

Synchronizing Field Data in ArcGIS Runtime for WPF Using Feature Services

Mark Cederholm
UniSource Energy Services
Flagstaff, Arizona

Field Applications at UNS

- GPS crew: ArcMap + GDB Replication

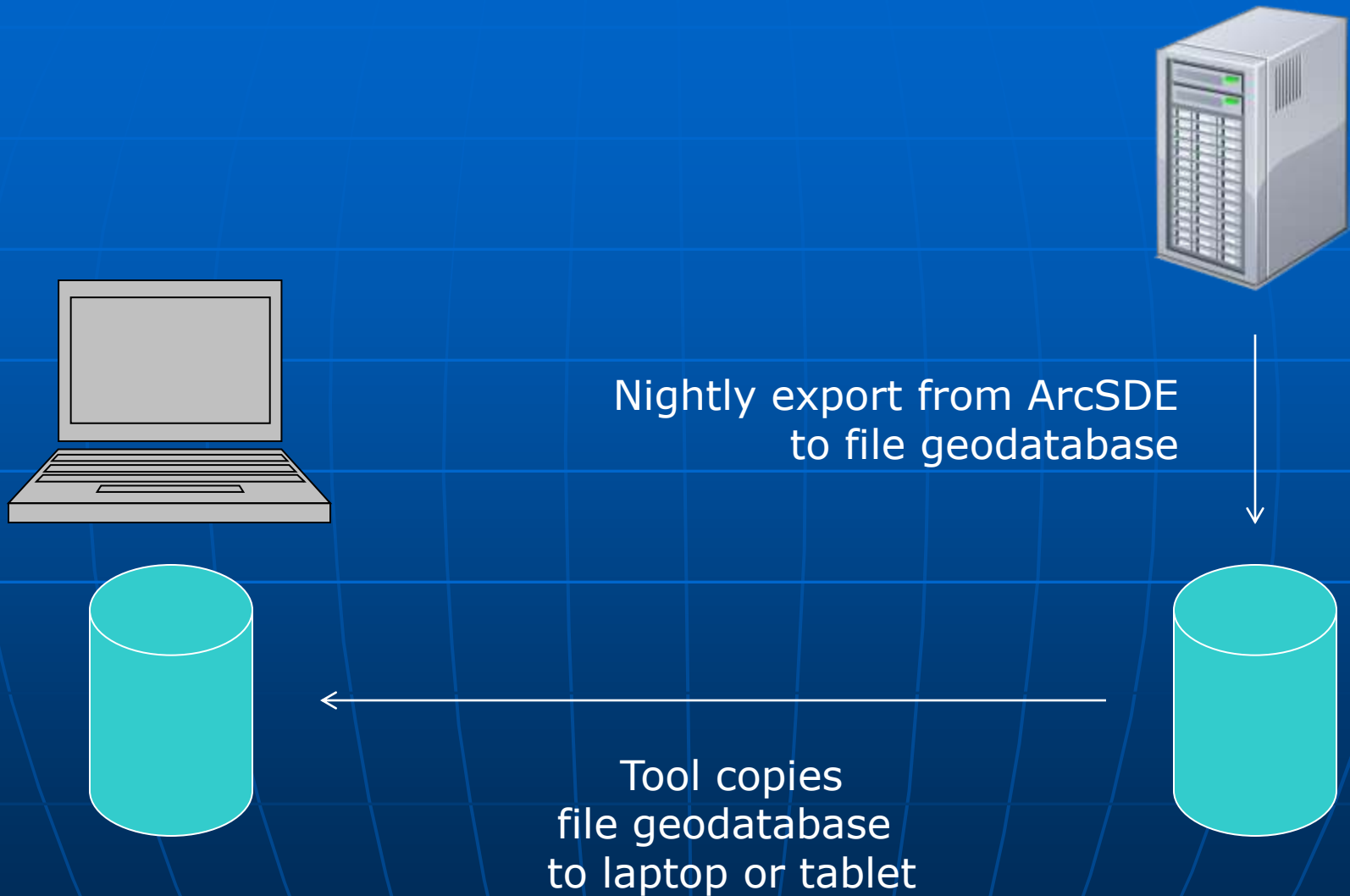
Other field apps:

- Markups (redlining)
- Gas pressure measurements
- Cathodic protection readings
- Future: leak survey, tree trimming, pole assembly inventory, phasing verification

Field App Requirements at UNS

- Rural AZ: no wireless or cell phone connectivity
- Full copy of operational data
- Create simple features and rows
- Update simple feature attributes
- Synchronize edits with master GDB

Operational Data

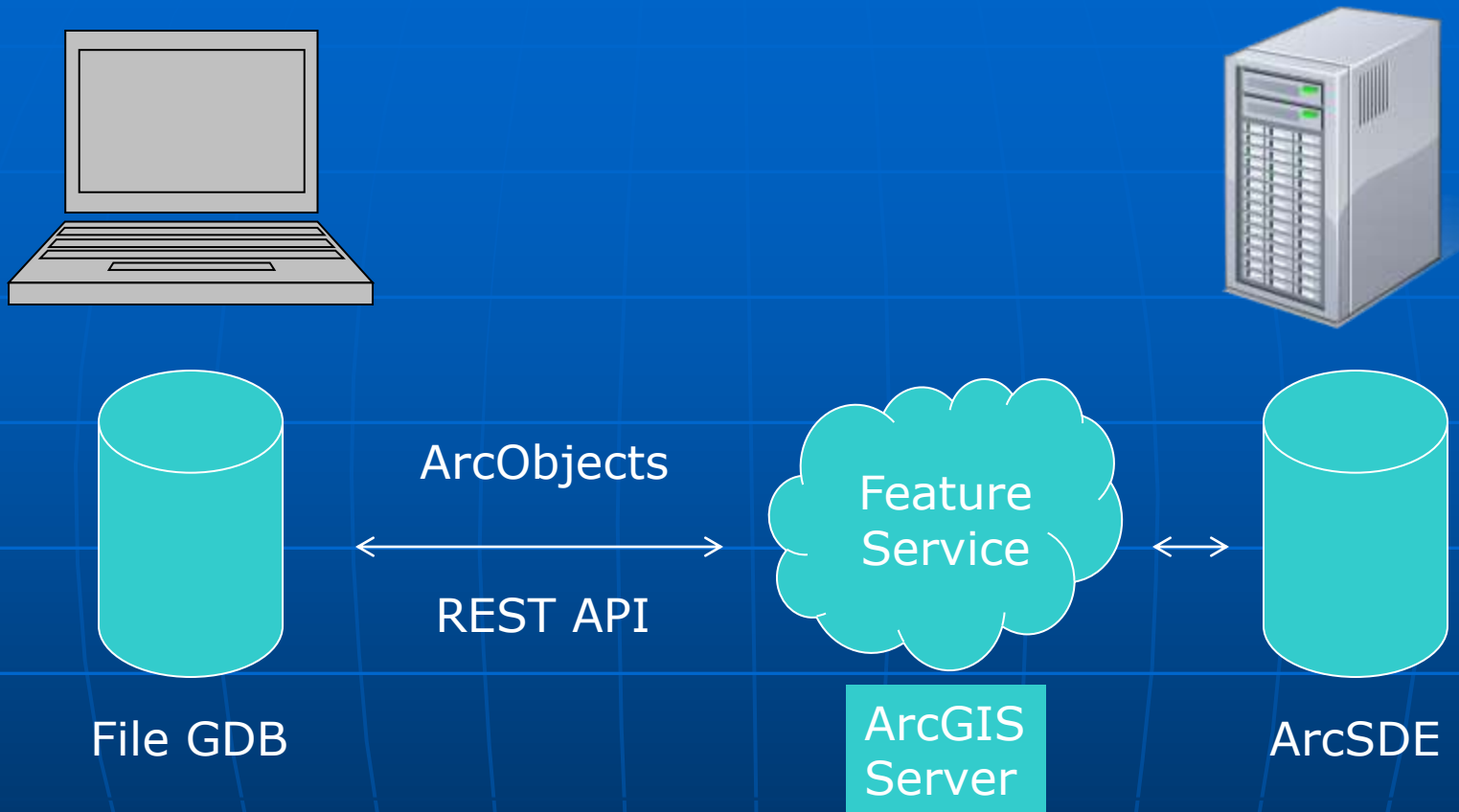


Field Data Edits (the 9.x way)

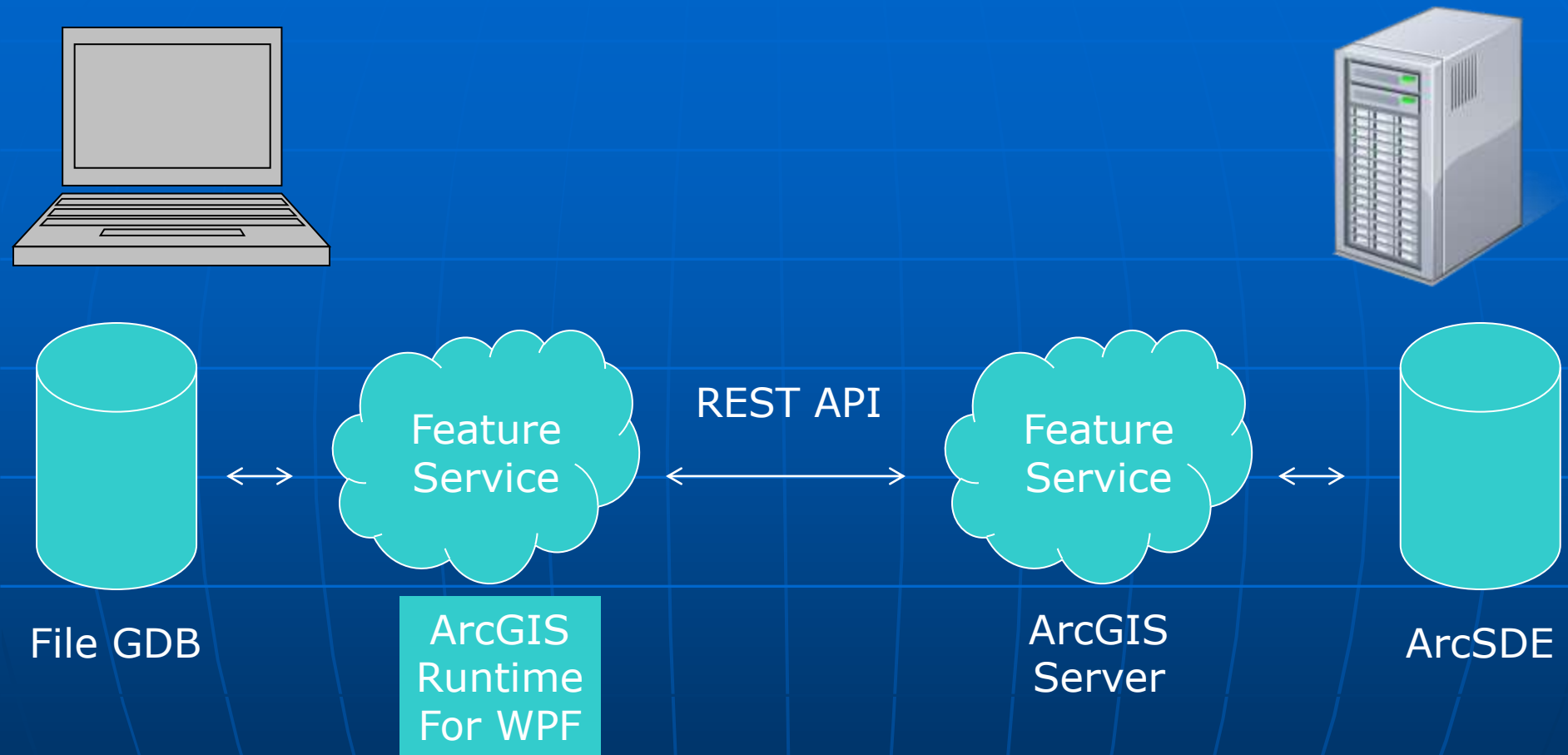


(Requires Oracle client and DB access)

Field Data Edits (the 10.0 way)



Field Data Edits (the Runtime way)



DEMO

Basic Logic, Part 1

- From edit log, get last sync date (if there is one)
- Query features by timestamp from remote service and send to local service
- Check if features received are adds or updates
- Update the sync date in the edit log

Basic Logic, Part 2

- From edit log, build lists of adds and updates to send
- Retrieve features and rows from local service and send to remote service with updated timestamp
- If all features are successfully sent, delete edits from log

Assumptions

- Uses 10.0 feature service functionality (not replica-aware)
- Records created or updated in the field are never deleted on the server
- Geometry types to be synchronized are 2D Points, Polylines, or Polygons
- Relationships between classes to be synchronized are through a field other than OBJECTID or GLOBALID

The ID Field

- A separate unique ID field must be populated whenever a feature or row is created
- This is because a feature service cannot add a feature with a specific global ID

The Timestamp Field

- The timestamp field must be updated any time a feature or row is created or edited
- When sending field edits to the remote service, the timestamp is updated to just before the current sync time
- Otherwise, other field people may not see the updates when they synchronize

Pros

- No Oracle client or DB access is required
- Runtime licensing is cheaper than ArcObjects

Cons

- Requires maintenance of ID and timestamp fields
- Performance is slower than using ArcObjects
- Receiving updates before sending raises the (albeit unlikely) possibility of overwriting local edits

Failed Experiments

File Geodatabase API:

- Goal: to boost performance updating local data
- Still requires a separate ID field
- Does not work gracefully with data opened by a map service (lock and spatial index issues)
- Really intended for importing and exporting data

Failed Experiments

REST API 10.1 (replication):

- Goal: to improve performance and eliminate the need for a timestamp
- ID field still required on local GDB
- Register and unregister work fine
- Cannot get synchronize to work (even Esri tech support is stumped!)

UNDER THE HOOD

Questions?

- Mark Cederholm
mcederholm@uesaz.com
- This presentation and sample code may be downloaded at:

<http://www.pierssen.com/arccgis10/runtime.htm>