Leveraging the Model-View-Presenter Pattern in Rich Client Applications

Patrick Paulin
Eclipse RCP Trainer and Consultant
RCP Quickstart

patrick@rcpquickstart.com

www.rcpquickstart.com/training/presentations/mvp-and-rcp-ew2008

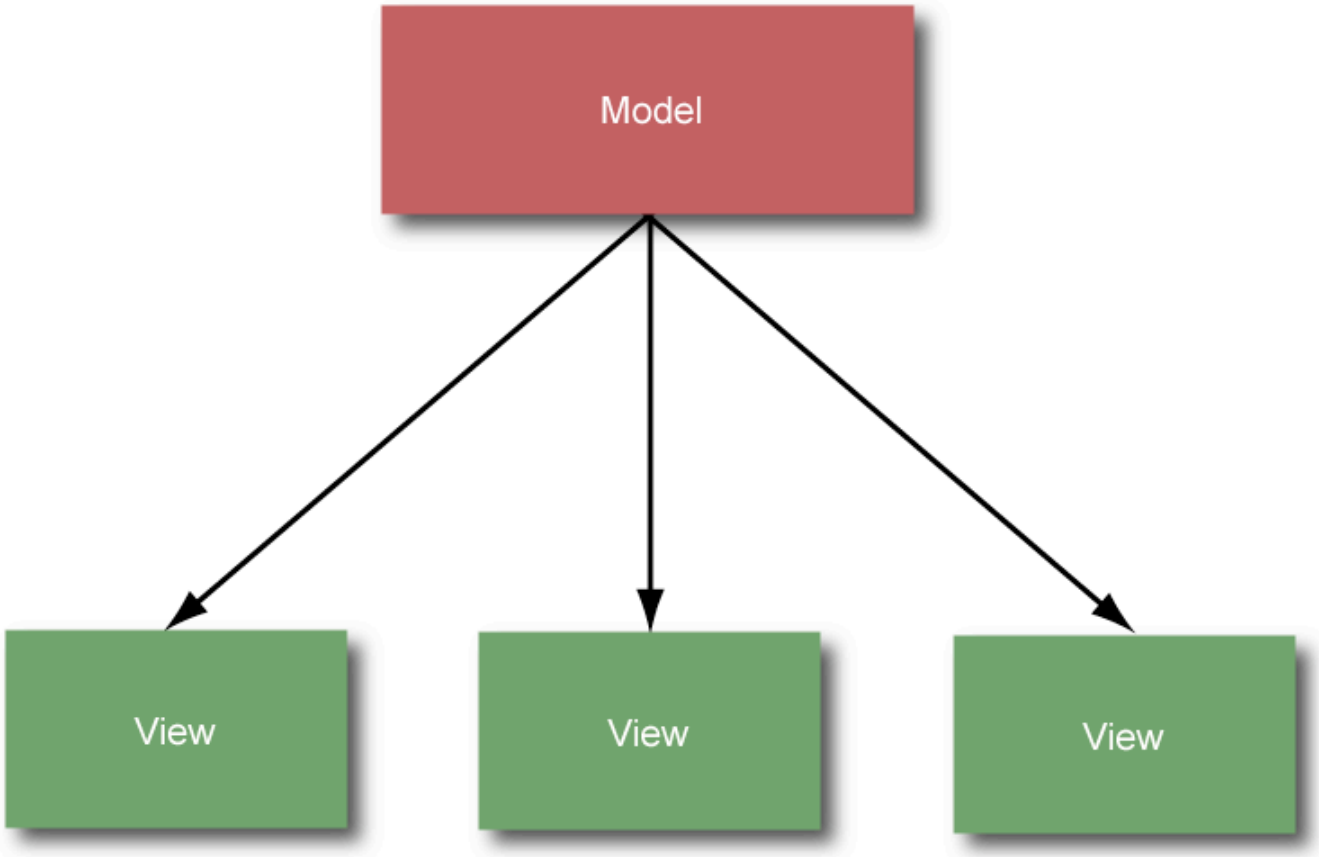
Agenda

- A brief history lesson
- Presenter pattern in detail
- MVP and RCP
- RAP / RCP dual use demo

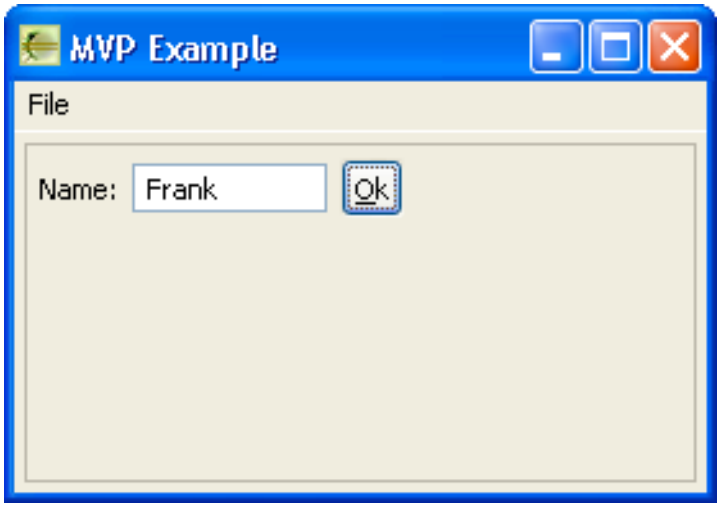
UI architecture

- Forms and controls
- Model-View-Controller (MVC)
- Model-View-Presenter (MVP)

Model / View separation



Forms and controls

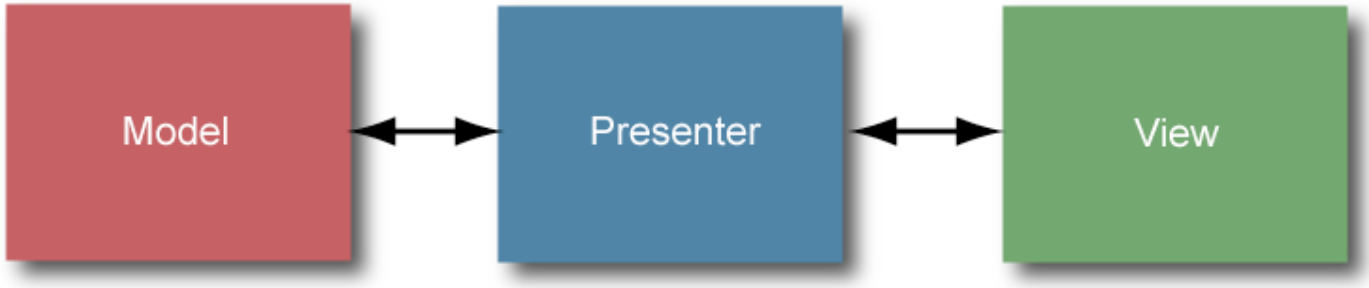



```
Text name = new Text(parent, SWT.BORDER);
name.addModifyListener(new ModifyListener()
        public void modifyText(ModifyEvent e) {
                model.setName(nameText.getText());
        }
});
```

Model-View-Controller (MVC)

What is a controller, anyway?

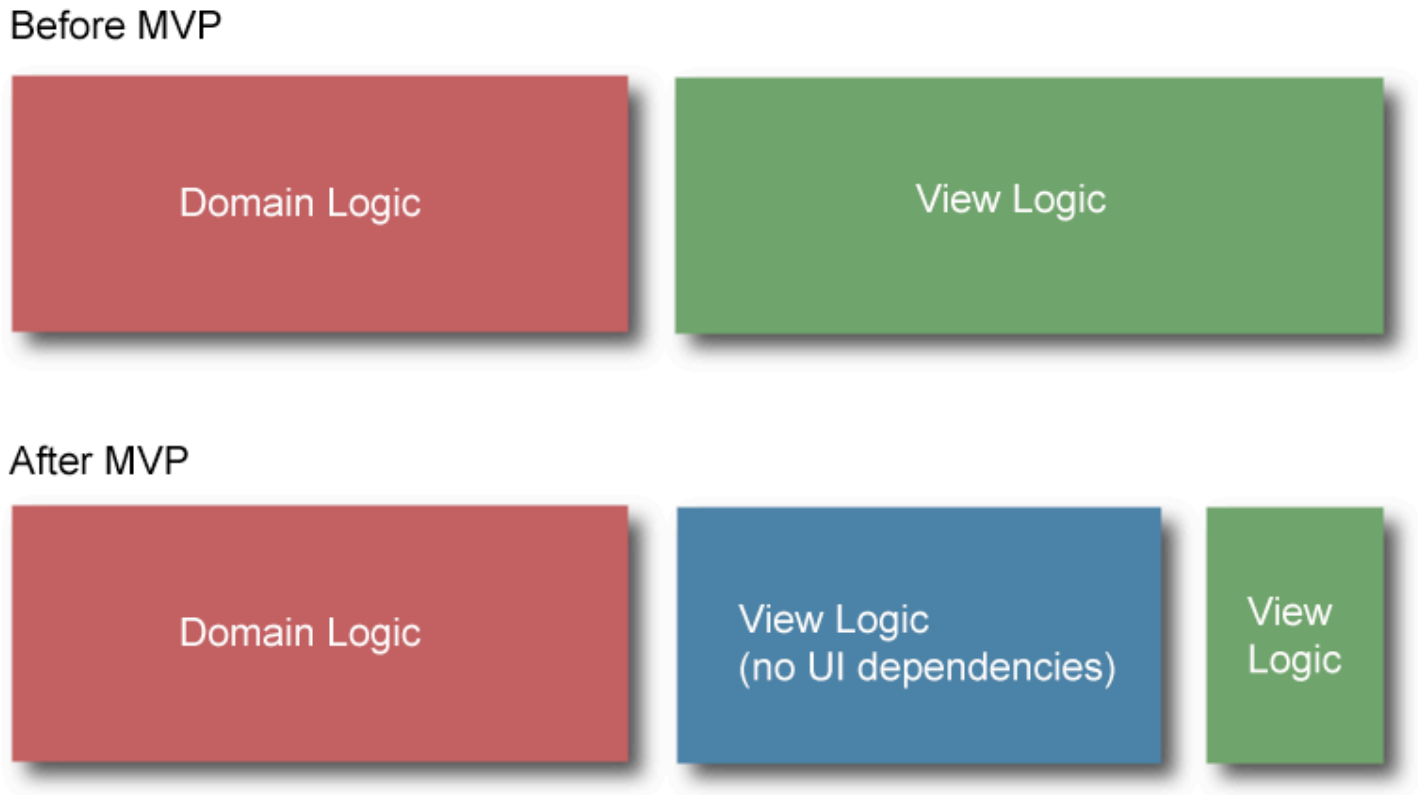
Model-View-Presenter (MVP)



Agenda

- A brief history lesson
- Presenter pattern in detail
- MVP and RCP
- RAP / RCP dual use demo

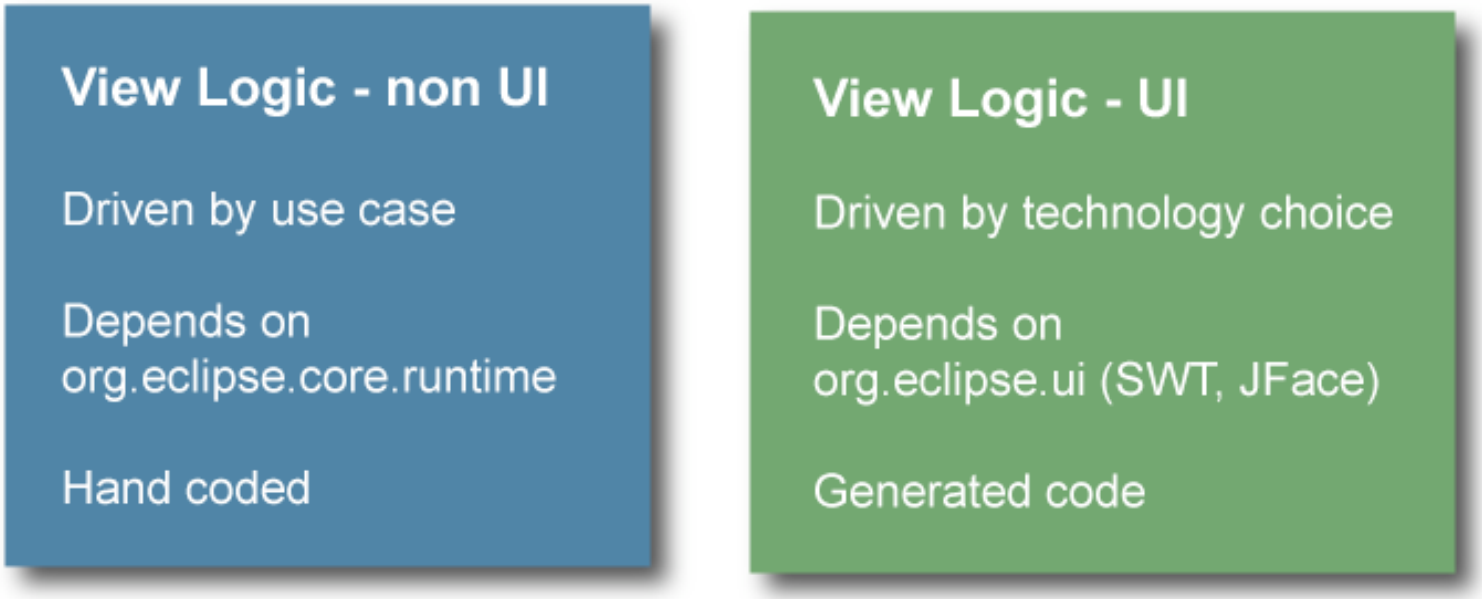
What is MVP?



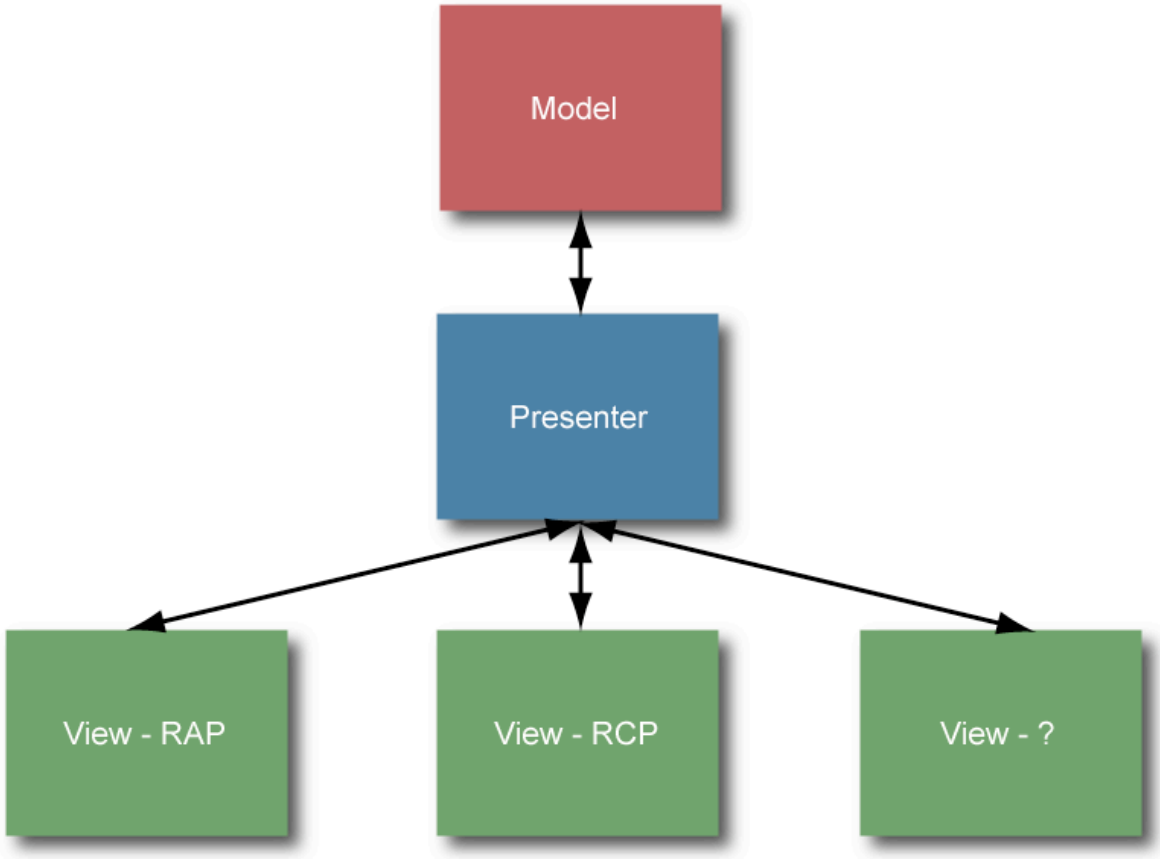
Why MVP?

- Manage complexity
- Flexibility
- Testability

Managing complexity



Flexibility



Testability is hard

- UI testing is hard
- Tooling isn't there yet
- Unit tests must be easy and quick to run

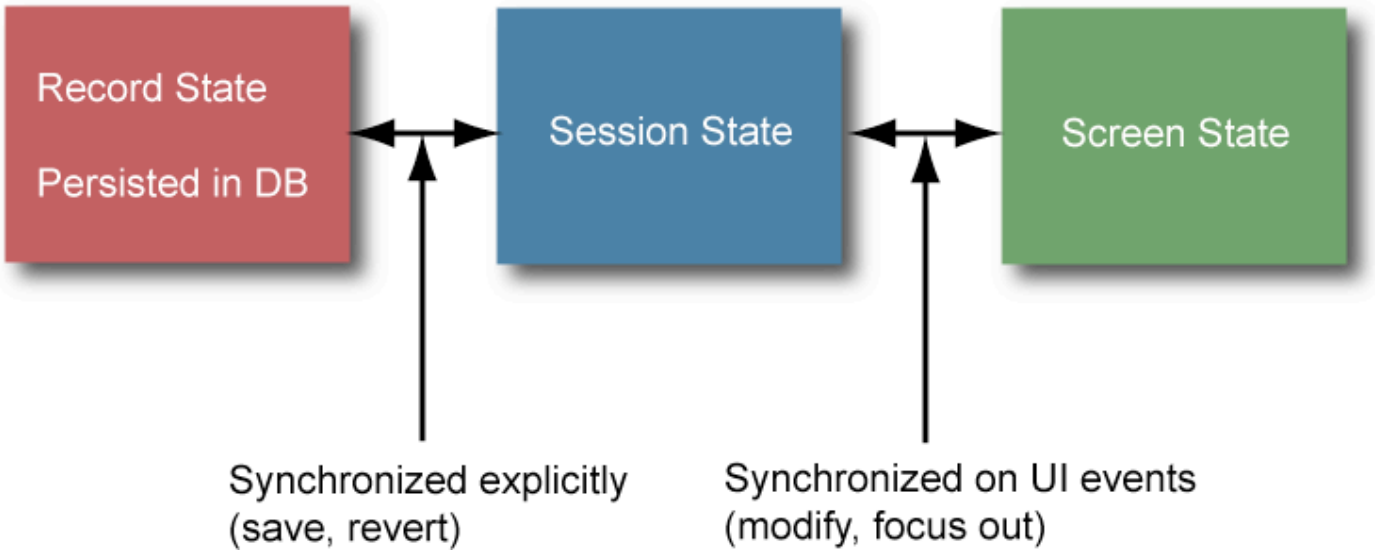
> *"Any object that is difficult to test should have minimal behavior."*
>
> - Martin Fowler

Acceptance testing

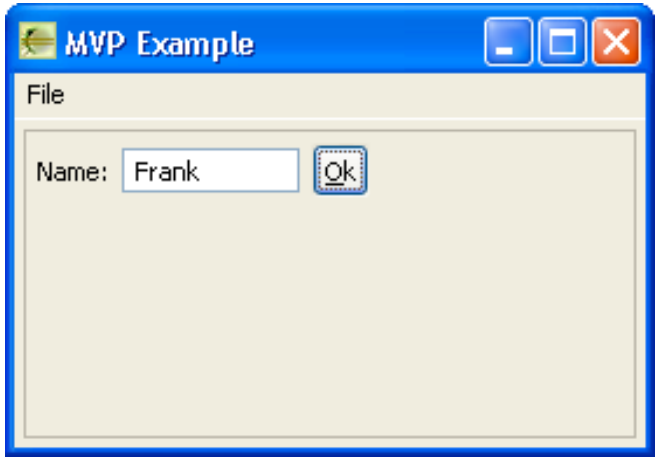
```
public class MyPresenter {

   setProperty(String property)
   getValidationMessages()
}
```

```
public class MyPresenterTest {

   @Test
   testUseCase() {
      MyPresenter p = new
         MyPresenter();

      p.setProperty("value");
      assertEquals(0,
         p.getValidationMessages().
         size())
   }
}
```

MVP and data

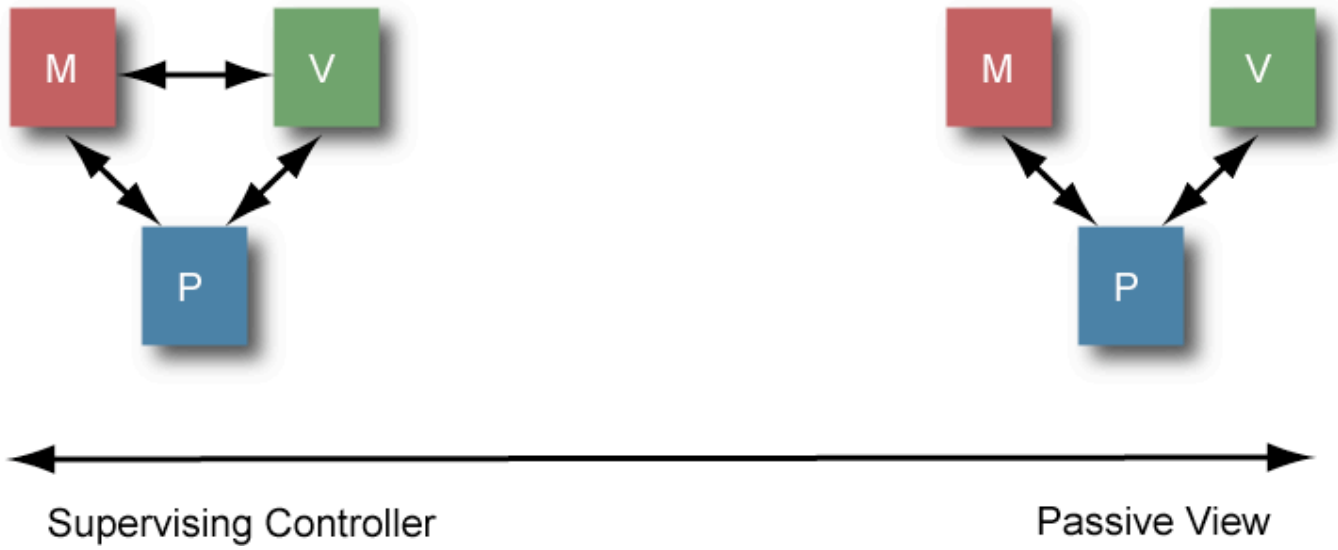


MVP and logic

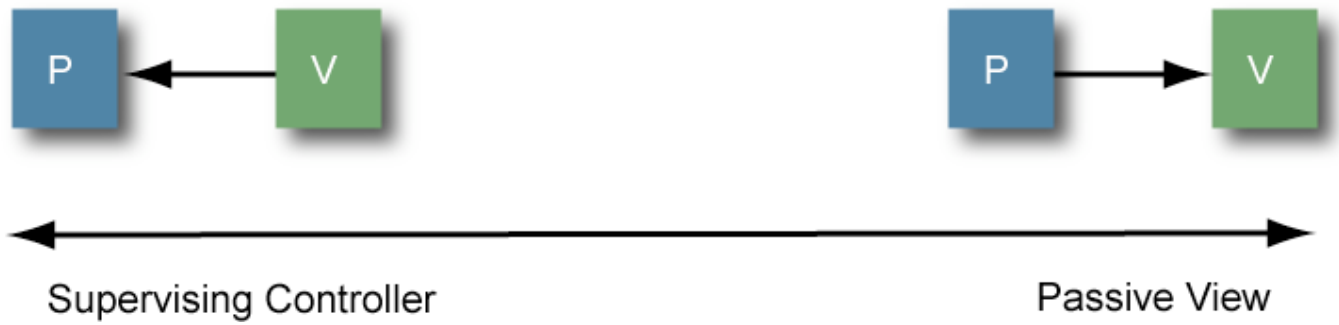



```
public class ExamplePresenter {

    public void setName(String name)

    public boolean getOkEnabled()

}
```

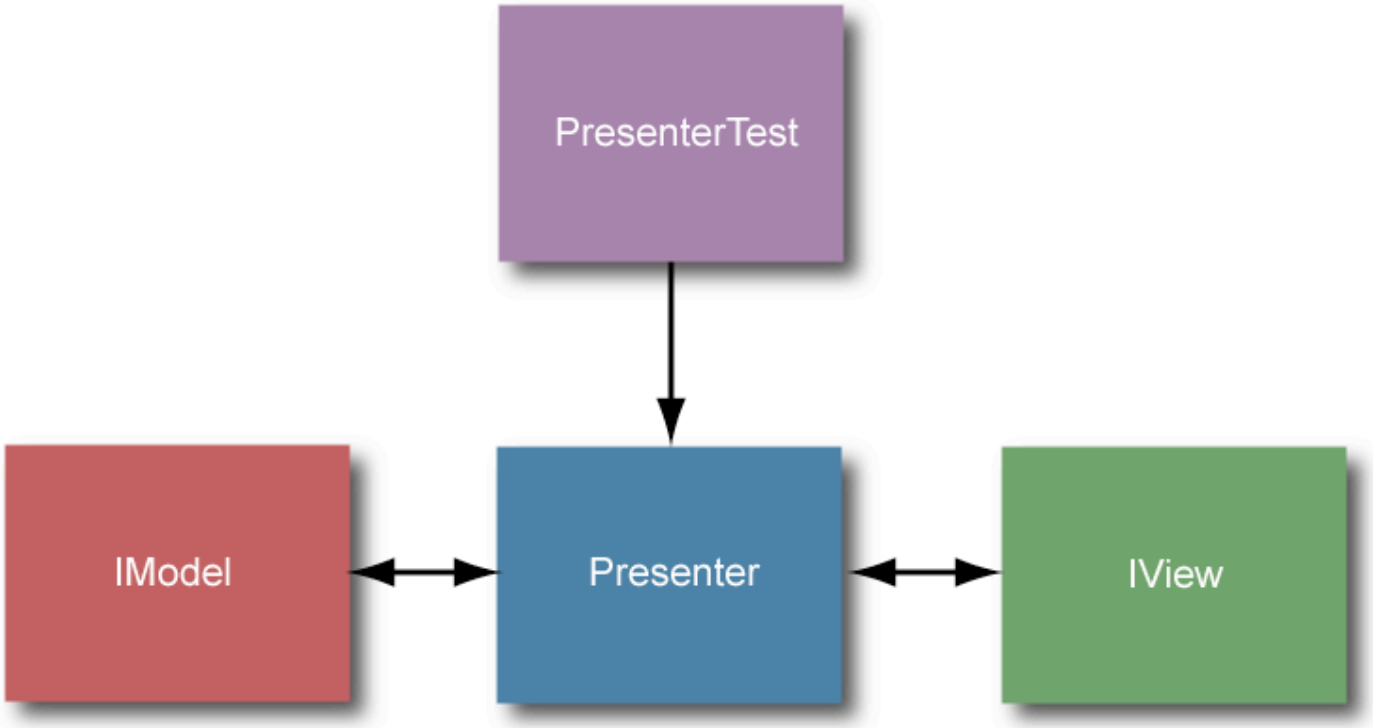
MVP continuum



Push vs pull



Passive view testing



Sample test case

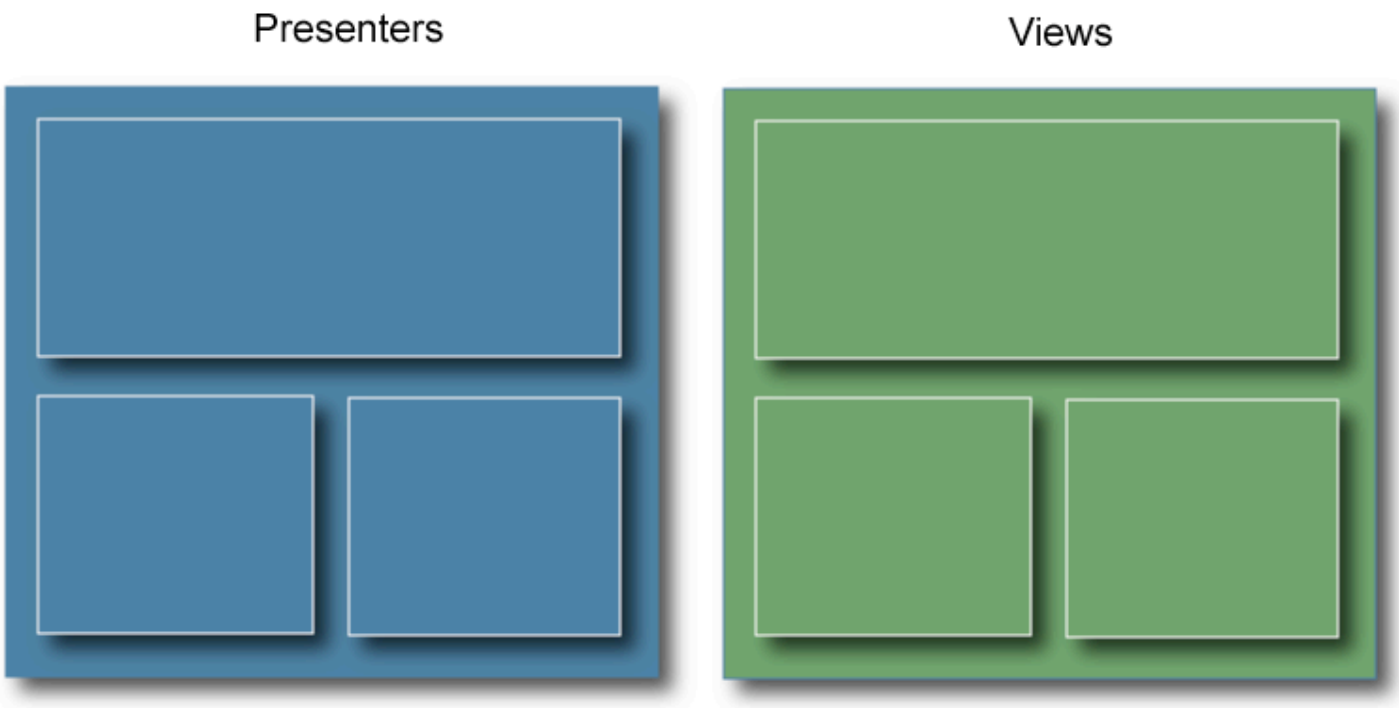
Presenter

```
public class ExamplePresenter {

   private IView view;

   public ExamplePresenter(IModel model,
      IView view) {
      this.view = view;

   public void setName(String name)
      this.view.setOkEnabled(name !=
         null);
```

Presenter Test Case

```
public class ExamplePresenterTest {

   protected void testSetName() {
     // create mock model and view

     ExamplePresenter presenter = new
       ExamplePresenter(mockModel,
       mockView);

     presenter.setName("Frank");

     assertTrue(mockView.isOkEnabled());
```

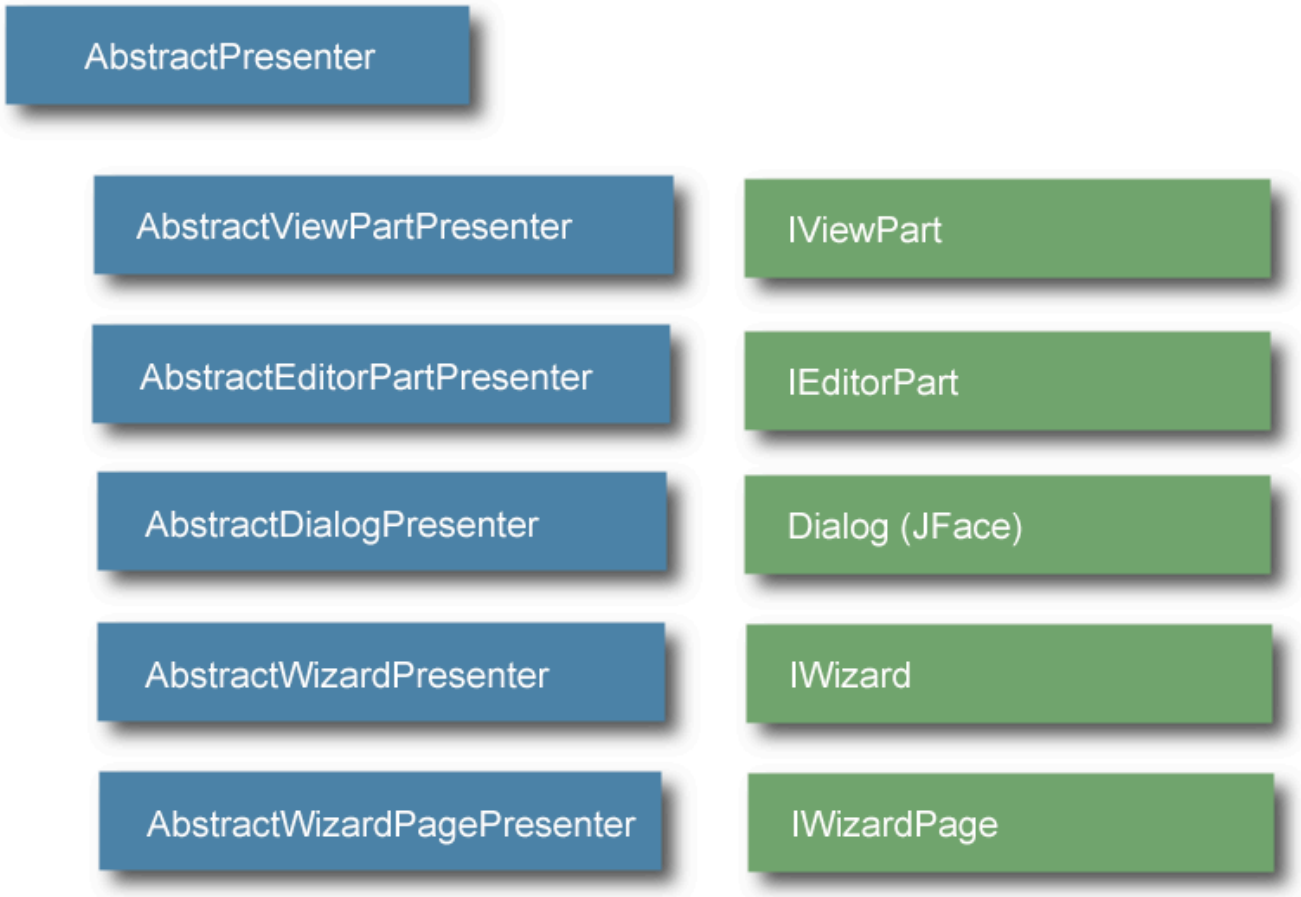
Composite presenters



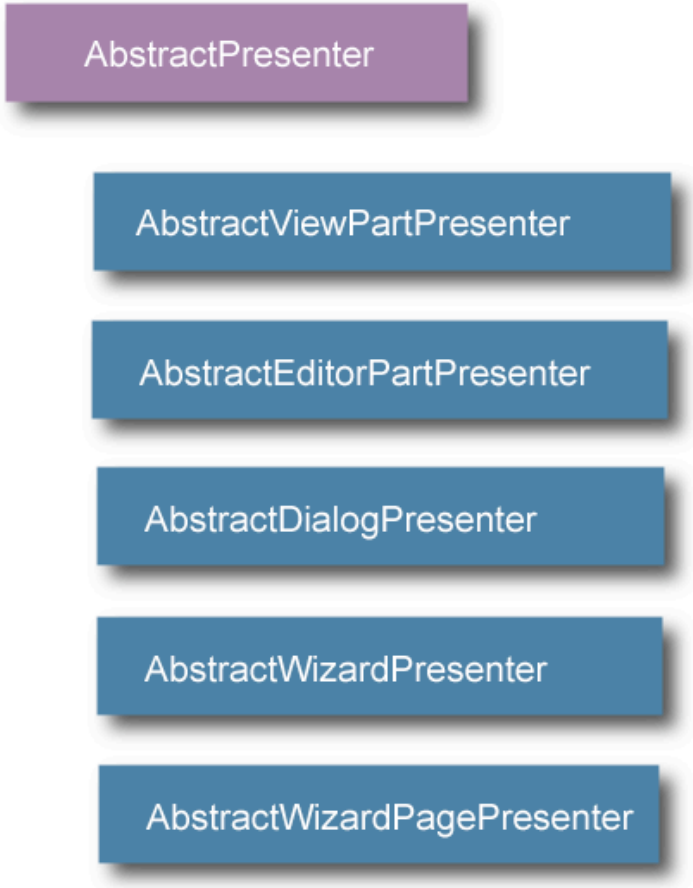
Agenda

- A brief history lesson
- Presenter pattern in detail
- MVP and RCP
- RAP / RCP dual use demo

Presenter hierarchy



AbstractPresenter



- Composite pattern
- Property change management (if data binding is used)

AbstractViewPartPresenter

- Create in IViewPart.init()
- Route saveState()
- Mementos
- Filtering
- Sorting

AbstractEditPartPresenter

- Create in IEditorPart.init() or extract from editor input
- Dirty state management
- Save and revert
- Form header messages
- Can have nested page and section presenters

AbstractDialogPresenter

- Create in constructor
- Handle button enablement
- Handle button presses

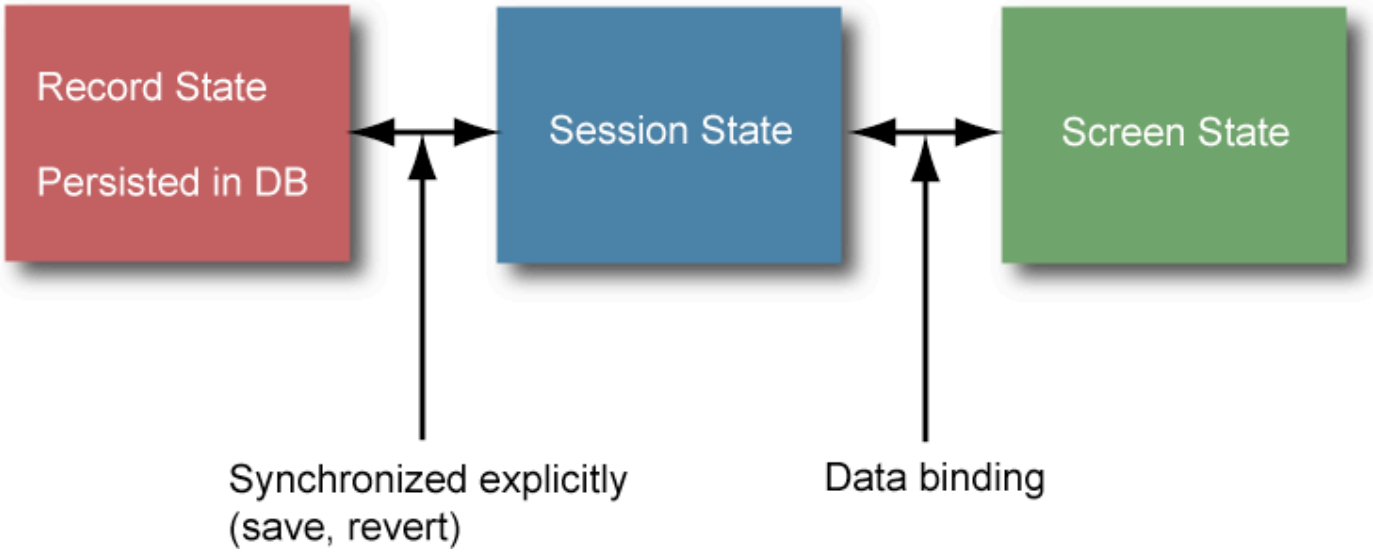
AbstractWizardPresenter

- Page management (page presenters or constants)
- Determines if wizard can finish
- Performs finish or cancel logic

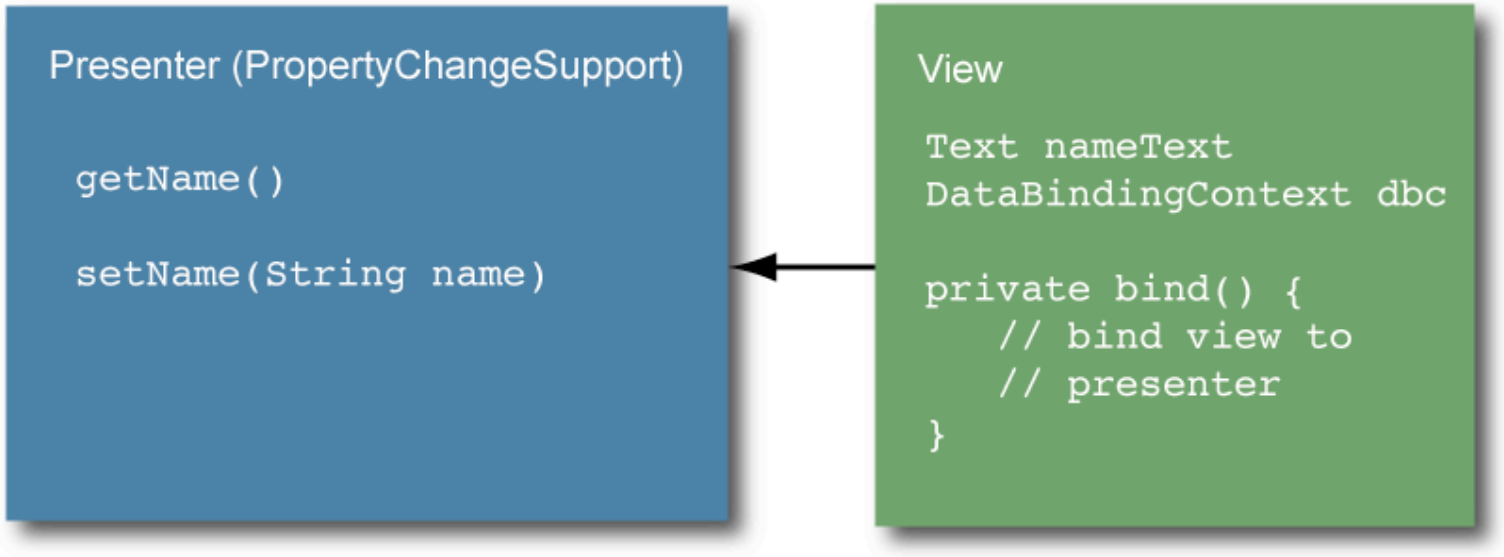
AbstractWizardPagePresenter

- Page management
- Handles isPageComplete()

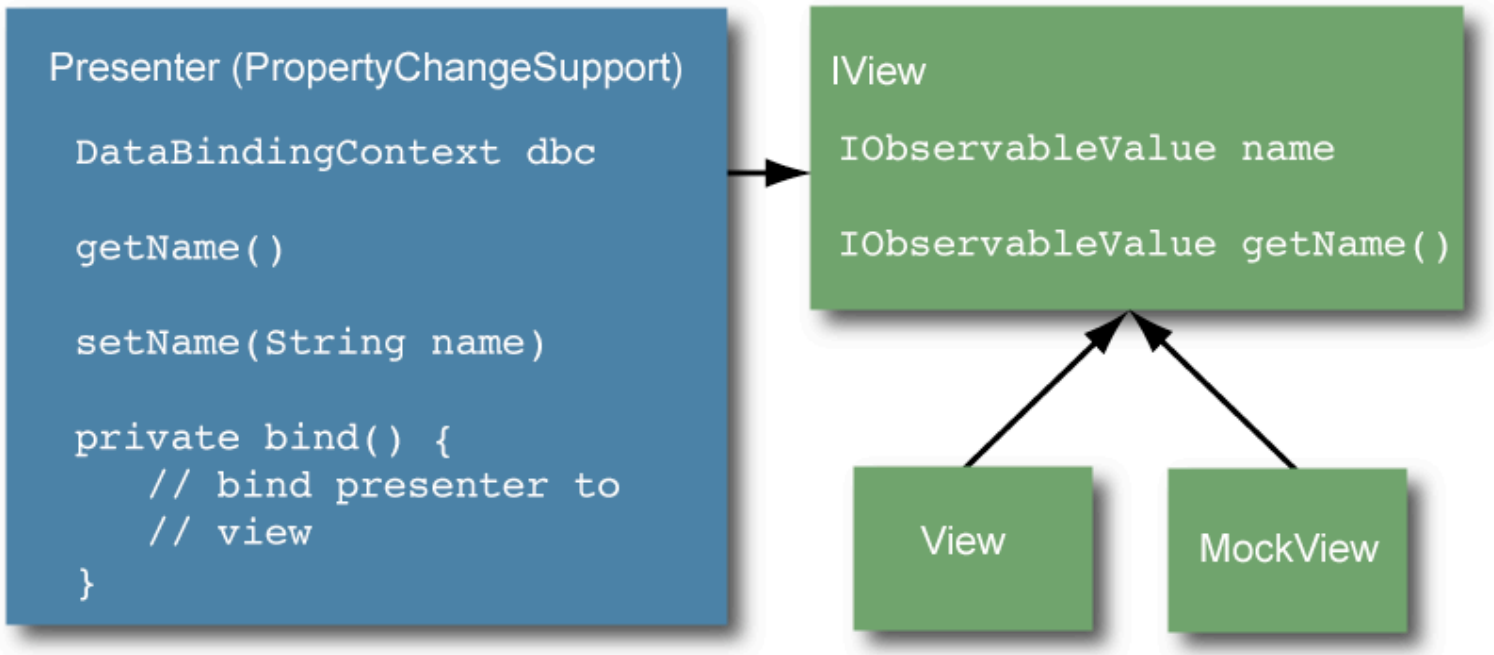
Data binding



Data binding - view in control

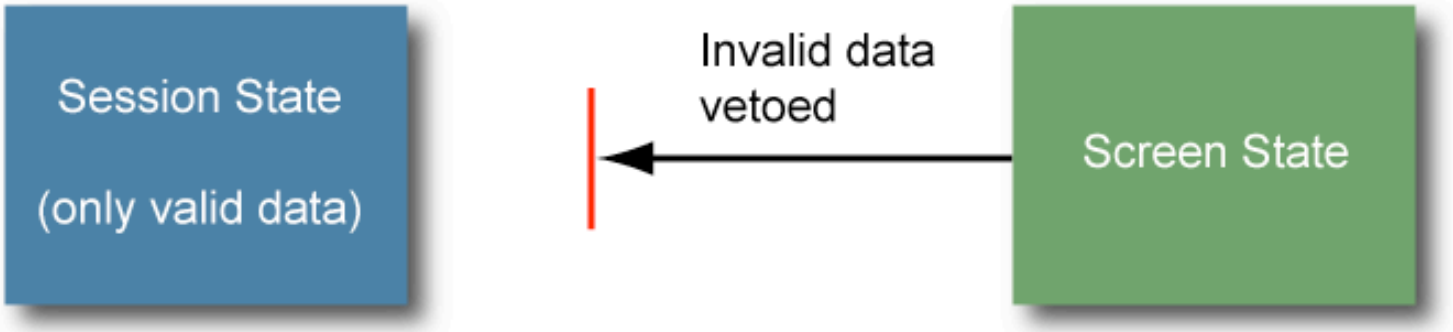


Data binding - presenter in control

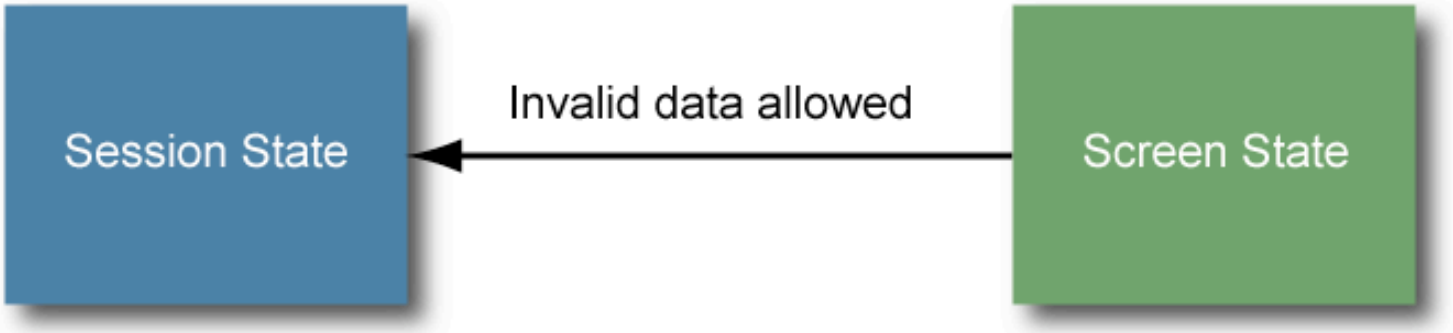


RCP Quickstart

Data binding validation



Presenter validation



Data binding - beyond data

- Control enablement
- Selections
- View still needs to call presenter to take action (e.g. okPressed)
- Presenter can set a listener into view if full control is desired

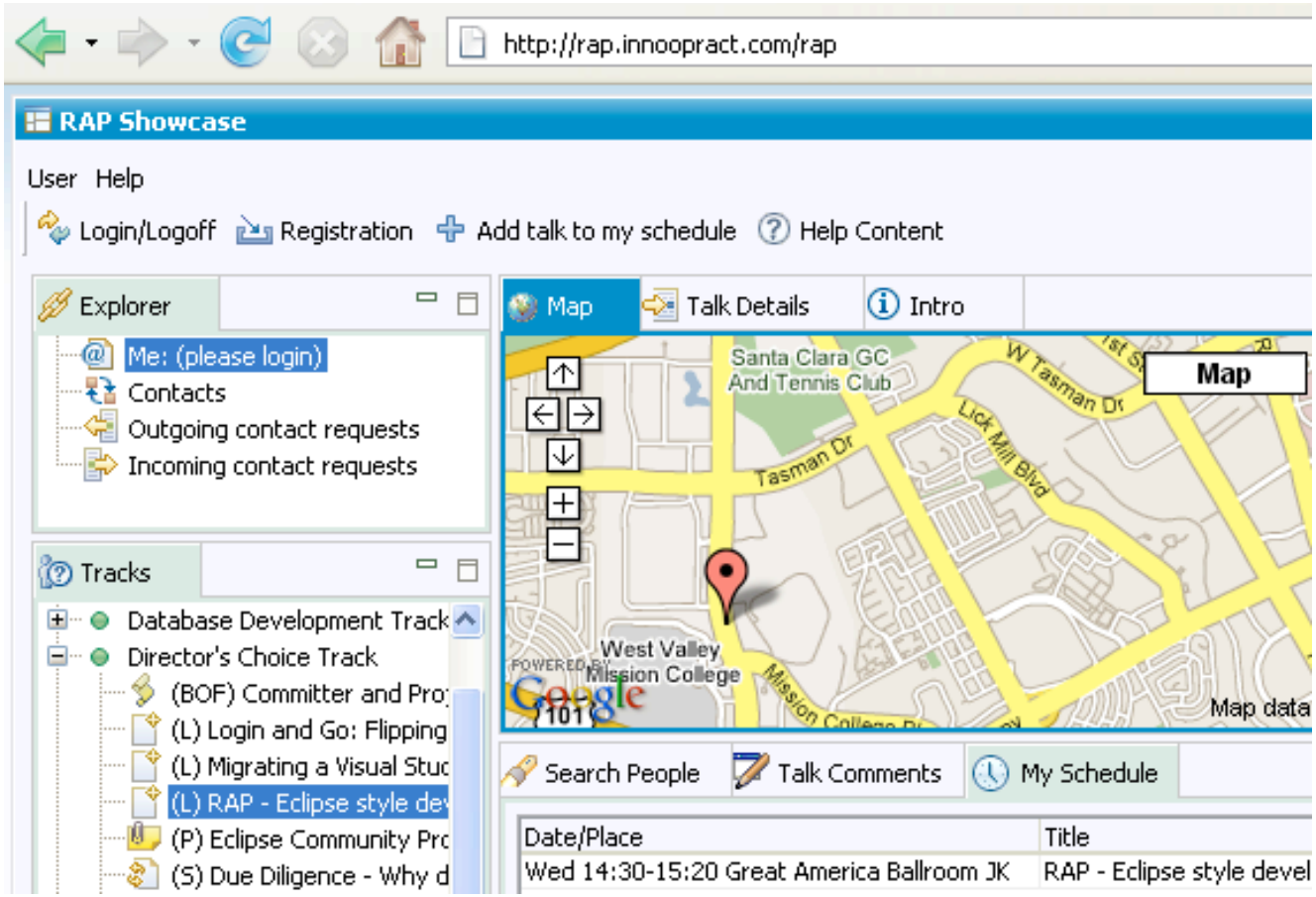
Agenda

- A brief history lesson
- Presenter pattern in detail
- MVP and RCP
- RAP / RCP dual use demo

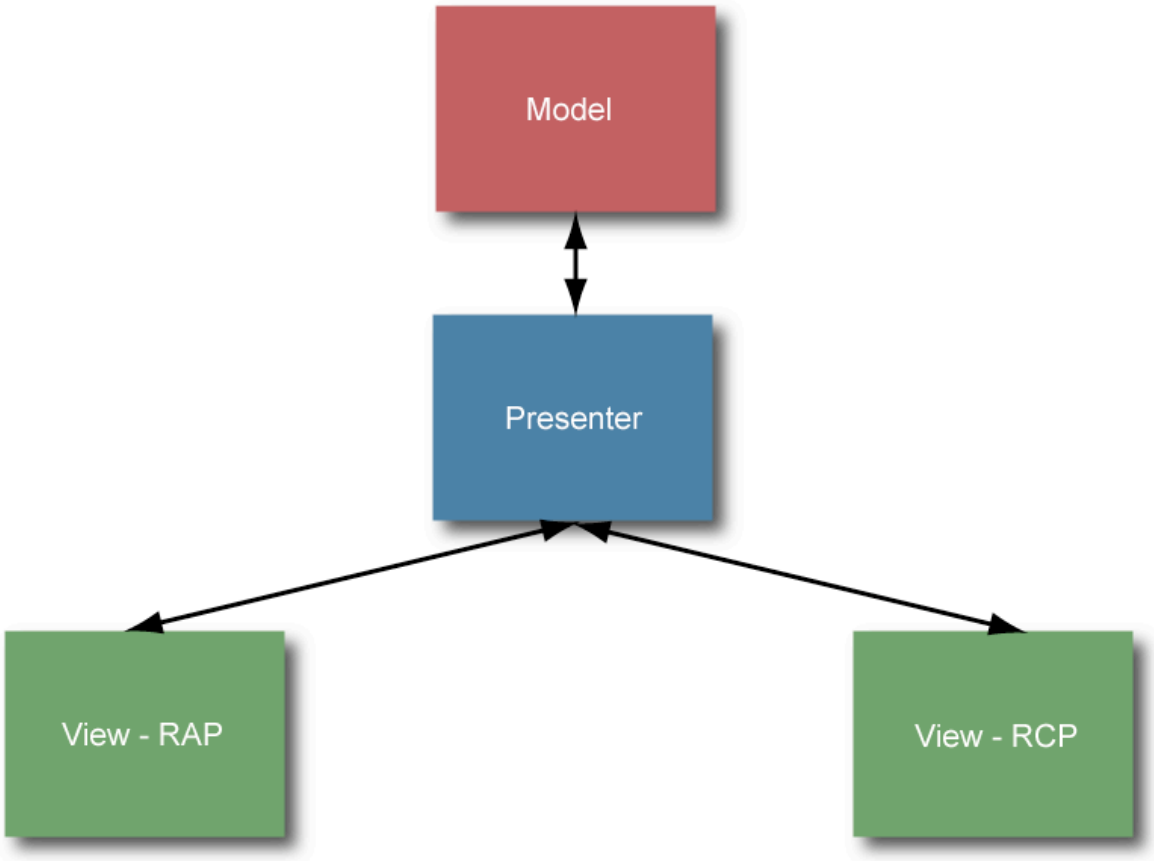
What is RAP?

- Rich Ajax Platform
- Parallel UI toolkit including SWT, JFace and RCP Workbench
- UI classes run on server and emit Javascript to browser
- Now supports data binding!
- http://www.eclipse.org/rap/

RAP Example



Presenter dual use



MVP Summary

- Replacement for MVC
- Flexibility and testability
- MVP and RCP work well together
- MVP prepares you for the future

More Information

Check out my website:

www.rcpquickstart.com

Detailed notes for presentation at:

www.rcpquickstart.com/training/presentations/mvp-and-rcp-ew2008

Email me:

patrick@rcpquickstart.com