# IT Systems Stack

## PoCRA-IITB

This document describes the basic IT stack developed and used by the IITB PocRA for various purposes. It is divided into Level 0 which are the basic databases with tools of access, Level 1, related to models and budgets and the construction of smart databases, i.e., those with spatio-temporal integrity. Level 2 are the higher level tools which are useful at the user-level. Finally, we have a list of Tools which have proved useful.

## Summary

| Level | Name | Description | Remarks |
|---|---|---|---|
| Level 0 | Raw Weather | Raw skymet data | |
| | Smooth Weather | Smoothed for missing items.<br>Allows various access api's including ET0 | |
| | GIS-MRSAC | Geo-morphology shape files. Tools: Zones. Stream Proximity. Vulnerability. Stream Order. | |
| | ADMIN-MRSAC | Administrative shape files | |
| | PoCRADB | Basic PoCRA DB | |
| | PoCRAVillage | Basic Village Database | |
| Level 1 | Point Model Daily | Old daily model. Various Look-ups. | App version available |
| | Point Model Hourly | New Hourly. Uses ET0 | |
| | Village Water Budget | Uses daily point model and PoCRAVillage | Plug-in and batch. |
| | Water Budget Hourly | Hourly model. In development. | Plug-in available |
| | AWS Smart Data | Includes analytic variables. Updated daily. | |
| | Raster Smart Data | Model data at raster. Updated daily. Uses daily model. | |
| | FFS Smart Data | Current status of FFS plots. | |
| | Village Smart | Village level Water-Balance data. | |

| | | | |
|---|---|---|---|
| | Data | | |
| Level 2 | MLP | Uses WB and village data on assets.<br>Uses Zones. | |
| | Dashboard | Uses Smart Raster, Weather, FFS. Uses PoCRADB, Admin. Uses PMU data on permissions. Allows smart views and data download for fixed attributes. | |
| | Contingency | Allows specification and viewing of contingencies. | |
| Tools | Charts | Uses MLP data | |
| | Soil Survey | Used by NBSSLUP and PMU | |

## PMU-IT Team addition

1. **GIS-Web-Enterprise Senior Architect.**
   - Overall ownership of Contingency, MLP and PMU Processes and Dashboard
   - Liaison with PMU-IT processes team and PMU
   - 7-10 years experience in database development, field data acquisition systems, web publishing and GIS
2. **GIS + Database Web Designer/Developer (1+1 positions)**
   - GIS Dashboard front-end/back-end designer, GIS based web services, GIS enabled databases
   - Contingency databases and process, geography databases of PMU such as FFS or DBT
   - 3-5 years experience in enterprise level GIS and Web publishing
3. **GIS-Geomorphology Programmer/Developer (2 positions)**
   - Ownership of water-budget, MLP and other services, development of new tools guided by expert
   - maintenance of spatial and temporal integrity of contingency related GIS-databases
   - 3-5 years experience in GIS and modeling, watershed programs and agro-met services

See: IT Stack for description of current PoCRA-IITB IT Stack (Working Document)

### Level 0 Basic Services

#### Weather

This includes basic data sets prepared by organizing skymet downloads in a suitable form. The “smooth” versions are those where missing station values are papered over by choosing

nearest functioning AWS. The datasets may be accessed by suitable api's used by several other programs.

| Name | Description | Remarks | Code | Status 1 VM1 | Status 2 IITB |
|---|---|---|---|---|---|
| **Weather** | | | W | | |
| **Databases** | | | | | |
| **Raw Weather** | Buffered data fetched from Skymet. | Circle-wise various parameters | RW | | |
| **Methods** | | | | | |
| Update | Periodically fetches from skymet and updates | Executed Daily | RWM1 | | |
| **Needs** | | | | | |
| skymet_api | Api to access data from skymet | https://www.skymetweather.com/media/apiawscurrentdata-mahavedh.php?date=2018-10-18 | skymet_api | OK | |
| **APIs** | | | | | |
| api_rwget1 | Allows fetching of raw data hourly | | RWA1 | | |
| api_rwget2 | Allows fetching of raw data daily | | RWA2 | | |
| **Used By** | | | | | |
| **Smooth Weather** | Buffered and smoothened Skymet<br><br>Master AWS-Village table - yearly<br>Master AWS FFS table - yearly<br>Smooth weather database | All parameters without gaps | SW | Over VM2 in update.py it can be found as a function load_data_in | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | _smoothed_skymet_data_table(), | |
| **Methods** | | | | | |
| Update | Works on RW and fills gaps | Executed daily | SWM1 | | |
| **Needs** | | | | | |
| RW | RW and its methods are inputs | | | | |
| **APIs** | | | | | |
| api_swget1 | Allows fetching of raw data hourly | | SWA1 | | |
| api_swget2 | Allows fetching of raw data daily | | SWA2 | | |
| api_ET0 | Computes ET0 | | ET0 | | |
| **Used By** | | | | | |
| | | | | | |

**MRSAC**

This was the basic datasets obtained from MRSAC for project area. The geo-morphology includes soil data, slope, LU and LC. The DEM, watershed and drainage contain stream level data.Computation of zones is a tool developed for this data set. ADMIN and PoCRADB cover the administrative and basic PoCRA data sets.

| Name | Description | Remarks | Code | Status 1<br>VM1 | Status 2<br>IITB |
|---|---|---|---|---|---|
| **MRSAC** | | | W | | |
| **Databases** | | | | | |
| **GIS** | Various layers obtained from MRSAC | Soils, DEM, Drainage,WS | MRSAC | | |

| | | | | | |
|---|---|---|---|---|---|
| **Methods** | | | | | |
| Zones | Formation of zones. | Output is available in shapefile format with zone id | Zone | Zones have been created for all the villages. Python script was shared eralier | |
| Stream Proximity | Creates stream proximity regions. | Output is available in Shapefile format with predefined buffers based on stream order | StreamProx | To be checked | |
| Stream Order | Creates and orders stream segments | Output is in shapefile with stream order obtained using strahler order. | StreamOrder | Plugin is available | |
| **Needs** | | | | | |
| | | | | | |
| **APIs** | | | | | |
| api_acces_prop | Allows fetching of properties for a given lat-long | | LatLon1 | | |
| **Used By** | | | | | |
| | | | | | |
| **ADMIN** | Various administrative boundaries and census data | Admin. attributes | ADMIN | | |
| **Methods** | | | | | |
| | | | | | |
| **Needs** | | | | | |
| | | | | | |
| **APIs** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| **Used By** | | | | | |
| | | | | | |
| **PoCRADB** | Various layers obtained from PoCRA | Villages, Clusters others | PoCRADB | | |
| **Methods** | | | | | |
| | | | | | |
| **Needs** | | | | | |
| | | | | | |
| **APIs** | | | | | |
| | | | | | |
| **Used By** | | | | | |
| | | | | | |
| **PoCRAVillage** | MLP data obtained from village | Assets, demographics. | PoCRAVillage | | |
| **Methods** | | | | | |
| | | | | | |
| **Needs** | | | | | |
| | | | | | |
| **APIs** | | | | | |
| | | | | | |
| **Used By** | | | | | |
| | | | | | |

**Level 1**

**Point Models**

| Name | Description | Remarks | Code | Status 1 VM1 | Status 2 IITB |
|---|---|---|---|---|---|
| **Point Models** | | | PM | | |
| **PMD** | The daily point model | Inputs: Point properties and weather, crop | PMD | | |
| **Look Ups** | | | | | |
| Soil, Crop | | | | | |
| **Needs** | | | | | |
| api_swget1 | Smooth weather | | | | |
| api_access_prop | Point properties | | | | |
| **APIs** | | | | | |
| api_update_point_daily | Updates the properties of a point for a given duration assuming initial values as input | | PMDupdate | | |
| **Used By** | | | | | |
| **FarmWaterD** | App version of daily point model | | FarmWaterD | | |
| **PMH** | The hourly point model | Inputs: Point properties and weather, crop | PMH | | |
| **Look Ups** | | | | | |
| Soil, Crop | | | | | |
| **Needs** | | | | | |
| api_swget2 | Smooth weather | | | | |
| api_access_prop | Point properties | | | | |
| ET0 | Hourly computation of ET0 | | | | |
| **APIs** | | | | | |
| api_update_point_hourly | Updates the properties of a point for a given duration | | PMHupdate | | |

| | | | | | |
|---|---|---|---|---|---|
| | assuming initial values as input | | | | |
| **Used By** | | | | | |
| **FarmWaterH** | App version of hourly point model | | FarmWaterH | | |

**Water Balance Tools**

| Name | Description | Remarks | Code | Status 1 VM1 | Status 2 IITB |
|---|---|---|---|---|---|
| **Water Budgets** | | | WB | | |
| **WBD** | The regional water budget. running on a daily point model.<br>Outputs regional stocks and flows. Vulnerability. | Inputs:region, cadastral, and cropping data, seasons | PID | | |
| **Methods** | | | | | |
| Epoch 1 | Creates water budget epochs in space-time for budgeting purposes | Crop Duration WB, regional stocks | Epoch1 | | |
| Sampler 1 | Samples points in regions, calls point model and aggregates. | Grid-based sampler | Sampler1 | | |
| **Needs** | | | | | |
| api_update_point_daily | Daily point model on various locations | | | | |
| MRSAC | Region and cadastral. | | | | |
| **APIs** | | | | | |
| api_regional_WBD | Reports regional WB-daily in standard attributes | | R-WBD | | |
| **Used By** | | | | | |
| **MLP** | Planning Tool | | MLP | | |
| **WBH** | The regional water budget. running on hourly point | Inputs:region, cadastral, and | PIH | | |

| | | | | | |
|---|---|---|---|---|---|
| | model. | cropping data, seasons | | | |
| **Methods** | | | | | |
| Epoch 1 | Creates water budget epochs in space-time for budgeting purposes | Crop Duration WB, regional stocks | Epoch1 | | |
| Sampler 1 | Samples points in regions, calls point model and aggregates. | Grid-based sampler | Sampler1 | | |
| **Needs** | | | | | |
| api_update_point_hourly | Daily point model on various locations | | | | |
| MRSAC | Region and cadastral. | | | | |
| **APIs** | | | | | |
| api_regional_WBH | Reports regional WB-hourly in standard attributes | | R-WBH | | |
| **Used By** | | | | | |
| **MLP** | Planning Tool | | MLP | | |

**Spatio-temporal Smart Databases**

| Name | Description | Remarks | Code | Status 1 VM1 | Status 2 IITB |
|---|---|---|---|---|---|
| **Spatio-Temporal Databases** | | | STDB | | |
| **Databases** | | | | | |
| **Smart AWS** | Various attributes such as dry-spell, cumulative rainfall etc. | Circle-wise various parameters | SmartAWS | | |
| **Methods** | | | | | |
| Update | Temporal logic and auxiliary variables. | Executed Daily | AWS_Update1 | | |
| **Needs** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| api_swget1 | Hourly smooth weather data | | | | |
| | | | | | |
| **APIs** | | | | | |
| | | | | | |
| **Used By** | | | | | |
| Contingency | | | | | |
| Dashboard | | | | | |
| **SmartRaster** | Model Attributes at Raster Level | Point-wise various parameters | SmartRaster | | |
| **Methods** | | | | | |
| Sampler 2 | Creates and maintains grid of points | | Sampler2 | | |
| Update | Temporal logic. | Executed Daily | Raster_Update 1 | | |
| **Needs** | | | | | |
| api_update_point_hourly | Hourly point model update | | | | |
| | | | | | |
| **APIs** | | | | | |
| Raster_Access | Access a specific lat-long data for a specified interval | | | | |
| **Used By** | | | | | |
| Contingency | | | | | |
| Dashboard | | | | | |

**Level 2**

**Micro-Level Planning**

| Name | Description | Remarks | Code | Status 1 VM1 | Status 2 IITB |
|---|---|---|---|---|---|
| **Micro Level Planning** | | | MLP | | |
| **Databases** | | | | | |
| **MLP** | Village Level database of assets, cropping pattern etc., with basic stocks and flows. | Village level planning outputs | MLP-DB | | |
| **Methods** | | | | | |
| Update | Uses regional water balance and other water conservation equations to prepare WB | | | | |
| **Needs** | | | | | |
| api_regional_WBD | Daily WB | | | OK | |
| PoCRAVillage | Village database: cropping pattern, assets | | | | |
| GIS_Zones | Zone decomposition | | | | |
| **APIs** | | | | | |
| | | | | | |
| **Used By** | | | | | |
| | | | | | |

**Dashboard**

| Name | Description | Remarks | Code | Status 1 VM1 | Status 2 IITB |
|---|---|---|---|---|---|
| **Dashboard** | | | MLP | | |
| **Dashboard** | . Allows smart views and data download for fixed attributes | | Dashboard | | |
| **Methods** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Update | Updates from smart databases | | | | |
| DataDownload | Allows data download based on PMU permissions | | | | |
| **Needs** | | | | | |
| SmartAWS | AWS with key contingency indicators | | | | |
| SmartRaster | Raster with model data | | | | |
| SmartVillage | Village level updates | Village level aggregate model outputs<br><br>Village level model update api | | | |
| MLP | Village level assets etc. | | | | |
| PoCRA_Admin, PoCRA_PMU | For permissions etc. | | | | |
| **APIs** | | | | | |
| | | | | | |
| **Used By** | | | | | |
| | | | | | |

| Name | Description | Remarks | Code | Status 1<br>VM1 | Status 2<br>IITB |
|---|---|---|---|---|---|
| **Soil Survey** | | | SS | | |
| **Soil Survey** | Allows collection of farm level data. | Attributes like soil depth, crops, yields etc. | | ok | |
| **Methods** | | | | | |
| Update | Submit database on server VM1 | Separate database/table is created in VM | | ok | |

| | | | | | |
|---|---|---|---|---|---|
| DataDownload | Allows data download from server in csv format | Script is running on the server which gives access to download the data | | ok | |
| **Needs** | | | | | |
| SSO login | SSO login by cluster assistants for entering data to different villages | API provided by runtime. This is the same as used in other apps. | | | |
| Android APK | Code for running the application and data submission to server | Code repository is shared with PMU and runtime | | ok | |
| PHP script | For data insertion to database | | | ok | |
| PHP script | For image upload to the server from apk. | | | ok | |
| | | | | | |
| **APIs** | | | | | |
| | | | | | |
| **Used By** | | | | | |
| | Cluster assistants | | | | |

**Future tools and services**

1. Stream order tool 2.0 - All the drainage related assets will be snapped to the stream network. After this step, existing streams will be split into two parts based upon the location of the asset. Stream id's and stream order will be reassigned.
2. These new streams with assets will be added to the stream flow model. This will help in understanding or predicting the storage in these assets as per the rainfall runoff events.