|  |  |
| --- | --- |
| * Establish data catalog supporting the repository. The catalog should support OGC/CSW | * Elastic Search will be the database, same as Esri geoportal-server-catalog. * The repository will be based upon Elastic Search Python bindings. No Java programming is needed, just some Python scripts (Python is required for geometric processing). * Standard defaults to FGDC, as used by datasets on HKMS 2.0 (see below).      * Metadata profile for each file type. * For 3D models, one metadata file for each model instead of the entire archive. Otherwise, it makes little sense. * Some metadata are auto-completed by the system depending on the file type, such as file format name, file size, extent, centroid, ‘place’, etc. These fields will be read-only. * Versioning |
| * Develop a Web application for LandsD users to retrieve, browse and download available records and map services from the data catalog. | * XSLT files transforms XML to readable HTML. (from Esri geoportal-server-catalog) |
| * Create a OGC compliant CSW |  |
| * Upload Data to the catalog |  |
| * Convert uploaded data into map service |  |
| * Search uploaded data from catalog |  |
| * Preview available map services in web application | Thumbnails for JPG, PNG, BMP, PDF etc.  Thumbnail if size of the mesh is too large (below 20 tiles combined, file size > 1 GB) |
| * Download Data | ✓ |
| * Interfacing with CSDI portal |  |
| * Data include but not limited to OSGB, OBJ, CityGML, GeoTIF, Tif, MMS, Web services, FileGDB, Shapehile, PDF, IFC etc. | * Helsinki CityGML mesh: <http://3d.hel.ninja/data/citygml/> |
| * Capable for geo-referencing the footprint | ✓  Server (Python script) converts a supported 3D object into a 2D shapefile.  For instance, this highway model:    The orange polygon is its footprint: |
| * Data Interoperability extension for server (Convert Upload/ Download 3D Spatial Data) | Conversion from 3ds, dae, fbx, glb, gltf, ifc, obj, osgb, skp, wrl, x3d formats to dae, fbx, glb, obj, osgb formats. |