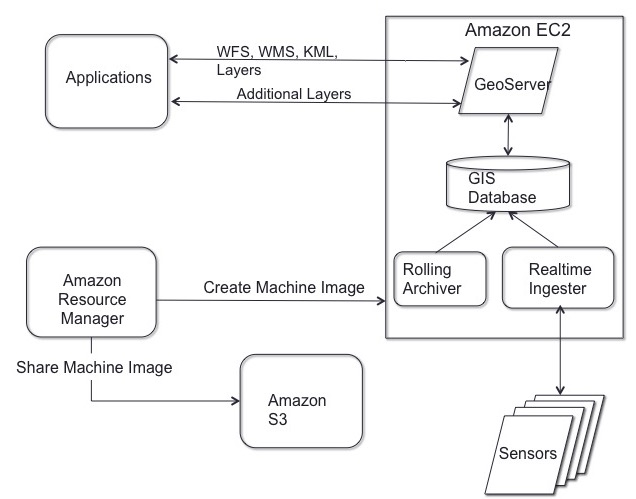
**CHORDS Implementation - UAH**

The overall architecture of CHORDS is shown in Figure 1. We have an instance of the implementation based on this architecture running on the CHORDS server. That instance is currently supporting two different data sources as described below.



**Figure 1. Current Architecture of CHORDS implementation**

**1. UM Hydrology Sensor Data**

**1.1 Extraction**

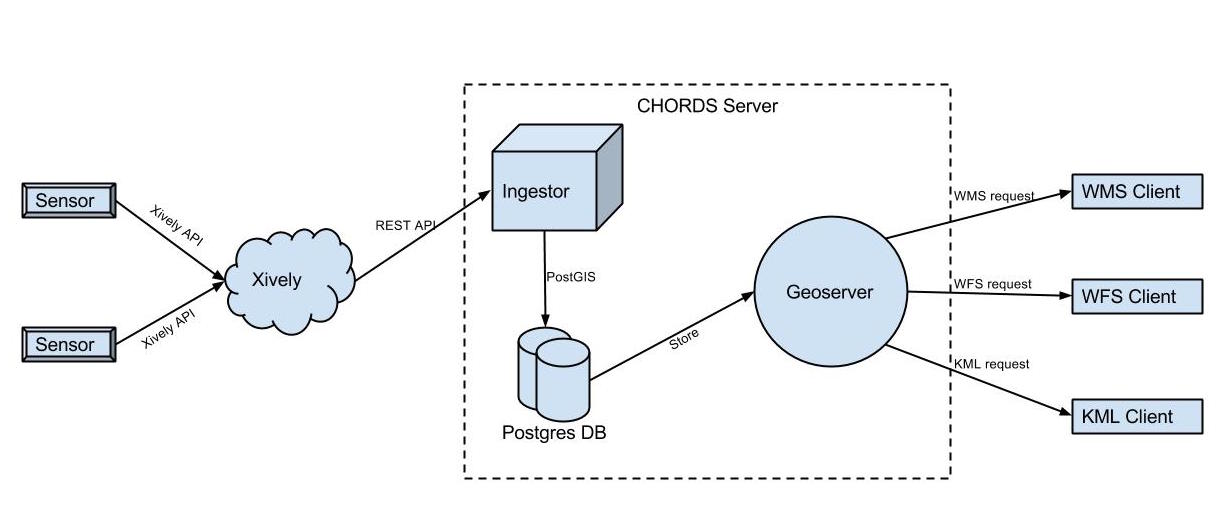
The data generated by sensors located at different place are pushed to Xively Servers over the internet. Xively is Platform As A Service (PAAS) for Internet of Things (IOT). Our ingestor program written in Java makes REST request to Xively to pull the data from its server. Our program uses Apache HTTP Client for Java to make REST request. Eventually, sensors will push the data to CHORDS server directly.

**1.2 Processing**

The pulled geolocated data is then processed, formatted and pushed to PostgresDB with PostGIS extensions on the CHORDS server. The json response received from Xively server is processed using Jackson JSON library for Java.

**1.3 Publishing**

GeoServer generates data store for geo-located data present in PostgresDB. This store is then published which can then be accessed through WFS client like GeoExplorer (We have an instance of GeoExplorer running on CHORDS server). GeoServer allows us to access the published data in many OGC webservices including WMS, WFS, and KML.



**Figure 2. UM Hydrology Sensor Data Publishing Workflow**

**2. CSU CHILL Radar Data**

**2.1 Extraction**

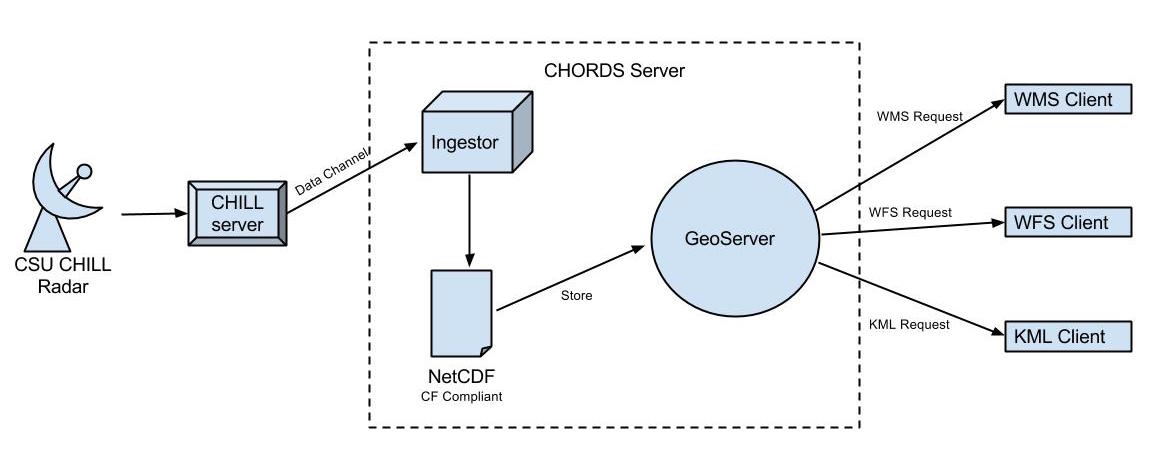
The radar data generated by CSU-CHILL radar is served in real-time via its server (vchill.chill.colostate.edu) at port 2511. Our ingestor program written in Java establishes a data channel with the server and receives real-time data from the server.

**2.2 Processing**

This real-time data is then processed, formatted and saved as CF compliant NetCDF file in the CHORDS server. To generate the NetCDF file we are using ucar NetCDF library for Java.

**2.3 Publishing**

GeoServer generates data store from the saved NetCDF to provide OGC standard services.



**Figure 3. CSU CHILL Radar Data Publishing Workflow**

**\*\*\*We are also working on an OpenLayer-based mapping interface to visualize CHORDS data.**

**Cloud and Software Configurations**

Amazon EC2 Instance Type: m1.large

Operating System: Red Hat

Public IP: 50.0.129.182

Apache Tomcat running on port 8080 to support GeoServer and GeoExplorer

Apache Web Server running on port 80 to support OpenLayer-based visualization application

Port 5432 is open to support PostGreSQL database