

# Textual AND graphical modeling: Xtext editors within Sirius Diagrams

Niko Stotz, [niko.stotz@altran.com](mailto:niko.stotz@altran.com)

EclipseCon Europe, 2018-10-24

alTRAN

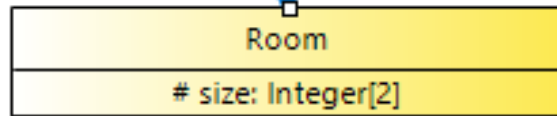
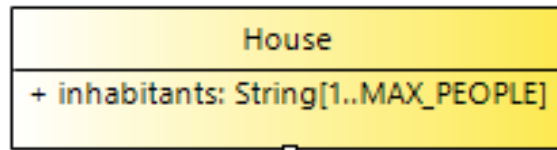
# Goals

# Xtext within Sirius Diagram

MAX_ROOMS
23

MAX_PEOPLE
42

MORE_CONSTANT
MAX_ROOMS



rooms [1..MAX\_ROOMS]

- MAX\_PEOPLE
- MAX\_ROOMS
- ]

Constant **MAX\_ROOMS**

Press 'F2' for focus

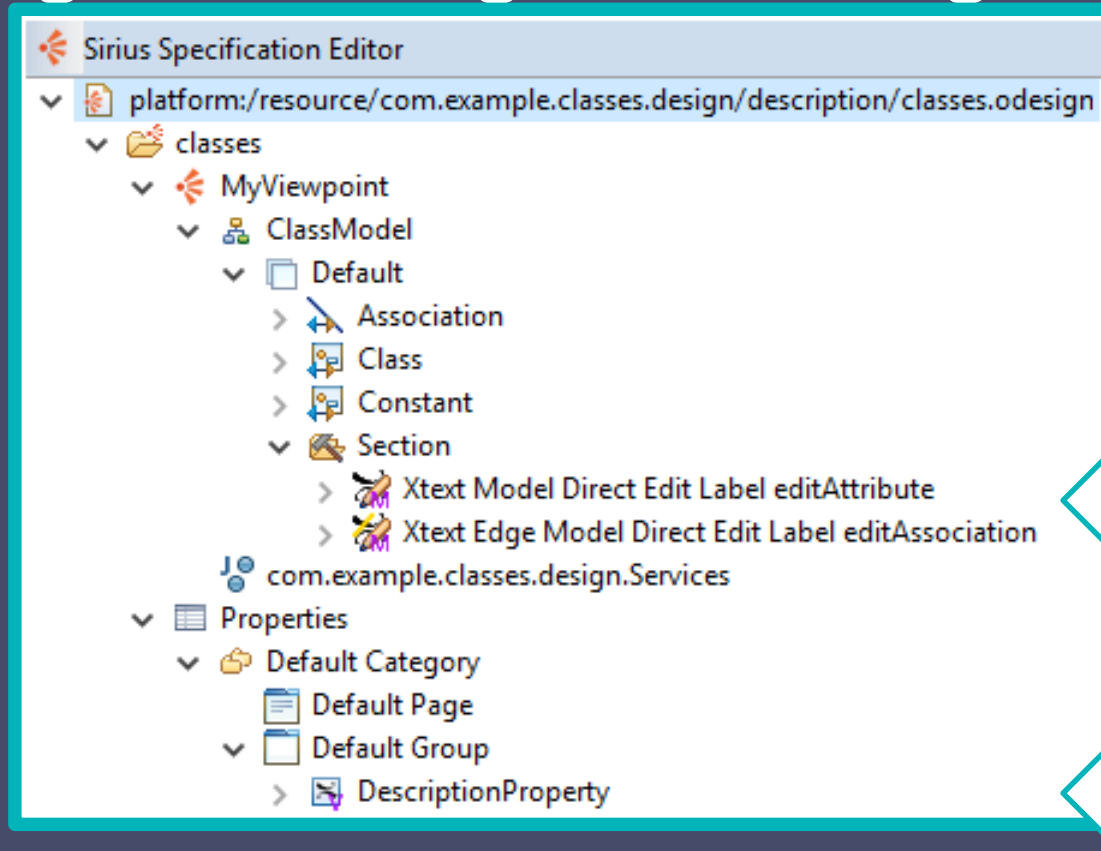
# Xtext within Properties View

The screenshot shows the Properties View of an IDE. On the left, there is a sidebar with the following categories: Page, Semantic, Style, and Appearance. The 'Page' category is selected, and the 'Description' property is expanded. The description text is displayed in a text area and contains two paragraphs of Xtext code:

```
<p>A <i>Room</i> is <b>not</i> available!</p>  
<p>Please try at another place.
```

The first paragraph contains an Xtext error: a red squiggly line under the closing tag of the bold tag (`</i>`) indicates a mismatch. A yellow tooltip box is positioned over this error, containing the text: `mismatched input '</i>' expecting '</b>'`. Below the tooltip, it says "Press 'F2' for focus".

# Configured in regular Odesign Model



# Without Changes to Xtext

```
grammar com.example.Classes with org.eclipse.xtext.common.Terminals

generate classes "http://www.example.com/Classes"

ClassModel:
  content+=Content*
;

Content:
  Constant
  | Class
  | Association
;

Constant:
  Description?
  '...'
;
```

```
module com.example.GenerateClasses

import org.eclipse.xtext.xtext.generator.*
import org.eclipse.xtext.xtext.generator.model.project.*

var rootPath = ".."

Workflow {

  component = XtextGenerator {
    configuration = {
      project = StandardProjectConfig {
        baseName = "com.example.classes"
        rootPath = rootPath
        eclipsePlugin = {
          enabled = true
        }
      }
    }
  }
}
```

```
package com.example

class ClassesRuntimeModule extends AbstractClassesRuntimeModule {
}
```

# Demo

# Config



# Xtext Language Injector

```
public class ClassesLanguageInjector implements
IXtextLanguageInjector {
    @Override
    public Injector getInjector() {
        return ClassesActivator.getInstance()
            .getInjector(
                ClassesActivator.COM_EXAMPLE_CLASSES
            );
    }
}
```

# Language Injector Extension Point

```
<plugin>
<extension point=
"com.altran.general.integration.xtextsirius.runtime.xtextLanguageInjector">
  <injector
    id=
    "com.example.classes.design.ClassesLanguageInjectorId"
    class=
    "com.example.classes.design.ClassesLanguageInjector"/>
</extension>
</plugin>
```

# Odesign Tool / Widget

The image displays two screenshots of the Odesign Tool interface, showing menu paths for creating new elements.


**Top Screenshot:** The 'Section Tools' menu is open, and the 'New' option is selected. The resulting dropdown menu lists the following options:


- Direct Edit Label
- Xtext Edge Model Direct Edit Label
- Xtext Edge Value Direct Edit Label
- Xtext Model Direct Edit Label (highlighted)
- Xtext Value Direct Edit Label

**Bottom Screenshot:** The 'Default Group' menu is open, and the 'New Widget' option is selected. The resulting dropdown menu lists the following options:

- Text
- Xtext Model Text (highlighted)
- Xtext Model Text Area
- Xtext Value Text
- Xtext Value Text Area

# Odesign Details

Properties 

 **Xtext Edge Model Direct Edit Label editAssociation**

**General**

**Id\*:**  **Label:**

**Documentation**

**Injector Id:**

**Selected Features:**  ...

**Editable Features:**  ...

**Lines:**  single-line  multi-line

# Features

# Example Model as Text

```
constant MAX_ROOMS = 23
constant MAX_PEOPLE = 42

class House {
  + inhabitants:
  string[1..MAX_PEOPLE]
}

[[[
<p>rooms are important</p>
]]]
association rooms House --> Room
[1..MAX_ROOMS]
```

```
[[[
<p>A <i>Room</i> is <b>not</b>
available!</p>
<p>Please try at another place.</p>
]]]
class Room {
  [[[
  <p>size doesn't really matter</p>
  ]]]
  # size: integer[2..2]
}
```

# Example Model as Tree

The screenshot displays a software development tool interface. At the top, a tab labeled 'test.classes' is open. Below it, a tree view shows the following structure:

- platform:/resource/classes/test.classes
  - Class Model
    - Constant MAX\_ROOMS
    - Constant MAX\_PEOPLE
    - Constant MORE\_CONSTANT
    - Class House
      - Attribute inhabitants
        - String Type
        - Integer Literal 1
        - Constant Ref
      - Class Room
        - Attribute size
        - Association rooms

Below the tree view, a 'Properties' panel is visible, showing the following table:

Property	Value
Description	[[[ <p>A <i>Room</i> is <b>not</b> available!</p> <p>Please try at another place.</p> ]]]
Name	Room

# Xtext Editing Model String Property

The screenshot shows the Xtext editor interface. On the left, there is a sidebar with tabs for 'Page', 'Semantic', 'Style', and 'Appearance'. The main editor area is titled 'Description' and contains the following HTML-like text:

```
<p>A <i>Room</i> is <b>not</i> available!</p>  
<p>Please try at another place.</p>
```

A validation error message is displayed in a yellow box over the text: 'mismatched input '</i>' expecting '</b>'. Below the message, it says 'Press 'F2' for focus'. The error points to the closing tag of the italicized word 'Room'.

The screenshot shows the 'Properties' view in Xtext. The tree view on the left shows a 'Class Room' with sub-properties 'Attribute size' and 'Association rooms'. The 'Properties' table below is highlighted with an orange border and contains the following data:

Property	Value
Description	[[[<p>A <i>Room</i> is <b>not</i> available!</p> <p>Please try at another place.</p> ]]]
Name	Room



# Text from Model Property

**<p>A *Room* is **not** available!</p>**  
**<p>Please try at another place.</p>**

# Document Completed from Config

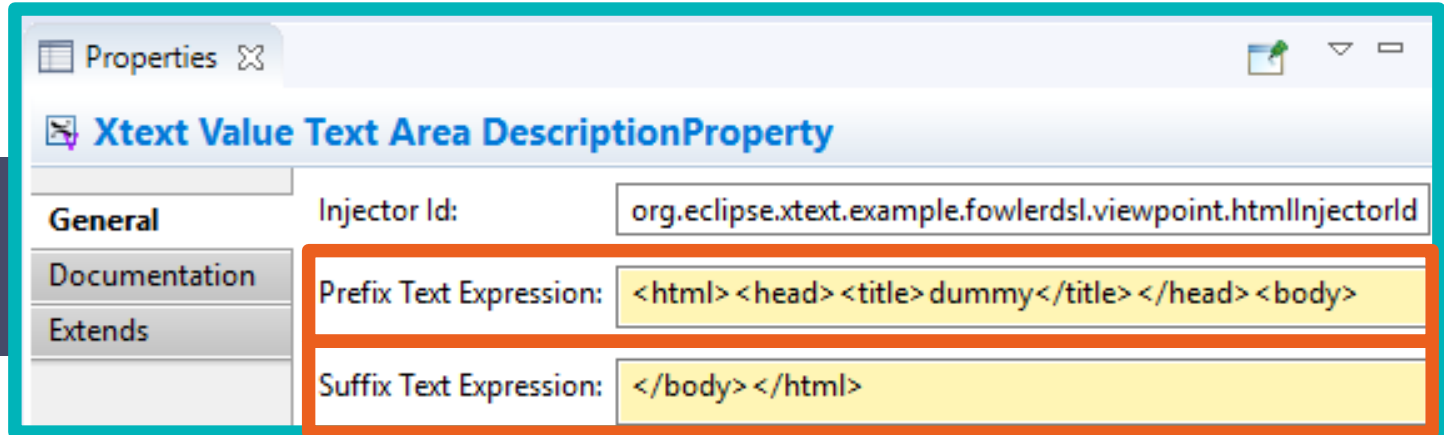
**<html>**

**<head>**

**<title>dummy</title>**

**</head>**

**<body>**



The screenshot shows the Eclipse IDE's Properties window for the `Xtext Value Text Area DescriptionProperty`. The window has a title bar with a close button and a maximize button. Below the title bar, there is a tab labeled "Properties" and a small icon. The main content area is divided into sections: "General", "Documentation", and "Extends". The "General" section is selected and contains the following properties:

- Injector Id: `org.eclipse.xtext.example.fowlerdsl.viewpoint.htmlInjectorId`
- Prefix Text Expression: `<html> <head> <title> dummy</title> </head> <body>`
- Suffix Text Expression: `</body> </html>`

The Prefix and Suffix Text Expression fields are highlighted with a yellow background and an orange border.

**</body>**

**</html>**

# Show Only Relevant Part to User

```
<html>
```

```
<head>
```

```
<title>dummy</title>
```

```
</head>
```

```
<body>
```

```
<p>A Room is not available!</p>
```

```
<p>Please try at another place.</p>
```

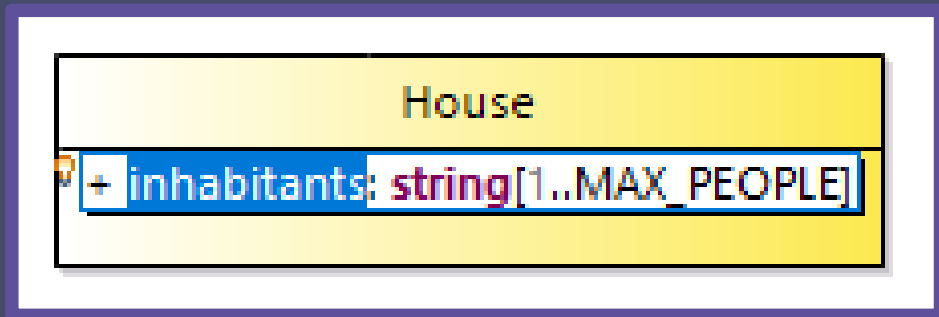
```
</body>
```

```
</html>
```

# Xtext Editing Model Itself

The screenshot shows an IDE window titled 'test.classes'. The left sidebar displays a tree view of the project structure. Under 'platform:/resource/classes/test.classes', the 'Class Model' is expanded to show 'Class House'. The 'Attribute inhabitants' is selected and highlighted with an orange box. The 'Properties' panel at the bottom shows the details for the selected attribute:

Property	Value
Description	
Name	inhabitants
Visibility	+



# Pre-selecting Feature Text

test.classes

- platform:/resource/classes/test.classes
  - Class Model
    - Constant MAX\_ROOMS
    - Constant MAX\_PEOPLE
    - Constant MORE\_CONSTANT
    - Class House
      - Attribute inhabitants
        - String Type
        - Integer Literal 1
        - Constant Ref
  - Class Room
    - Attribute size
    - Association rooms

Properties

Property	Value
Description	
Name	inhabitants
Visibility	+

```
House  
- inhabitants: string[1..MAX_PEOPLE]
```

Section

- Xtext Model Direct Edit Label editAttribute

Properties

Xtext Model Direct Edit Label editAttribute

General

Id\*: editAttribute

Documentation

Injector Id: com.example.classes.design.ClassesLanguageInjectorId

Selected Features: Attribute.name

# Editing Only Subset of Features

▼ Attribute size

- Integer Type
- Integer Literal 2
- Integer Literal 3

Properties ✕

Property	Value
Description	<p>size doesn't really matter</p>
Name	size
Visibility	#

Room

# size: integer[2..3]

Xtext Model Direct Edit Label editAttribute

Properties ✕

**Xtext Model Direct Edit Label editAttribute**

General	Id*:	<input type="text" value="editAttribute"/>
Documentation	Injector Id:	<input type="text" value="com.example.classes.design.ClassesLanguageInjectorId"/>
	Editable Features:	<input type="text" value="Attribute.name, Attribute.visibility, Attribute.lowerBound, Attribute.upperBound, Attribute.type"/>

# Editing Only Subset of Features

Attribute size

- Integer Type
- Integer Literal 2
- Integer Literal 3

Property	Value
Description	<p>size doesn't really matter</p>
Name	size
Visibility	#

Room

# size: integer[2..3]

Xtext Model Direct Edit Label editAttribute

Properties

Xtext Model Direct Edit Label editAttribute

General

Id\*: editAttribute

Documentation

Injector Id: com.example.classes.d...n.ClassesLanguageInjectorId

Editable Features: Attribute.name, Attribute.visibility, Attribute.lowerBound, Attribute.upperBound, Attribute.type

# Editing Only Subset of Features

The screenshot shows a tree view on the left with 'Attribute size' expanded, containing 'Integer Type', 'Integer Literal 2', and 'Integer Literal 3'. Below it is a 'Properties' table with the following content:

Property	Value
Description	<p>size doesn't really matter</p>
Name	size
Visibility	#

Red boxes highlight the 'Name' and 'Visibility' rows, with a '2' next to 'Name' and a '1' next to 'Visibility'.

The screenshot shows a code editor with a yellow background. The text '# size: integer[2..3]' is highlighted with a blue selection box. An orange box highlights the '#' character, with a '1' below it. Another orange box highlights the 'size' text, with a '2' below it.

The screenshot shows a configuration dialog for 'Xtext Model Direct Edit Label editAttribute'. The 'Id\*' field contains 'editAttribute'. The 'Injector Id' field contains 'com.example.classes.d...n.ClassesLanguageInjectorId'. The 'Editable Features' field contains 'Attribute.name', 'Attribute.visibility', 'Attribute.lowerBound', 'Attribute.upperBound', and 'Attribute.type'. Red boxes highlight 'Attribute.name' and 'Attribute.visibility', with a '2' next to 'Attribute.name' and a '1' next to 'Attribute.visibility'.



# Editing Only Subset of Features

The screenshot shows the Xtext Model Direct Edit interface. At the top, a tree view displays the 'Attribute size' node, which is expanded to show three sub-nodes: 'Integer Type', 'Integer Literal 2', and 'Integer Literal 3'. The 'Integer Type' node is highlighted with an orange box and a red circle containing the number 3. Below the tree view is a 'Properties' table with the following content:

Property	Value
Description	<p>size doesn't really matter</p>
Name	size
Visibility	#

The 'Name' and 'Visibility' rows are highlighted with orange boxes and red circles containing the numbers 2 and 1, respectively.

The screenshot shows the Xtext Model Direct Edit interface for the 'Room' class. The class name 'Room' is displayed in a yellow box. Below it, the attribute '# size integer 2..3' is highlighted with an orange box and a red circle containing the number 3. The attribute name 'size' is highlighted with an orange box and a red circle containing the number 2, and the attribute value 'integer 2..3' is highlighted with an orange box and a red circle containing the number 1.

The screenshot shows the Xtext Model Direct Edit interface for the 'Xtext Model Direct Edit Label editAttribute' class. The class name is displayed in a blue box. Below it, the 'Properties' table is shown with the following content:

Property	Value
Id*	editAttribute
Injector Id:	com.ample.classes.d...n.ClassesLanguageInjectorId
Editable Features:	Attribute.name, Attribute.visibility, Attribute.lowerBound, Attribute.upperBound, Attribute.type

The 'Attribute.name' and 'Attribute.visibility' features are highlighted with orange boxes and red circles containing the numbers 2 and 1, respectively. The 'Attribute.type' feature is highlighted with an orange box and a red circle containing the number 3.

# Editing Only Subset of Features

This screenshot shows two parts of an IDE interface. At the top, a tree view under 'Attribute size' has three items: 'Integer Type' (circled 3), 'Integer Literal 2' (circled 4), and 'Integer Literal 3'. Below this is a 'Properties' table with the following content:

Property	Value
Description	<p>size doesn't really matter</p>
Name	size (circled 2)
Visibility	# (circled 1)

This screenshot shows a code editor with a yellow background. A class named 'Room' is visible. Below the class name, an attribute is defined as '# size integer 2,3'. The attribute name 'size' is circled 1, the visibility '#' is circled 2, the type 'integer' is circled 3, and the values '2,3' are circled 4.

This screenshot shows the 'Xtext Model Direct Edit Label editAttribute' dialog box. It has a 'Properties' section and a 'General' section. The 'General' section contains the following fields:

- Id\*:** editAttribute
- Injector Id:** com.ample.classes.d. n.ClassesLanguage.ctorId (circled 4), with 'com' circled 2 and 'n.ClassesLanguage' circled 1.
- Editable Features:** Attribute.name, Attribute.visibility, Attribute.lowerBound, Attribute.upperBound, Attribute.type (circled 3).

# Editing Only Subset of Features

Attribute size

- Integer Type (3)
- Integer Literal 2 (4)
- Integer Literal 3 (5)

Properties

Property	Value
Description	<p>size doesn't really matter</p>
Name (2)	size
Visibility (1)	#

Room

# size: integer 2,3

1 2 3 4 5

Xtext Model Direct Edit Label editAttribute

Properties

Xtext Model Direct Edit Label editAttribute

General

Id\*: editAttribute

Documentation

Injector Id: com.ample.classes.d...n.ClassesLanguage...ctorId (2, 1, 4, 5, 3)

Editable Features: Attribute.name, Attribute.visibility, Attribute.lowerBound, Attribute.upperBound, Attribute.type

# Editing Only Subset of Features

Tree view structure:

- Attribute size
  - Integer Type
  - Integer Literal 2
  - Integer Literal 3

Properties table:

Property	Value
Description	<p>size doesn't really matter</p>
Name	size
Visibility	#

Room

# size: integer[2..3]

Xtext Model Direct Edit Label editAttribute

Properties

**Xtext Model Direct Edit Label editAttribute**

**General**

Id\*:

**Documentation**

Injector Id:

Editable Features:

# Different Grammar to Cut Middle Features

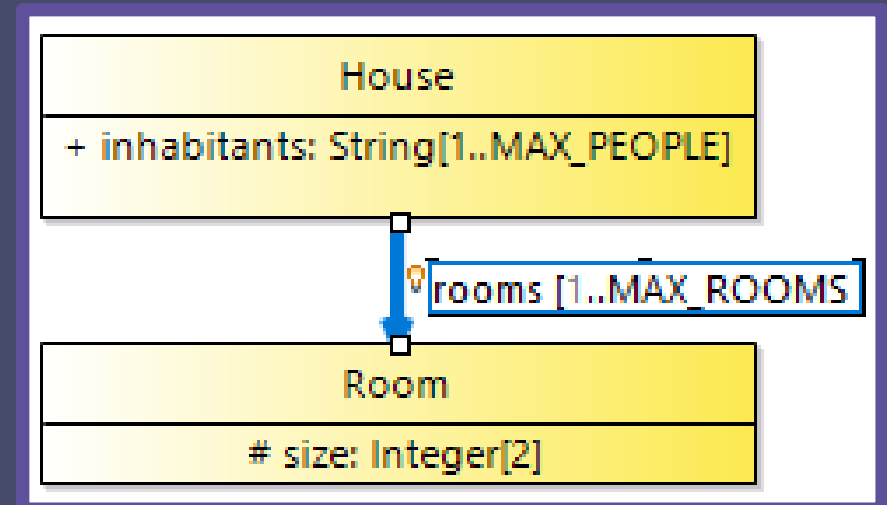
association rooms **House --> Room** [1..MAX\_ROOMS]

Association rooms

- Integer Literal 1
- Constant Ref

Properties

Property	Value
Description	<p>rooms are important</p>
Name	rooms
Source	Class House
Target	Class Room



# Gotchas

- Editing identifying property
- Working with erroneous models
- Heuristics to find terminals adjacent to features
- Size of Xtext edit box

# Extensive Documentation

<https://altran-mde.github.io/xtext-sirius-integration.io/userguide/index.html>



## Xtext / Sirius Integration User Guide

Niko Stotz – [niko.stotz@altran.com](mailto:niko.stotz@altran.com) – Version 0.22.1-SNAPSHOT

### Table of Contents

1. Terminology
2. Overview
3. Examples
  - 3.1. `.odesign` Contents
  - 3.2. `plugin.xml` Contents
  - 3.3. Properties Views
4. Reference
  - 4.1. Editor Placement
    - 4.1.1. Diagram / Direct Editor
    - 4.1.2. Property View / Property Editor
  - 4.2. Editor Contents
    - 4.2.1. Editing Models
    - 4.2.2. Editing Values
  - 4.3. Editor Infos [!]
    - 4.3.1. Injector
    - 4.3.2. Single-line vs. Multi-line
    - 4.3.3. PrefixText and SuffixText
    - 4.3.4. Editable Features
    - 4.3.5. Ignored Nested Features
    - 4.3.6. Pre-selected Features
    - 4.3.7. Position of Affected Edge Label
    - 4.3.8. PrefixTerminals and SuffixTerminals
  - 4.4. Additional Model Operations
  - 4.5. Handling Syntax Errors
5. Advanced Capabilities
  - 5.1. Using a Different Grammar for Editing Model Contents
  - 5.2. Force Serialization of Whitespaces
  - 5.3. Force Ignored Nested Features to be Transient
  - 5.4. Constrain the Global Scope to all Ecore Resources from the Sirius Session
6. Versions
7. Known Issues

# Source Code & Samples on GitHub

<https://github.com/altran-mde/xtext-sirius-integration>

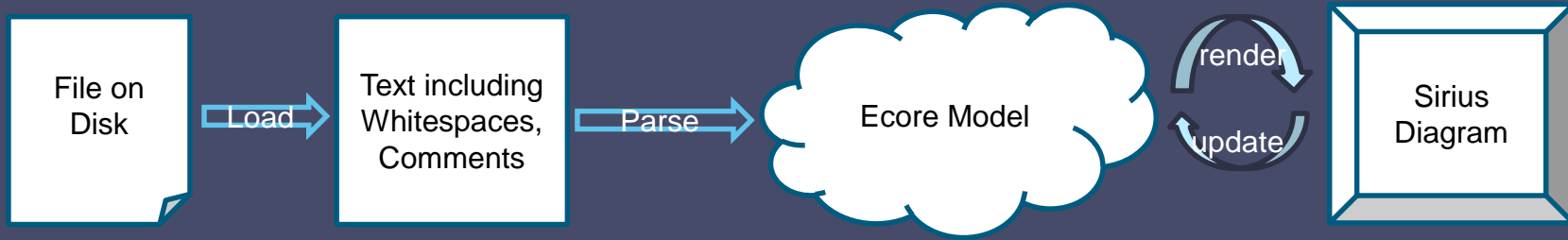




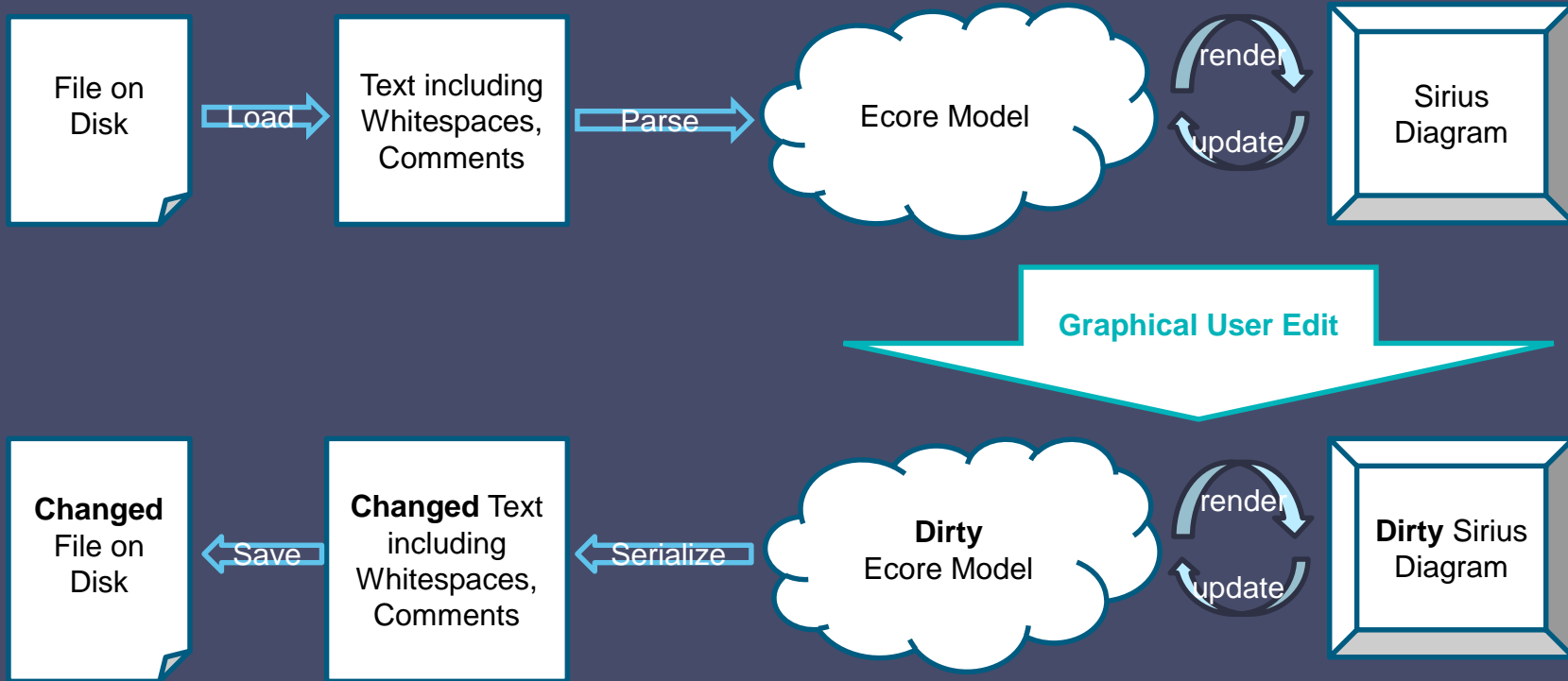
# Future: Contribute to Sirius?



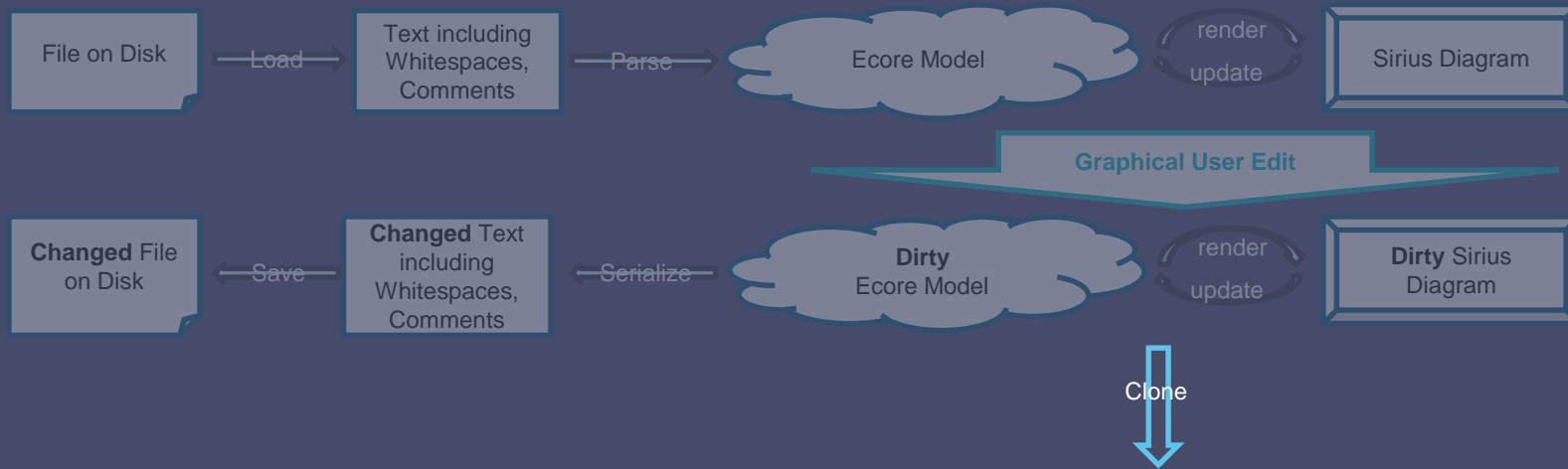
# Lifecycle of Ecore Model



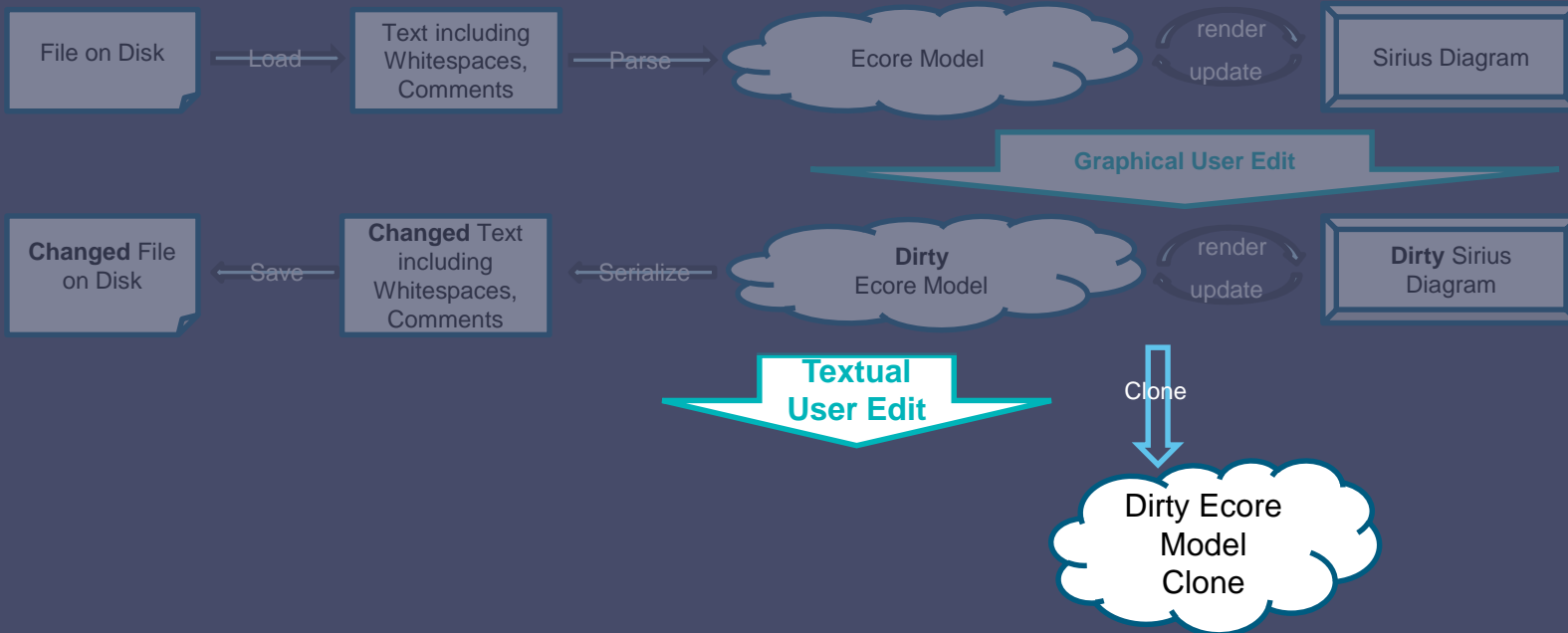
# Lifecycle of Ecore Model



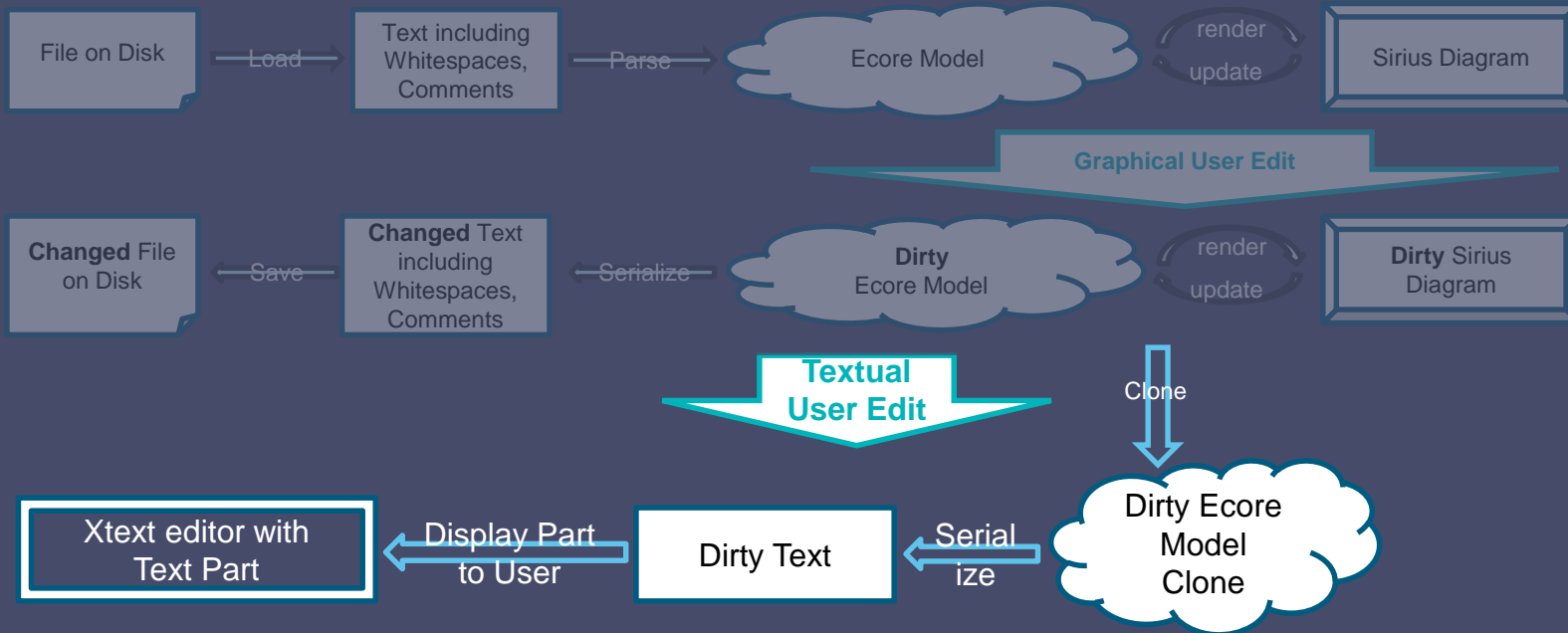
# Lifecycle of Textual User Edit



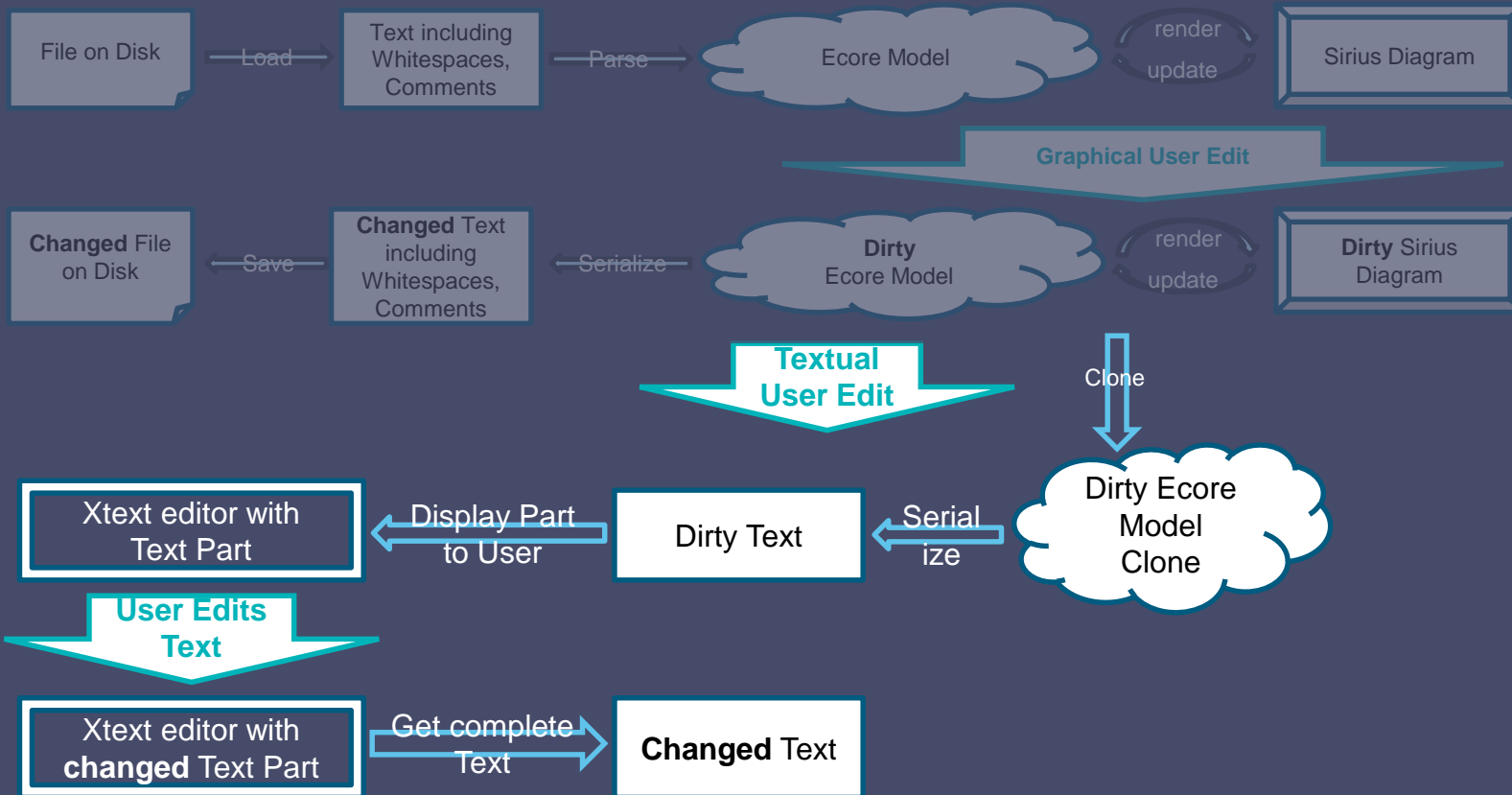
# Lifecycle of Textual User Edit



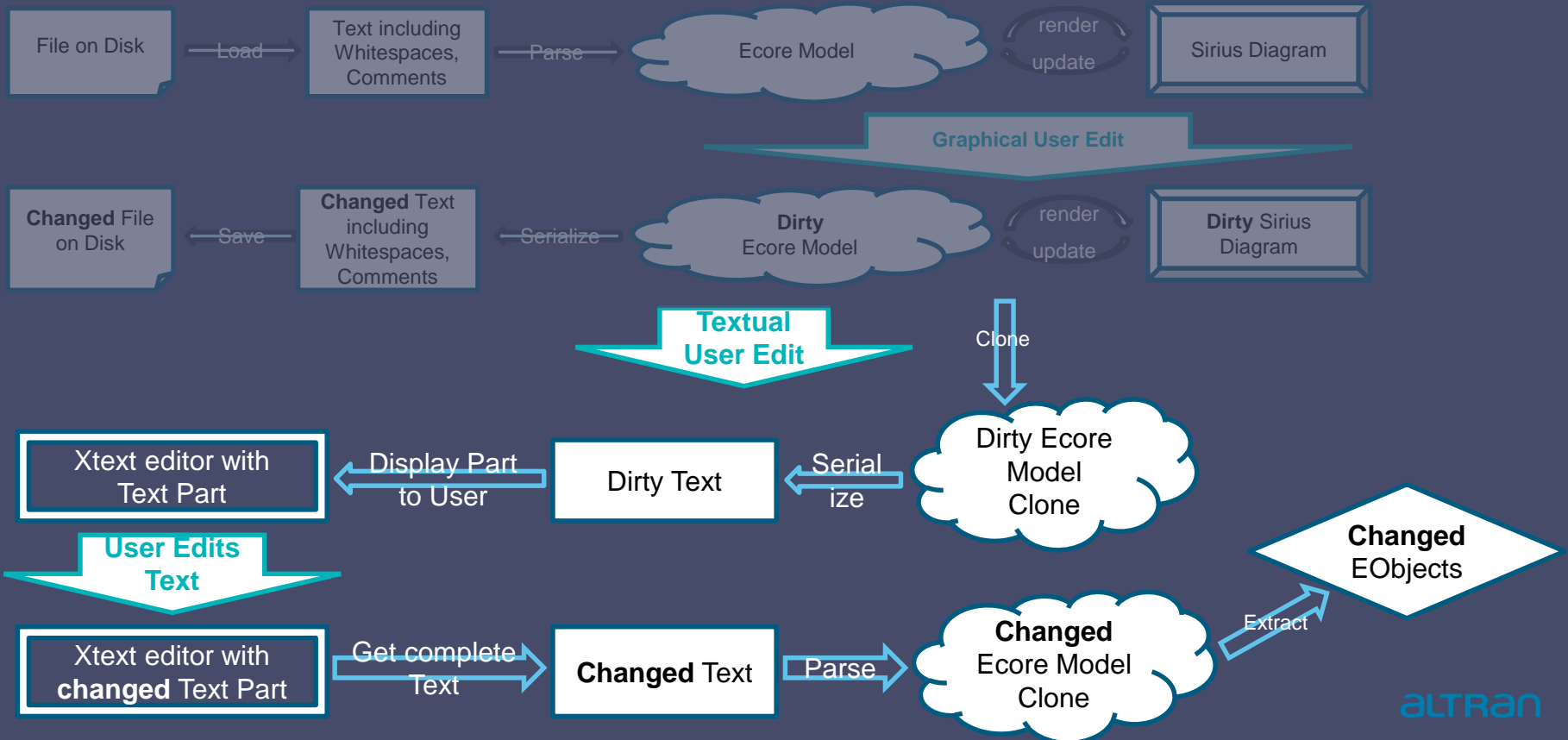
# Lifecycle of Textual User Edit



# Lifecycle of Textual User Edit

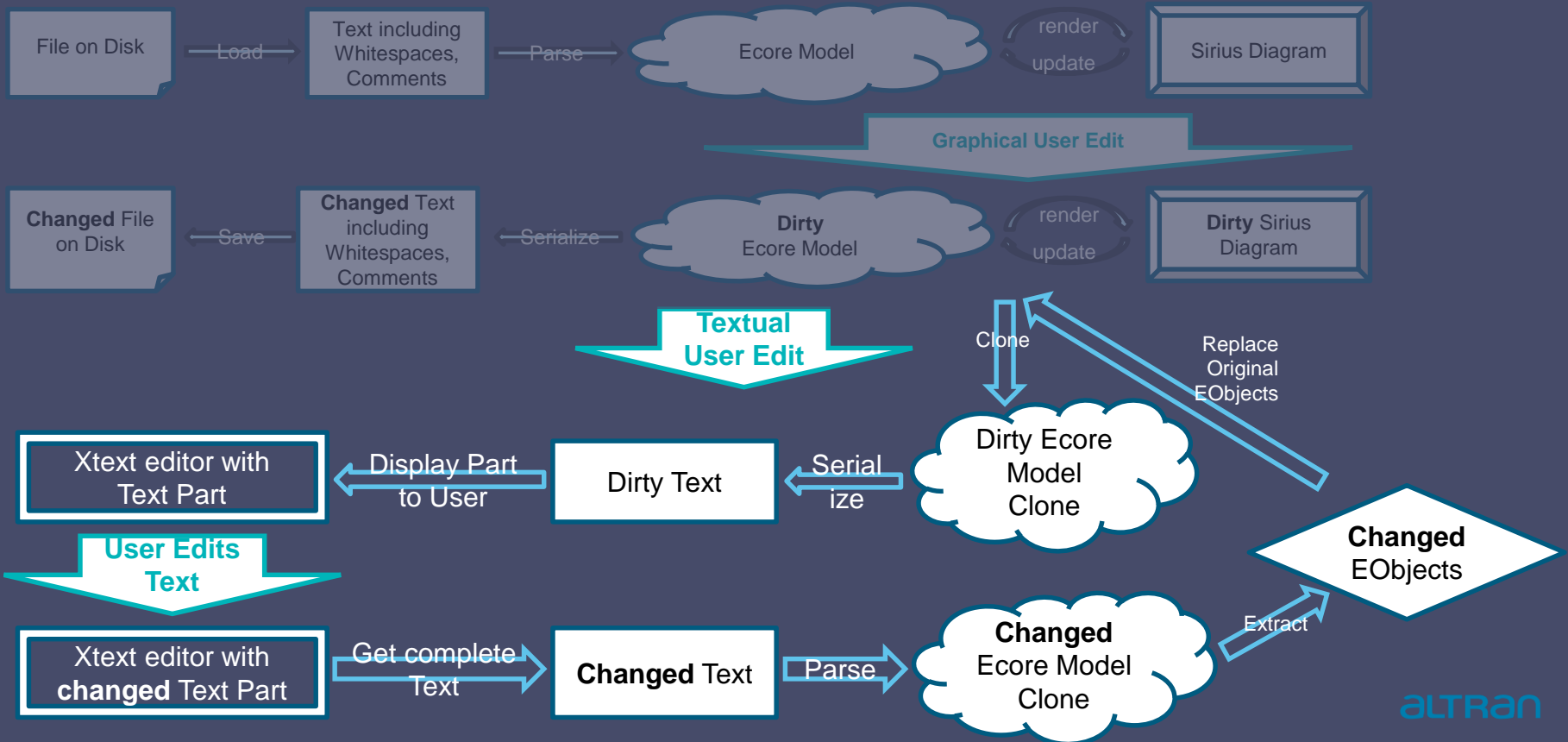


# Lifecycle of Textual User Edit





# Lifecycle of Textual User Edit



# References from Edited Model Part

## Original Model

### events

event1 [10 .. pi]

### constants

e 271

pi 314



# References from Edited Model Part

## Original Model

### events

event1 [10 .. pi]

### constants

e 271

pi 314

Textual  
User Edit

## Cloned Model

### events

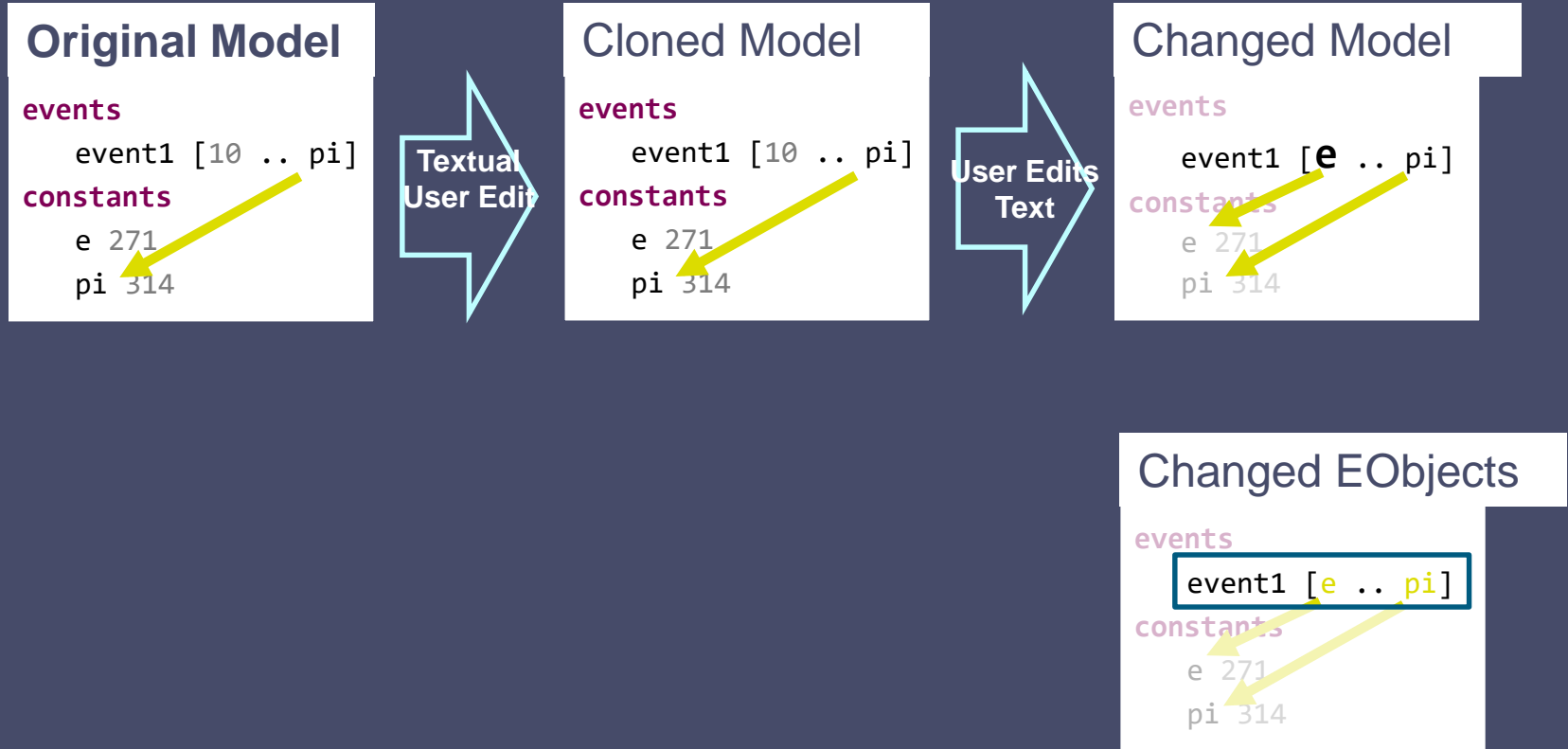
event1 [10 .. pi]

### constants

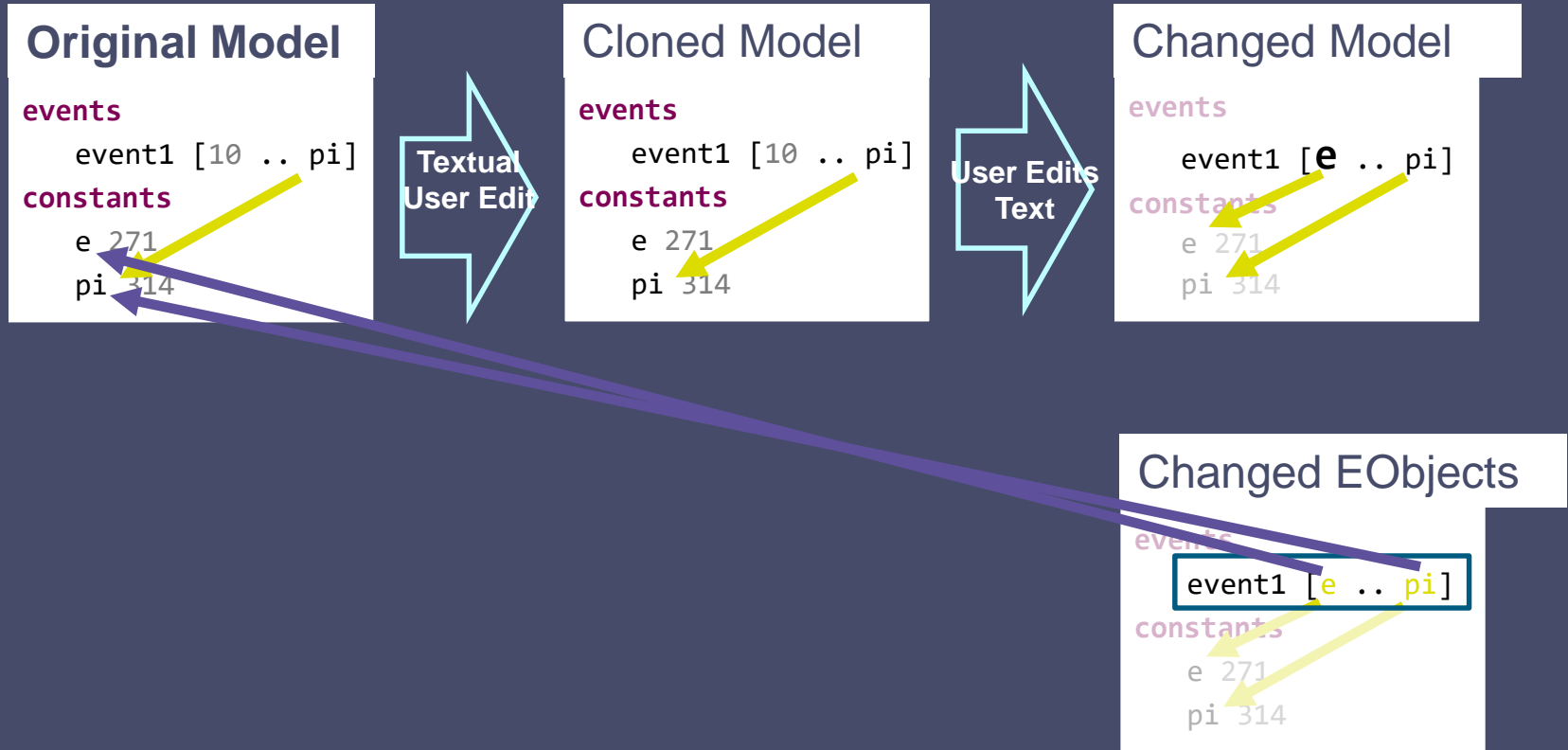
e 271

pi 314

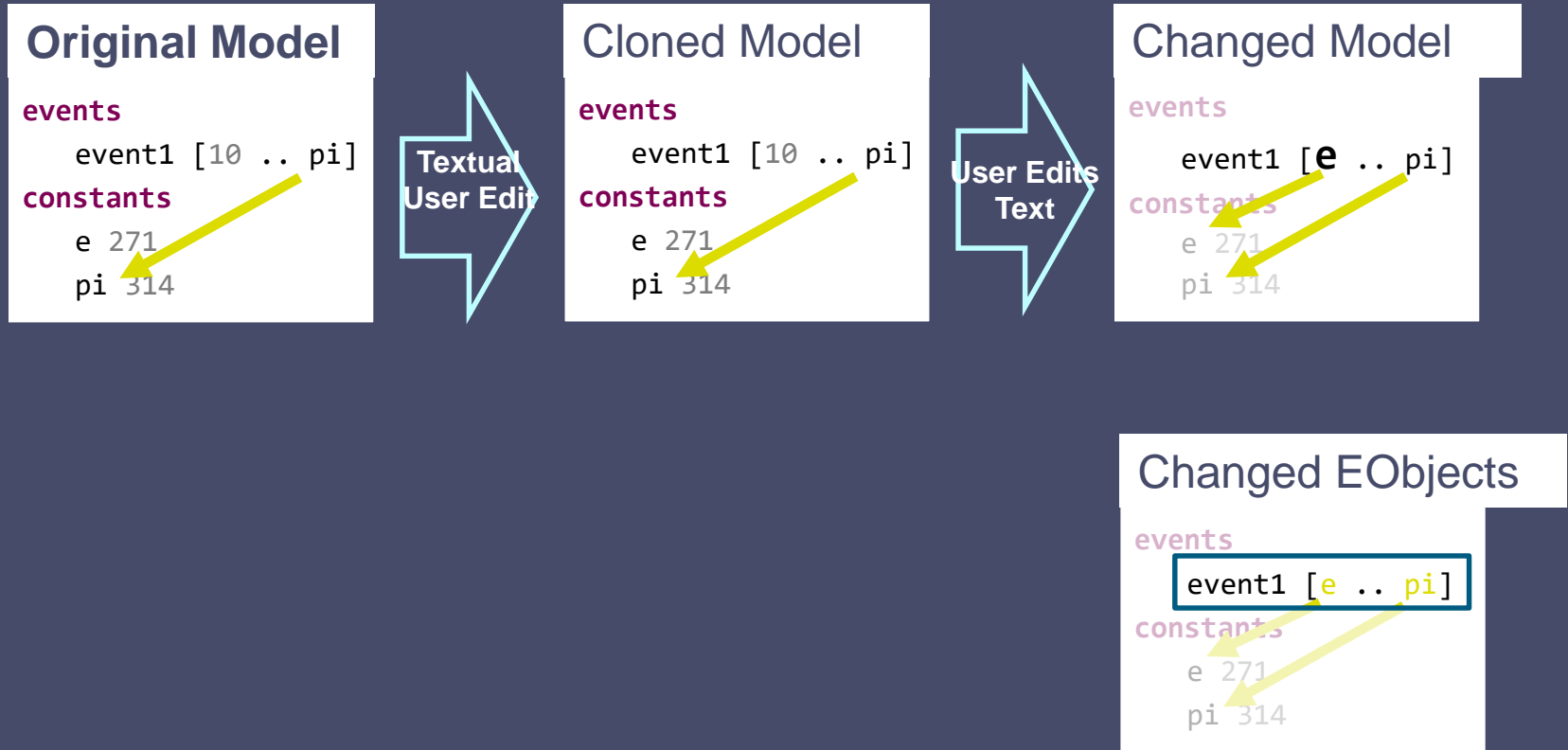
# References from Edited Model Part



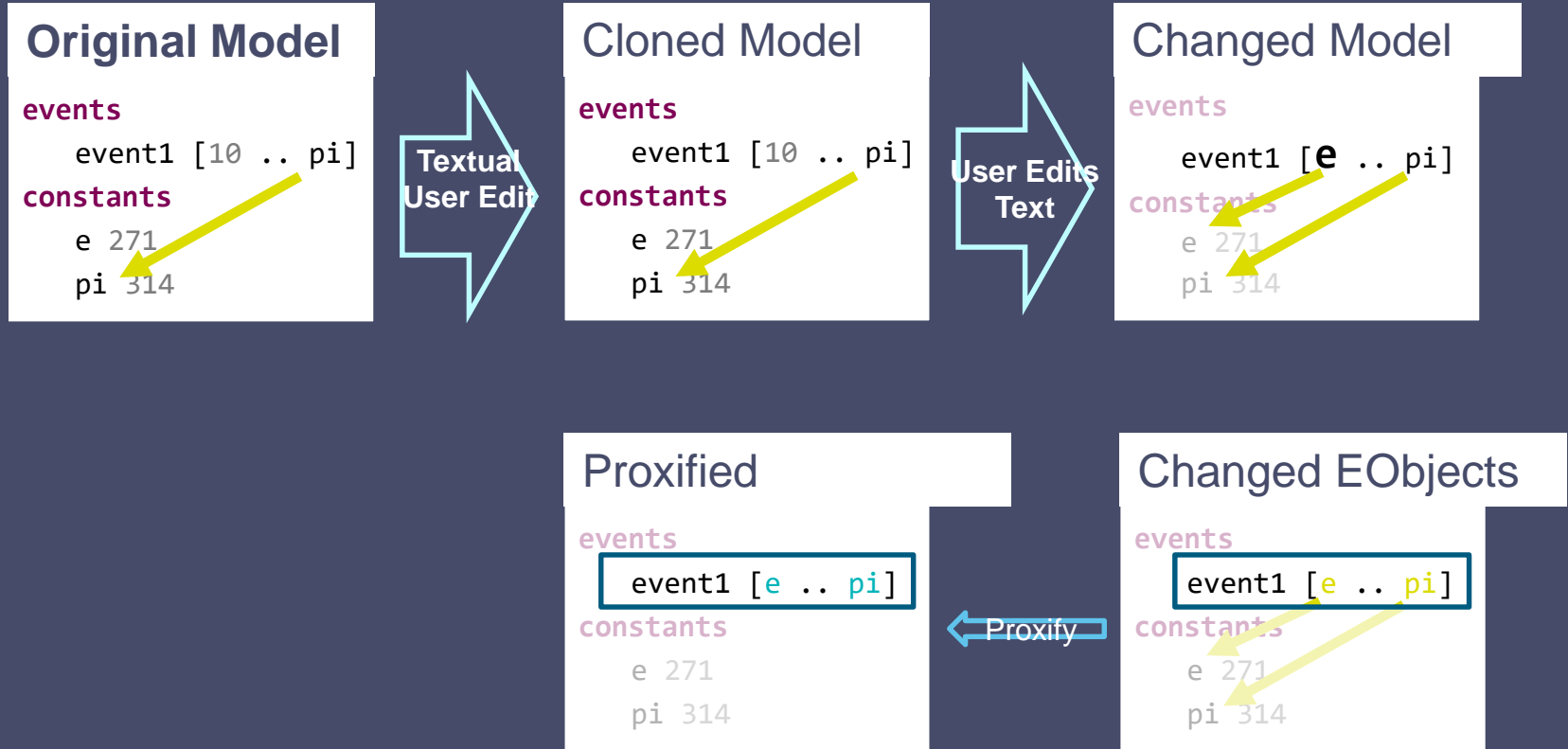
# References from Edited Model Part



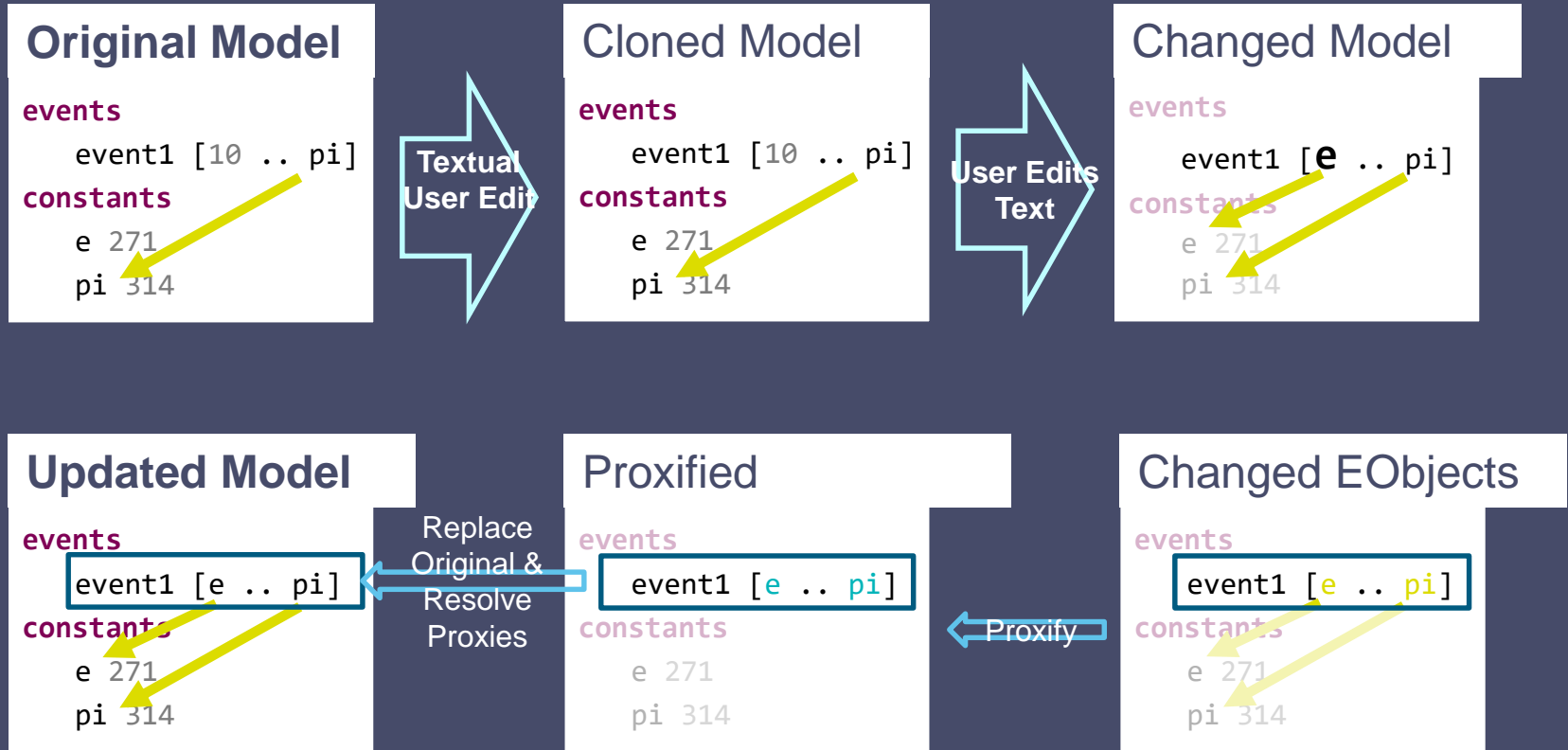
# References from Edited Model Part



# References from Edited Model Part

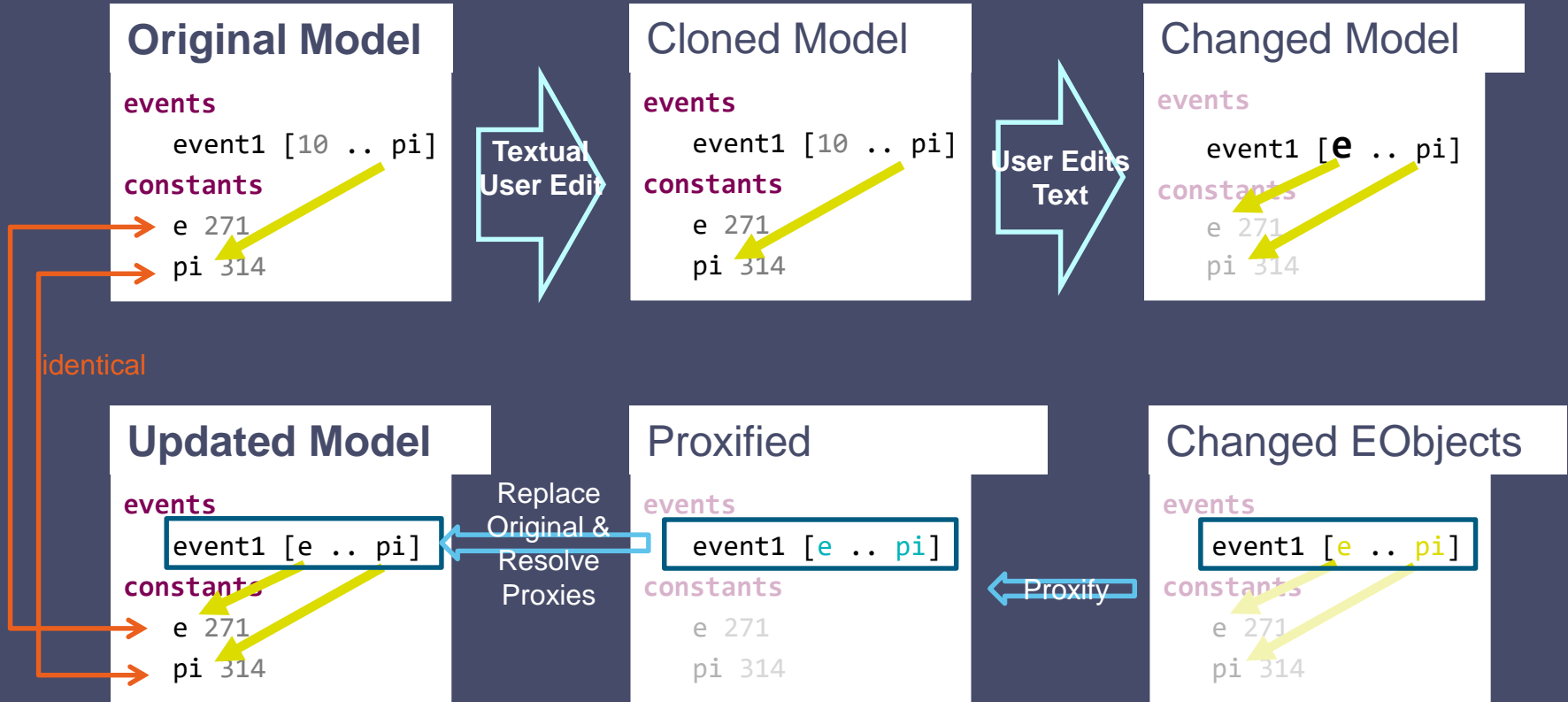


# References from Edited Model Part





# References from Edited Model Part



# Evaluate the Sessions

Sign in and vote at [eclipsecon.org](https://eclipsecon.org)

-1

0

+1