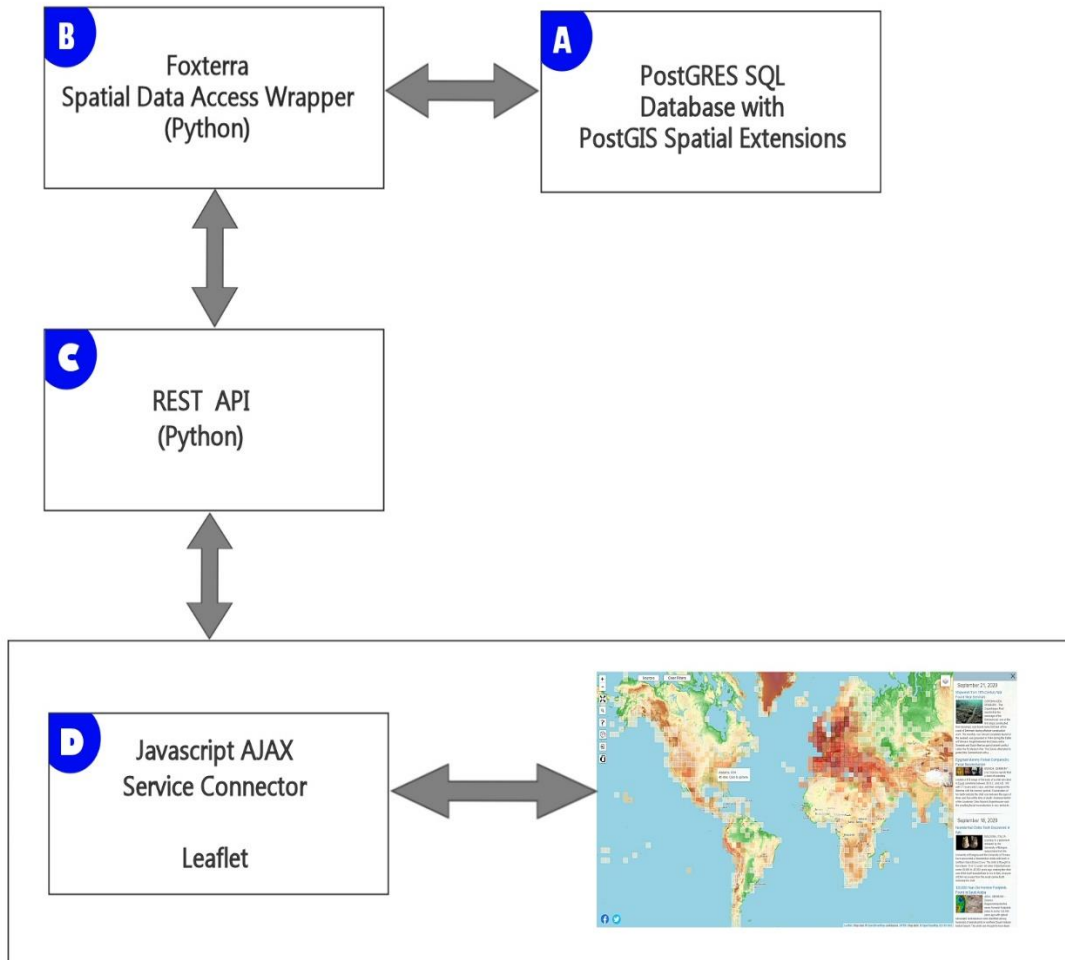


Foxterra Technology Overview

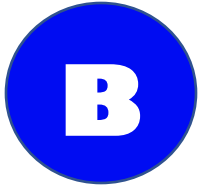




PostgreSQL / PostGIS Extension Database

- Advanced SQL database
- Open sourced with active developer community
- Optimized - very fast spatial queries
- Complex – requires expert PostGIS specialists for serious development tasks

```
-- Select data from income class including polygon intersection fraction
SELECT A.uniqueid, G.gid, A.geocode, A.income, A.bracket, A.bracket_pct, G.geom,
ST_AsText(G.geom),ST_Area(ST_intersection(ST_Transform(geom,utmzone(ST_Centroid(geom))),
ST_Transform(ST_MakePolygon(ST_GeomFromText('LINESTRING(-114.107437134 51.0983477909,
-114.107437134 51.0534808838,-114.169921875 51.0534808838,-114.169921875 51.0983477909,
-114.107437134 51.0983477909)', 4326)), utmzone(ST_Centroid(geom)))))) /
ST_Area(ST_Transform(geom, utmzone(ST_Centroid(geom)))) AS _x
FROM income_2016 A, census_boundary G
WHERE ST_Intersects(( ST_GeomFromText('POLYGON((-114.107437134 51.0983477909,
-114.107437134 51.0534808838,-114.169921875 51.0534808838,-114.169921875 51.0983477909,
-114.107437134 51.0983477909)'),4326)),geom)
AND A.geocode=G.geocode
```



Foxterra Spatial Data Access Wrapper

- Hides the complexity of PostGIS queries
- Consistent interface for any database geometry
- Result set based on requested rectangle, polygon or circle
- Returns fraction of intersected rectangle, polygon or circle
- Returns 'n' features closest to specified geometry
- Geocoding using open source and proprietary geocoding engines
- Reverse geocoding using open source and proprietary geocoding engines
- Built in geoJSON generation of result sets

```
# Use a built in intersection method
h = Income()
aggregate = 0.0
result = h.fractionIntersectPolygon(refPts)
while result:
    frac = h._intersectFraction
    # Do something with the Income class contents
    aggregate += frac * h._minGrossIncome
    result = h.fractionIntersectPolygon(refPts)
```



REST API (Python)

- Accepts input parameters, typically from AJAX request
- Instantiates a Foxterra Spatial Access class
- Uses Foxterra Spatial Access methods
- Accumulates results
- Returns result set to the AJAX request as:
 - JSON
 - GeoJSON
 - Text
 - Image

```
# Return a text buffer containing HTML to the AJAX request

bfr = '<div class="" + titleCss + "">Closest Major Projects</div>'
if loc != '':
    bfr += '<div class="locHeader">Centered at ' + loc + '</div>'
a = AnalyticsReporting()
bfr += a.ClosestMajorProjects(ctrPtArray, 20, loc)
print( bfr )
```



Javascript AJAX Service Connector

- Makes AJAX request to REST API
- Parses returned request buffer
 - Post results to map using Leaflet methods
 - Build HTML content
- Javascript content contains Leaflet layer builders
 - Instantiate Leaflet layer(s)
 - Build layer control
 - Metadata driven

```
function randomSite() {
  var theUrl = serviceUrl + "fetch_random_site.py";
  $.ajax(
  {
    url: theUrl,
    type: 'POST',
    dataType: 'json',
    cache: true,
    success: function(jsonData)
    {
      flyToSiteObj = eval(jsonData);
      map.flyTo([flyToSiteObj.latitude, flyToSiteObj.longitude], 12,
        {'duration':2.0, 'easeLinearity':1.0, 'noMoveStart':false});
    },
    error: function (jqXHR, ajaxOptions, thrownError) {
      message("Please try again", 10);
    }
  });
}
```

```
# Single layer entry for layer instantiation and control

Graticules_20_Degrees = {"order":90, "select": False, "opacity": 1, "tooltip": "",
"pop_up":"","username": "", "datagrid": False, "format": "SHP", "image": "",
"heatMap": False, "surrogate": "", "locale": "", "open": False, "minZoom": 0,
"legend": "", "maxZoom": 18, "attribution": "", "visible": False, "cssClass":
"selectLine", "type": "static", "text": "Graticules 20 Degrees", "imagerySet": "",
"password": "", "filters": False, "zindex": 90, "owner": "", "source": "", "color":
"black", "weight": 1, "geometry": "polyline", "URL": "spatial/ne_10m_graticules_20",
"layerControl":True}
```