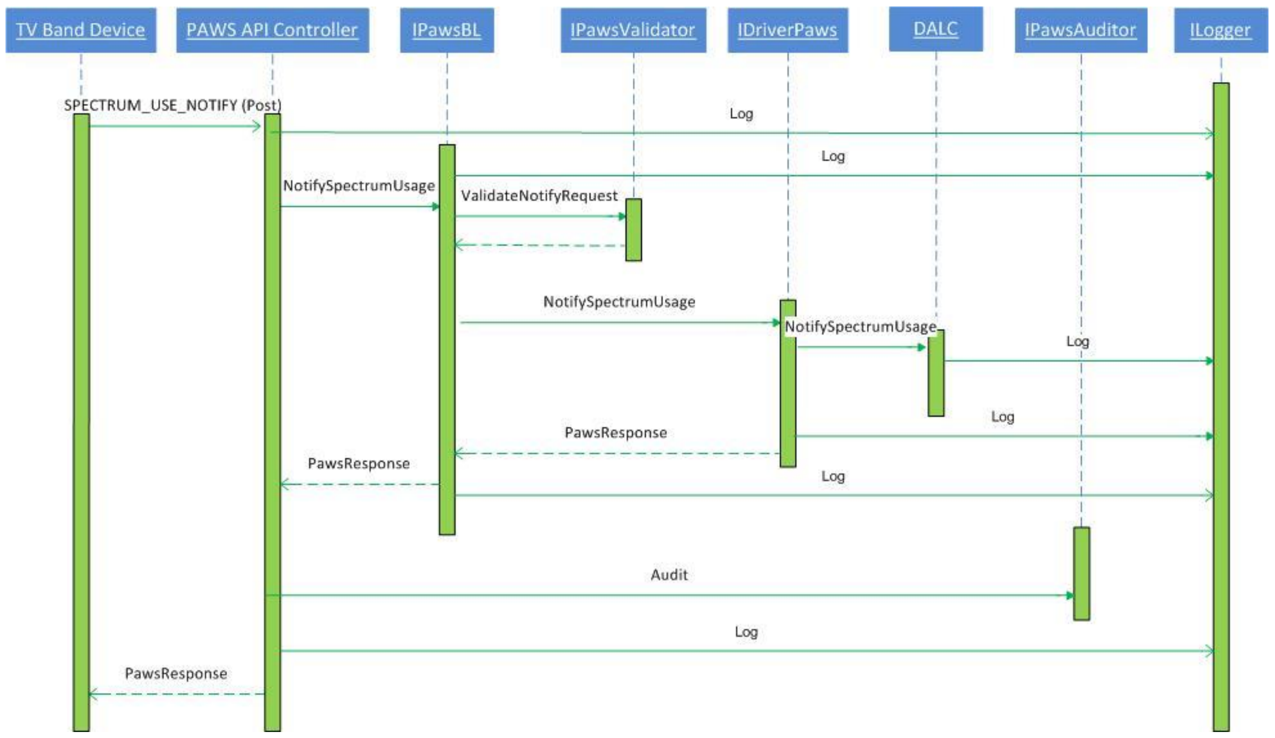

TV Band Device
PAWS API Controller
IPawsBL
IPawsValidator
IDriverPaws
DALC
IPawsAuditor
ILogger
INIT_REQ (Post)
Log
Initialize
Log
ValidateInitializationRequest
Initialize
Initialize
Log
Ruleset Info
Log
PawsResponse
Log
PawsResponse
Audit
Log
PawsResponse

AVAIL_SPECTRUM_REQ, AVAIL_SPECTRUM_BATCH_REQ

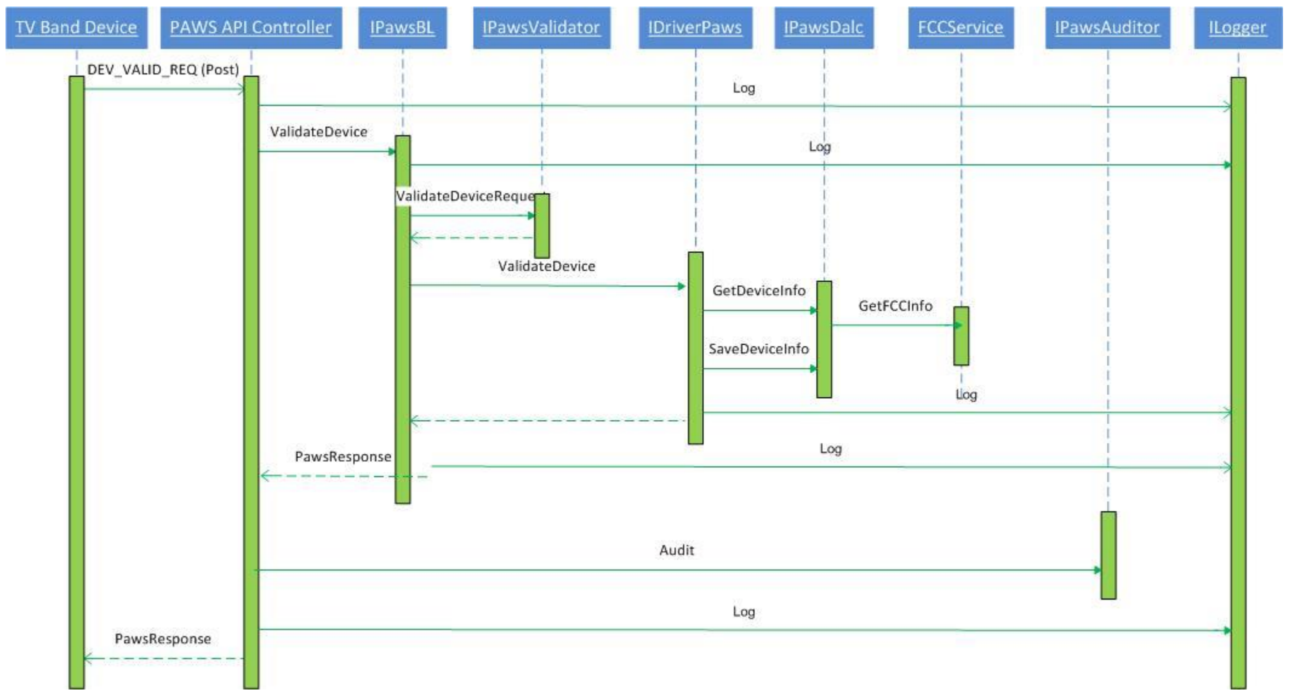
Note that the interactions between classes are identical for these two PAWS requests. For the AVAIL_SPECTRUM_BATCH_REQ, the Whitespace Driver iterates through the list of locations that is present in the request, while for the AVAIL_SPECTRUM_REQ, the driver just processes the single location that is present in the request.



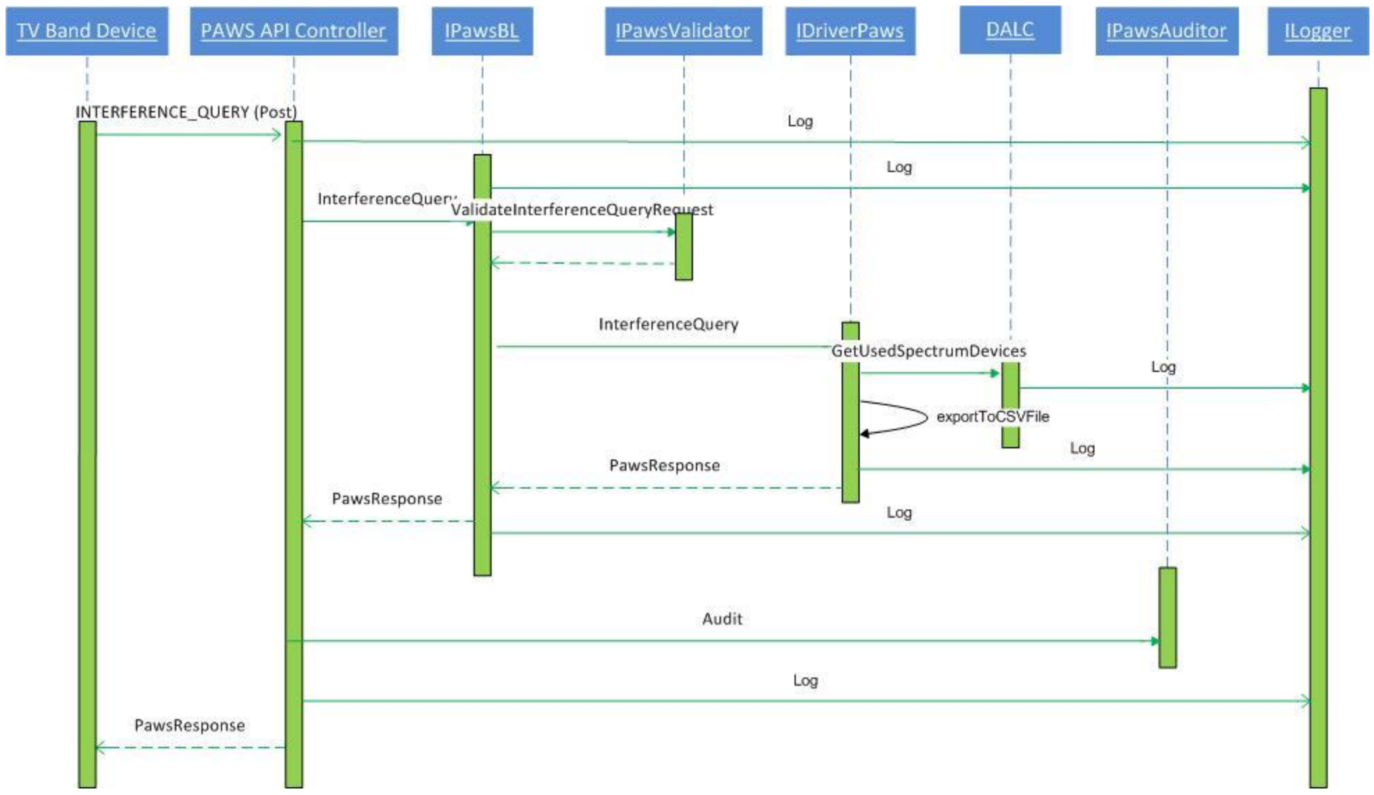
SPECTRUM_USE_NOTIFY



DEV_VALID_REQ



INTERFERENCE_QUERY_REQ (Ofcom only)



5.16.2 Region Management Web Service

The following web methods will be exposed by the region management web service:

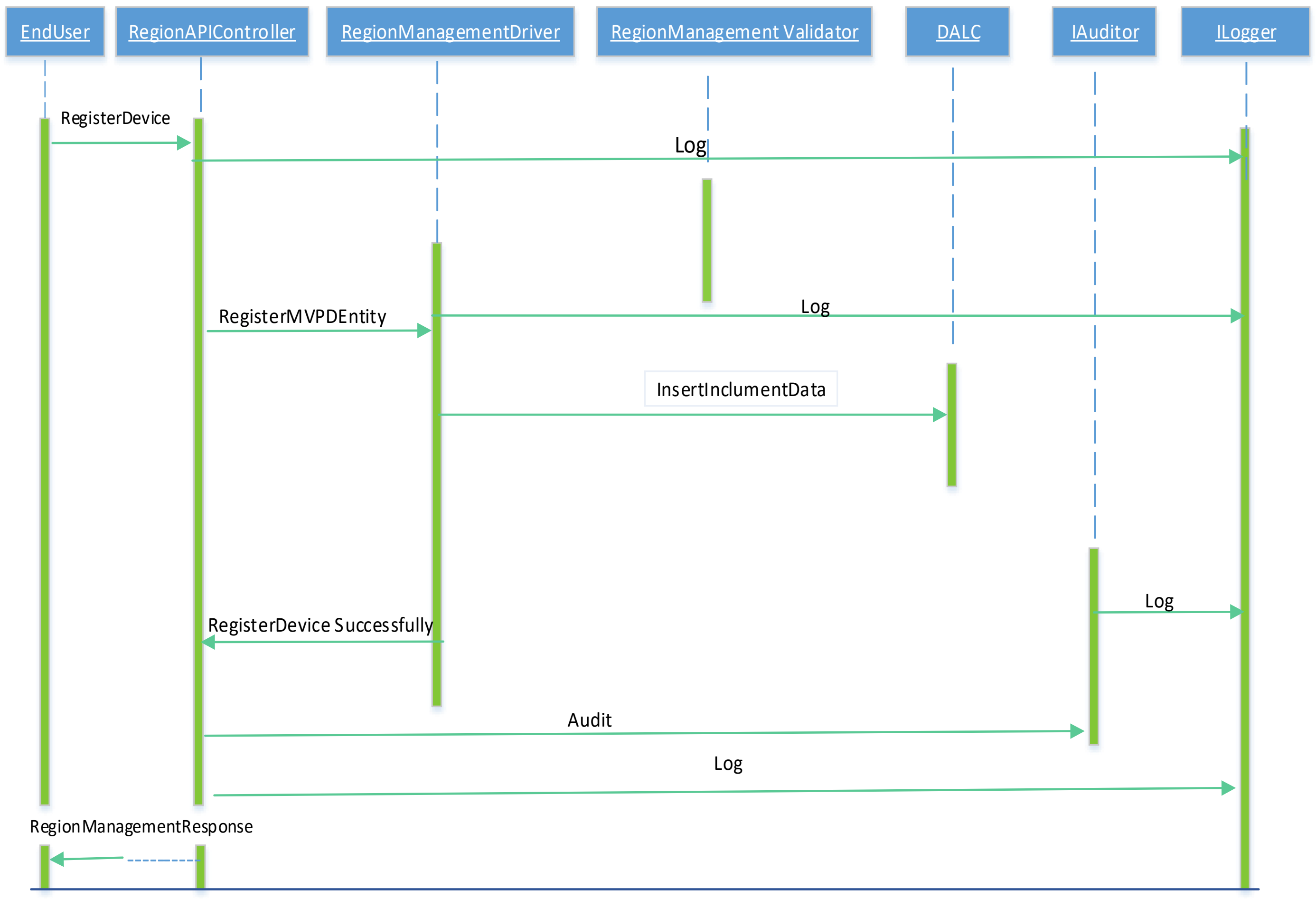
- **Register Device** – Allows a user to add a new incumbent for protection from interference from White Space devices. The incumbent types that can be registered are MVPD, LP Aux (both licensed and unlicensed) and Temp BAS.
- **DeleteIncumbent -** Deletes a given incumbent from the registration database.
- **GetDeviceList** – returns a list of all protected devices (incumbents such as TV Towers, wireless mics, …) that occupy channels at a given location. This method will return multiple arrays of incumbents, with a separate array for each incumbent type. The resulting list will contain information for all of the incumbents that are determined to be protected for the specified position as a result of the channel calculations.
- **GetChannelList** – returns a list of all available "free" channels at a given location. The returned list of channels will indicate the type(s) of device that are permissible at that location (i.e. fixed, mobile, none).
- **ExcludeChannel** – Exclude white space registration for a given list of channels in a given location (a set of lat/long points will be passed in for the excluded channels).
- **ExcludeIds** – Excludes a list of Ids from reserving white spaces (Ids will be region specific such as FCC Id).
- **GetULSCallSigns -** returns a list of ULS call signs that can be used for registration of licensed LPAux devices.
- **GetULSFileNumbers** – returns a list of ULS File Numbers that can be used for registration of an unlicensed LP Aux device.
- **SearchMvpdCallSigns** – used to get the list of TV stations call signs that can be used for registration of an MVPD site.
- **GetPublicData**– – returns a list of incumbents (ProtectedEntity objects) that is contained in the whitespace database. Used to provide public visibility to the contents of the whitespace database
- **GetPublicDataWithEvents** – returns a list of incumbents (ProtectedEntityWithEvents objects) that is contained in the whitespace database. Used to provide public visibility to the contents of the whitespace database.
- **GetAuthorizedDeviceModels**– returns a list of the wireless device models that have been approved by the FCC
- **GetContourData –** – returns the set of 360 lat/long pairs that define the protection contours for a particular incumbent.

Assembly Name	**Microsoft.Whitespace.RegionManagement.dll**
Source Directory	src\Whitespace\RegionManagement
Root Namespace	Microsoft.Whitespace.RegionManagement
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Common, Microsoft.Whitespace.Driver, Microsoft.Whitespace.UserManager

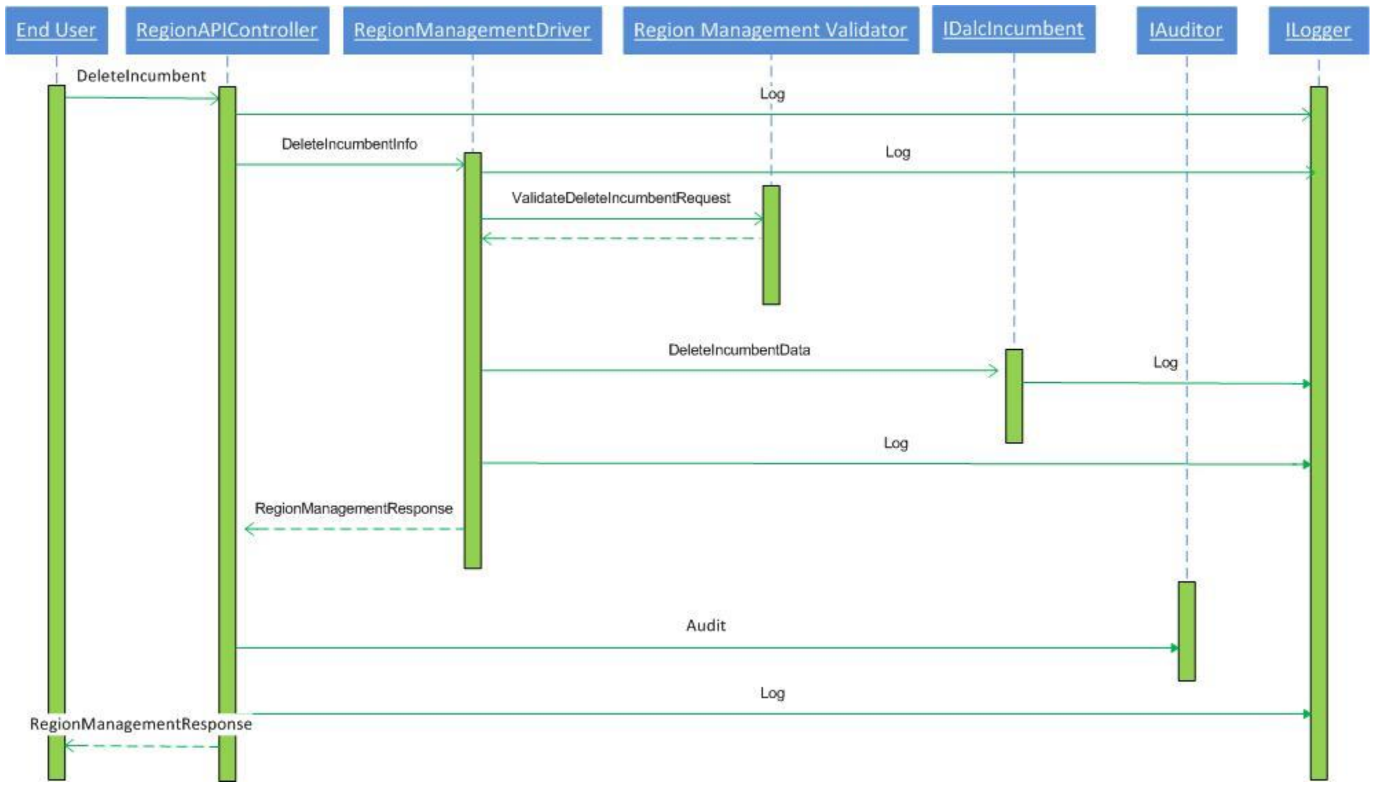
5.16.2.1 Sequence Diagrams

The sequence diagrams below provide a high-level overview of each of the methods of the Region Management service.

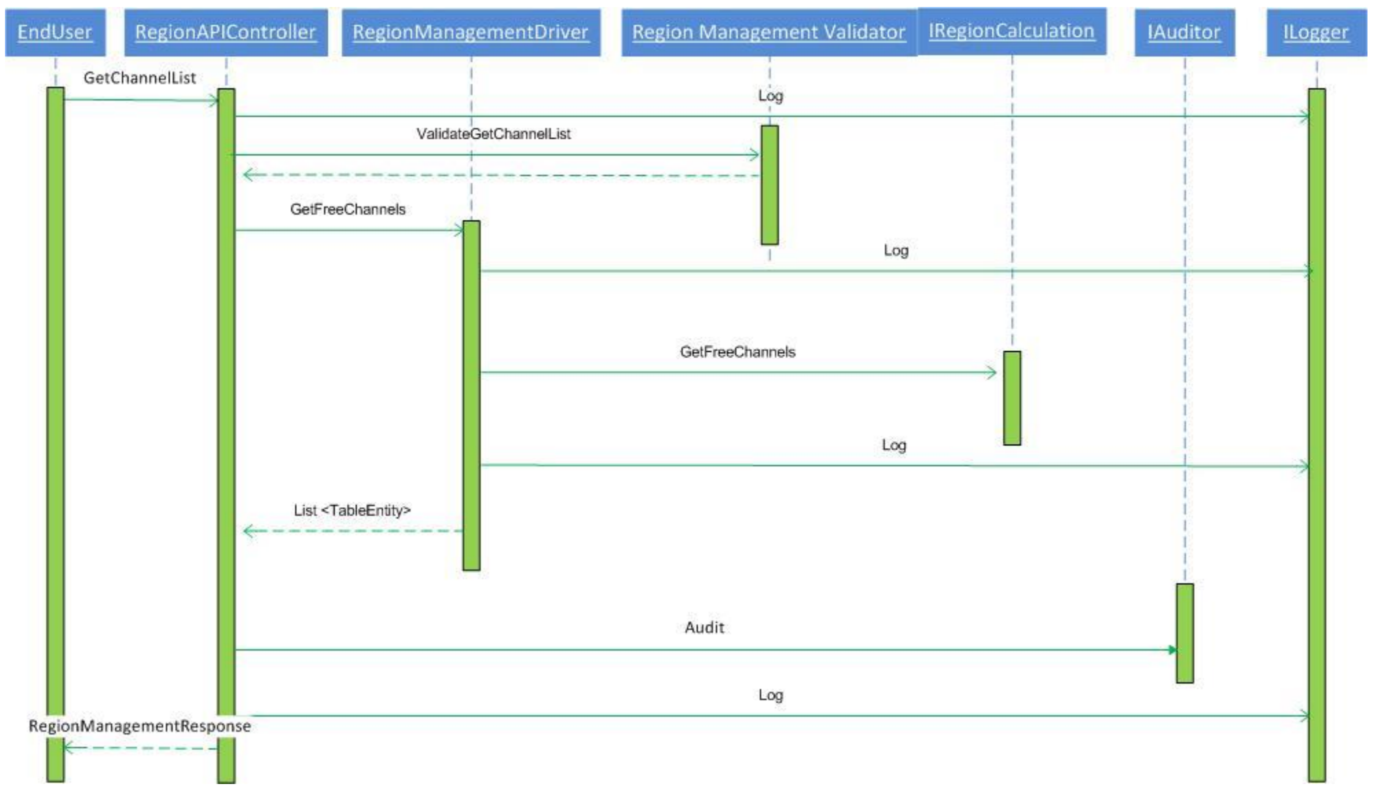
Register Device



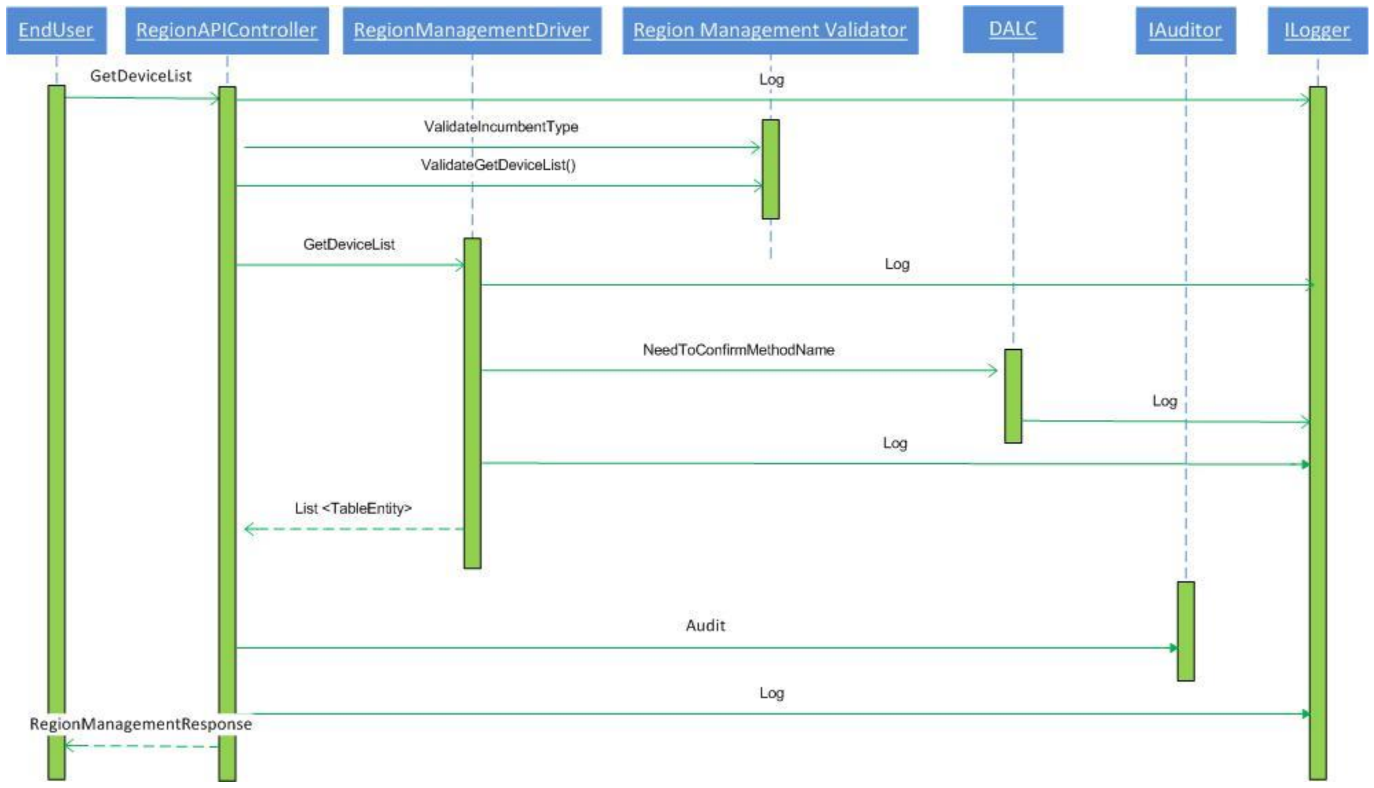
Delete Incumbent


End User
RegionAPIController
RegionManagementDriver
Region Management Validator
IDalcIncumbent
IAuditor
ILogger
DeleteIncumbent
Log
DeleteIncumbentInfo
Log
ValidateDeleteIncumbentRequest
DeleteIncumbentData
Log
Log
RegionManagementResponse
Audit
Log
RegionManagementResponse

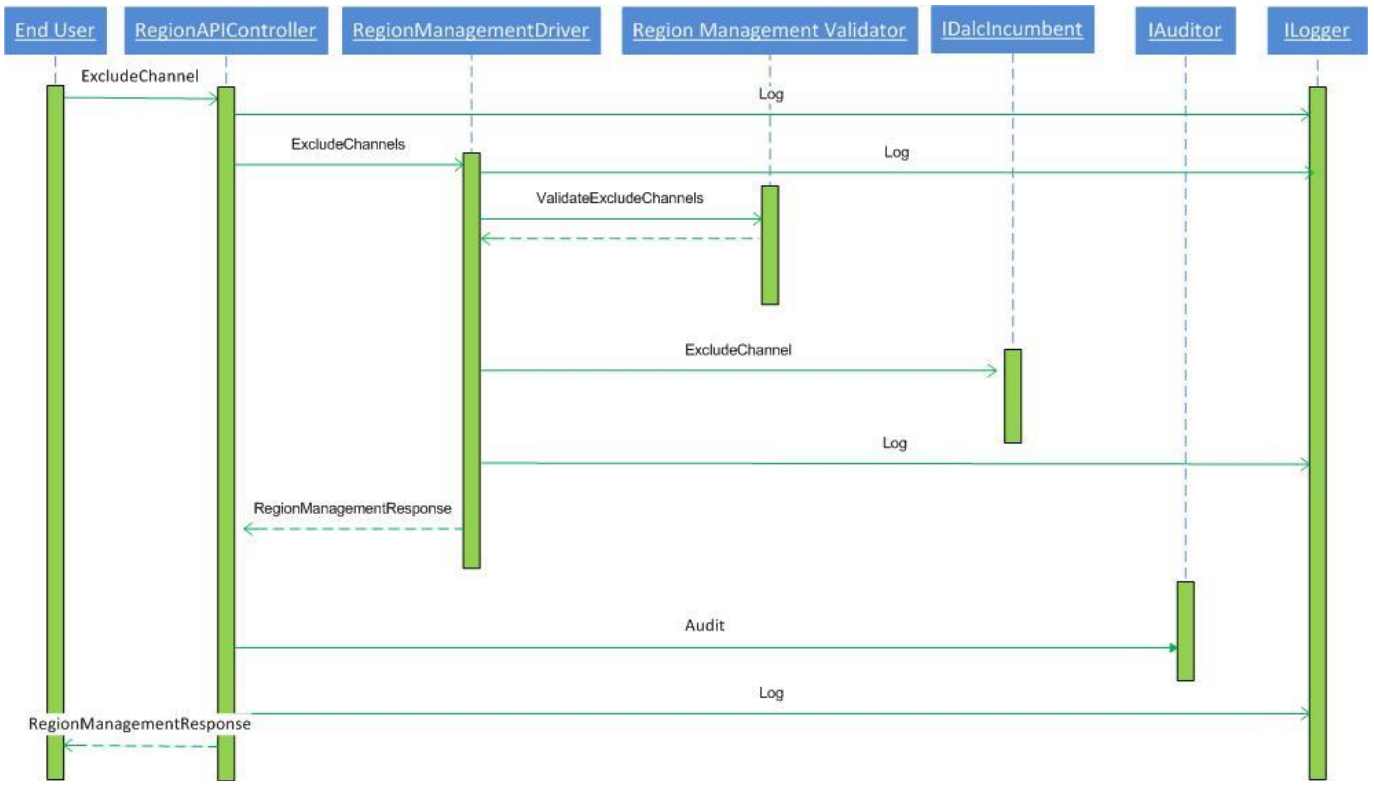
GetChannelList



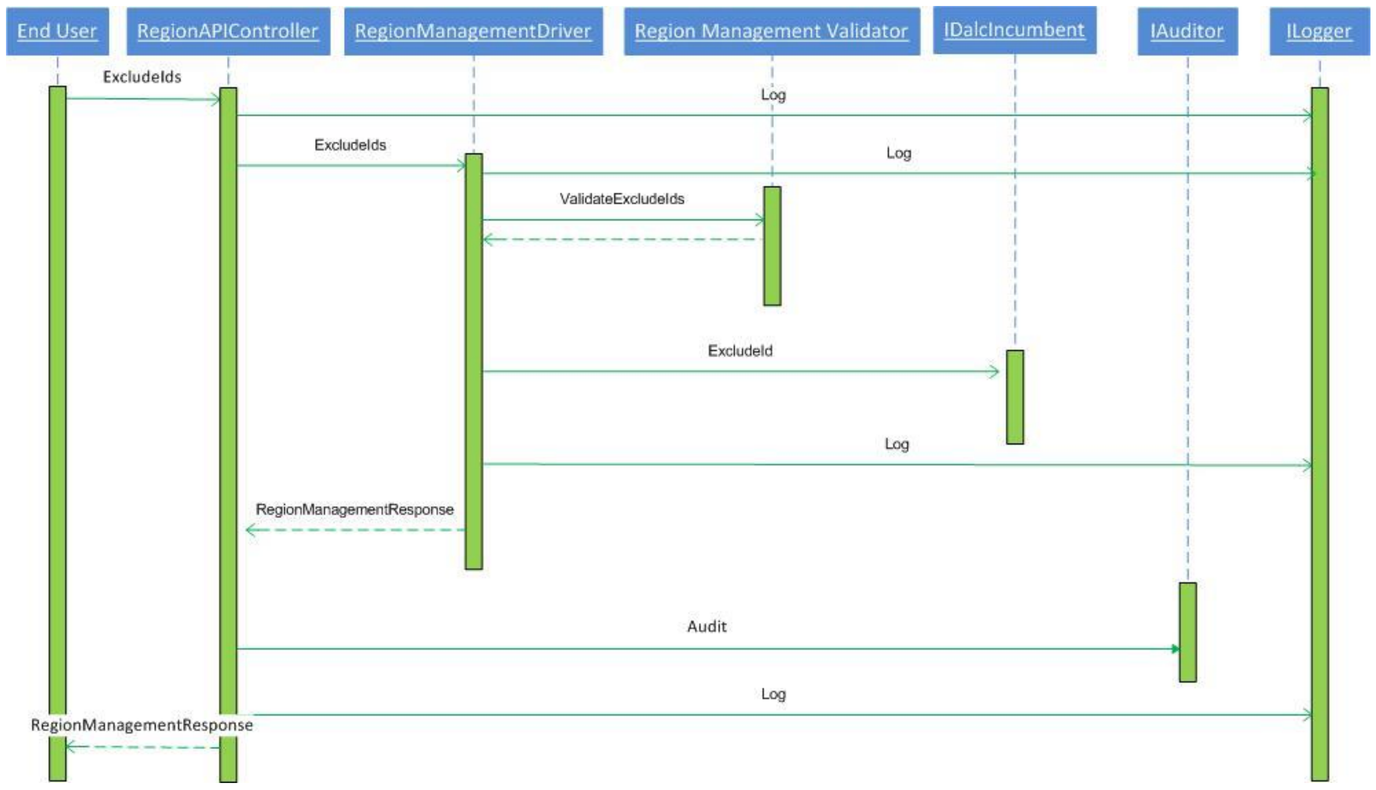
GetDeviceList



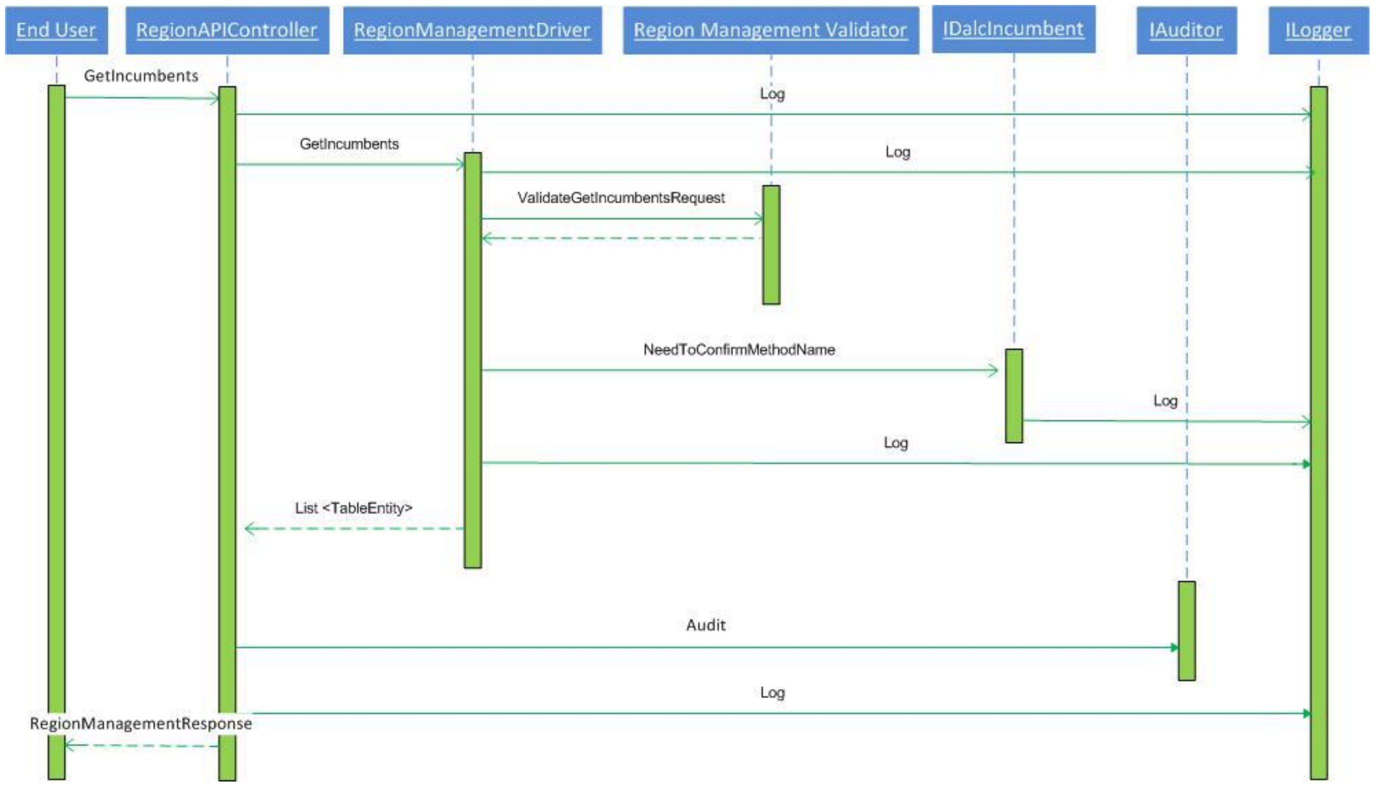
ExcludeChannel



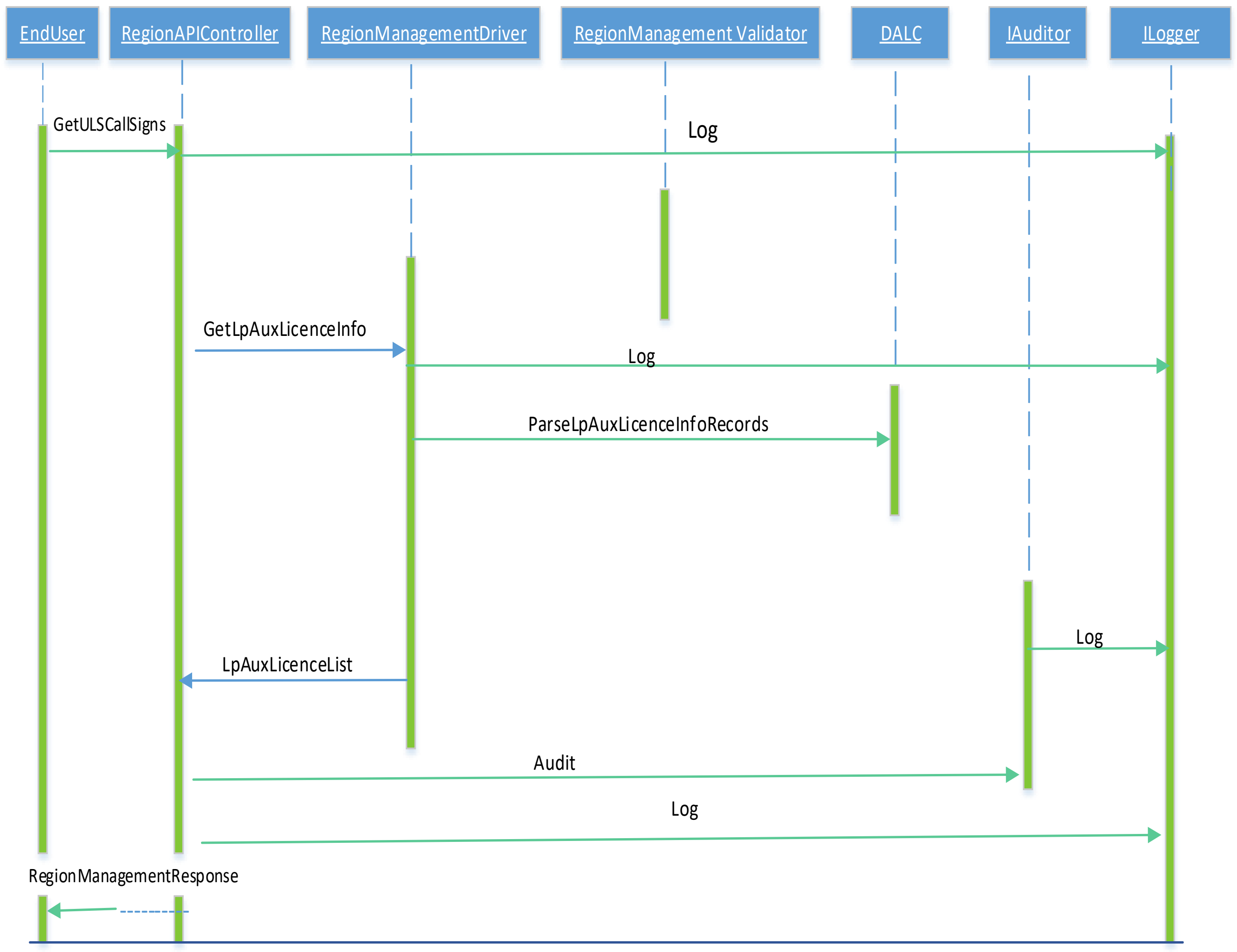
ExcludeIds


End User
RegionAPIController
RegionManagementDriver
Region Management Validator
IDalcIncumbent
IAuditor
ILogger
ExcludeIds
Log
ExcludeIds
Log
ValidateExcludeIds
ExcludeId
Log
RegionManagementResponse
Audit
Log
RegionManagementResponse

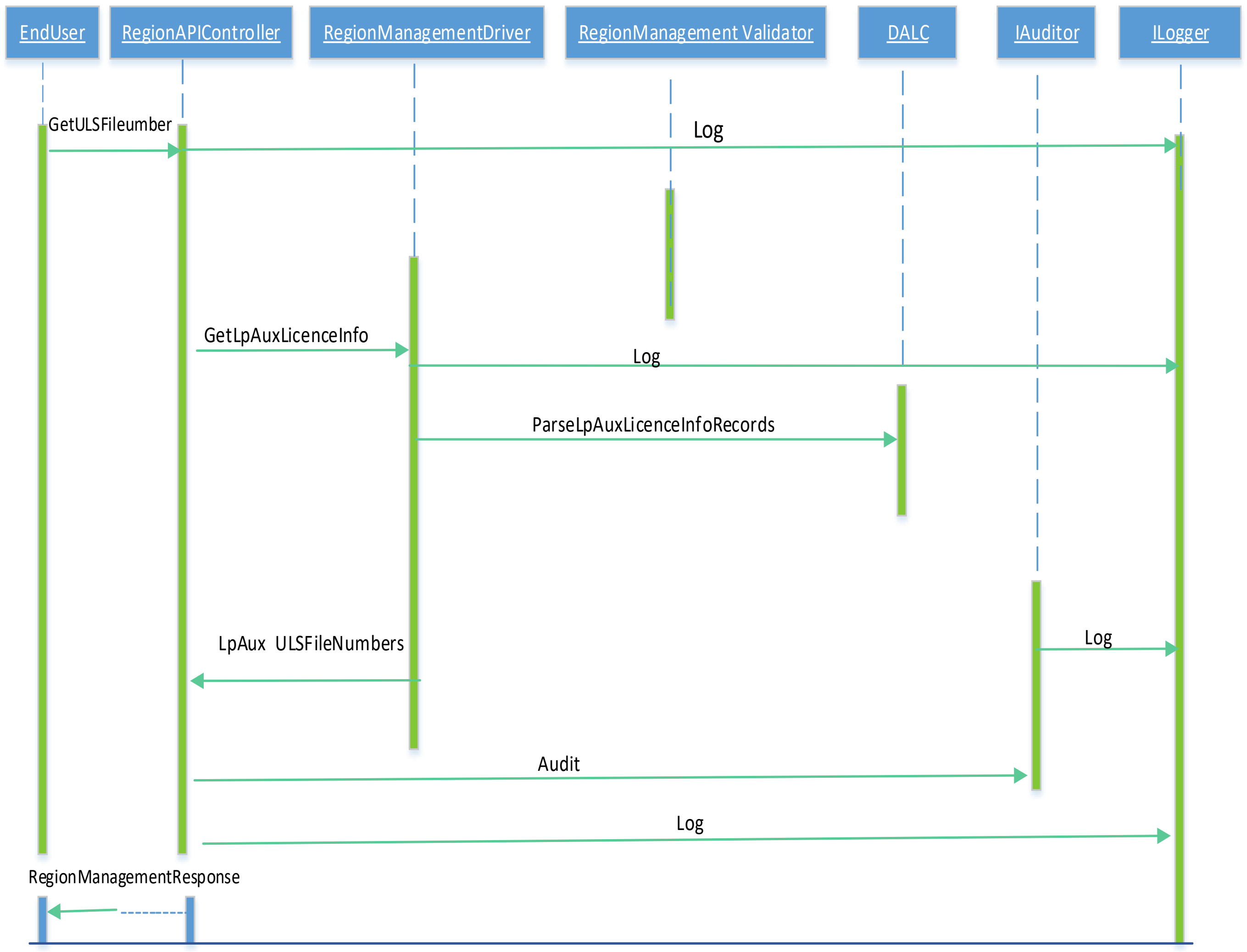
GetIncumbents



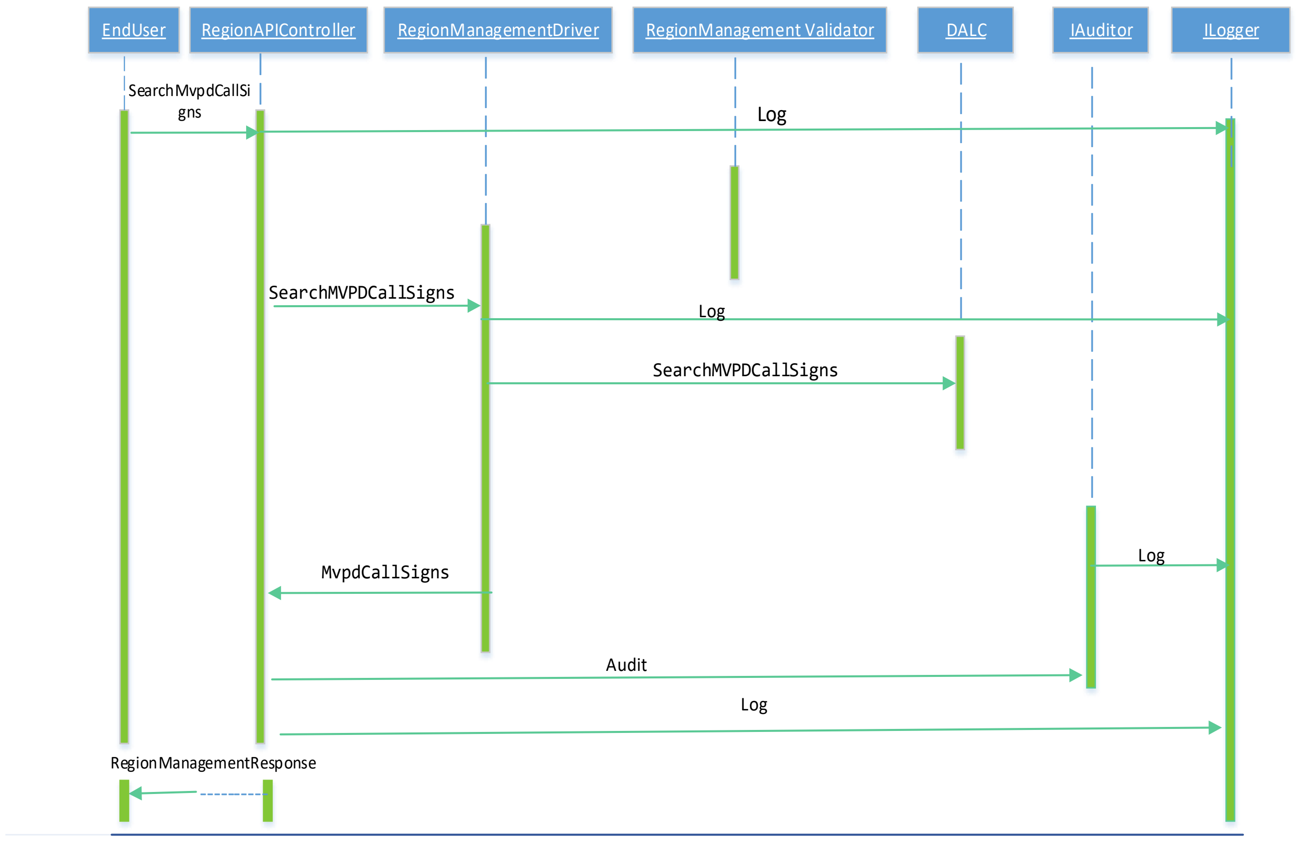
GetULSCallSign



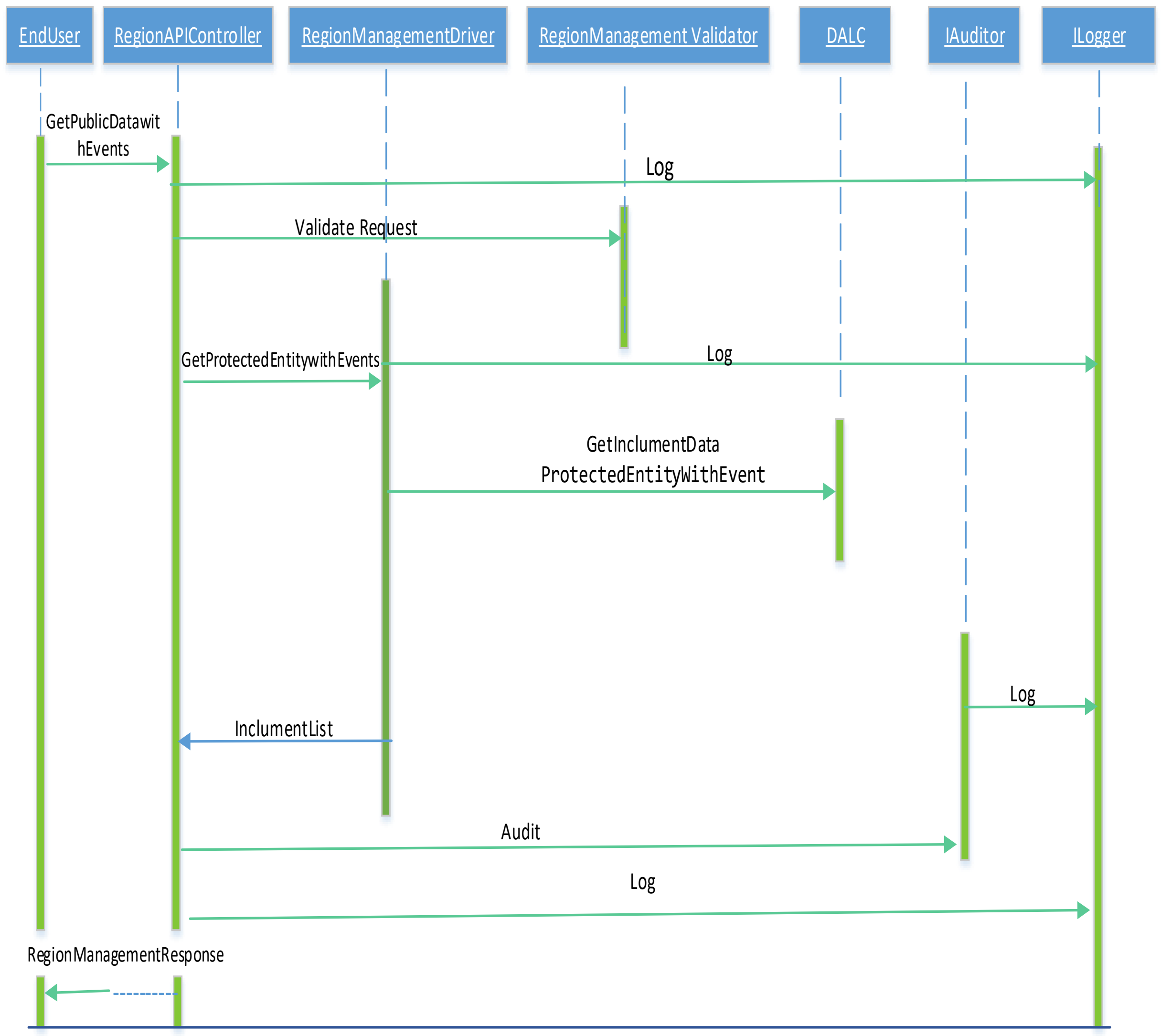
GetULSFileNumbers



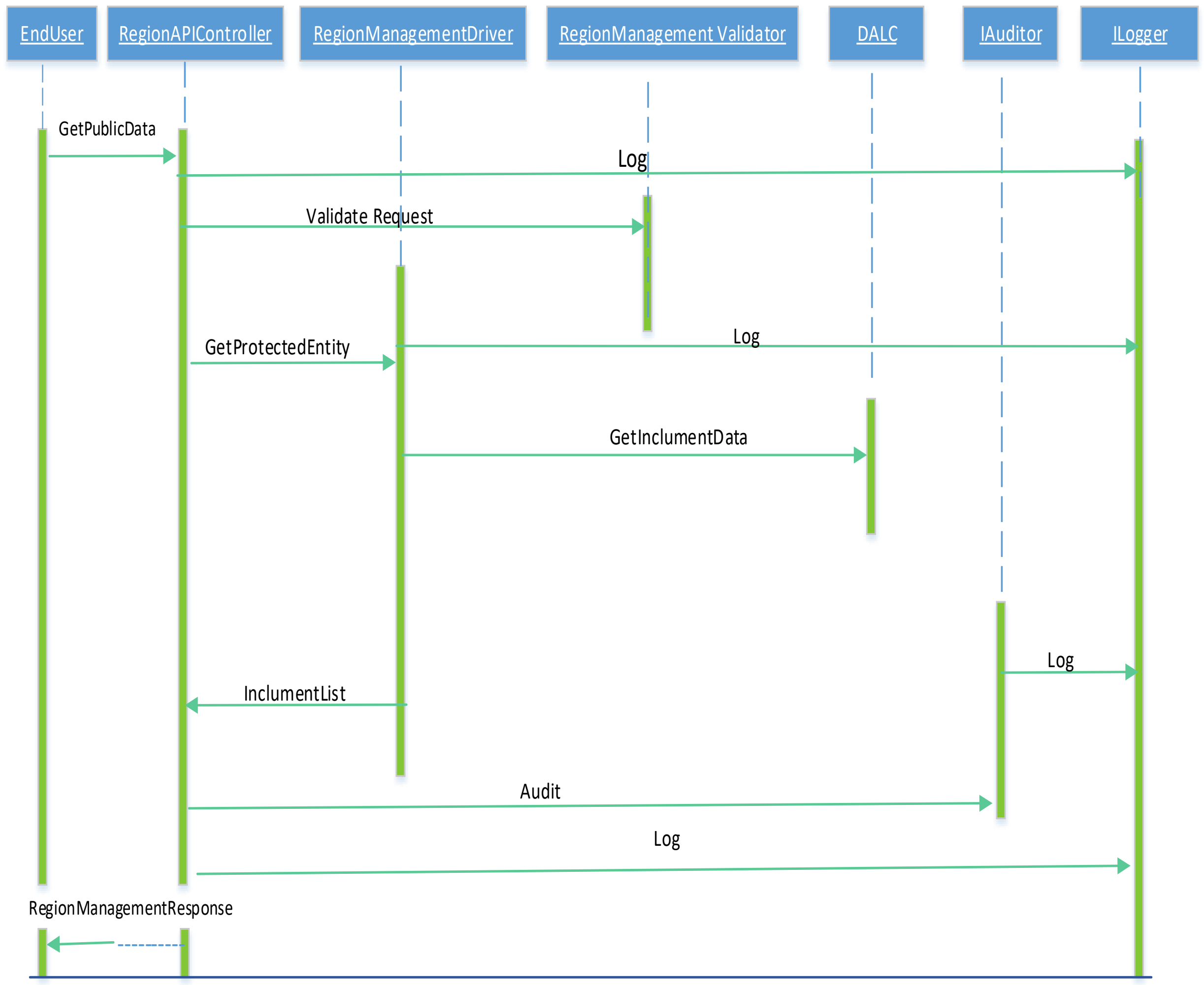
SearchMvpdCallSigns



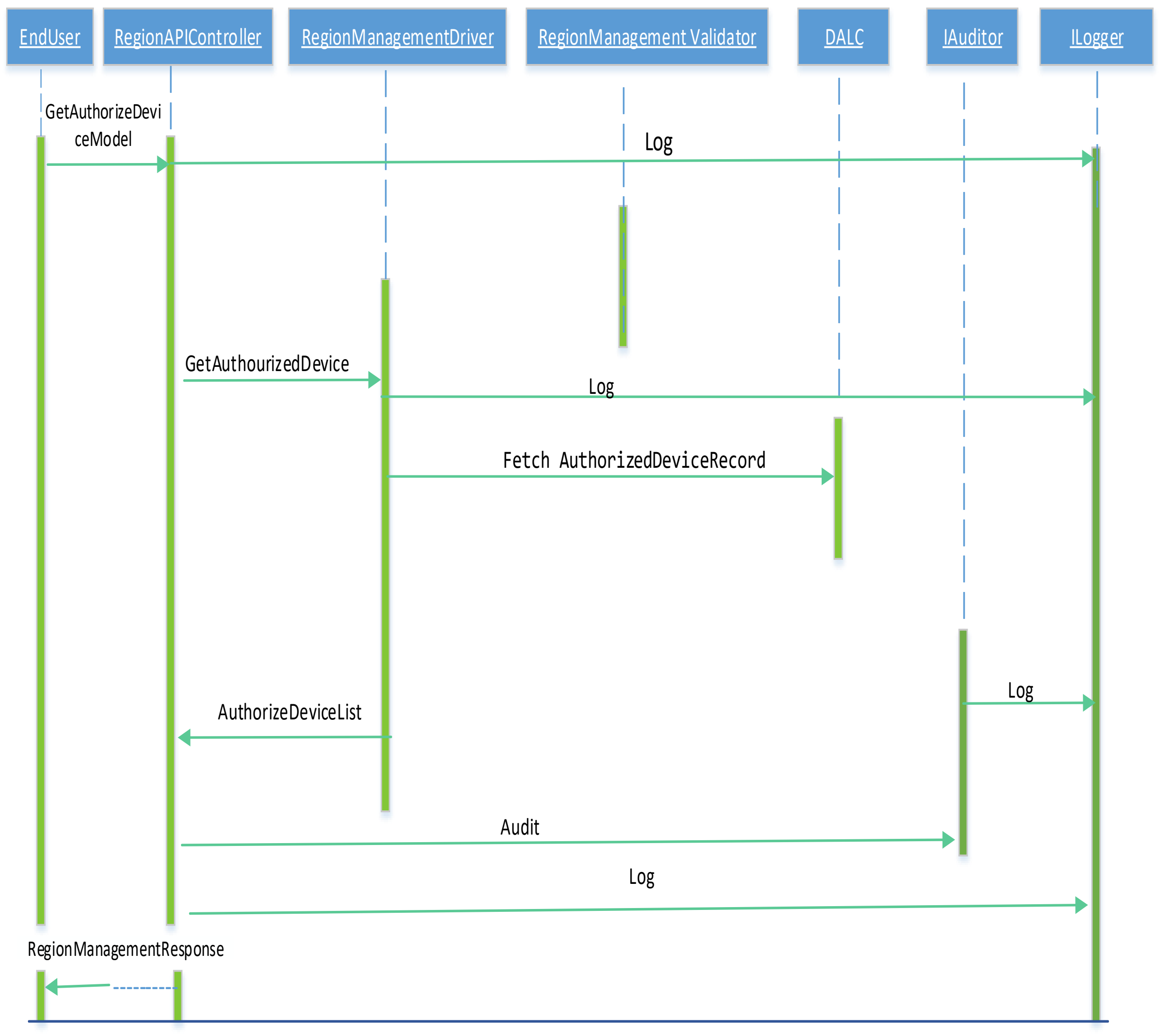
GetPublicDataWithEvents



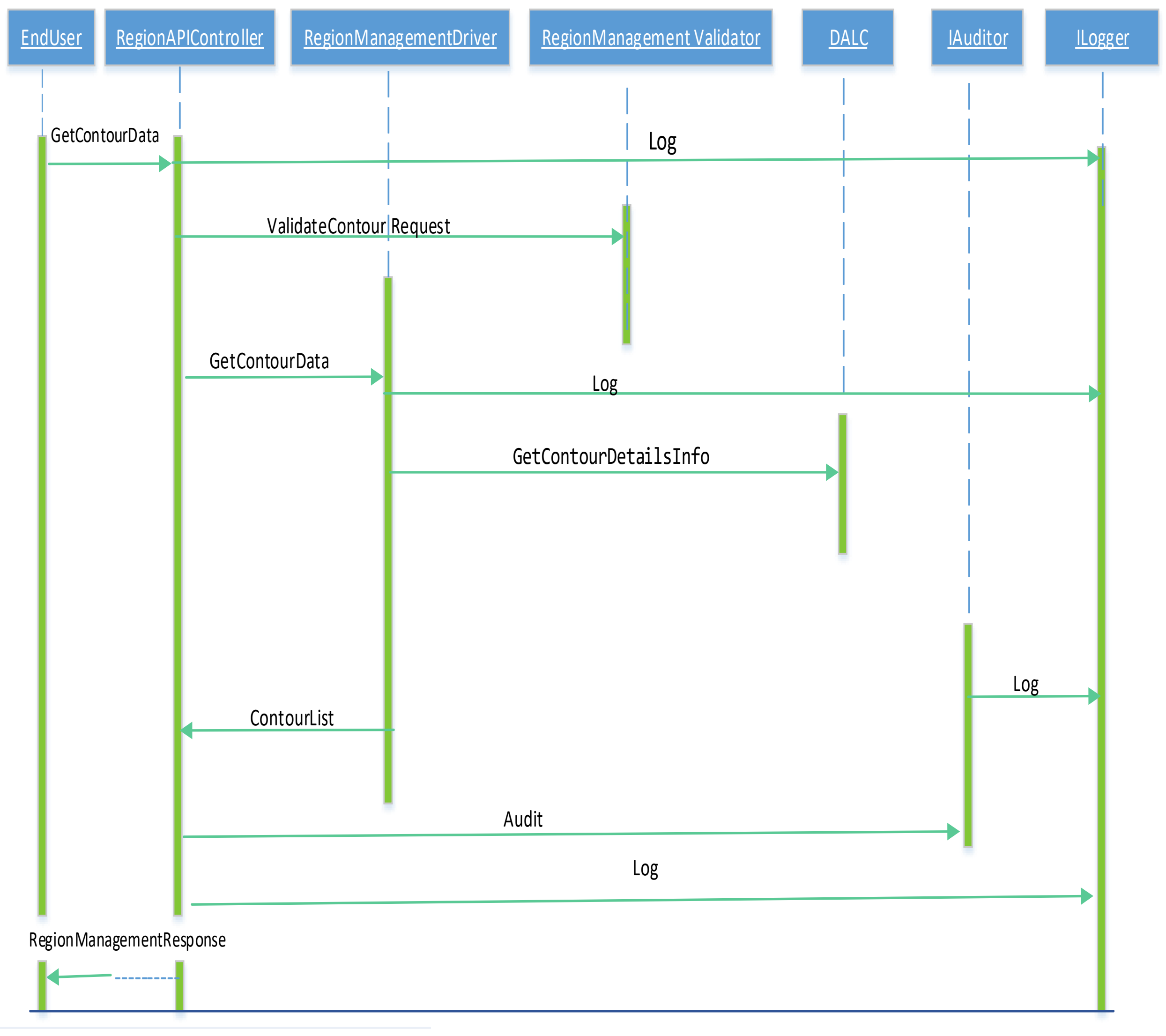
GetPublicData


EndUser
RegionAPIController
RegionManagementDriver
RegionManagement Validator
DALC
IAuditor
ILogger
GetPublicData
Log
Validate Request
GetProtectedEntity
Log
GetInclumentData
Log
InclumentList
Audit
Log
RegionManagementResponse

GetAuthorizedDeviceModels

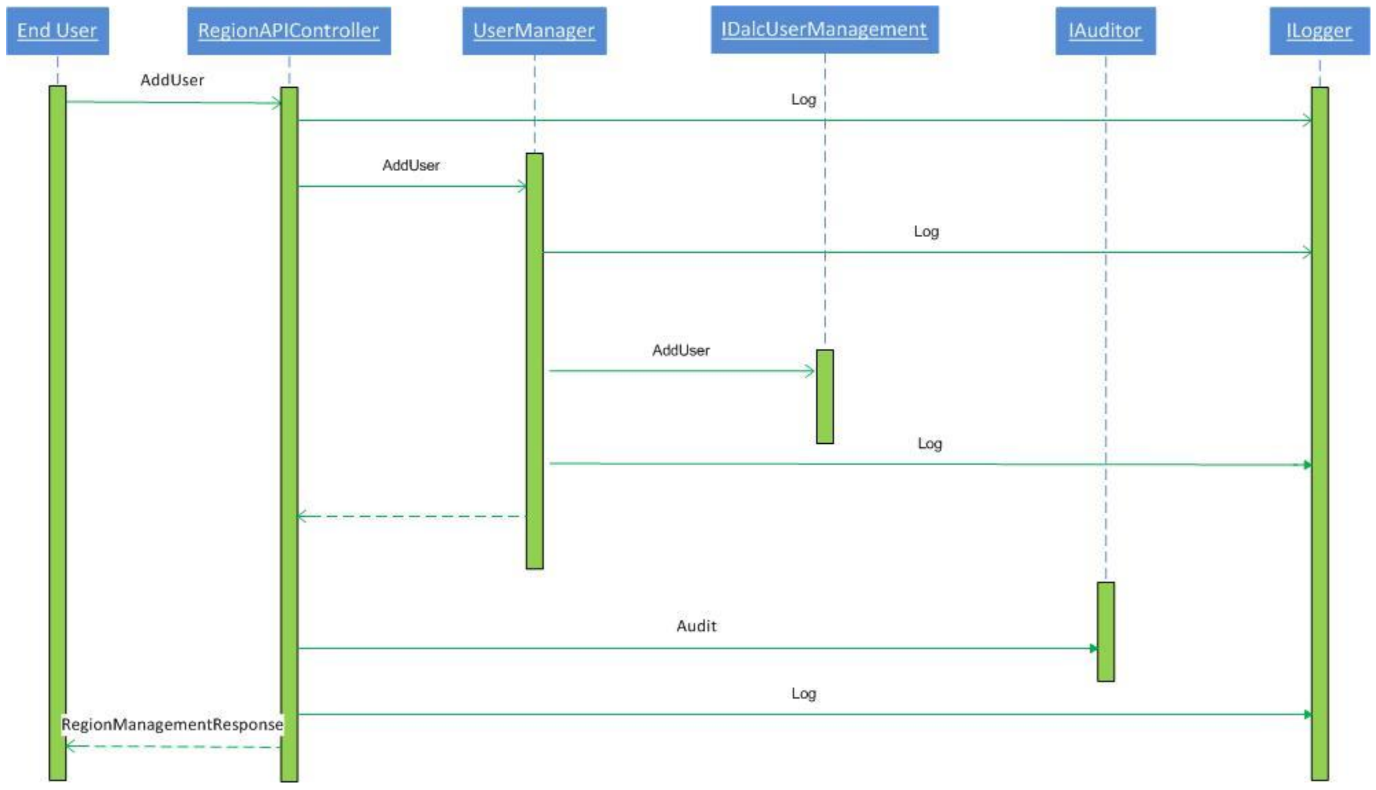


GetContourData



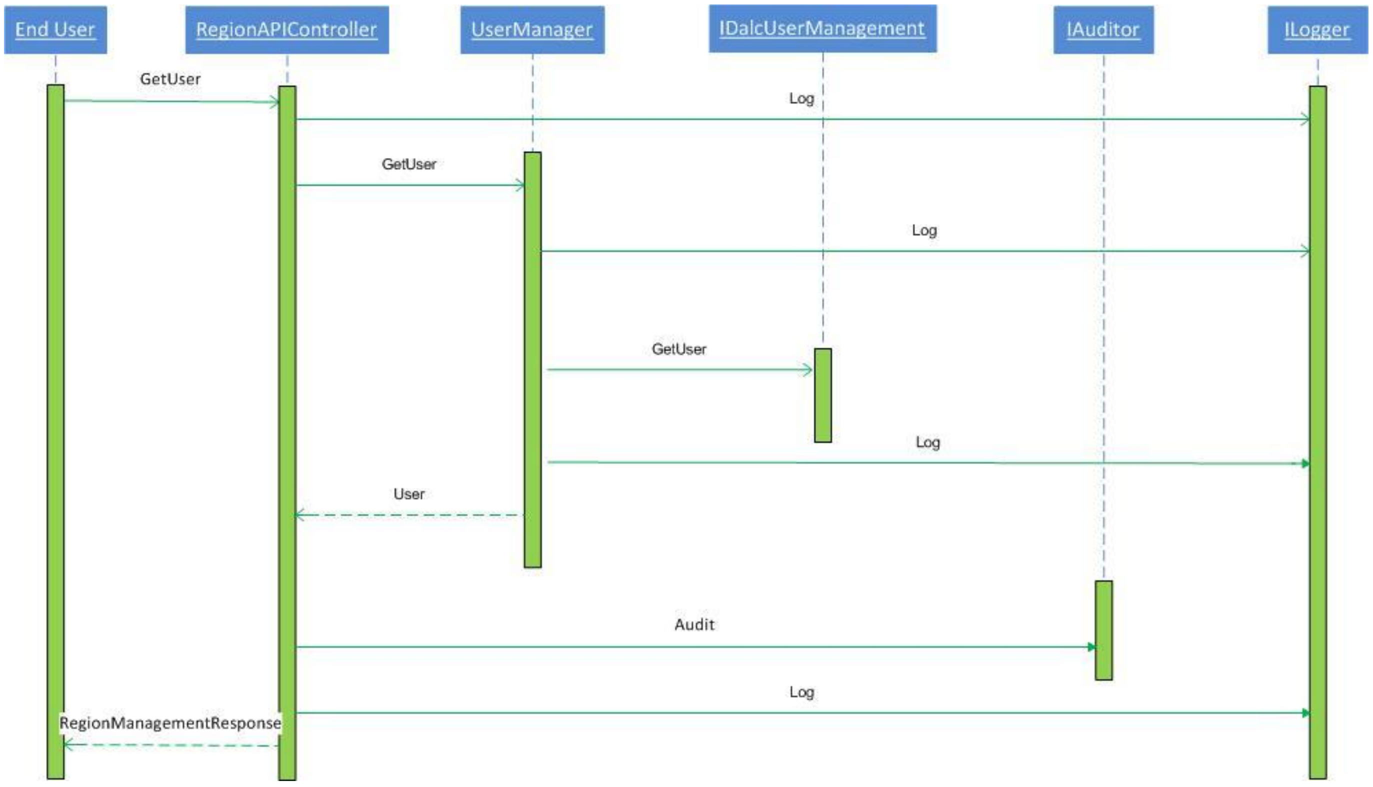
Add User

Note – this method is not used in the current implementation of the application but it is left in the document to provide details in case it is needed later.


End User
RegionAPIController
UserManager
IDalcUserManagement
IAuditor
ILogger
AddUser
Log
AddUser
Log
AddUser
Log
Audit
Log
RegionManagementResponse

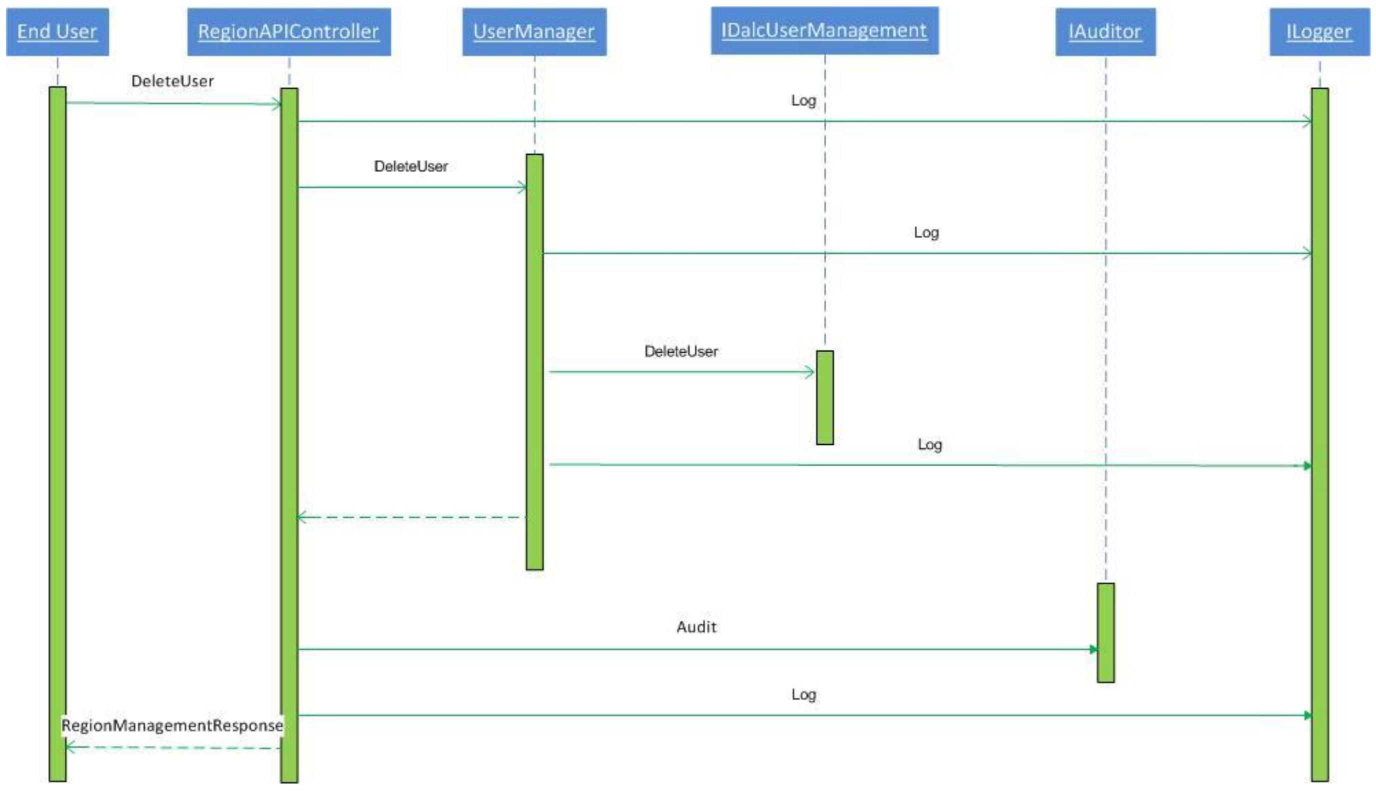
Get User

Note – this method is not used in the current implementation of the application but it is left in the document to provide details in case it is needed later.



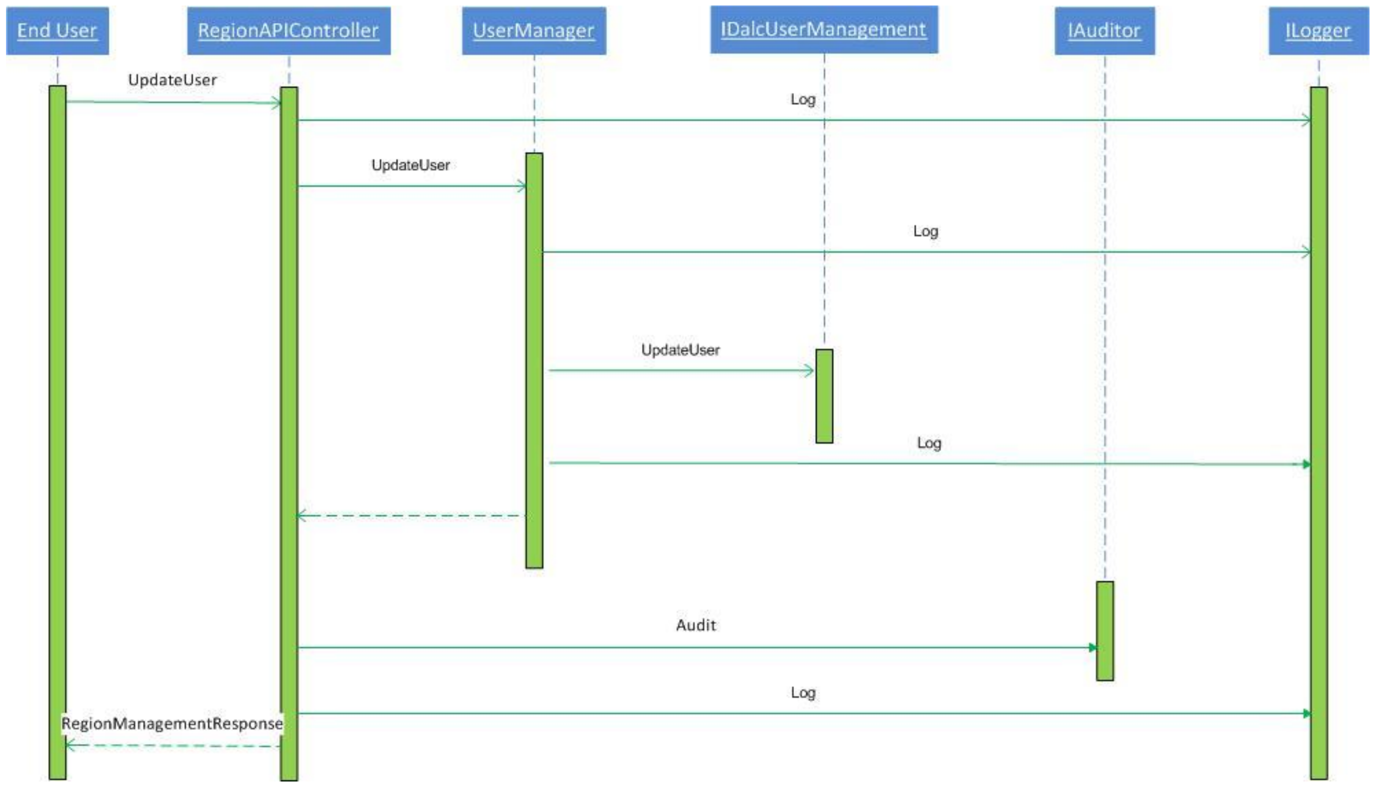
Delete User

Note – this method is not used in the current implementation of the application but it is left in the document to provide details in case it is needed later.



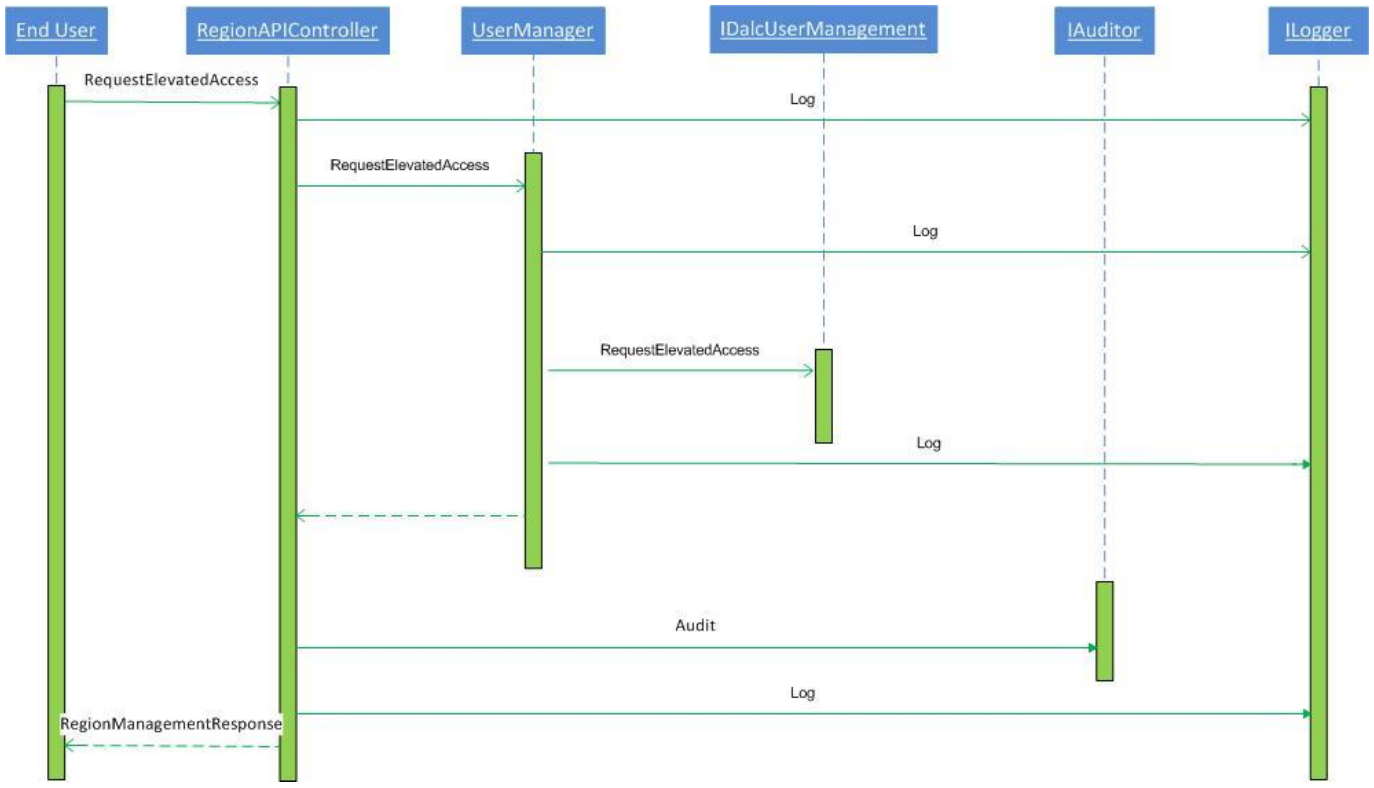
UpdateUser

Note – this method is not used in the current implementation of the application but it is left in the document to provide details in case it is needed later.



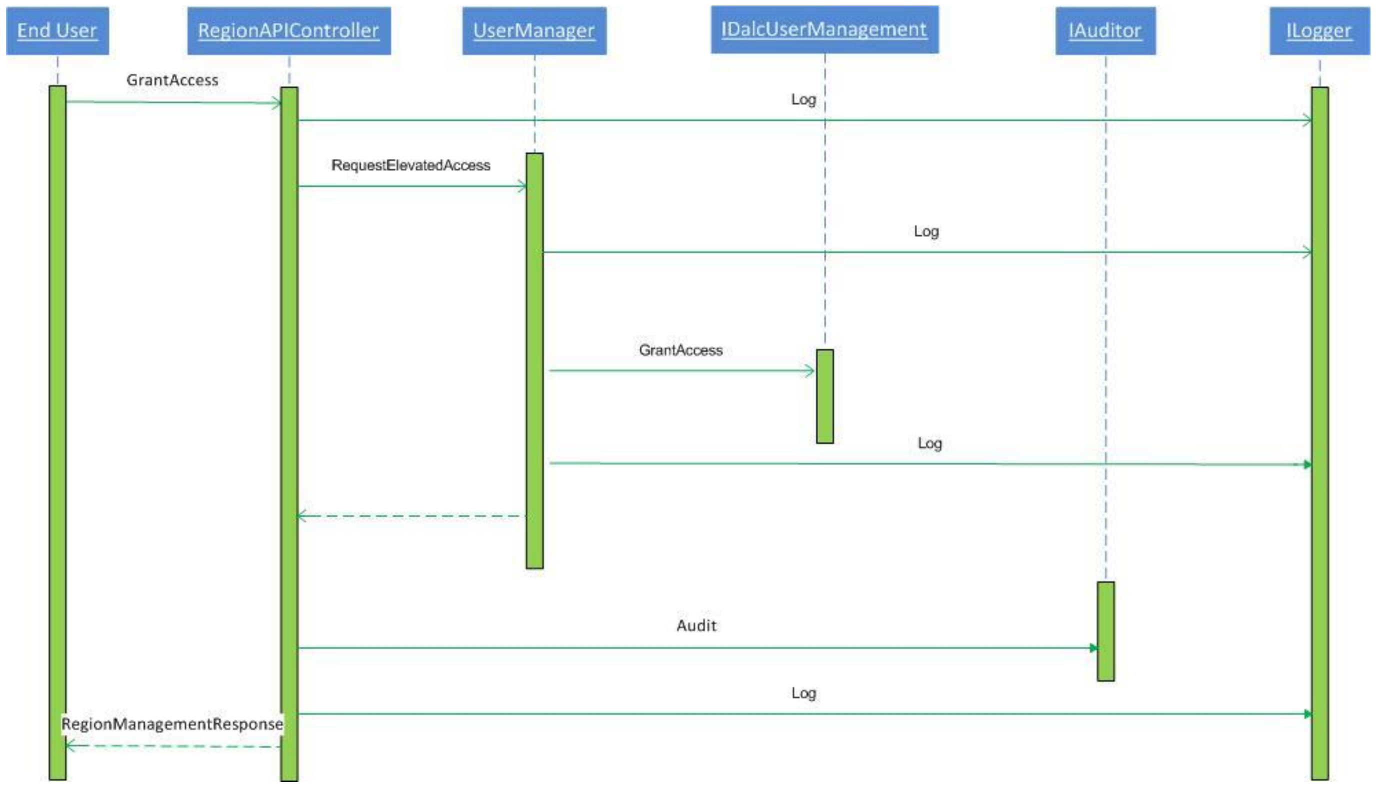
Request Elevated Access

Note – this method is not used in the current implementation of the application but it is left in the document to provide details in case it is needed later.



Grant Access

Note – this method is not used in the current implementation of the application but it is left in the document to provide details in case it is needed later.

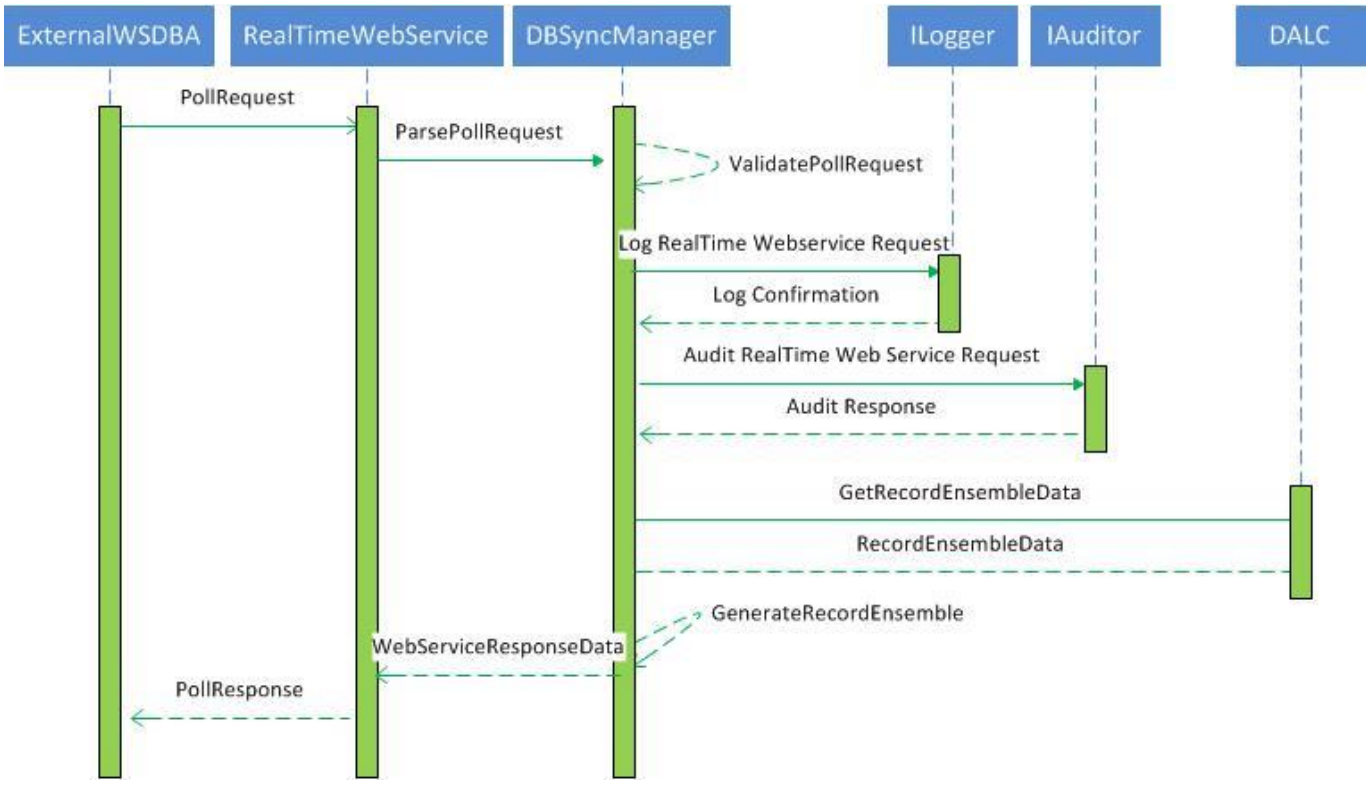


5.16.3 DB Admin Web Service

The DB Admin Web Service will expose one method called: RealTimePollRequest. This method takes one parameter (XML string) that will identify the "poll" request. The XML schema format for the poll request and the XML response is documented in "Section 4 – Real Time Web Service" of the Database-to-Database Synchronization Interoperability Spec.

Assembly Name	**Microsoft.Whitespace.Sync.Database.Service**
Source Directory	src\Whitespace\Sync\Database\Service
Root Namespace	Microsoft.Whitespace.Sync.Database.Service
Dependencies	Microsoft.Whitespace.Entities.dll, Microsoft.Whitespace.Common, Microsoft.Whitespace.Sync.Database.Manager

The following sequence diagram shows the high level view of the DB Admin web service



6. Test

6.1Sample Test Configuration Implementation

Note – this section of the document will be updated when the final set of Unit Test classes for the application has been created.

For unit testing purposes, we do not want to have a config file that is saved to an azure table (instead, it would be ideal to have everything stored locally within the test harness). The following is an outline of a sample test configuration class that can be used where config values are retrieved from the application's web.config or app.config file. Also, any updated config values that are made during the exectuion are temporarily stored in a private member variable dictionary called "overrideValues". Below is an general outline of the TestConfiguration:

```
///<summary>
/// A Test Configuration class that uses the app.config or web.config to store all config values.
///</summary>
publicclassTestConfiguration : IConfiguration
{
privateconststring CurrentRegionIdFieldName = "RegionId";
privateconststring UnityConfigFieldName = "UnityConfig";

///<summary>
///The overrideValues dictionary is used to temporarily save updated config
/// values since
///</summary>
privateDictionary<string, string> overrideValues = newDictionary<string, string>();

publicint CurrentRegionId
   {
get
      {
returnint.Parse(ConfigurationManager.AppSettings[CurrentRegionIdFieldName]);
      }
   }

publicstringthis[string name]
   {
get
      {
if (overrideValues.Keys.Contains(name))
         {
return overrideValues[name];
         }
else
         {
return System.Configuration.ConfigurationManager.AppSettings[name];
         }
      }

set
      {
overrideValues[name] = value;
      }
   }

publicIUnityContainerCreateUnityContainer()
   {
IUnityContainer container = newUnityContainer();

// Initialize the container via the XmlUnityConfigurationSection by reading the xml config values.
XmlReader reader = XmlReader.Create(new System.IO.StringReader(this[UnityConfigFieldName]));
XmlUnityConfigurationSection streamConfig = newXmlUnityConfigurationSection();
streamConfig.DeserializeFromXml(reader);
streamConfig.Configure(container);

return container;
   }
}
```

7. Application Tables

7.1Organization of Tables Within the Application

The whitespace application includes two major categories of tables, as follows:

- Region-specific tables that contain data specific to a given region's deployment of the whitespace application
- Application-wide tables that contain data that is used throughout the application and is not specific to a given region.

The table names for region-specific tables contain a base table name, prefixed by a region identifier. The table below shows sample table names for both types of tables.

Table	Type	Description
Audit	Application Wide	Table used for recording audit information throughout the application
RGN1ConfigSettings	Region-specific	Contains configuration settings for the application as deployed for RGN1 (FCC)
RGN5ConfigSettings	Region-specific	Contains configuration settings for the application as deployed for RGN5 (Ofcom)

7.2List of Tables Used

The table below provides a list of all of the Azure tables that are used within the Whitespace database application.

Table Name/Base Table Name	Type	Sample Region-Specific Name	Description	Applies to Regions
Audit	Application Wide		Table used for recording audit information throughout the application	All
BroadcastAuxiliaryStations	Region-Specific	RGN1BroadcastAuxiliaryStations	Contains list of broadcast auxiliary stations requiring protection. Pouplated in FCC region sync process	FCC Only

CDBSAntennaPattern	Region-Specific	RGN1CDBSAntennaPattern	Antenna pattern data from FCC CDBS database	FCC Only
CDBSCanadaTvEngDataTable	Region-Specific	RGN1CDBSCanadaTvEngDataTable	TV Engineering data for Canada	FCC Only
CDBSTranslatorData	Region-Specific	RGN1CDBSTranslatorData	List of TV translators requiring keyhole protection on their receive channel	FCC Only
CDBSUSMexicoTVEngData	Region-Specific	RGN1CDBSUSMexicoTVEngData	US and Mexico TV stations requiring protection	FCC Only
ConfigSettings	Region-Specific	RGN1ConfigSettings	Configuration settings for the deployment	All
DataCacheUpdateStatus	Region-Specific	RGN1DataCacheUpdateStatus	Contains information regarding status of data that is cached in the application	All
DBAdminInfo	Region-Specific	RGN1DBAdminInfo	List of other whitespace DBAs the application communicates with in DBSync process	FCC Only
ExcludedChannels	Region-Specific	RGN1ExcludedChannels	Channels for which whitespace access is excluded in a given region	All
ExcludedIds	Region-Specific	RGN1ExcludedIds	Devices for which whitespace usage is excluded	All
FixedTVBDRegistration	Region-Specific	RGN1FixedTVBDRegistration	Registration data for fixed TV band devices	
InitializedDeviceDetails	Region-Specific	RGN1InitializedDeviceDetails	Contains information about TV band devices that have submitted device initialization requests through PAWS	All
LpAuxRegistration	Region-Specific	RGN1LpAuxRegistration	Registration data for lp aux devices	FCC Only

LPAuxRegistrationDetails	Region-Specific	RGN1LPAuxRegistrationDetails	Contains details regarding lp aux device registration such as start/stop time of events etc. Child table of LPAuxRegistration table	FCC Only
MVPDRegistration	Region-Specific	RGN1MVPDRegistration	Registration data for MVPD sites	FCC Only
MVPDWaiverCallSigns	Region-Specific	RGN1MVPDWaiverCallSigns	List of MVPD call signs that have received a waiver that allows them to be protected beyond the 80KM limit for keyhole protection. Maintained manually.	FCC Only
OffshoreRadioService	Region-Specific	RGN1OffshoreRadioService	List of locations where offshore radio service is protected	FCC Only
PMSESyncStatus	Region-Specific	RGN1PMSESyncStatus	Contains control information for Ofcom PMSE synchronization process	Ofcom only
RadioAstronomyData	Region-Specific	RGN1RadioAstronomyData	List of radio astronomy sites requiring protection	FCC Only
RegionSyncStatus	Region-Specific	RGN1RegionSyncStatus	Contains control information for FCC region sync processes	FCC Only
RuleSetInformation	Region-Specific	RGN1RuleSetInformation	Contains validation rules for FCC whitespace devices	FCC Only
TBandProtection	Region-Specific	RGN1TBandProtection	List of Tband sites requiring protection	FCC Only
TempBASRegistration	Region-Specific	RGN1TempBASRegistration	Registration data for TempBAS sites	FCC Only
TransactionIDIssued	Region-Specific	RGN1TransactionIDIssued	List of transaction ids generated in DB sync process	FCC Only

TranslatorWaiverCallSigns	Region-Specific	RGN1TranslatorWaiverCallSigns	Contains a list of call signs for translators that are beyond the 80KM limit from their parent call sign but still require protection because they have received a waiver	FCC Only
TVReceiveSiteRegistration	Region-Specific	RGN1TVReceiveSiteRegistration	TV receive site registration data. The FCC specification describes the registration of this entity type and exchange in DBSync process but it is not currently used in the FCC process	FCC Only
ULSLicensedLPAuxData	Region-Specific	RGN1ULSLicensedLPAuxData	Licensed LP aux registration data downloaded from FCC ULS database	FCC Only
ULSPLCMRSData	Region-Specific	RGN1ULSPLCMRSData	Private Land Mobile/Commercial Mobile registration data downloaded from FCC ULS database	FCC Only
ULSUnLicensedLPAuxData	Region-Specific	RGN1ULSUnLicensedLPAuxData	Unlicensed LP aux registration data downloaded from FCC ULS database	FCC Only
DttSyncStatus	Region-Specific	RGN5DttSyncStatus	Contains control information for Ofcom availability data syncronization process	Ofcom only
p1h0150	Region-Specific	RGN5p1h0150	Ofcom DTT availability data. See details below	Ofcom only
p1h0150v01112013	Region-Specific	RGN5p1h0150v01112013	Ofcom DTT availability data. See details below	Ofcom only

PMSEAssignments	Region-Specific	RGN5PMSEAssignments	List of PMSE devices requiring protection for Ofcom	Ofcom only
UnscheduledAdjustments	Region-Specific	RGN5UnscheduledAdjustments	Data received from Ofcom regarding unscheduled adjustments to availability data	Ofcom only
WADLogsTable	Application-wide		Trace Log	All

7.3Ofcom DTT Availability Data

Ofcom provides availability data regarding the maximum EIRP values that are permitted for white space devices in a given channel and location. The data is stored in multiple tables separated using the following attributes:

Attribute	Description	Table Name Parameter Value
Emissions Class	Identifies the White Space Device emissions class for this set of availability data	p
Height	Height in centimeters above ground for this set of availability data	h

The tables contain the maximum EIRP values for each channel, from 21 through 60, by Easting/Northing. For each emissions class, there is also a default values table that is used when the white space device does not report its height. The table below provides some example table names illustrating the organization of the data.

Table Name	Contents
RGN5p1h0150	Availability data for devices with emission class 1, height up to 150 centimeters
RGN5p1h0500	Availability data for devices with emission class 1, height between 151 and 500 centimeters
RGN5p1hDDDD	Availability data for devices with emission class 1, default values when height is not reported