

ESRI Developer Summit

March 22–25, 2010
Palm Springs, CA

Editing GeoDatabases over the Web using ArcGIS Server

Roop Goyal, Sud Menon



Schedule

Presentation: ~60 minutes.

Q&A: ~15 Minutes



Please ensure that your cell phones are switched off.
Thanks!

Outline

- Editing over the Web
- Feature Service Architecture and APIs
- Feature Service Information Model
- Authoring and Publishing a Feature Service
- Other GeoDatabase Editing Solutions
- Conclusions

Editing over the Web

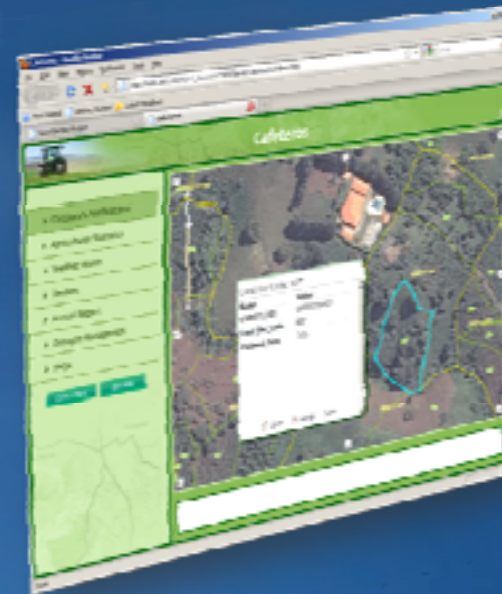
- Allows people to easily contribute and edit geographic information :
 - Volunteered Geographic Information (VGI)
 - User Generated Content (UGC)
 - Crowd sourcing of information
- Over the Internet using a web-browser or mobile device
- Applications include :
 - 311 applications to gather citizen reported incidents
 - Citizen Science
 - Emergency Response
 - ...

Editing over the Web.. (contd.)

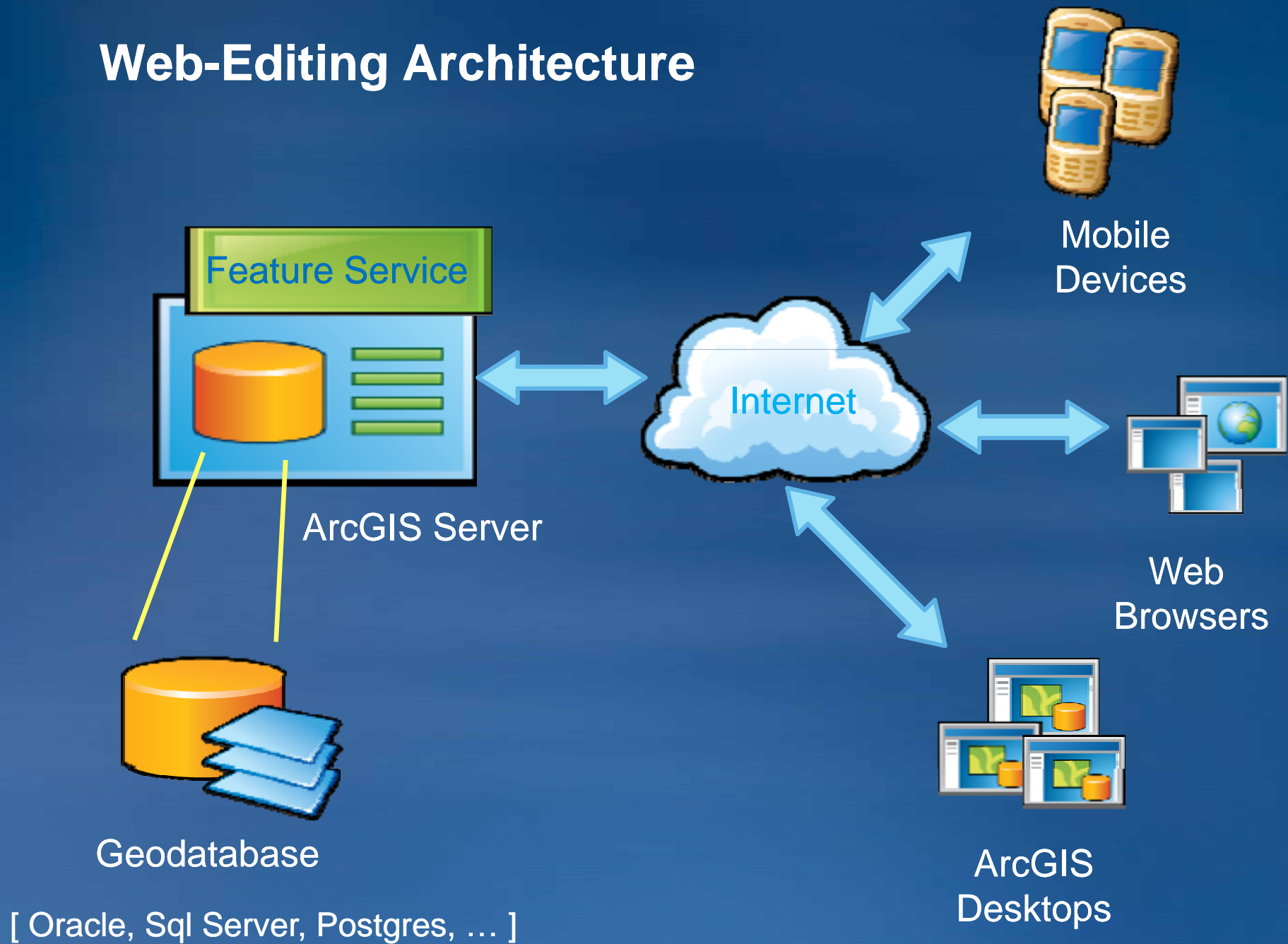
- VGI Applications
 - Sketch point, line, and polygon features on a map
 - Provide an appropriate Description
 - Attach photos, videos
- Structured Editing
 - Authoritative Information Models
 - Edit features with
 - Predefined attribute schema
 - Restrictions on the attribute values
- Feature Service (new at 10) supports both types of information, including combinations

ArcGIS Server 10 - Web Editing

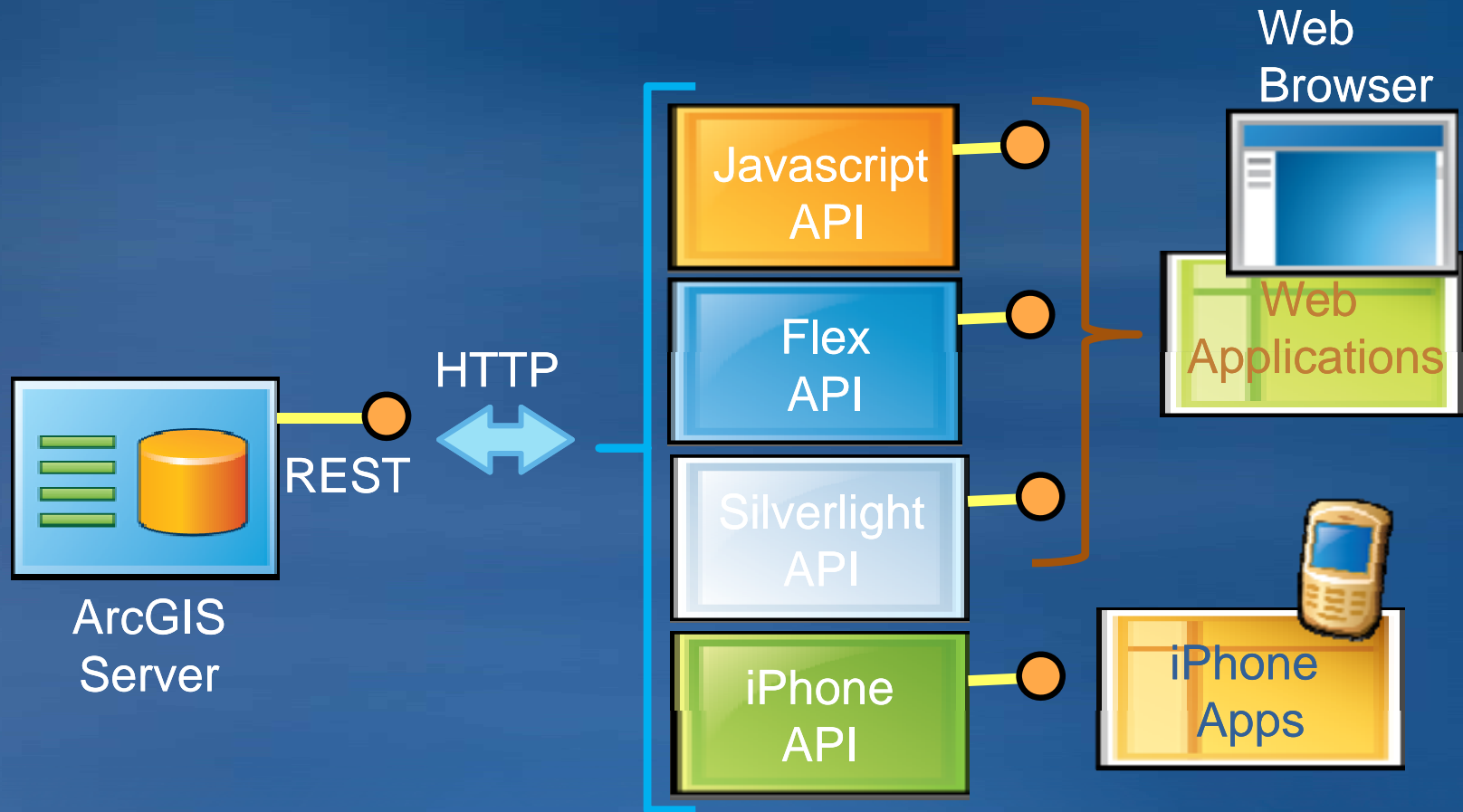
- Extends editing to all web clients
 - Javascript, Flex, Silverlight, iPhone, ...
- Ability to build light and intuitive user experiences for web users
- Scalable editing for large user communities



Web-Editing Architecture



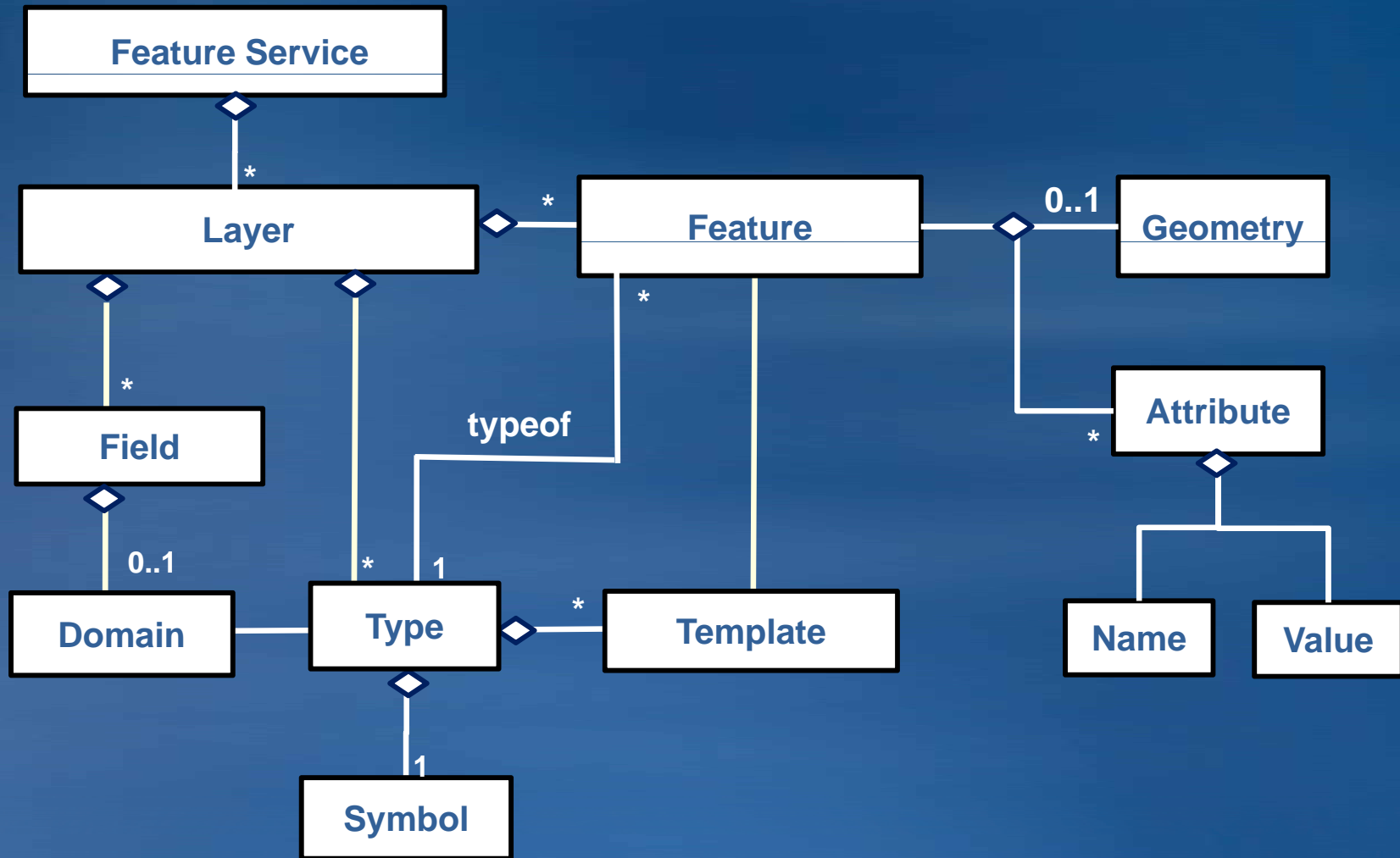
APIs for Web Editing



Feature Service Information Model

- Features are used to represent real world entities such as schools, parcels and valves.
- Features include symbology, attributes, relationships and attachments.
- Features are organized into Layers and Types.
- Each Layer has a set of Types.
- Each Type has a set of prototypical instances called Templates.

Feature Service Information Model



Demo 1: Wild Fire Demo

- Quick intro of the whole process
- <http://jaladhi/ArcGIS/rest/services/Wildfire/FeatureServer>
 - Re-run the Flex Sample

Demo 2: REST resource view of Layers [with Types, Templates, Domains]

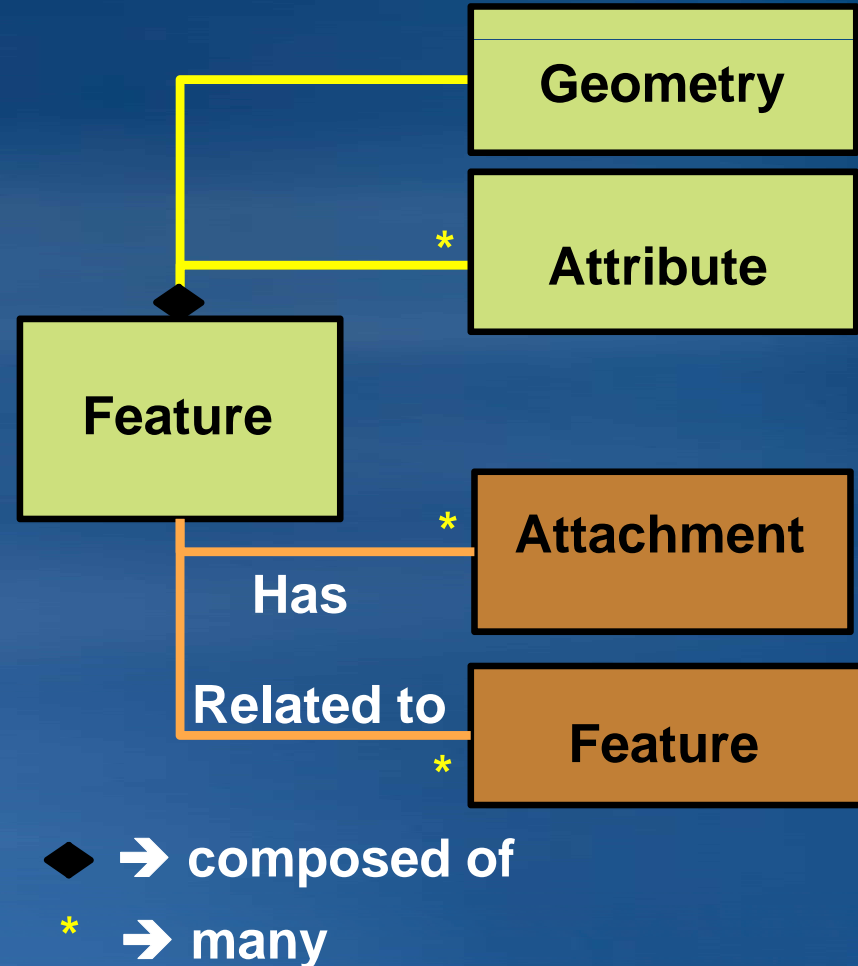
- Homeland operations
- <http://jaladhi/ArcGIS/rest/services/HomelandOperations/FeatureServer/2>
- <http://jaladhi/ArcGIS/SDK/REST/index.html?catalog.html>

Operations on a Layer

- Query
- Query Related Records
- Add Features
- Delete Features
- Update Features
- Apply Edits

Feature

- Geometry
- Attributes
- Time attributes are supported
 - Time metadata is in Layer
- The feature can have
 - Attachments
 - Relationships with the features in other layers/ tables



Attachments

- Attachments can be used to attach media and documents to a feature
 - Picture, video, pdf , ...
 - Stored in the geodatabase in an associated table

- A feature can have multiple attachments.
 - Attachments can be added and deleted
 - When the object is deleted, attachments are automatically removed

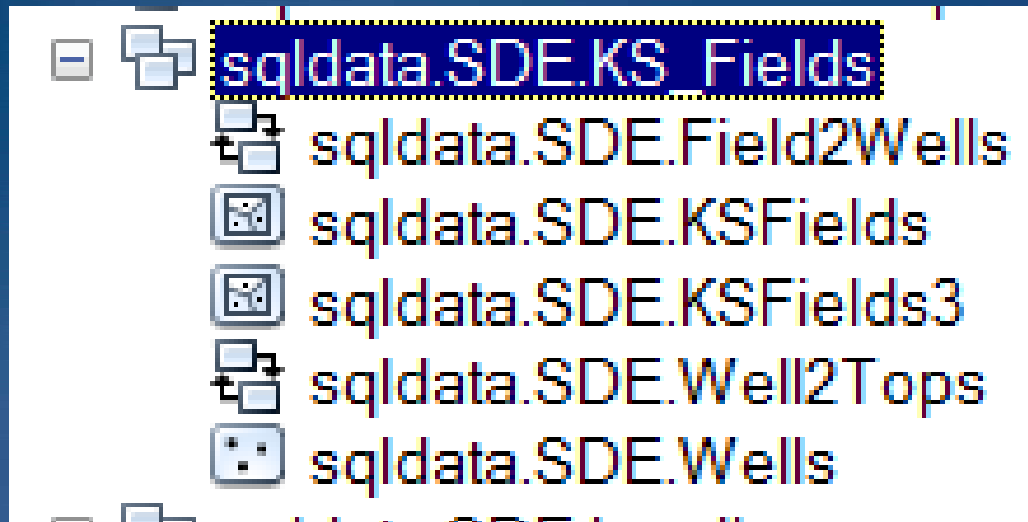
Attachments (contd.)

- A feature has a list of lightweight AttachmentInfos.
- Applications can fetch AttachmentInfos to obtain :
 - names, types and sizes of attachments to present to the user
- Full Attachment can be downloaded if the user decides to view or download it

Relationships

- A feature may be related to features in other layers
 - Related features are queried from the **source** layer by supplying the ids of the source features
 - The returned features belong to a **target** layer
- Information on available relationships is available at the layer/table level
- Relationships are maintained in the geodatabase
- Deleting features deletes applicable relationships

Relationships Example: View In GeoDatabase



Relationship Class Properties

General Rules

Name: sqldata.SDE.Field2Wells

Type: Simple

Cardinality: 1 - M

Notification: None (no messages propagated)

Origin Table/Feature Class

Name: sqldata.SDE.KSFields3

Primary Key: FIELD_KIDN

Foreign Key: FIELD_KID

Destination Table/Feature Class

Name: sqldata.SDE.Wells

Labels

Forward: Field 2 Wells

Backward: Wells 2 Field

Relationships Example: View in REST Services

ArcGIS Services Directory

[Home](#) > [OilWells \(FeatureServer\)](#) > [sqldata.SDE.Wells](#)

[Help](#) | [API Reference](#)

Feature Layer: sqldata.SDE.Wells (ID: 0)

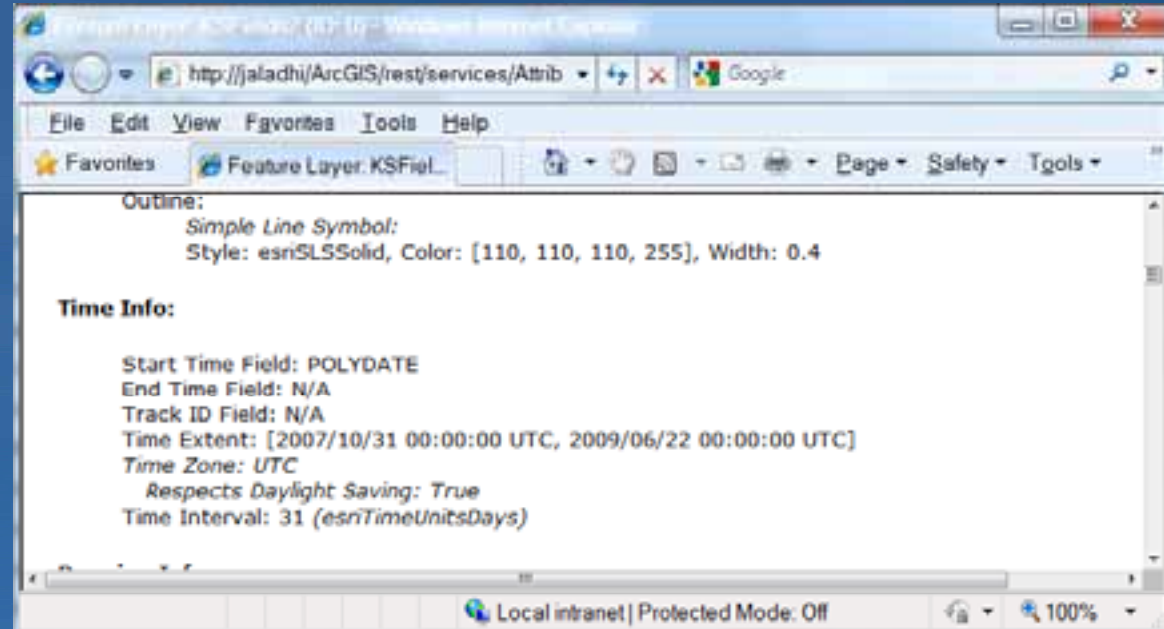
Relationships:

- Well 2 Tops (3) -- *Related To* [sqldata.SDE.KSTOPS](#) (2)
- Wells 2 Field (2) -- *Related To* [sqldata.SDE.KSFields3](#) (1)

Relationship Labels are used to name the Relationships in Feature Service

Time

- Time information is available in Layer properties
 - Start and End field, TimeExtent, TimeReference...
- Can query based on time .
- Can set time for the time fields and add/ update time aware features.



Table

- A Service has Tables, just like it has Layers
- A single layer / table resource in REST
 - Table does not have symbols.
 - Features/Objects queried out of aTable don't have geometry.
- Tables can be queried for the objects
- Edits can be made via the standard operations
- Supports Attachments, Relationship, and Time

Demo 3: Attachment, Relationship, Table, and Time

- Building Attachments,
 - Show the Attachment in ArcCatalog
 - REST view and
 - Flex Sample
- Oil Wells relationships with Fields and TOPS table
 - Relationship in ArcCatalog
 - Relationship on the REST page
 - Online Flex Sample: FeatureLayer/ Relationship samples in
- Earthquake Layer for Time
 - Layer's Time properties in ArcMap
 - Time properties in REST resource view
 - Flex Sample for Earthquake animation

Demo 3: Attachments and Relationships

- Building Attachments,
 - Show the Attachment in ArcCatalog
 - REST view and
 - Flex Sample
- Related Record Query on Fields Layer

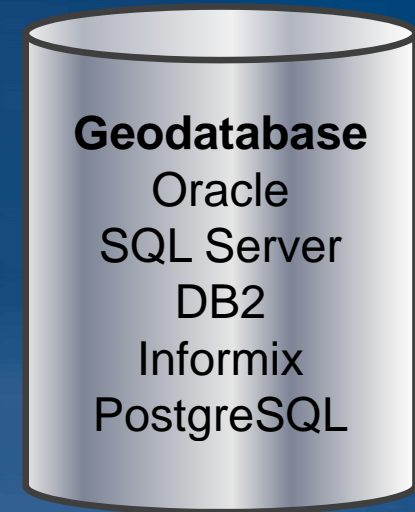
Publishing a Feature Service

- A Feature Service is defined by a Map and the Geodatabase that it references
- Author
 - Author the GeoDatabase
 - Author the Map
- Publish the map, enabling feature access
- Use resulting feature service in web/ mobile apps

Types of Geodatabases

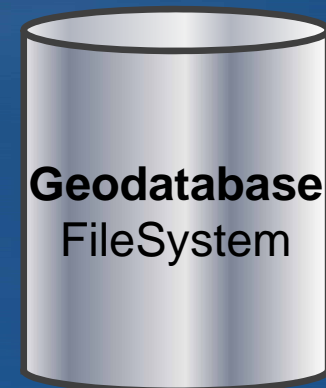
- Enterprise Geodatabases

- Stored in a spatially enabled enterprise RDBMS
- Multiuser editing model
- Extended with versioning , long transactions, historical archiving, replication



- File Geodatabases

- Single user editing model
- Used for GIS Project work
- Supports large local datasets with high performance



Authoring the GeoDatabase

- Connect to an Enterprise GeoDatabase
 - Multi user transactional database
 - Oracle, MS SQL Server, PostgreSQL, Informix, DB2
- Define **Geodatabase Schema** :
 - Create **Point, Line, and Polygon Feature Classes** for the Layers and any non spatial **Tables**
 - To constrain a field's value to valid ones, add **Domains**
 - To model Type specific domains and default values add **Subtypes** to your feature classes
 - To model Relationships, create **Relationship Classes**
 - To model Attachments add **Attachments**.

Authoring the GeoDatabase (contd.)

- Can use pre-existing Geodatabases
- Can start with Pre-existing features

Authoring the Map

- Add Feature Classes and Tables to ArcMap
- Set the field properties for each Layer
 - Visibility
 - Read-Only
 - Aliases
- Define time fields for Layers, if applicable
- Define Types for Layers by setting up a Unique Value Renderer (UVR) on a field with distinct values
 - UVR field will be the TypeID field for the layer
 - UVR field must be the geodatabase subtype field if present

Authoring the Map (contd.)

- If no types (i.e a single type) then define a Simple Renderer for the Layer
- For line FeatureClasses
 - Use simple line symbols,
 - Other symbols will get converted to simple line symbols
- For polygon FeatureClasses, use simple or picture fill symbols,
 - others symbols will get converted to simple fill symbols

Authoring the Map (contd.)

- By Default a Template is created for each
 - Type
 - If the Layer has types / a Unique Value Render
 - Layer
 - if no types / a Simple Renderer
- Create additional templates for the type or layer, as needed
- Edit attribute values in templates as needed

Publishing the Map

- Publish the Map in ArcCatalog
- Enable the 'Feature Access' Capability for the Map Service
- Enable Query and Edit operations for this Capability
- Create the Service

- ArcGIS Server Security applies to Feature Services
 - Set permissions on the published service

- Browse REST Services folder and find the Service' url
 - <http://jaladhi/ArcGIS/rest/services/Buildings/FeatureServer>

- Use REST url in Web Application/ iPhone Application

Geometry Service

- Provides basic geometric operations for use by thin clients
- A stateless computational service, not bound to any data on the server
- Takes and returns arrays of geometries
- Any ArcGIS Server can host a Geometry Service
 - Administrator manages the number of instances (threads)
 - Can be scaled out across the server cluster
- Exposes the ESRI Topology / Geometry engine to the web

Geometry Service

<http://<rest-services-url>/<serviceName>/GeometryServer>

[Simplify](#) - Returns an array of simplified geometries.

[Project](#) - Returns an array of projected geometries.

[Buffer](#) - Returns an array of polygons at the specified distances for the input geometry. An option is available to union buffer polygons at each distance.

[Areas and lengths](#) - Calculates areas and perimeter lengths for each polygon specified in the input array.

[Lengths](#) - Calculates the lengths of each polyline specified in the input array.

[Relation](#) - Determines the pairs of geometries from the input geometry arrays that participate in the specified spatial relation.

[Label Points](#) - Calculates an interior point for each polygon specified in the input array.

[Convex Hull](#) - Returns the Convex Hull of the input geometry.

[Densify](#) - Densifies geometries by plotting intermediate points between existing vertices.

[Generalize](#) - Returns generalized (Douglas-Poiker) versions of the input geometries.

[Cut](#) - Splits the input polyline or polygon where it crosses a cutting polyline.

[Offset](#) - Constructs the offset of the given input polyline based on an offset distance.

[Auto Complete](#) - Simplifies the process of constructing polygons that are adjacent to other polygons.

[Reshape](#) - Reshapes a polyline or a part of a polygon using a reshaping line.

[Trim / Extend](#) - Trims / Extends each polyline specified in the input array, using the user specified guide polylines.

[Union](#) - Constructs the set-theoretic union of the input geometries.

[Intersect](#) - Constructs the set-theoretic intersection between an array of geometries and another geometry.

[Difference](#) - Constructs the set-theoretic difference between an array of geometries and another geometry.

Creating your Web or Mobile Apps

- In this session we used Flex Samples from the beta Resource Center
- Focus of this session has been the feature service
- Javascript, Flex, Silverlight and iPhone API sessions have more information about Web and Mobile App Development
- Demo Apps in this session used
 - Feature Service REST URL
 - World street map's REST URL pointing to ArcGIS Online

Editing using Desktop

- Scalable editing solution for web-browsers
- Supports ArcMap as well
- Easy user interaction flow for Desktop users
- Same symbology and templates as in the map doc
- Advanced editing tools in ArcMap

Editing using Desktop: User Interaction

- Connect to ArcGIS Server
- Add a MapService to ArcMap.
- Zoom to the desired extent.
- Select the Layer(s) that you want to edit

Editing using Desktop: User Interaction (contd.)

- *Create Local Copy for Editing.* ArcMap
 - fetches the Layers and just the necessary data and
 - puts them in your local GeoDatabase
- Edit the data in ArcMap,
- Synchronize the edits with Feature Service.

Editing Geodatabases – Different Approaches

Solution	Connection Type	Symbology	Client
Web ADF	Stateful	Yes	Web-Browser
WFS-T	Stateless	No	WFS clients
GeoData Service	Stateless	Yes (in 10)	ArcMap Engine Apps
Mobile Service	Stateless	Yes	ArcGIS Mobile
Feature Service	Stateless	Yes	Web-Browser iPhone ArcMap

Editing Geodatabases – Web ADF

- **Web ADF**
 - **JSP (Java) or ASP.Net Application Development**
 - **ADF makes a server side, stateful connection to the Geodatabase**
 - **Uses remote ArcObject calls to edit the geodatabase**
 - **Large Server Side Session, expensive to Scale**

Editing Geodatabases – Geodata and Mobile Services

- **GeoDataService**
 - Supports Server to Server Replication
 - Supports Server to Client Replication and Editing
 - Equivalent to Feature Service from an ArcMap editing client perspective
- **Mobile Service**
 - Supports editing and synchronization via ArcGIS Mobile (Windows)

Editing Geodatabases – WFS-T

- **WFS-T**
 - Supports editing based on the OGC specification
 - Uses GML Simple Features Profile
 - Edit features using a WFS client

Feature Service – Editing Model

- Direct Editing over the Web
- Simple Transaction Model
 - Each call to ApplyEdits is one transaction
 - Last In Wins
 - Can edit both short transaction (non versioned) and versioned data
- Can be combined with Versioning
 - Web Editors share a version
 - Default Version or “web” version
 - Back Office reconciles and posts “web” version
 - Use non-versioned for maximum scalability / very large scale web use

Feature Service – Editing Model (contd.)

- Can be combined with Disconnected Editing
 - Exposed to Desktop Clients (ArcMap) at this Release
 - Applications can checkout data
 - Periodically Synchronize

Feature Service - Security

- **Feature Services follow the standard ArcGIS Server Security model.**
- **You can :**
 - **Manage the list of users**
 - **Allow only authenticated users to use the server**
 - **Based on credentials supplied by the user**
 - **Manage user permissions on all GIS Services**
 - **Allow only authorized users to use GIS Services**
 - **Ensure that all communication with a GIS Service is encrypted (https)**

Conclusions

- ArcGIS 10 introduces a new Feature Service
- Designed for web editing and gathering of user generated content.
- Supports a rich feature information model including symbology, types, domains, relationships and attachments
- All edits flow into and can be managed in the back end GeoDatabase
- Exposed in a wide variety of clients via easy to use Web APIs

Sessions Related to the Presentation

- WFS: Tue, Mar 23, 1:00pm-2:15pm, Smoketree A - E (PSCC)
- GeoDatabase Replication:
 - Wed, Mar 24, 1:00pm-2:15pm, Catalina/Madera (RH)
- ArcGIS Mobile: Tue, Mar 23, 4:30pm-5:45pm, Primrose A (PSCC)
 - Thu, Mar 25, 8:30am-9:45am, Primrose A (PSCC)
 - Thu, Mar 25, 10:15am-11:30am, Catalina/Madera (Renais...)
- ADF:
 - .Net: Thu, Mar 25, 10:15am-11:30am, Primrose C/D (PSCC)
 - Java: Thu, Mar 25, 10:15am-11:30am, Smoketree A - E (PSCC)
- ArcGIS Server Security:
 - .Net: Wed, Mar 24, 2:45pm-4:00pm, Primrose C/D (PSCC)
 - Java: Wed, Mar 24, 2:45pm-4:00pm, Smoketree A - E (PSCC)

Sessions Related to the Presentation (contd.)

- REST: Tue, Mar 23, 1:00-2:15pm, Pasadena/Ventura/Sierra (R H)
- Javascript API
 - Wed., March 24, 4:30pm-5:45pm, Primrose B (PSCC)
 - Thu., March 25, 1:30pm-2:45pm, Primrose B (PSCC)
- Flex API
 - Wed, March 24, 10:30am-11:45am, Primrose A (PSCC)
 - Thu, March 25, 1:30pm-2:45pm, Pasadena/Ventura/Sierra (RH)
- Silverlight API
 - Wed, March 24, 2:45pm-4:00pm, Primrose B (PSCC)
 - Thu, March 25, 1:30pm-2:45pm, Catalina/Madera (RH)
- iPhone
 - Wed, March 24, 4:30pm-5:45pm, Pasadena/Ventura/Sierra (RH)
 - Thu, March 25, 1:30pm-2:45pm, Primrose C/D (PSCC)

Questions?

- Thanks for your attention!