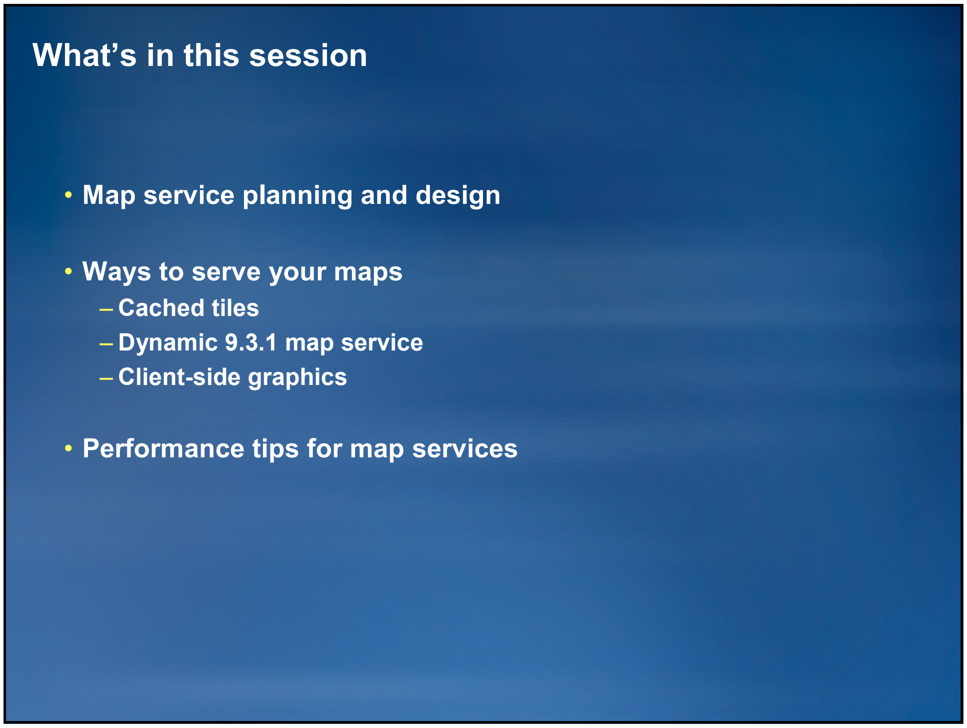
Best Practices for Designing Effective Map Services

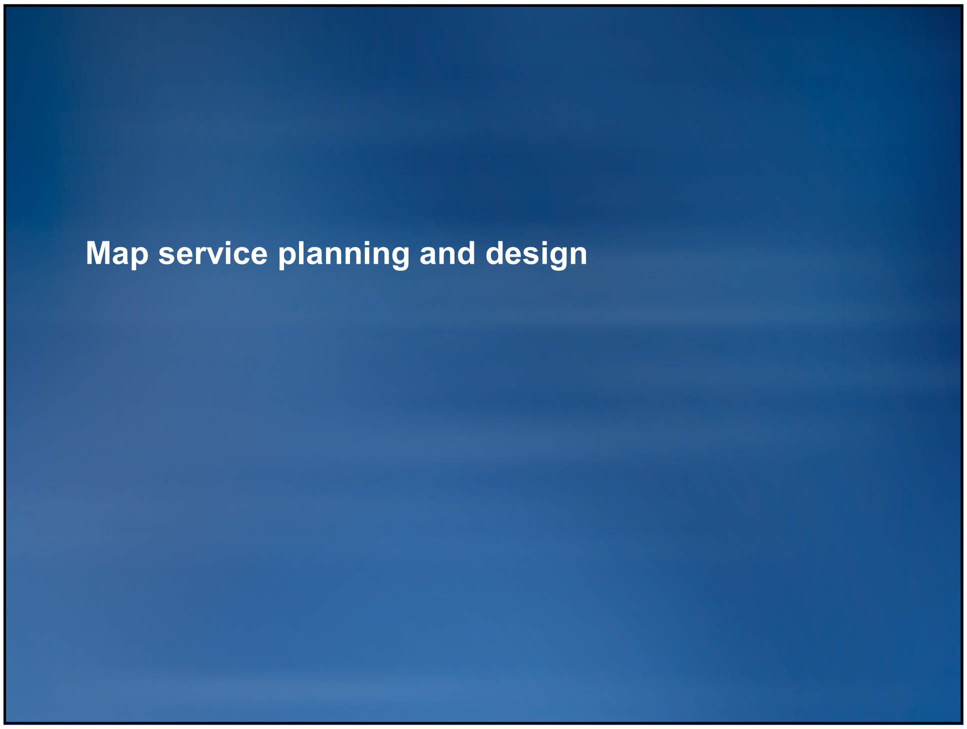
Rex Hansen

Lead Product Engineer - .NET Server
ESRI Redlands

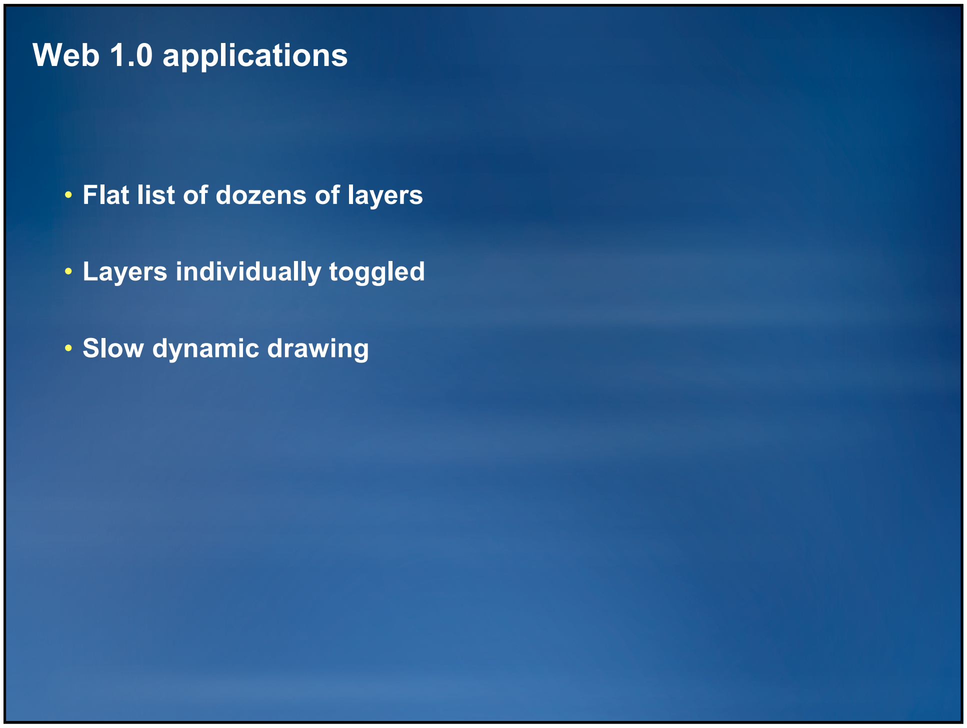


What’s in this session

- Map service planning and design
- Ways to serve your maps
  - Cached tiles
  - Dynamic 9.3.1 map service
  - Client-side graphics
- Performance tips for map services

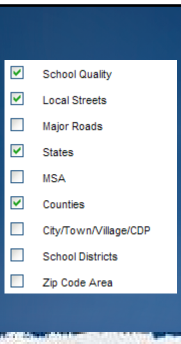


Map service planning and design

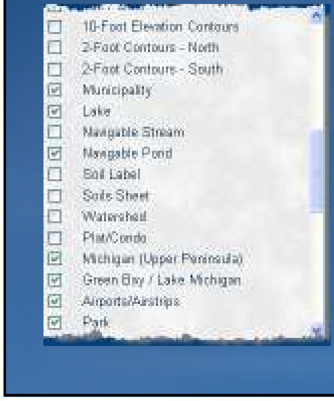


Web 1.0 applications

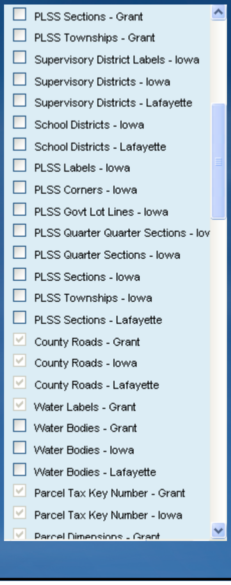
- Flat list of dozens of layers
- Layers individually toggled
- Slow dynamic drawing



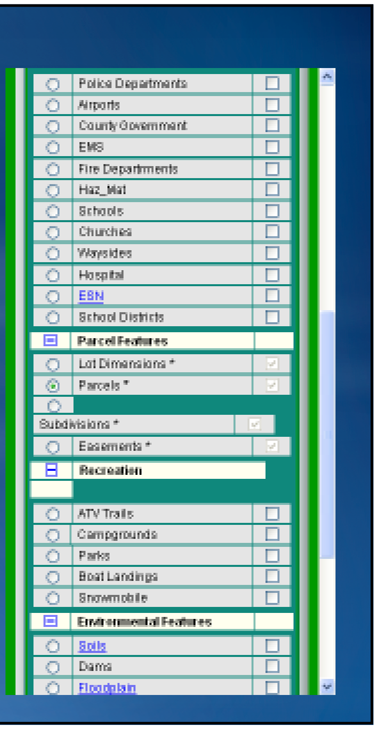
- ☑ School Quality
- ☑ Local Streets
- □ Major Roads
- ☑ States
- □ MSA
- ☑ Counties
- □ City/Town/Village/CDP
- □ School Districts
- □ Zip Code Area



- □ 10-Foot Elevation Contours
- □ 2-Foot Contours - North
- □ 2-Foot Contours - South
- ☑ Municipality
- ☑ Lake
- □ Navigable Stream
- ☑ Navigable Pond
- □ Soil Label
- □ Soils Sheet
- □ Watershed
- □ Plat/Condo
- ☑ Michigan (Upper Peninsula)
- ☑ Green Bay / Lake Michigan
- ☑ Airports/Airstrips
- ☑ Park



- □ PLSS Sections - Grant
- □ PLSS Townships - Grant
- □ Supervisory District Labels - Iowa
- □ Supervisory Districts - Iowa
- □ Supervisory Districts - Lafayette
- □ School Districts - Iowa
- □ School Districts - Lafayette
- □ PLSS Labels - Iowa
- □ PLSS Corners - Iowa
- □ PLSS Govt Lot Lines - Iowa
- □ PLSS Quarter Quarter Sections - Iov
- □ PLSS Quarter Sections - Iowa
- □ PLSS Sections - Iowa
- □ PLSS Townships - Iowa
- □ PLSS Sections - Lafayette
- ☑ County Roads - Grant
- ☑ County Roads - Iowa
- ☑ County Roads - Lafayette
- ☑ Water Labels - Grant
- □ Water Bodies - Grant
- □ Water Bodies - Iowa
- □ Water Bodies - Lafayette
- ☑ Parcel Tax Key Number - Grant
- ☑ Parcel Tax Key Number - Iowa
- ☑ Parcel Dimensions - Grant



○	Police Departments	□
○	Airports	□
○	County Government	□
○	EMS	□
○	Fire Departments	□
○	Haz_Mat	□
○	Schools	□
○	Churches	□
○	Waysides	□
○	Hospital	□
○	EBN	□
○	School Districts	□
⊟	**Parcel Features**	
○	Lot Dimensions *	☑
●	Parcels *	☑
○		
Subdivisions *		☑
○	Easements *	☑
⊟	**Recreation**	
○	ATV Trails	□
○	Campgrounds	□
○	Parks	□
○	Boat Landings	□
○	Snowmobile	□
⊟	**Environmental Features**	
○	Soils	□
○	Dams	□
○	Floodplain	□

Web 2.0 maps

- Base map + operational layers in logical groups
- Small number of layers to turn on and off
- High-performance blend of
  - Cached tiles
  - Dynamic layers
  - Client-side graphics

HIGH PERFORMANCE BLEND $9

SOY POWER
soy milk, banana, mango, pineapple, vanilla myoplex

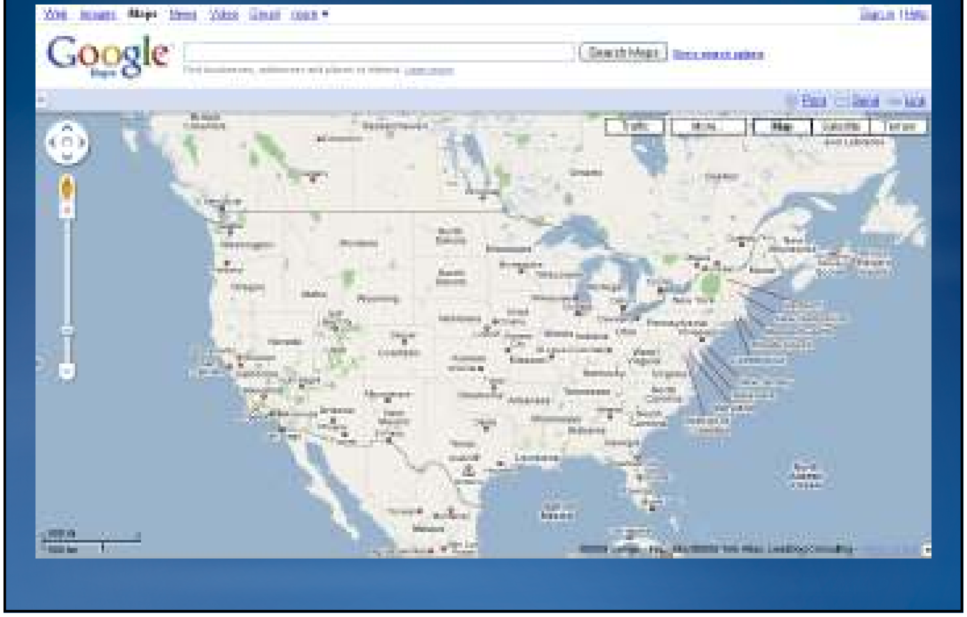
PROTEIN PUNCH
skim milk, banana, chocolate myoplex

MONA-VIE 2oz SHOT $7
açai, wolfberry, purple grape

SUPPLEMENT SHOT $2
ginseng gingko echinacea
lecithin aloe

http://www.brguestrestaurants.com/restaurants/menus/BF_BREAKFAST_1.16.09.pdf

Case study: Google Maps



Google Maps base maps

“Map”

- Highways
- Streets
- Ferries
- Railroads
- Transit centers
- Cities
- Parks
- Military reservations
- Municipal boundaries
- Lakes
- Rivers
- Golf courses
- Hospitals
- Shopping centers
- Airports
- Colleges
- Cemeteries
- Amusement parks

“Terrain”

- Shaded relief
- Vegetation
- Highways
- Streets
- Cities
- Parks
- Military reservations
- Municipal boundaries
- Lakes
- Rivers
- Golf courses
- Hospitals
- Shopping centers
- Airports
- Colleges
- Cemeteries
- Amusement parks

“Satellite”

- Imagery

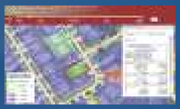
Google Maps operational layers

- Street overlay for imagery
- Traffic
- Photos
- Videos
- Wikipedia
- StreetView coverage
- Transit info (query on click)

Some ArcGIS Server examples

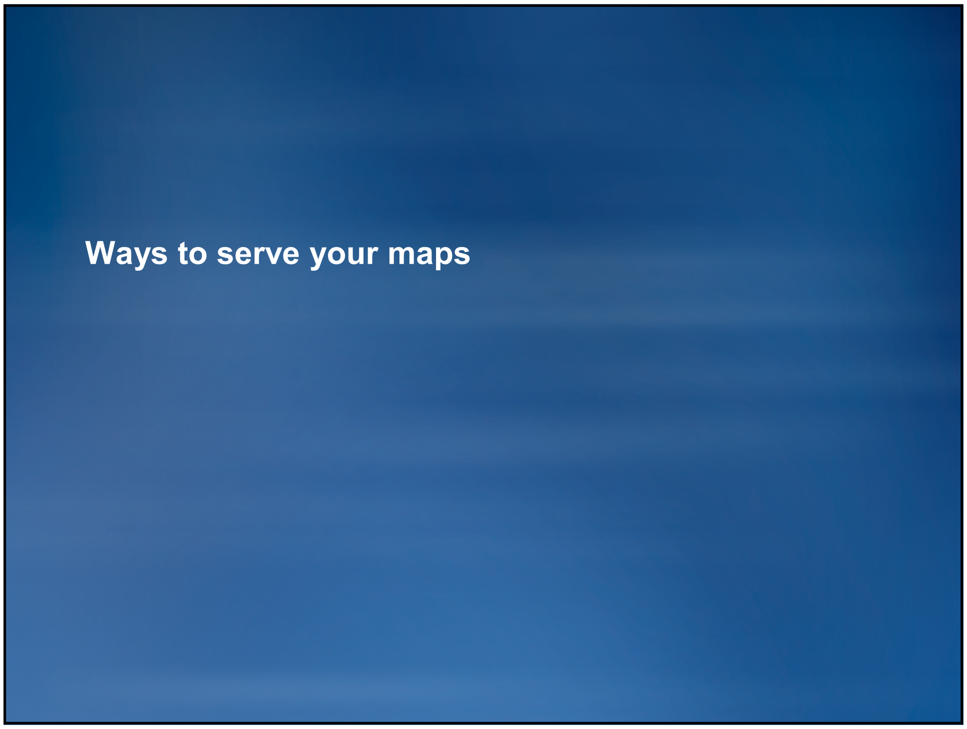
- Orange County Property Appraiser Map

- Palm Beach County Property Map

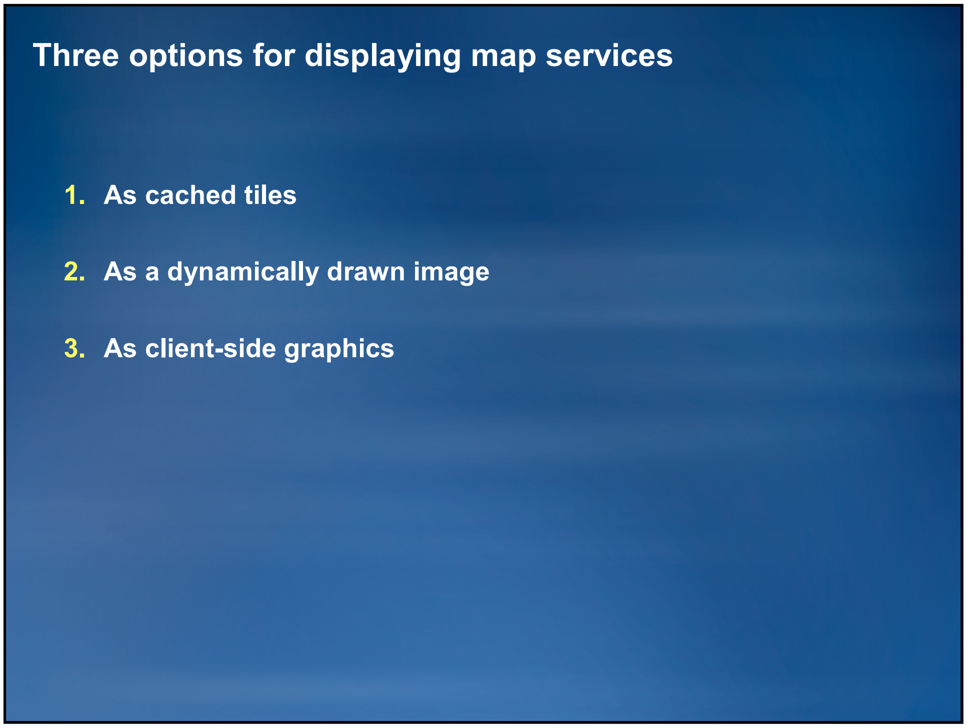


- Solar Boston

- City of Greeley Property Information Map



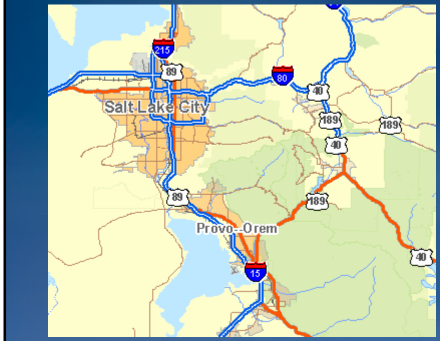
Ways to serve your maps

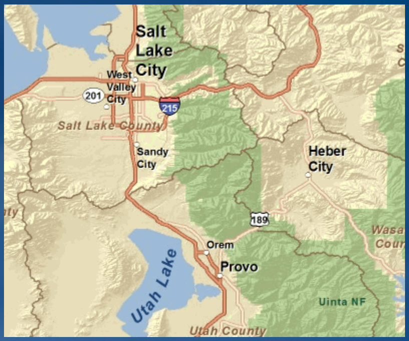


Three options for displaying map services

1. As cached tiles
2. As a dynamically drawn image
3. As client-side graphics

Internet users expect the performance of cached maps





What users expected 10 years ago

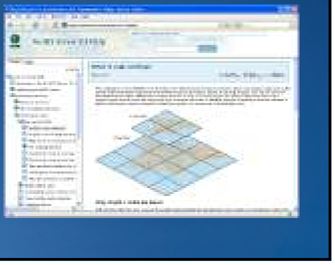
- Dynamically drawn map
- Slow
- Compromised cartography

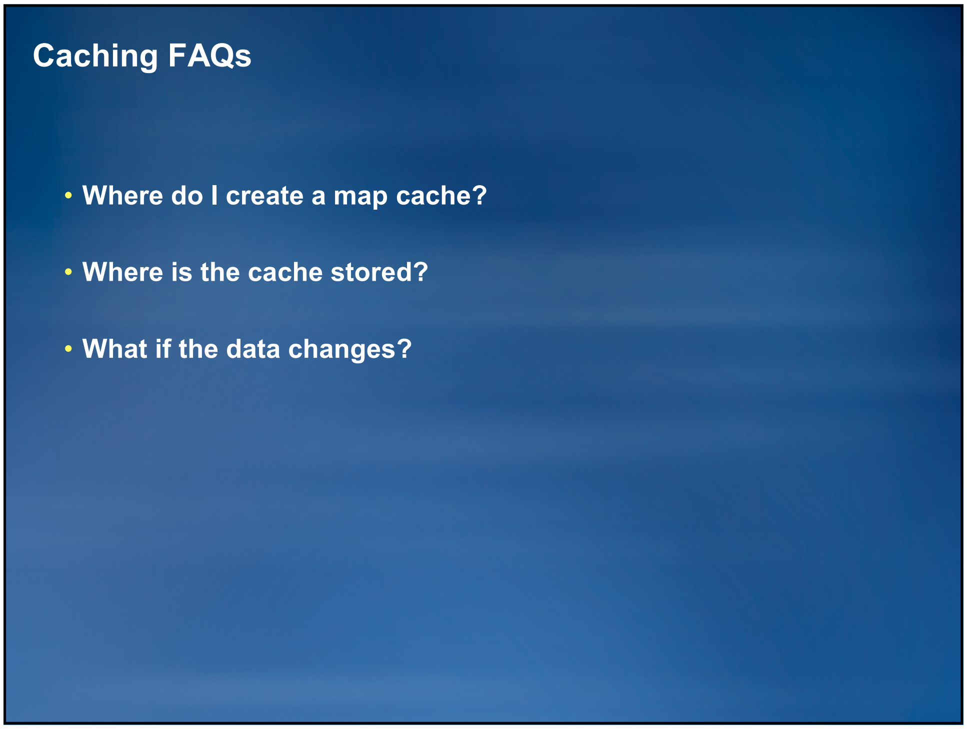
What users expect today

- Cached map
- Fast
- Beautiful cartography

Cached tiles

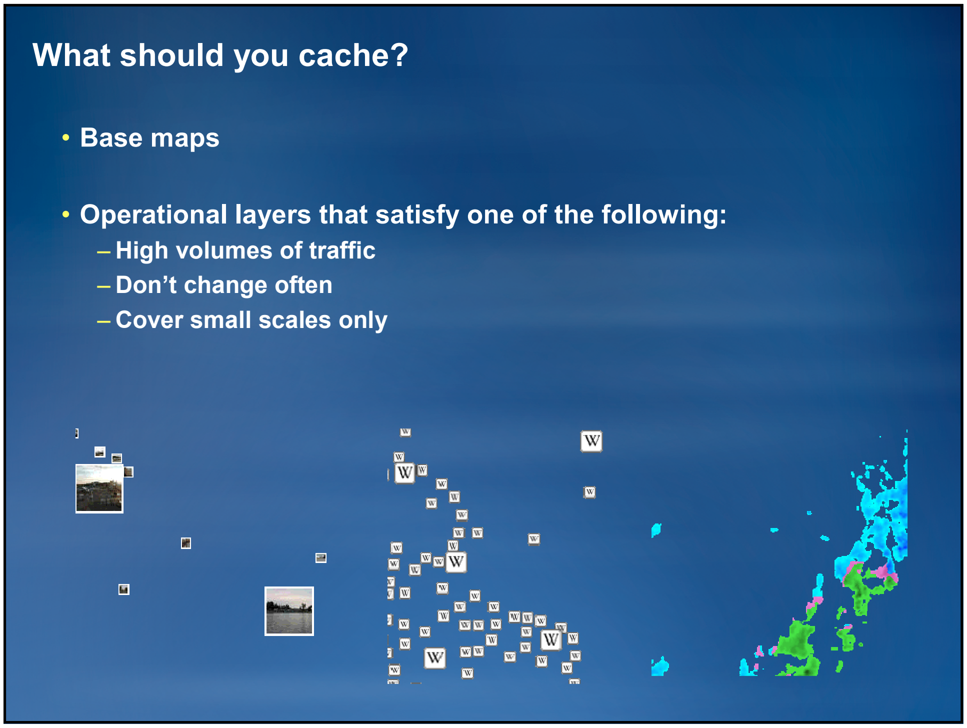
- Pre-draw map tiles and serve them to clients
- Best performance and scalability
- Standard for online maps (Google, VE, Yahoo, etc)
- Requires you to create and maintain cache





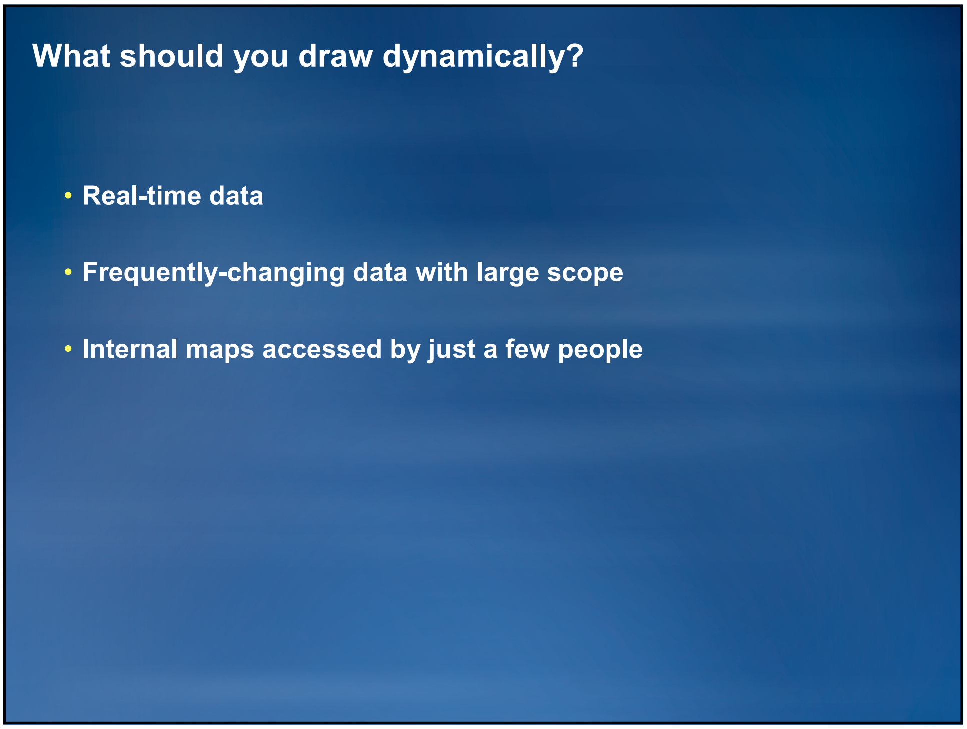
Caching FAQs

- Where do I create a map cache?
- Where is the cache stored?
- What if the data changes?



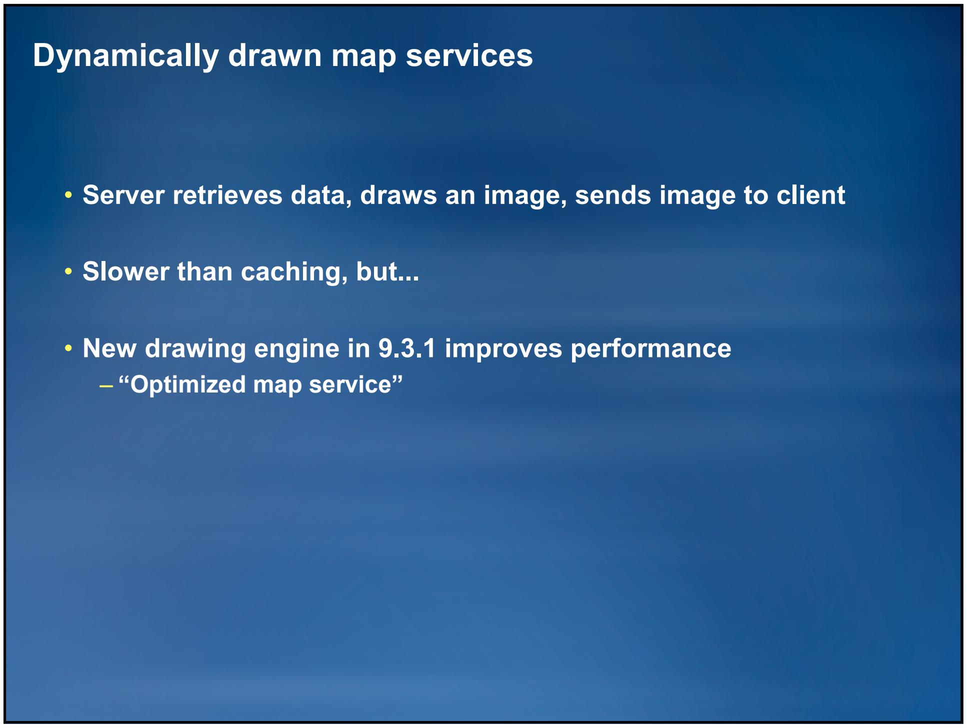
What should you cache?

- Base maps
- Operational layers that satisfy one of the following:
  - High volumes of traffic
  - Don’t change often
  - Cover small scales only



What should you draw dynamically?

- Real-time data
- Frequently-changing data with large scope
- Internal maps accessed by just a few people

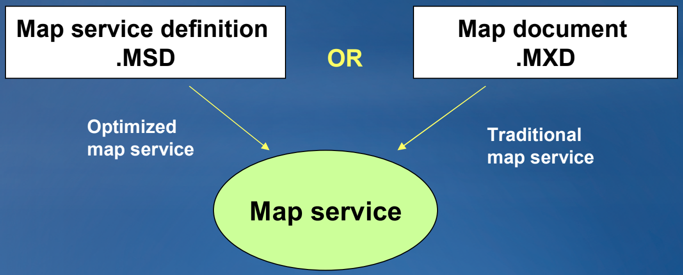


Dynamically drawn map services

- Server retrieves data, draws an image, sends image to client
- Slower than caching, but...
- New drawing engine in 9.3.1 improves performance
  - “Optimized map service”

Map services at 9.3.1

- 9.3.1 map services optionally use new, faster drawing engine
- New file type: Map service definition (MSD)



Map service definition .MSD

OR

Map document .MXD

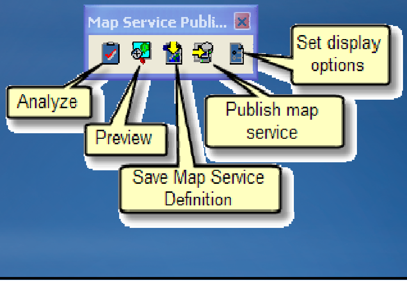
Optimized map service

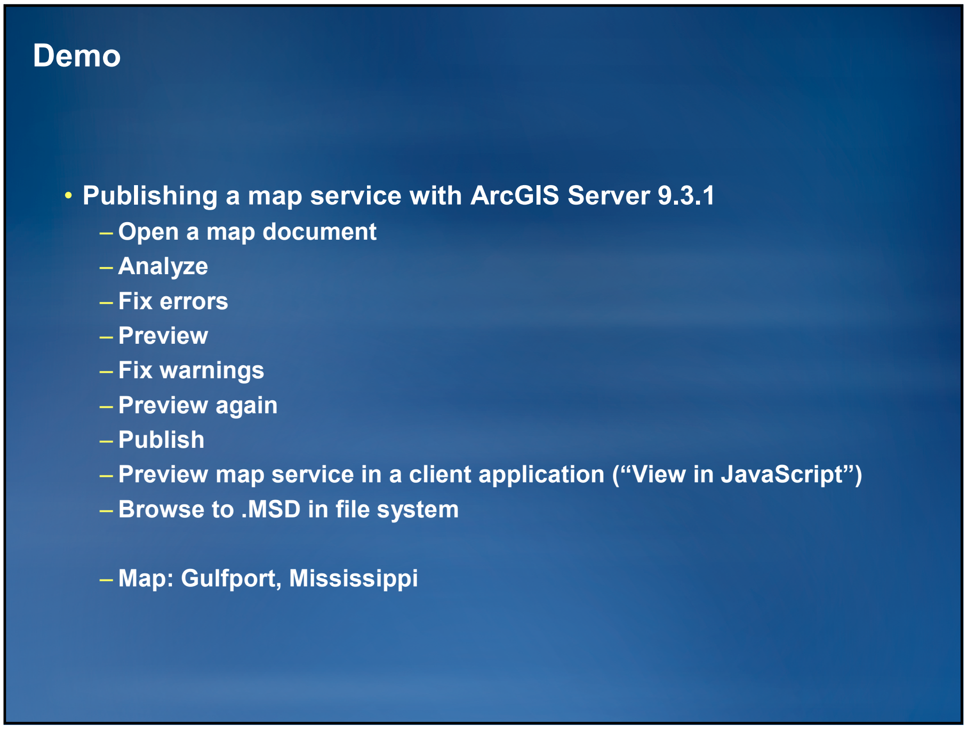
Traditional map service

Map service

Optimized map service

- Faster dynamic drawing than ArcIMS
- Supports a subset of the most common layer types
- Prepare and publish with new Map Services Publishing Toolbar in ArcMap

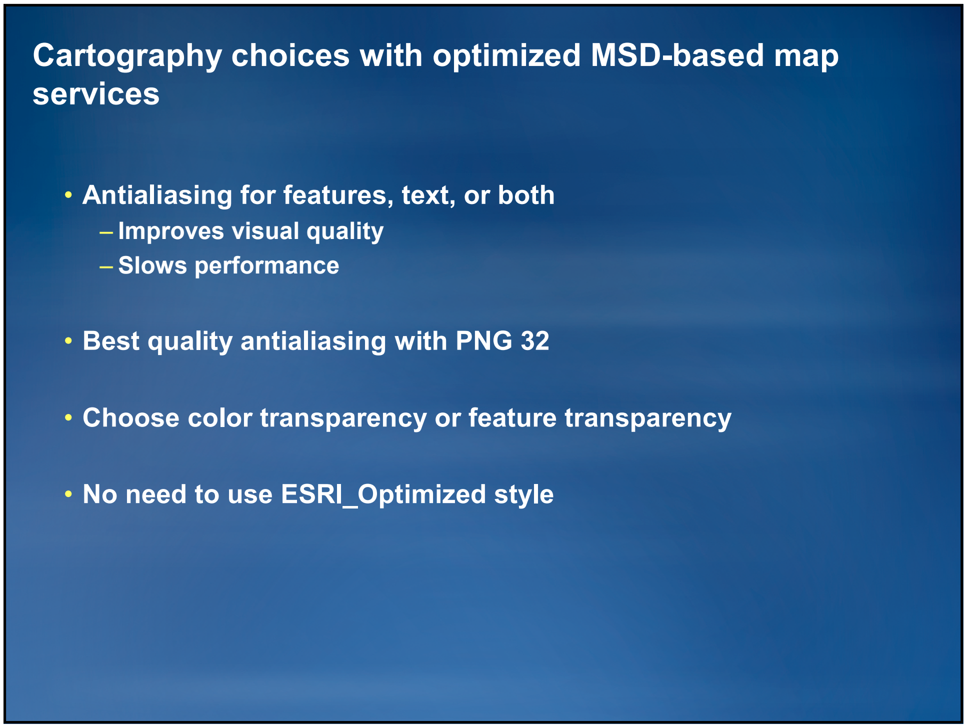




Demo

- Publishing a map service with ArcGIS Server 9.3.1
  - Open a map document
  - Analyze
  - Fix errors
  - Preview
  - Fix warnings
  - Preview again
  - Publish
  - Preview map service in a client application (“View in JavaScript”)
  - Browse to .MSD in file system

  - Map: Gulfport, Mississippi



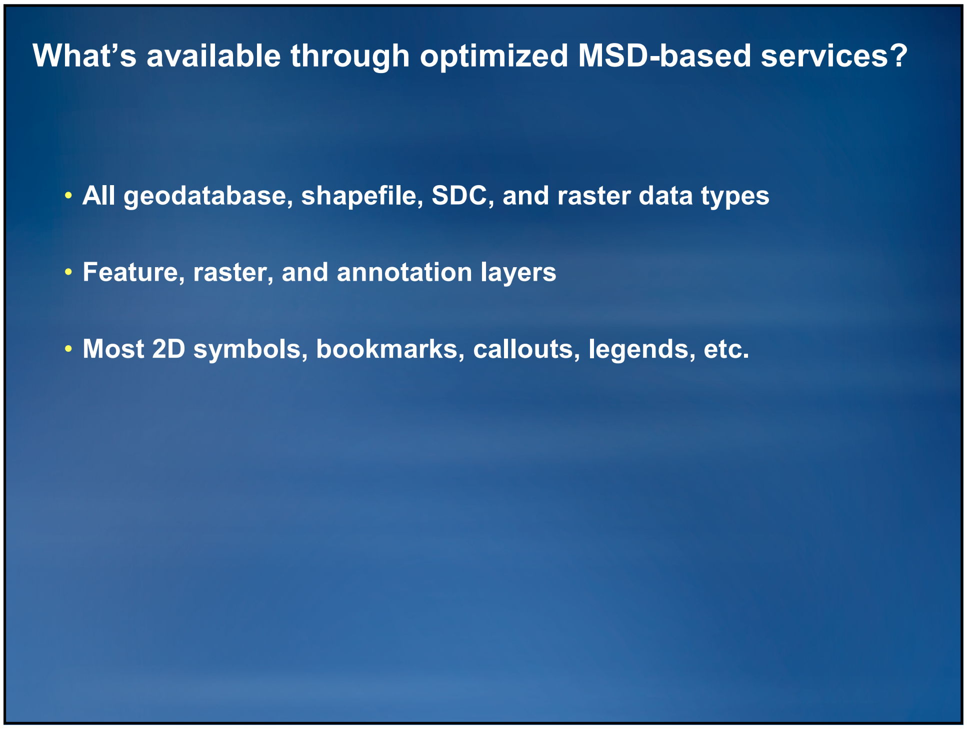
Cartography choices with optimized MSD-based map services

- Antialiasing for features, text, or both
  - Improves visual quality
  - Slows performance

- Best quality antialiasing with PNG 32

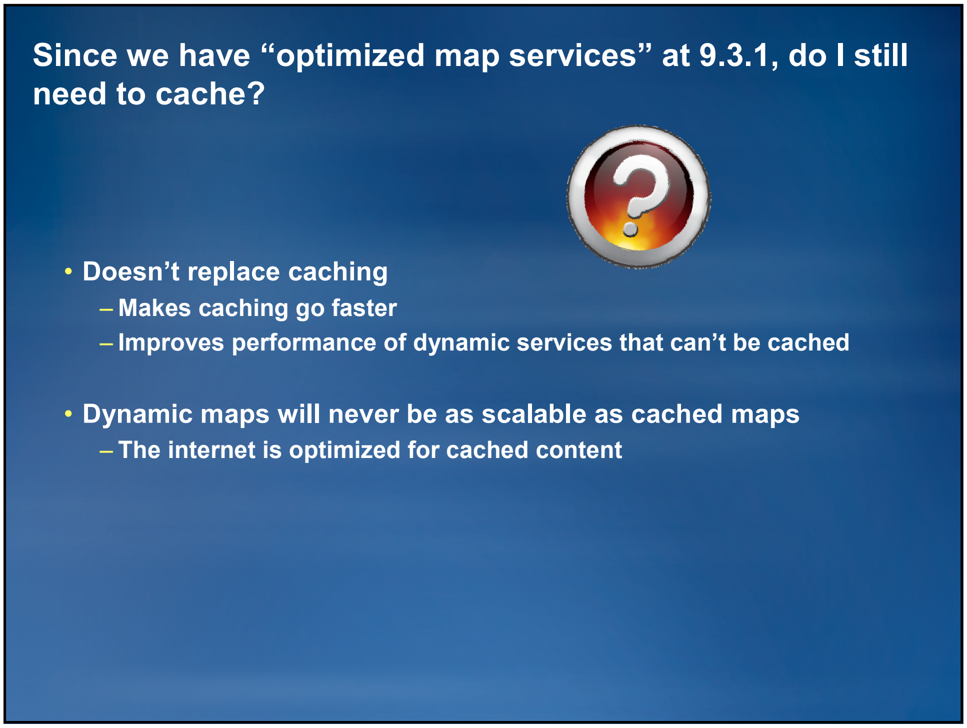
- Choose color transparency or feature transparency

- No need to use ESRI_Optimized style



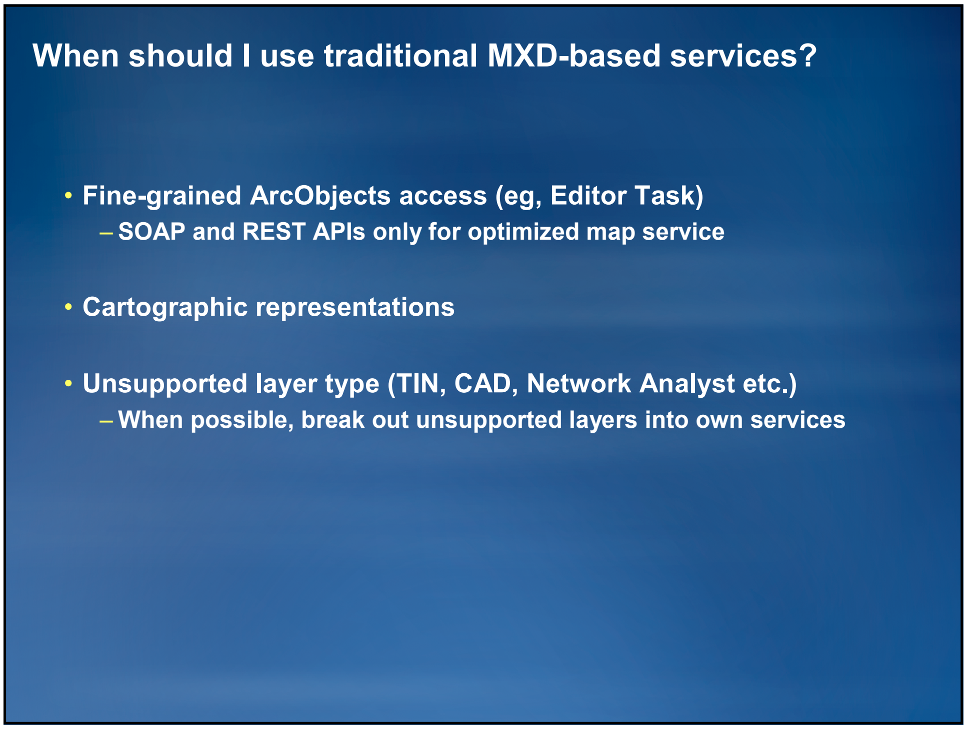
What’s available through optimized MSD-based services?

- All geodatabase, shapefile, SDC, and raster data types
- Feature, raster, and annotation layers
- Most 2D symbols, bookmarks, callouts, legends, etc.



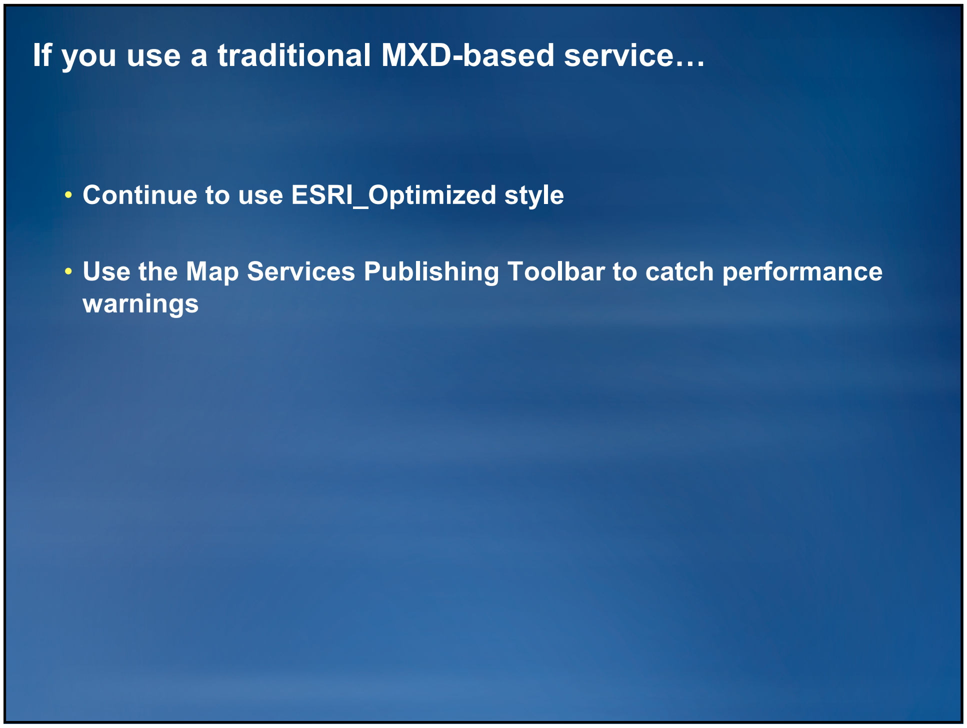
Since we have “optimized map services” at 9.3.1, do I still need to cache?

- Doesn’t replace caching
  - Makes caching go faster
  - Improves performance of dynamic services that can’t be cached
- Dynamic maps will never be as scalable as cached maps
  - The internet is optimized for cached content



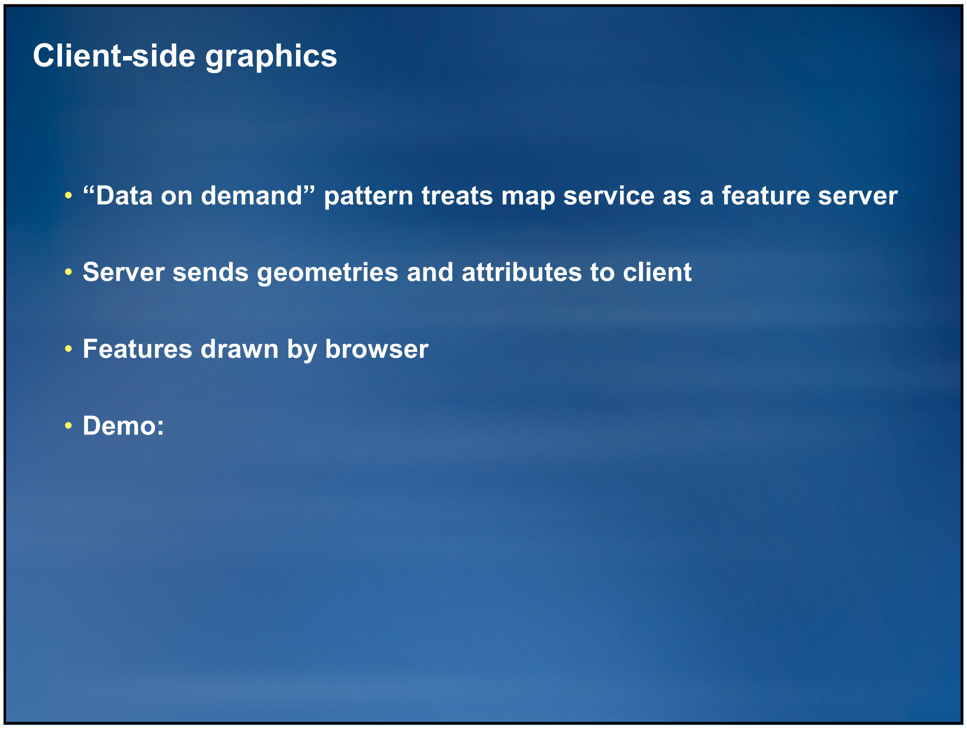
When should I use traditional MXD-based services?

- Fine-grained ArcObjects access (eg, Editor Task)
  - SOAP and REST APIs only for optimized map service
- Cartographic representations
- Unsupported layer type (TIN, CAD, Network Analyst etc.)
  - When possible, break out unsupported layers into own services



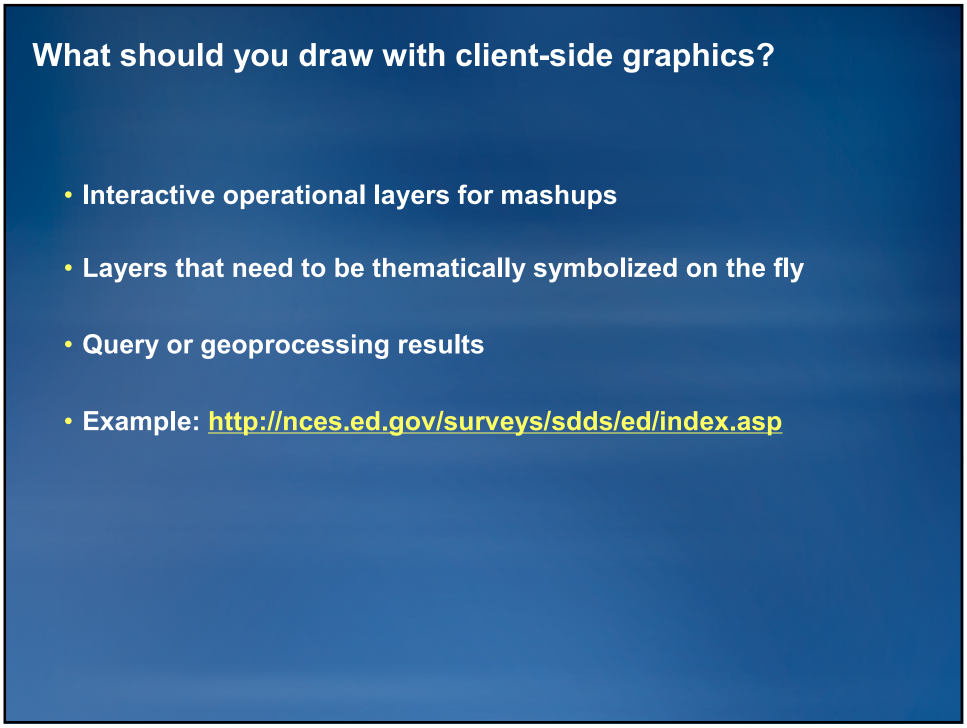
If you use a traditional MXD-based service…

- Continue to use ESRI_Optimized style
- Use the Map Services Publishing Toolbar to catch performance warnings



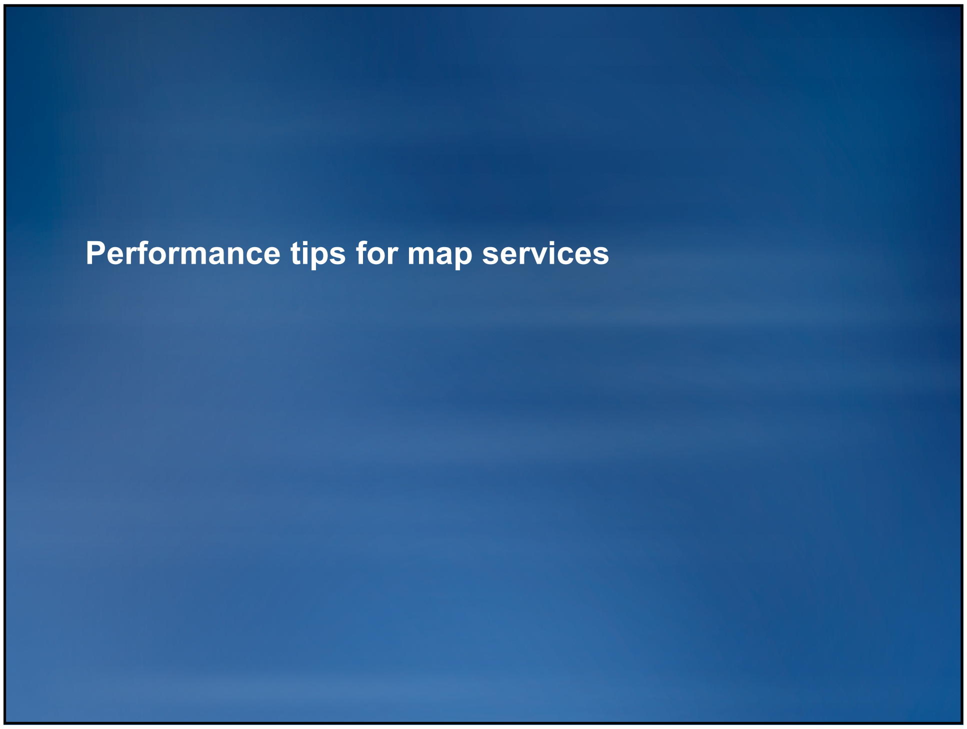
Client-side graphics

- “Data on demand” pattern treats map service as a feature server
- Server sends geometries and attributes to client
- Features drawn by browser
- Demo:

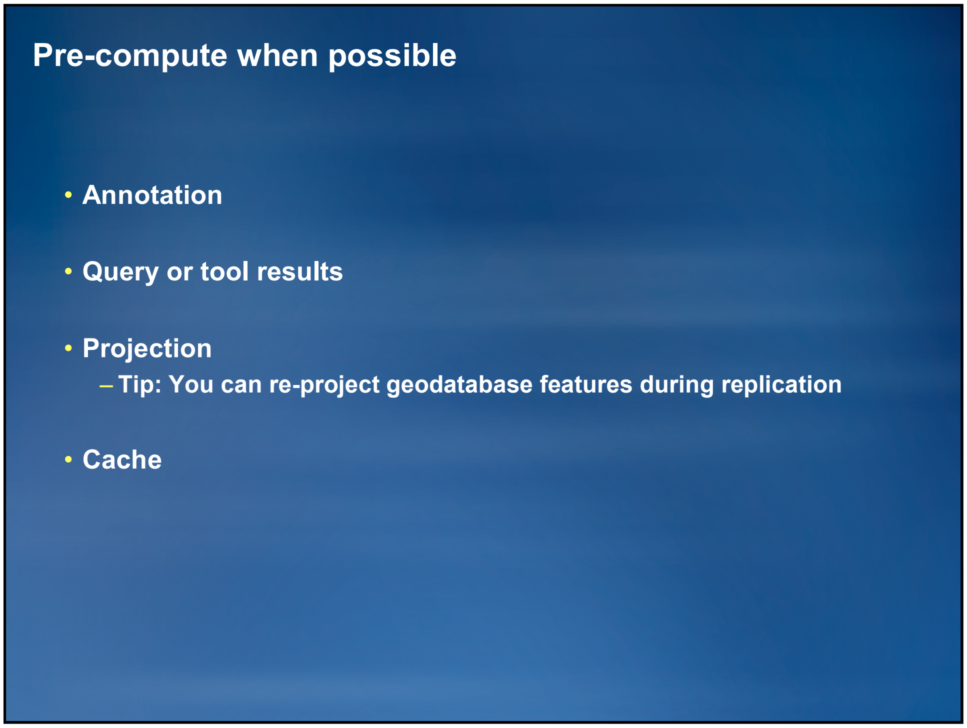


What should you draw with client-side graphics?

- Interactive operational layers for mashups
- Layers that need to be thematically symbolized on the fly
- Query or geoprocessing results
- Example: http://nces.ed.gov/surveys/sdds/ed/index.asp

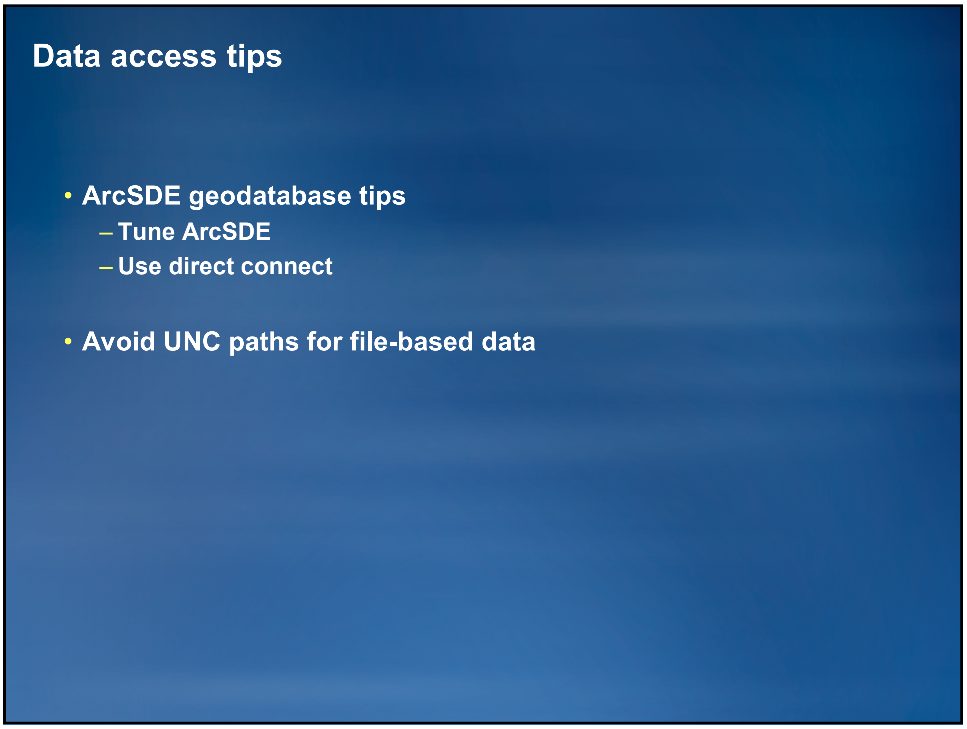


Performance tips for map services



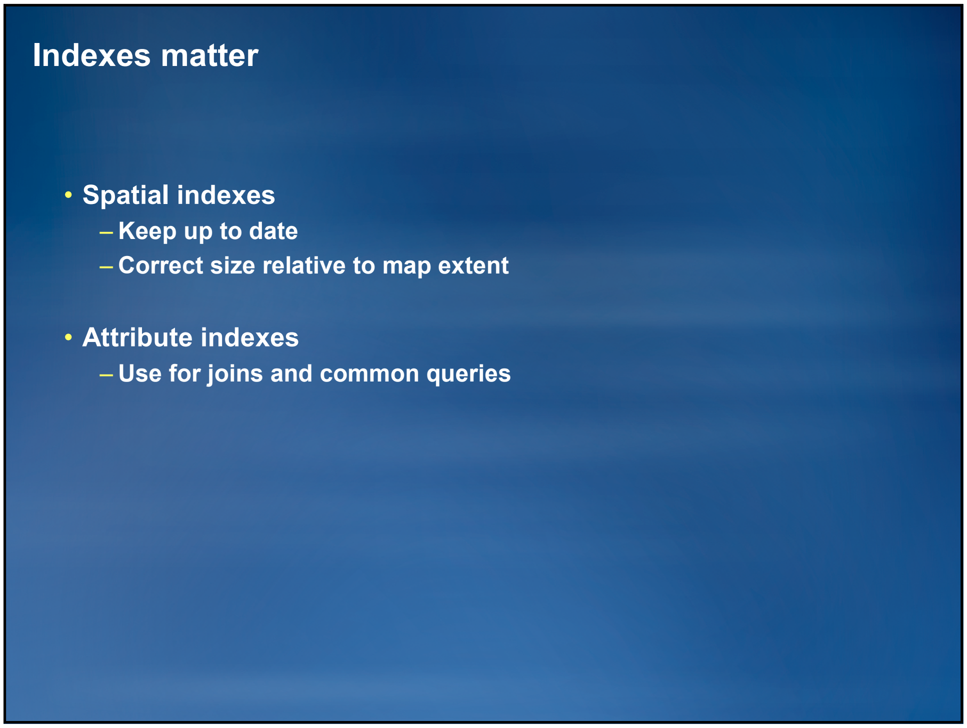
Pre-compute when possible

- Annotation
- Query or tool results
- Projection
  - Tip: You can re-project geodatabase features during replication
- Cache



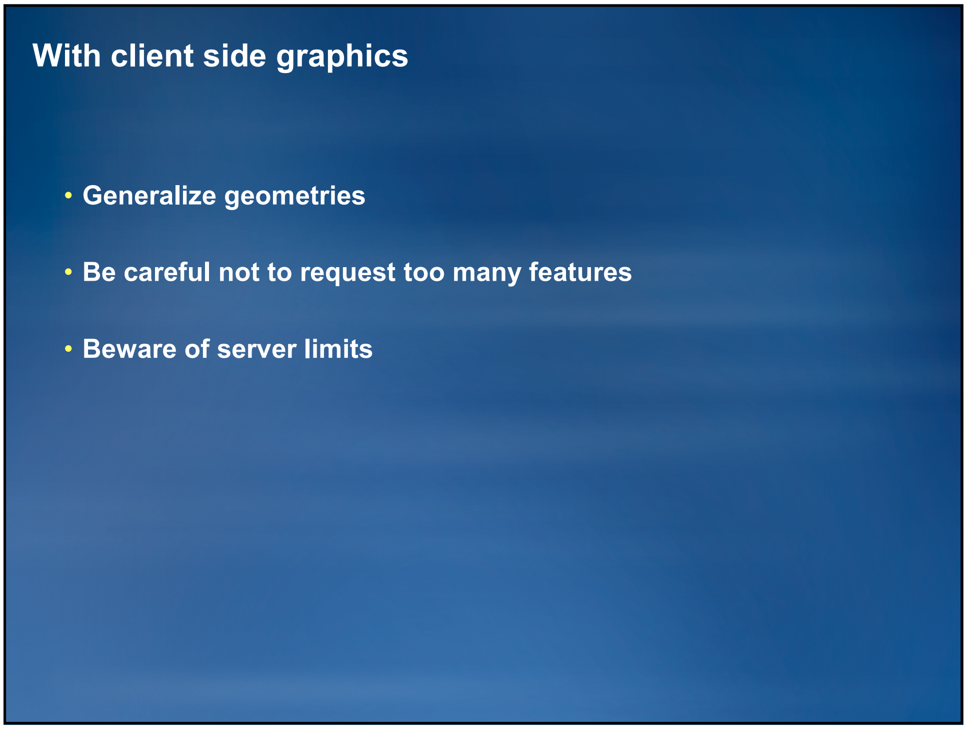
Data access tips

- ArcSDE geodatabase tips
  - Tune ArcSDE
  - Use direct connect
- Avoid UNC paths for file-based data



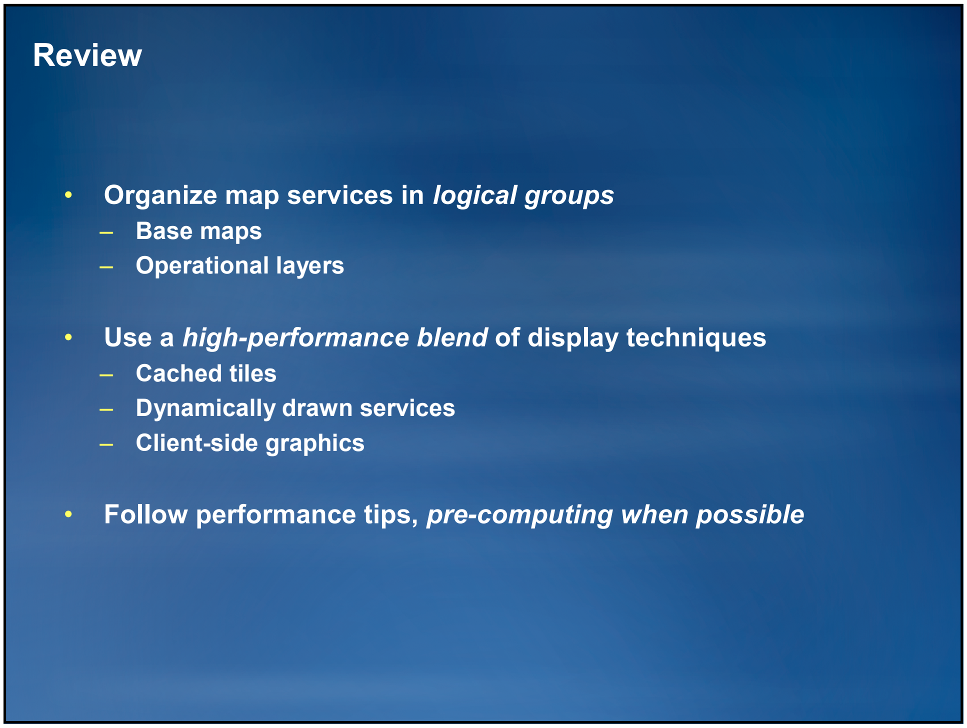
Indexes matter

- Spatial indexes
  - Keep up to date
  - Correct size relative to map extent
- Attribute indexes
  - Use for joins and common queries



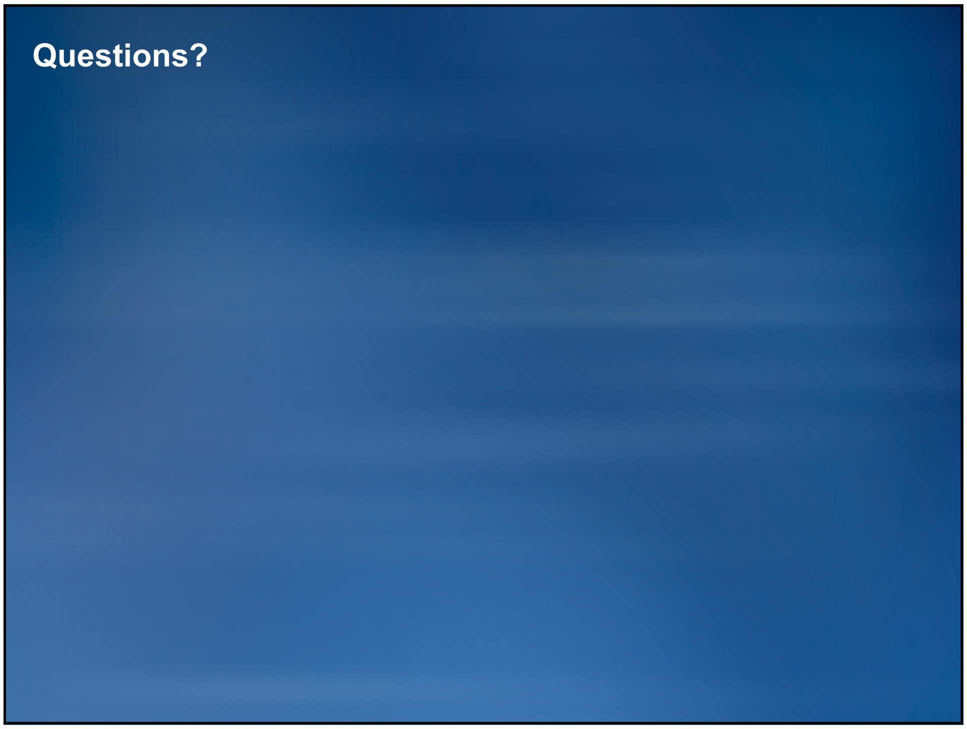
With client side graphics

- Generalize geometries
- Be careful not to request too many features
- Beware of server limits



Review

- Organize map services in *logical groups*
  - Base maps
  - Operational layers
- Use a *high-performance blend* of display techniques
  - Cached tiles
  - Dynamically drawn services
  - Client-side graphics
- Follow performance tips, *pre-computing when possible*



Questions?