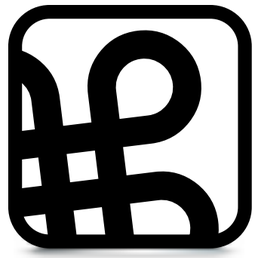


Macoun

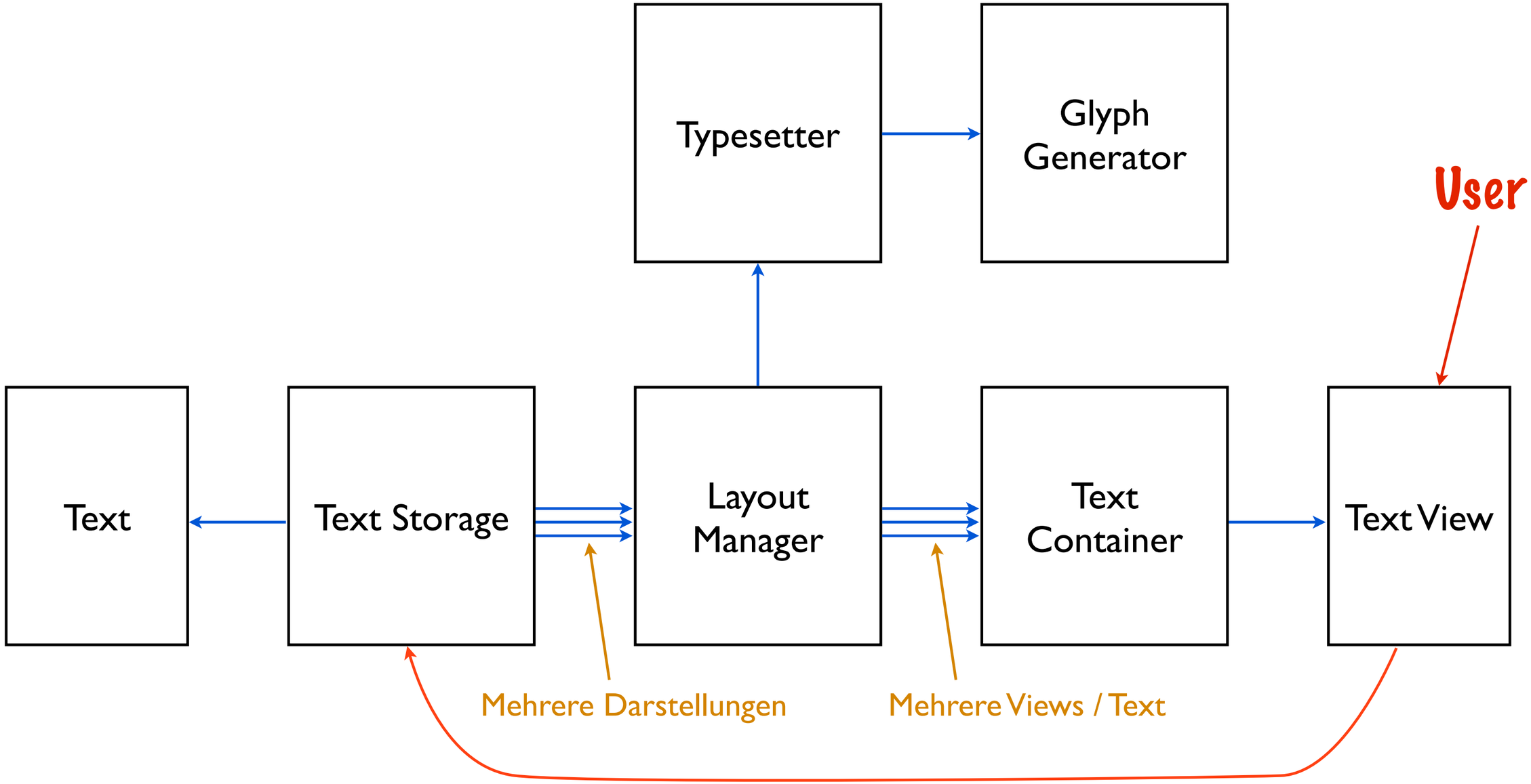
Wir bauen einen Texteditor

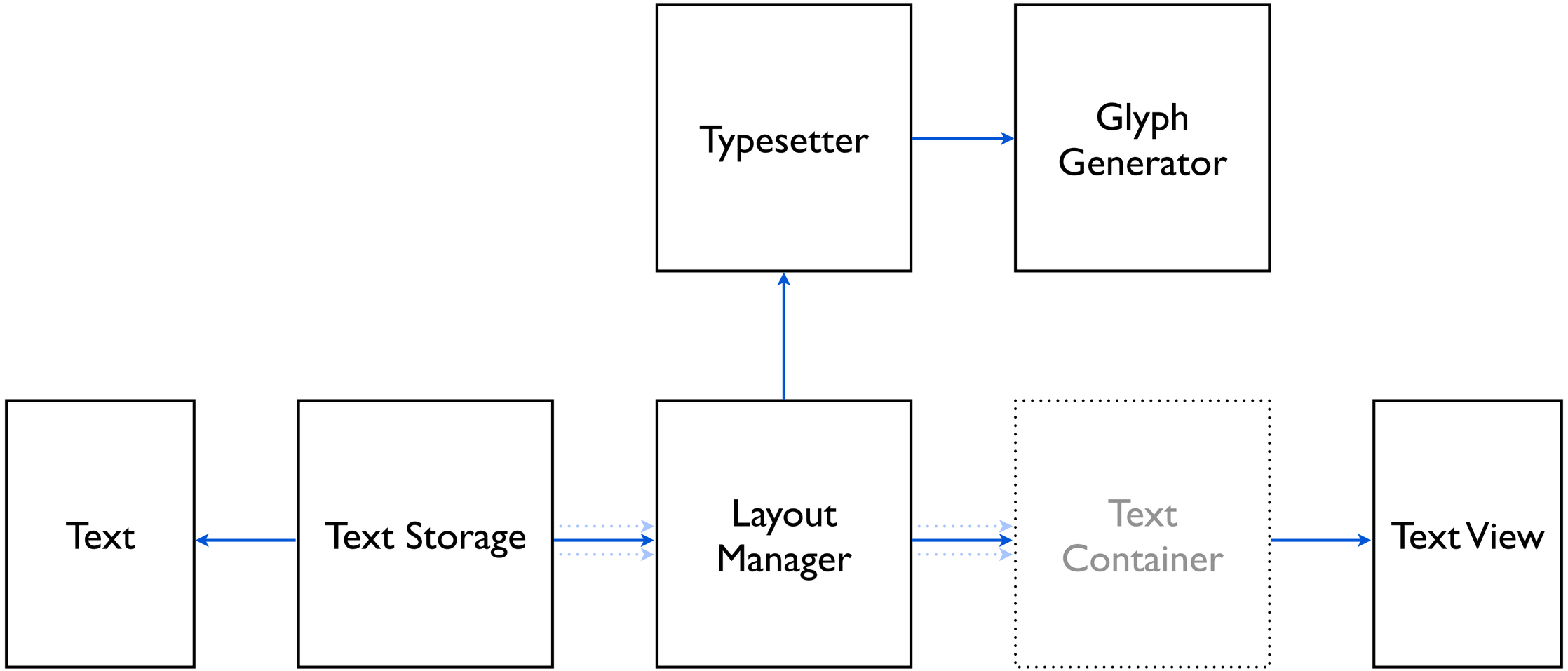
Max Seelemann

Das bin ich!

Demo

Architektur

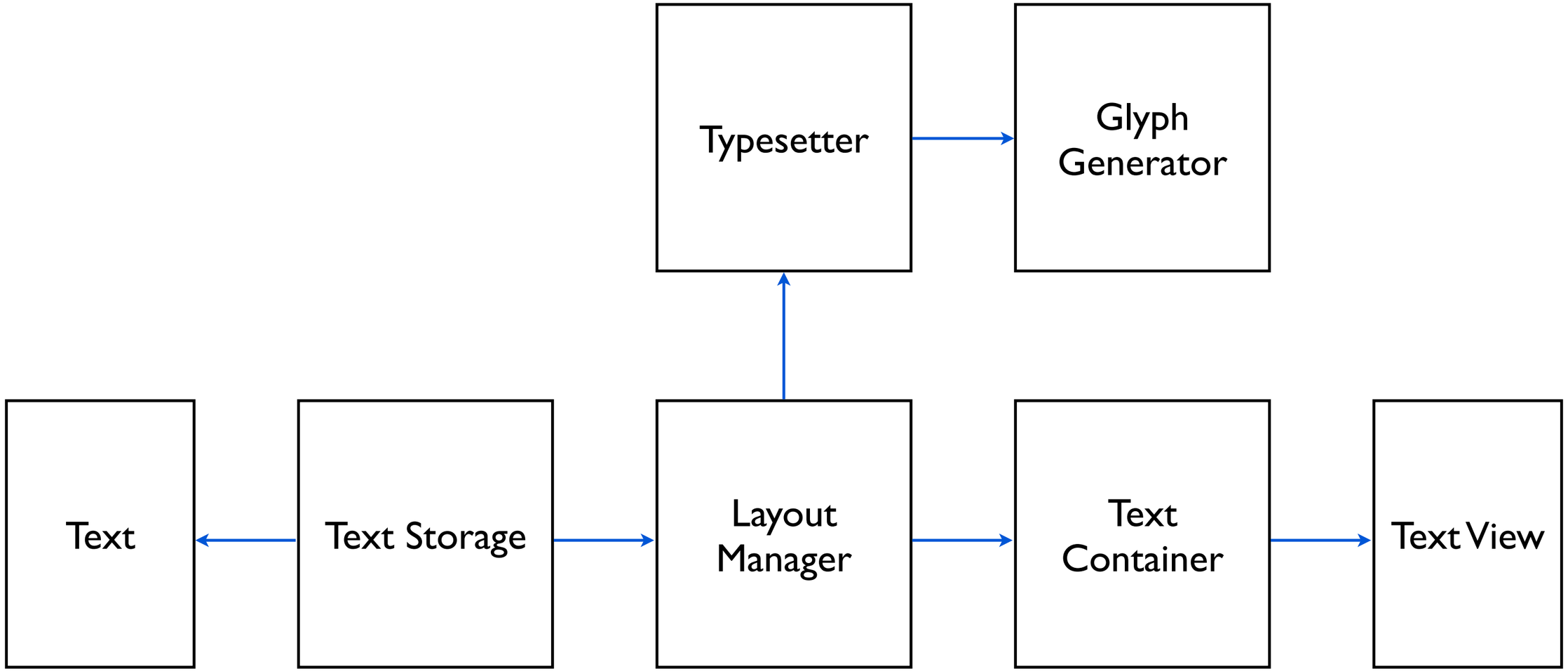

Typesetter
Glyph Generator
User
Text
Text Storage
Layout Manager
Text Container
Text View
Mehrere Darstellungen
Mehrere Views / Text

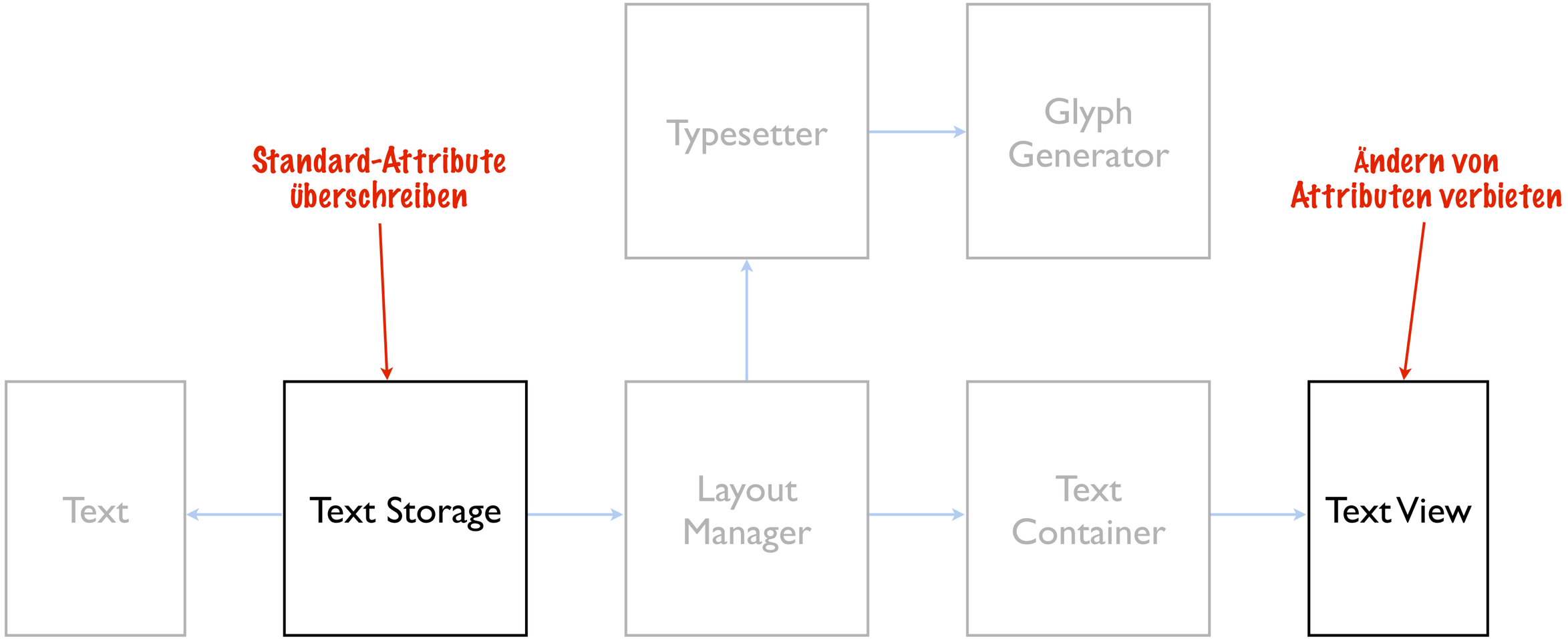

Typesetter
Glyph Generator
Text
Text Storage
Layout Manager
Text Container
Text View

Aufgabe:

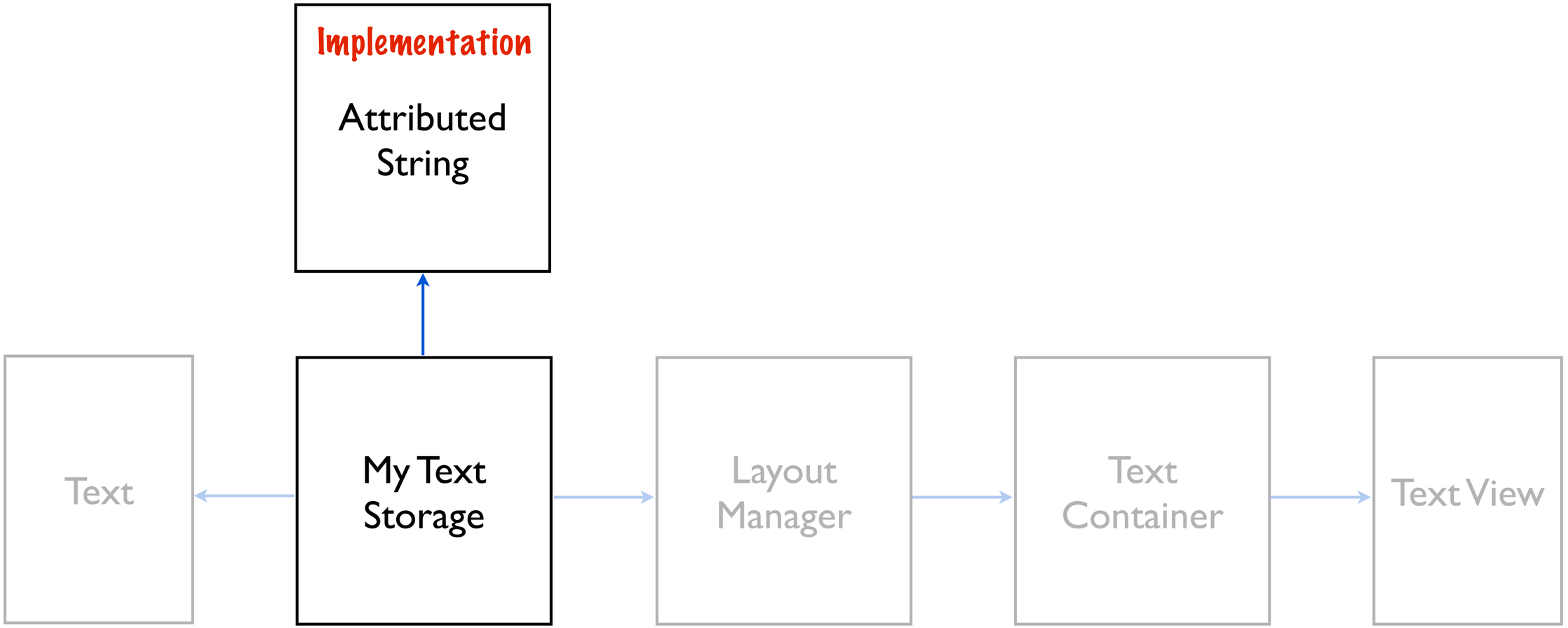
“Syntax Highlighting”

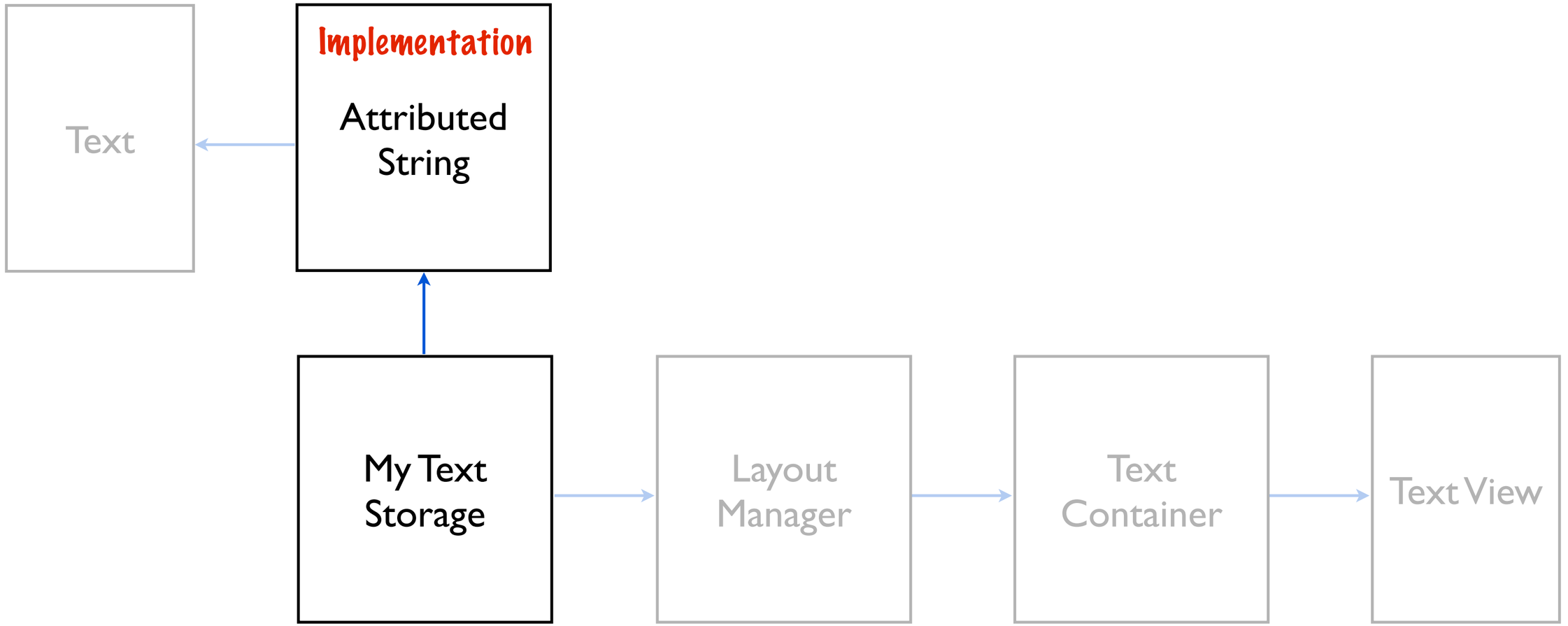
```
Vivamus et turpis in dui blandit pulvinar nec dignissim diam. Nulla
scelerisque posuere _***viverra***_. Quisque nibh nunc, consequat vel luctus in,
consectetur vitae mauris. Nunc **__lacinia__** malesuada mauris, sed egestas
nunc fermentum nec. Suspendisse potenti. Vestibulum ante ipsum primis in
faucibus orci luctus et ultrices posuere cubilia Curae; Vestibulum enim.
```


Typesetter
Glyph Generator
Text
Text Storage
Layout Manager
Text Container
Text View




```
- (void)setUsesFontPanel:(BOOL)flag;
```


Implementation
Attributed String
Text
My Text Storage
Layout Manager
Text Container
Text View


Text
Implementation
Attributed String
My Text Storage
Layout Manager
Text Container
Text View

Attribute überschreiben

- ~~Beim Lesen überschreiben~~

```
- (NSDictionary *)attributesAtIndex:(NSUInteger)location effectiveRange:(NSRangePointer)range
{

}
```

- Beim Schreiben setzen

```
- (void)replaceCharactersInRange:(NSRange)range withString:(NSString *)replacement
{

}
```

Attribute überschreiben

```
- (void)replaceCharactersInRange:(NSRange)range withString:(NSString *)replacement
{
    [_implementation replaceCharactersInRange:range withString:replacement];

    // Text system notification
    NSInteger changeInLength = replacement.length - range.length;
    [self edited:NSTextStorageEditedCharacters range:range changeInLength:changeInLength];


    /*
     Text scannen
     */
    NSRange attrRange = /* … */;
    NSDictionary *attrs = @{ NSForegroundColorAttributeName: NSColor.redColor};
    [_cache setAttributes:attrs range:range];

    [self edited:NSTextStorageEditedAttributes range:attrRange changeInLength:0];
}
```

Problem:

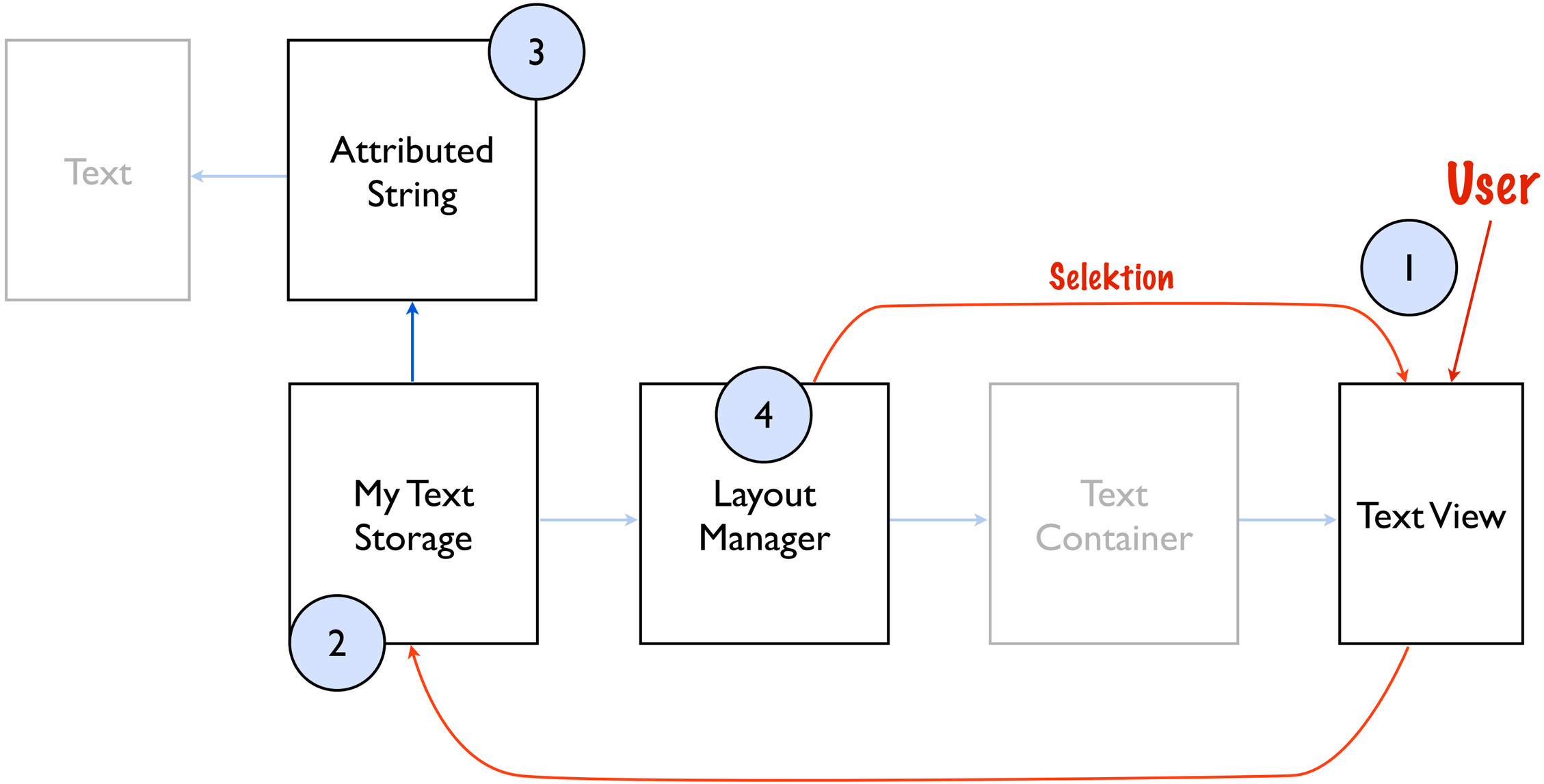
Text Selektion

```
Cum sociis natoque penatibus et magnis dis parturient montes.
```

```
Cum sociis natoque *penatibus et magnis dis parturient montes.
```

```
Cum sociis natoque **penatibus et magnis dis parturient montes.
```

Selektion


Text
3
Attributed String
User
1
Selektion
My Text Storage
2
4
Layout Manager
Text Container
Text View

Änderungen

```
[self edited:NSTextStorageEditedAttributes range:attrRange changeInLength:0];
```

```
- (void)textStorage:(NSTextStorage *)str edited:(NSUInteger)editedMask range:(NSRange)newCharRange
        changeInLength:(NSInteger)delta invalidatedRange:(NSRange)invalidatedCharRange
```

```
- (void)_fixSelectionAfterChangeInCharacterRange:(NSRange)range changeInLength:(NSInteger)delta
```

Private

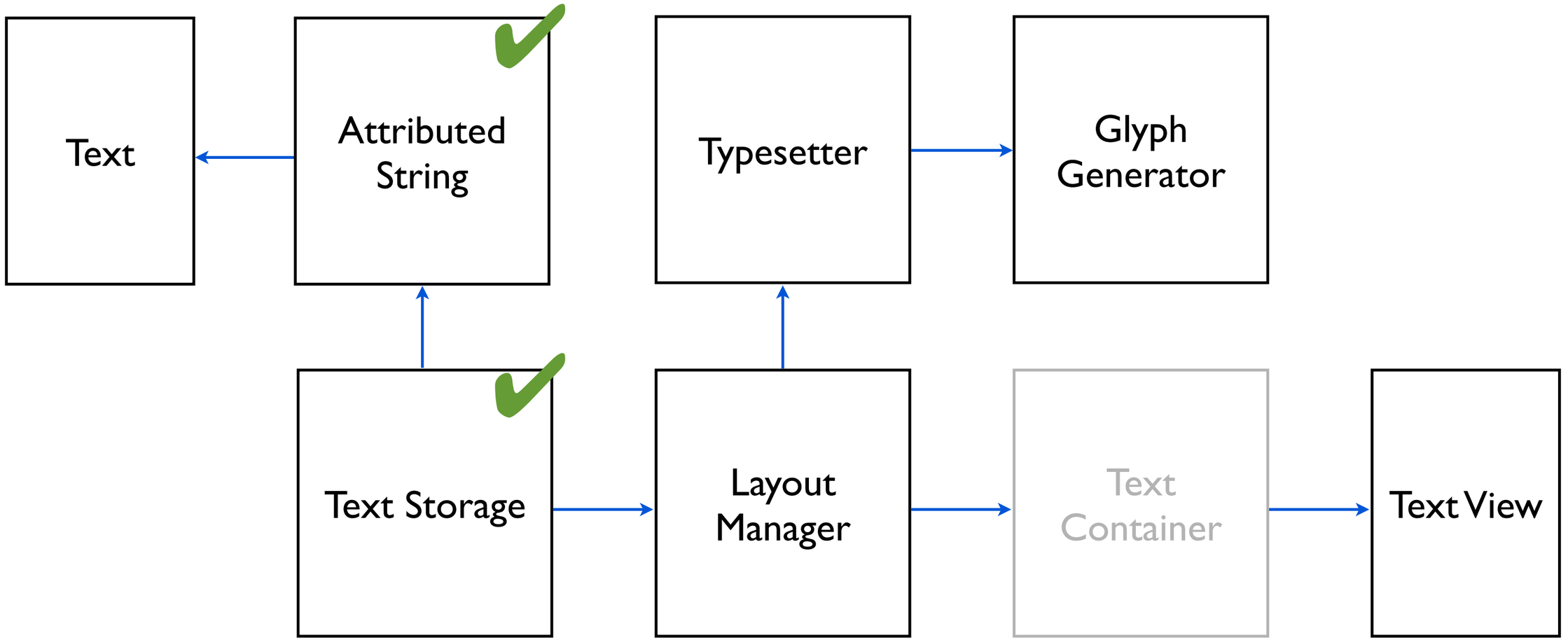
“Echte” Änderungen merken

- Text Storage:

```
- (void)replaceCharactersInRange:(NSRange)range withString:(NSString *)replacement
{
    self.userEditedRange = range;
    [_implementation replaceCharactersInRange:range withString:replacement];

    ...
```

- Layout Manager

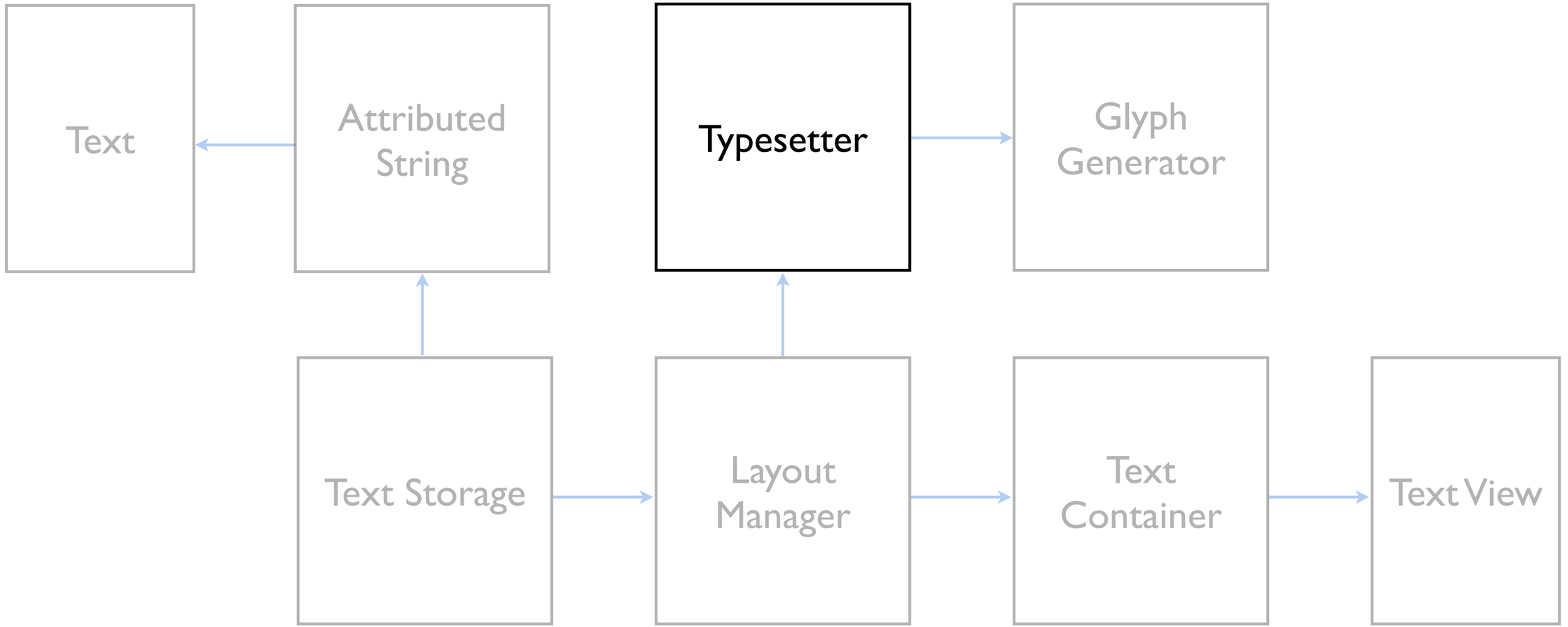
```
- (void)_fixSelectionAfterChangeInCharacterRange:(NSRange)range changeInLength:(NSInteger)delta
{
    if (self.textStorage.userEditedRange.location != NSNotFound)
        range = self.textStorage.userEditedRange;

    [super _fixSelectionAfterChangeInCharacterRange:range changeInLength:delta];
}
```


Text
Attributed String
Typesetter
Glyph Generator
Text Storage
Layout Manager
Text Container
Text View

Aufgabe:

Tolle Typographie

```
Cum sociis natoque penatibus et magnis dis parturient montes.
# Überschrift
Suspendisse potenti. Vestibulum.
```


Text
Attributed String
Typesetter
Glyph Generator
Text Storage
Layout Manager
Text Container
Text View

Einrückung berechnen

```
- (void)beginParagraph
{
    NSRange range = [self.layoutManager.textStorage.string paragraphRangeForRange: self.paragraphCharacterRange];

    /* Compute spacing */
    _paragraphInset = ...;
    _lineInset = ...;

    [super beginParagraph];
}
```

Momentane Einrückung

```
- (void)beginLineWithGlyphAtIndex:(NSUInteger)glyphIndex
{
    [super beginLineWithGlyphAtIndex: glyphIndex];

    // Update front inset
    NSRange range = [self.layoutManager.textStorage.string paragraphRangeForRange: self.paragraphCharacterRange];
    NSUInteger paragraphStart = [self glyphRangeForCharacterRange:range actualCharacterRange:NULL].location;

    _currentInset = (glyphIndex == paragraphStart) ? _paragraphInset : _lineInset;
}
```

```
- (void)getLineFragmentRect:(NSRectPointer)lineFragmentRect usedRect:(NSRectPointer)lineFragmentUsedRect
        remainingRect:(NSRectPointer)remainingRect forStartingGlyphAtIndex:(NSUInteger)startingGlyphIndex
        proposedRect:(NSRect)proposedRect lineSpacing:(CGFloat)lineSpacing
        paragraphSpacingBefore:(CGFloat)paragraphSpBefore paragraphSpacingAfter:(CGFloat)paragraphSpAfter
{
    [super getLineFragmentRect:lineFragmentRect usedRect:lineFragmentUsedRect remainingRect:remainingRect
        forStartingGlyphAtIndex:startingGlyphIndex proposedRect:proposedRect lineSpacing:lineSpacing
        paragraphSpacingBefore:paragraphSpBefore paragraphSpacingAfter:paragraphSpAfter];

    // Setting a non-zero X coordinate hangs up the text system
    if (lineFragmentRect)
        lineFragmentRect->size.width -= _currentInset;
    if (lineFragmentUsedRect && lineFragmentRect)
       lineFragmentUsedRect->size.width = fminf(lineFragmentUsedRect->size.width, lineFragmentRect->size.width);
}

- (void)willSetLineFragmentRect:(NSRect *) lineFragmentRect forGlyphRange:(NSRange)glyphRange usedRect:(NSRect
*)lineFragmentUsedRect baselineOffset:(CGFloat *)baselineOffset
{
    lineFragmentRect->origin.x += _currentInset;
    lineFragmentUsedRect->origin.x += _currentInset;
}
```

```
- (void)getLineFragmentRect:(NSRectPointer)lineFragmentRect usedRect:(NSRectPointer)lineFragmentUsedRect
        remainingRect:(NSRectPointer)remainingRect forStartingGlyphAtIndex:(NSUInteger)startingGlyphIndex
        proposedRect:(NSRect)proposedRect lineSpacing:(CGFloat)lineSpacing
        paragraphSpacingBefore:(CGFloat)paragraphSpBefore paragraphSpacingAfter:(CGFloat)paragraphSpAfter
{
    [super getLineFragmentRect:lineFragmentRect usedRect:lineFragmentUsedRect remainingRect:remainingRect
        forStartingGlyphAtIndex:startingGlyphIndex proposedRect:proposedRect lineSpacing:lineSpacing
        paragraphSpacingBefore:paragraphSpBefore paragraphSpacingAfter:paragraphSpAfter];

    // Setting a non-zero X coordinate hangs up the text system
    if (lineFragmentRect)
        lineFragmentRect->size.width -= _currentInset;
    if (lineFragmentUsedRect && lineFragmentRect)
        lineFragmentUsedRect->size.width = fminf(lineFragmentUsedRect->size.width, lineFragmentRect->size.width);
}

- (void)willSetLineFragmentRect:(NSRect *) lineFragmentRect forGlyphRange:(NSRange)glyphRange usedRect:(NSRect
*)lineFragmentUsedRect baselineOffset:(CGFloat *)baselineOffset
{
    lineFragmentRect->origin.x += _currentInset;
    lineFragmentUsedRect->origin.x += _currentInset;
}
```

Aufgabe:

Noch Tollere Typographie

```
Cum sociis natoque penatibus et magnis dis parturient montes.
   1. Vestibulum enim.
   2. Quisque nibh nunc.
   3.    1. Fusce tincidunt erat
         2. Sit amet magna porttitor
Suspendisse potenti. Vestibulum.
```

Cum sociis natoque penatibus et magnis dis parturient montes.

1. Vestibulum enim.
2. Quisque nibh nunc.
3. 1. Fusce tincidunt erat
   2. Sit amet magna porttitor

Suspendisse potenti. Vestibulum.

```
magnis dis parturi
  1. Vestibulum en
  2. Quisque nibh
  3.    1. Fusce ti
           2. Sit amet
                porttito
```

magnis dis parturi

1. Vestibulum en

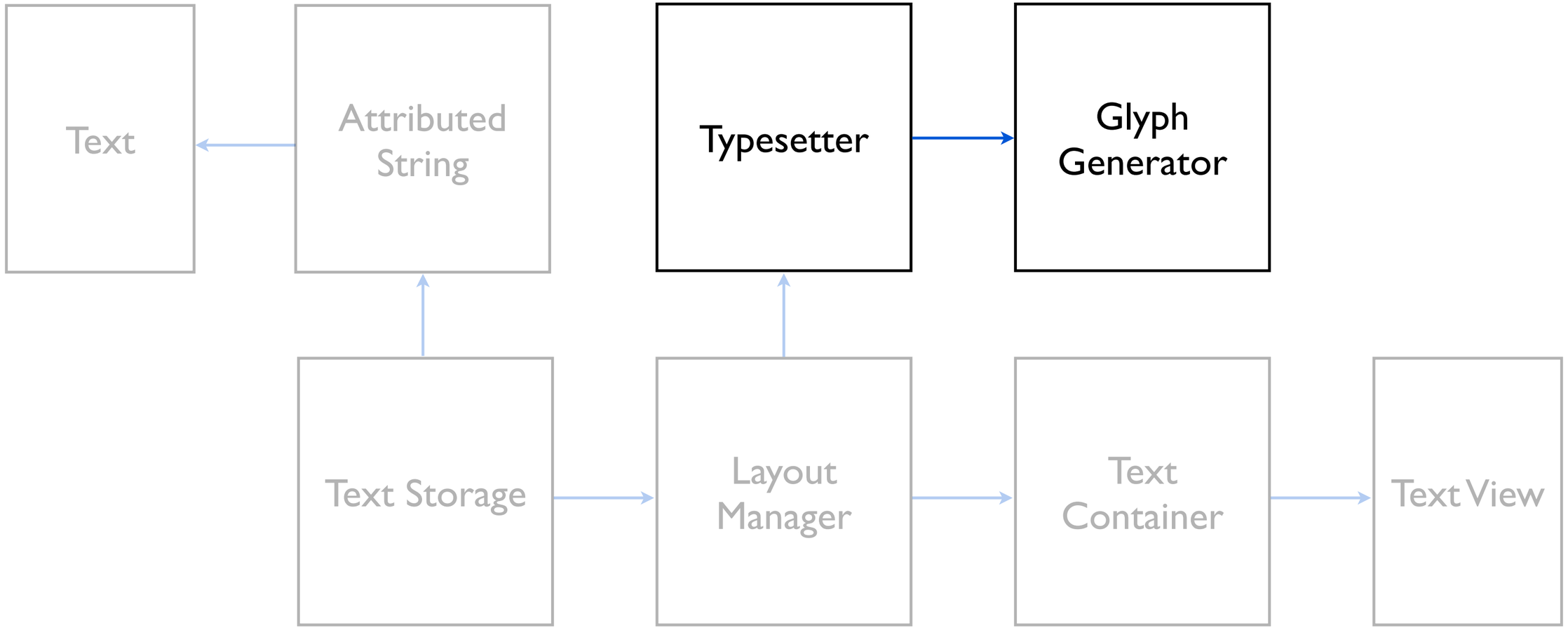
2. Quisque nibh

3. 1. Fusce ti

2. Sit amet

porttito

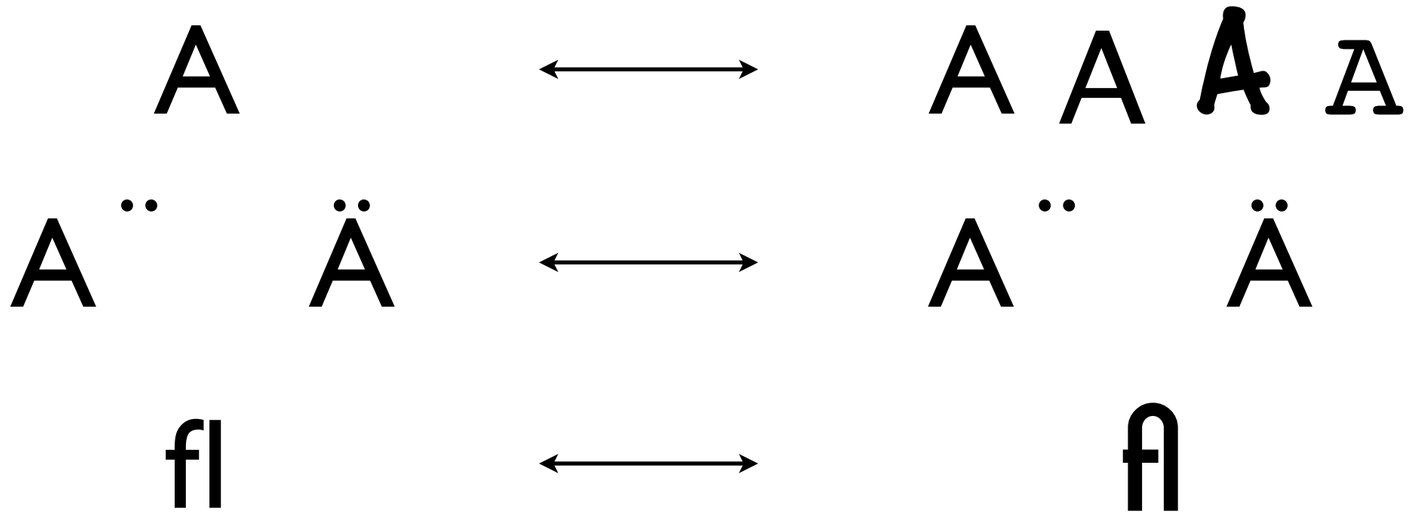
Suspendisse potent


Text
Attributed String
Typesetter
Glyph Generator
Text Storage
Layout Manager
Text Container
Text View

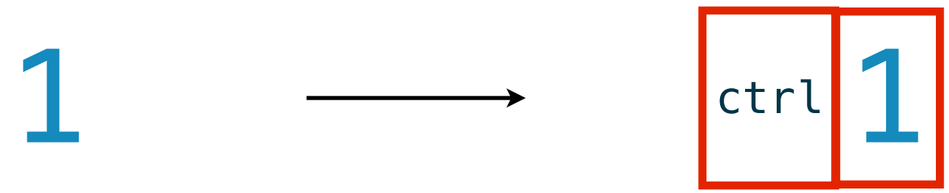
Zeichen != Zeichen

Characters

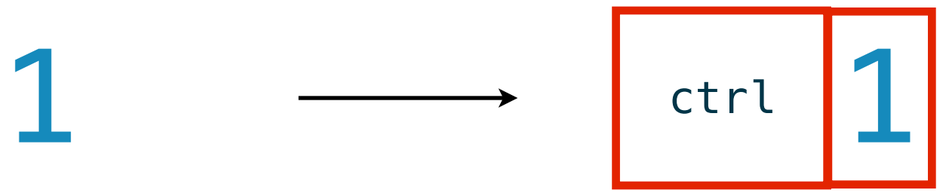
Glyphs



Typesetter



Typesetter



Glyph Generator

```
- (void)generateGlyphsForGlyphStorage:(id<NSGlyphStorage>)glyphStorage desiredNumberOfCharacters:
            (NSUInteger)nChars glyphIndex:(NSUInteger *)glyphIndex characterIndex:(NSUInteger *)charIndex
{
    NSRange generatedRange = NSMakeRange(*charIndex, 0);

    // Generate glyphs as usual
    [[NSGlyphGenerator sharedGlyphGenerator] generateGlyphsForGlyphStorage:glyphStorage
                         desiredNumberOfCharacters:nChars glyphIndex:glyphIndex characterIndex:charIndex];
    generatedRange.length = *charIndex - generatedRange.location;

    // Enumerate generated range
    for (NSRange visibleRange in /* Logik */) {
        NSUInteger firstGlyphIndex = [(NSLayoutManager *)glyphStorage
                                                    glyphIndexForCharacterAtIndex:visibleRange.location];

        [(NSLayoutManager *)glyphStorage insertGlyph:NSControlGlyph atGlyphIndex:firstGlyphIndex
                                                                  characterIndex:visibleRange.location];
        (*glyphIndex)++;
    }];
}
```

Glyph Generator

```
- (void)generateGlyphsForGlyphStorage:(id<NSGlyphStorage>)glyphStorage desiredNumberOfCharacters:
            (NSUInteger)nChars glyphIndex:(NSUInteger *)glyphIndex characterIndex:(NSUInteger *)charIndex
{
    NSRange generatedRange = NSMakeRange(*charIndex, 0);

    // Generate glyphs as usual
    [[NSGlyphGenerator sharedGlyphGenerator] generateGlyphsForGlyphStorage:glyphStorage
                         desiredNumberOfCharacters:nChars glyphIndex:glyphIndex characterIndex:charIndex];
    generatedRange.length = *charIndex - generatedRange.location;

    // Enumerate generated range
    for (NSRange visibleRange in /* Logik */) {
        NSUInteger firstGlyphIndex = [(NSLayoutManager *)glyphStorage
                                                    glyphIndexForCharacterAtIndex:visibleRange.location];

        [(NSLayoutManager *)glyphStorage insertGlyph:NSControlGlyph atGlyphIndex:firstGlyphIndex
                                                              characterIndex:visibleRange.location];

        (*glyphIndex)++;
    }];
}
```

Typesetter

```
- (NSTypesetterControlCharacterAction)actionForControlCharacterAtIndex:(NSUInteger)charIndex
{
   if (/* Char index modified */)
       return NSTypesetterWhitespaceAction;
   else
       return [super actionForControlCharacterAtIndex: charIndex];
}


- (NSRect)boundingBoxForControlGlyphAtIndex:(NSUInteger)index forTextContainer:(NSTextContainer *)container
proposedLineFragment:(NSRect)rect glyphPosition:(NSPoint)glyphPosition characterIndex:(NSUInteger)charIndex
{
   CGFloat location = /* Precomputed position */;

   CGRect bounding = NSZeroRect;
   bounding.origin = glyphPosition;
   bounding.size.width = location - glyphPosition.x;

   return bounding;
}
```

magnis dis parturi

1. Vestibulum en

2. Quisque nibh

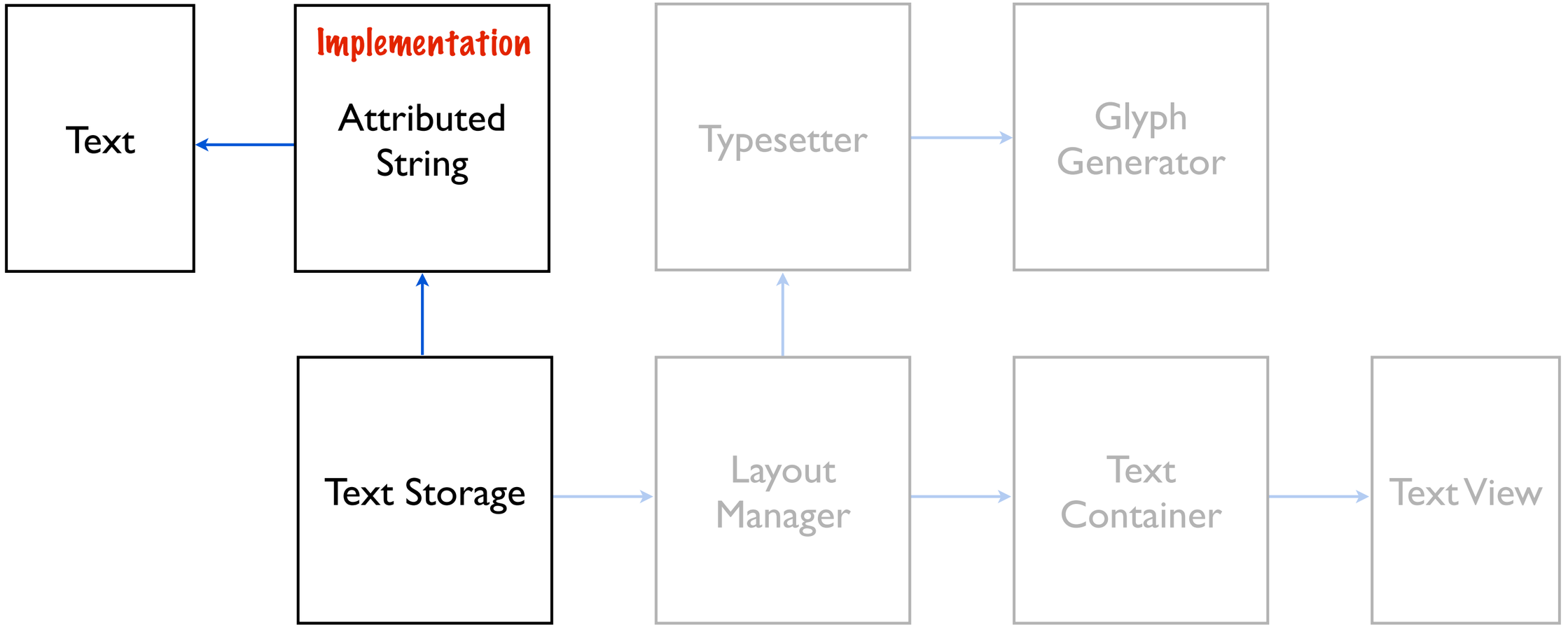
3. 1. Fusce ti

2. Sit amet

porttito

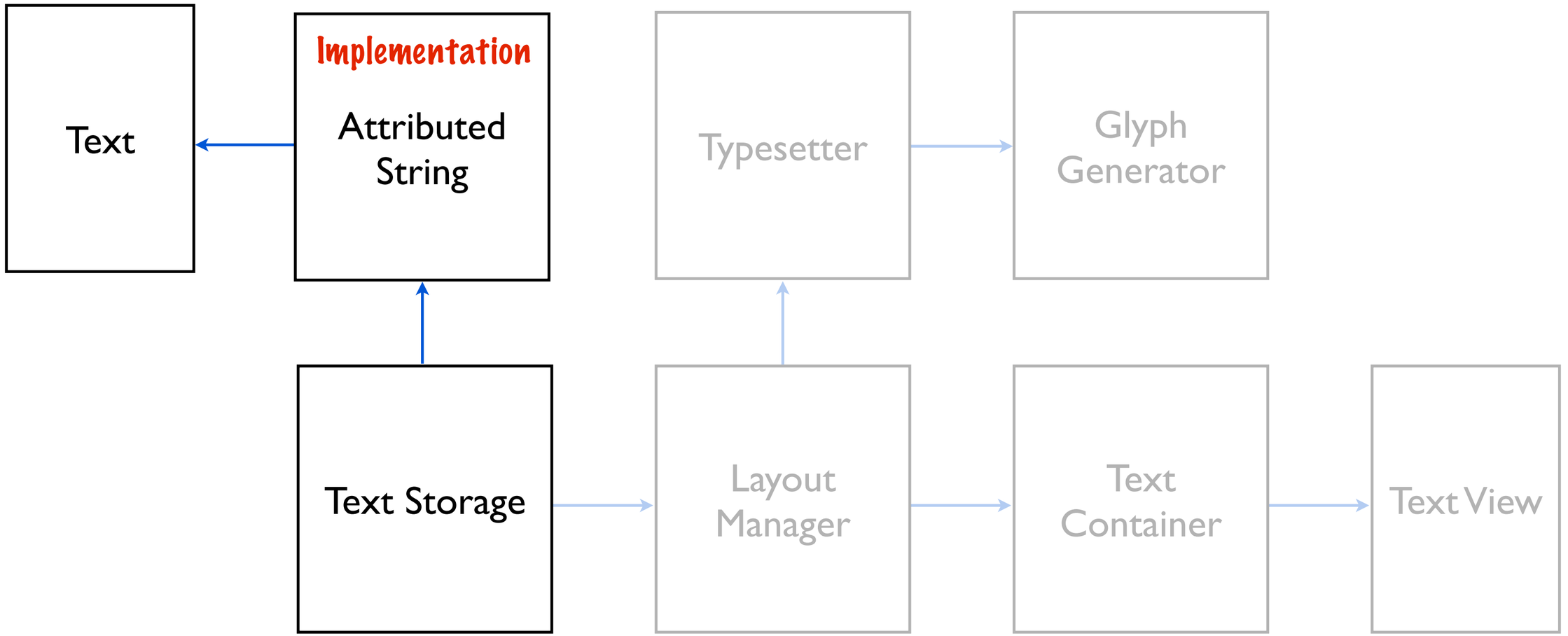
Logischer Schritt:

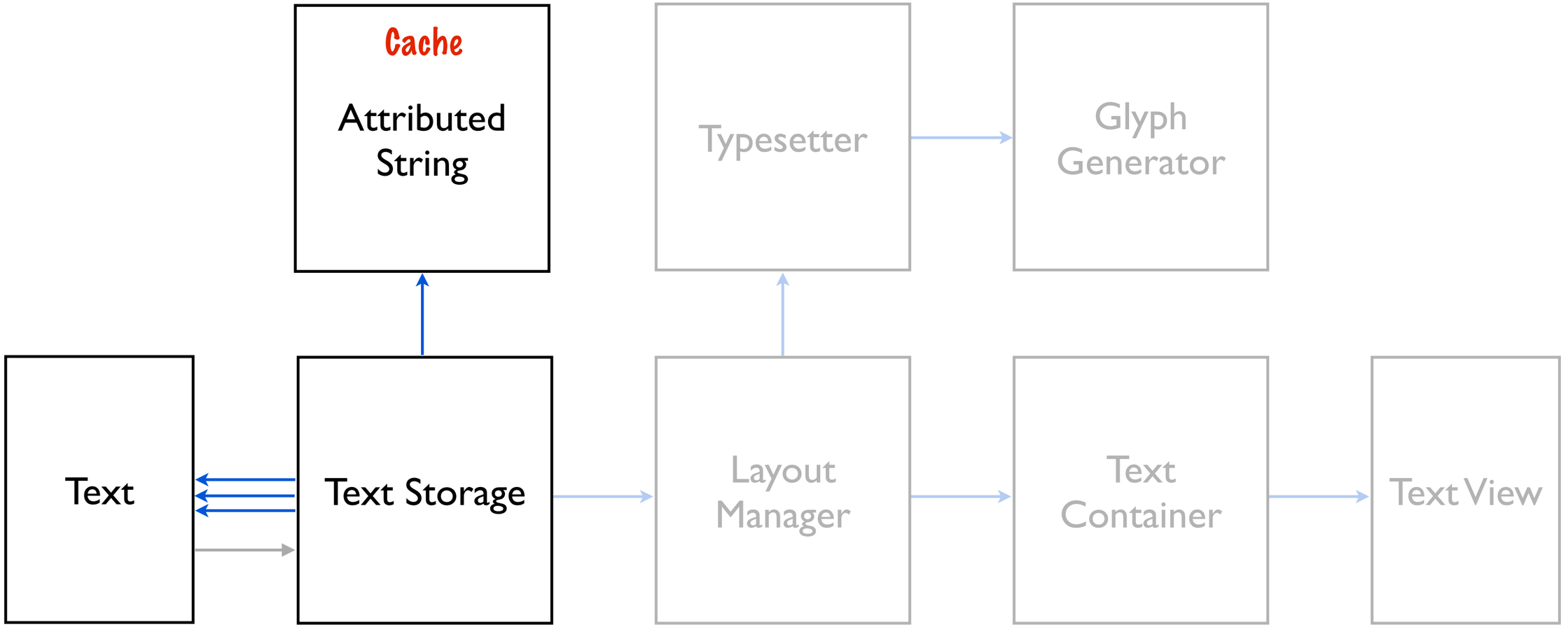
Eigenes String

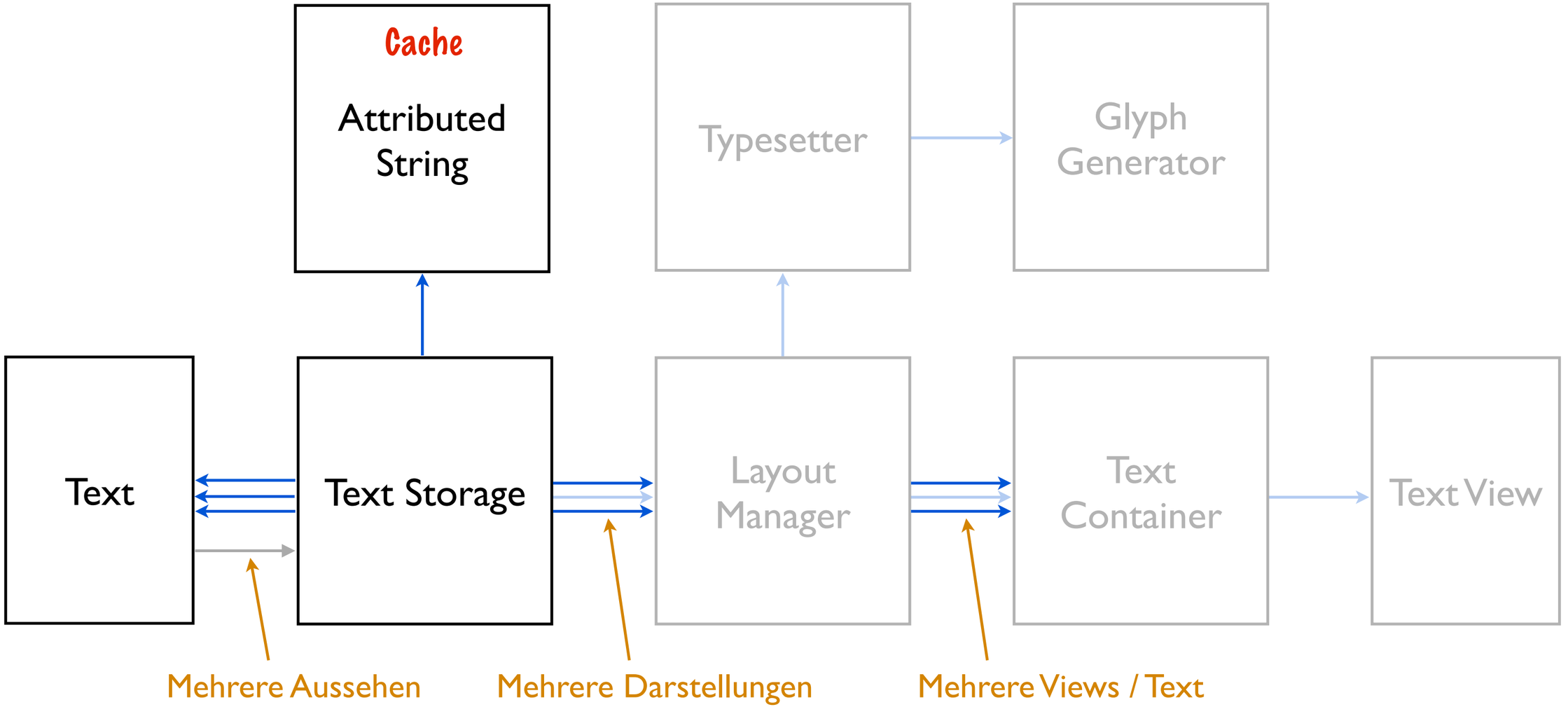

Text
Implementation
Attributed String
Typesetter
Glyph Generator
Text Storage
Layout Manager
Text Container
Text View

Gute Gründe

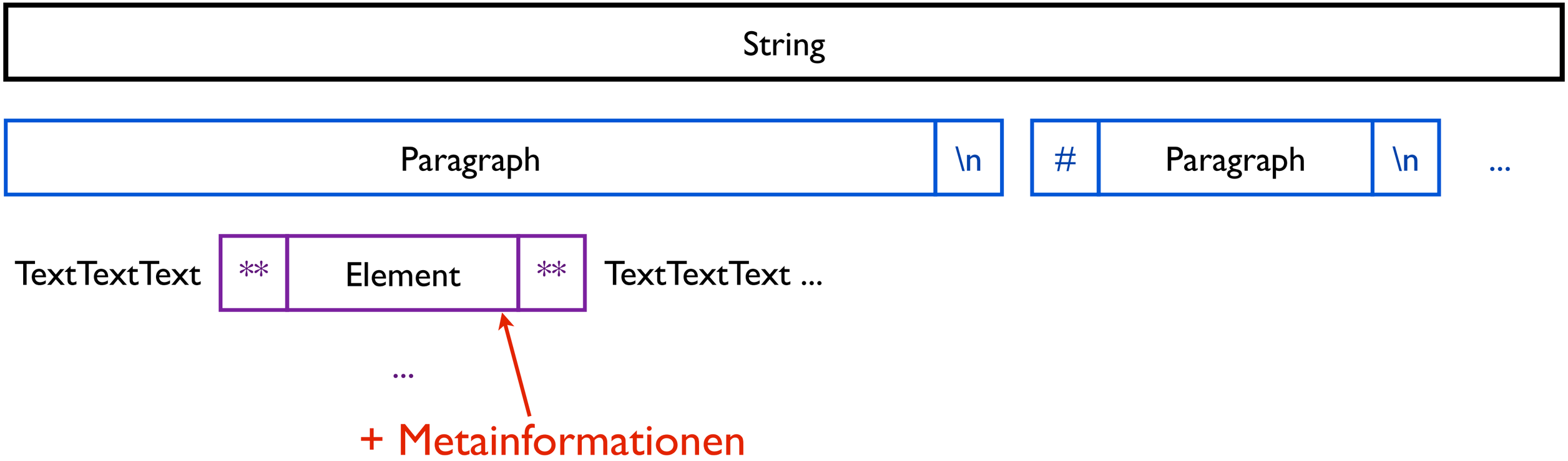
- VIEL Textmining
- Effizienter Zugriff
- Verschiedene Darstellungen
- Abbilden von zusätzlichen Informationen


Text
Implementation
Attributed String
Typesetter
Glyph Generator
Text Storage
Layout Manager
Text Container
Text View


Cache
Attributed String
Typesetter
Glyph Generator
Text
Text Storage
Layout Manager
Text Container
Text View


Cache
Attributed String
Typesetter
Glyph Generator
Text
Text Storage
Layout Manager
Text Container
Text View
Mehrere Aussehen
Mehrere Darstellungen
Mehrere Views / Text

Baum-Basiertes String

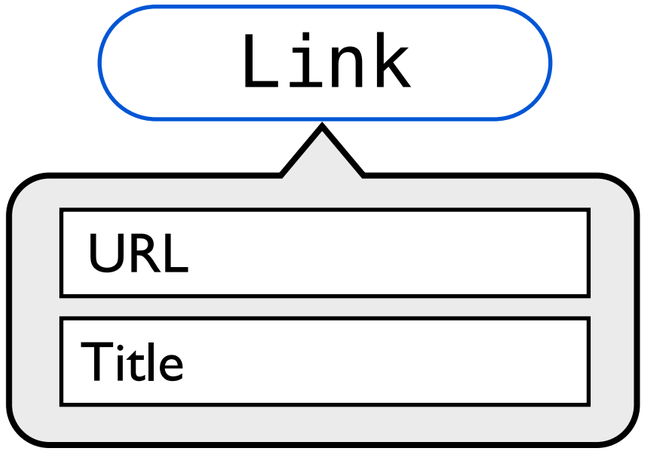


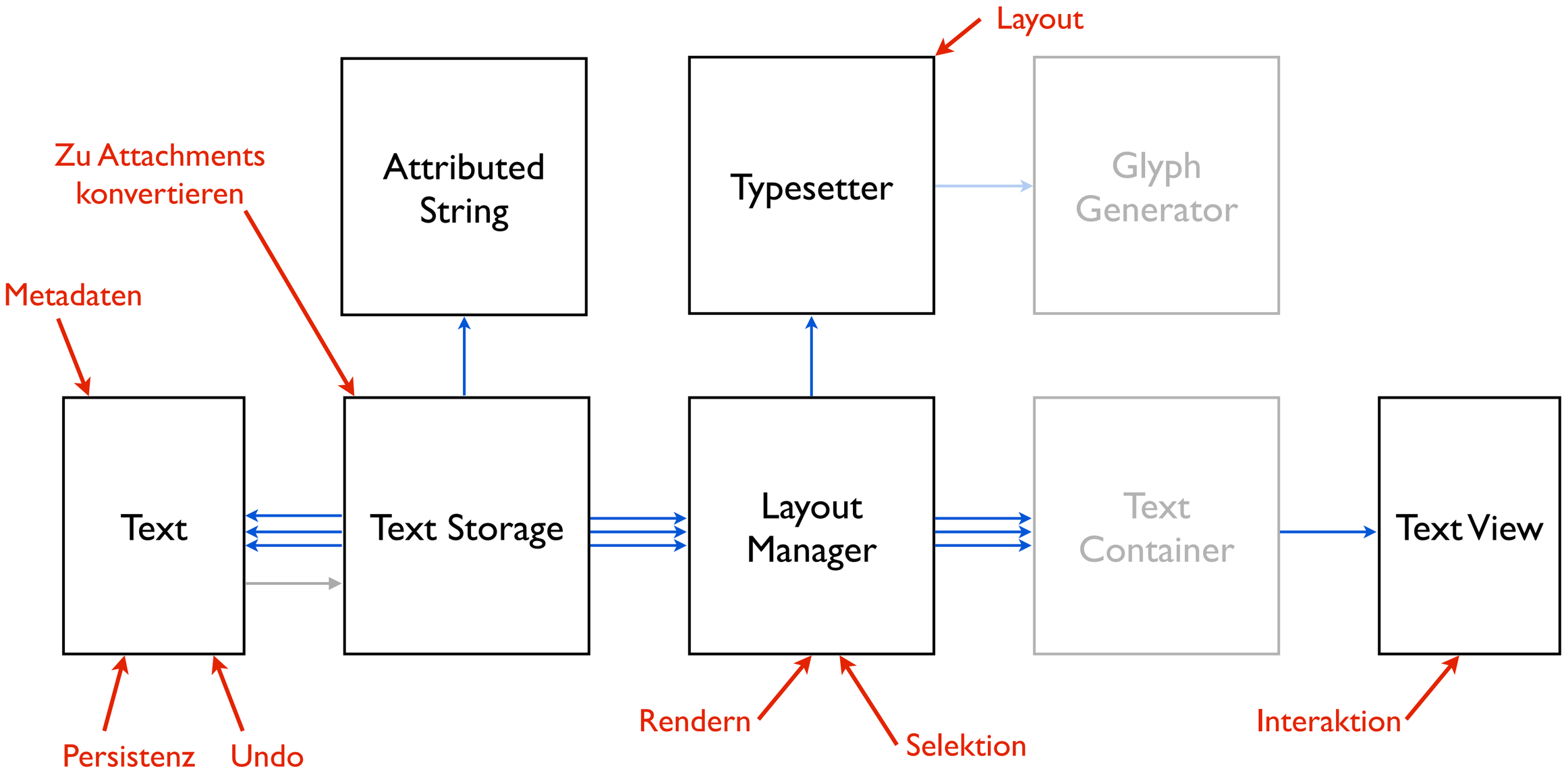
Die Herausforderung

Metadaten

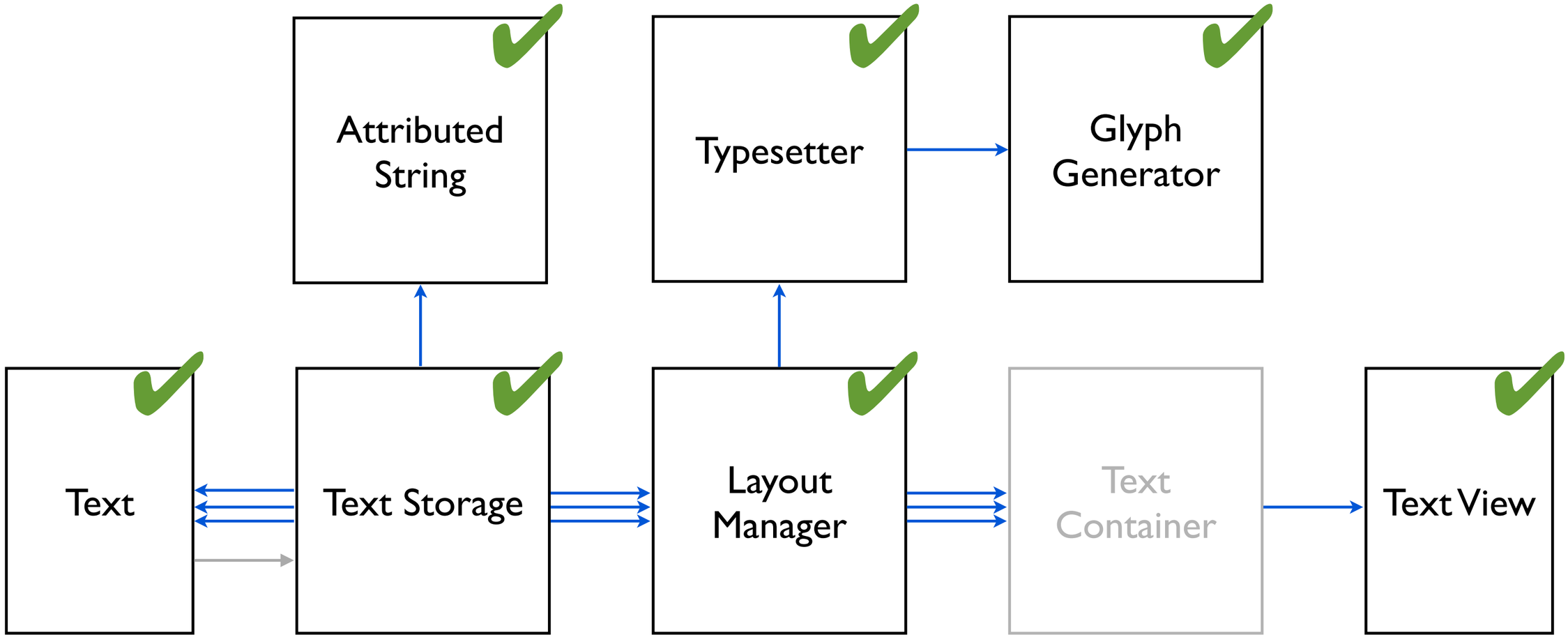
Meta vs. Syntax

```
[ Link ] ( URL ) “Title” )
```



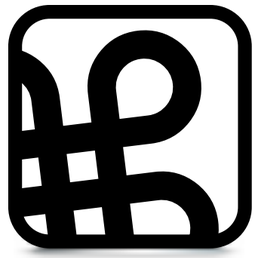

Layout
Attributed String
Typesetter
Glyph Generator
Zu Attachments konvertieren
Metadaten
Text
Text Storage
Layout Manager
Text Container
Text View
Persistenz
Undo
Rendern
Selektion
Interaktion

Demo?


Attributed String
Typesetter
Glyph Generator
Text
Text Storage
Layout Manager
Text Container
Text View

Fragen?

Vielen Dank



Macoun