

# GUI Architecture

This page is for a general discussion of a new GUI architecture.  
(Under construction, feel free to add random thoughts)

## Overview

It should be remembered that GeoTools2 exists to allow the creation of applications. It is not intended to actually be one. Because of this, any 'architecture' should not place too many restrictions on how a developer might want to use it. A developer should be able to embed the GUI components we create inside almost any platform structure.

The main issue to be considered is - how to interact with a renderer? i.e. how to navigate, how to style, how to query and how to edit a map shown in a display.

## Requirements

The overall architecture should (probably) be platform neutral so as to allow SWT and Swing implementations

It needs to be easy to extend with new tools / variants of tools

It needs an event / notification system to automatically update the renderer should any of the following change:

- data
- filters
- styling
- selection / highlighting

It should allow for coordination between multiple views.

It should follow the MVC pattern - the entire state (visible extent, styling, selections...) should be external to any viewer component

It should support undo/redo as soon as possible (implies command/action pattern of some form)

Where possible selections / actions should be expressed using Filters - i.e. a select tool that drags out a box should actually generate an intersects or contains filter rather than working out the selected features for itself.

## Discussion

The convention when thinking about MVC is that there is one model for the view. However, I think we need to think about having multiple models...

e.g. The style, area of interest, data and current selections should be separate models

## Use Cases

## Other architectures

There are many other projects out there that face the same issues. The following links point to design documents and javadocs for other projects' GUI designs from which we might beg/borrow/adapt solutions.

- [geoapi](#) - Specifically the org.opengis.go.\* packages
- [OpenMap](#) - Their [API](#) docs
- [JUMP](#) - [API Docs](#)

- [GISToolkit](#)
- [Deegree](#), current CVS, no API docs