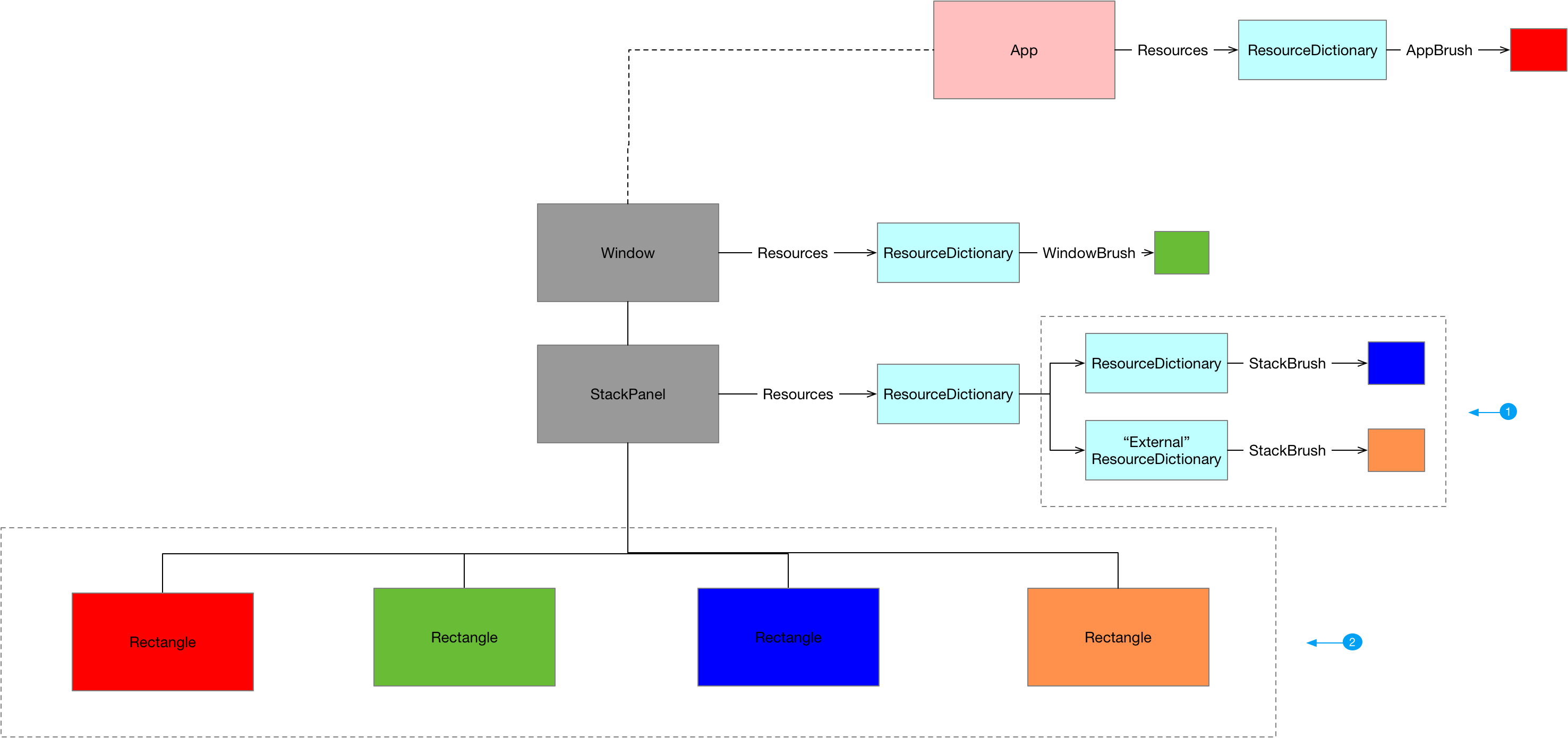
Resources

Sharing .NET objects

Resource Dictionaries provide a standard mechanism for packaging and accessing resources in WPF.

## Resources use the logical tree



1. Resource dictionaries can be merged
2. Framework Elements can access resources from the resources collections of parents in the logical tree (and resources defined in the application itself )

Resources can be accessed as either static or dynamic resources

Below we share the source code for this example

Listing ExternalResourceDict.xaml



Listing App.xaml



Listing MainWindow.xaml



## Resource resolution

We access objects stored in the resource mechanism from XAML using either Static or Dynamic markup extensions.

StaticResource markup extension

* Can only look for resources by string. Either x:Key or x:Name
* Processes key by searching for it in all parent resource dictionaries
* Does not support forward references from resource dictionaries
* Evaluated at load time

DynamicResource markup extension

* Processes key by creating an expression that is evaluated at runtime
* Evaluated at runtime
* Can only be used to set values on DependencyProperties or Freezables

### Dynamic Resource Resolution

A DynamicResource markup extention will perform the following steps to try and resolve a key.

1. Look in the current elements resource dictionary
2. If the element has a style set look in the style’s resource dictionary
3. If the element has a template set look in the template’s resource dictionary
4. Walk back up the logical tree looking in ancestors’ resource dictionaries

The following diagram show how this works for a piece of XAML we sh

Listing 4MainWindow.xaml



Listing 5MainWindow.xaml.cs



Questions

Which types have a Resources property?

FrameworkElement

FrameworkContentElement

Application.Resources

What kind of objects are typically stored in ResourceDictionaries?

Templates, Styles, Brushes

What kind of object can be stored in Resource Dictionaries?

Any .NET object. If we want to instantiated it from XAML it needs to be XAML friendly(Default public constructor etc.)

Why use ResourceDictionary

Provide standardised location to store and access shared objects

Compare assembly resources with WPF resources

Typically assembly resources are chunks of binary data such as images and sound files that we want to access. WPF resources are .NET objects which we want to store in a resource dictionary so we can access them from multiple places