APACHECON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014

Apache SIS and standards in Geospatial information

Martin Desruisseaux
martin.desruisseaux@geomatys.com

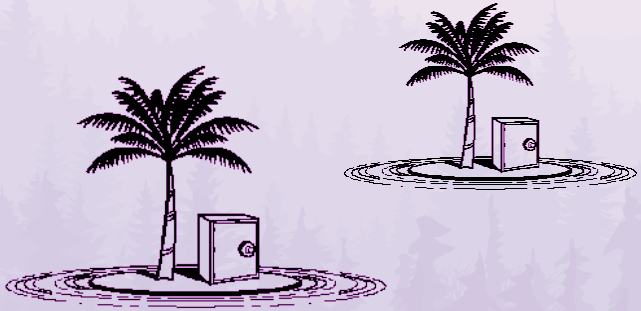
Presented For The Apache Foundation By
LINUX FOUNDATION

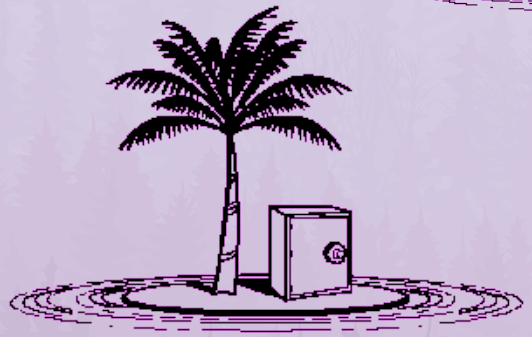
Agenda

- The barrier in geospatial data exchanges
- The Open Geospatial Consortium (OGC)
- The metadata model (ISO 19115)
- The referencing model (ISO 19111)
- From UML to Java interfaces: GeoAPI
- Apache Spatial Information System (SIS)
- Roadmap

Interoperability issues

- Incompatible data
- Incompatible system
- Data fragmentation
- Redundancy





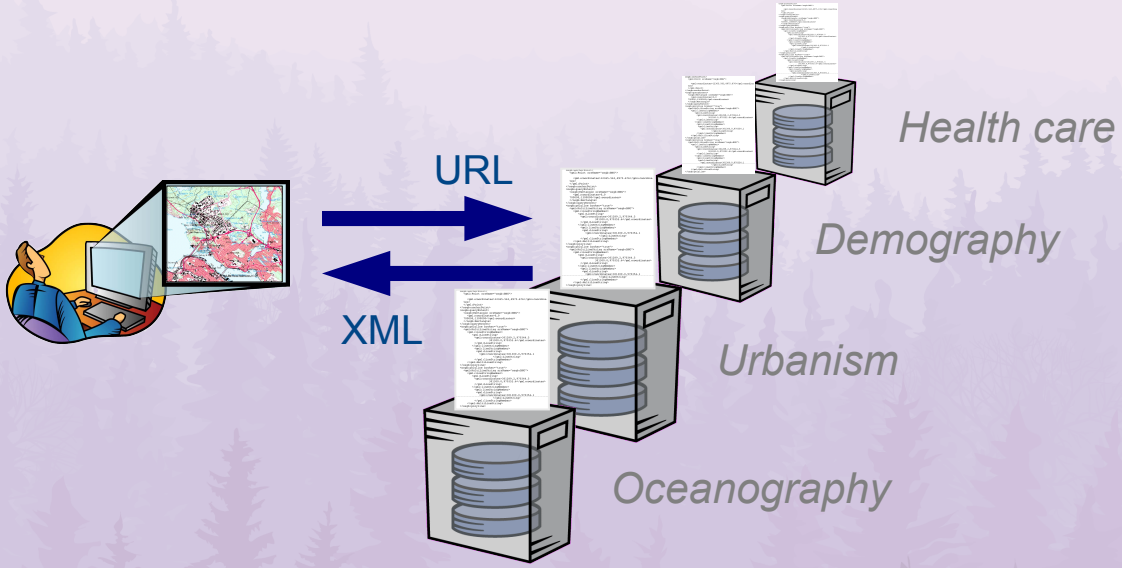
Need for:

- Share maps on the web
- Deliver data to different systems easily
- Common language for geospatial data and services

Use case: cholera

Propagation depends on:

- Population density
- Population migration
- City infrastructure
  (water supply and disposal)
- Copepod and shrimp shell



Some communities are not well aware of each other. → Mixing data are difficult.

Open Geospatial Consortium

APACHECON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014

Vision:

A world in which everyone benefits from the use of geospatial information and supporting technologies.

Mission:

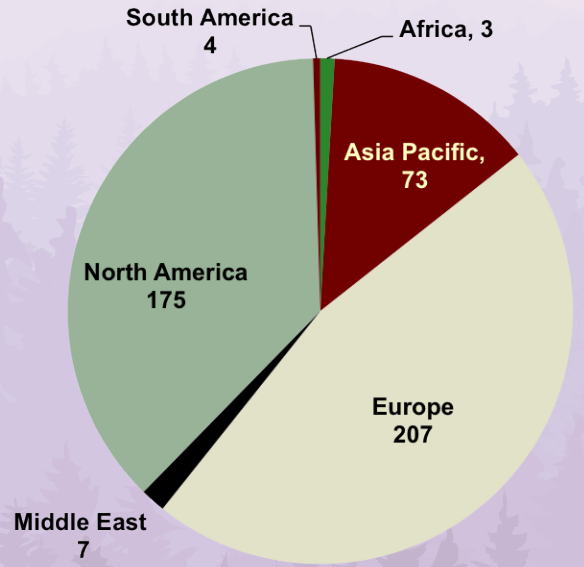
Advance the development and use of international standards and supporting services that promote geospatial interoperability.

http://www.opengeospatial.org

The OGC at a glance

Not-for-profit, international voluntary consensus standards organization; leading development of geospatial standards.

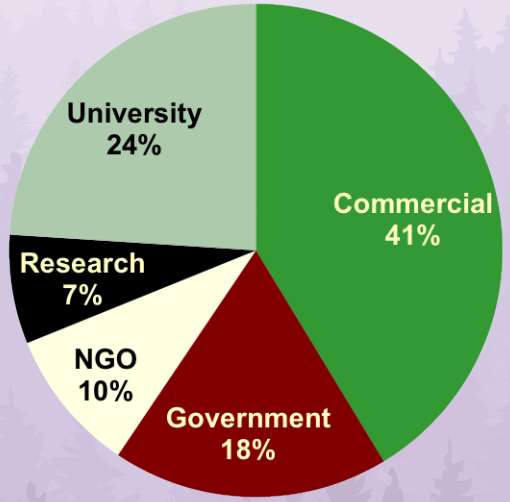
- Founded in 1994
- 470+ members and growing
- 40 standards
- Hundreds of product implementations
- Broad user community implementation worldwide
- Collaborative activities with professional associations



The OGC at a glance

Not-for-profit, international voluntary consensus standards organization; leading development of geospatial standards.

- Founded in 1994
- 470+ members and growing
- 40 standards
- Hundreds of product implementations
- Broad user community implementation worldwide
- Collaborative activities with professional associations



Alliance partners

… and others

What OGC provide

Standards

- Defined by UML
- Often jointly with ISO

XML schemas

- Derived from UML
- http://schemas.opengis.net/

```
<complexType name="GeodeticCRSType">
 <complexContent>
  <extension base="AbstractCRSType">
   <sequence>
    <choice>
     <element ref="ellipsoidalCSProperty"/>
     <element ref="cartesianCSProperty"/>
     <element ref="sphericalCSProperty"/>
    </choice>
    <element ref="geodeticDatumProperty"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>
```

CITE tests

- Web service tests
- File validations

UML as our source for API

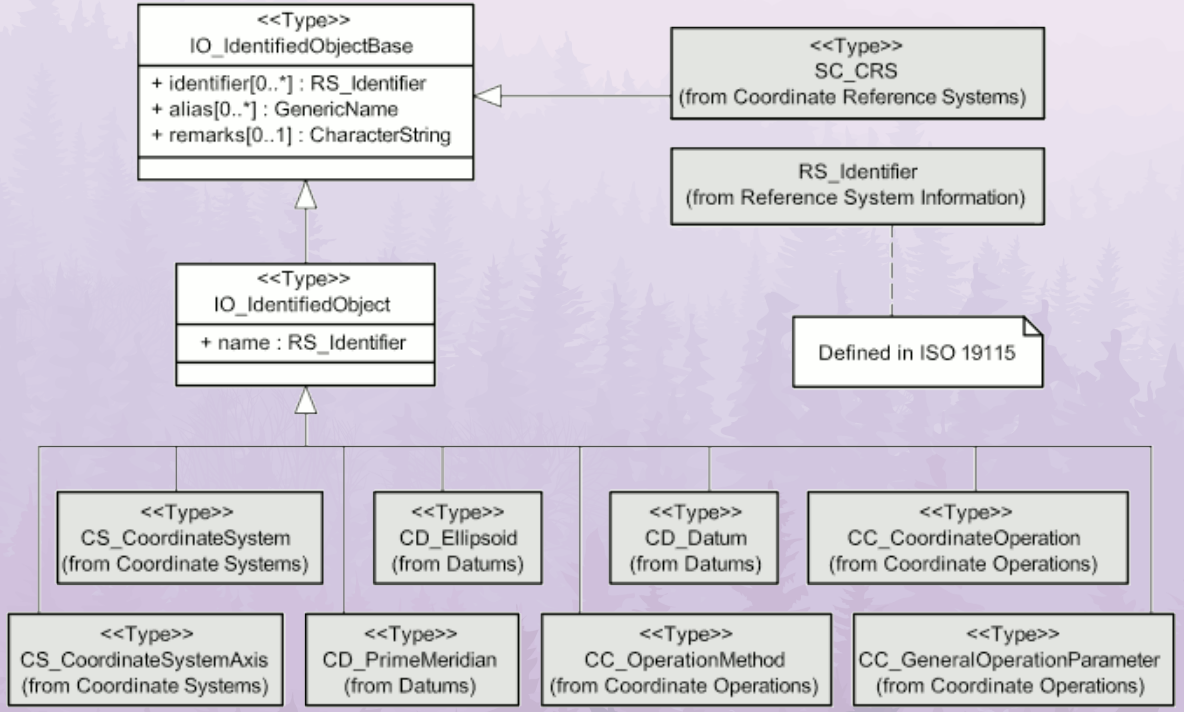
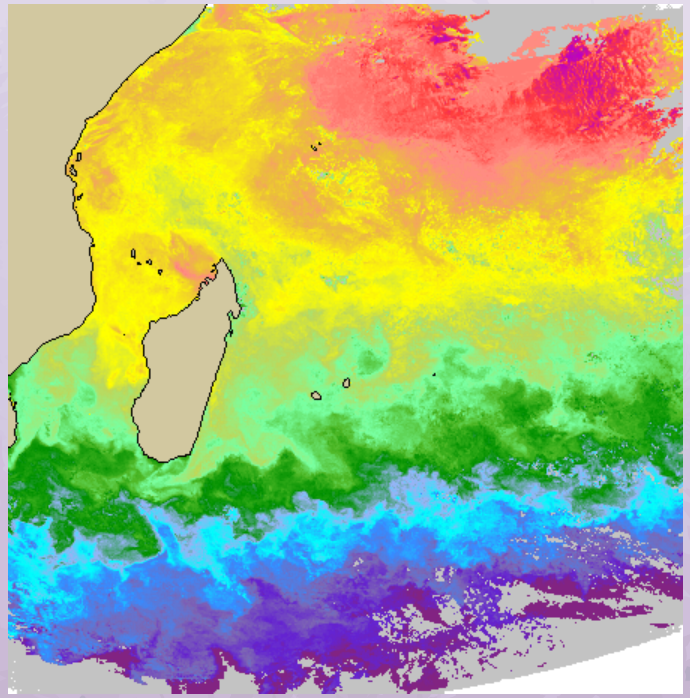


Figure 5 — IO_IdentifiedObject package

Metadata

Metadata

Data example:



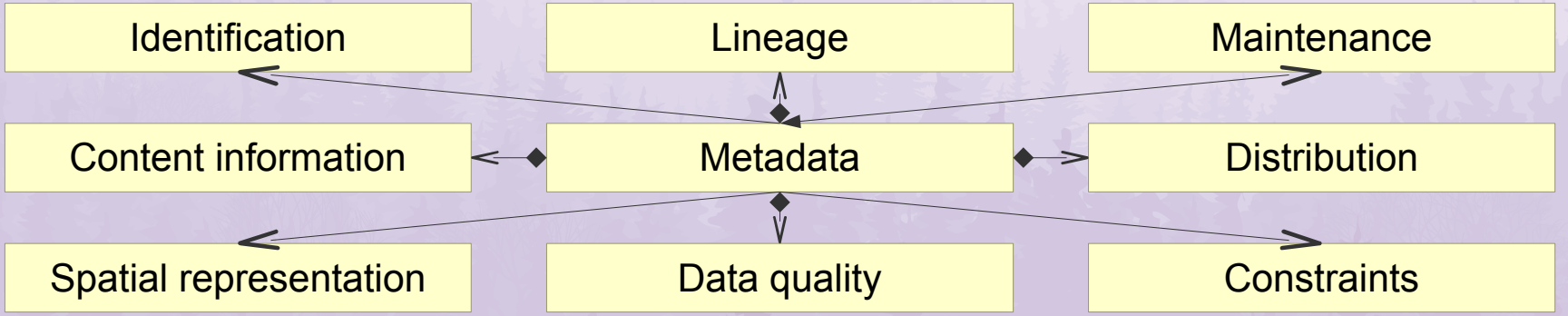
But...

- What is measured?
- Where and when is it?
- What is the quality?
- What are the usage restrictions?

Without metadata, geospatial data are useless

ISO 19115 model

- Describe information or resource that can have geographic extents
- 156 interfaces or enumerations (GeoAPI 3.0)



Root packages:

- `org.opengis.metadata` for the interfaces
- `org.apache.sis.metadata.iso` for the implementation classes

Metadata.toString() example

```
Metadata
  ├─Metadata standard name……… ISO 19115-2 Geographic Information
  ├─Metadata standard version……… ISO 19115-2:2009(E)
  ├─Hierarchy level……… Dataset
  ├─Identification info
  │   ├─Citation
  │   │   ├─Title……… Downscaling data from CERFACS
  │   │   └─Cited responsible party
  │   │       ├─Individual name……… Global Change Team
  │   │       ├─Organisation name……… CERFACS
  │   │       ├─Contact info
  │   │       │   ├─Address
  │   │       │   │   └─Electronic mail address…… globc@cerfacs.fr
  │   │       │   └─Online resource
  │   │       │       ├─Linkage……… http://www.cerfacs.fr/globc/
  │   │       │       ├─Protocol……… http
  │   │       │       ├─Application profile……… web browser
  │   │       │       └─Function……… Information
  │   │       └─Role……… Originator
  │   ├─Descriptive keywords
  │   │   ├─Keyword (1 of 4)……… climate
  │   │   ├─Keyword (2 of 4)……… scenarios
  │   │   ├─Keyword (3 of 4)……… downscaling
  │   │   ├─Keyword (4 of 4)……… CERFACS
  │   │   └─Type……… Theme
  │   └─Extent
  │       └─Geographic element
  │           ├─West bound longitude……… 5°19′44.96990203857″W
  │           ├─East bound longitude……… 10°47′34.12445068359″E
  │           ├─South bound latitude……… 41°19′05.4242177611″N
  │           └─North bound latitude……… 51°07′17.4568587976″N
  ├─Content info
  │   └─Dimension
  │       ├─Sequence identifier……… tas
  │       └─Descriptor……… air_temperature
```

Since 2013 in Europe, new data produced by governments shall have their metadata published

(the INSPIRE initiative)

Referencing by coordinates

From Earth to map

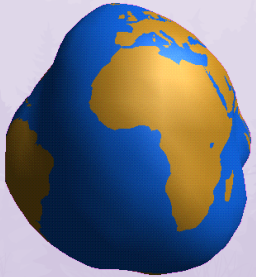
Earth

Difference:

- 9,000 meters for highest mountains
- 11,000 meters for deepest abysses

Geoid

Close to mean sea level



Difference:

- < 200 meters
- Depends on the chosen ellipsoid

Ellipsoid

Idealized surface

Map projection

Distortions:

- Angles or surfaces
- Distances, *etc.*

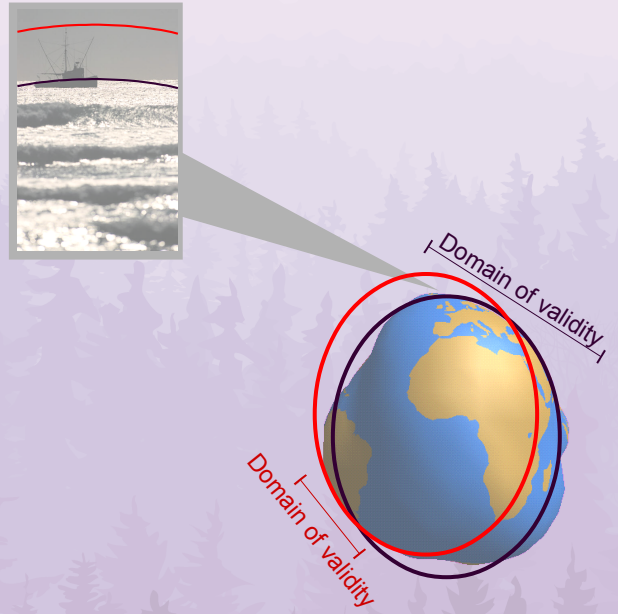
Ellipsoid and datum

Ellipsoid:

- Semi-major axis (equatorial radius)
- Semi-minor axis (polar radius)

Datum:

- Ellipsoid center relative to Earth's center of mass
- Ellipsoid axis orientations
- Established by national or scientific authorities
- May be global or valid only in a local area



454 geodetic datum in EPSG database

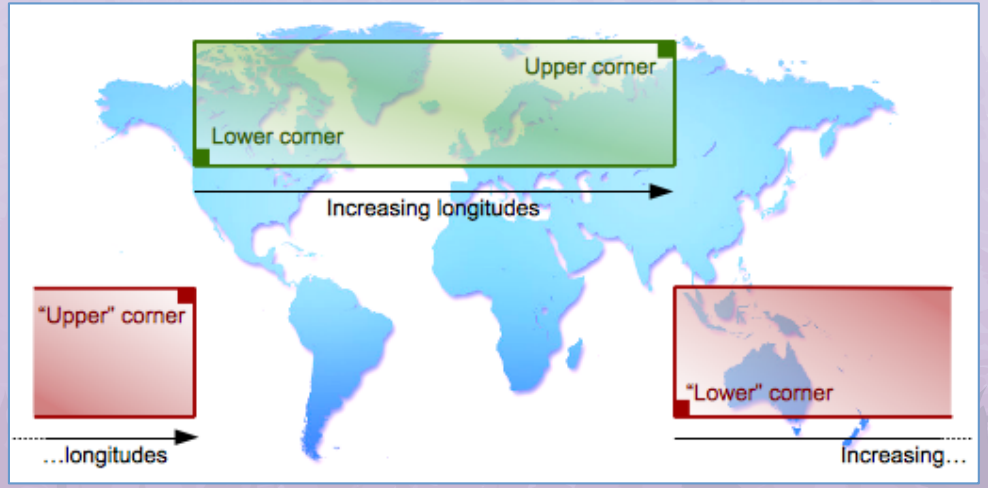
Differences from centimeters to a few kilometers

Coordinate system

- Axes with their ranges, units and abbreviation
- Mathematical relationships (distance, angle)

Ranges may wraparound

Not only longitude (e.g. climatology calendar)



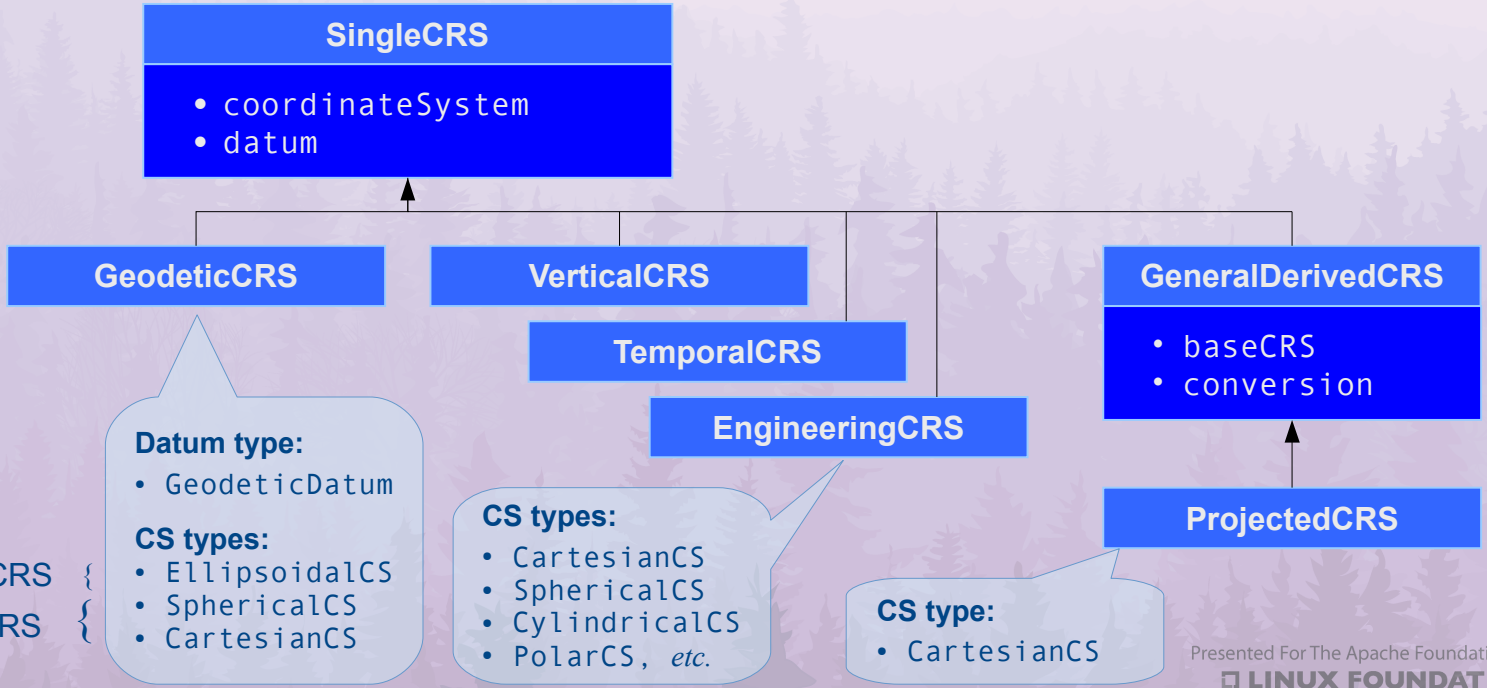
Names are constrained:

- Geodetic latitude (φ)
- Geodetic longitude (λ)
- Gravity-related height (h)
- Easting or Westing (E, W)
- Northing or Southing (N, S)
- *… others …*

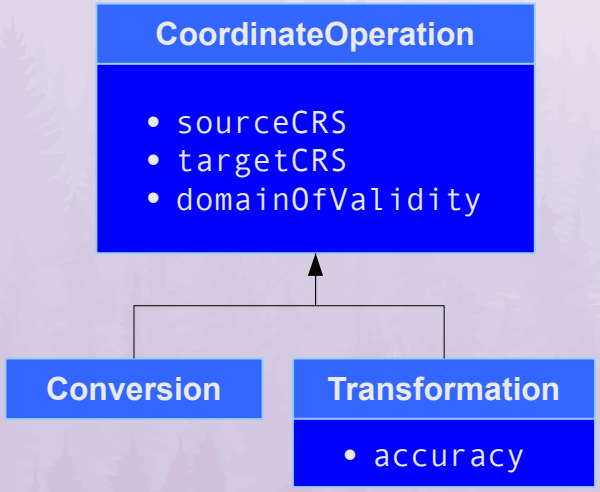
Up to 20 km difference between *geodetic* and *geocentric* latitude

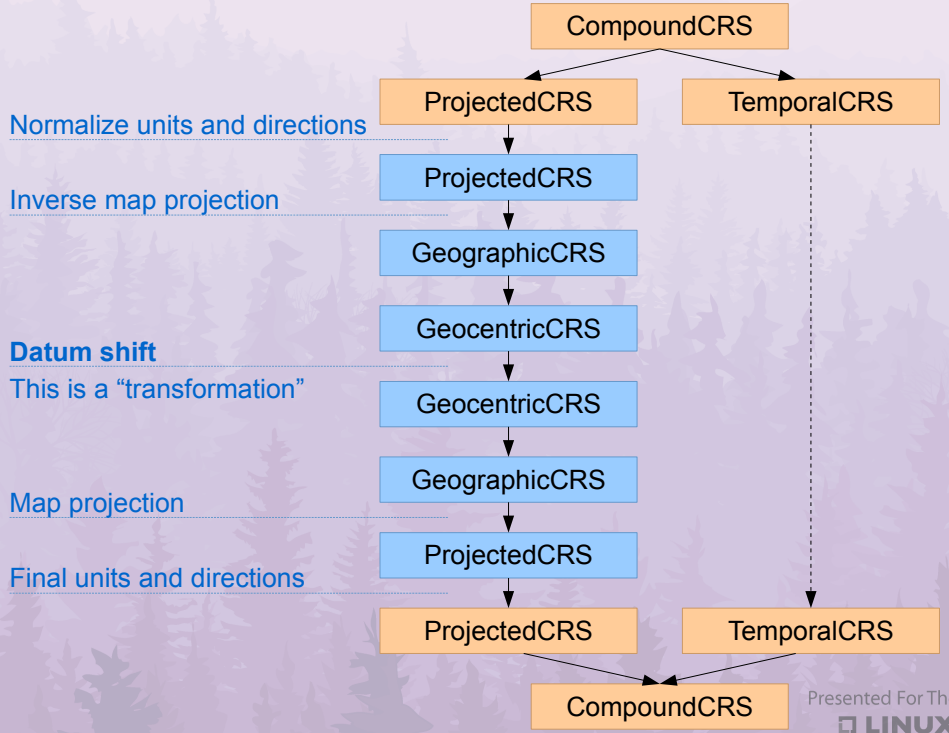
Coordinate reference system

- Coordinate system related to Earth through datum
- 4906 definitions in EPSG database



Coordinate operation





Hide the complexity

Use EPSG database

http://epsg-registry.org/

⚠ Be cautious with other sites

```
CoordinateReferenceSystem crs = CRS.forCode("EPSG:3395");
```

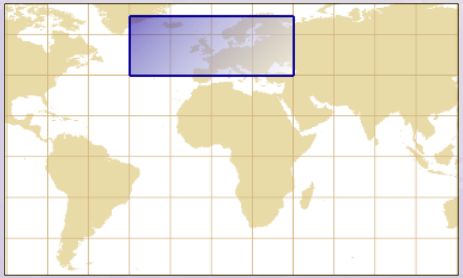
Everything needed for a World Mercator projection

Let SIS figures out the transformation path (targeted for SIS 0.5)

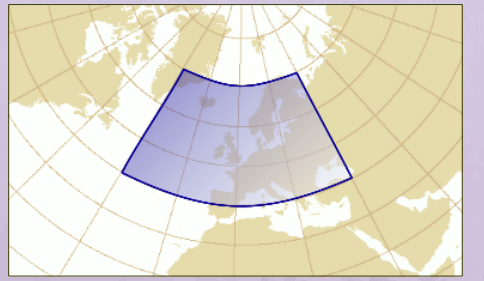
```
CoordinateOperation op = CRS.findOperation(
        sourceCRS, targetCRS, areaOfInterest);
```

Envelope transformation

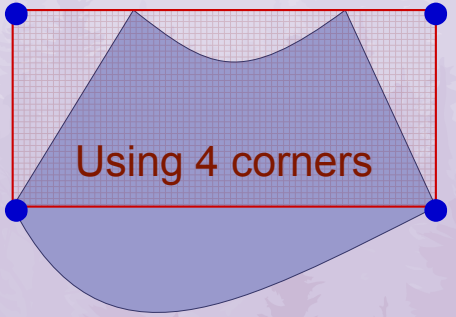
Source

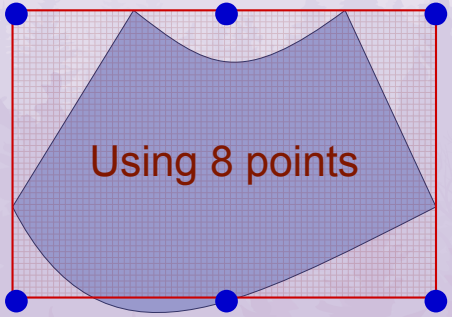


Target



- Target envelope shall contain fully the theoretical shape
- Common approach is to sample many points

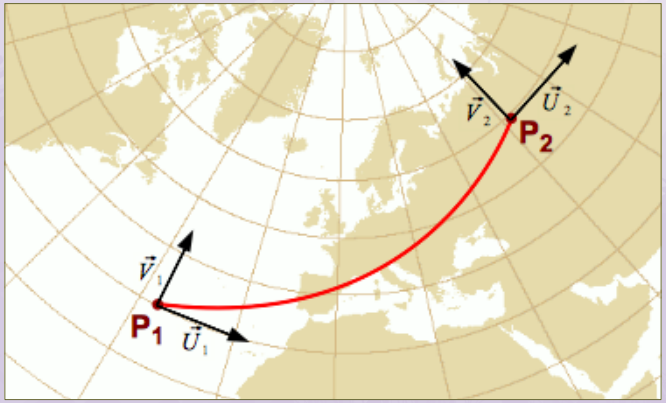


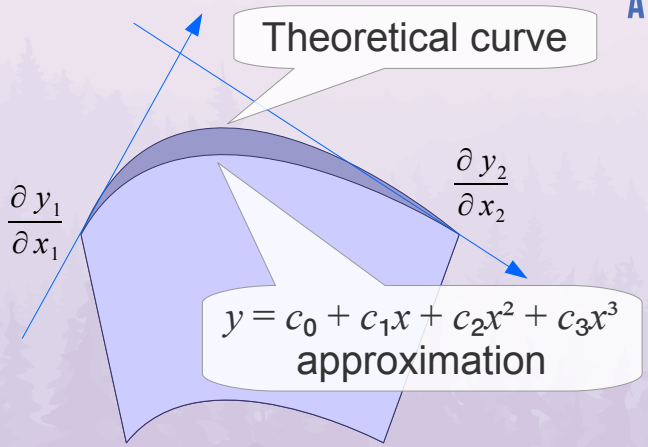


Compromise hard to find

Some libraries use 160 points

Using transform derivatives



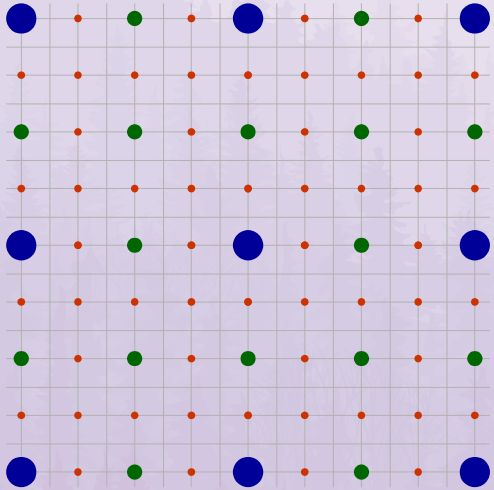


1. Given two points and their derivatives, get the equation of a cubic curve
2. Locate the curve's extrema using the derivative of above cubic equation
3. Transform only the point aligned with that extrema

Fast, accurate, a SIS 0.5 advantage

16 points + derivatives often more accurate than sampling 160 points

Raster resampling



- Video cards are very fast with affine transforms.
- Transforming individual pixels in Java is slow.
- Map projections are not affine transforms,
  but small blocks are approximatively affine.
- Iteratively decrease blocks size until they are affine enough.
- Each pass requires ~3 times more points than all previous passes.

Alternative:

- *Without derivatives,* final pass is only to figure out that it was not necessary.
- *With derivatives*, we can estimate if we need an other pass before doing it,
  thus using only ⅓ the amount of points of first alternative.

APACHECON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014

From UML to Java API

Presented For The Apache Foundation By
LINUX FOUNDATION

Metadata — XML fragment


```
<gmd:MD_Metadata>
  <gmd:identificationInfo>
    <gmd:MD_DataIdentification>
      <gmd:citation>
        <gmd:CI_Citation>
          <gmd:citedResponsibleParty>
            <gmd:CI_ResponsibleParty>
              <gmd:contactInfo>
                <gmd:CI_Contact>
                  <gmd:onlineResource>
                    <gmd:CI_OnlineResource>
                      <gmd:linkage>
                        <gmd:URL>http://www.opengeospatial.org</gmd:URL>
                      </gmd:linkage>
                    </gmd:CI_OnlineResource>
                  </gmd:onlineResource>
                </gmd:CI_Contact>
              </gmd:contactInfo>
            </gmd:CI_ResponsibleParty>
          </gmd:citedResponsibleParty>
        </gmd:CI_Citation>
      </gmd:citation>
    </gmd:MD_DataIdentification>
  </gmd:identificationInfo>
</gmd:MD_Metadata>
```

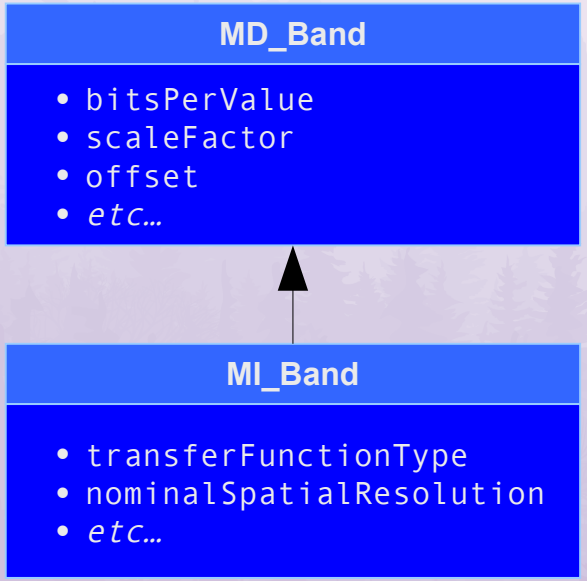
Consistently used pattern:

```
<gmd:property>
  <gmd:PACKAGE_Type>
```

Java bindings compiled from schema duplicate every classes

```
metadata.getIdentificationInfo().getMD_DataIdentification()
        .getCitation().getCI_Citation()
        .getCitedResponsibleParty().getCI_ResponsibleParty()
        .getContactInfo().getCI_Contact()
        .getOnlineResource().getCI_OnlineResource()
        .getLinkage().getURL();
```

Metadata historic



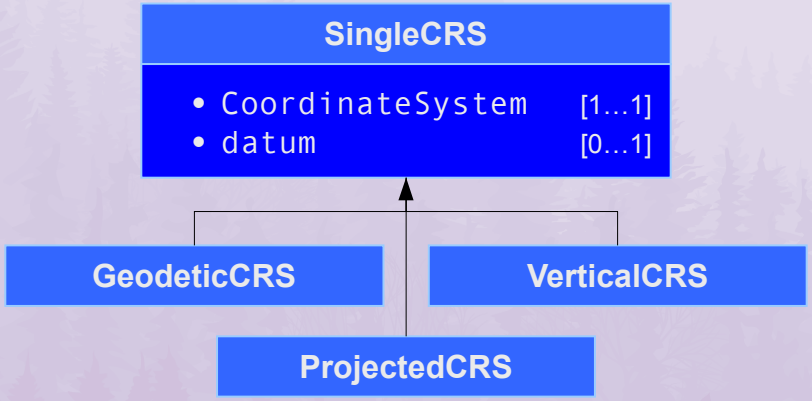
Imagery extension (ISO 19115-2) has been completed separately.

Merged in a single `Band` interface for simplicity

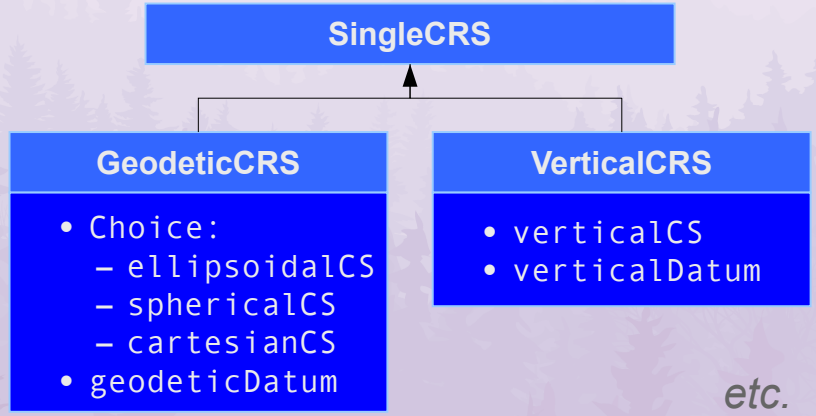
Apache SIS separates properties at XML marshalling time

Referencing

From UML



From XML schema



etc.

Single CRS associated to exactly one Datum:

- *From XML schema:* `getGeodeticDatum()`, `getVerticalDatum()`, *etc.*
- *From UML:* `getDatum()` with covariant return type.

API comparison

Given: `CoordinateReferenceSystem crs = ...;`
Wanting: `CoordinateSystem cs;`

From UML

```
cs = crs.getCoordinateSystem();
```

From XML schema

```
if (crs instanceof GeodeticCRS) {
    GeodeticCRS geodeticCRS = (GeodeticCRS) crs;
    cs = geodeticCRS.getEllipsoidalCS();
    if (cs == null) {
        cs = geodeticCRS.getSphericalCS();
        if (cs == null) {
            cs = geodeticCRS.getCartesianCS();
        }
    }
} else if (crs instanceof VerticalCRS) {
    VerticalCRS verticalCRS = (VerticalCRS) crs;
    cs = verticalCRS.getVerticalCS();
} else // etc.
```

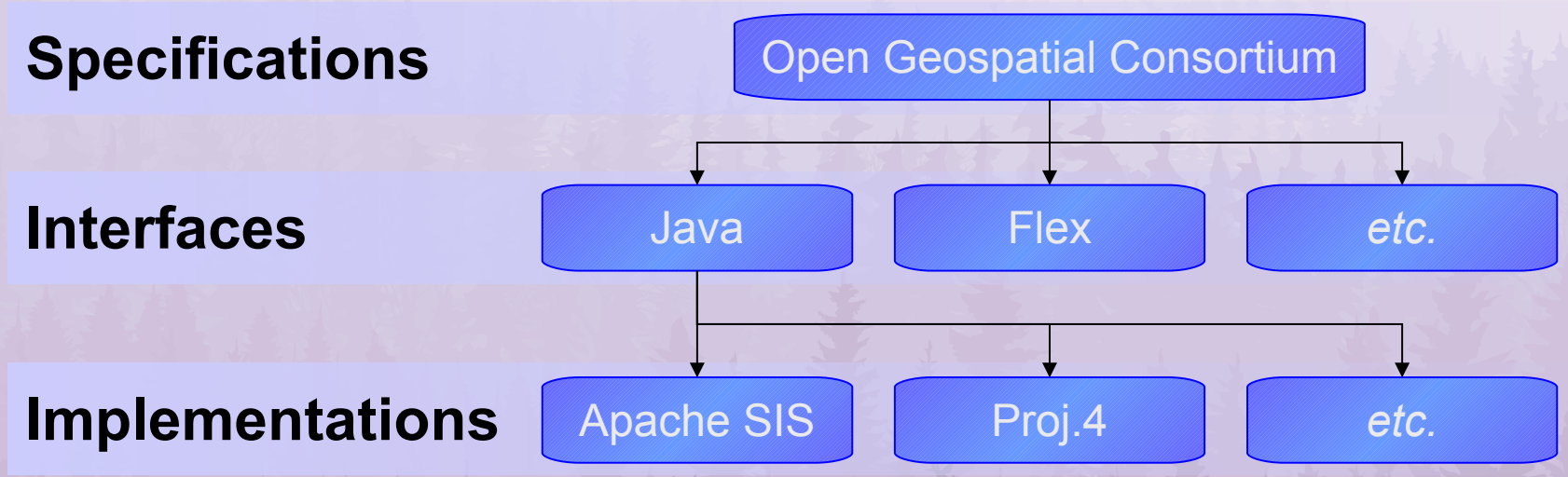
Why no automated process

- Avoid doubling depth of metadata tree
- One overridden method where appropriate
  - No distinct methods for the same attribute
- Union (*à la* C/C++) replaced on a case-by-case basis
  - By class hierarchy adjustment when it makes sense
  - Sometime no workaround
- Methods can compute their value
  - All single CRS have a datum, even if indirectly in projected CRS case
- Hide duplications caused by historical reasons

Goal: make it easier for API users
Not necessarily easier for implementors

GeoAPI interfaces

- Between OGC/ISO specifications and implementations
- Java language for now, but other languages are possibles
- Different implementations for the same set of interfaces

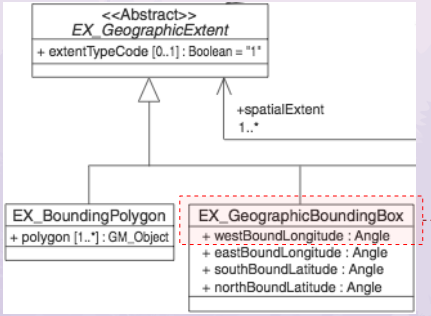


Interface example


```java
/**
 * Abstract coordinate reference system, consisting
 * of a single coordinate system and a single datum.
 *
 * @since GeoAPI 2.0
 */
@Classifier(ABSTRACT)
@UML(identifier="SC_SingleCRS", specification=ISO_19111)
public interface SingleCRS extends CoordinateReferenceSystem {
    /**
     * Returns the coordinate system associated to this CRS.
     */
    @UML(identifier="coordinateSystem", specification=ISO_19111, obligation=MANDATORY)
    CoordinateSystem getCoordinateSystem();

    /**
     * Returns the datum associated directly or indirectly to this CRS.
     */
    @UML(identifier="datum", specification=ISO_19111, obligation=OPTIONAL)
    Datum getDatum();
}
```

Interface example

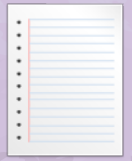


UML to API

	Name / Role Name	Short Name	Definition
344.	westBoundLongitude	westBL	western-most coordinate of the limit of the dataset extent, expressed in longitude in decimal degrees (positive east)

```
@UML(identifier="EX_GeographicBoundingBox", specification=ISO_19115)
public interface GeographicBoundingBox extends GeographicExtent {
    /**
     * Returns the western-most coordinate of the limit of the
     * dataset extent. The value is expressed in longitude in
     * decimal degrees (positive east).
     */
    @UML(identifier="westBoundLongitude", obligation=MANDATORY)
    double getWestBoundLongitude();
    // etc...
}
```

Spec. to javadoc



Resources

Test case example

testCassiniSoldner

```
public void testCassiniSoldner()
                    throws FactoryException,
                           TransformException
```

Tests the "*Cassini-Soldner*" (EPSG:9806) projection method. First, this method transforms the point given in the *Example* section of the EPSG guidance note and compares the MathTransform result with the expected result. Next, this method transforms a random set of points in the projection area of validity and ensures that the inverse transform and the derivatives are coherent.

The math transform parameters and the sample coordinates are:

Parameter	Value
semi-major axis	6378350.8704 m
semi-minor axis	6356675.0184 m
Latitude of natural origin	10.441666666666666°
Longitude of natural origin	-61.333333333333333°
False easting	86501.46392052001 m
False northing	65379.0134283 m

Source ordinates	Expected results
61°20'00"W 10°26'30"N	430000.00 links 325000.00 links
60°00'00"W 10°00'00"N	66644.94 links 82536.22 links

1 link = 0.66 feet
1 feet = 0.3048 metre

Throws:

FactoryException - If the math transform can not be created.
TransformException - If the example point can not be transformed.

See Also:

AuthorityFactoryTest.testEPSG_2314()

---

testLambertConicConformal1SP

```
public void testLambertConicConformal1SP()
                           throws FactoryException,
                                  TransformException
```

Same tests executed on various implementations

More than anti-regression tests

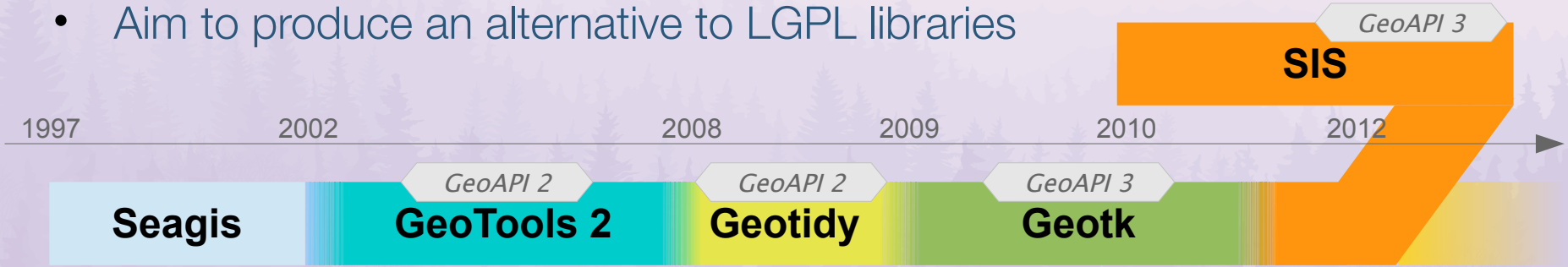
APACHECON
DENVER
WESTIN DENVER DOWNTOWN
APRIL 7-9, 2014

Apache SIS

Presented For The Apache Foundation By
LINUX FOUNDATION

History

- Apache SIS created in 2010
- Aim to produce an alternative to LGPL libraries



- Geotk has its root in 1997
- Code migration began in september 2012
- 94000 lines of code in April 2014
- About 1 million lines of code remaining

OGC/ISO standards

ISO	OGC	Topic	GeoAPI	Apache SIS
19103		Conceptual schema language	3.0.0	0.3
19115		Metadata (including imagery and gridded data extension)	3.0.0	0.3
19139		Metadata — XML schema		0.3
19111	08-015	Spatial referencing by coordinates	3.0.0	0.4 (partially)
19162	12-063	Well Known Text (WKT) representation of reference systems		0.4 (partially)
19136	07-036	Geographic Markup Language (GML)		0.4 (partially)
19107		Geometry (1 to 3 dimensional)	pending	
19123	07-011	Coverage geometry and functions	pending	Pending port from the Geotk project.
19156	10-004	Observation and measurement	pending	
13249		SQL spatial		
19128	06-042	Web Map Service (WMS)		Pending port from the Constellation project.
19142	09-025	Web Feature Service (WFS)		

And more...

SIS branches

Commits

Branch name	Target platform	GeoAPI version
JDK8	Java 8	GeoAPI 3.1-SNAPSHOT (including the pending part)
JDK7	Java 7	
JDK6	Java 6	
trunk	Java 6	GeoAPI 3.0.0 release

- SIS releases are created from trunk
- Trunk will follow GeoAPI latest releases

Strong points

- XML bindings hide some GML complexity
- *Well Known Text* (WKT) version 1 and 2
- Direct connection to an EPSG database
- “Late binding” referencing engine
- Transformations use derivatives for accuracy
- Extended precision arithmetic in critical parts
- Run *Geospatial Integrity of Geoscience Software* (GIGS) tests
- Extensive Javadoc

As of April 2014, not all those features are ported to SIS.
The port will be completed before the end of the year.
In the meantime, all the above exist in the Geotk project.

Roadmap

- SIS 0.3 — August 2013
  - *Metadata* (ISO 19115)
  - XML bindings for metadata (ISO 19139)
  - NetCDF-CF bridge to metadata
- SIS 0.4 — April 7th, 2014
  - First part of *Referencing by Coordinates* (ISO 19111) — basic classes
  - *Well Known Text* (WKT) version 1 and 2 formatting for basic classes
- SIS 0.5 — Summer 2014
  - Complete *Referencing by Coordinates* (ISO 19111) with map projections
  - Start of Features model and Shapefile reader
- SIS 0.6

Issues

Distributing EPSG database with SIS: Is it compatible with Apache 2 license?

- This is data, not software — we can not rewrite it.
- **Data of critical importance to Apache SIS.**
- EPSG conditions:
  - Can not sale EPSG tables alone (okay to sale them with SIS or other product).
    This condition would emerge only if user downloads SIS and deletes everything except EPSG.
  - If data are modified, can not be called “EPSG” anymore.
    Apache has the same policy for their own softwares.

Class name collisions with an other project that forked the `org.opengis` packages.

Apache SIS uses the official standard.

Links

- Apache SIS
  http://sis.apache.org
- GeoAPI
  http://www.geoapi.org
- Open Geospatial Consortium
  http://www.opengeospatial.org
- Geospatial integrity of geoscience software (GIGS)
  http://www.epsg.org/gigs.html

Thanks!

Merci!

ありがとうございます