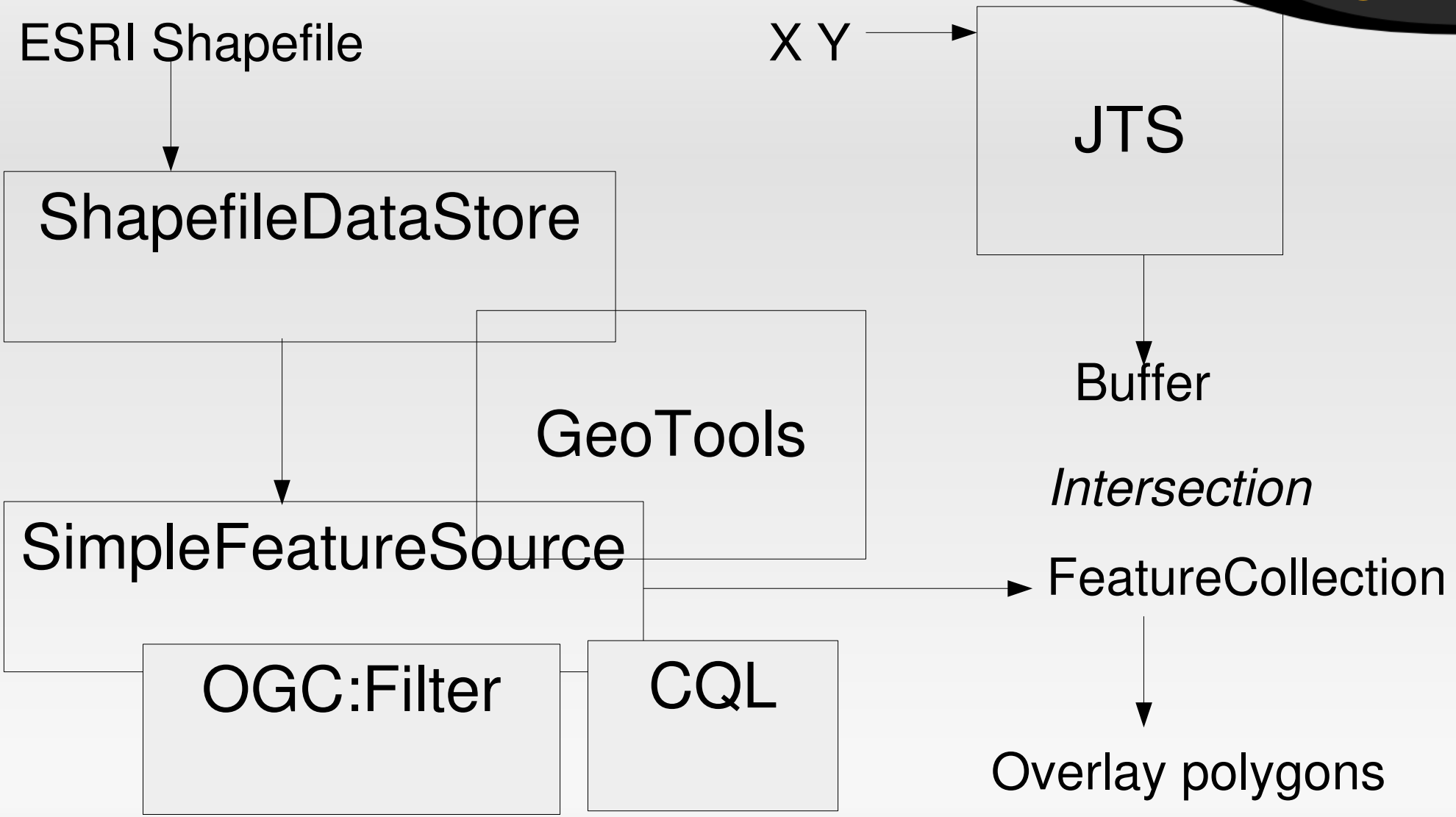
Spatial operations

Introduction

- Find overlay of buffer around defined point
- Create buffer around defined point
- Calculate overlay between buffer polygon and plygons in layer
- Count area of overlay

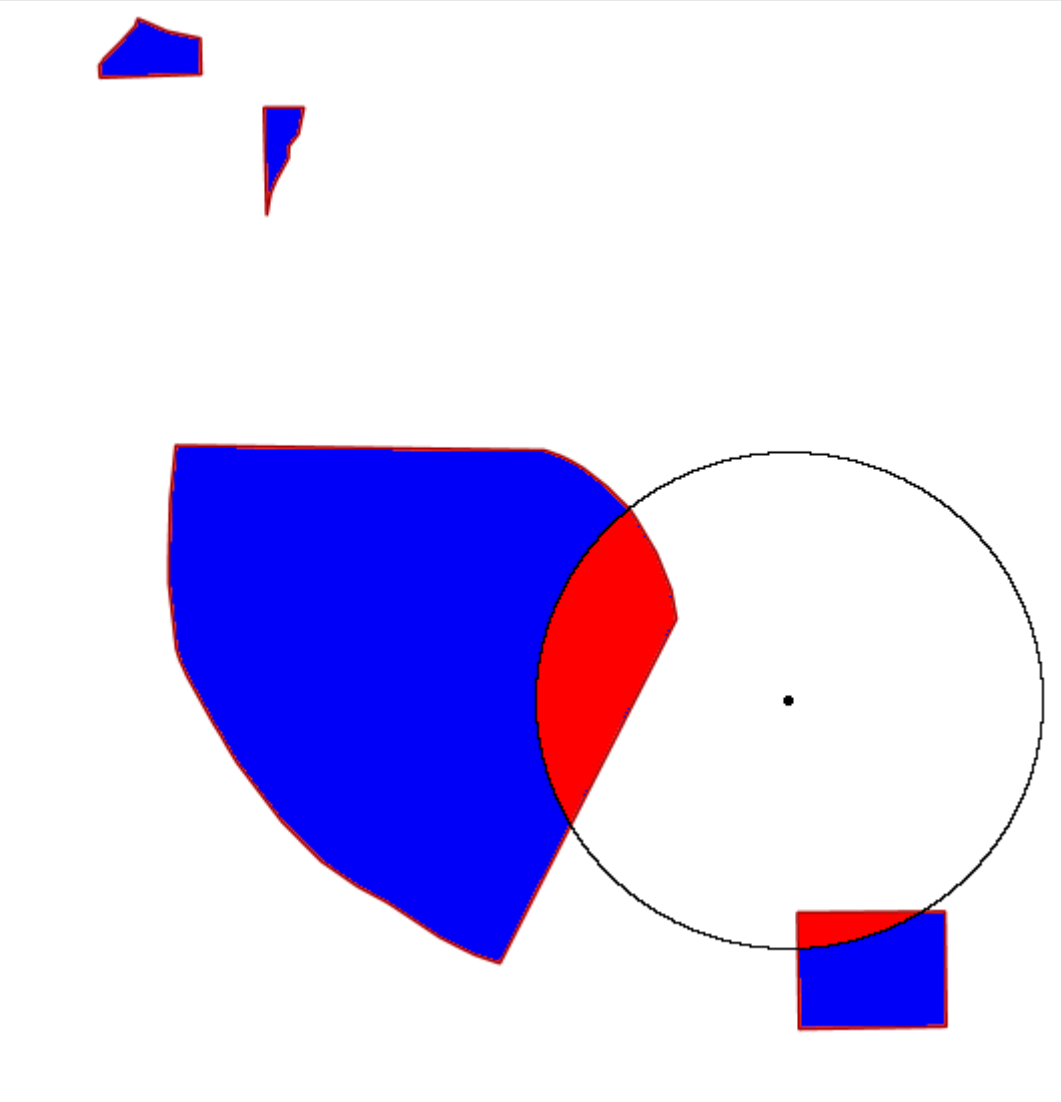
Principle



Steps

- Connect to plygons layer
- Read coordinates and distance from input
- Create buffer around point
- Overlay buffer with features in layer
- For each overlay calculate area

Sample



ubuntu

Sample output

Finded objects: Object: Area

1: 0.0

2: 0.0

4: 1550081.2884726578

3: 5678137.145570624

Read Shapefile

```
ShapefileDataStore sfds = new
  ShapefileDataStore(new
  URL("file:///data/install/geoserver/geoserv
  er-2.8.2/data_dir/data/sf/restricted.shp"));

SimpleFeatureSource fs =
  sfds.getFeatureSource("restricted");
```

Create point

```
GeometryFactory gf = new
  GeometryFactory();

String xy[] = pointString.split(" ");

Point point = gf.createPoint(new
  Coordinate(Double.parseDouble(xy[0]),
  Double.parseDouble(xy[1])));
```

Buffer

```
Polygon p1 = (Polygon)
  point.buffer(distance);
```

ubuntu

Overlay

```
SimpleFeatureIterator sfi =
  fs.getFeatures().features();

while (sfi.hasNext()) {

 SimpleFeature sf = sfi.next();

 MultiPolygon mp2 = (MultiPolygon)
  sf.getDefaultGeometry();

 Polygon p2 = (Polygon) mp2.getGeometryN(0);

 Polygon p3 = (Polygon) p2.intersection(p1);

 names = names + "\n" + sf.getAttribute("cat") +
  ": " + p3.getArea();

}
```

Add to service

```
@DescribeProcess(title="overlayWPS",
   description="Creates buffer around point and overlays
   it with polygon layer. Returns areas of overlay.")

public class OverlayWPS implements GeoServerProcess {

 ...

}
```

Add to service / 2

```
@DescribeResult(name="result",
  description="output result")

   public String
  execute(@DescribeParameter(name="point",
  description="point") String point,
  @DescribeParameter(name="distance",
  description="distance to search") double
  distance) {

        Examples e = new Examples();

        return e.overlay(point, distance);

   }
```

Testing WPS

Constructeur de requête WPS

Constructeur pas à pas de requête WPS.

Choisir process

gs:OverlayWPS

Creates buffer around point and overlays it with polygon layer. Returns areas of overlay. (WPS DescribeProcess)

Entrées du process

point* - String

point

600000 4920000

distance* - Double

distance to search

5000

Sorties du process

result* - String

output result

☑ Generate

ubuntu

Input for WPS

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS" xmlns:xsi=
  <ows:Identifier>gs:OverlayWPS</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>point</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>600000 4920000</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distance</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>5000</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Output WPS

```
Nalezené objekty: Objekt:Plocha překryvu
1: 0.0
2: 0.0
4: 1550081.2884726578
3: 5678137.145570624
```

ubuntu