



uDig

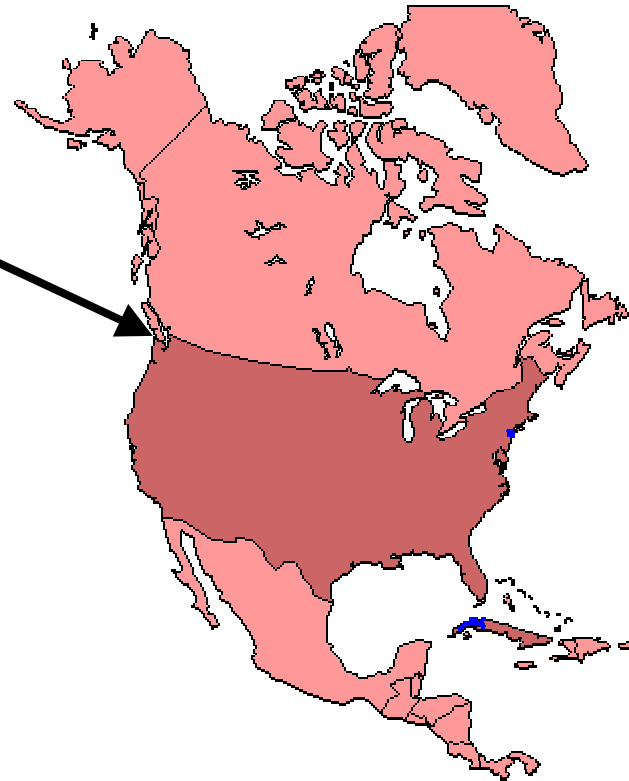
User-friendly Desktop Internet GIS



Paul Ramsey
Jody Garnett
Jesse Eichar

Refractions Research

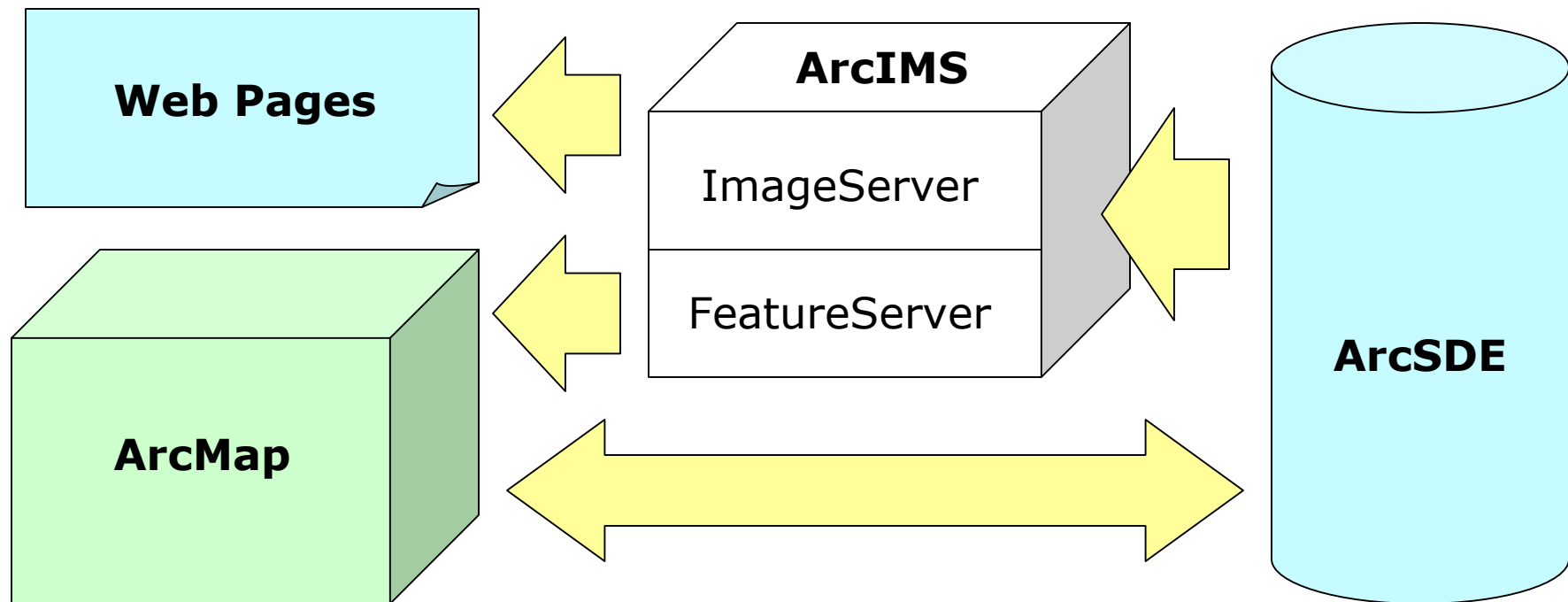
- Victoria, BC, Canada
- Spatial Systems Consulting
- Open Source Software
- PostGIS
uDig / GeoTools



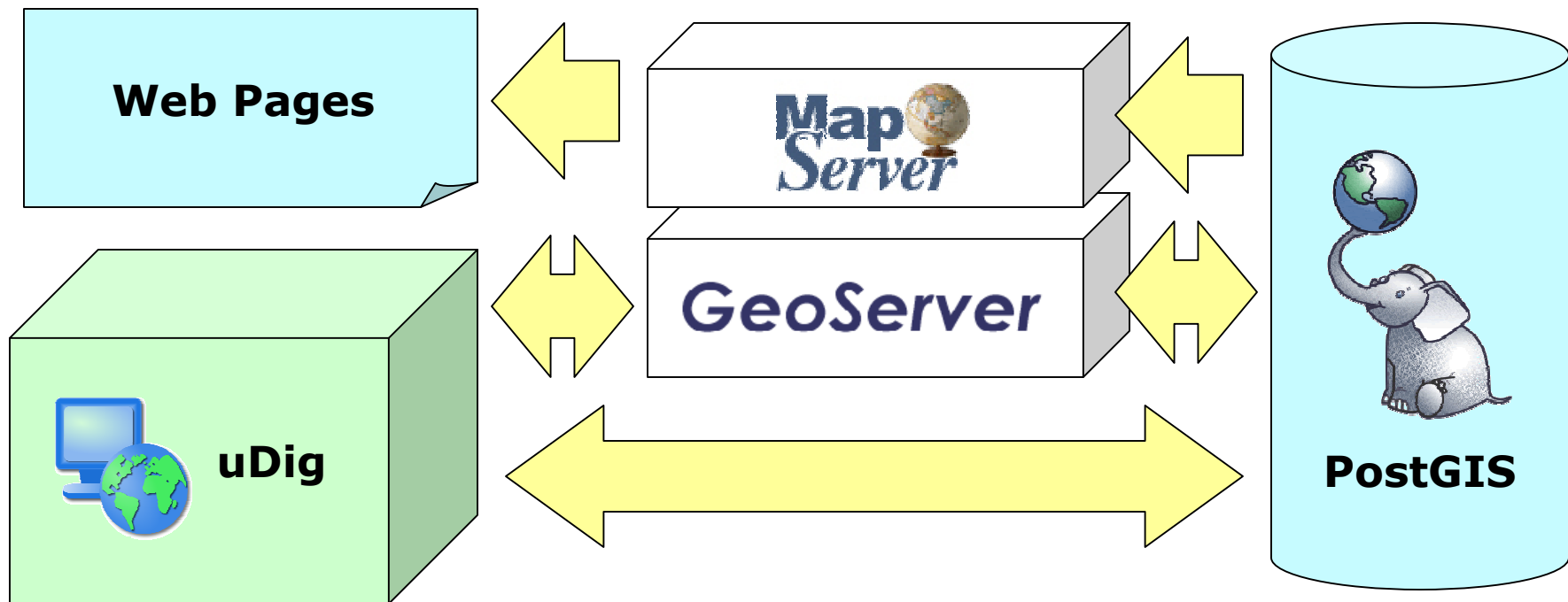
Geospatial Architecture

- Spatial Database
 - Concurrency
 - Transactions
 - Seamlessness
- Internet Publishing
 - Feature Access
 - Map Access
- Data Manipulation
 - Direct Access
 - Editing
 - Cartography

ESRI Architecture



Open Source Architecture



Missing Link for OpenGIS

- “Integrated Client”
- Ability to directly view WMS
- Ability to directly edit WFS
- Ability to search catalogues
- Ability to integrate standard GIS data
- Hides complexity of network access

Missing Link for Open Source

- “Standard GIS Functionality”
- Ability to directly edit GIS data
- Ability to connect to PostGIS, Mapserver, GeoServer
- Ability to create paper cartography
- Ability to integrate with proprietary infrastructures

Gimme a "u"!

- "User Friendly"
 - Sensible Defaults
 - Use Preferences for Complexity
 - Automatic Integration
 - Coordinate Reference Systems
 - Formats, Services
 - Drag and Drop Everywhere
 - Hide Differences

Gimme a "D"!

- "Desktop"
 - Not a Web Application
 - Desktop Look and Feel
 - Windows, Linux, Apple OS/X
 - Desktop Integration
 - Cut and Paste, External Drag and Drop
 - Desktop Installation
 - One Click Installers

Gimme an "i"!

- "Internet"
 - Consume Remote Data and Services
 - View OGC Web Map Server
 - View/Edit OGC Web Feature Server
 - View ESRI ArcIMS Server *
 - Multi-catalogue Search
 - Treat Local and Internet Layers Equally

* Coming Soon

Gimme a "g"!

- "GIS"
 - Platform Extensibility
 - Platform "De-stensibility"
 - Hooks for Analytical Plugins
 - Data Editing
 - Standard GIS Data Sources
 - Paper Cartography

Whaddaya Got ?!?

wDdig

DEMO...

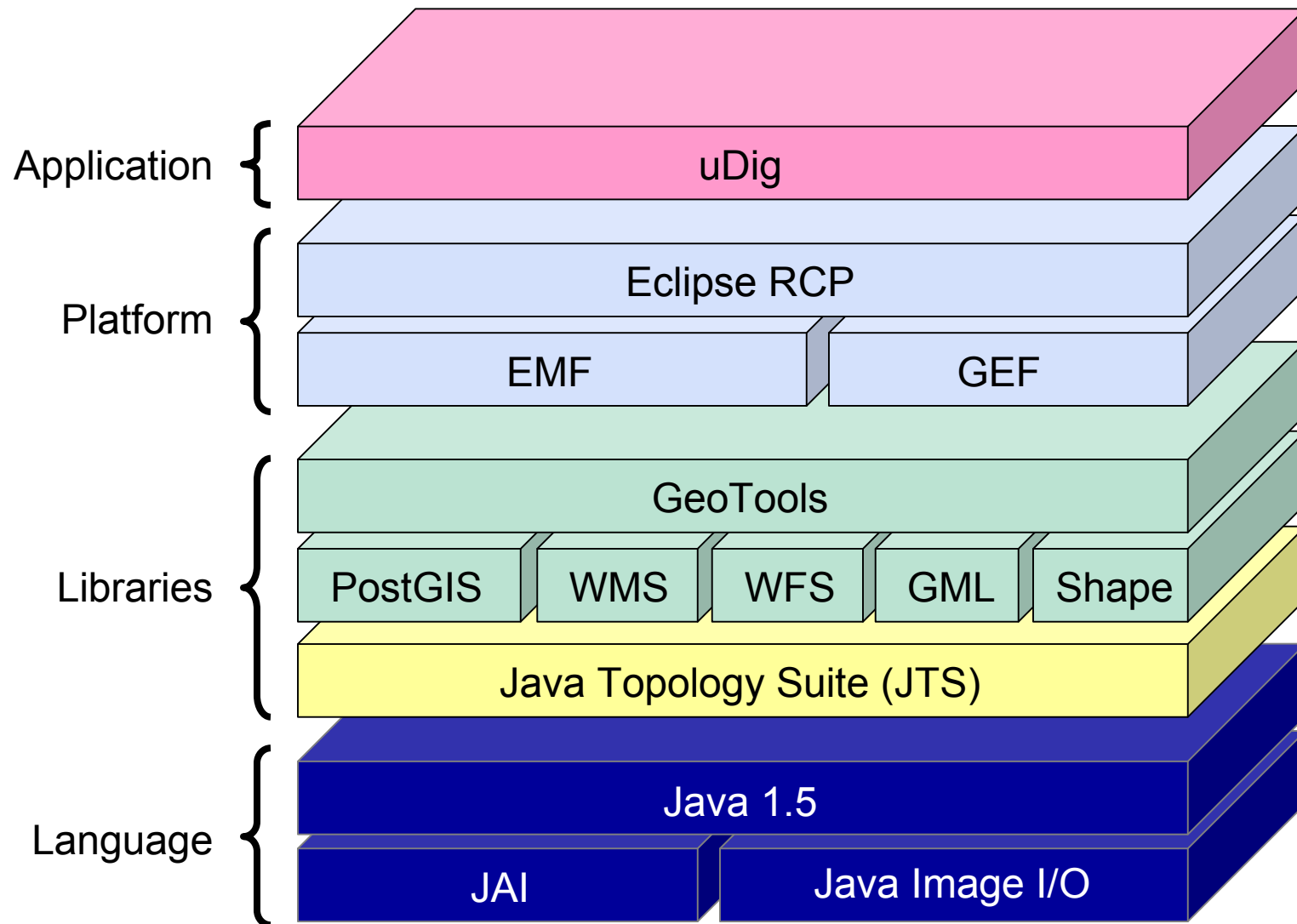
MUM/EOGEO 2005

Jody Garnett

- uDig Development Team Leader
- GeoTools Project Management Committee (PMC) Member
- GeoServer contributor
- Anachronistic pugilist
- Technology Platforms and Decisions We Made



uDig Architecture



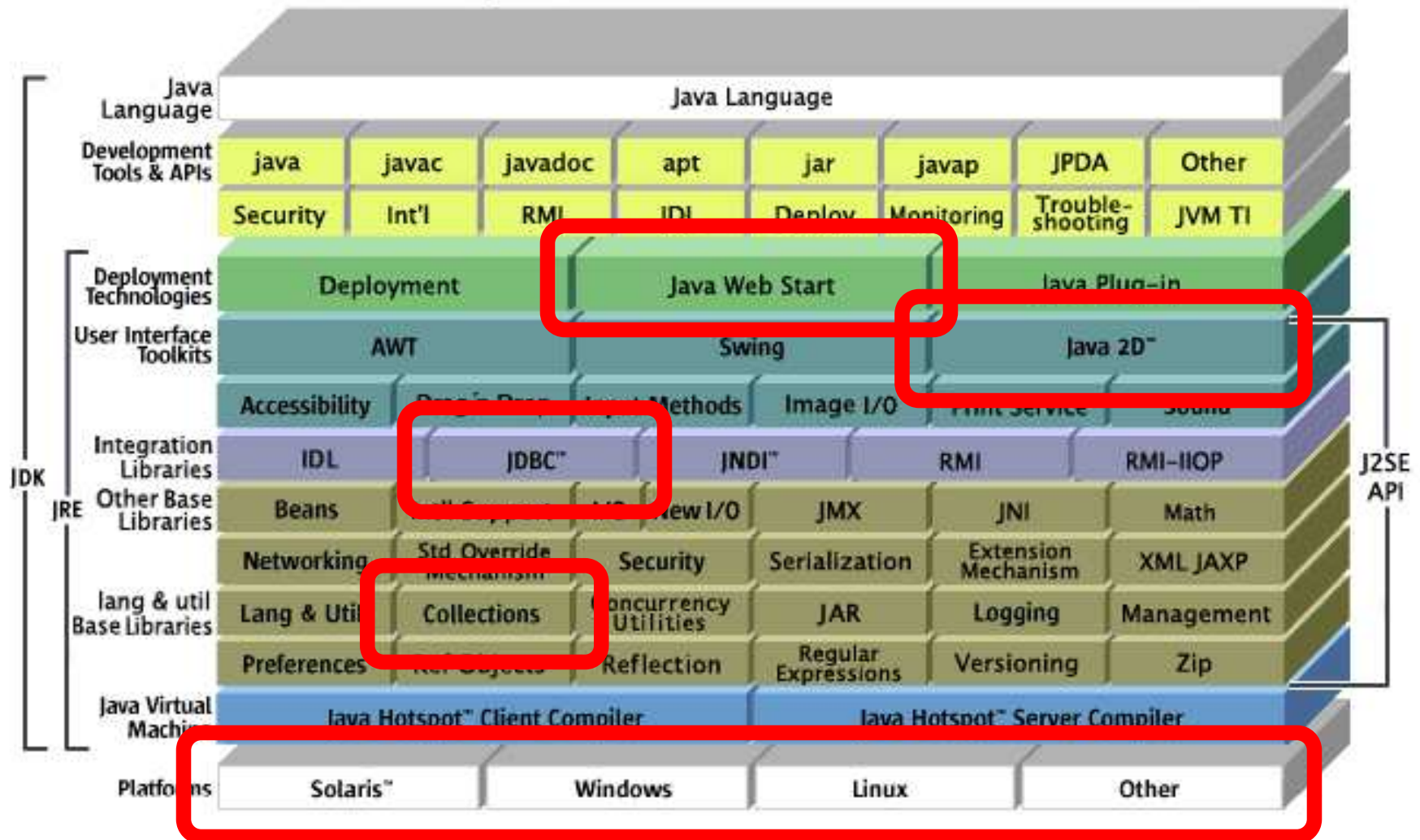
Java 1.5

- Risks
 - Compatibility with other Java libraries that are 1.4 only
 - Danger of the bleeding edge
- Rewards
 - New language features
 - type narrowing
 - enumerations
 - More explicit API for uDig
 - Better Linux support for JAI under SWT



Java 1.5

Java™ 2 Platform Standard Edition 5.0



GeoTools

- Risks
 - Needed to add a **GML** parser
 - Had to coordinate a scheduled project with an unscheduled community

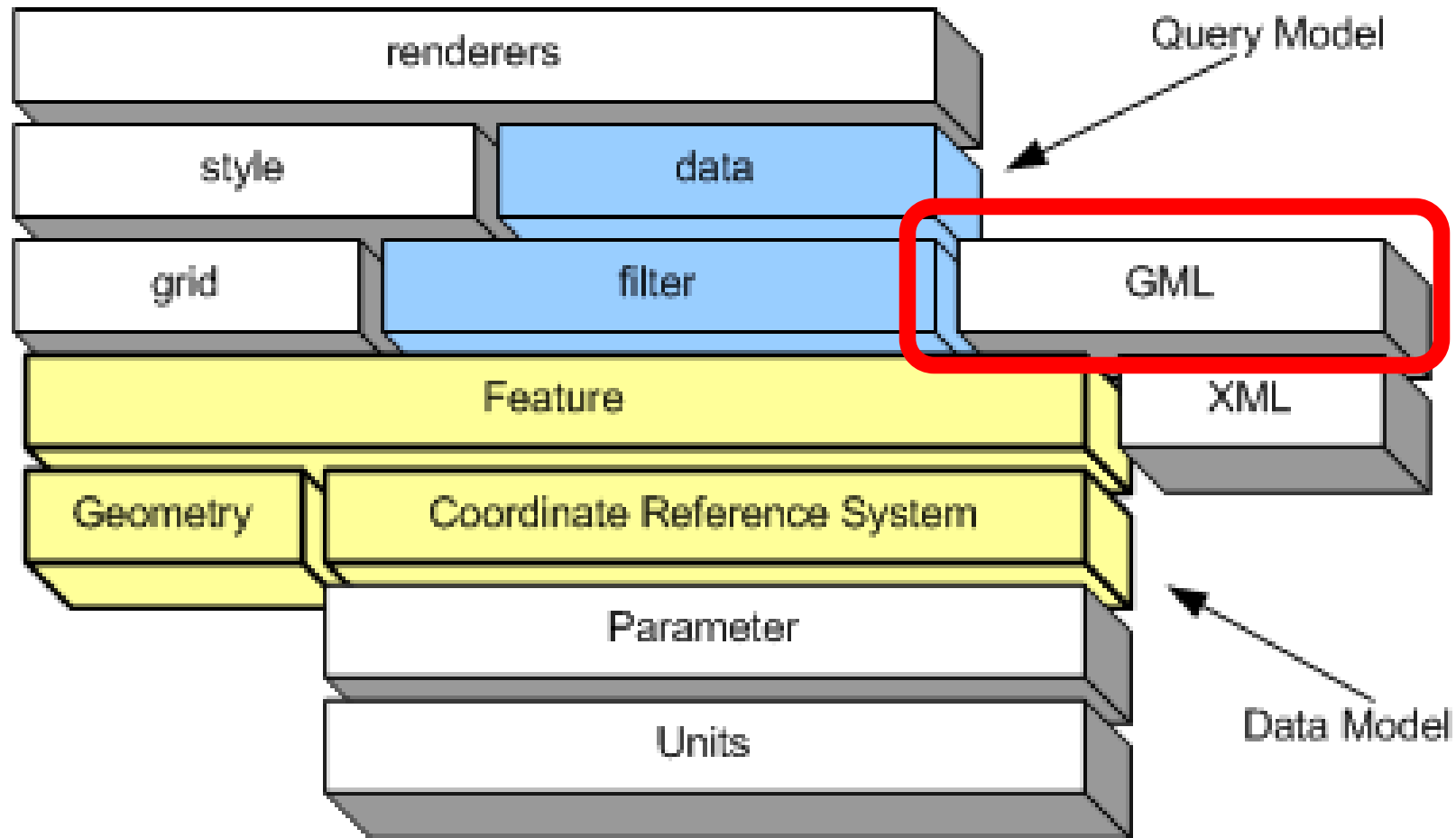
GeoTools

The open source Java GIS toolkit



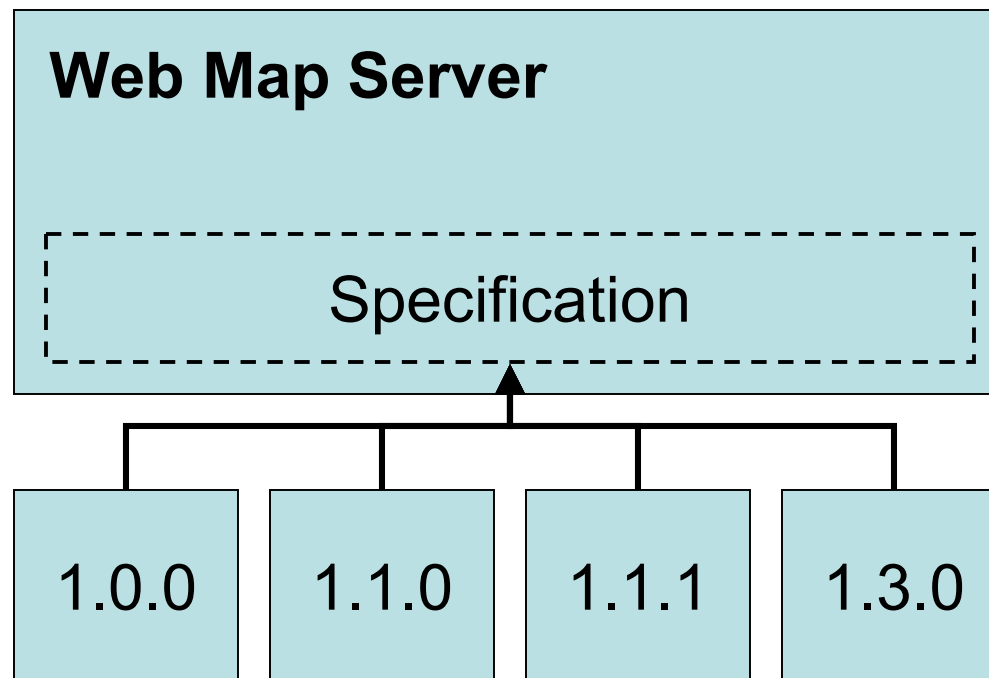
- Rewards
 - Larger base of developers
 - Rich API
 - Standards (OGC, ISO) compliant by design

GeoTools



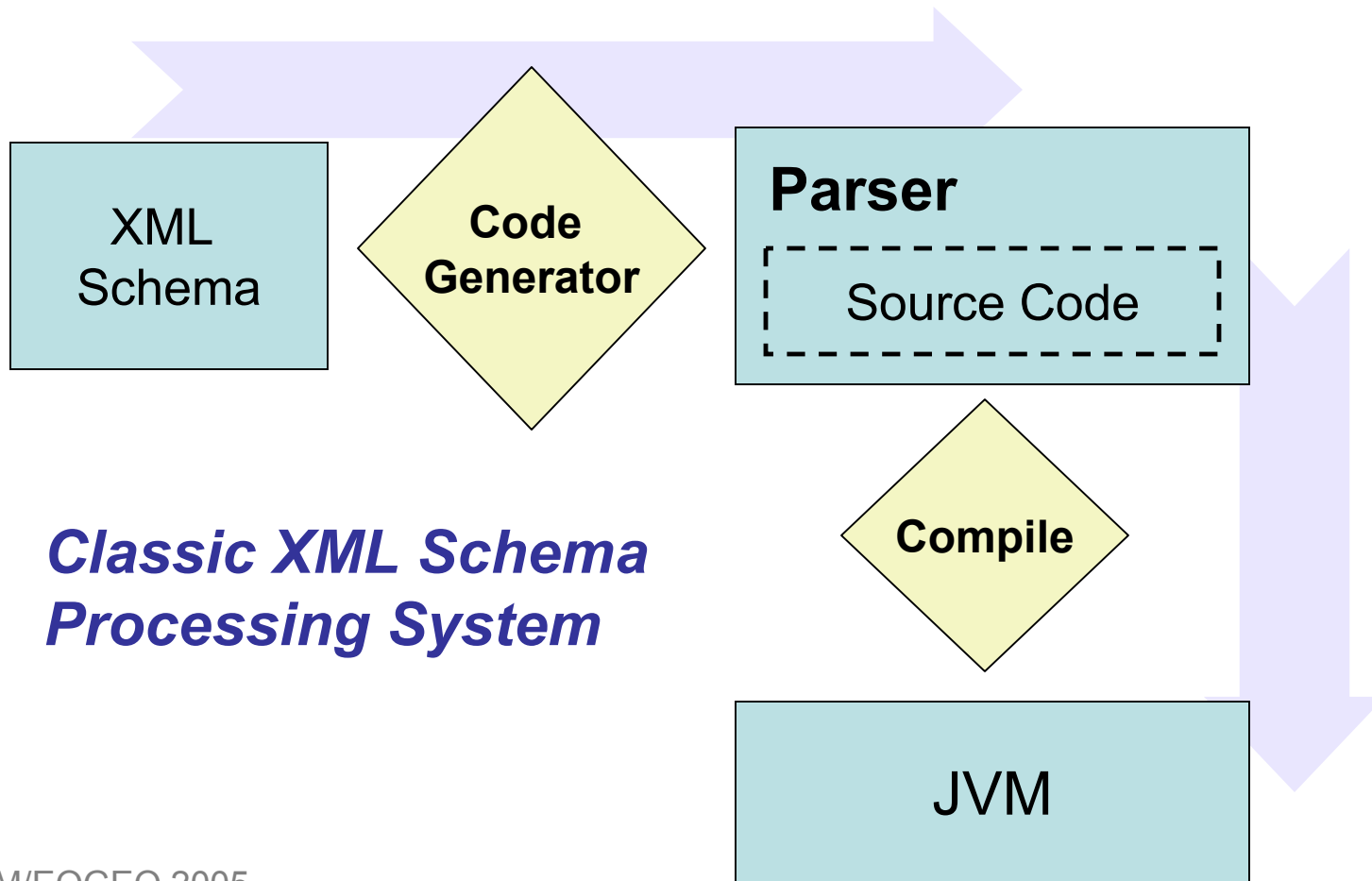
GeoTools – WMS Client

- We wrote generic WMS client
 - WMS has 4 versions!



GeoTools – WFS Client

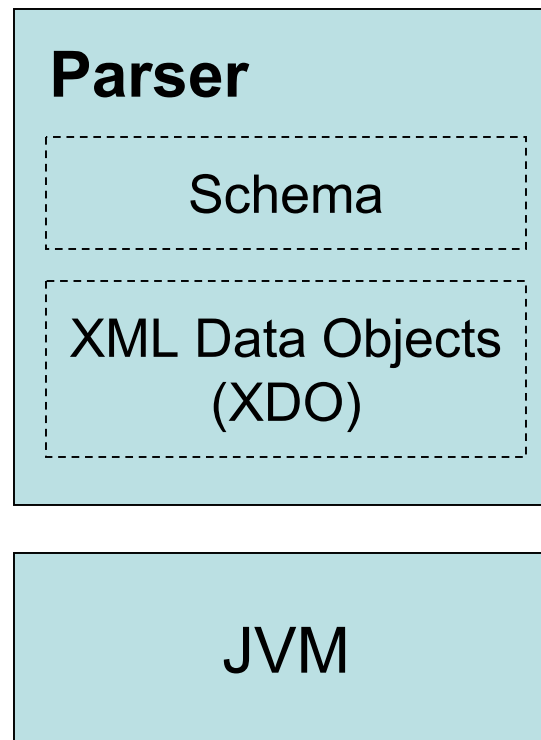
- We wrote generic WFS client
 - GML handling is hard!



GeoTools – WFS Client

- We wrote generic WFS client
 - GML handling is hard!

*Our XML Schema
Processing System*



Eclipse RCP

- Risks

- Bleeding edge when we started
- Not Swing
- SWT integration with core Java



- Rewards

- Plug-in environment pre-defined
- Documented platform
- Attractive and extensible by design

Eclipse RCP

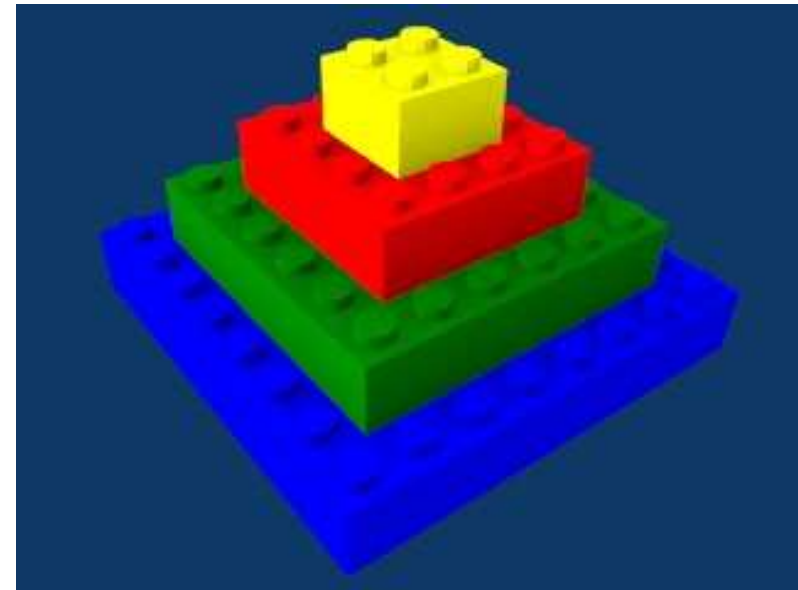
Eclipse RCP is more than an application framework system, it is an **platform** that not only allows extension via plug-ins, but also organizes the plug-ins into a **rational structure**.

Framework

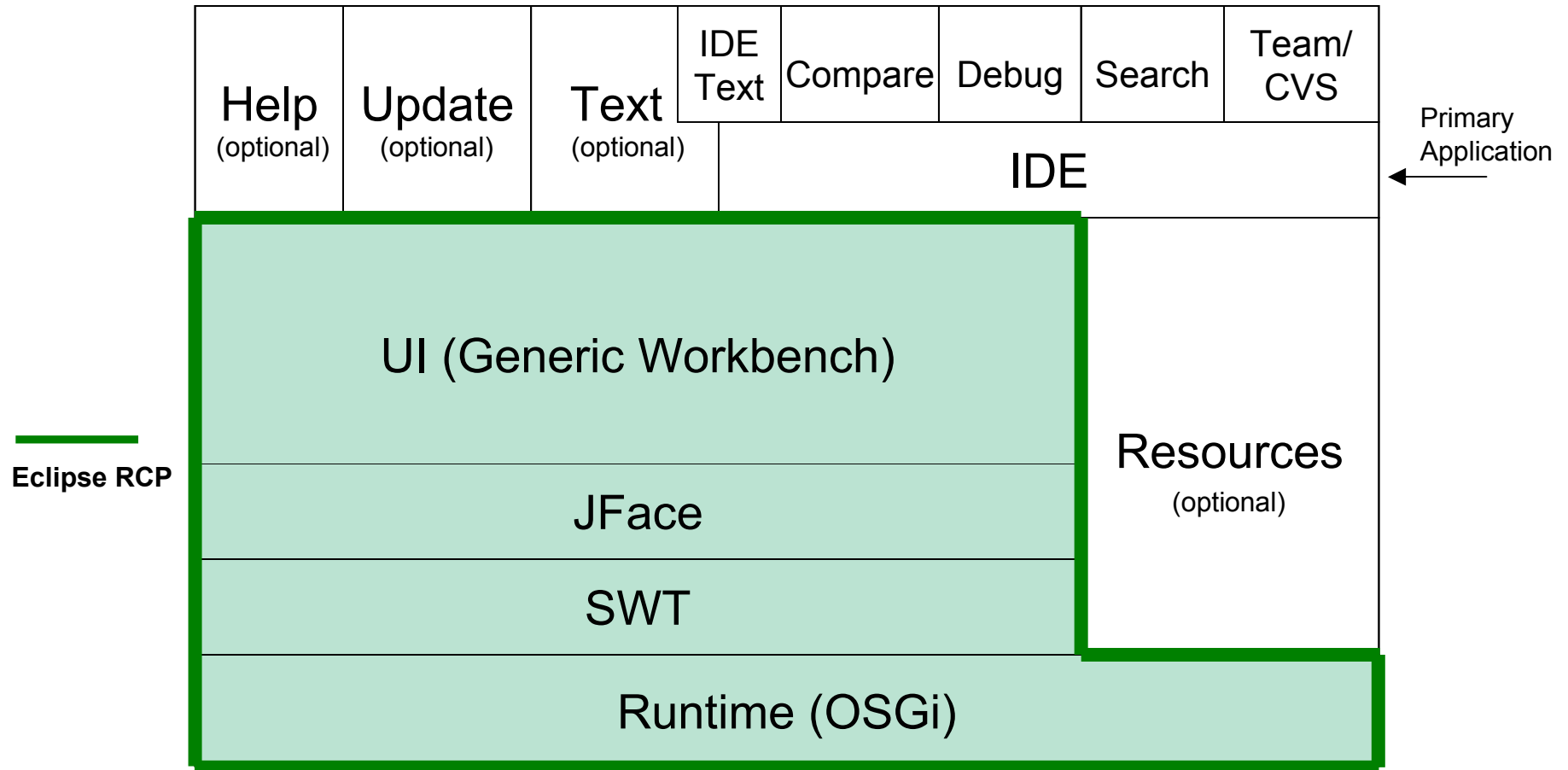


vs

Platform



Eclipse RCP

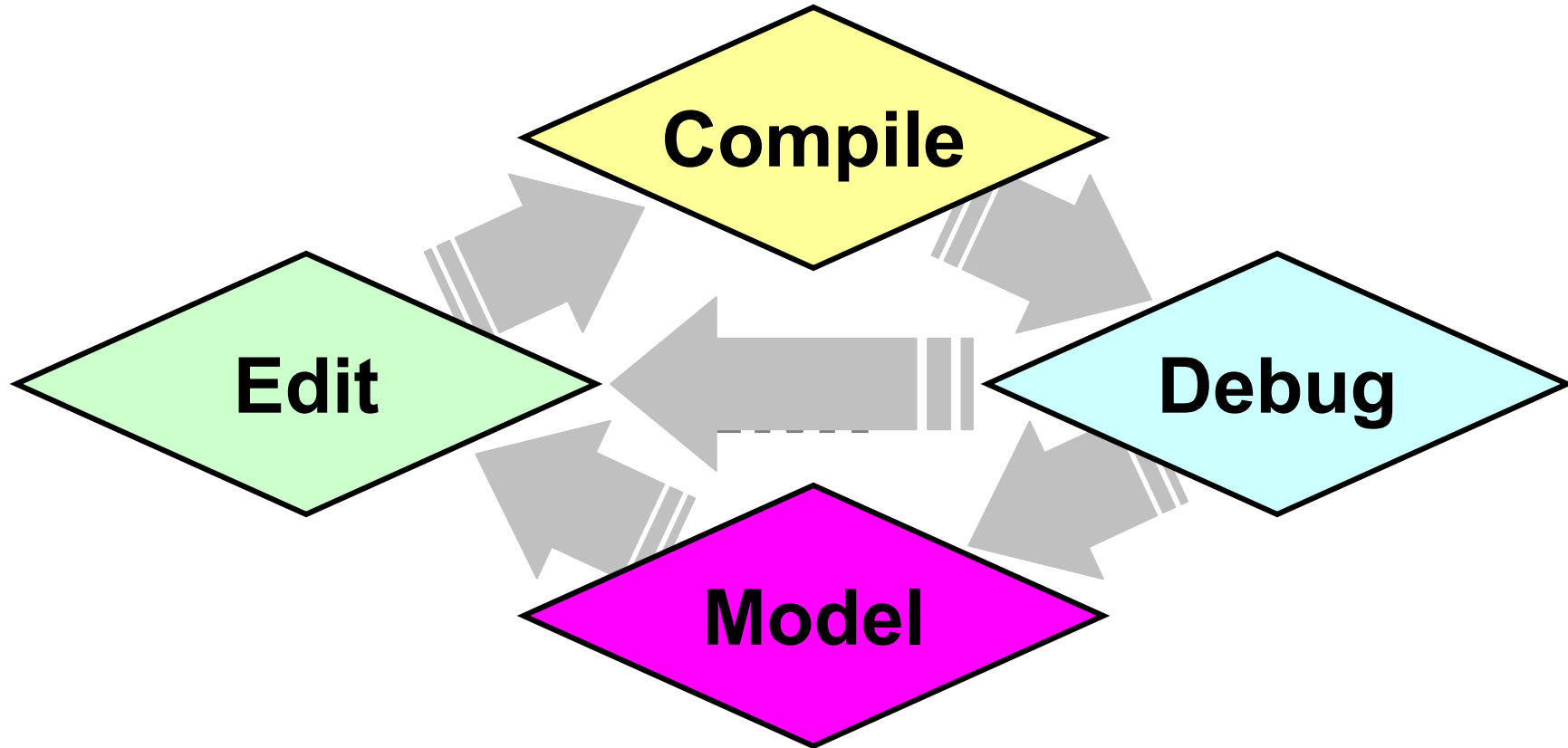


Eclipse EMF

- Risks
 - Eclipse Modelling Framework (EMF)
 - New technology
 - More complex API
- Rewards
 - Integrates modelling into programming environment
 - Allows rapid changes in application model
 - Weeks versus Hours
 - Drag-n-drop, events, persistence all built-in



Eclipse EMF

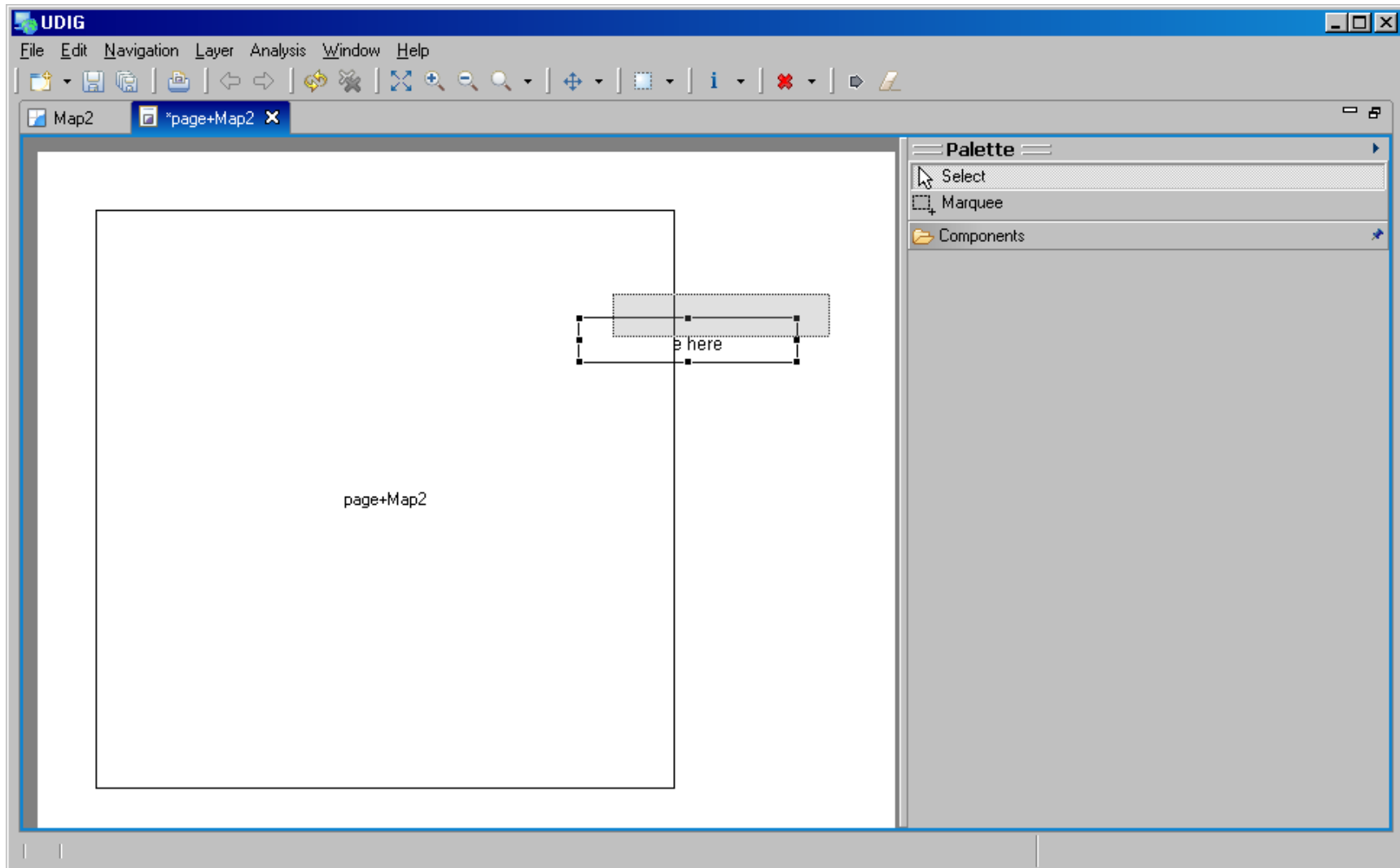


Eclipse GEF

- Risks
 - Graphical Editing Framework (EMF)
 - New technology
- Rewards
 - Pre-built graphical manipulation toolkit
 - Future use in GIS process model builder



Eclipse GEF



Development Environment

Component	Product
IDE	Eclipse
Build System	Eclipse / PDEBuild
Version Control	Subversion
Real Time Build	Cruise Control
Wiki / Documentation	Confluence
Bug Tracking	JIRA

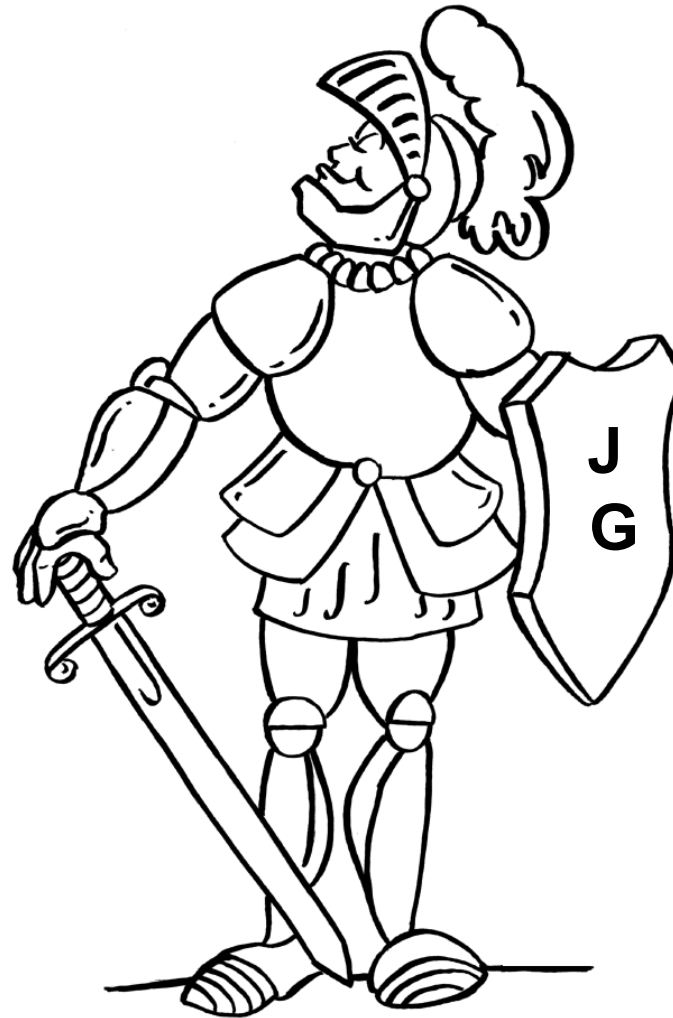
Mistakes

Ooops!	Aaaahh!
Catalog 2.0	CGDI / Google XML
Drupal	Confluence
Scope	Release early, release less
Java 2D Render	OpenGL Draw2D

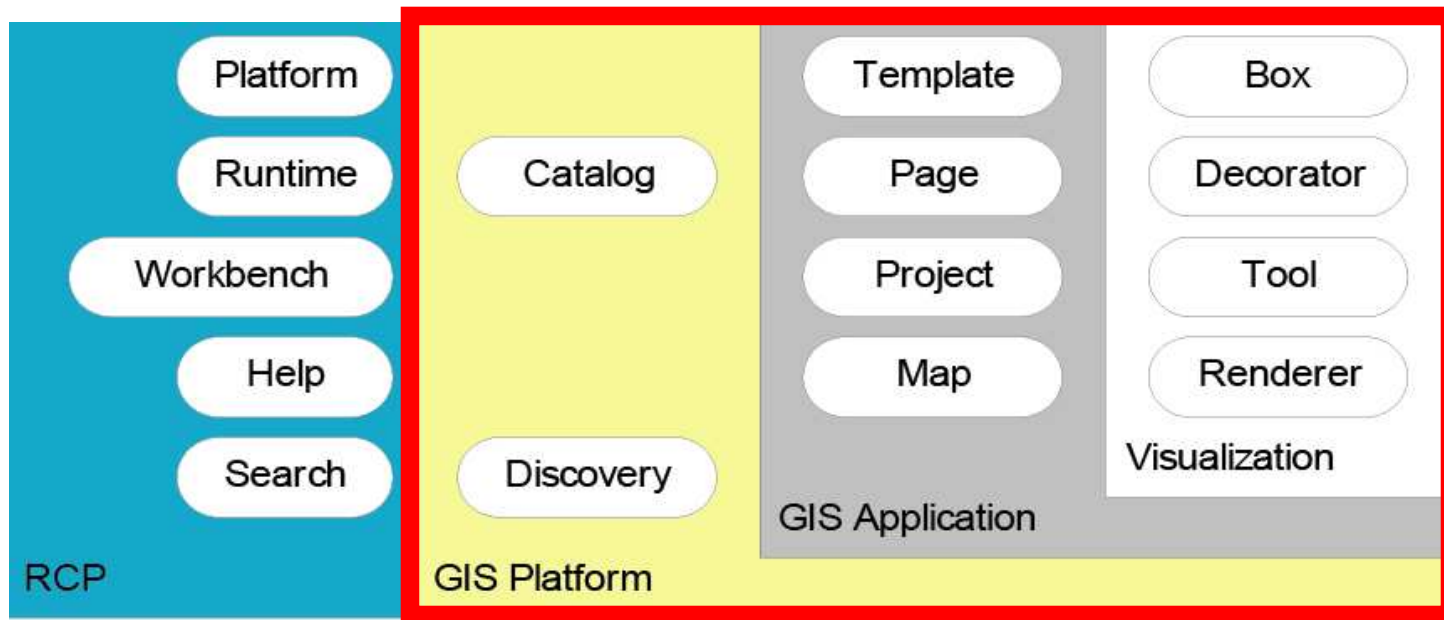
How to Contribute

- Everyone
 - Join the udig-devel list
- Developers
 - Quickstart
 - Source Code
- Users (Future Developers)
 - Download
 - Test / Bug Report
 - Translate
 - Supply Crazy Ideas (Toolkits, 3D, GML3!)

Questions?



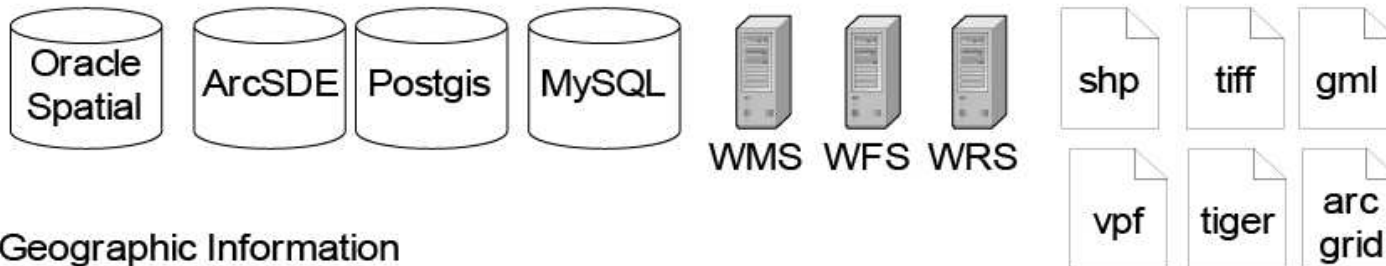
Architecture of uDig



GIS Toolkits

Geotools

GeoAPI



Geographic Information

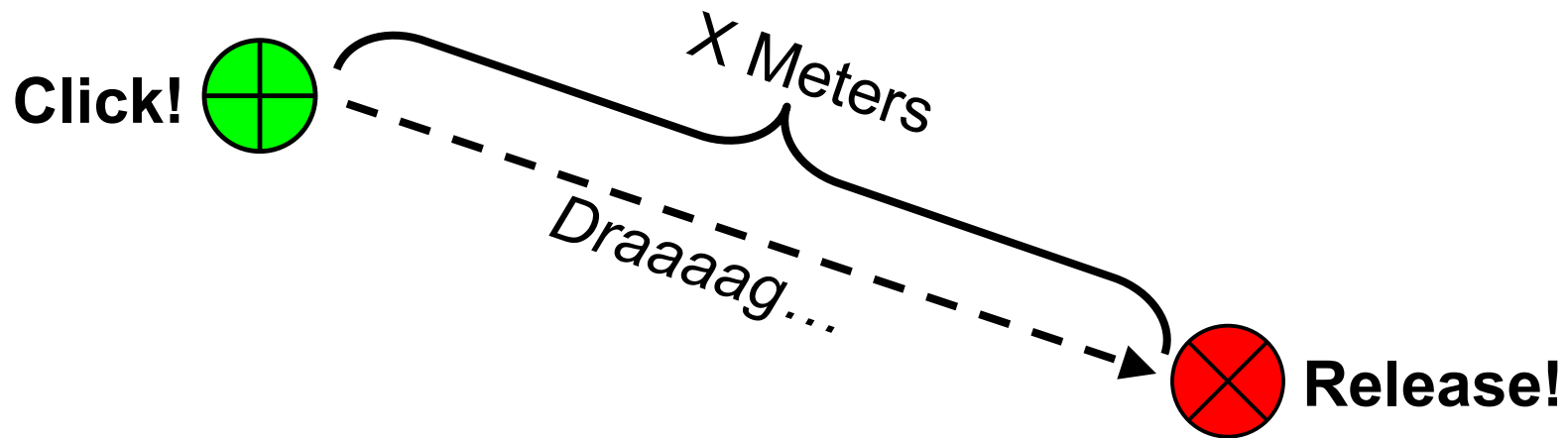
Jesse Eichar

- Developed uDig Core Model
- Developed uDig Renderer
- GeoTools Committer
- Solver of Hard Problems
- Brazilian pugilist
- Example of Extending uDig



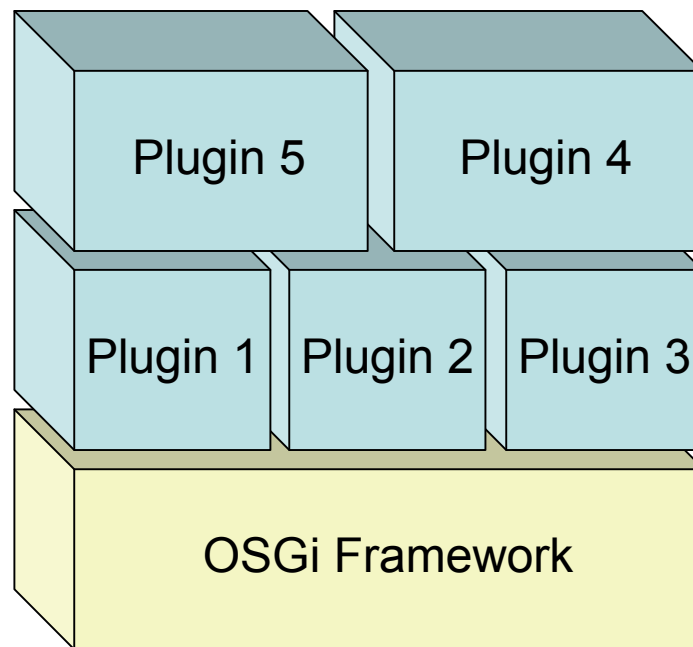
Goal

- A tool that returns the ground distance between the point the mouse is clicked and the point the mouse is released.



Plug-ins

- Basic building block of Eclipse and uDig
- Have dependencies on and provide facilities for other plug-ins



Extension Points

- Every plug-in uses extension points
- Some plug-ins **provide** extension points
- Extension points provide organization in a potentially chaotic situation:
 - Applications built entirely of different plug-in components
 - Each plug-in must know its role, and advertise it, so that others know too
 - Each plug-in must know what other plug-ins it depends on
 - Stay in your sandbox!

Extension Points

The screenshot shows the Eclipse IDE interface. The title bar reads "Plug-in Development - net.refractions.udig.distanceTool - Eclipse Platform". The menu bar includes "File", "Edit", "Navigate", "Project", "Run", "Window", and "Help". The toolbar contains various icons for file operations and development actions. The editor area shows several open files: "MapEditor.java", "CursorPosition.java", "net.refractions.u...", and "messages.properties".

The "Extensions" view is active, displaying a tree of "All Extensions" on the left and "Extension Element Details" on the right. The tree shows the following structure:

- net.refractions.udig.project.ui.tool
 - Measurements (category)
 - distance (modalTool)
- org.eclipse.ui.commands
 - Measurement Tools (command)
- org.eclipse.ui.bindings

Buttons for "Add...", "Edit...", "Up", and "Down" are visible next to the tree. The "Extension Element Details" panel shows the following configuration for the "modalTool" extension:

id*:	net.refractions.udig.distanceTool.distanceTool
tooltip:	Measure the surface distance between two points
class:	net.refractions.udig.distanceTool.Dist <input type="button" value="Browse..."/>
icon:	<input type="text"/> <input type="button" value="Browse..."/>
name:	distance
inMenu:	true
onToolBar:	true
categoryId:	net.refractions.udig.measurements
commandIds:	<input type="text"/>
commandHandler:	<input type="text"/> <input type="button" value="Browse..."/>

At the bottom, the "Body Text" view is collapsed. The bottom status bar shows tabs for "Overview", "Dependencies", "Runtime", "Extensions", "Extension Points", "Build", "MANIFEST.MF", "plugin.xml", and "build.properties".

uDig Extension Points

- Resources
- Renderers / Styles
- Operations
- MapGraphics
- Tools
- Drag'n'Drop
- FeatureEditor

Distance Tool

- Will use the uDig **Tool** extension point
- **Tool** extension point provides:
 - A Context (simple API to Map)
 - A Panel (the Map Editor UI)
- **Tools** must extend an AbstractTool:
 - Modal Tool (“info”)
 - Action Tool (“zoom in”)
 - Background Tool (“current coords”)

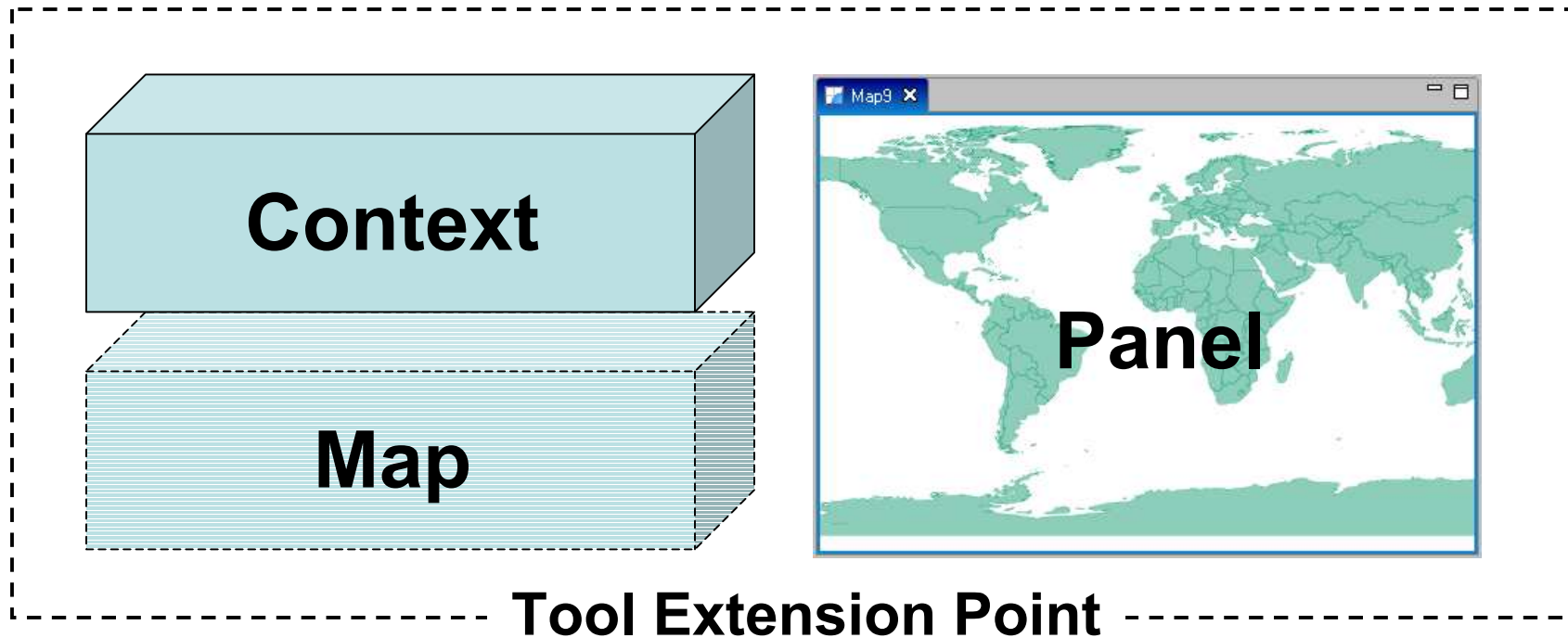


MUM/EOGEO 2005



Distance Tool

SimpleDistanceTool



SimpleDistanceTool

- Small tool, so just one class with three methods:
- mouseDown
 - Mark the button click
- mouseUp
 - Mark the button release and calculate the distance
- displayOnStatusBar
 - Get the calculated distance into the workbench status bar

DistanceTool

```
package net.refractions.udig.distanceTool;

import org.eclipse.jface.action.IStatusLineManager;
import org.geotools.referencing.CRS;
import com.vividsolutions.jts.geom.Coordinate;
import net.refractions.udig.project.ui.render.displayAdapter.MapMouseEvent;
import net.refractions.udig.project.ui.tool.AbstractModalTool;
import net.refractions.udig.project.ui.tool.ModalTool;

public class DistanceTool extends AbstractModalTool implements ModalTool {
    public DistanceTool() {
        // Register for mouse events
        // Options are: MOUSE, MOTION, WHEEL
        super(MOUSE);
    }
}
```

Coordinate start;

...

mousePressed

```
public void mousePressed(MapMouseEvent e)
{
    start=getContext().pixelToWorld(e.x, e.y);
}
```

mouseUp

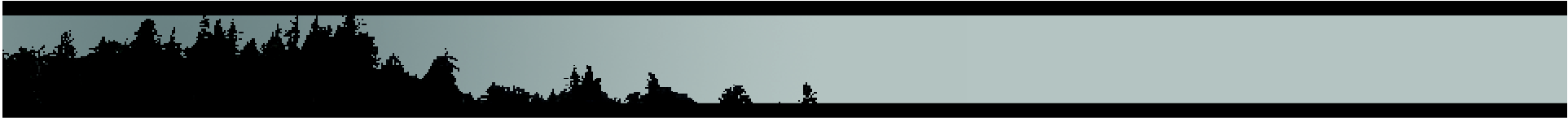
```
public void mouseReleased(MapMouseEvent e) {  
    Coordinate end=getContext().pixelToWorld(e.x, e.y);  
    try {  
        double distance=CRS.distance(  
            start, end, getContext().getCRS());  
        displayOnStatusBar(distance);  
    }  
    catch (Exception e1) {  
        displayError();  
    }  
}
```

displayError

```
private void displayError() {  
    final IStatusLineManager statusBar = getContext().getStatusBar();  
    if( statusBar==null )  
        return; // shouldn't happen if the tool is being used.  
    getContext().updateUI(new Runnable() {  
        public void run() {  
            statusBar.setErrorMessage("Unable to calculate the distance  
        }  
    });  
}
```

displayOnStatusBar

```
private void displayOnStatusBar(double distance) {
    final IStatusLineManager statusBar =
        getContext().getStatusBar();
    if( statusBar==null ) return;
    int totalmeters=(int)distance;
    final int km=totalmeters/1000;
    final int meters=totalmeters-(km*1000);
    float cm = (float) (distance-totalmeters)*10000;
    cm = Math.round(cm);
    final float finalcm=cm/100;
    getContext().updateUI(new Runnable(){
        public void run() {
            statusBar.setMessage(
                "Distance = "+km+", "+meters+"m "+finalcm+"cm");
        }
    });
}
```

Demonstration

Questions?

