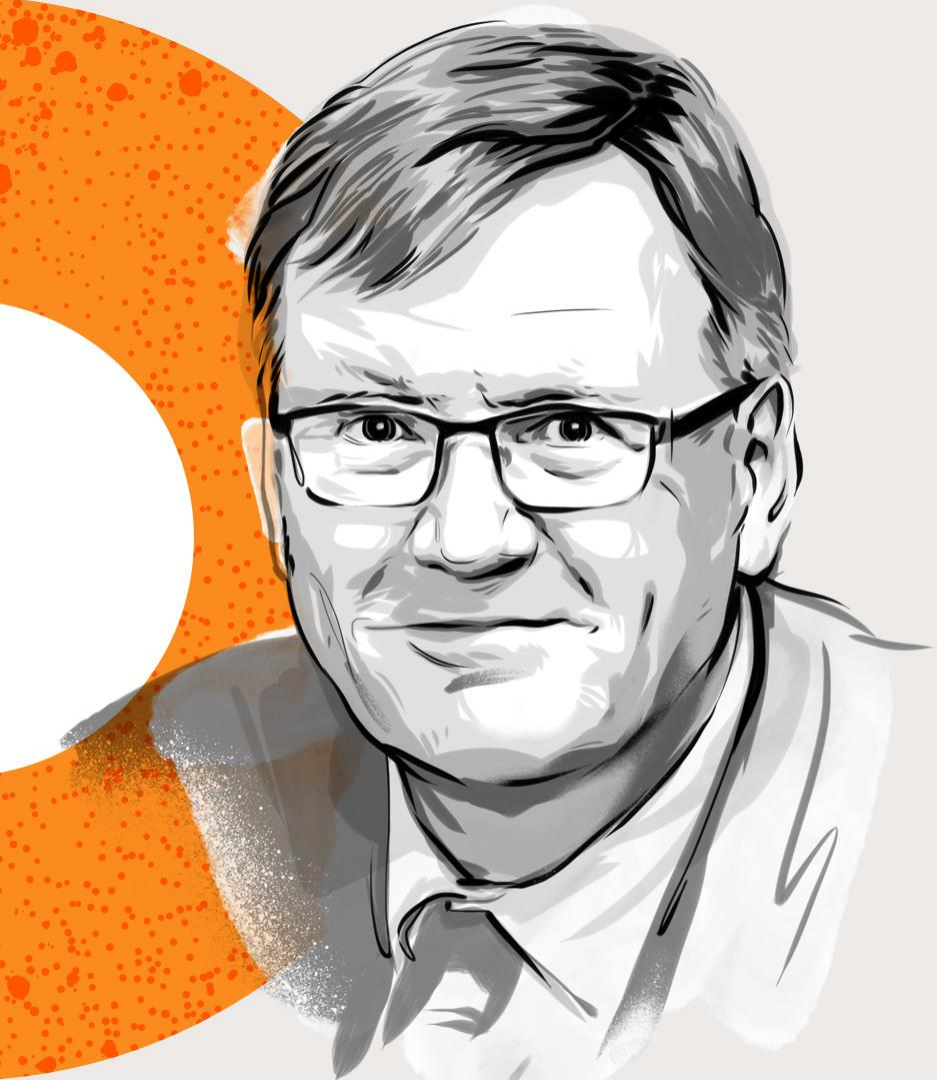


UGM Boston Oct. 4th 23

Chemaxon

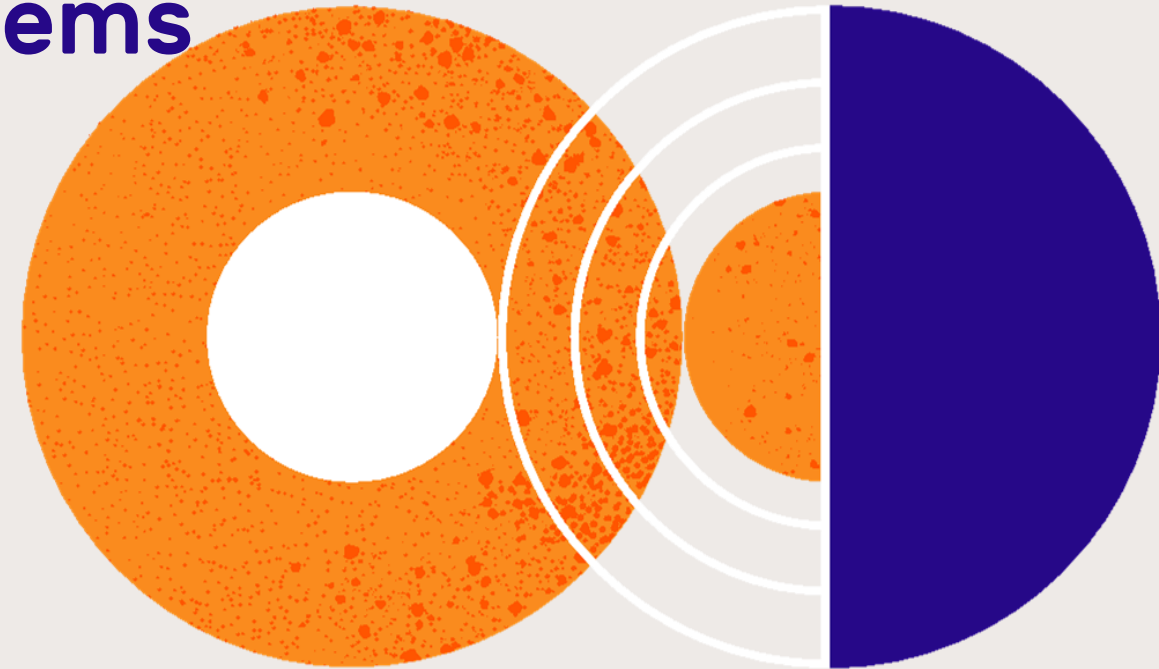
ChemistryView: how Pfizer simplifies its legacy desktop chemistry systems

David Klatte
Senior Director of Scientific Solution Engineering, Pfizer

ChemistryView: how Pfizer simplifies its legacy desktop chemistry systems

UGM Boston Oct. 4th 23
Chemaxon

David H. Klatte, Ph.D.
Senior Director
Scientific Solution
Engineering, Pfizer Digital



Overview

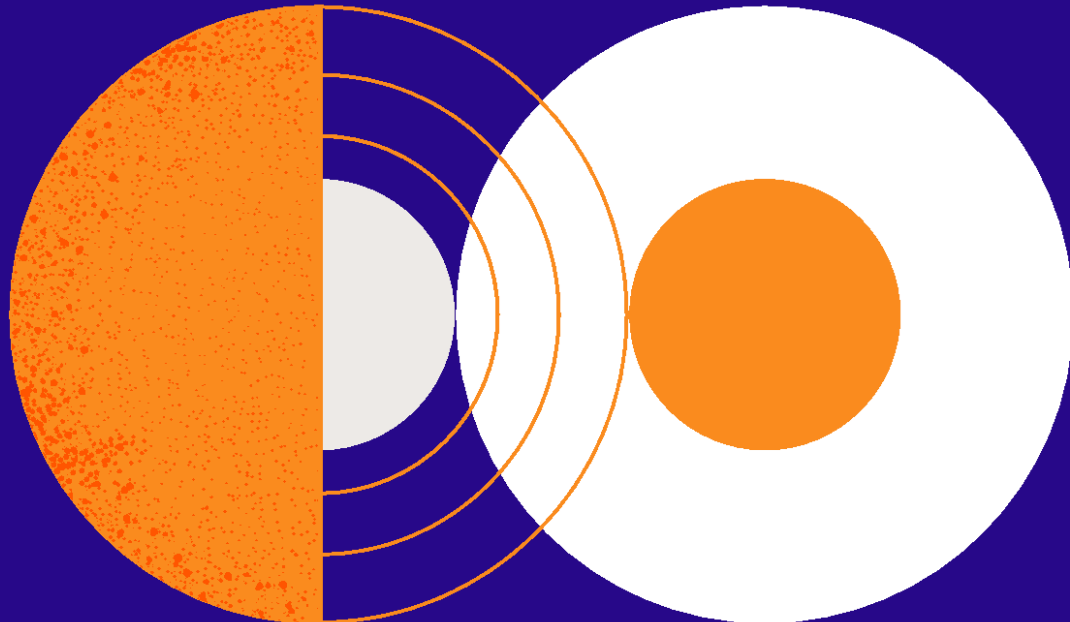
1. The Use Case. What is our environment and what problem were we trying to solve?
2. ChemistryView. A synopsis of our solution.

UGM Boston Oct. 4th 23

Chemaxon

Part 1: The Use Case

The problem we were trying to solve.



Pfizer’s Desktop Java Apps

We have quite a few

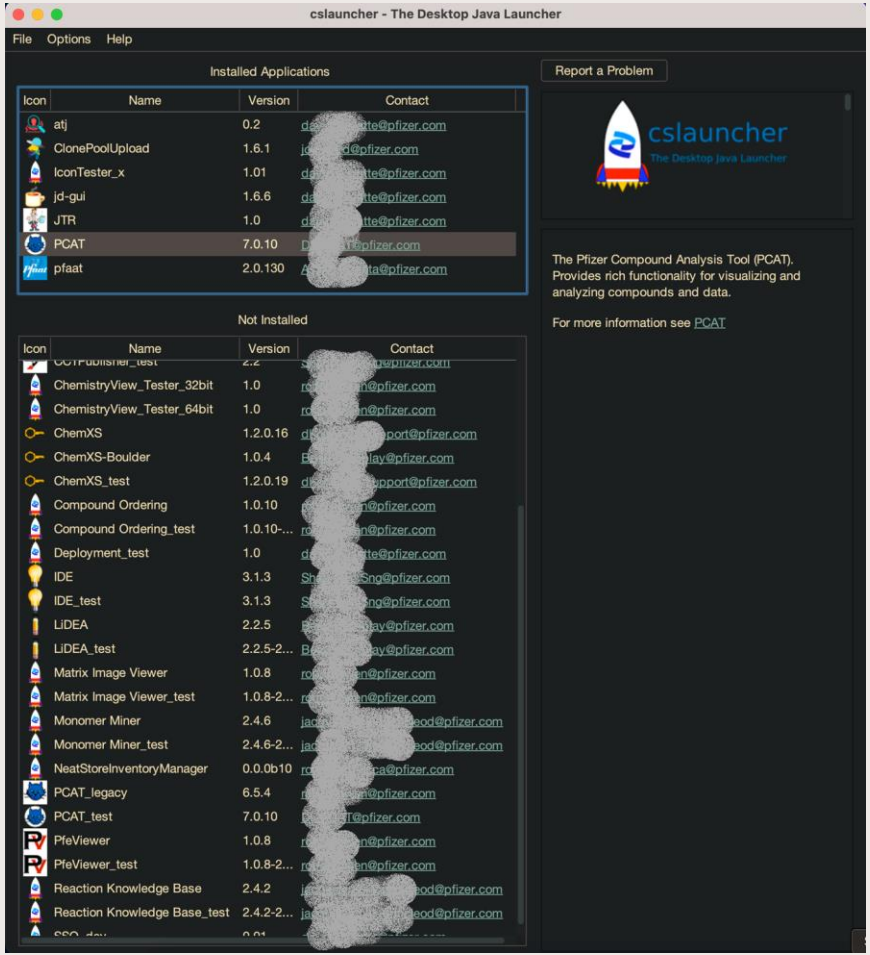
While we rarely build new solutions this way, we still do so when utilizing the power of the user’s desktop is a key requirement.

And we have a lot of legacy Java desktop applications that are still used and are actively maintained.

cslauncher

We have so many that we have built a dedicated java launcher to manage them.

Many of these are chemistry centric applications.



Cslauncher…

- Is not the topic of this presentation, but it is the environment into which the Java applications using ChemistryView are deployed.
    - A brief overview of what it is may be useful.

Cslauncher…

- Manages Java versions (7 to 17)
  - We have total control over java versions – they are not installed/integrated with the operating system
  - JREs are installed only as needed and on demand

Cslauncher...

- Allows for Platform variability
    - Specify different libraries or start up parameters for different platforms. This allows use, for example, of native libraries specific to each platform.

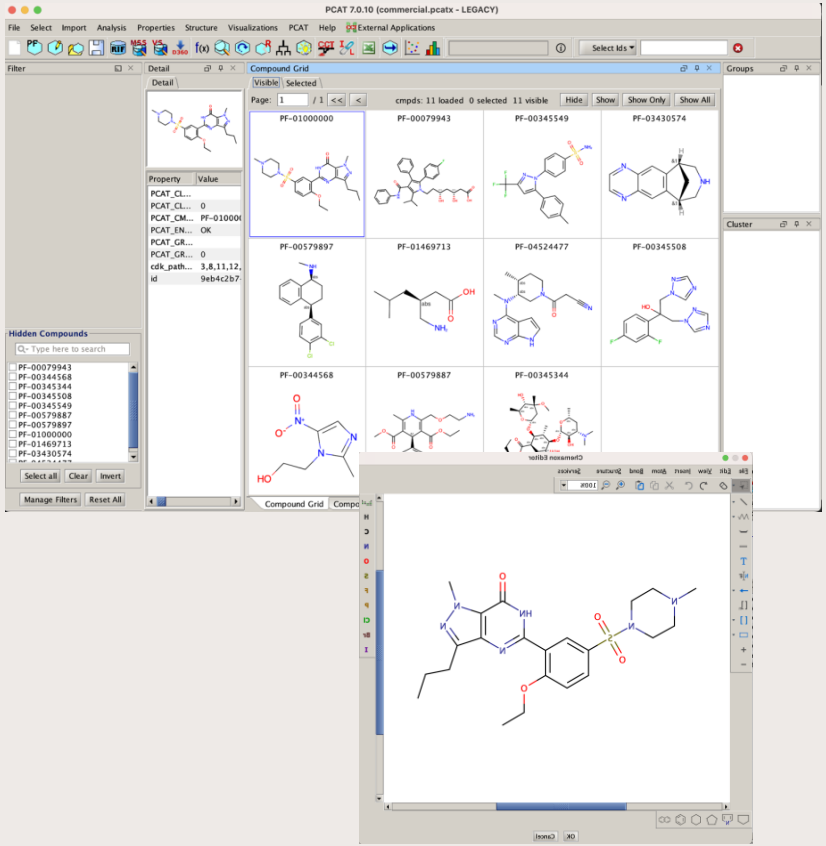
Cslauncher...

- Permits start up customization via profile
  - Works for different platforms automatically, but you could also have special high memory profiles for power users.
  - Anything you could do with a custom Java command-line is possible.

ChemistryView Requirements

Chemistry applications tend to share many common requirements

Goal: Build a single library that meets these common requirements that can be reused across applications.



Requirement: Cross Platform

Red Hat Enterprise Linux 8

Sometimes the preferred (chemistry) solution/library/tool is not available on every platform.

Support Major Platforms

Mac OS

Requirement: User Preference

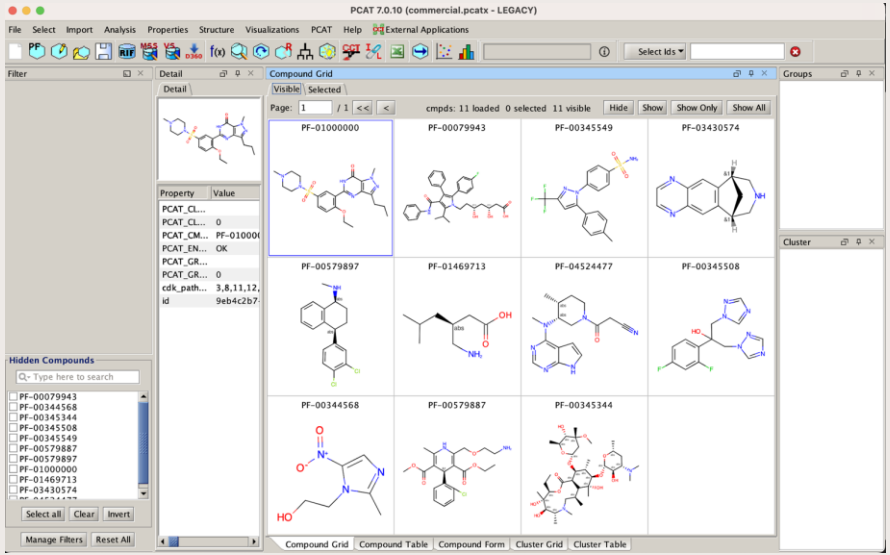
For each chemistry requirement, there is the possibility that users will have different needs.

Support a plugin architecture.

Requirement: Structure Display

All chemistry desktop apps need to be able to display structures. But users have different preferences/needs and sometimes want custom viewers. And sometimes platform is an issue.

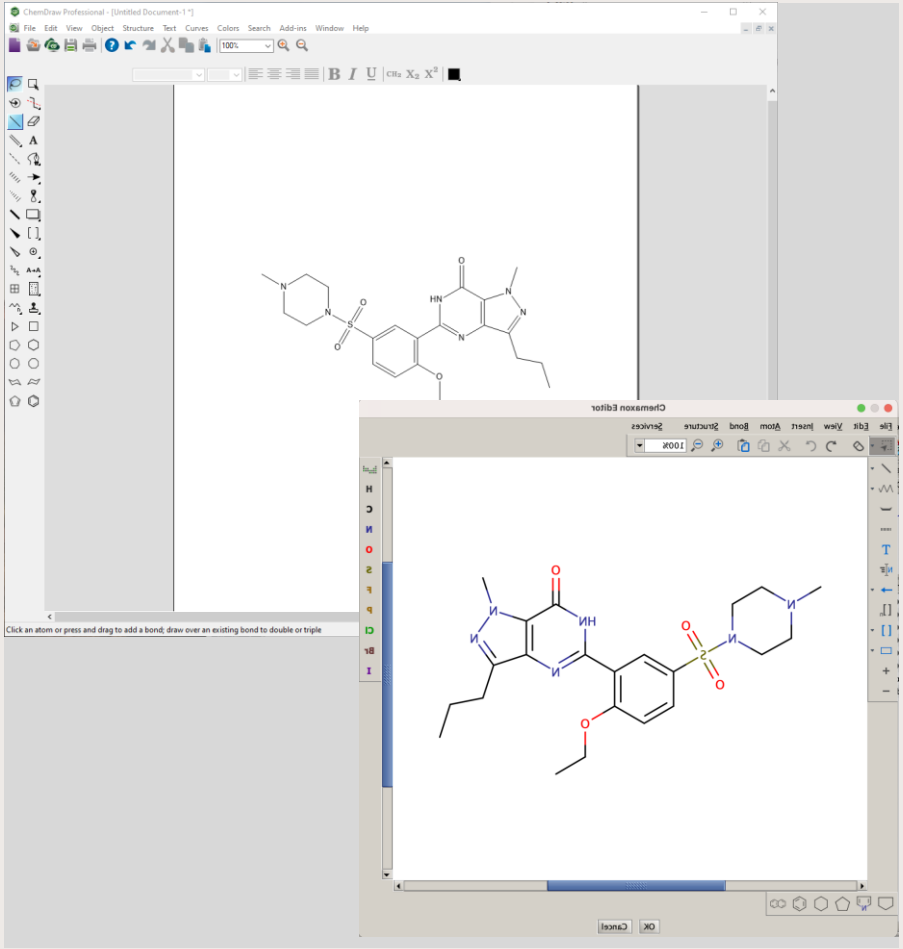
Support multiple viewer plugins.



Requirement: Structure Editing

Most chemists prefer ChemDraw, but it is windows only. Chemaxon MarvinBeans sketcher is excellent, and it works everywhere.

Support multiple chemistry drawing plugins



Requirement: Chemistry Toolkit

While some apps will have very specific needs, all will need general molecule methods that are original document centric and that handle enhanced stereochemistry correctly.

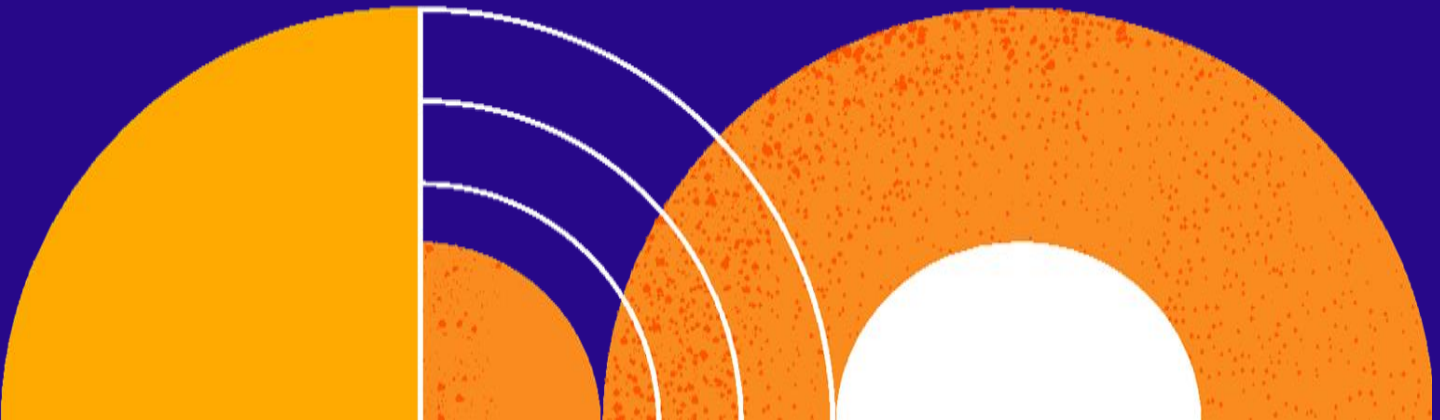
Support Basic Chemistry Toolkit

UGM Boston Oct. 4th 23

Chemaxon

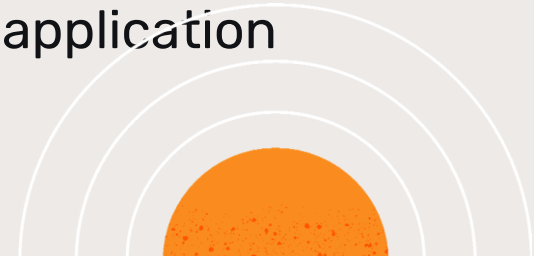
ChemistryView

The Java library design



Overview

A Java swing component for editing and rendering chemistry related objects such as molecules and reactions. This component utilises a plugin-based architecture that can be used to specify what technology the view uses for renderer/editing this chemical information, for example a ChemDraw editor (or a Marvin editor) and a render that displays the information as an image (or MarvinView, or OEDepict). The plugins used by the view can be changed at run time based on current application needs.



ChemistryView Interfaces

ChemistryView is based around a set of Java Interfaces (or abstract classes), each of which has multiple concrete implementations.

ChemistryDocumentEditor Interface

Methods for opening a chemistry sketcher

```
/***
 * A ChemicalSketcherEditor allows for a user to edit a ChemicalReaction in a sketching context
 *
 */
public interface ChemistryDocumentEditor {
  public boolean isEnabled();
  public void openEditor(ChemistryView chemistryView) throws ChemistryViewPluginException,
      ChemistryViewConversionException, ChemistryViewFormatException;
  public void setEditorPreferredSize(Dimension windowSize) throws ChemistryViewPluginException;
  public void onWindowReize(Dimension newSize) throws ChemistryViewPluginException;
  public boolean userClickedOK();
  public void setNameFieldEnabled(boolean enabled);

}
```

ChemistryDocument Renderer Interface

A much more complicated interface that supports a host of interactions with a ChemistryView Object (editable or not)

```
public interface ChemistryDocumentRenderer extends PreferenceChangeListener {

  public JComponent getChemicalRendererComponent();
  public Dimension getPreferredSize();
  public void setPreferredSize(Dimension size);
  public abstract void setIsEditable(boolean isEditable);
  public abstract boolean isEditable();
  public void setChemistryDocument(ChemistryDocument chemDoc)
      throws ChemistryViewConversionException;
  public abstract void setChemistryView(ChemistryView view);
  public abstract List<JMenuItem> getRenderingMenuItems();
  public abstract boolean supportsChangingAbsoluteFlagVisibility();
  public abstract void showAbsoluteFlag(boolean show);
  public abstract boolean getAbsoluteFlagIsVisible();
  public abstract boolean getDisplaysAtomNumbers();
  public abstract void setDisplaysAtomNumbers(boolean displays);
  public abstract boolean supportsChangingColorScheme();
  public abstract void setRenderingColorScheme(AtomColoring scheme);
  public abstract AtomColoring getMolColorScheme();
  public abstract boolean supportsChangingAtomSize();
  public abstract void setDisplayAtomSize(double size);
  public abstract double getDisplayAtomSize();
  public abstract boolean supportsChangingHydrogenDisplay();
  public abstract void setHydrogenDisplay(ExplicitHydrogens hyd);
  public abstract ExplicitHydrogens getHydrogenDisplay();
  public abstract boolean supportsChangingPeptideDisplay();
  public abstract void setPeptideDisplay(PeptideDisplay pd);
  public abstract PeptideDisplay getPeptideDisplay();
  public abstract boolean supportsChangingExpandingSGroups();
  public abstract void setExpandSGroups(boolean state);
  public abstract boolean getExpandSGroups();
  public abstract boolean supportsChangingChiralitySupport();
  public abstract void setChiralitySupport(ChiralitySupport csFlag);
  public abstract ChiralitySupport getChiralitySupport();
  public abstract boolean supportsChangingRenderingStyle();
  public abstract void setRenderingStyle(Rendering rendering);
  public abstract Rendering getRendering();
  public abstract void setBackground(Color bg);
  public abstract void setSelectionColor(Color color);
  public abstract Color getSelectionColor();
  public abstract Component getPaintableComponent();
  public abstract boolean getSupportsSelection();
  public abstract boolean getSelectionEnabled();
  public abstract void setSelectionEnabled(boolean selection);
  public abstract List<Integer> getSelectedAtomIndicies();
}
```

ChemistryViewPlugin

The core plugin class with multiple implementations

```
public abstract class ChemistryViewPlugin {

  public static boolean isSupported() {
    return true;
  }
  public ChemistryEditorResizeDelegate editorResizeDelegate;
  public abstract String getPluginName();
  public abstract String getPluginDescription();
  public abstract boolean supportsRendering();
  public abstract boolean supportsEditing() throws ChemistryViewPluginException;
  public abstract ChemistryDocumentEditor createEditor() throws ChemistryViewPluginException;
  public abstract ChemistryDocumentRenderer createRenderer();
  public abstract Image getIcon();
  public abstract boolean requiresLicense();
  public abstract void setLicense(Object licenseData);
  /** etcetera */
}
```

ChemistryView Toolkit Interface

A host of methods for manipulating and converting molecules and reactions.

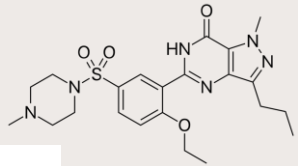
Again, with multiple concrete implementations.

```
public interface ChemistryViewToolkit {

  public List<ChemistryDocumentFormat> getSupportedDocumentInputFormats();
  public boolean supportsDocumentInputFormat(ChemistryDocumentFormat format);
  public List<ChemistryDocumentFormat> getSupportedDocumentOutputFormats();
  public boolean supportsDocumentOutputFormat(ChemistryDocumentFormat format);
  public List<ChemistryDocumentFormat> getSupportedDocumentOutputImageFormats();
  public boolean supportsDocumentImageOutputFormat(ChemistryDocumentFormat format);
  public List<ChemistryObjectFormat> getSupportedMoleculeOutputFormats();
  public boolean supportsMoleculeOutputFormat(ChemistryObjectFormat format);
  public boolean supportsReactionOutputFormat(ChemistryObjectFormat format);
  public List<ChemistryObjectFormat> getSupportedReactionOutputFormats();
  public Object getBlankDocument();
  public ChemistryDocumentFormat getDefaultDocumentFormat();
  public byte[] getBlankDocumentTextFormat();
  public Object getDocumentInternalRepresentation(byte[] data, ChemistryDocumentFormat format)
      throws ChemistryViewConversionException;
  public Object cloneDocumentInternalRepresentation(Object representation)
      throws ChemistryViewConversionException;
  public byte[] exportDocumentToFormat(Object representation, ChemistryDocumentFormat format)
      throws ChemistryViewConversionException;
   public byte[] exportDocumentToDataFlavor(Object representation, DataFlavor flavor)
      throws ChemistryViewConversionException;
  public StandardizationResult standardizeMolecule(Object representation, StandardizeType type)
      throws ChemistryViewConversionException;
  public boolean isEmptyDocument(Object representation);
  public boolean isReactionDocument(Object representation);
  public List<ChemistryMolecule> getAllMoleculesFromDocument(Object representation)
      throws ChemistryViewConversionException;
  public Object getBlankObject();
  public ChemistryObjectFormat getDefaultObjectFormat();
  public byte[] getBlankObjectTextFormat();
  public Object getObjectInternalRepresentation(byte[] data, ChemistryObjectFormat format)
      throws ChemistryViewConversionException;
  public Object cloneObjectInternalRepresentation(Object representation)
      throws ChemistryViewConversionException;
  public byte[] exportObjectToFormat(Object representation, ChemistryObjectFormat format)
      throws ChemistryViewConversionException;
  public byte[] exportObjectToImageFormat(
      Object representation, ImageFormat format, int width, int height, String extraFormatting)
      throws ChemistryViewConversionException;

  /** And many more methods */
}
```

ChemistryView Components

The most important bits that make it work

- **Chemaxon Marvin** – the basis of *three* plugins
  - display plugin
  - structure editor plugin
  - toolkit plugin



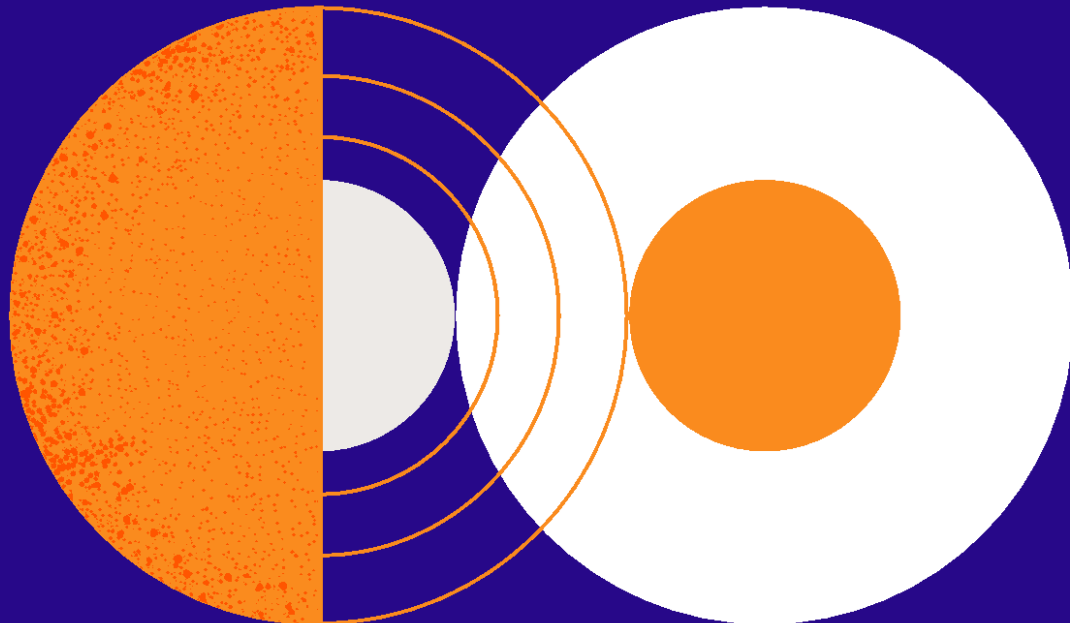
- CDL – an in-house developed structure *editor plugin*
- ChemDraw – Windows structure *editor plugin*
- OpenEye OEChem – general *toolkit plugin*

UGM Boston Oct. 4th 23

Chemaxon

ChemistryView Benefits

It is a game changer



Software Reuse: New Features

- Features added become available to all applications that use the library
  - New viewer or editor types
  - New features (e.g. drag and drop)
  - 64-bit Java Support

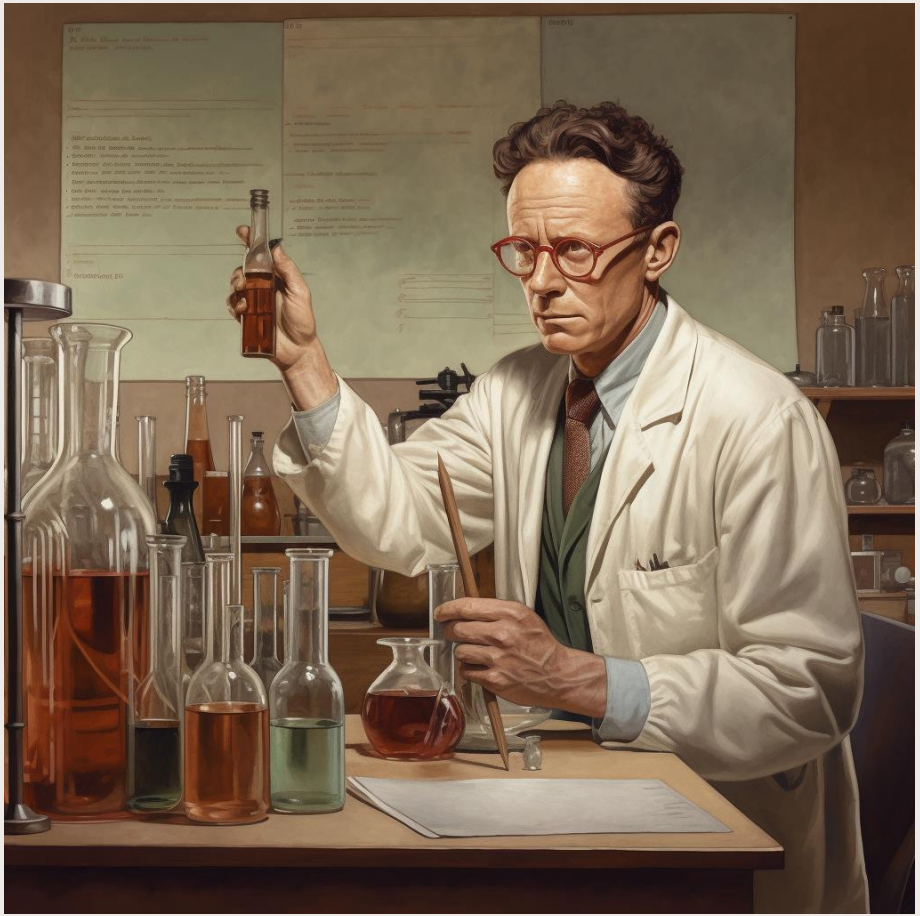
Software Reuse: Insulation

The layer of abstraction between our apps and commercial libraries means we can manage in one place changes to:

- APIs
- Licenses/availability
- New vendors

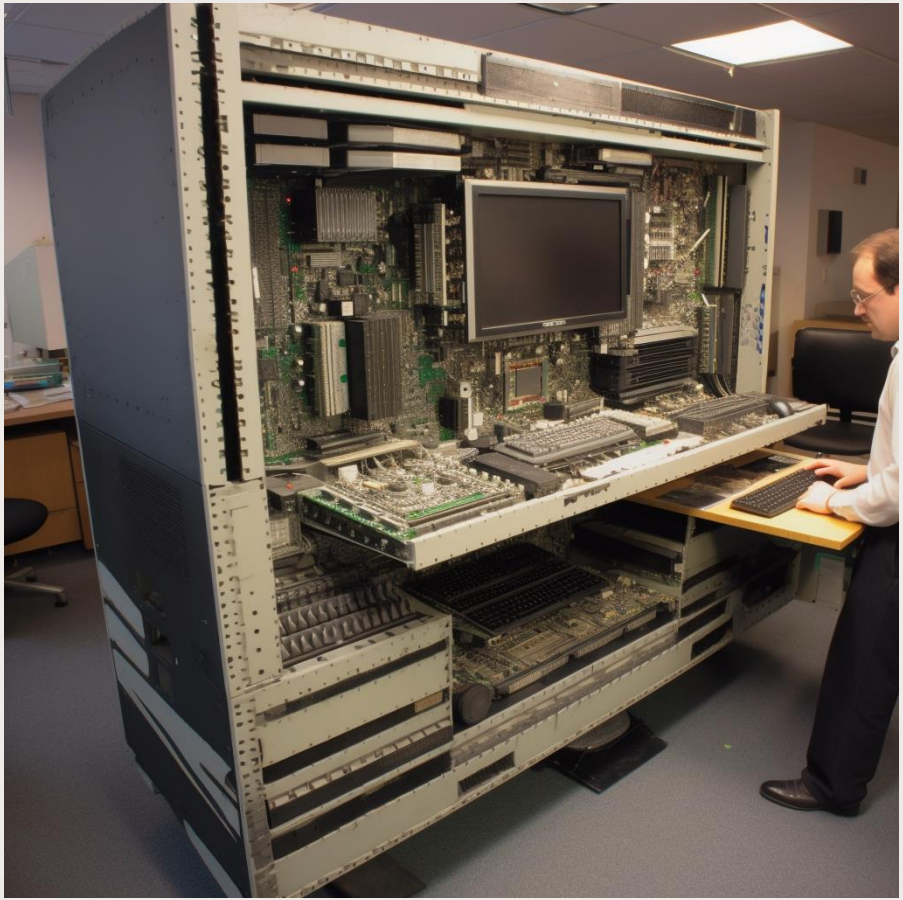
Simplification

Not every developer needs to be an expert at everything that goes into ChemistryView – they can just use it.



Huge Benefits

ChemistryView has dramatically simplified the problem of providing chemistry view and editing in a complex environment of cross platform chemistry applications.



Acknowledgements

ChemistryView was written by Robert Owen, Ph.D.

UGM Boston Oct. 4th 23

Chemaxon

Thank you

David H. Klatte, Ph.D.

david.h.klatte@pfizer.com