DEVELOPER SUMMIT

March 10–13

WELCOME

Map Caching in ArcGIS for Server

Eric Rodenberg

Tom Brenneman

Schedule

- Planning and building a cache
- Architectures for caching
- Caching in the cloud

- We will hold periodic breaks for questions
- #DEVSUMMIT #CACHE

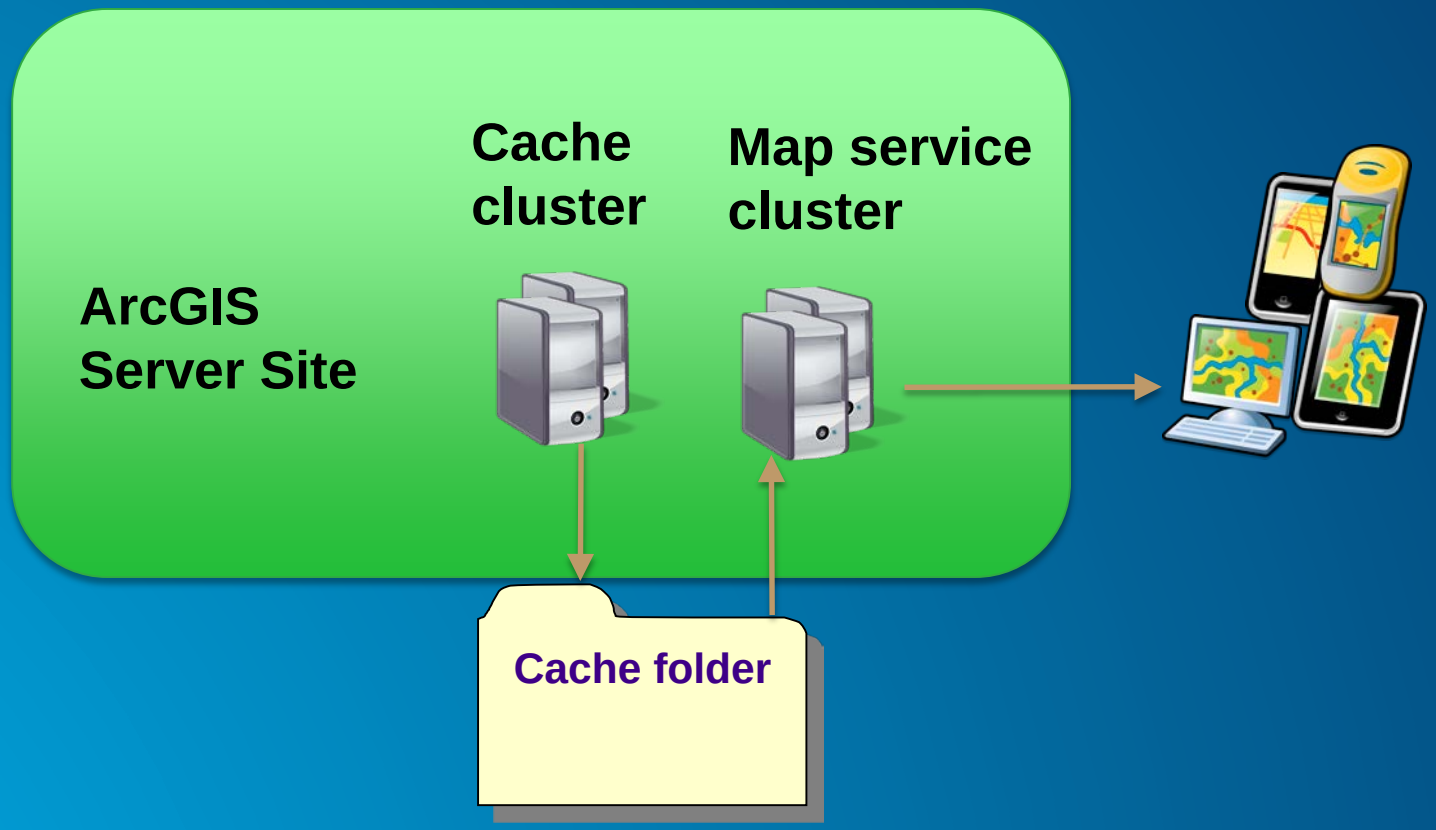
Demo of caching with ArcGIS

- CachingTools geoprocessing service
- Defining cache properties
  - Build automatically or manually
  - Min and max cached scales
  - Image format
- Calculating cache size
- Asynchronous caching
- Real-time cache status updates
- Viewing job status and fixing problems
- Visualizing cache creation over time

Short break for questions

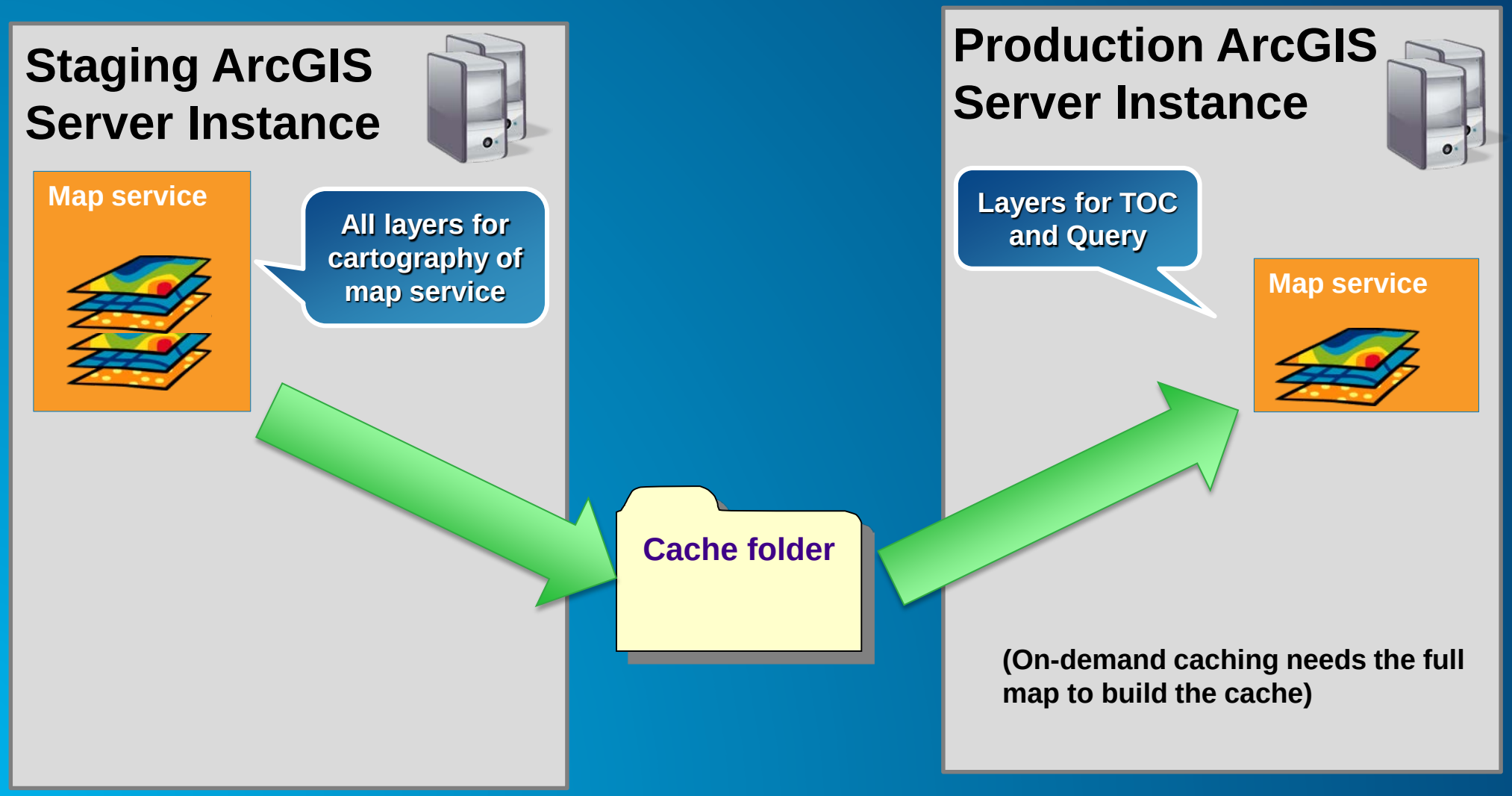
Architecture for caching and cache updates

Clusters allow you to isolate cache creation



Cache cluster can scale out while the cache job is running

Update a cache using a staging server

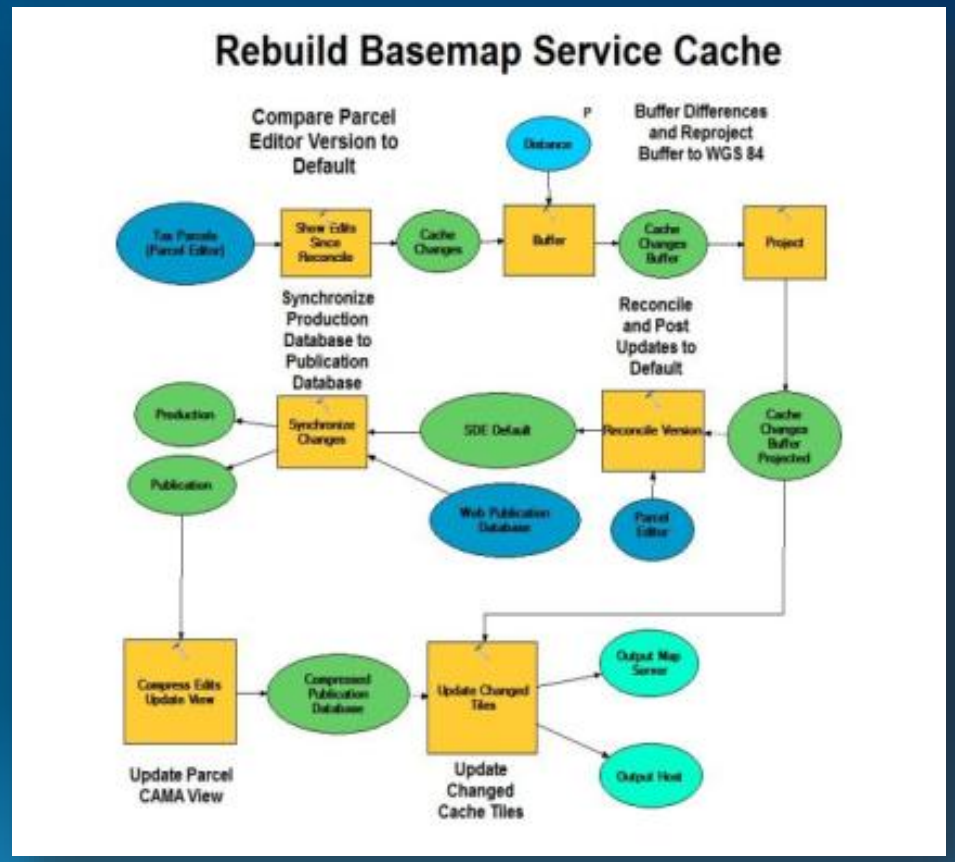


Cache update strategies

- Rebuild the entire cache
  - Size of cache
  - Time to cache
- Rebuild specific tiles
  - Rebuild at specific scales
  - Rebuild areas based on change detection

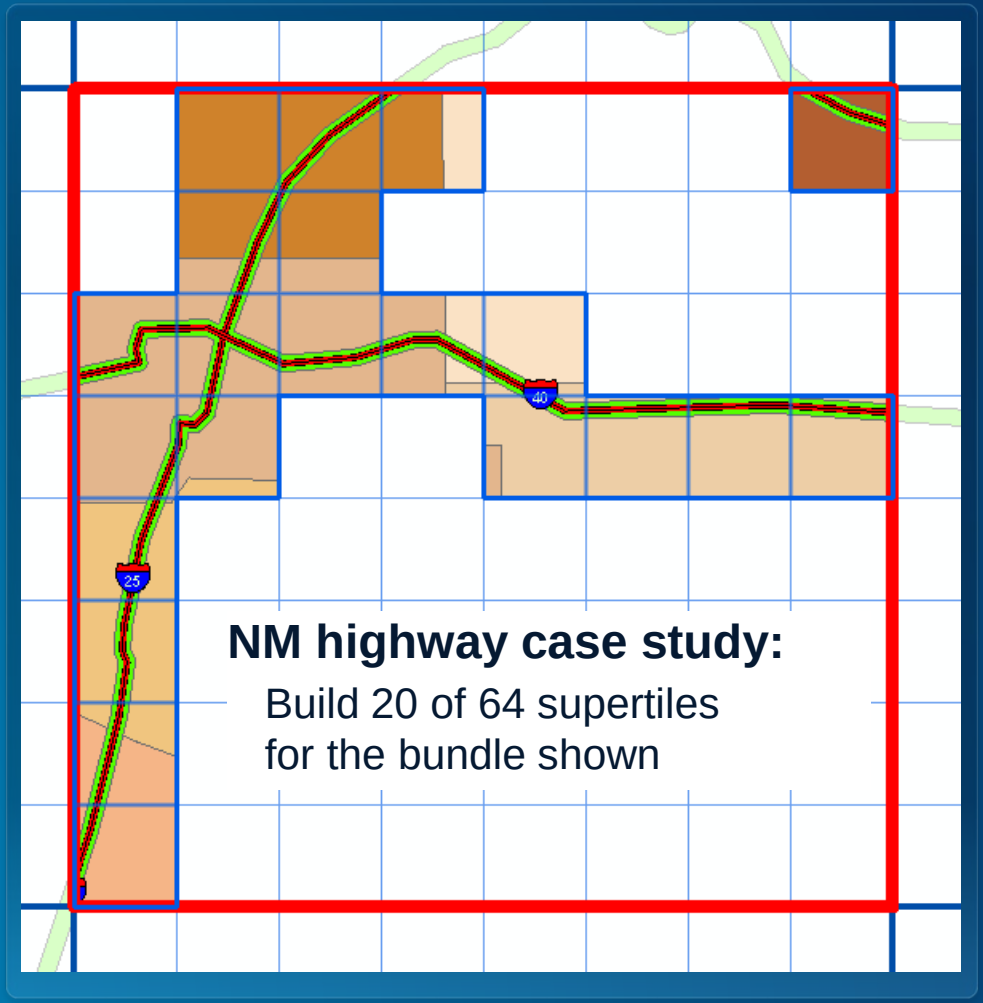
Cache Update Automation

- Use Model Builder to script Cache Update Automation
  - Rebuild Specific Tiles
  - Export to Python
  - Schedule Run Time
- Tools
  - Data Comparison Toolset
    - Feature Compare
    - Detect Feature Changes for lines (New at 10.2.1)
- Other samples
  - Compare feature classes
  - Show edits since reconcile
- Sample available on GitHub
  - https://github.com/Erodenberg/UpdateTileCache



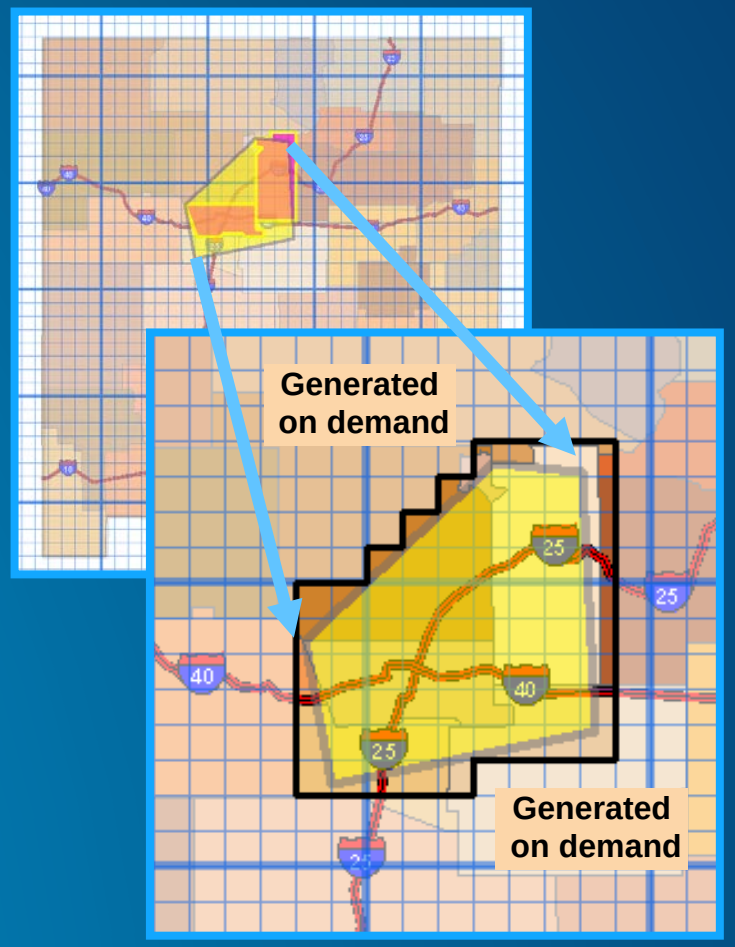
How much of a cache should you generate?

- Cache by feature
  - Geographic elements
  - Generates tiles for intersecting supertiles
- Saves on…
  - Generation time
  - Processor resource
  - Disk usage



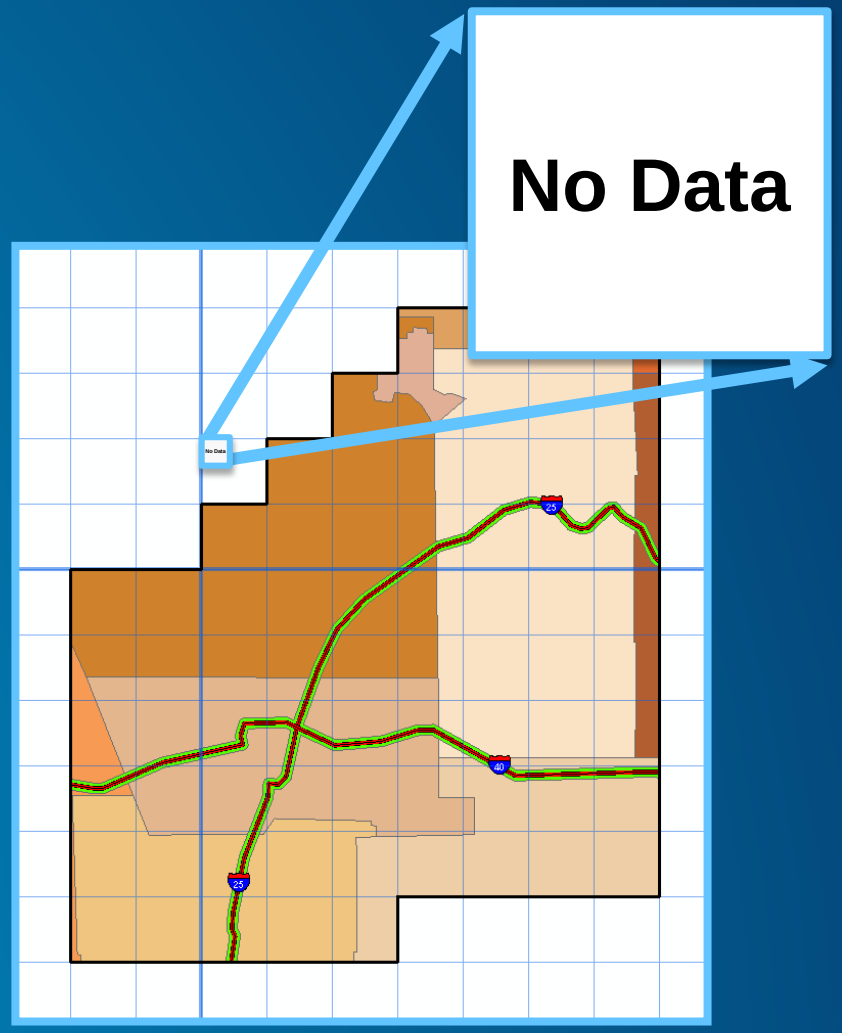
Pre-create coupled with cache on demand

- Pre-create high use areas
  - Population centers
  - Parks, roads, attractions
- Features
  - Cover popular extent
  - Generate key tiles
  - All others generated on demand



Handling tiles you do NOT create

- Create “No Data” tile
  - Same image format (JPG or PNG)
  - Same size (256 x 256)
  - Save in cache folder
    …\<dataframe>_alllayers
- No Data Tile Help Documentation



Other caching techniques

Image Services
ArcGIS Online
Tile Packages
Multiple tile servers

What is image service caching

- Fast access to images as a tiled service
  - Out performs / scales mosaic dataset and raster dataset
  - Imagery is not processed on the fly
- Uses image extension

Why should I cache image services

- Improved performance for basic images
  - Can not modify mosaic methods
- Skip overview generation
  - Tiles generate from large scales to small scales
- Improve performance for slow formats
  - Recommended for highly compressed formats – e.g. JPEG2000, MrSID
- Caching image services is much faster than caching map services with imagery
  - Faster rendering engine for imagery

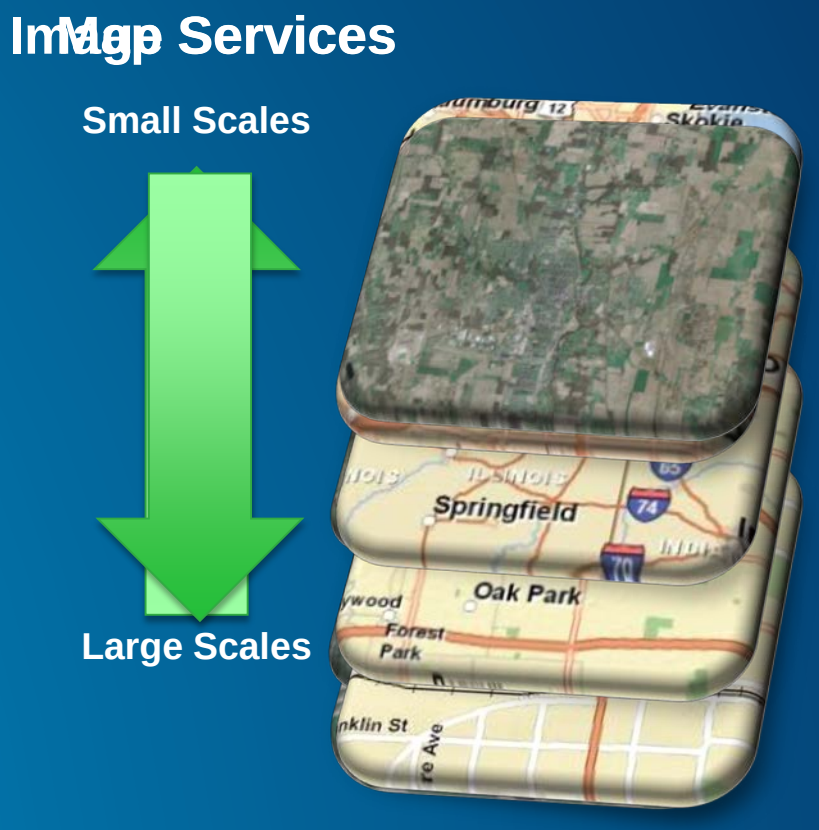


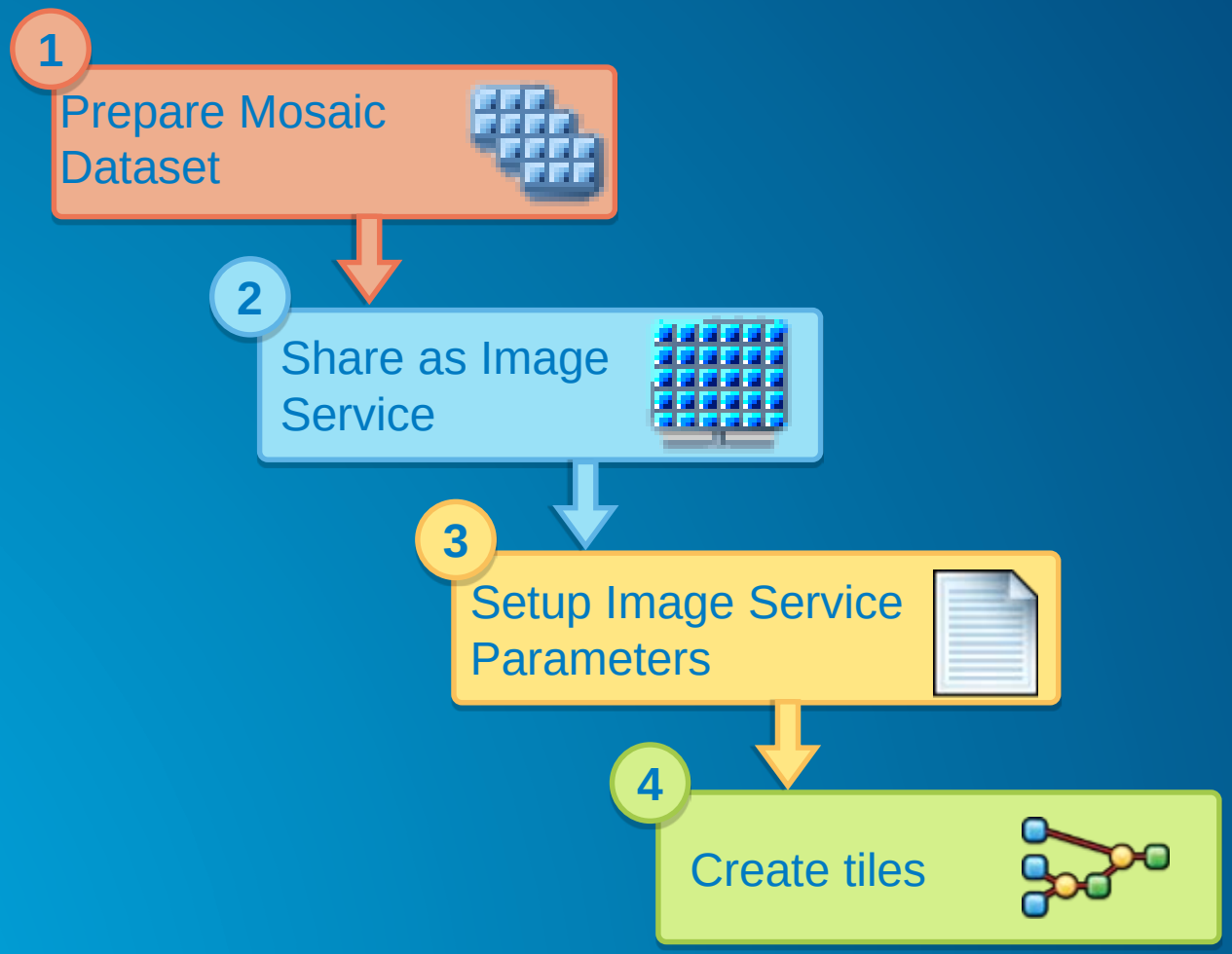
Image resolution and cache scales

- ArcGIS Server chooses the optimal scales for Imagery
  - ArcGIS will not exceed raster resolution
- Neither should you!
  - ArcGIS Server resamples imagery exceeding raster resolution
  - ArcGIS Desktop zoom to raster resolution
- Scale based on 96 DPI

Scale (Ft) = (x/12) * 96 Scale (m) = (x/0.0254) * 96

Cell Size(Ft)	Scale 1:X	Cell Size (m)	Scale 1:X
.25	288	0.3	1133.858
.5	576	0.6	2267.717
1	1152	1	3779.528
3	3456	3	11338.58
15	17280	15	56692.91

Building an image service cache



Choosing the best image format

- Vector maps
  - PNG
    - Even with lots of colors or hillshades
- Imagery
  - No transparency
    - JPEG (start with quality = 55)
  - Need transparency at the edges
    - Mixed (use the same quality as JPG)

Which one looks better?

JPEG 90 – 25KB

JPEG 90 – 30KB

Switching to a Dynamic Request

- Supported by Map and Image Services & the Web API's
- Force a Cached Service to Render Dynamic
  - Use ArcGISDynamicMapServiceLayer(Cached Service URL)
- Render cached tiles from smaller scales levels 0 – 15
- Render dynamic services at larger scale levels 16 – 19
  - Large scales = less area to draw features = faster

Scale Levels 10 – 17 Cached

Scale Levels 18 – 19 Forced Dynamic Draw

Switching to dynamic display

http://arcgiswebapps.s3-website-us-east-1.amazonaws.com/applications/Caching/DynamicDraw/index.htm

Code: https://github.com/Erodenberg/DynamicDraw

Caching in ArcGIS Online

- ArcGIS Online subscription allows for caching
- No need to worry about capacity
- Charged by tile creation and storage
- Two approaches
  - Upload data to AGOL
    - Build and store cache with AGOL
  - Upload tile package to AGOL
    - Build cache on premise but store with AGOL
- Understanding credit usage: http://www.esri.com/software/arcgis/arcgisonline/credits

Why create a tile package?

- Local cache for Desktop and Runtime applications
- Transport a map cache
- Upload a map cache to ArcGIS Online

Creating a tile package

- ArcMap Options > Sharing > Enable ArcGIS Runtime tools
- Two options for creation
  - Create tile package within ArcMap
    - Single processor
    - File > Share As > Tile Package
  - Create cache with server
    - Tile Cache > Export Tile Cache
      - Uses Parallel Processing Factor Geoprocessing Environment setting

Using multiple domains

- With multiple services
  - Use a different domain for each services
- With one service
  - API's support multiple web services endpoints for a single layer

```
var layer = new esri.layers.ArcGISTiledMapServiceLayer(
        "http://www.mydomain.com/ArcGIS/rest/services/myservice/MapServer",
        { tileServers: [
        " http://cache1.mydomain.com/ArcGIS/rest/services/myservice/MapServer ",
        " http://cache2.mydomain.com/ArcGIS/rest/services/myservice/MapServer "]
        });
```

- Use with small cache tiles
- Can reduce browser caching and result in more HTTP connections

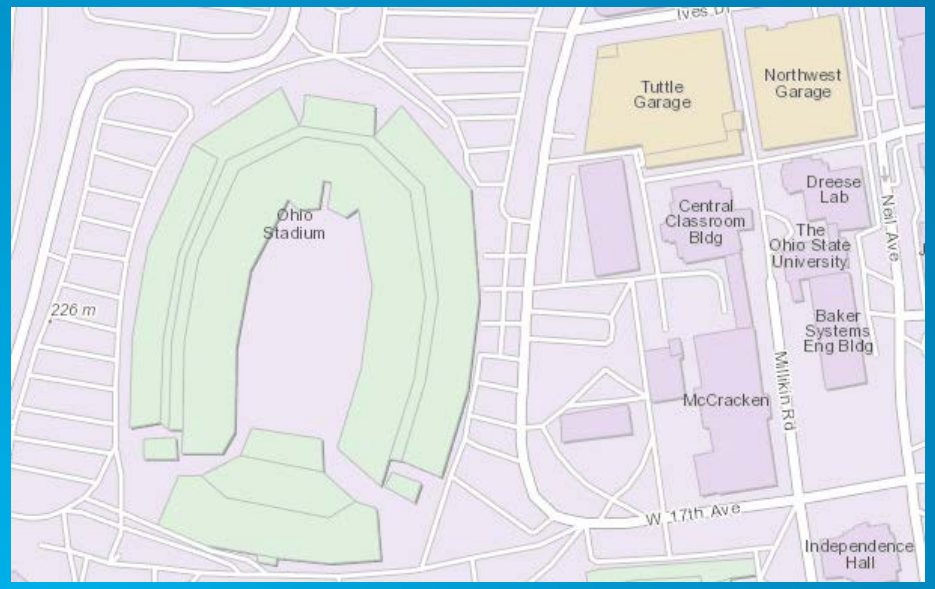
Multiple domains

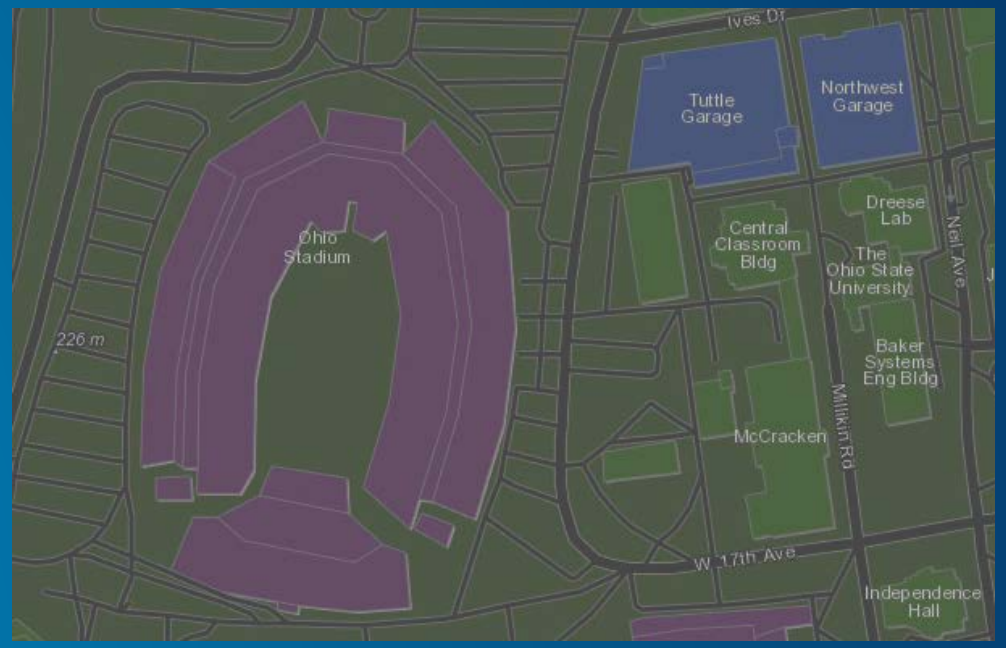
http://arcgiswebapps.s3-website-us-east-1.amazonaws.com/applications/Caching/MultipleDomains/index.html

Code:
https://github.com/Erodenberg/MultipleDomains

CSS Filters

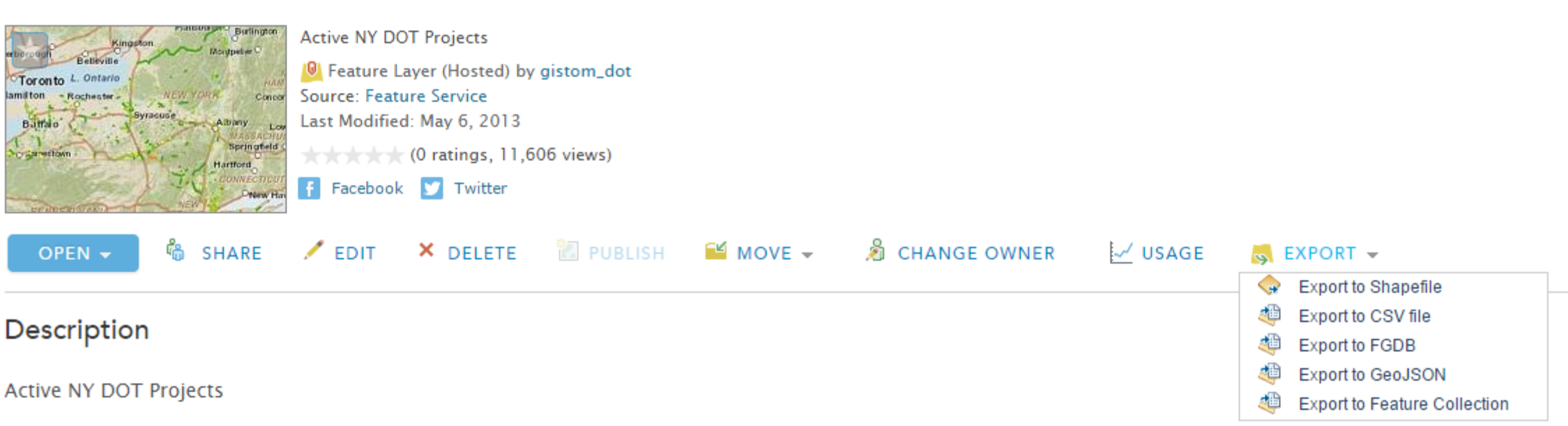
- .layerTile class added to cached tiles by JS API
- Add .layerTile class to your CSS stylesheet to change the look of the tiles





Feature Collections in ArcGIS Online

- Feature caching of feature services
  - Read only copy with independent access control
- Improved performance and scalability
- API's consume it as a RO feature service



Feature collections and CSS filters

http://arcgiswebapps.s3-website-us-east-1.amazonaws.com/applications/Caching/CSSFilters/Index.html

Rate This Session

www.esri.com/RateMyDevSummitSession

esri

Understanding our world.