# Downloading Source Raster Tiles from Image Services in Virginia

**Wendy Stout**, Geospatial Program Coordinator,
Virginia Geographic Information Network

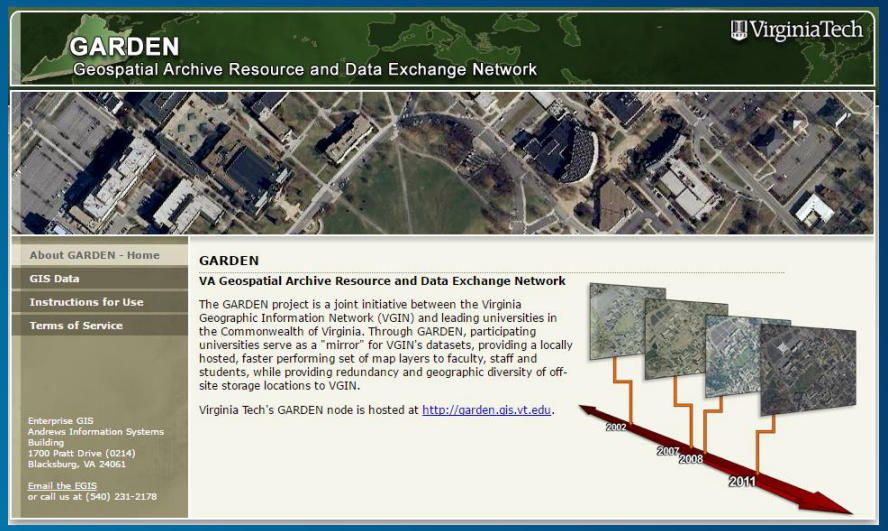
**Seth Peery**, Sr. GIS Architect,
Virginia Tech

# Virginia Geographic Information Network

- VGIN was established in 1997 in Virginia Code to "foster the creative utilization of geographic information and oversee the development of a catalog of GIS data available in the Commonwealth".
- Some of the core functions of VGIN are:
  - Develop & recommend policies & guidelines required to support state and local government exchange, acquisition, storage, use, sharing and distribution of geographic or base map data and related technologies
  - Compile a data catalog consisting of descriptions of GIS coverages maintained by individual state and local government agencies
  - Setting priorities for the development of state digital geographic data and base maps that meet the needs of state agencies, institutions of higher education, and local governments
  - Provide services, geographic data products, and access to the repository at rates established by the Division
  - Source: Virginia Code § 2.2-2026 and § 2.2-2027

# Geospatial Archive Resource and Data Exchange Network

## VGIN/Virginia Tech Partnership




- GARDEN is the **G**eospatial **A**rchive **R**esource and **D**ata **E**xchange **N**etwork, a federation of partners to provide access to geospatial data for users in the Commonwealth of Virginia.
- GARDEN provides a common framework of geospatial infrastructure that can be replicated across universities, such that a sustainable and resilient network of GARDEN nodes can be created and maintained.
- GARDEN provides universities with complete access to the latest geospatial data for the Commonwealth, fostering education and research in a variety of disciplines.
- GARDEN provides VGIN with backup and redundancy of services which it provides to the larger geospatial community within the Commonwealth.
- Both parties benefit from a sharing of technologies and best practices.

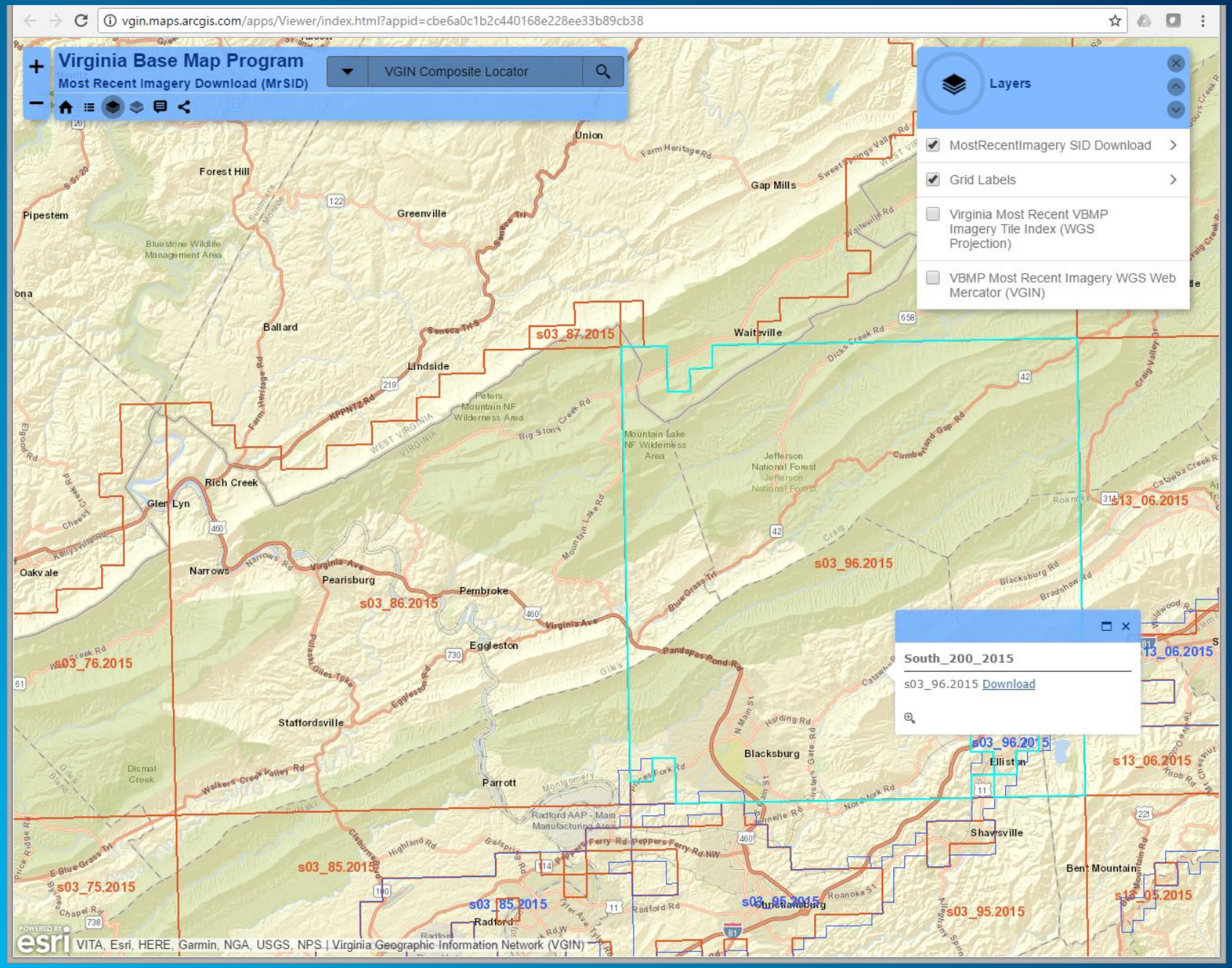
Virginia Tech has partnered with VGIN to become one of the Commonwealth's most active and useful GARDEN sites. Through this partnership, VGIN hopes to tackle some of its data sharing challenges with innovative solutions.

# Rationale

- VGIN contracts for the collection of orthoimagery of one third of the state every year over a 4 year cycle (no acquisition on 4th year).
- Imagery products are delivered to localities as TIFF and MRSID
- There is a constituency of users who need source image files but do not directly receive an imagery deliverable (e.g., researchers, contractors)
- The workflow for obtaining the imagery and other data from VGIN required human interaction and effort to identify, copy and deliver the files.
- There was a need to implement a better solution and develop a self service delivery capability for full source imagery.

## Current Best Practices - VGIN Data Sharing Approach

- VGIN was spending a significant amount of time filling MrSID imagery and LiDAR data requests, taking time away from our other responsibilities.
- To resolve this issue, we stood up 2 ArcGIS online applications – One for LiDAR and one for MrSID Imagery.
- These self service data transfer applications allow users requesting data to be able to research and download from our server the data they needed without VGIN having to assist.
- Just this year to date it saved us from having to fill about 50,000 pieces of data requested each month.
- Seamless map caches available to the public at large are also created for the entire state and served on gismaps.vita.virginia.gov
- VGIN still has a need for a downloadable solution for TIFF data.


vgin.maps.arcgis.com/apps/Viewer/index.html?appid=cbe6a0c1b2c440168e228ee33b89cb38
Virginia Base Map Program
Most Recent Imagery Download (MrSID)
VGIN Composite Locator
Layers
MostRecentImagery SID Download
Grid Labels
Virginia Most Recent VBMP Imagery Tile Index (WGS Projection)
VBMP Most Recent Imagery WGS Web Mercator (VGIN)
South_200_2015
s03_96.2015 Download
s03_87.2015
s03_86.2015
s03_76.2015
s03_96.2015
s13_06.2015
s03_85.2015
s03_75.2015
s03_95.2015
Union
Forest Hill
Pipestem
Greenville
Gap Mills
Ballard
Waiteville
Lindside
Rich Creek
Glen Lyn
Narrows
Pearisburg
Pembroke
Eggleston
Staffordsville
Blacksburg
Parrott
Elliston
Shawsville
Radford
Christiansburg
Bent Mountain
VITA, Esri, HERE, Garmin, NGA, USGS, NPS | Virginia Geographic Information Network (VGIN)

gismaps.vita.virginia.gov/arcgis/rest/services/MostRecentImagery/MostRecentIm

**ArcGIS REST Services Directory** Login |

Home > services > MostRecentImagery > MostRecentImagery_WGS (MapServer) Help | API Re

JSON | SOAP | WMS | WMTS

# MostRecentImagery/MostRecentImagery_WGS (MapServer)

**View In:** ArcGIS JavaScript ArcGIS.com Map Google Earth ArcMap ArcGIS Explorer

**View Footprint In:** ArcGIS.com Map

**Service Description:** Orthoimagery collected in Spring 2013, 2014 or 2015 (whichever is most available) by the VBMP program for Virginia. The imagery is displayed in true color (RED, GREEN The spatial reference is WGS 1984 Web Mercator (Auxiliary Sphere). The imagery is tiled at 12 l from 1:4,622,324 to 1:2,257. The imagery was collected to meet ASPRS Class 1 orthoimagery s Most areas were collected at a 1-ft ground sample distance (GSD), with some urban areas upgr 6-inch and 3-inch GSD. - "Any determination of topography or contours, or any depiction of phy improvements, property lines or boundaries is for general information only and shall not be used design, modification, or construction of improvements to real property or for flood plain determi Subsection C of § 54.1-402

**Map Name:** Layers

Legend

All Layers and Tables

**Layers:**

- VBMP (0)

**Description:**

**Copyright Text:**

**Spatial Reference:** 102100 (3857)

**Single Fused Map Cache:** true

**Tile Info:**

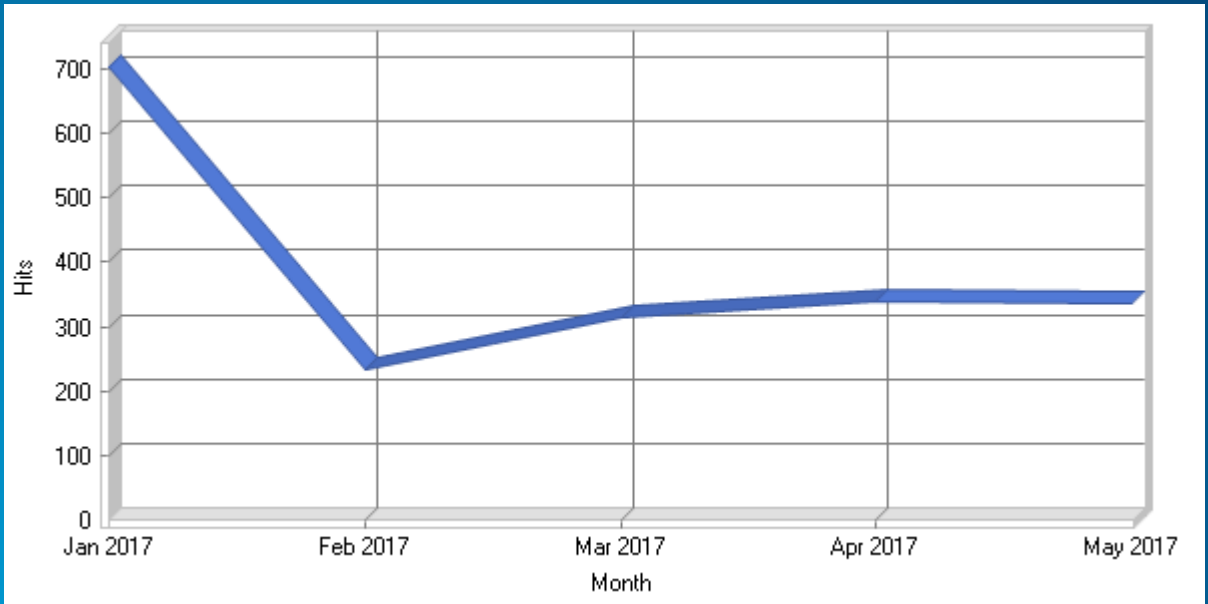
- **Height**: 256
- **Width**: 256
- **DPI**: 96
- **Levels of Detail:** *12*
  - **Level ID:** 0 [ Start Tile, End Tile ]
    - Resolution: 1222.992452562495
    - Scale: 4622324.434309
  - **Level ID:** 1 [ Start Tile, End Tile ]
    - Resolution: 611.4962262813797
    - Scale: 2311162.217155
  - **Level ID:** 2 [ Start Tile, End Tile ]
    - Resolution: 305.74811314055756
    - Scale: 1155581.108577
  - **Level ID:** 3 [ Start Tile, End Tile ]
    - Resolution: 152.87405657041106

gismaps.vita.virginia.gov/arcgis/rest/services/MostRecentImagery/MostRecentIm

**ArcGIS JavaScript API: MostRecentImagery/MostRecentImagery_WGS**

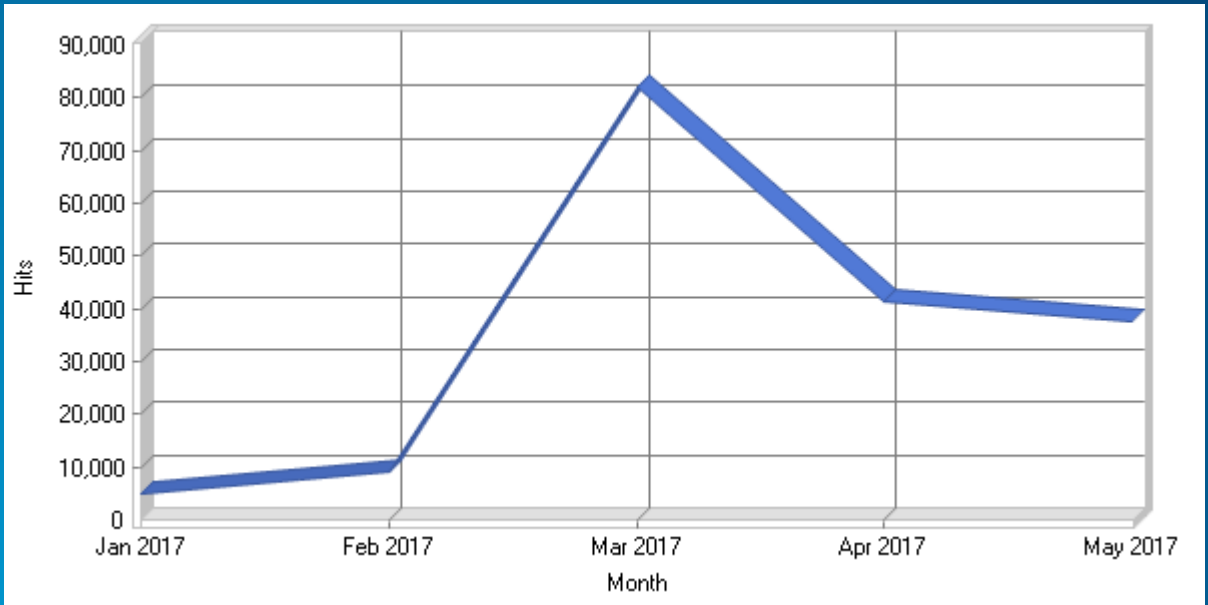
Built using the ArcGIS JavaScript A

# MrSID Usage Statistics




| Month | Hits | Incomplete Requests | Bandwidth (KB) |
|---|---|---|---|
| Jan-17 | 702 | 8 | 457,170,712 |
| Feb-17 | 232 | 10 | 158,171,551 |
| Mar-17 | 315 | 30 | 224,275,769 |
| Apr-17 | 339 | 21 | 254,895,199 |
| May-17 | 336 | 6 | 252,936,233 |
| Total | 1,924 | 75 | 1,347,449,466 |

# LiDAR Usage Statistics




| Month | Hits | Incomplete Requests | Bandwidth (KB) |
|---|---|---|---|
| Jan-17 | 4,892 | 6 | 46,022,249 |
| Feb-17 | 9,247 | 45 | 604,871,720 |
| Mar-17 | 82,022 | 70,139 | 80,449,788 |
| Apr-17 | 41,579 | 26,607 | 145,349,625 |
| May-17 | 37,542 | 23,705 | 153,623,657 |
| Total | 175,282 | 120,502 | 1,030,317,042 |

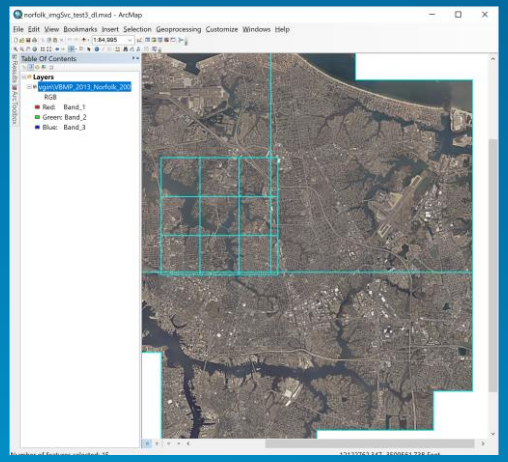
# Functional Requirements for an interactive TIFF Download Utility

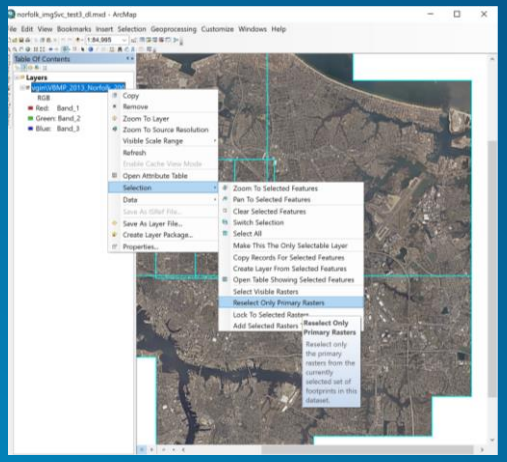
- Interactive web map to allow users to define AOI for download
- AOI selection should return full tiles that intersect without clipping
- Rate limiting should be imposed at the presentation tier
- Download package creation can be synchronous or asynchronous
- Users should not need to know or look up filenames/tile IDs; only inputs should be
  - Product type (collection year)
  - AOI (rectangle intersecting tiles)

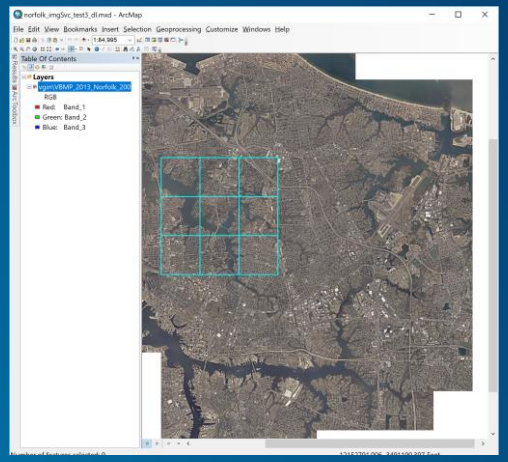
# Candidate Implementation Patterns

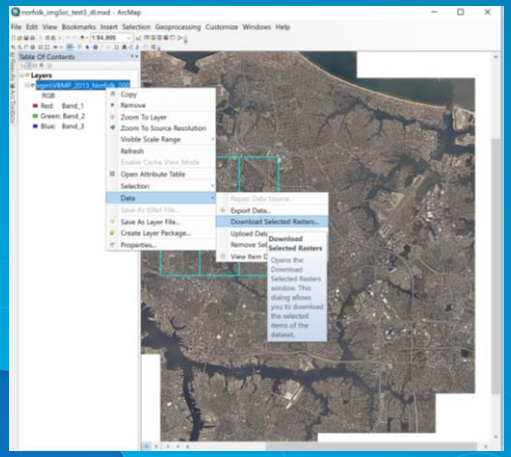
1. TIFF imagery online, backing Image Service
    1. ArcMap Download
        1. Build-in functionality
        2. Requires client software, learning curve
    2. Web Application
        1. Image service renders in JSAPI web app
        2. <ImageService>/query,/download,/file methods of REST API are used to extract source files
        3. Requires GP svc for assembly of ad hoc download packages; asynchronous; return link to /arcgisoutput relative .zipfile URL
2. TIFF imagery stored elsewhere
    1. Near-line tape (e.g., SGI DMF) / S3 / Glacier / etc
    2. Latency and storage/data transfer cost varies
    3. Requires indexing to be done by other means (e.g., traditional tile index FC)
    4. Separation of Download source from Display source (e.g., map cache)

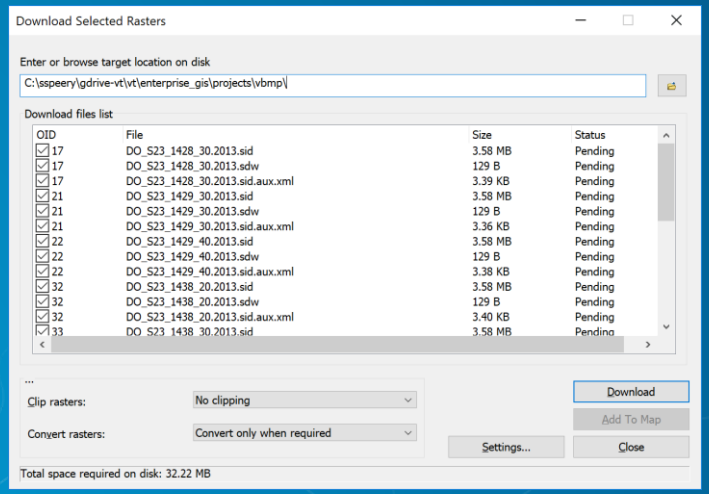
# Functionality already built into ArcMap



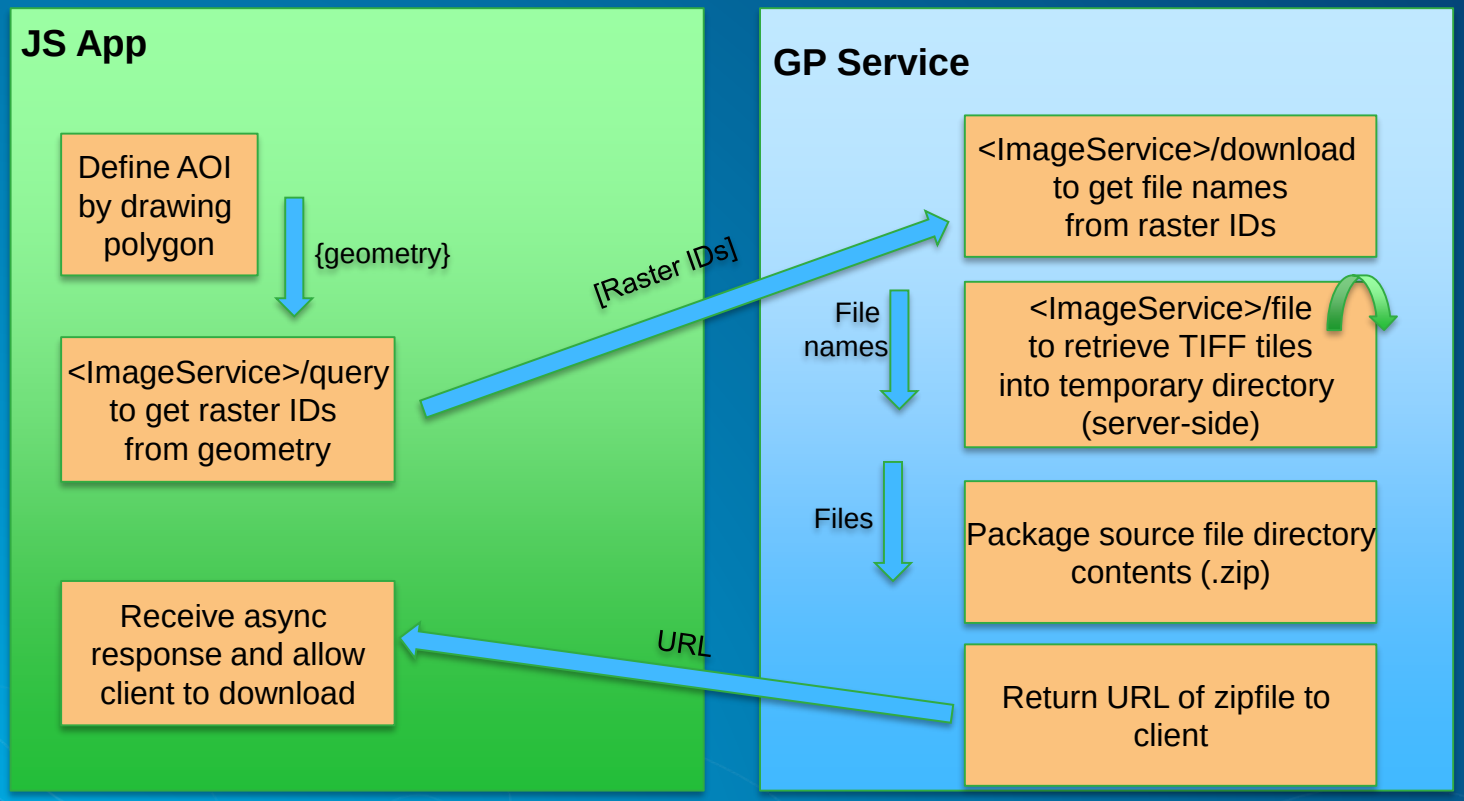




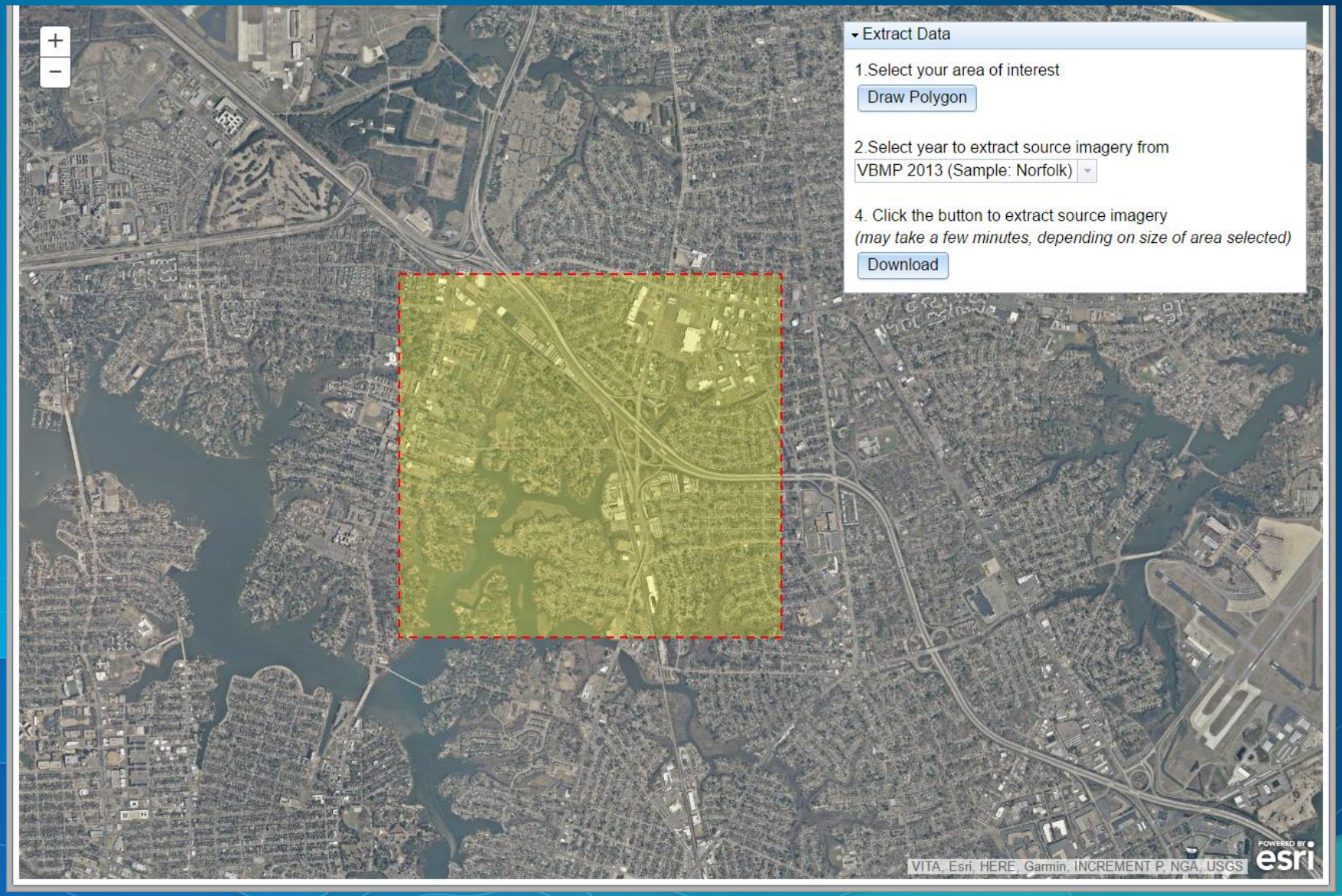




# Architecture of a comparable Web-based Image Service Download Implementation



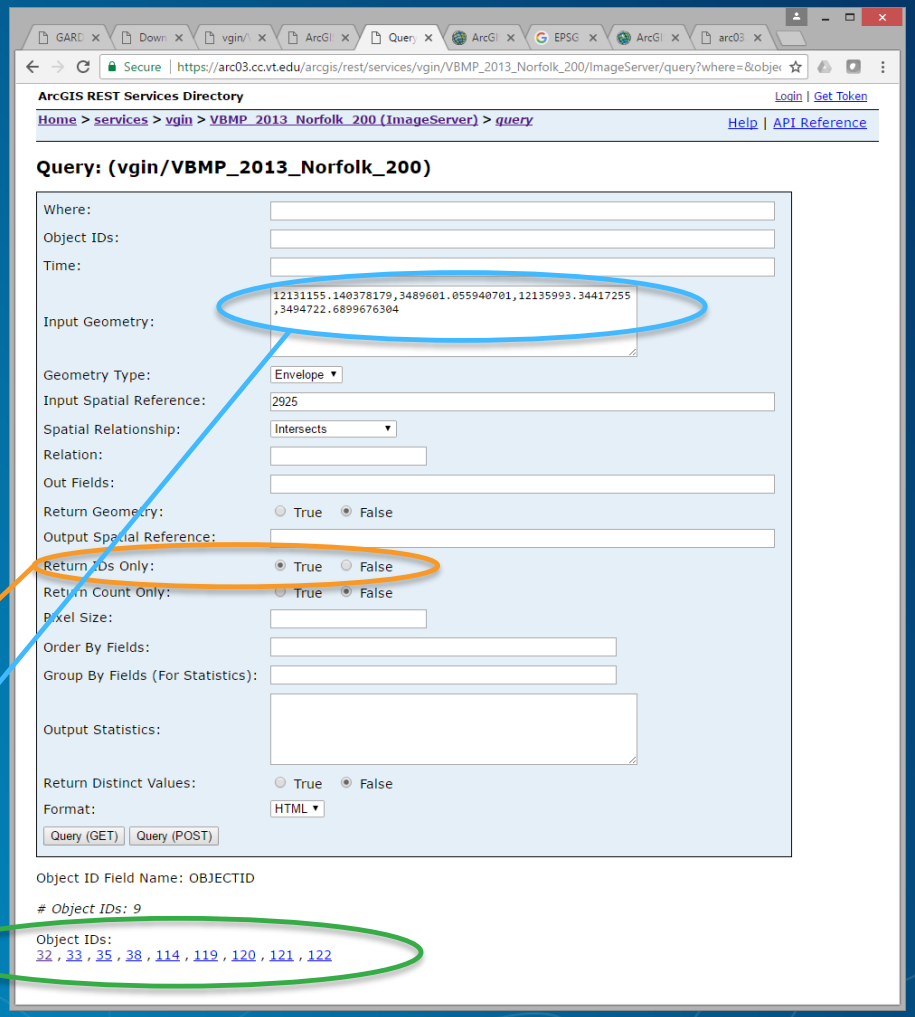
# UI Overview


Extract Data
1.Select your area of interest
Draw Polygon
2.Select year to extract source imagery from
VBMP 2013 (Sample: Norfolk)
4. Click the button to extract source imagery
(may take a few minutes, depending on size of area selected)
Download
VITA, Esri, HERE, Garmin, INCREMENT P, NGA, USGS
POWERED BY esri

# REST API step #1: Query for Raster IDs based on selection envelope

1. **Input AOI from interactive selection**
2. **Select "return IDs only" to get array of IDs for use in download() method**
3. **Format here is HTML for demonstration; the real request and response are:**

https://arc03.cc.vt.edu/arcgis/rest/services/vgin/VBMP_2013_Norfolk_200/ImageServer/query?geometry=12131155.140378179%2C3489601.055940701%2C12135993.34417255%2C3494722.6899676304&geometryType=esriGeometryEnvelope&inSR=2925&spatialRel=esriSpatialRelIntersects&returnGeometry=false&returnIdsOnly=true&f=json



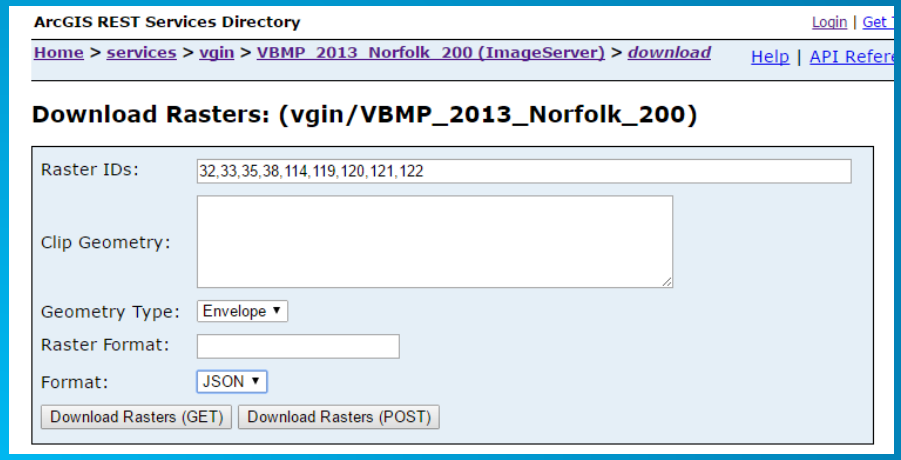

**Response:**

{"objectIdFieldName":"OBJECTID","objectIds":[32,33,35,38,114,119,120,121,122]}

# REST API Step #2: Get Physical Filenames

- Using the array of Raster IDs from previous step we can obtain the (relative) path names to the source files
- Because this is a REST request we can issue it from the JS app or from a Python GP Task

https://arc03.cc.vt.edu/arcgis/rest/services/vgin/VBMP_2013_Norfolk_200/ImageServer/download?rasterIds=32%2C33%2C35%2C38%2C114%2C119%2C120%2C121%2C122&geometry=&geometryType=esriGeometryEnvelope&format=&f=pjson





```
{
 "rasterFiles": [
  {
   "id": ".\\Norfolk\\GeoTIFF\\DO_S23_1438_20.2013.tif",
   "size": 3749880,
   "rasterIds": [
    32
   ]
  },
  {
   "id": ".\\Norfolk\\GeoTIFF\\DO_S23_1438_20.2013.tfw",
   "size": 129,
   "rasterIds": [
    32
   ]
  },
  {
   "id": ".\\Norfolk\\GeoTIFF\\DO_S23_1438_20.2013.tif.aux.xml",
   "size": 3479,
   "rasterIds": [
    32
   ]
  },

. . . .
```

## Processing the Download

- Yes, we *could* iterate over the JSON output of <ImageService>/download and use ImageService/file to download each individually
- *Don't do that.*

- Ad hoc zip file assembly for aggregated file downloads require a GP service with access to the physical filesystem
- Requires additional computational resources
- Can be done with Python os, zipfile functions
- Getting data back to the client
  - Must be async; this takes some time
  - write to arcgisoutput or arcgisjobs dir; then send URL to client while they wait
  - Treat a download as an "order"; write to arcgisoutput or arcgisjobs dir; then e-mail URL to client (useful for long running jobs, does not require client to stay in download UI (remember seamless.usgs.gov?)

# Design Considerations (1)

- A multistep process is needed to extract the source files
  - <Imageservice>/query : Identify the File IDs
  - <Imageservice>/download : does not really download anything; gets paths
  - <Imageservice>/file : actually downloads a file
    - NOTE: if running GP service on same server, and relative path can be resolved to a host disk physical path, download package can be prepared using python OS module instead of the Imagerservice/file method, preventing congestion of network interface
    - A call to an Image Service's REST API function <Imageservice>/file results in outbound data transfer (may become an issue in metered PaaS environments
  - Image Services essentially serve an indexing function
  - There are some significant drawbacks to the Image Service Download Pattern:
    - TIFF source must be online and attached to the ArcGIS Server instance hosting the Image Service
    - Caches are much lighter weight and faster for display (Image Service itself not strictly needed for display of reference images in UI tier)
    - Demand for TIFF product may not justify storage penalty

# Design Considerations (2)

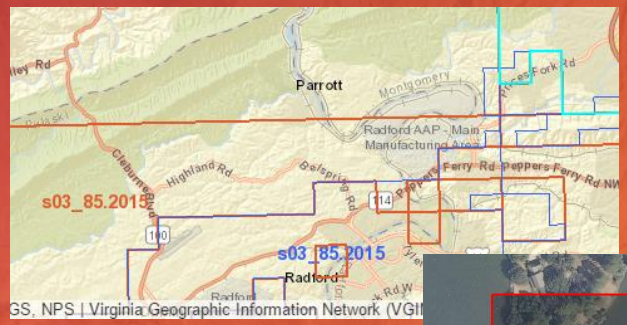
**Hybrid approaches (/performance improvements /next steps)**

***Enabled by loosely coupled nature of REST API calls that drive the platform***

1. **Replace Image Service with Map cache for fast visualization in lightweight JS API app that retains same application logic to facilitate AOI delineation**
2. **Use of S3 buckets / Glacier for storage of TIFF sources + traditional tile index feature classes**

**Assembly of download packages for end user is still problematic…**

1. **Solution #1: individual file downloads; annoying, prone to error, incongruent with client use cases**
2. **Solution #2: pre-created file collections; delete what you don't need once on local system**
3. **Solution #3: custom, ad-hoc assemblies of tiles as .zip**




# Demonstration

Comparing Imagery Access Patterns