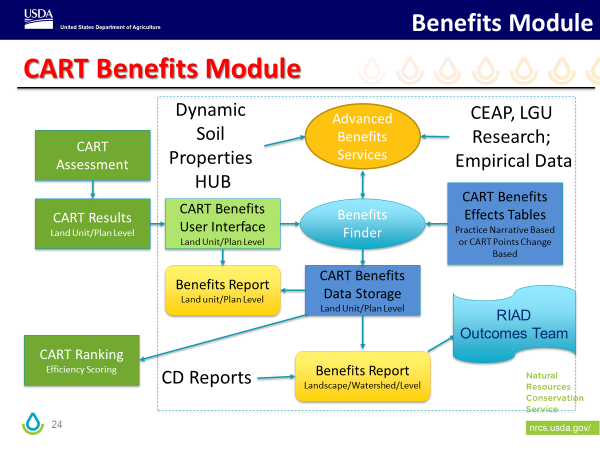
**Dynamic Soil Properties (DSP) Hub Technical Requirements - EDAPT Exemption**

The DSP Hub is fundamentally an innovative and experimental high-end geospatial data science workbench that builds new data products from a wide variety of existing data sets. The following background, technical requirements, example workflows, and suggested solutions are an attempt to clearly document the unsuitability of the EDAPT platform and administrative controls for the DSP Hub. Flexibility and agility to rapidly respond to customer requests for science-based soil property data is a critical customer need at the Deputy Chief, Chief, and Under Secretary- level.

**Conceptual Background**

* Dynamic soil properties are soil properties that change rapidly under the influence of land management, with the focus on conservation practice effects on soil (e.g. soil organic carbon)
* Conservation practice effects on soil properties is another way to refer to “environmental benefits” or “outcomes” of conservation practices
* The DSP Hub is the first to feed the CD/CART “Advanced Benefits Services” (diagram, right)
* The DSH Hub will establish data standards and a peer review process to embed a science-based foundation into conservation program and practice evaluation
* Science, statistics, and careful data structuring, data stewardship, and model integration will provide high-quality, defensible (i.e. authoritative) “benefits” estimates

**Situational analysis**

* USDA is under tremendous pressure to provide “outcomes” and “environmental benefits” as outlined in the 2018 Farm Bill over 20 times.
* The FY 2021 Omnibus Appropriations Bill (COVID-19 economic relief bill) added more urgency with the specific focus on soil carbon management
* The Biden Administration has a strong climate change agenda, with expectations of soil carbon being a major focus of that effort
* Existing USDA data sets are often on outdated technology, siloed databases and applications, and even on individual desktops.
* There are many different applications and reference tables that need to be gathered, cleaned, and transformed to create the “advance benefits services” core authoritative reference data sets
* In order to quickly gather and wrangle existing data assets from a variety of source applications, the DSP Hub will need a “Swiss Army Knife” approach with multiple flexible tools to wrangle the legacy datasets into a usable and integrated dataset.
* This agility and flexibility will allow for critical gleaning of data from decades of application-building with less-than-optimal data alignment and management

**DSP Hub Administration Controls Requirements**

* Speed and flexibility – DSP Hub will produce data products rapidly for urgent initiatives, which means that soil data scientists need to be able to pull in various geospatial and other data sources quickly and perform analysis and testing without the limitation of processing approvals
* DSP Hub products will focus on external process models and various reference data for those models which do not run in EDAPT
* Direct access (manual or API) to servers and applications will be needed (example, serving data to CART and CD processes, see diagram above)
* Flexibility for many different avenues to directly access data for models, such as access to internal servers and existing application data, transactional data, web browsers, and other disparate data sources (e.g. COMET or CART reference tables need to be downloaded and quickly added to the analysis within an hour)
* Direct data access is needed without approval wait time to add new datasets to complete analytical products for customers
* Functions, features, and data sources need to be added on a regular basis (potentially daily during a specific analytics request) to develop new models and new data sets
* The DSP Hub needs flexibility to create jobs to process data, including developing custom machine learning rules that match unique datasets and applications.
* There will be a zRoles structure for data products, peer reviews, approvals and workflow-type data administration
* DSP Hub will be using and producing authoritative data. The source datasets used for analysis in the DSP Hub environment must be authoritative and have a clear owner and version control in place.

**Approach to Logic, AI, ML, and Rules Engines Requirements**

* Ability to experiment with DSP data on multiple machine learning/predictive engines/fuzzy logic to support exploring developing algorithms and automation to predict outcomes under conservation practices
* DSP requires a dedicated tabular and spatial development database that can be administered by end users with full system administration privileges as a working area to support development of DSP
* Because DSP requires multiple artificial intelligence, analytic, and statistical tools, DSP requires a user interface and automation to reduce manual work interacting with multiple disparate tools
* An adaptable rules engine that can be directly adjusted by the DSP Hub owners will be needed to serve the needs of the DSP Hub workbench.
* There may need to be more than one adaptable rules engine
* Some workbench analysis and rules writing may evolve into broader rules during data exploration and wrangling. The DSP Hub owners will use SQL like a logical puzzle, with constant adaptation and scientific/conceptual analysis to achieve the desired end product

**Processing capacity**

* Predictive analytics often will need to run scenarios potentially thousands of times on the same land unit.
* Multiple soil geospatial layers will be processed at the same time without significant loss in performance.
* It is expected that the need will be for software that can efficiently utilize as many CPU cores as needed (48 or 64 core) with 512 GB RAM for typical analysis
* Consideration of RAID (Redundant Array of Inexpensive Disks) for reducing bottlenecks on spatial data analysis and operations
* DSP Data Hub is expected to grow to up to 1000 PBs
* DSP Data Hub will contain more than 100 quintillion records of data
* DSP requires a computational engine that can process 100 trillions of transactions per hour
* Due to the large, complex transactional needs, DSP computational engine is required to run server side (analytic tools on client-side would not meet transactional volume nor handle the size of the data)
* Ability to process multiple vector files – typical size 200 Gigabytes each, which typically would have files analyzed at one time
* Specific solutioning/software “wishlist” from experienced soil data scientists is included in Appendix A
* Ability to process multiple raster files – typical size 2 Terrabytes with potentially 150 analyzed at one time
* Ability to process tabular data files – typical size up to a Pedabyte, which would need to be analyzed with other raster and vector files

**Workflow Requirements**

* The data management and scientific processes will need a workflow management. The DSP Hub will need to process intake requests, job status, approvals, routing work, alerting, conditional branching of work, and returning work when not approved
* A typical workflow would be:
  1. Customer request
  2. Determine objectives (data planning, e.g. refinement of need and product expectations)
  3. Data gathering from many disparate sources (tabular, geospatial, external data, internal data, model reference data, and many more)
  4. Feasibility analysis for data product
  5. Data cleaning and wrangling
  6. Test analyses with algorithms and visualizations
  7. Review with customer
  8. Submit data product for review (scientific review, program review, as needed)
  9. Draft workflow for individual data product
  10. Develop SOP and data standard if needed
  11. Establish data provisioning (could be to EDAPT, API, embed within CART, models, etc)
  12. Establish update frequency and data management protocols
  13. Automate the process if needed
* A sample workflow for the initial product testing is attached as Appendix B

**The DSP Hub platform (stack/workbench) capability summary (MUST haves):**

1. Platform must support robust geospatial analysis
2. Data visualization needs to be included in every part of the process (loading, analysis, testing statistical models and algorithms)
3. Platform has to be able to quickly ingest/import raw data and allow for tagging and processing of data, adding additional data such as chain of custody, data source information, and other metadata. Some of this should be automated but need flexibility to do manually. Chain of custody should include a quality check/approvals for upload.
4. Platform has to support developing statistics/algorithms/rules to perform analysis on existing reference data and raw data
5. Tool needs to support manual querying and filtering to curate and aggregate data
6. Tool needs to support automated data mining and machine learning
7. Ability to aggregate and analyze data for some simple quick summaries (counts, averages, quick stats) from the database
8. Data unification and joining from multiple data sets to create new datasets and performing spatial and tabular joins
9. Ability to create and attribute data products (outputs) sets that will be used in various models
10. Analysis tools that allows for complex statistical framework (building and developing new models)
11. The tool includes a “test mode” with data visualization (charts, graphs, maps) to develop and test statistical models and algorithms
12. Tool needs to support generating and providing different output types, including customization of outputs
13. Data curation for external users allows for automated decision-making on systems that an organization uses to manage its interactions with customers, employees and suppliers.
14. The tool should allow for users to perform the above without doing scripting
15. Options to use R Studio server and publish to shiny app
16. Options for Python and SQL
17. API services
18. Predictive modeling

**Appendix A – Software solutions discussed that would support the effort**

* OGC file format and metadata
* Mainstream Linux distribution (Debian, Red Hat, etc.)
* PostgreSQL + PostGIS
* GRASS GIS
* R Studio Server
* MS SQL server for keeping current applications running
* Mapserver
* GDAL
* RStudio Server
* Apache WWW server
* Rules engine might have to be a google earth engine or something similar or more flexible.
* Platform to tie everything together (most important but still TBD)

**Solutioning background discussion from experts in manipulating soil data – in their own words**

PostgreSQL + PostGIS

Robustness – Any enterprise level database software, including PostgreSQL, is going to be orders of magnitude more robust (safe from crashes, data corruption, etc) than a file-based data storage. Commercial database server companies compete on the basis of robustness, security, and speed. If their databases fail it can be national news and billions of dollars are at stake so they have the best and the brightest software engineers, they can hire making sure that any possibility of crashing or security glitches are minimized. And open-source database server projects are not much different. In fact, most of the web is based on open source database servers because of their economic feasibility.

Speed – Many GIS operations can be performed in PostGIS itself, and since PostGIS is running on a server and servers are generally much more powerful than a desktop computer you may see significant performance gains and may be able to purchase less expensive computers for GIS users in your organization.

We need high-performance, multi-threaded software and libraries that can scale to very large vector, raster, and tabular data sources. This software should by fully compliant with OGC file format and metadata.

SQL Server sitting adjacent to PostgreSQL has demonstrably better spatial handling, indexing, manipulation and serialization capacity. Its ability to work with R vector data is more efficient to what we can do now. SQL server and Postgre don’t interfere with each other.

The DSP hub is similar in many ways to the NASIS modernization. The DSP hub will be transactional as well and even more complex as far as aggregating, modeling and the querying capabilities. Data from the DSP hub would be imported from NASIS but put back into NASIS. It’s transactional in a way that it will be constantly being manipulated. Data would be imported and entered in the DSP hub. It’s no different than the NASIS system but far more data hungry and processing power. If the NASIS modernization isn’t in the data lake, why would the DSP hub be in it as well.

PostGIS + PostGRE SQL

With a spatial database working with large datasets becomes possible. Not only easier, but sometimes it’s almost impossible to work on larger datasets without a database. Have you ever tried to open 2 gb csv file? Or tried to do some geoprocessing for a 800 mb Geo? Did you even know that Shapefiles (ESRI product) have a size limit? Of course, you can tackle some of these issues by using Geopackage or some other file formats, but in general PostGIS is the optimal tool for handling big (geospatial) data than using ESRI ArcGIS. A good example is ESRI portal and CART. There is lag and compatibility issues with SQL server when using just one layer.

As a rule of thumb, the more data a query has to fetch and more operations the database has to do (ordering, grouping etc), it becomes slower and thus less efficient. An efficient SQL query only fetches the rows and columns it really needs. SQL can work like a logical puzzle, where you really have to think thoroughly what you want to achieve. Under the ESRI framework this isn’t really achievable.

Linix

Having it on a linix distribution we cut out a tremendous amount of overhead that has occurred by the operating system. You don’t need a graphical interface. This is machine is made for crunching data and request that come over the network. If you do want to do some serious spatial analytics, automate your processes or in any way move your way of working with spatial data to the next level, using PostGIS and especially spatial PostGRE SQL is the way to go.

**Appendix B - Example Workflow of a single product needed for the MVP (currently in development)**

DSP Hub process needs for MVP

1. Ingest quality-reviewed activity data (such as results of laboratory analysis soil lab test results from the location (tabular and not standardized data - likely .csv).
2. Ingest geospatial data on the location of the activity data
3. Ingest covariate from various data sources
   * National Planning and Agreements/CART data
   * Individual PRISM datasets (raster datasets, a series of layers with a single variable)
   * Model outputs (final or interim, such as COMET-model emissions per scenario)
   * Geospatial soil data and attributes selected (vector/SSURGO/NASIS)

* NASIS Lab Data Mart – curate data from NASIS for the analysis
* CIG On-farm Soil Health Demonstration data (Excel spreadsheets?)
* Other geospatial data for covariates at the location of the project site

1. Data processing to get into the same data structure/framework (spatial and temporal) for the analysis
2. Load, extract/transform model reference data for identified analyses
3. Develop derivative data from the analysis (could be an interim data product for testing or further analysis)
4. Allow selection from a library of rulesets/functions/codes to:
   1. Existing stored coded processing step/function/ruleset that can be used for a variety of datasets (library of custom rules and functions for SHAPE scoring, COMET, aggregation of property by conservation practices etc.)
      1. Identify necessary rules, crosswalks and data needs
         1. INPUT data
         2. Covariate data
         3. Etc.
      2. Define AOI or other method to define scope
      3. Format INPUT data for analysis (cross-walk inputs into common fields, DSPs and mgmt. info)
      4. Retrieve and process covariate spatial Information (external (PRISM) and internal (SSURGO fields)
      5. Apply previously developed statistical analysis (R SHAPE function)
      6. Output result (SHAPE score with mgmt. info)
      7. Visualize results (maps, graphs by mgmt. info etc.)
      8. Use result as new input for further analysis (Advanced Map or Analyze scores by new covariates)
   2. Development of new output/products- iterative
      1. Identify necessary data, rules and crosswalks
      2. Organize possible covariate info
      3. Develop cross-walks for input and covariate info
         1. Use stored ‘roestta stone’
         2. Develop new cross-walks between datasets
      4. Develop statistical model for analysis
      5. Visualize and test interim results
      6. Define AOI or other method to define inference (where product is viable)
      7. Iterate
      8. Finalize as a known process /ruleset with all data, cross-walk and stats needed for future use
      9. QA/QC of new process/ruleset
      10. Make available for users to re-use as ‘known process’

So specifically for SHAPE

1. SHAPE - soil health scoring
   1. On a quarterly basis, ingest quality-reviewed results of laboratory analysis from project sites
   * Core dataset
     + CIG OF-SHDT
   * Additional data
     + CAP soil health testing
     + SPSD - Lab Data Mart
     + Other cooperating labs?
   1. Identify necessary rules and crosswalks
      1. Updated shape data inputs
      2. Updated crosswalks
      3. Updated stats models
   2. Format INPUT data for analysis
      * 1. join OFSHDT, CAP and LDM data into common fields - DSPs and mgmt. categories and info
   3. Retrieve and process covariate spatial Information
      * 1. PRISM - MAT and MAP
        2. SSURGO - surface texture and suborder
           1. Apply rule to reclassify for statistical analysis
   4. Apply Previously developed statistical analysis
      * 1. R function - developed by researchers and peer reviewed
   5. Output result (SHAPE score with all other information attached)
      1. Visualize results (maps, graphs by mgmt. info etc.)
   6. Perform some sort of QA/QC process
   7. Publish individual results and visualization back to users
   8. Use results as new input for further analysis (Advanced Map or Analyze scores by new covariates)
      1. Aggregate and analyze by categories (mgmt. categories)
      2. Aggregate and analyze by spatial areas (states)
      3. Aggregate and analyze by other covariates (SSURGO, MLRAs, PRISM derivaties)
      4. Create visualizations
      5. QA/QC
   9. Review and approval process
   10. Publish aggregation and maps
       1. SHD
       2. SPSD
       3. Users
       4. Other Stakeholders