GeoServer Cartographic Rendering

New features for map makers

Andrea Aime

aaime@opengeo.org

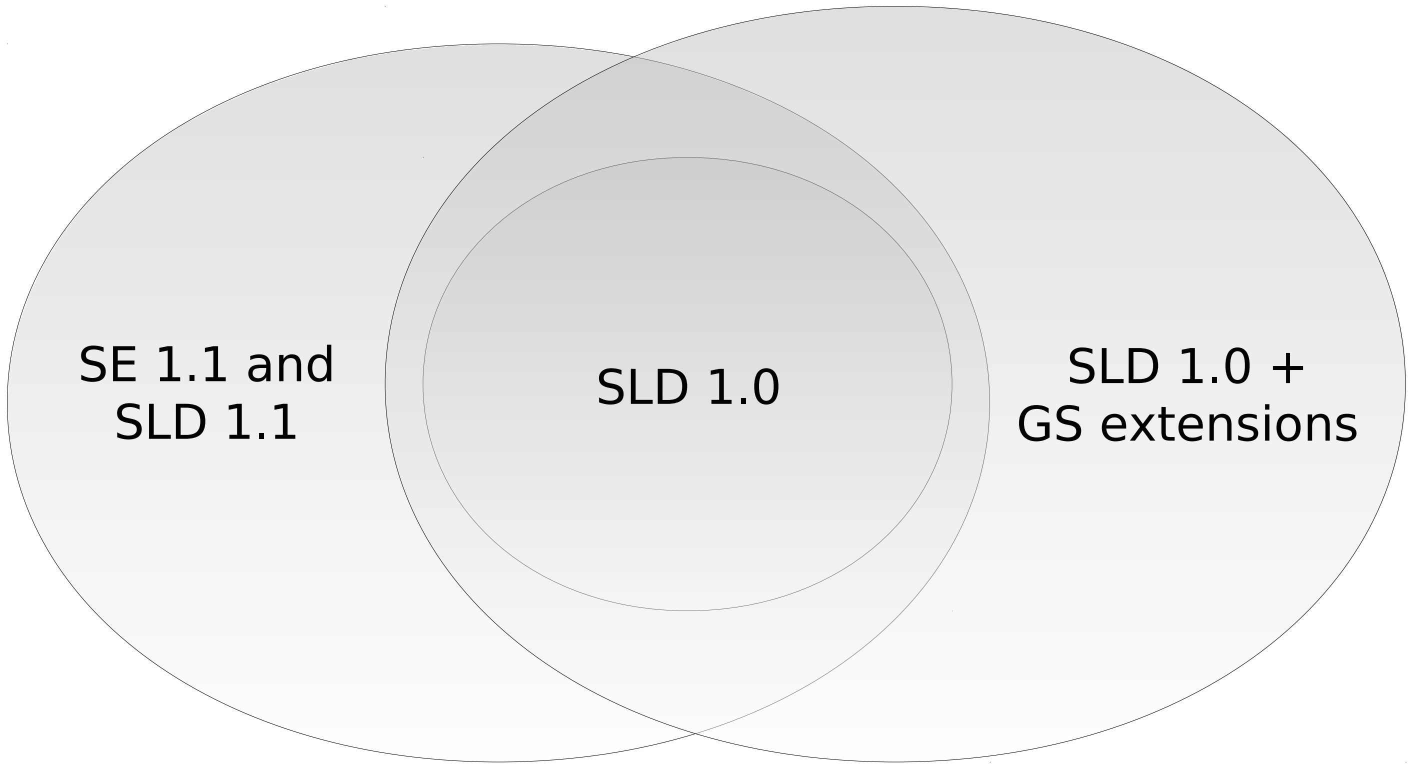
OpenGeo
http://opengeo.org

OpenGeo
http://opengeo.org

Three SLDs

http://opengeo.org

SLD 1.0, 1.1, and GS one



http://opengeo.org

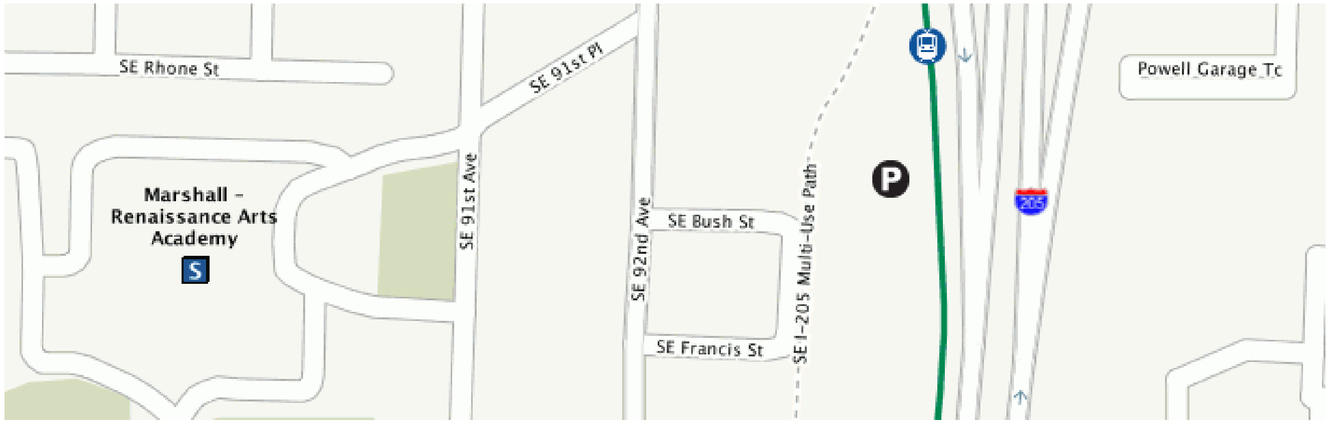
SE 1.1 improvements

- Symbolizers in real world units (uom)
- Selected geometry transformations: offsets, buffers
- External symbol sets (support for decoravite fonts)
- Functions: numeric, date and string formatting, categorization, interpolation, and recoding

OpenGeo
http://opengeo.org

GeoServer improvements

- GeoServer extended SLD 1.0 over time by adding a number of vendor extensions
- Some shared with SE 1.1, some unique
- That's the content of this presentation!



OpenGeo
http://opengeo.org

Summary

- Recent improvements
- Filter functions
- Geometry transformations
- Labeling

1.0	SLD 1.0	2.0	GeoServer 2.0.x
1.1	SLD 1.1 only	2.1	GeoServer 2.1.x (trunk)
GS	GeoServer specific		

OpenGeo
http://opengeo.org

Improvements

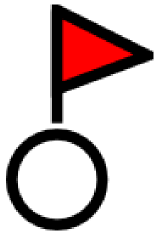
Things we were missing or not doing quite right

OpenGeo
http://opengeo.org

Graphic strokes (finally!)

- Graphic stroke: replicate an image along a line

```
...
<LineSymbolizer>
  <Stroke>
    <GraphicStroke>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource xlink:type="simple" xlink:href="burg02.svg" />
          <Format>image/svg+xml</Format>
        </ExternalGraphic>
        <Size>
          <ogc:Literal>20</ogc:Literal>
        </Size>
      </Graphic>
    </GraphicStroke>
  </Stroke>
</LineSymbolizer>
...
```



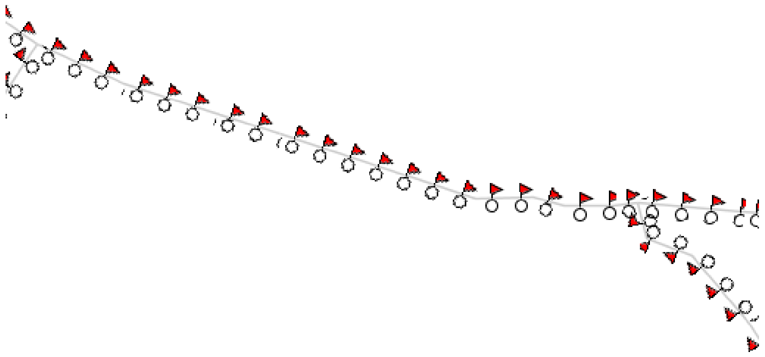
2.1

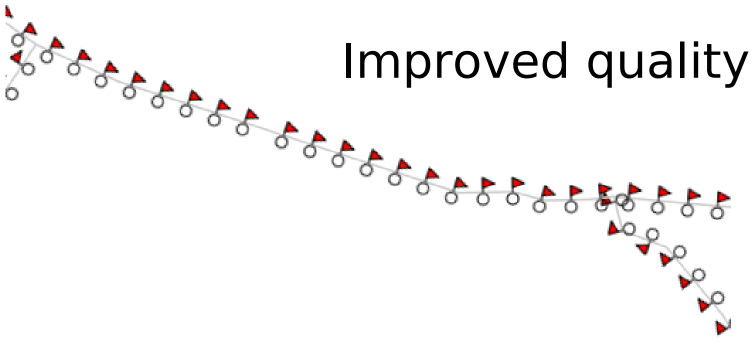
1.0

OpenGeo http://opengeo.org

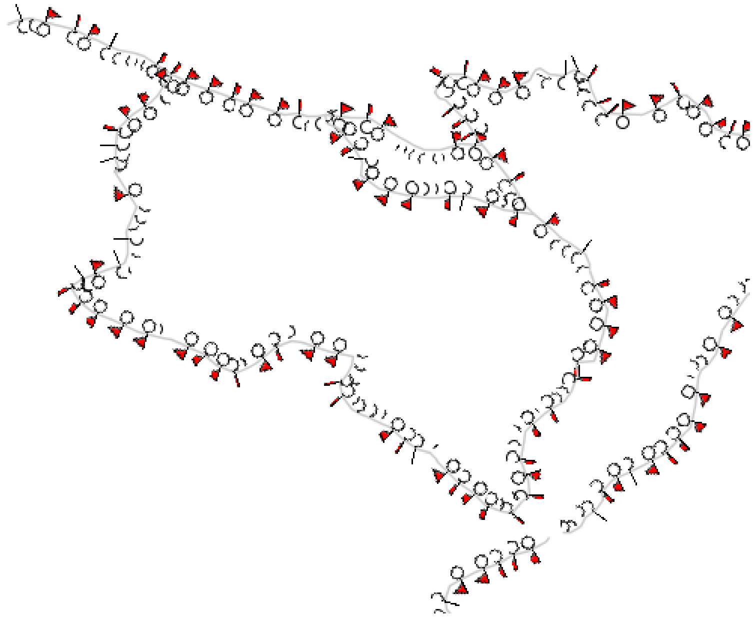
2.0.3

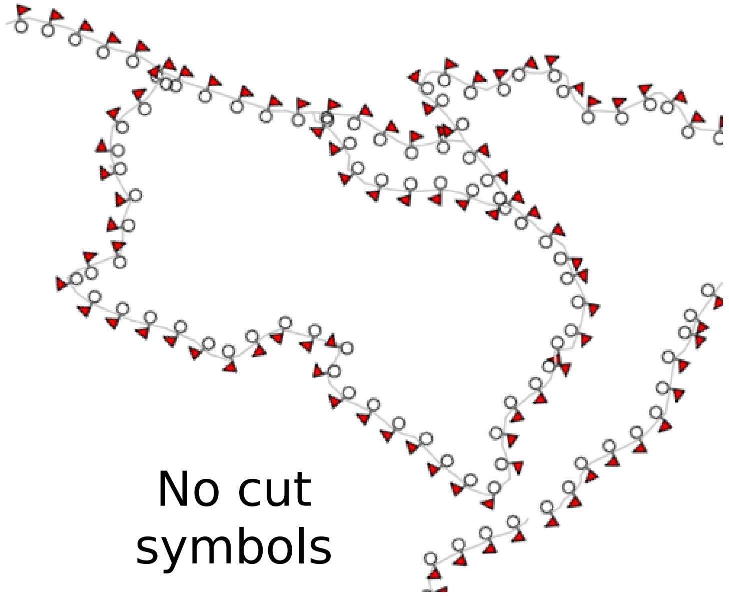
2.1.x (trunk)





Improved quality

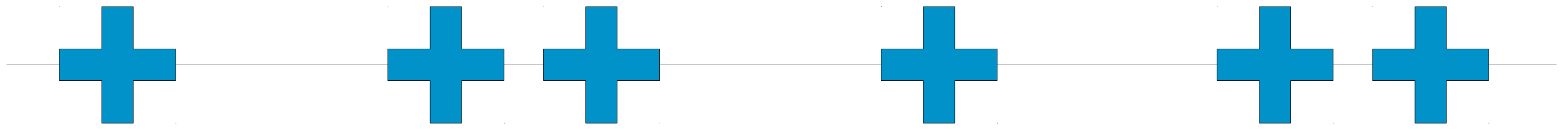




No cut symbols

… *adding dash-arrays*

- "The stroke-dasharray element encodes a dash pattern as a series of space separated floats"
- What about mixing dash array with graphic stroke? Spec does not say…



2.1

1.0 GS

OpenGeo http://opengeo.org

```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#000000</CssParameter>
        <CssParameter name="stroke-width">15</CssParameter>
        <CssParameter name="stroke-linejoin">round</CssParameter>
        <CssParameter name="stroke-linecap">round</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <GraphicStroke>
          <Graphic><Mark>
              <WellKnownName>Circle</WellKnownName>
              <Fill><CssParameter name="fill">#FFFFFF</CssParameter></Fill>
            </Mark><Size>5</Size></Graphic>
        </GraphicStroke>
        <CssParameter name="stroke-dasharray">5 35</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <LineSymbolizer>
      <Stroke>
        <GraphicStroke>
          <Graphic><Mark>
              <WellKnownName>Star</WellKnownName>
              <Fill><CssParameter name="fill">#FFFFFF</CssParameter></Fill>
            </Mark><Size>10</Size></Graphic>
        </GraphicStroke>
        <CssParameter name="stroke-dasharray">10 30</CssParameter>
        <CssParameter name="stroke-dashoffset">20</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Solid black line

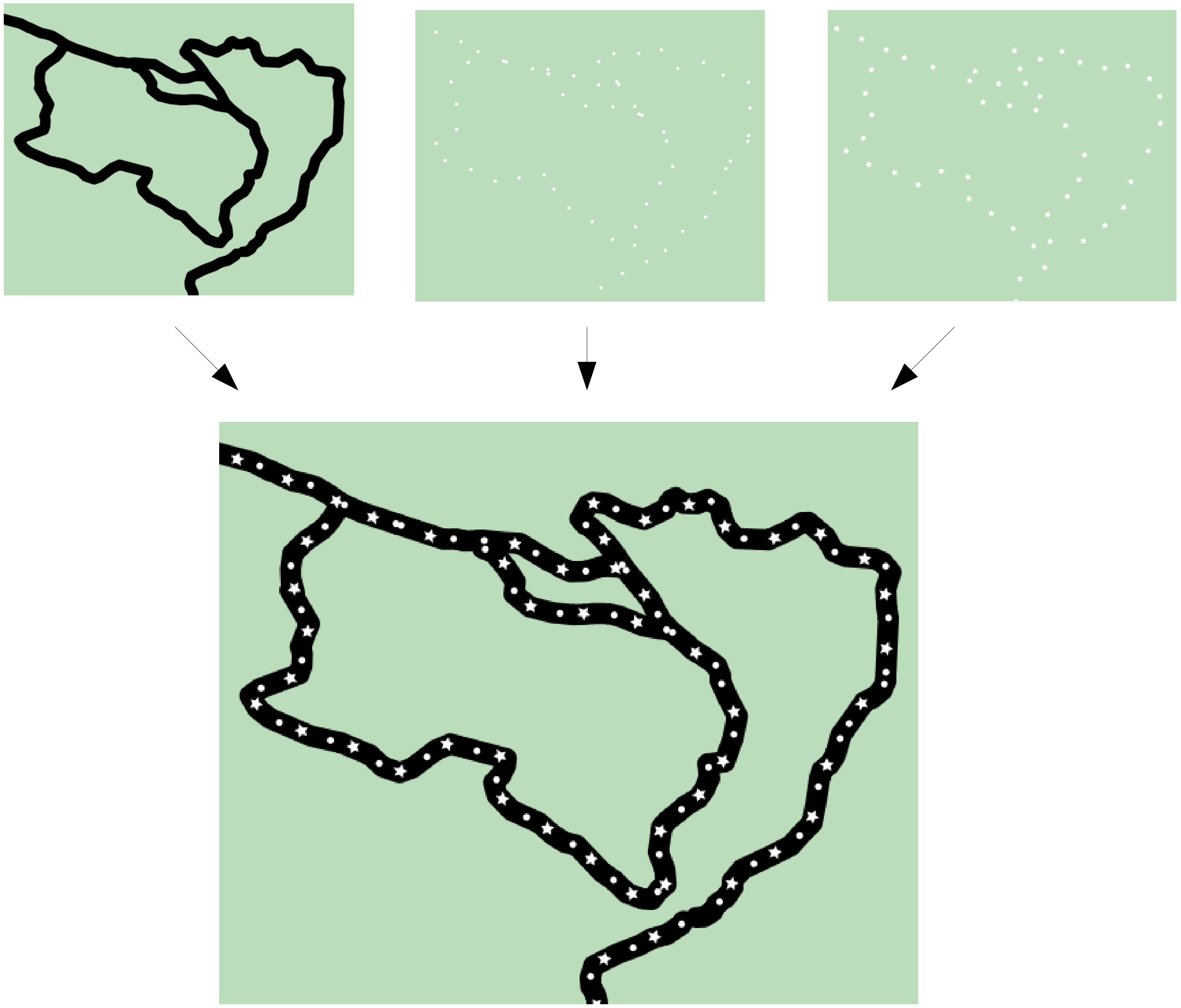
Repeated little white circle

Repeated star

2.1

1.0 GS

OpenGeo http://opengeo.org



2.1

1.0

GS

OpenGeo
http://opengeo.org

Full vector export

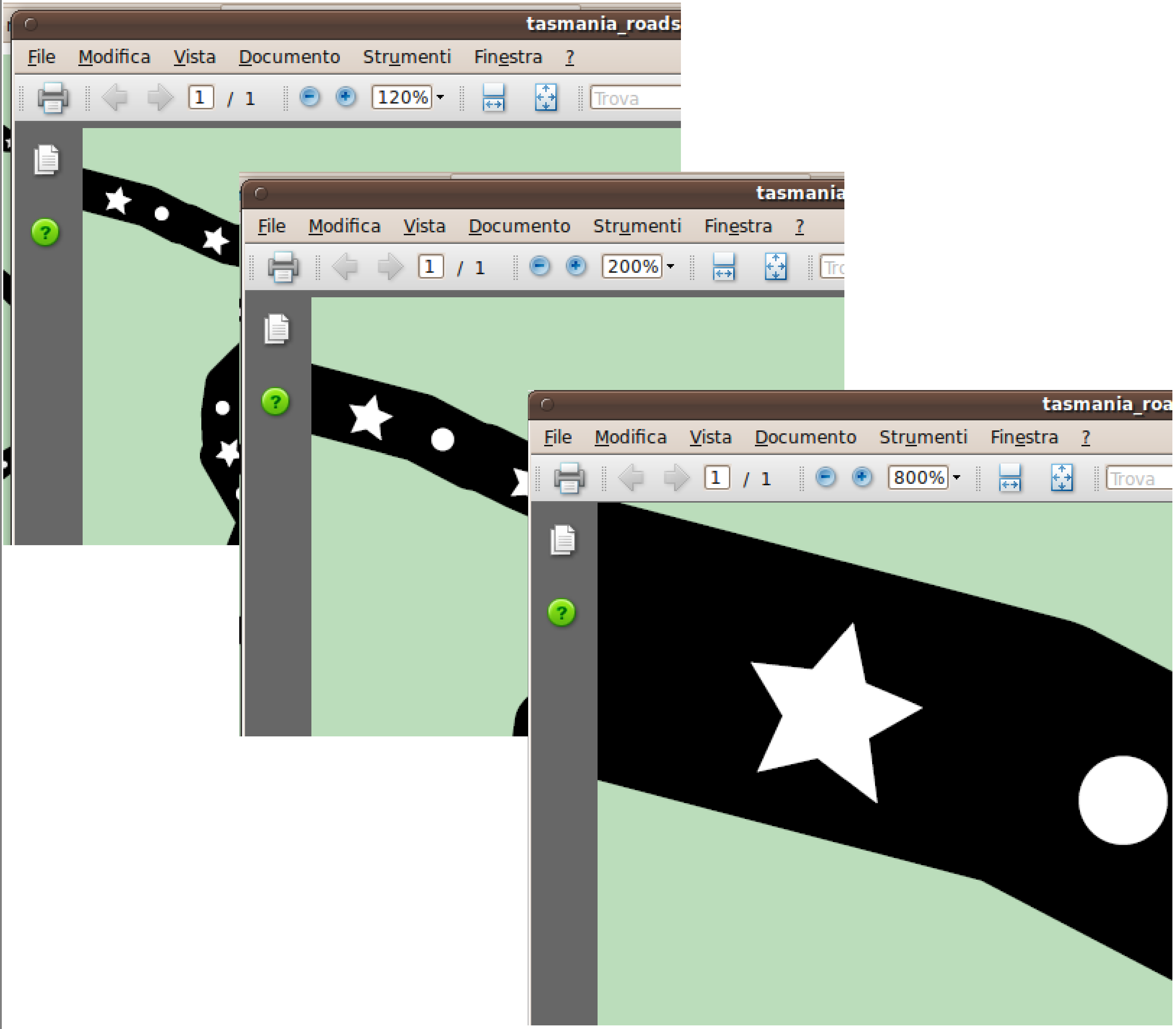
- SVG and PDF formats now provide full vector output
- Requires that all graphic are vector themselves: marks or SVG symbols
- History:
  - In 1.7.x all graphics were rasterized
  - In 2.0.x support vector output of point and polygon fills thanks to **Milton Jonathan** work
  - In trunk 2.1.x complete support (graphic strokes as well)

2.0

2.1

1.0

OpenGeo
http://opengeo.org


tasmania_roads
File Modifica Vista Documento Strumenti Finestra ?
1 / 1
120%
Trova
tasmania
File Modifica Vista Documento Strumenti Finestra ?
1 / 1
200%
tasmania_roa
File Modifica Vista Documento Strumenti Finestra ?
1 / 1
800%
Trova

Unit Of Measure

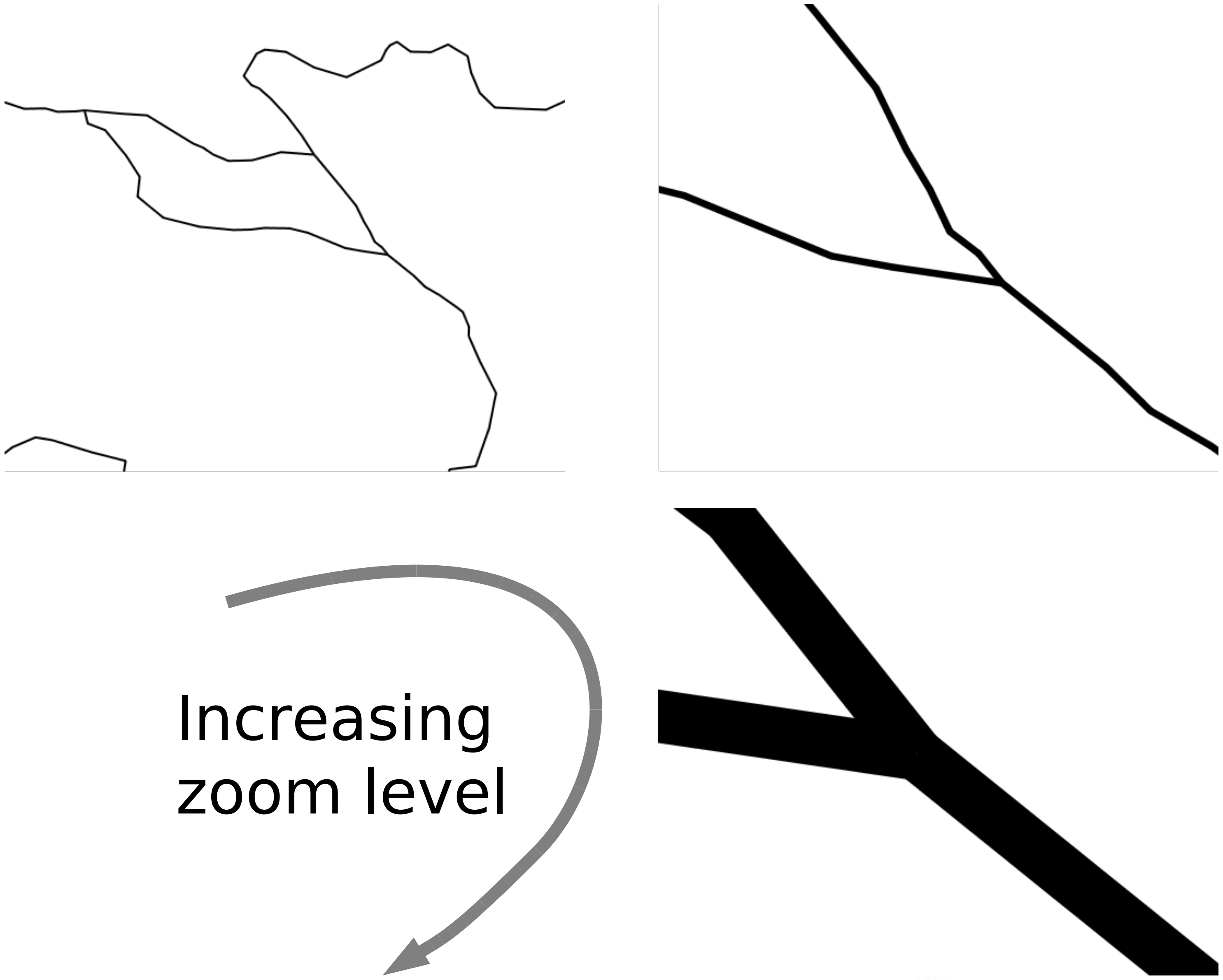
- SLD 1.0 supports only pixels
- SLD 1.1 has a **uom** attribute: pixels, meters or feet
- GeoServer SLD 1.0 accepts the UOM attribute anyways (thanks again to the work of **Milton Jonathan**)

```
<LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">
  <Stroke>
    <CssParameter name="stroke-width">500</CssParameter>
  </Stroke>
</LineSymbolizer>
```

2.1

1.1

OpenGeo
http://opengeo.org



2.1

1.1

OpenGeo
http://opengeo.org

The twilight zone

Stuff that is part of the SLD specification, yet it's not portable

OpenGeo http://opengeo.org

Leveraging SLD flexibility

- In SLD most elements are of the type **ogc:Expression**
  - Attribute names
  - Math (ogc:Add, ogc:Div, …)
  - **Call functions!**
- Functions are open ended!

$$e = f(x, y, z)$$

Filter functions

- The ***concept*** of filter function is **part of the OGC Filter spec**. A filter function is an expression with a name and a set of arguments
- However there are **no standardized functions in SLD 1.0**, and only a handful in SE 1.1
- GeoServer has **hundreds** built-in: http://docs.geoserver.org/stable/en/user/filter/function_reference.html

OpenGeo
http://opengeo.org

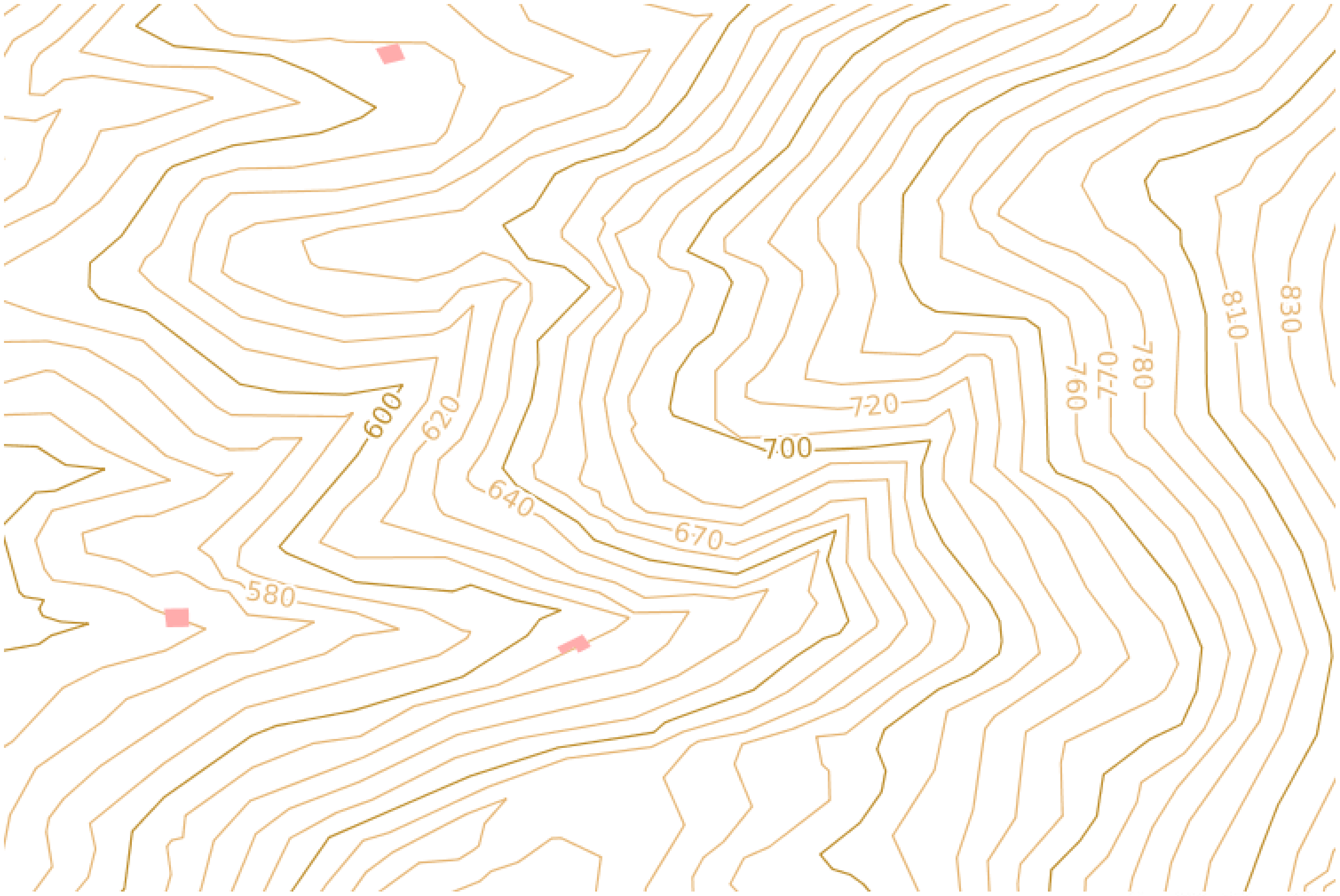
Filter function examples

- **Math**: abs, sin, cos, tan, floor, round, random, toDeegrees, toRadians, …
- **String**: strEqualsIgnoreCase, strLength, strReplace, strSubstring, strToLowerCase, strToUppercase, ...
- **Parsing** and formatting: dateFormat, numberFormat, …
- **Geometry** ones: intersects, union, ...

OpenGeo
http://opengeo.org

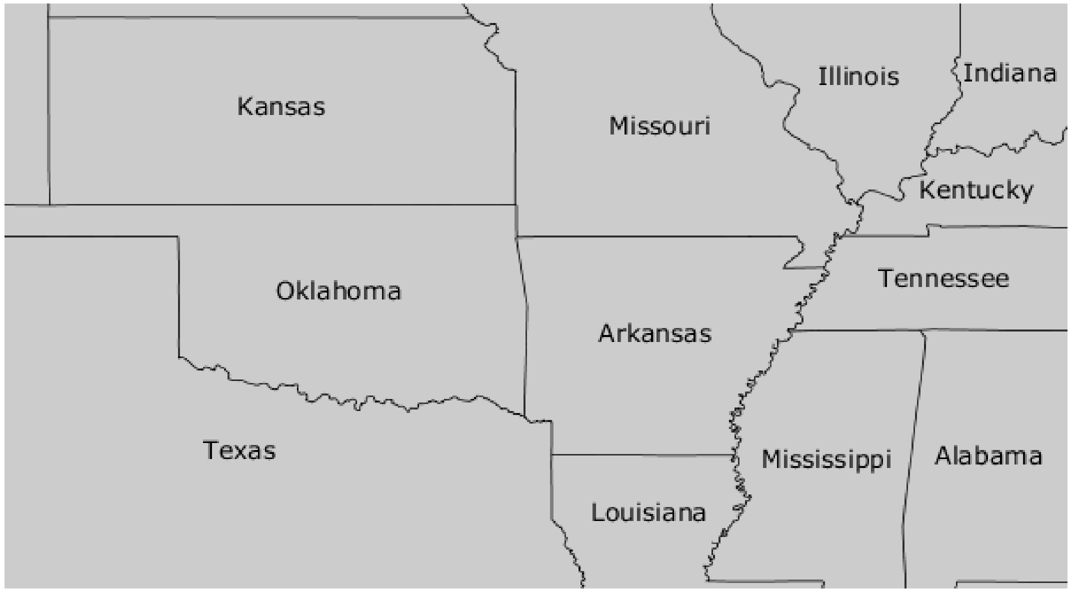
```
<TextSymbolizer>
   <Label>
     <ogc:Function name="roundDouble">
       <ogc:PropertyName>COTA_0201</ogc:PropertyName>
     </ogc:Function>
   </Label>
   ...
```

Float point field, would result in 620.0, sometimes in 619.999999999

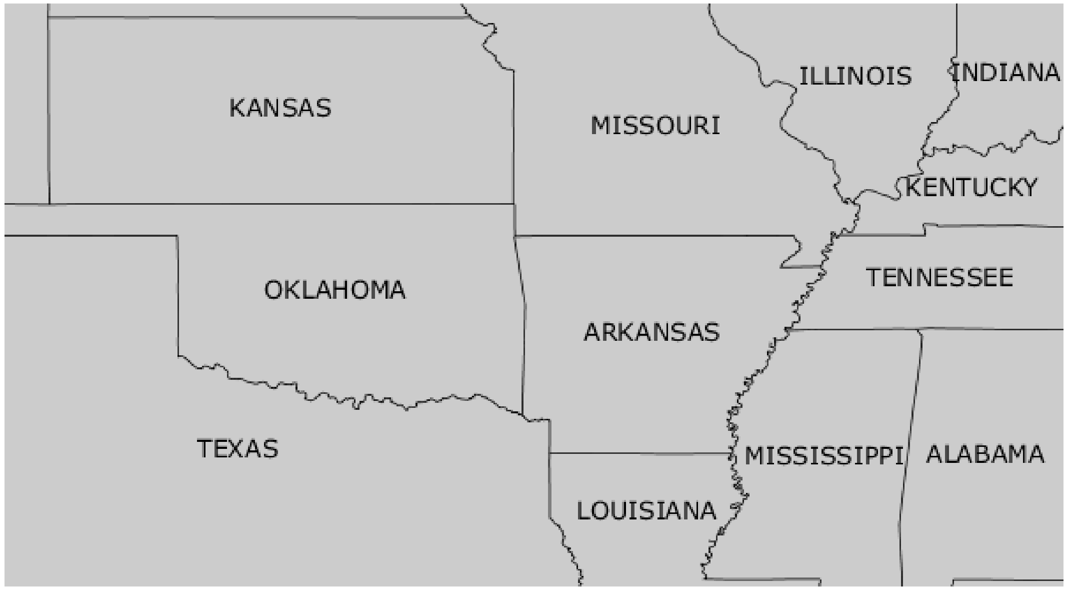


2.0

GS



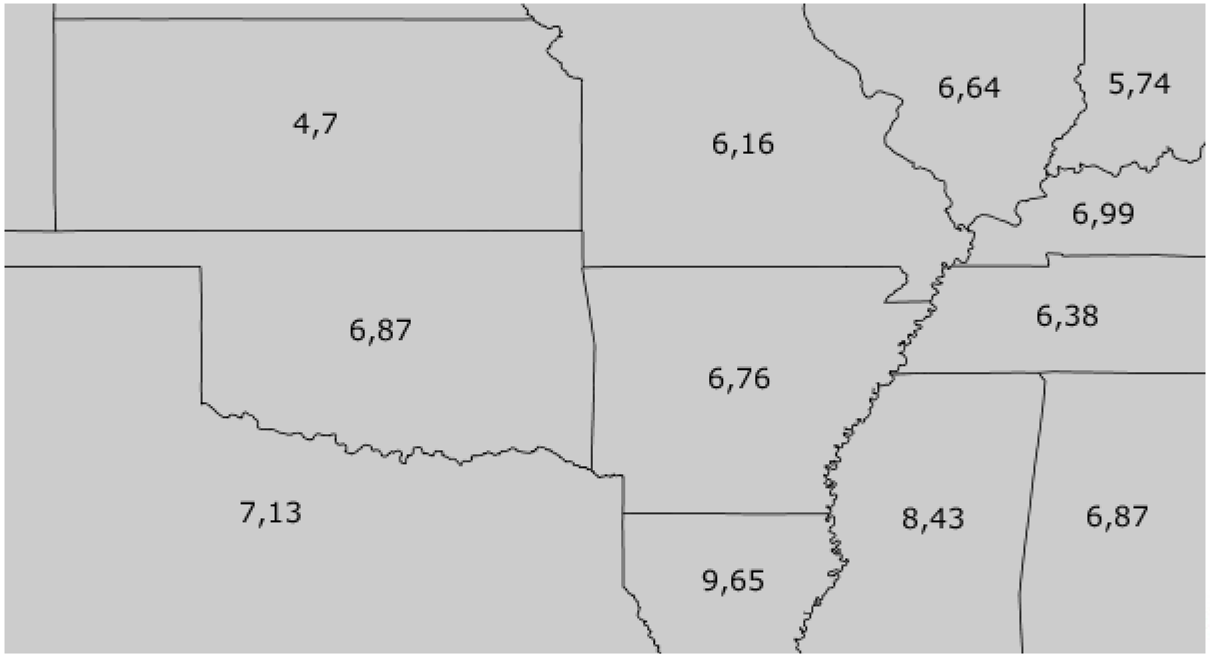
```
<Label>
  <ogc:Function name="strToUppercase">
   <ogc:PropertyName>STATE_NAME</ogc:PropertyName>
  </ogc:Function>
</Label>
```



2.0

GS

```
<Label>
  <ogc:Function name="numberFormat">
     <ogc:Literal>#.##</ogc:Literal>
     <ogc:Mul>
       <ogc:Div>
          <ogc:PropertyName>UNEMPLOY</ogc:PropertyName>
          <ogc:Add>
            <ogc:PropertyName>EMPLOYED</ogc:PropertyName>
            <ogc:PropertyName>UNEMPLOY</ogc:PropertyName>
          </ogc:Add>
       </ogc:Div>
       <ogc:Literal>100</ogc:Literal>
     </ogc:Mul>
  </ogc:Function>
</Label>
```



Format(“#.##”, UNEMPLOY / (EMPLOYED/UNEMPLOY))

2.0

GS

Geometry transformations

Not your grandpa's geometries

OpenGeo
http://opengeo.org

Geometry reference in SLD

- Each SLD/SE symbolizer has a “Geometry” element
- Used if you have many geometries among the attributes (not common)
- Has to be a ‹ogc:PropertyName›
- **Why? Can't I play with my geometry?**

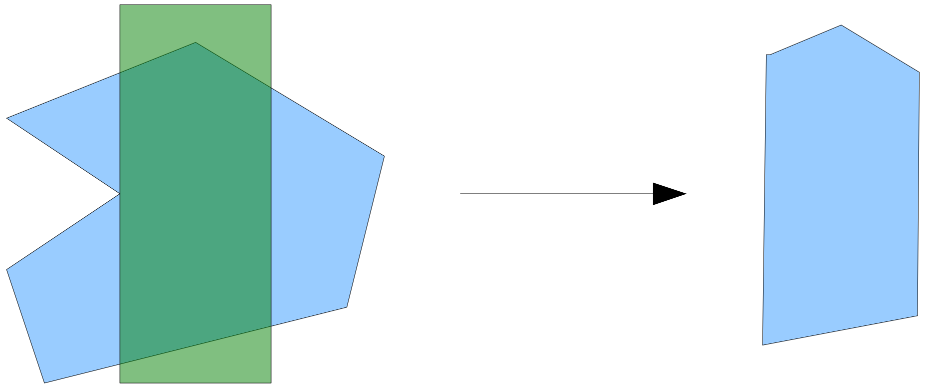
2.0

GS

OpenGeo
http://opengeo.org

Geometry transformations

- In GeoServer extended SLD, ‹Geometry› can be ogc:Function too
- You can transform the geometry before the renderer starts using it
- Extract vertexes, centroid, buffer, translate, intersect, ...

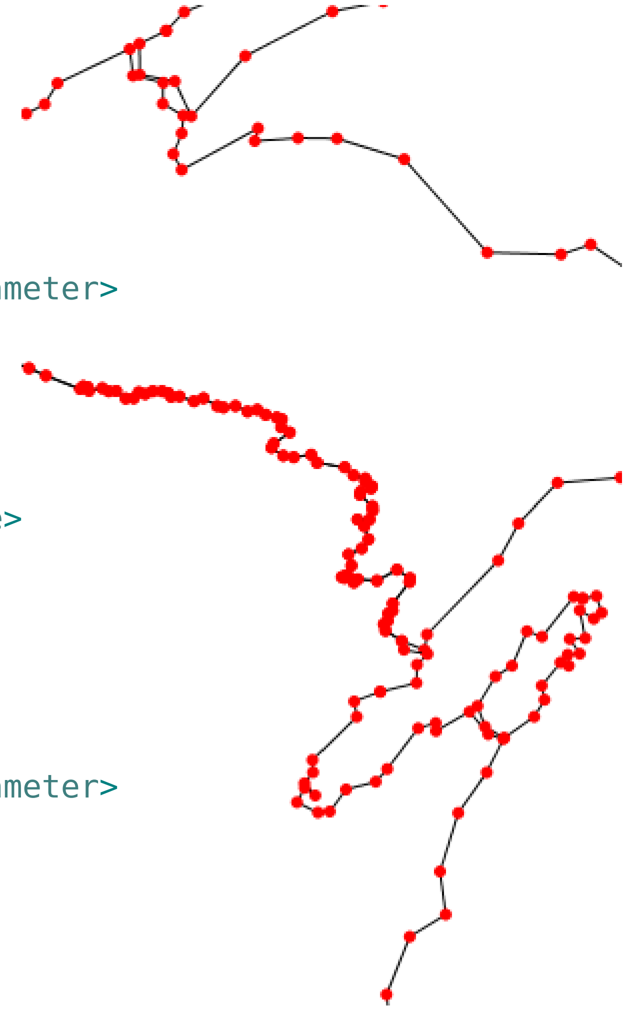


2.0

GS

OpenGeo
http://opengeo.org

```xml
<LineSymbolizer>
  <Stroke>
    <CssParameter name="stroke-width">0.5</CssParameter>
  </Stroke>
</LineSymbolizer>
<PointSymbolizer>
  <Geometry>
    <ogc:Function name="vertices">
      <ogc:PropertyName>the_geom</ogc:PropertyName>
    </ogc:Function>
  </Geometry>
  <Graphic>
    <Mark>
      <WellKnownName>circle</WellKnownName>
      <Fill>
        <CssParameter name="fill">#FF0000</CssParameter>
      </Fill>
    </Mark>
    <Size>6</Size>
  </Graphic>
</PointSymbolizer>
```

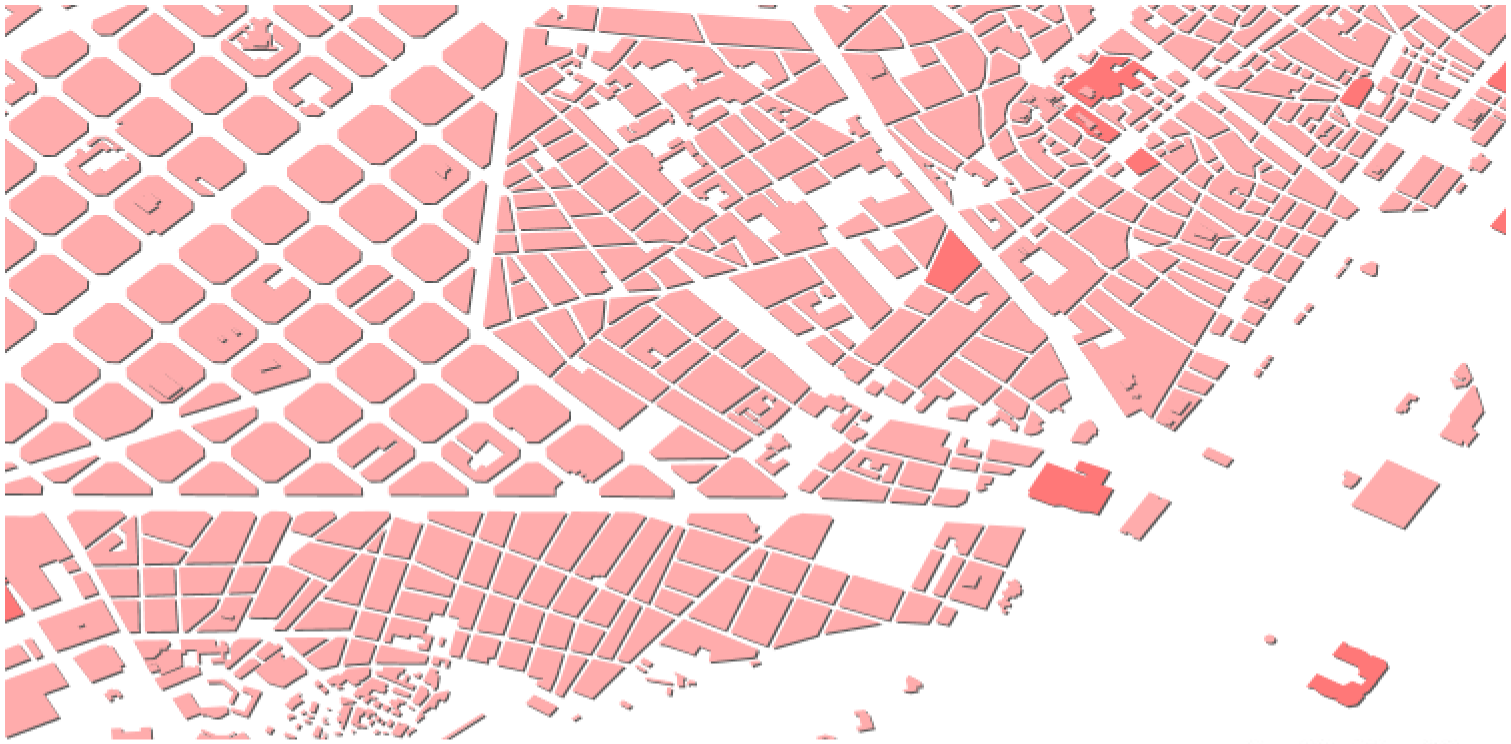


2.0

GS

OpenGeo
http://opengeo.org

```
<PolygonSymbolizer>
  <Geometry>
    <ogc:Function name="offset">
      <ogc:PropertyName>the_geom</ogc:PropertyName>
      <ogc:Literal>0.00004</ogc:Literal>
      <ogc:Literal>-0.00004</ogc:Literal>
    </ogc:Function>
  </Geometry>
  <Fill><CssParameter name="fill">#555555</CssParameter></Fill>
</PolygonSymbolizer>
<PolygonSymbolizer>
  <Fill><CssParameter name="fill">#ff7878</CssParameter></Fill>
</PolygonSymbolizer>
```



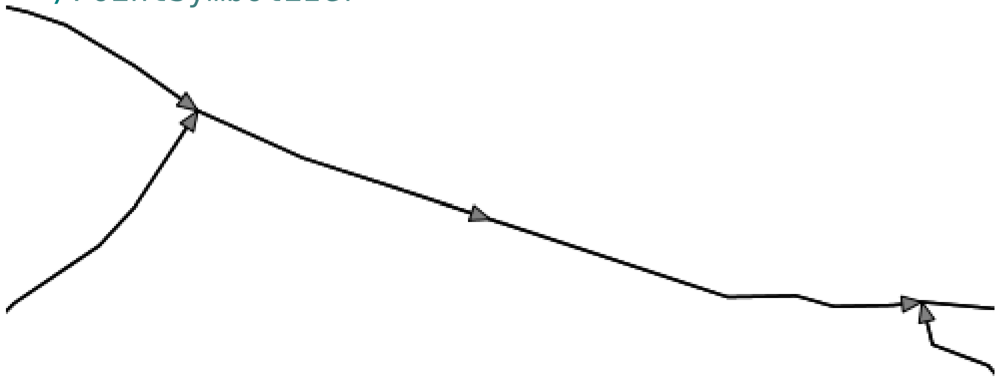
2.0

GS

```
<PointSymbolizer>
  <Geometry>
    <ogc:Function name="endPoint">
      <ogc:PropertyName>the_geom</ogc:PropertyName>
    </ogc:Function>
  </Geometry>
  <Graphic>
    <Mark>
      <WellKnownName>shape://carrow</WellKnownName>
      <Fill />
      <Stroke>
        <CssParameter name="stroke-width">1</CssParameter>
        <CssParameter name="stroke">#000000</CssParameter>
      </Stroke>
    </Mark>
    <Size>20</Size>
    <Rotation>
      <ogc:Function name="endAngle">
        <ogc:PropertyName>the_geom</ogc:PropertyName>
      </ogc:Function>
    </Rotation>
  </Graphic>
</PointSymbolizer>
```

Place a closed arrow at the end of the line

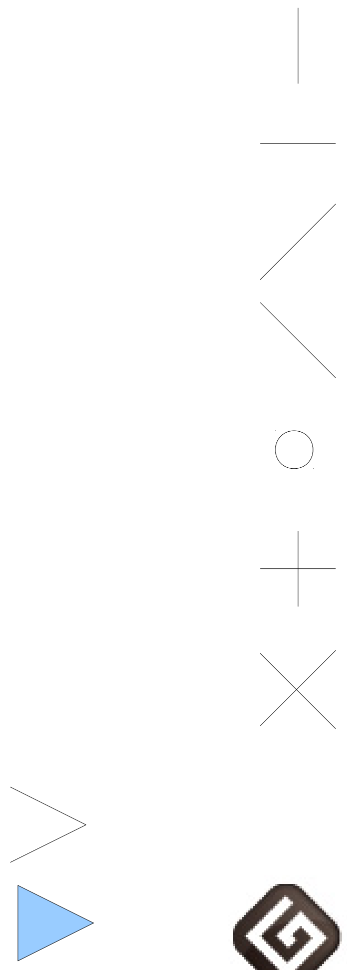
Rotate it along the line



OpenGeo
http://opengeo.org

The shape mark factory

- Simple general use shapes:
  - shape://vertline
  - shape://horline
  - shape://slash
  - shape://backslash
  - shape://dot
  - shape://plus
  - shape://times
  - shape://oarrow
  - shape://carrow

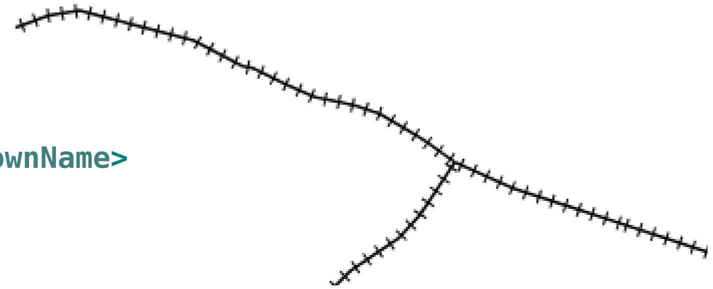


2.0

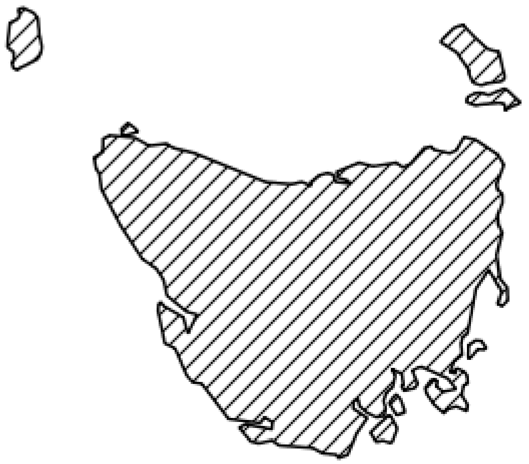
GS

OpenGeo
http://opengeo.org

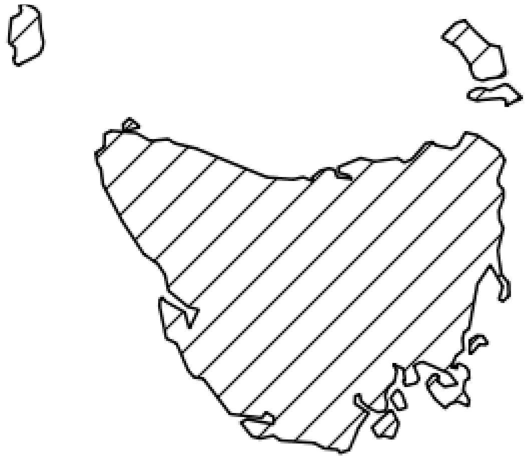
```
<LineSymbolizer><Stroke/></LineSymbolizer>
<LineSymbolizer>
  <Stroke>
    <GraphicStroke>
      <Graphic>
        <Mark>
          <WellKnownName>shape://vertline</WellKnownName>
          <Stroke/>
        </Mark><Size>7</Size>
      </Graphic>
    </GraphicStroke>
  </Stroke>
</LineSymbolizer>
```



```
<PolygonSymbolizer>
  <Fill>
    <GraphicFill><Graphic>
        <Mark>
          <WellKnownName>shape://slash</WellKnownName>
          <Stroke />
        </Mark><Size>10</Size>
    </Graphic></GraphicFill>
  </Fill>
  <Stroke/>
</PolygonSymbolizer>
```



```
<PolygonSymbolizer>
  <Fill>
    <GraphicFill><Graphic>
        <Mark>
          <WellKnownName>shape://slash</WellKnownName>
          <Stroke />
        </Mark><Size>20</Size>
    </Graphic></GraphicFill>
  </Fill>
  <Stroke/>
</PolygonSymbolizer>
```



2.0

GS

Map labeling

1001 vendor options

http://opengeo.org

SLD/SE status

- SLD/SE provides control for label along a line and position relative to a point
- Quite poor. What about:
  - Priority
  - Repetition
  - Label wrapping
  - Controlling placement heuristics
  - Mixing labels and graphics so that they behave as one (road plates)

GeoServer status

- More than a dozen vendor options to control and fine tune labeling
- Full list here: http://docs.geoserver.org/trunk/en/user/styling/sld-reference/labeling.html

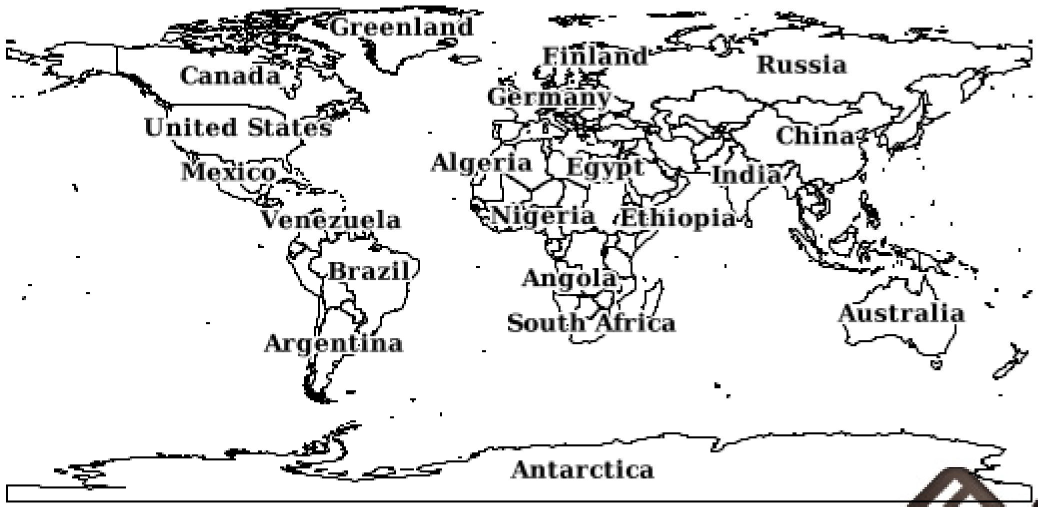
SW Fairmount Blvd
Marquam Nature Park
SW 13th Ave
SW 12th Ave
SW Marquam Hill Rd
SW Curry St
SW Mt Adams Dr
SW Gaines St
SW Marquam Hill Rd

OpenGeo
http://opengeo.org

Controlling priority

- ‹Priority› vendor element
- The higher the value, the sooner the label will be drawn (which makes it win in the conflict resolution game)

```
<Priority><ogc:PropertyName>POP2005</ogc:PropertyName></Priority>
```



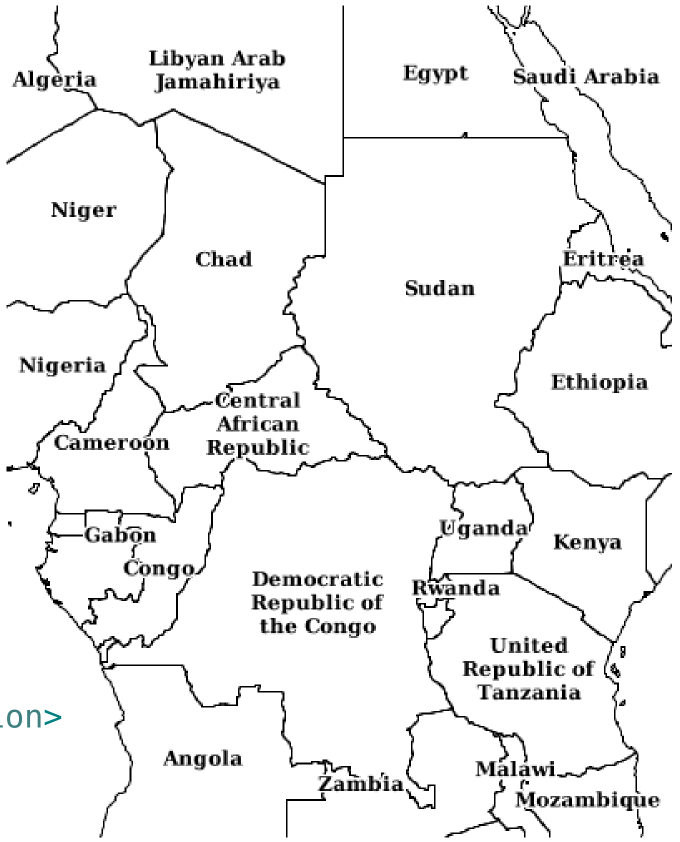
2.0

GS

Controlling label wrapping

- An option to wrap labels that exceed a certain length, in pixels

```
<VendorOption name="autoWrap">100</VendorOption>
```



2.0

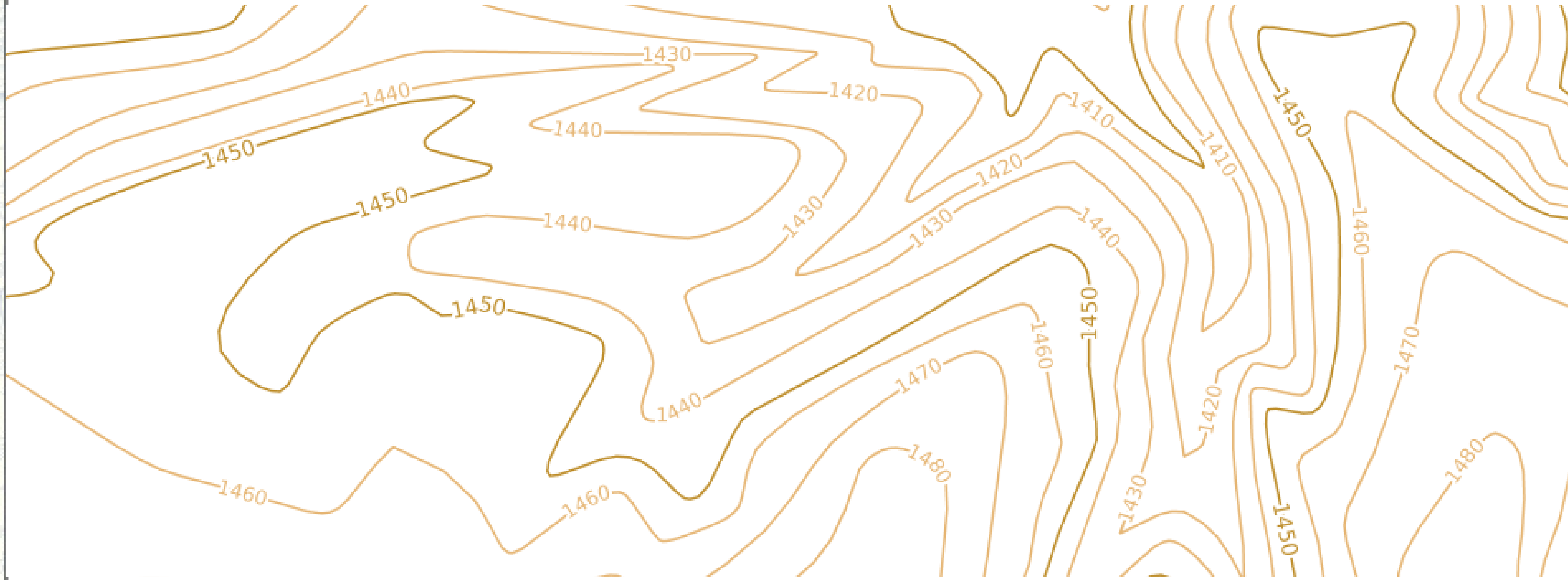
GS

OpenGeo
http://opengeo.org

Repeating and displacing

- Over long lines it's better to repeat the labels
- Displacing makes GS look for other places should the candidate label position be busy

```
<VendorOption name="followLine">true</VendorOption>
<VendorOption name="maxDisplacement">50</VendorOption>
<VendorOption name="repeat">300</VendorOption>
```



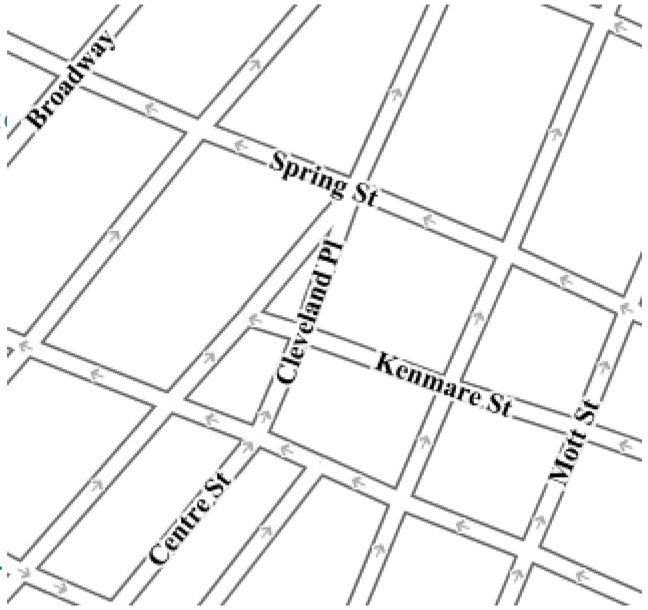
2.0

GS

Showing one way

- Labels are usually flipped to make them readable.
  If the char happens to be a directional arrow... that's not desirable

```
<TextSymbolizer>
   <Label>&#x2129;</Label>
   <Font>
     <CssParameter name="font-family">OpenSymbol</CssParamet
     <CssParameter name="font-size">10</CssParameter>
     <CssParameter name="font-weight">bold</CssParameter>
   </Font>
   <LabelPlacement>
     <LinePlacement>
     </LinePlacement>
   </LabelPlacement>
   <Halo>
     <Radius>
       <ogc:Literal>1</ogc:Literal>
     </Radius>
     <Fill>
       <CssParameter name="fill">#FFFFFF</CssParameter>
       <CssParameter name="fill-opacity">0.85</CssParameter>
     </Fill>
   </Halo>
   <Fill>
     <CssParameter name="fill">#AAAAAA</CssParameter>
   </Fill>
   <VendorOption name="maxDisplacement">100</VendorOption>
   <VendorOption name="forceLeftToRight">false</VendorOption>
 </TextSymbolizer>
```



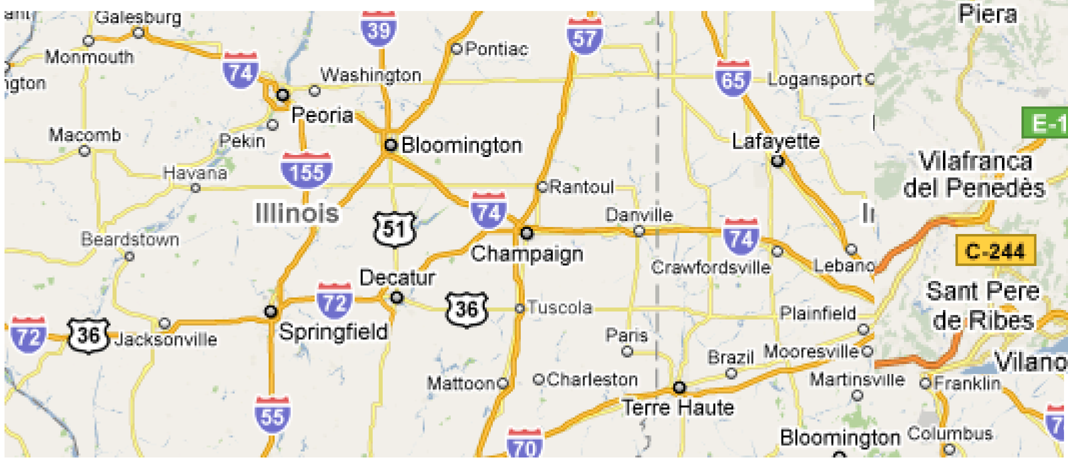
2.0

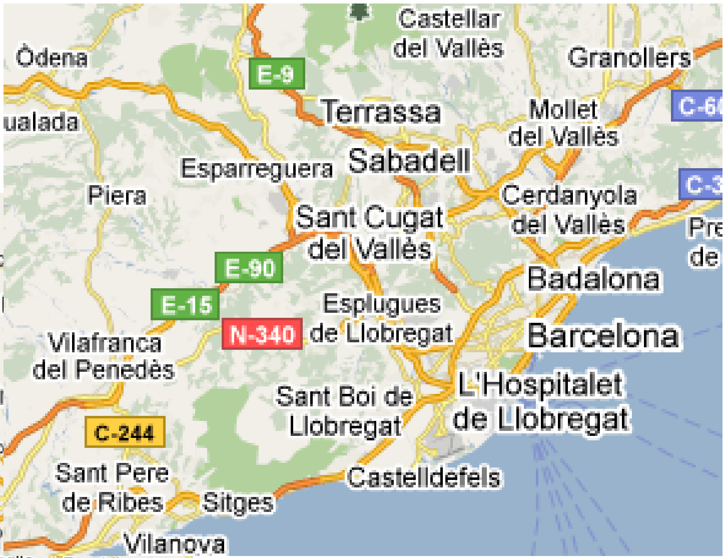
GS

OpenGeo
http://opengeo.org

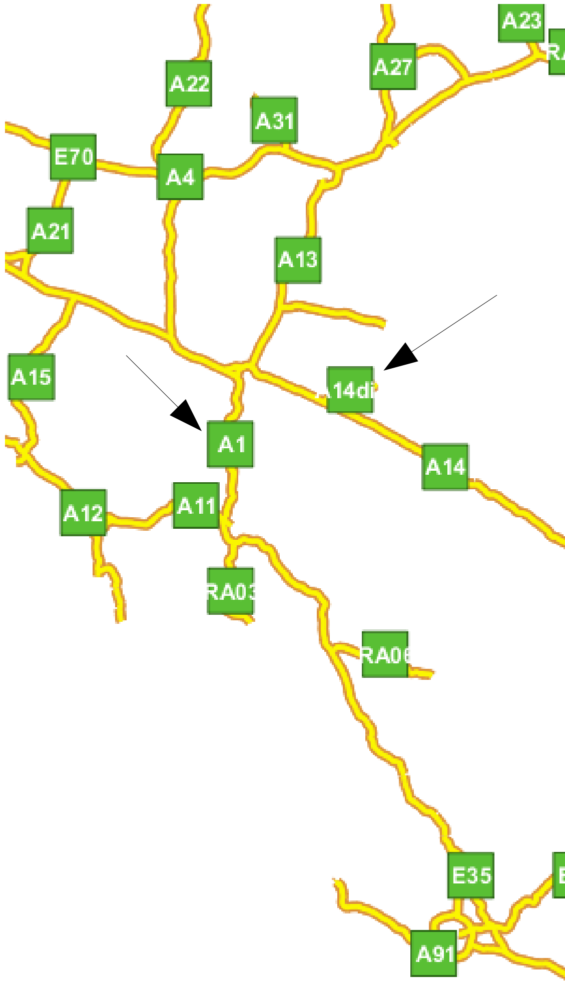
Mixing labels with graphics

- Typical case: road plate
- Either the road plate and the label show together, or none of them should
- Solution: include a Graphic element inside the TextSymbolizer!





```xml
<TextSymbolizer>
  <Label>
    <ogc:PropertyName>Nome_Secon</ogc:PropertyName>
  </Label>
  <Font>
    <CssParameter name="font-family">Arial</CssParameter>
    <CssParameter name="font-size">12</CssParameter>
    <CssParameter name="font-weight">bold</CssParameter>
  </Font>
  <LabelPlacement>
    <PointPlacement>
      <AnchorPoint>
        <AnchorPointX>0.5</AnchorPointX>
        <AnchorPointY>0.5</AnchorPointY>
      </AnchorPoint>
    </PointPlacement>
  </LabelPlacement>
  <Fill>
    <CssParameter name="fill">#FFFFFF</CssParameter>
  </Fill>
  <Graphic>
    <Mark>
      <WellKnownName>square</WellKnownName>
      <Fill>
        <CssParameter name="fill">#59BF34</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#2D6917</CssParameter>
      </Stroke>
    </Mark>
    <Size>24</Size>
  </Graphic>
</TextSymbolizer>
```



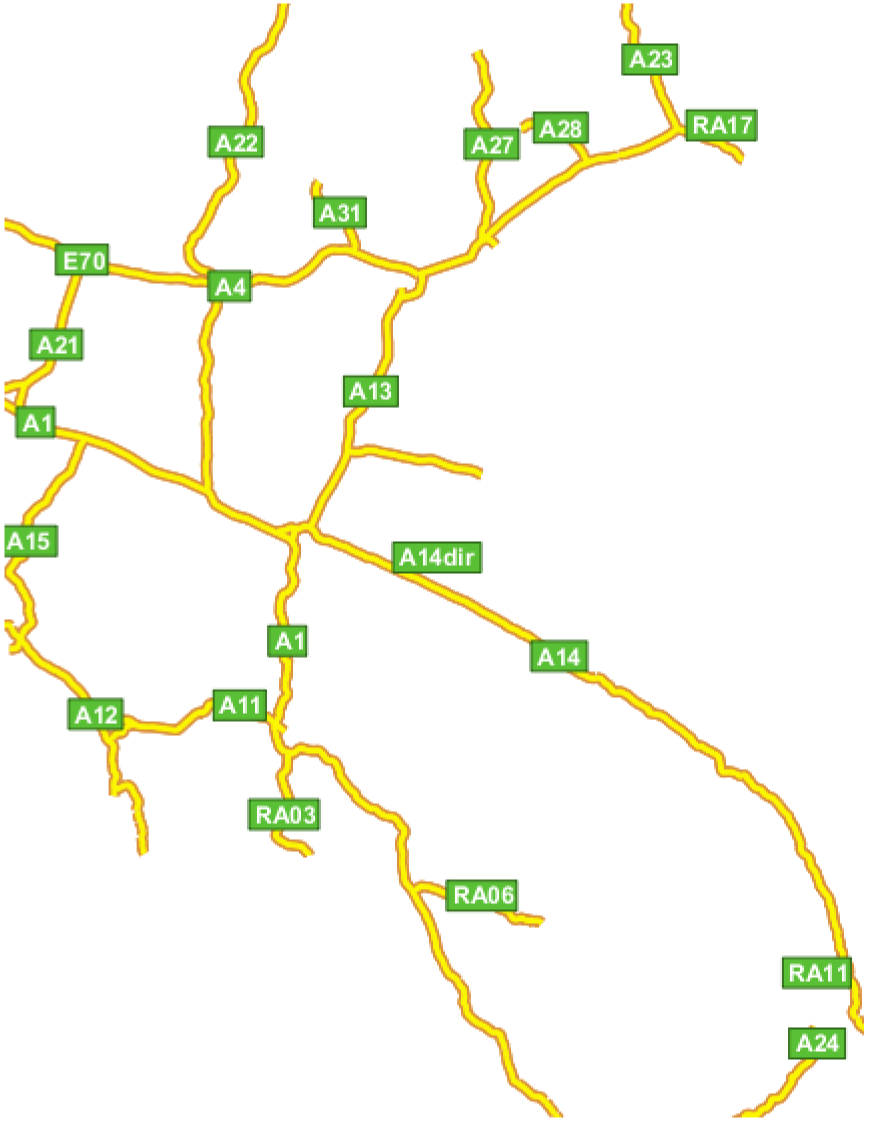
2.0

GS

Problem: the graphic size is fixed, the text one is dynamic! We could stretch it

OpenGeo
http://opengeo.org

```
<VendorOption name="graphic-resize">stretch</VendorOption>
<VendorOption name="graphic-margin">3</VendorOption>
```



Resize mode: none, proportional, stretch

2.1

GS

OpenGeo http://opengeo.org

Questions?

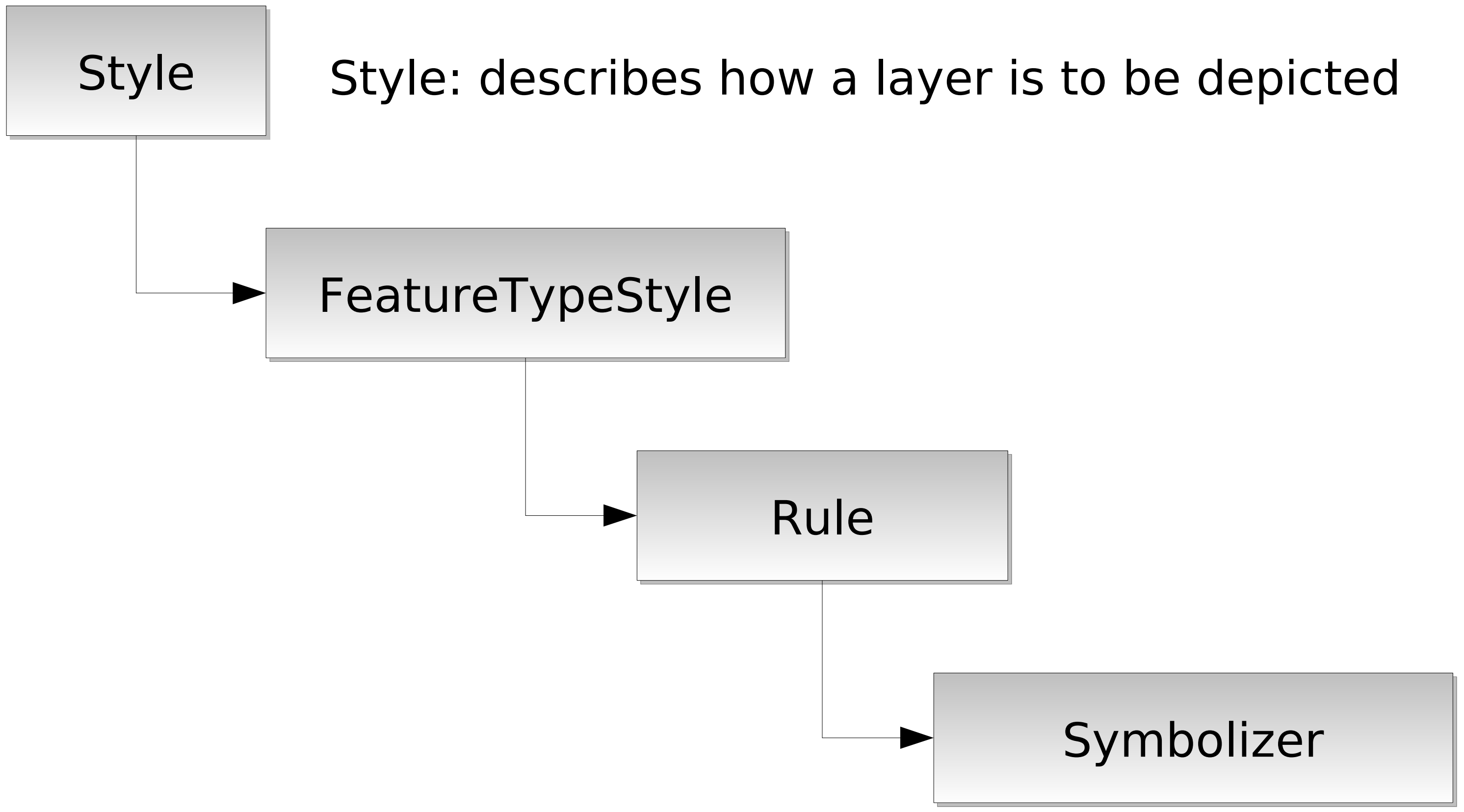
OpenGeo
http://opengeo.org

Extras

Lightning intro to SLD

http://opengeo.org

SLD basic elements



Style: describes how a layer is to be depicted

http://opengeo.org

FeatureTypeStyle

- “The FeatureTypeStyle defines the styling that is to be applied to a single feature type”
- “A map styler is expected to process all FeatureTypeStyles in the order that they appear, regardless, plotting one instance over top of another” (painter model)
- → Used mostly to force certain drawing order

OpenGeo
http://opengeo.org

Rule

- "Rules are used to group rendering instructions by feature-property conditions and map scales"
- So:
  - Scale dependencies
  - Filter by attribute
  - Rendering instructions that apply under the above conditions → symbolizers

OpenGeo
http://opengeo.org

Symbolizer

- "A Symbolizer describes how a feature is to appear on a map. The Symbolizer describes not just the shape that should appear but also such graphical properties as color and opacity."
- Five types of symbolizers:
  - Point: symbol, size, color, ...
  - Line: width, color, graphics along a line
  - Polygon: outline, fill (solid color or graphic based)
  - Text: label, font, placement
  - Raster: color table, gamma, histogram, ...
- A rule can contain multiple symbolizers

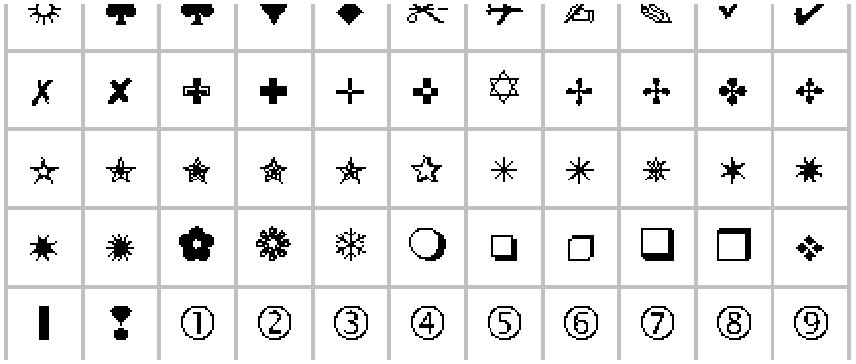
Dynamic symbolizers

Breaking out the mark and graphic cage

http://opengeo.org

Marks in SLD/SE

- Mark: a shape to be filled and stroked
- SLD 1.0:
  - “square”, “circle”, “triangle”, “star”, “cross”, and “x”
- SE 1.1: same, but also external symbol source and “mark index” (e.g. a decorative font + index inside of it)



OpenGeo
http://opengeo.org

Marks in GeoServer

- The well known name is a string, so it's **open ended**
- Our convention: `factory://name`
- Two factories available today:
  - shape
  - ttf
- More could be implemented, the API is pluggable

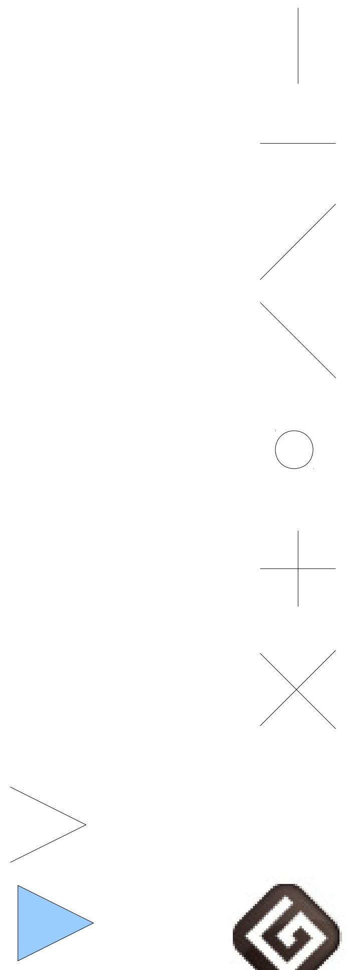
2.0

GS

OpenGeo
http://opengeo.org

The shape mark factory

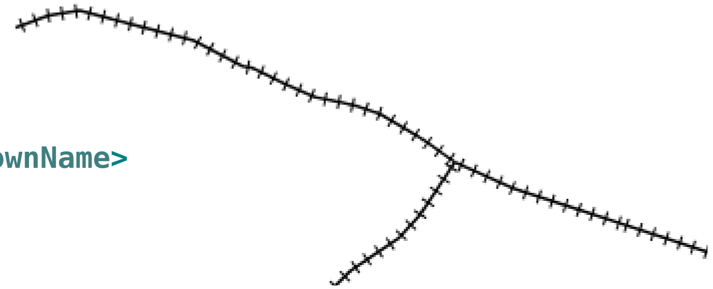
- Shapes intended to be hatch generators:
  - shape://vertline
  - shape://horline
  - shape://slash
  - shape://backslash
  - shape://dot
  - shape://plus
  - shape://times
  - shape://oarrow
  - shape://carrow



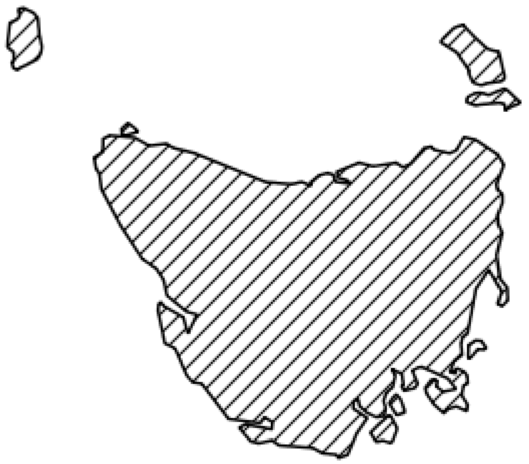
2.0

GS

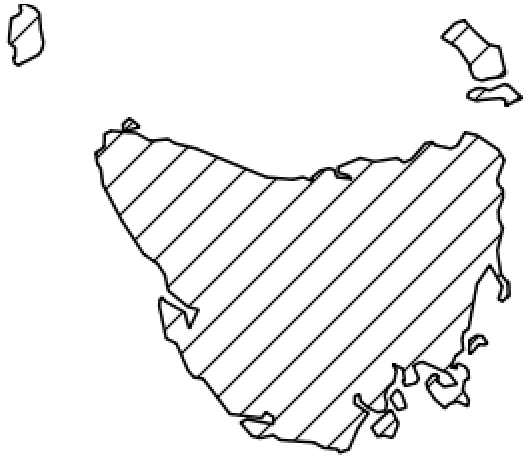
```
<LineSymbolizer><Stroke/></LineSymbolizer>
<LineSymbolizer>
  <Stroke>
    <GraphicStroke>
      <Graphic>
        <Mark>
          <WellKnownName>shape://vertline</WellKnownName>
          <Stroke/>
        </Mark><Size>7</Size>
      </Graphic>
    </GraphicStroke>
  </Stroke>
</LineSymbolizer>
```



```
<PolygonSymbolizer>
  <Fill>
    <GraphicFill><Graphic>
        <Mark>
          <WellKnownName>shape://slash</WellKnownName>
          <Stroke />
        </Mark><Size>10</Size>
    </Graphic></GraphicFill>
  </Fill>
  <Stroke/>
</PolygonSymbolizer>
```



```
<PolygonSymbolizer>
  <Fill>
    <GraphicFill><Graphic>
        <Mark>
          <WellKnownName>shape://slash</WellKnownName>
          <Stroke />
        </Mark><Size>20</Size>
    </Graphic></GraphicFill>
  </Fill>
  <Stroke/>
</PolygonSymbolizer>
```



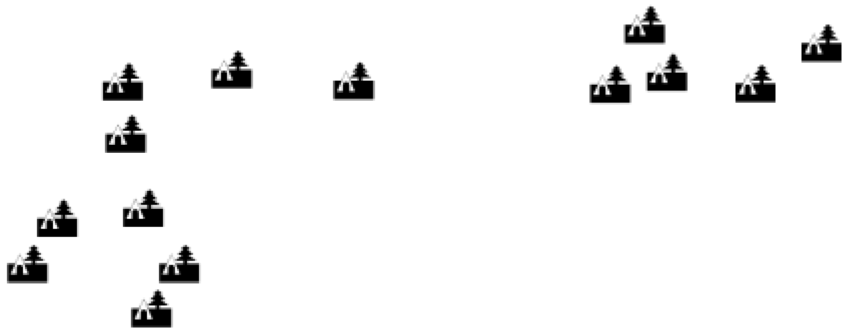
2.0

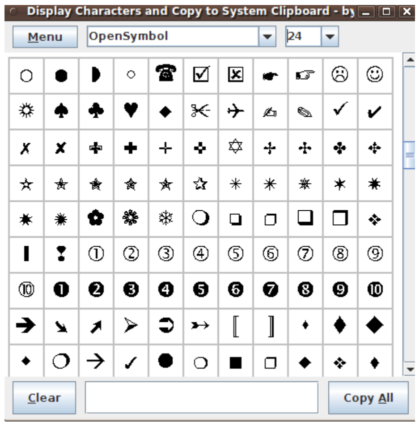
GS

The TTF mark factory

- Generates shapes out of decorative fonts
- Format is `ttf://fontname#charcode`

```
<PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>ttf://Webdings#0x0051</WellKnownName>
      <Fill>
         <CssParameter name="fill">#000000</CssParameter>
      </Fill>
    </Mark>
    <Size>20</Size>
  </Graphic>
</PointSymbolizer>
```





2.0

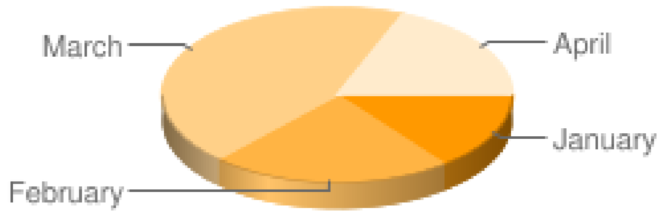
GS

OpenGeo
http://opengeo.org

External graphics

- URL to an image
- URL cannot have parameters → static image only!
- Compare with Google chart API → dynamic image!

```
http://chart.apis.google.com/chart?
cht=p3&chd=s:Uf9a&chs=250x100
&chl=January|February|March|April
```



OpenGeo http://opengeo.org

Enter dynamic symbolizers

- Dynamic symbolizers: **expand attribute names inside mark names and graphic URLs**
- **Expand full CQL expressions** (making math, formatting strings, calling functions)
- `${expression}/ ${attributeName}`

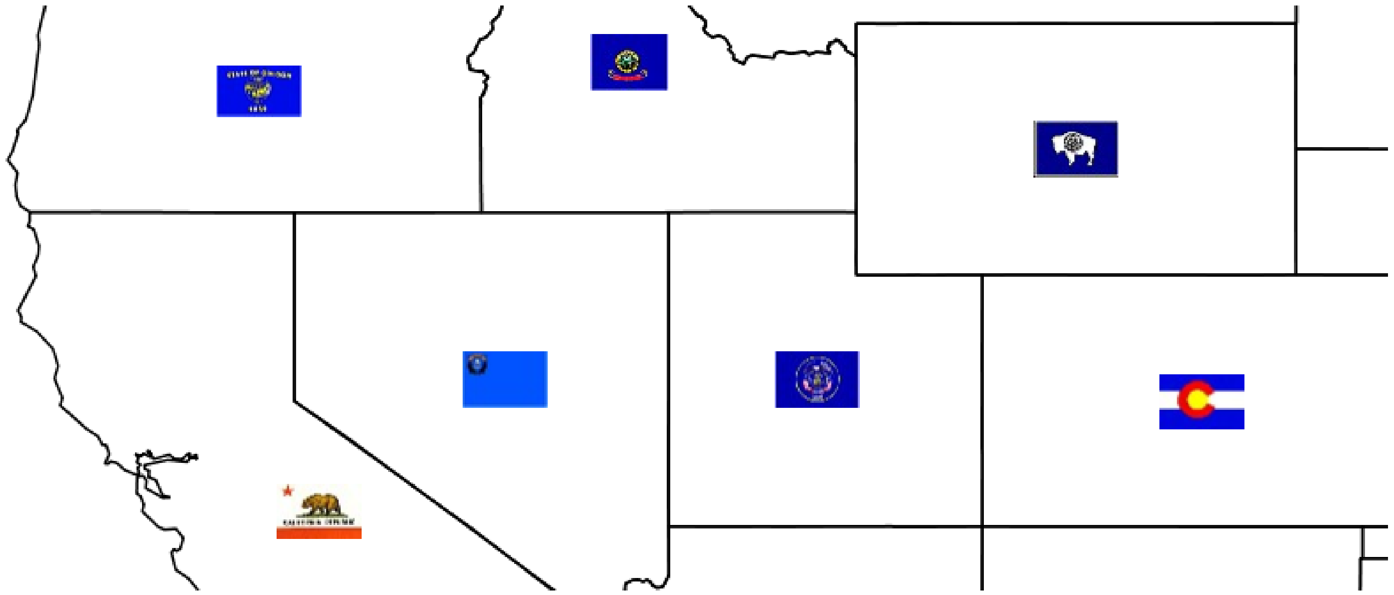
2.0

GS

OpenGeo
http://opengeo.org

Calling a filter function to lower case the state abbreviation

```
<ExternalGraphic>
  <OnlineResource xlink:type="simple"
    xlink:href="http://www.usautoparts.net/tn_${strToLowerCase(STATE_ABBR)}.jpg" />
  <Format>image/jpeg</Format>
</ExternalGraphic>
```



2.0

GS