

Backward Compatibility Maintenance in Bloc

Federico Lochbaum, Pablo Tesone, Guillermo Polito and Martín Dias



Me



miro

Me



The Team



Pablito



Guille



Martín

Bloc - Low-level UI framework for Pharo

AST-Core

- Tokens
- Pattern
- Formatter
- Nodes
- Visitors
- Manifest
- Notice
- Matching
- Type
- Exception
- Nodes - ErrorNodes
- Parser

AST-Core-Tests

Album

- Album-Alexandrie

Album-Examples

Album-Tests

Alexandrie-Base

OCAbstractFormatter

- OCSimpleFormatter

public interface

utilities

private

- private - formatting

initialization

visiting

accessing

formatArray:

formatBlock:

formatBlockArgumentsFor:

formatCommentsFor:

formatMethodBodyFor:

formatMethodCommentFor:

formatMethodPatternFor:

formatPragmasFor:

formatSelectorAndArguments:

formatSelectorAndArguments: firstSeparator: restSeparator:

formatSequenceNodeStatementsFor:

formatStatementCommentsFor:

Packages filter

1 formatMethodBodyFor: aMethodNode

2 self

3 indentAround: [

4 self

5 newLine;

6 formatMethodCommentFor: aMethodNode;

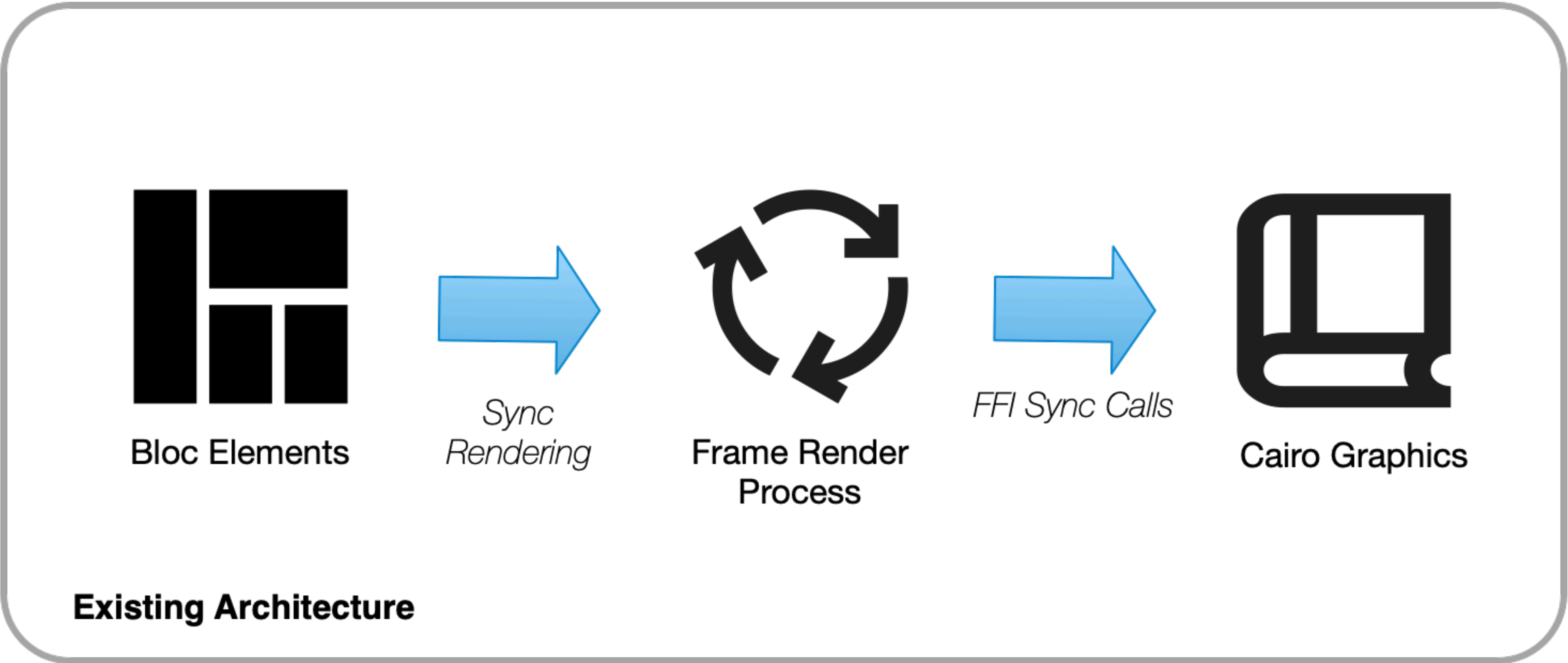
7 formatPragmasFor: aMethodNode;

8 visitNode: aMethodNode body]

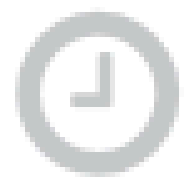
☐ Class side

Coming soon...

Current Render Architecture



Current Render Architecture



3:00 pm - 3:30 pm

Bloc for Pharo: What's New?

Bloc Elements

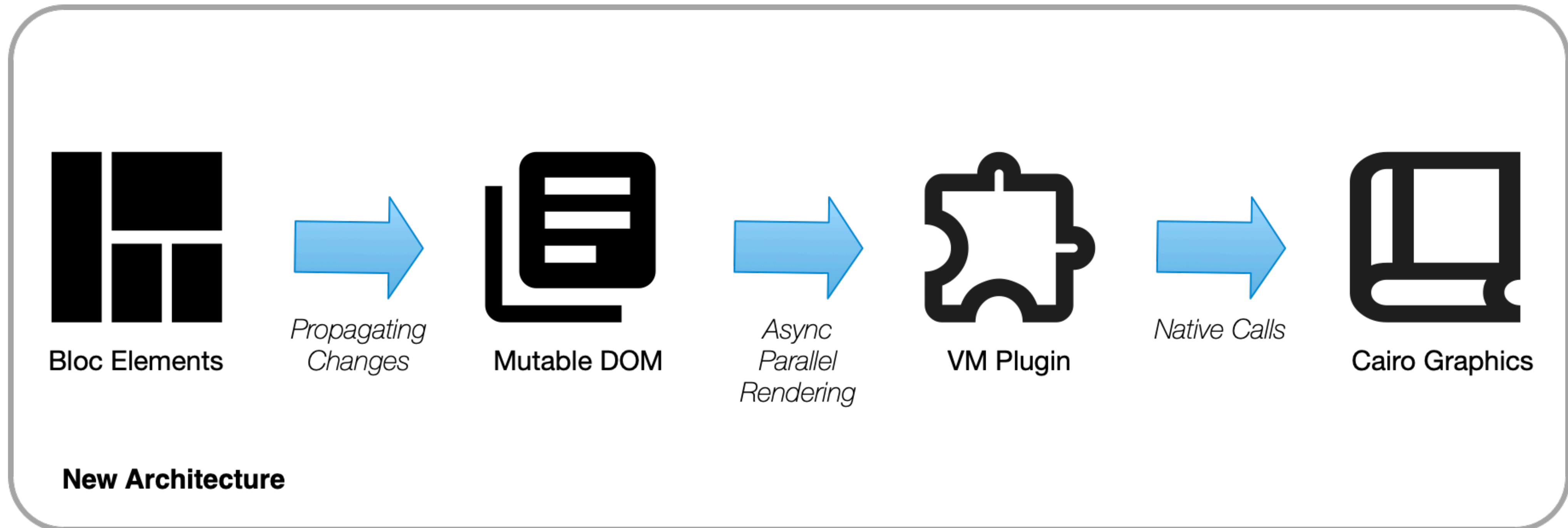
Rendering

Frame Render
Process

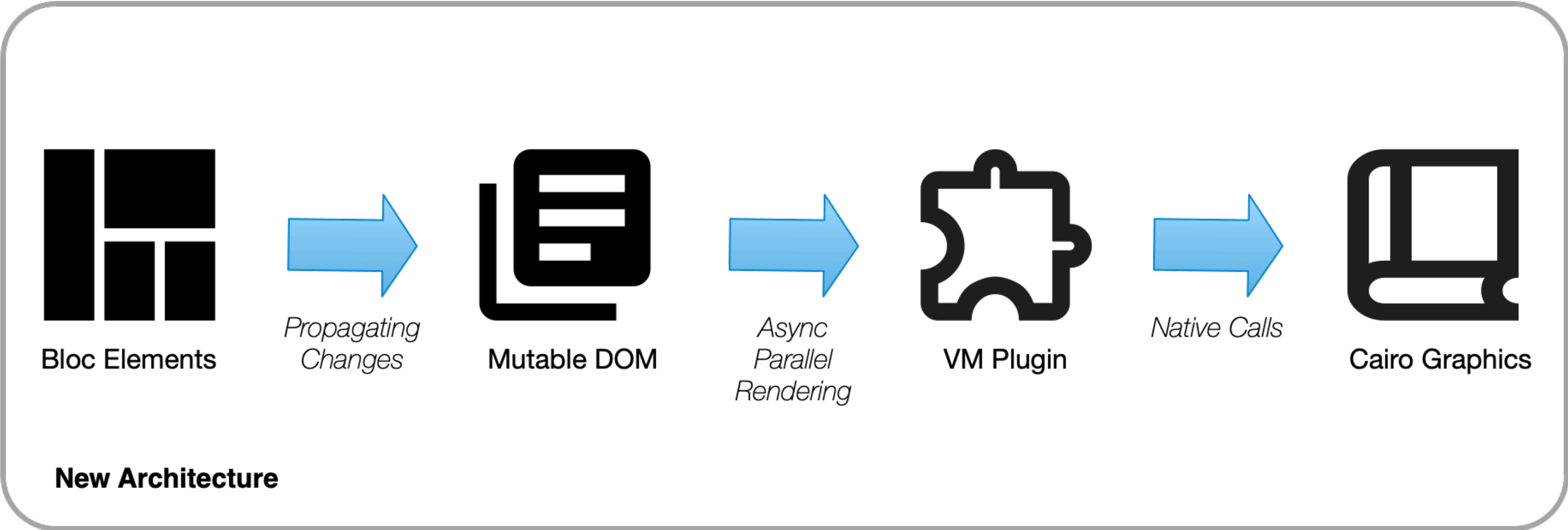
Cairo Graphics

Existing Architecture

Ladies and gentlemen, we have a new Bloc Render!



Ladies and gentlemen, we have a new Bloc Render!



(Yay!)

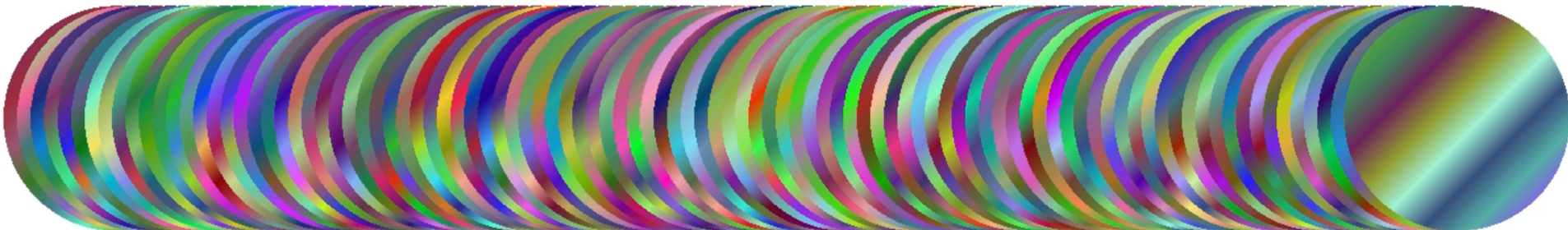
Now what ?

Do we replace it.
That's it ?

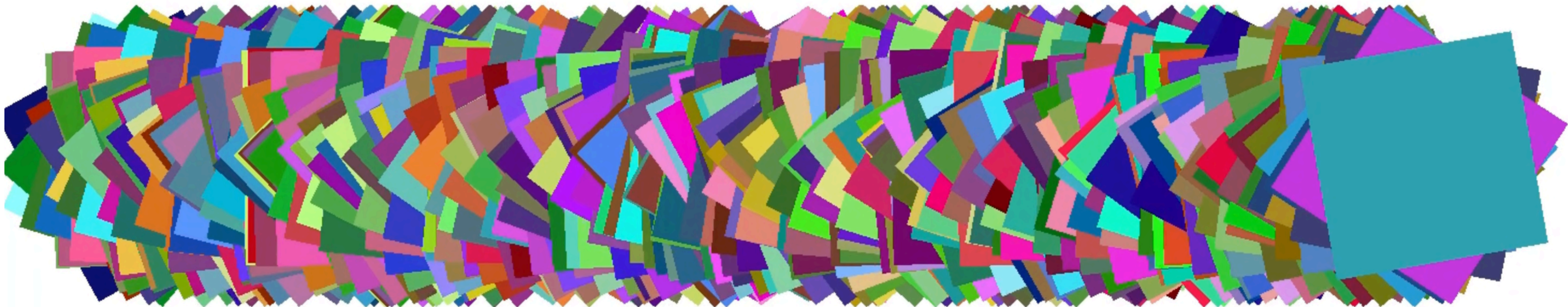
If we have tests...



We have some tests, yep (and benchmarks)



- Errors (0)
- Failures (10)
- Skipped tests (0)
- Passing tests (439)
- ExpectedFailures (0)
- Unexpected passed tests (0)



We have some tests,
yep
(and benchmarks)

**But, how do we reduce
the risk of backward
incompatibilities ?**

- Errors (0)
- Failures (10)
- Skipped tests (0)
- Passing tests (439)
- ExpectedFailures (0)
- Unexpected passed tests (0)

Rendering Problems

Rendering Problems

**Visual
Inconsistencies**

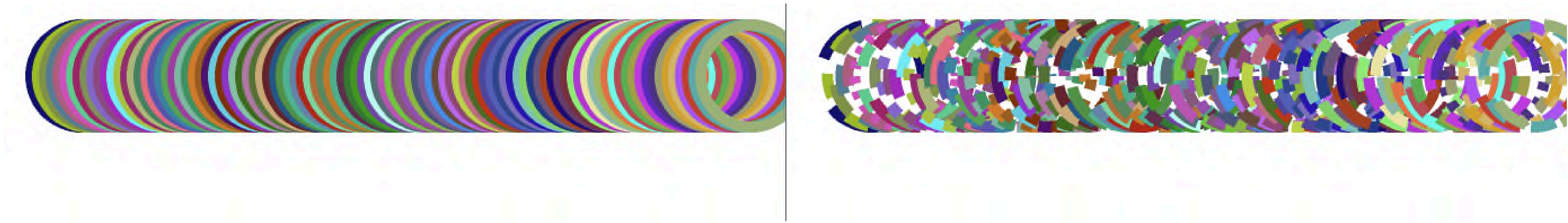
**Performance
Degradation**

**Performance
optimization
differences**

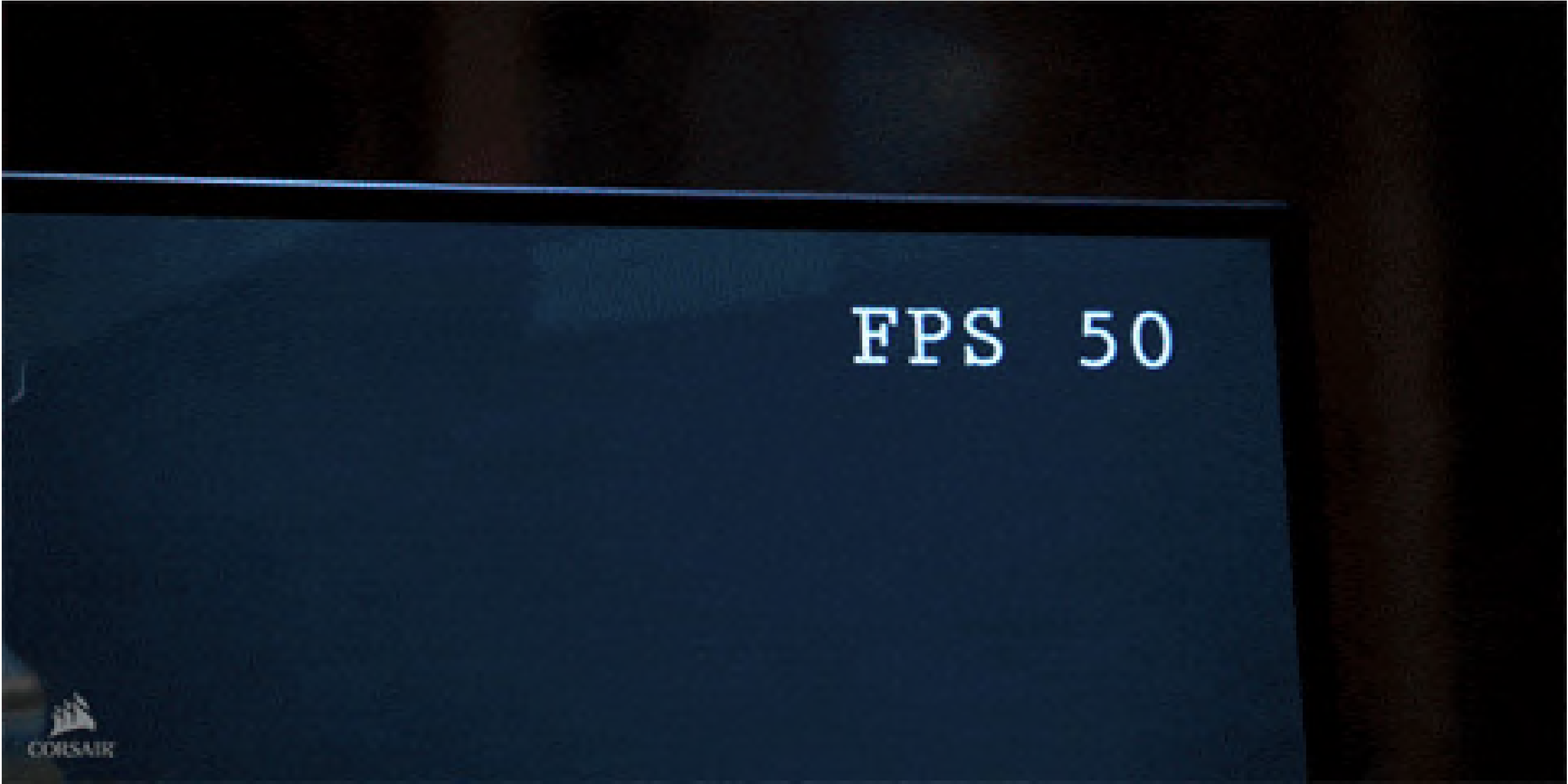
**DOM
Invalidations**

**Excessively
Memory
Consumption**

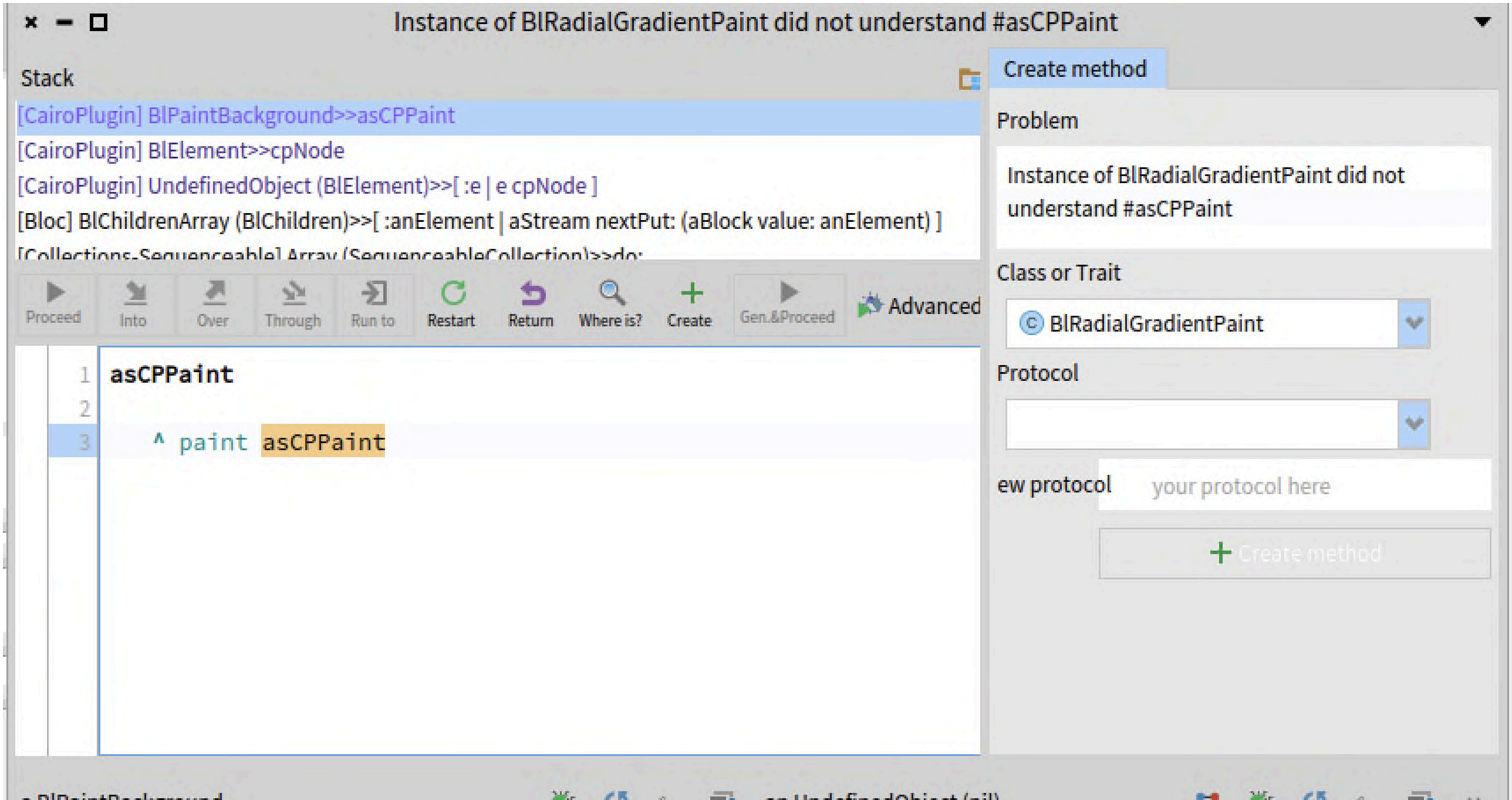
Visual Inconsistencies



FPS Drops



Feature Diversity Errors



Existing work

Visual-Aware Testing and Debugging for Web Performance Optimization

Xinlei Yang*
Tsinghua University, China

Wei Liu*
Tsinghua University, China

Hao Lin
Tsinghua University, China

Zenhua Li
Tsinghua University, China

Feng Qian
University of Minnesota, USA

Xianlong Wang
Tsinghua University, China

Yunhao Liu
Tsinghua University, China

Tianyin Xu
UIUC, USA

2023

Favocado: Fuzzing the Binding Code of JavaScript Engines Using Semantically Correct Test Cases

Sung Ta Dinh*, Haehyun Cho*, Kyle Martin†, Adam Oest‡, Kyle Zeng*, Alexandros Kapravelos†, Gail-Joon Ahn*§, Tiffany Bao*, Ruoyu Wang*, Adam Doupe*, and Yan Shoshitaishvili*

*Arizona State University, †North Carolina State University, ‡PayPal, Inc., §Samsung Research

{tdsung, haehyun, zengyhkyle, gahn, tbao, fishw, doupe, yans}@asu.edu

{kdmarti2, akaprav}@ncsu.edu, {aoest}@paypal.com

2021

METAMONG: Detecting Render-Update Bugs in Web Browsers through Fuzzing

Suhwan Song
Seoul National University
Seoul, South Korea
sshkeb96@snu.ac.kr

Byoungyoung Lee*
Seoul National University
Seoul, South Korea
byoungyoung@snu.ac.kr



2023

What do we propose?

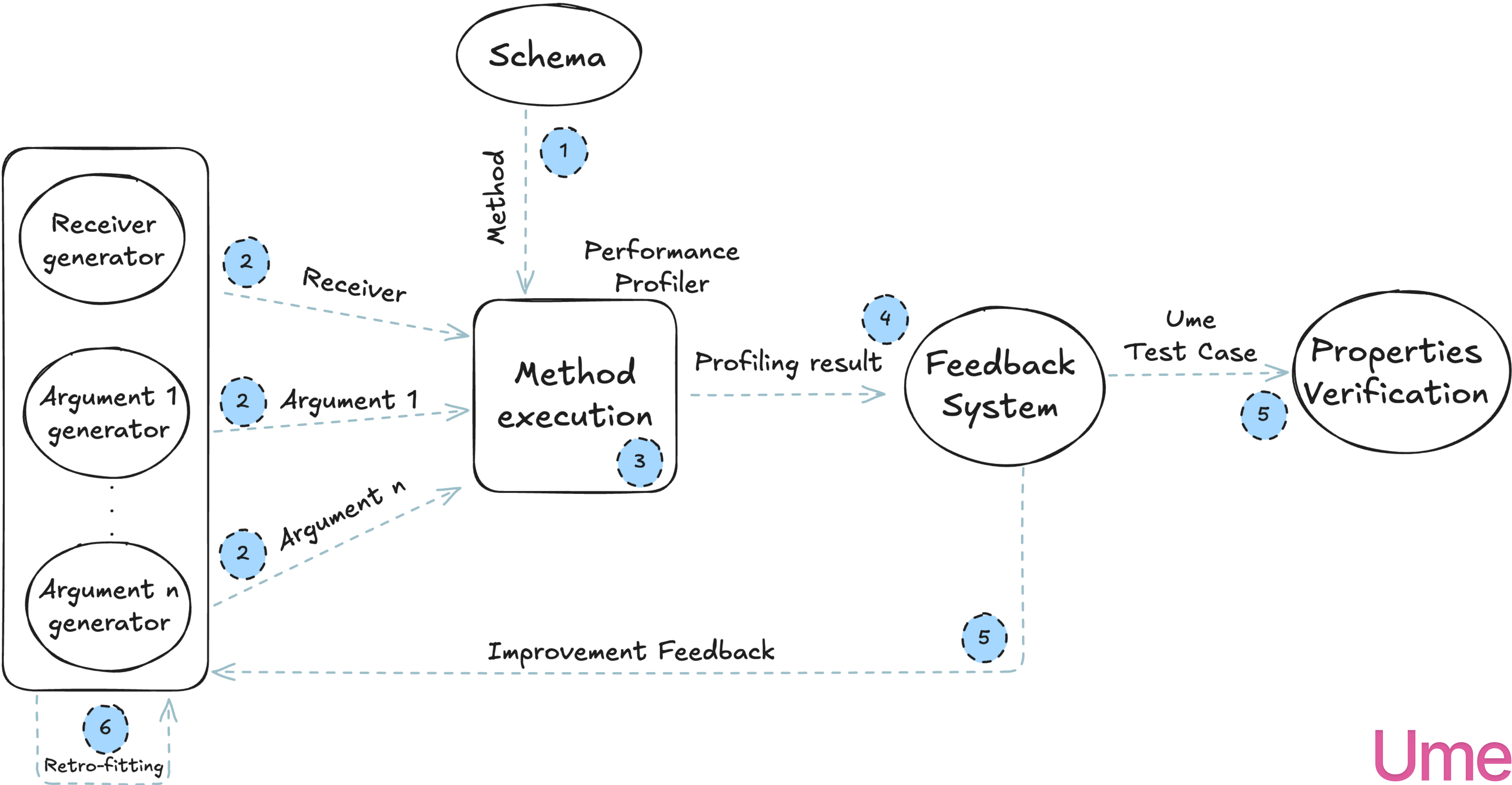
What do we propose?

Feature-based Guided
Grammar Fuzzing
+
Differential Testing

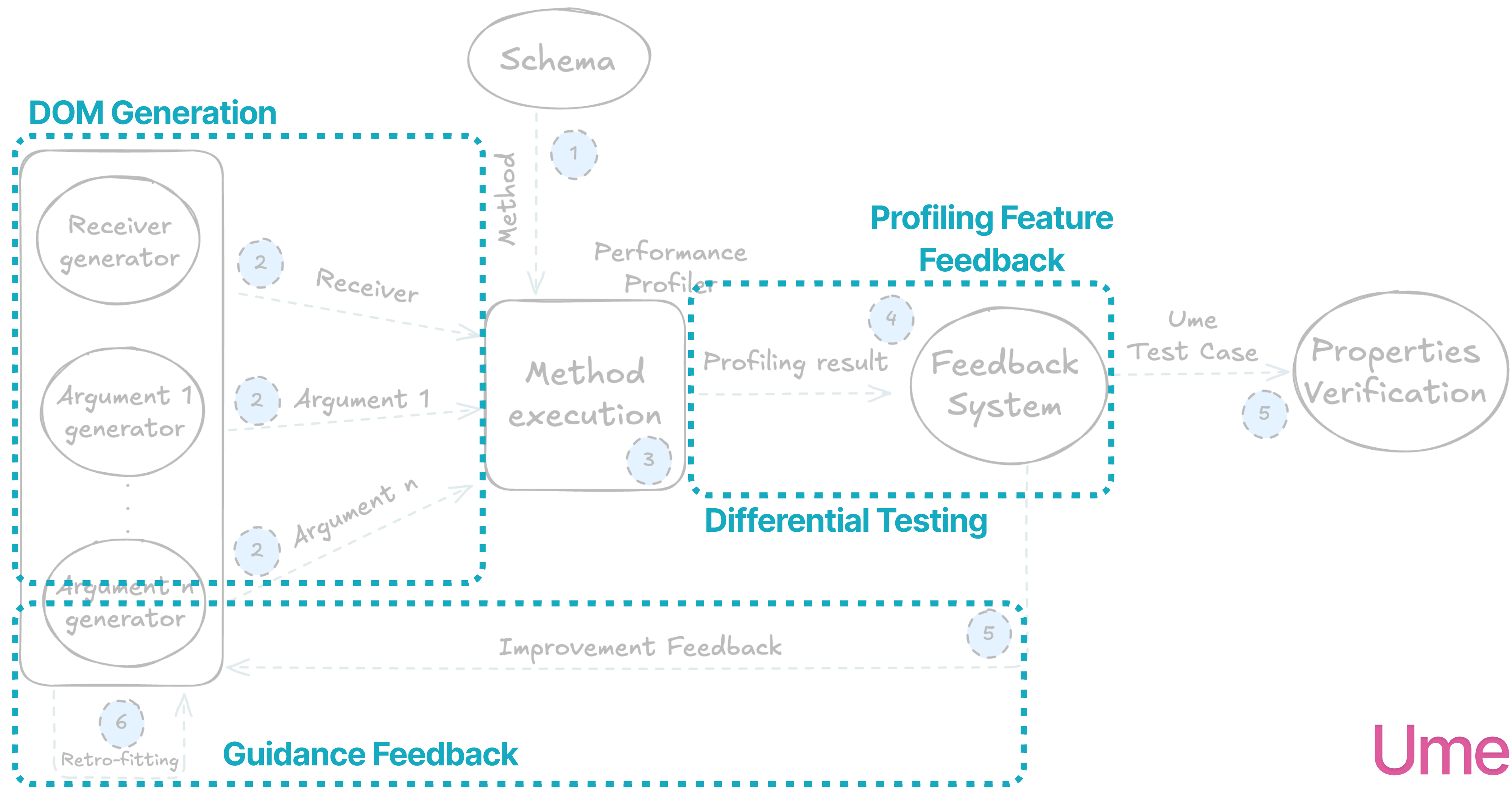
Automatic Test Generation & Guided Fuzzing



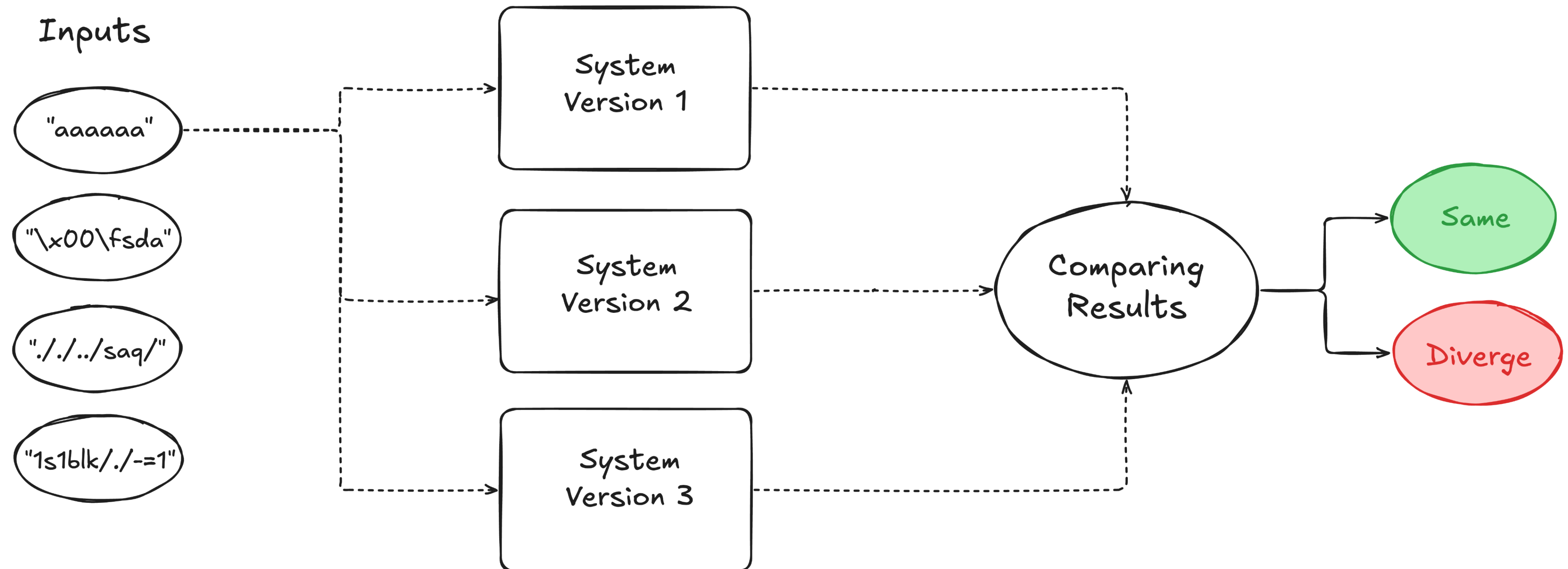
Ume Architecture



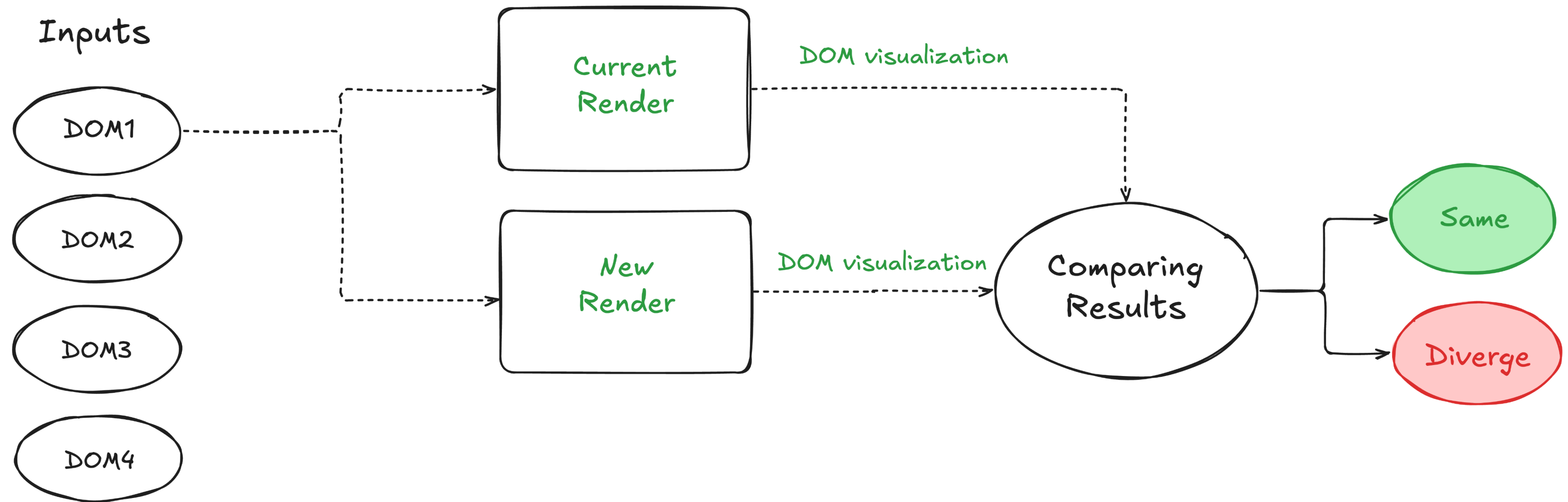
Guided Fuzzing



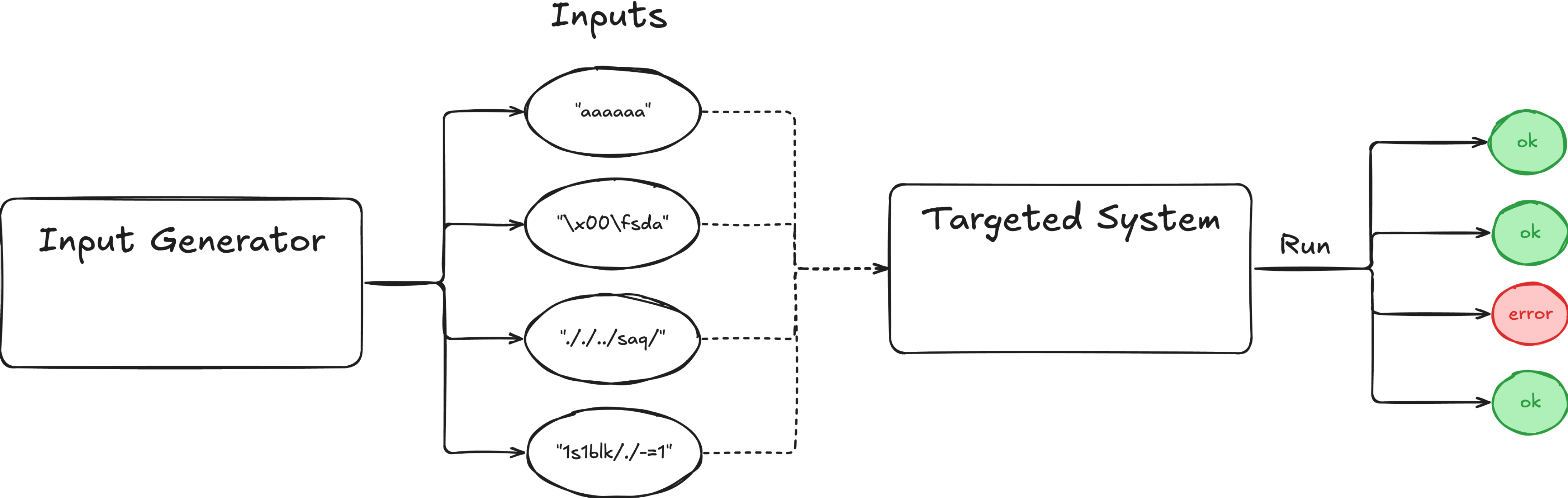
Differential Testing



Bloc Differential Testing



Fuzzing



Grammar-based Fuzzing

Regex --> Branch | Branch , '|', Regex

Branch --> " | Concat

Concat --> Modifier | Modifier, Concat

Modifier --> Pure | Pure, '?' | Pure, '*' | Pure, '+'

Pure --> `' | 'a' | 'b' | 'c' | '(' , ntRegex , ')'`

```
> 'c|c|.+(|)+c*|'
```

'a | b'

```
'a*c(bc+.)?.+|'
```

' | a? '

' | () + | '

' | c? | a?ab? | . '

'b*|a'

' | a? | | '

' .+a+c? '

'ac+b+ '

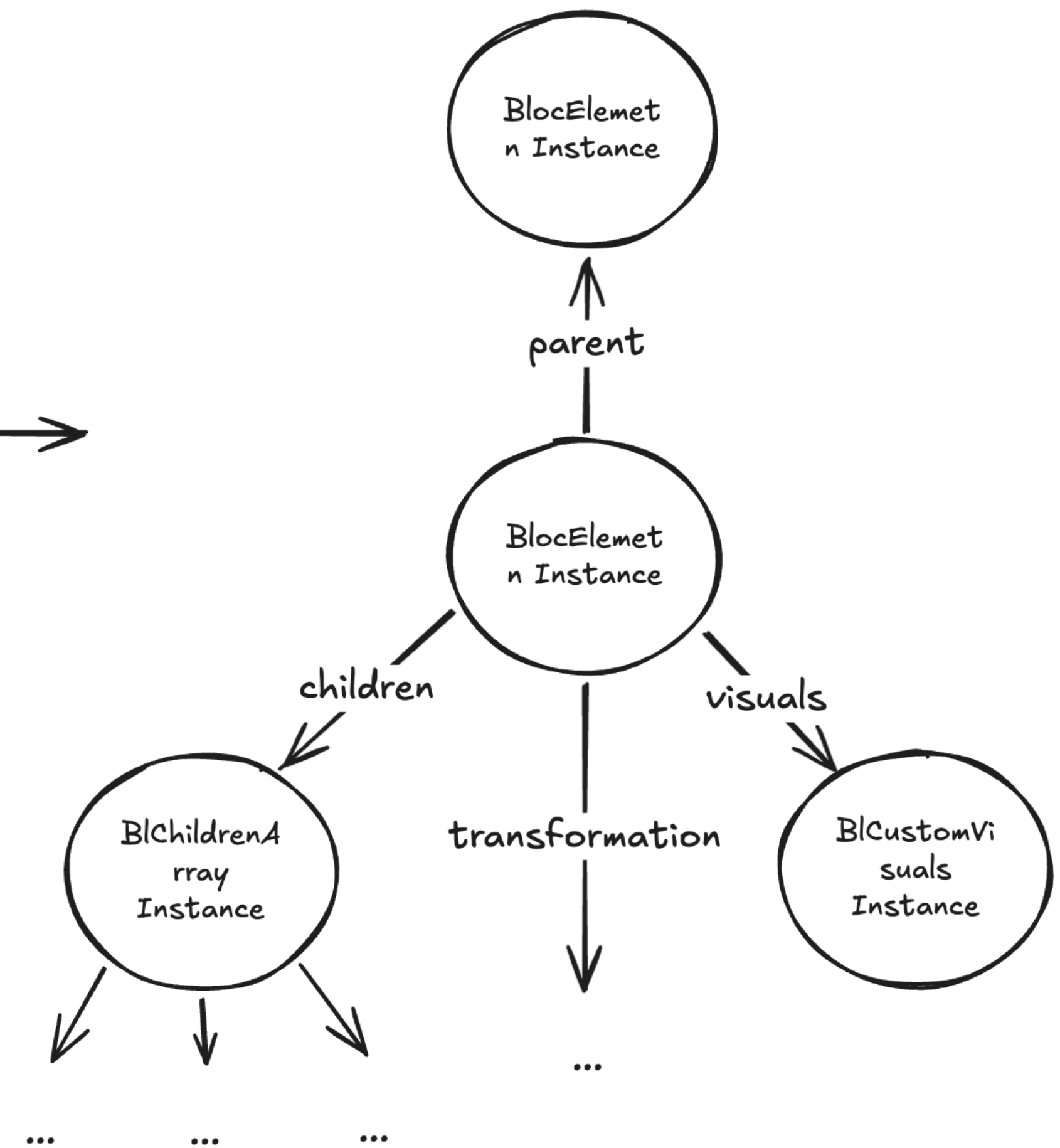
Objects Generation Grammars-based

```

BlocElement -> {
  children: BlChildrenArray,
  parent: BlocElement,
  visuals: (BlCustomVisuals | BlDefaultVisuals),
  transformation: ...
  ...
}

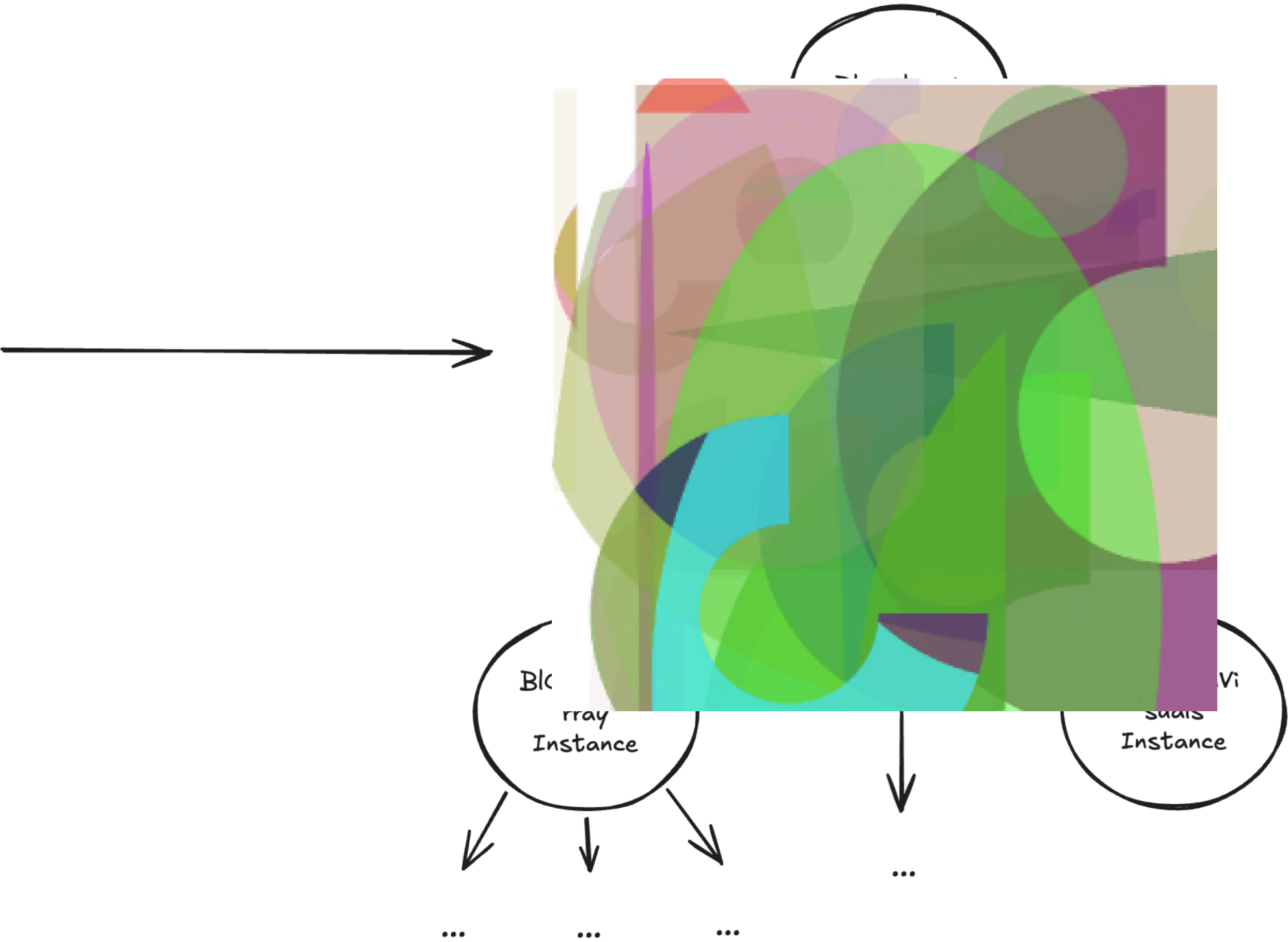
BlChildrenArray -> ...
...

```



Objects Generation Grammars-based

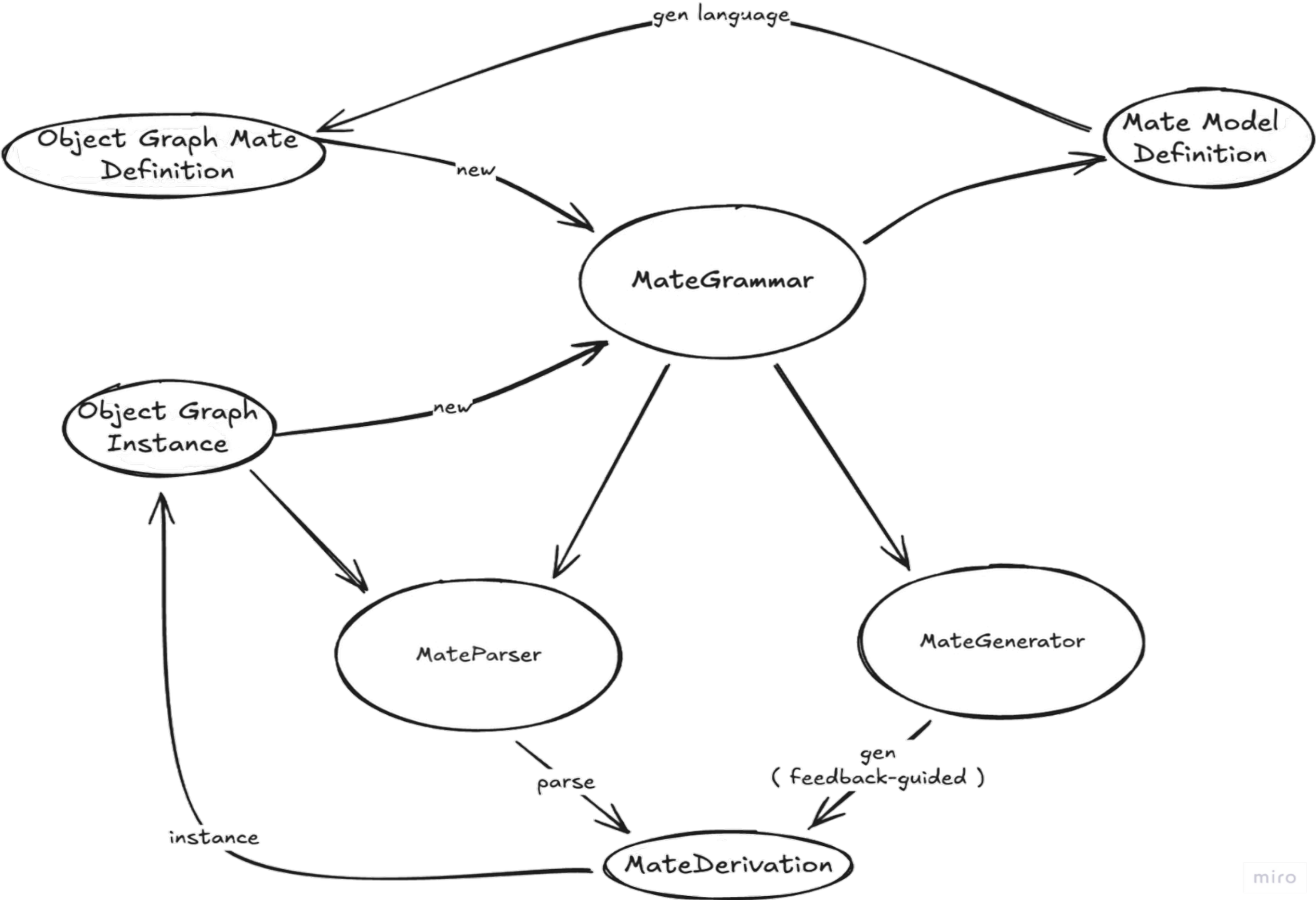
```
BlocElement -> {  
  children: BlChildrenArray,  
  parent: BlocElement,  
  visuals: (BlCustomVisuals | BlDefaultVisuals),  
  transformation: ...  
  ...  
}  
BlChildrenArray -> ...  
...
```



Object Graph Language and Object Graph Fuzzer



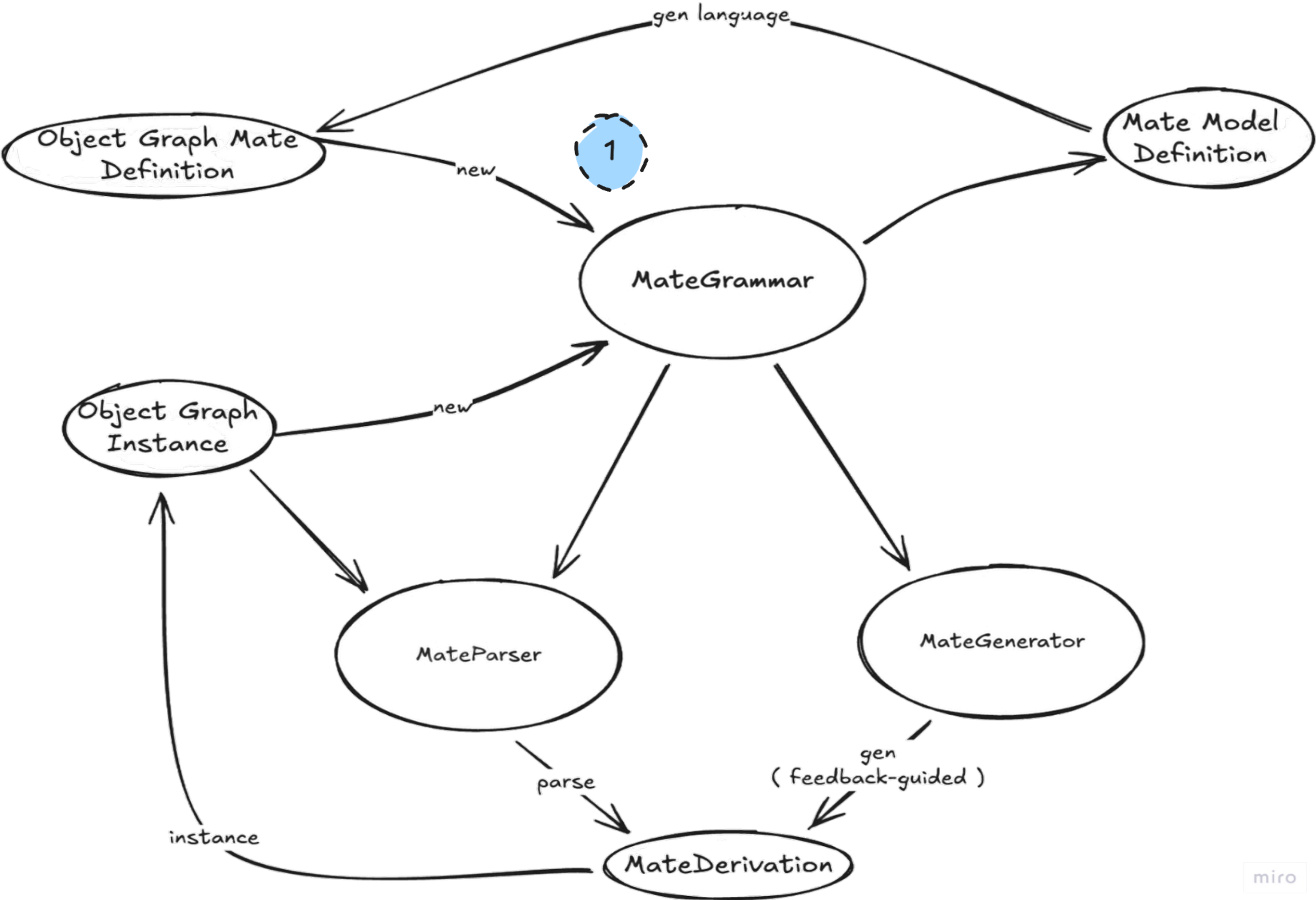
Mate Paths



miro



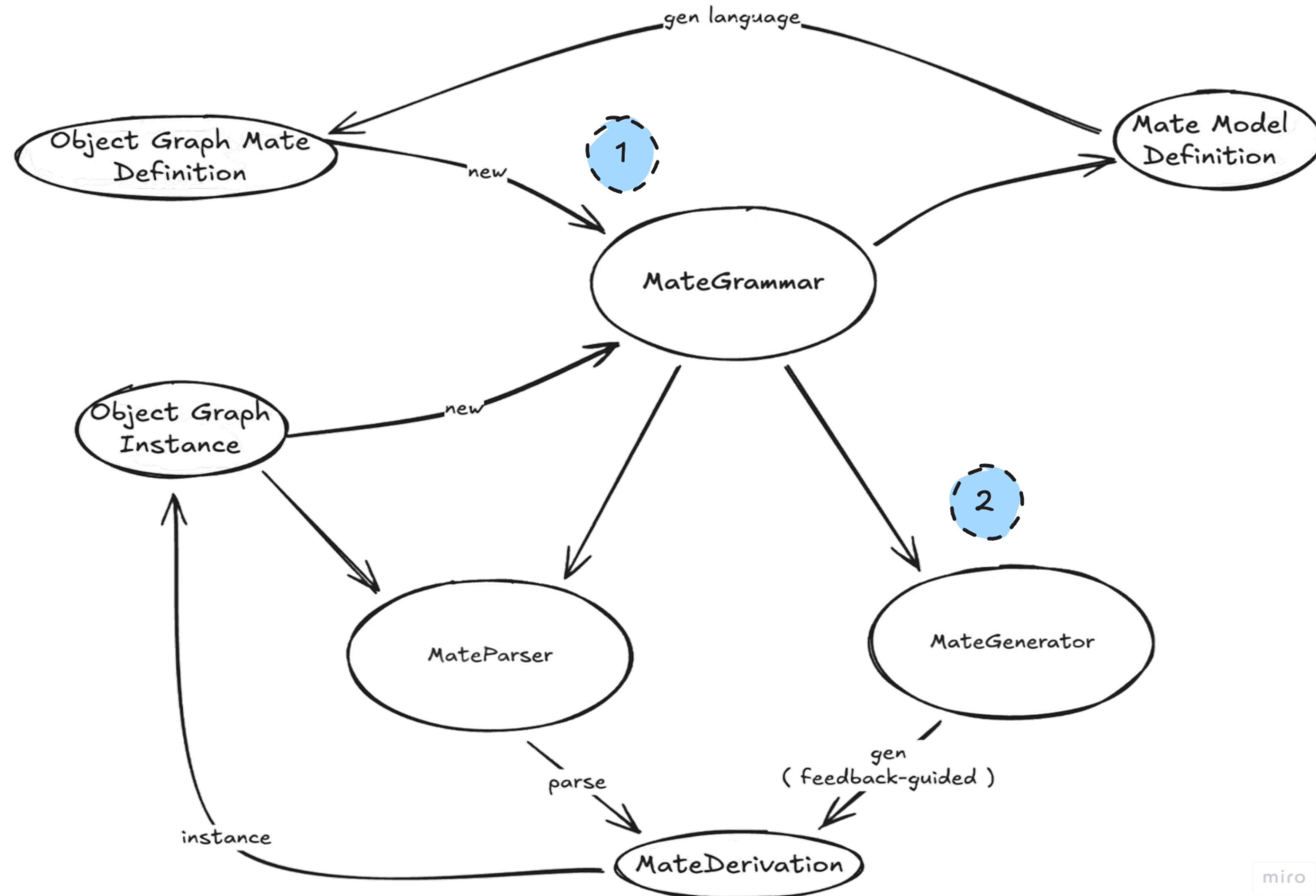
Mate OG Generation



miro



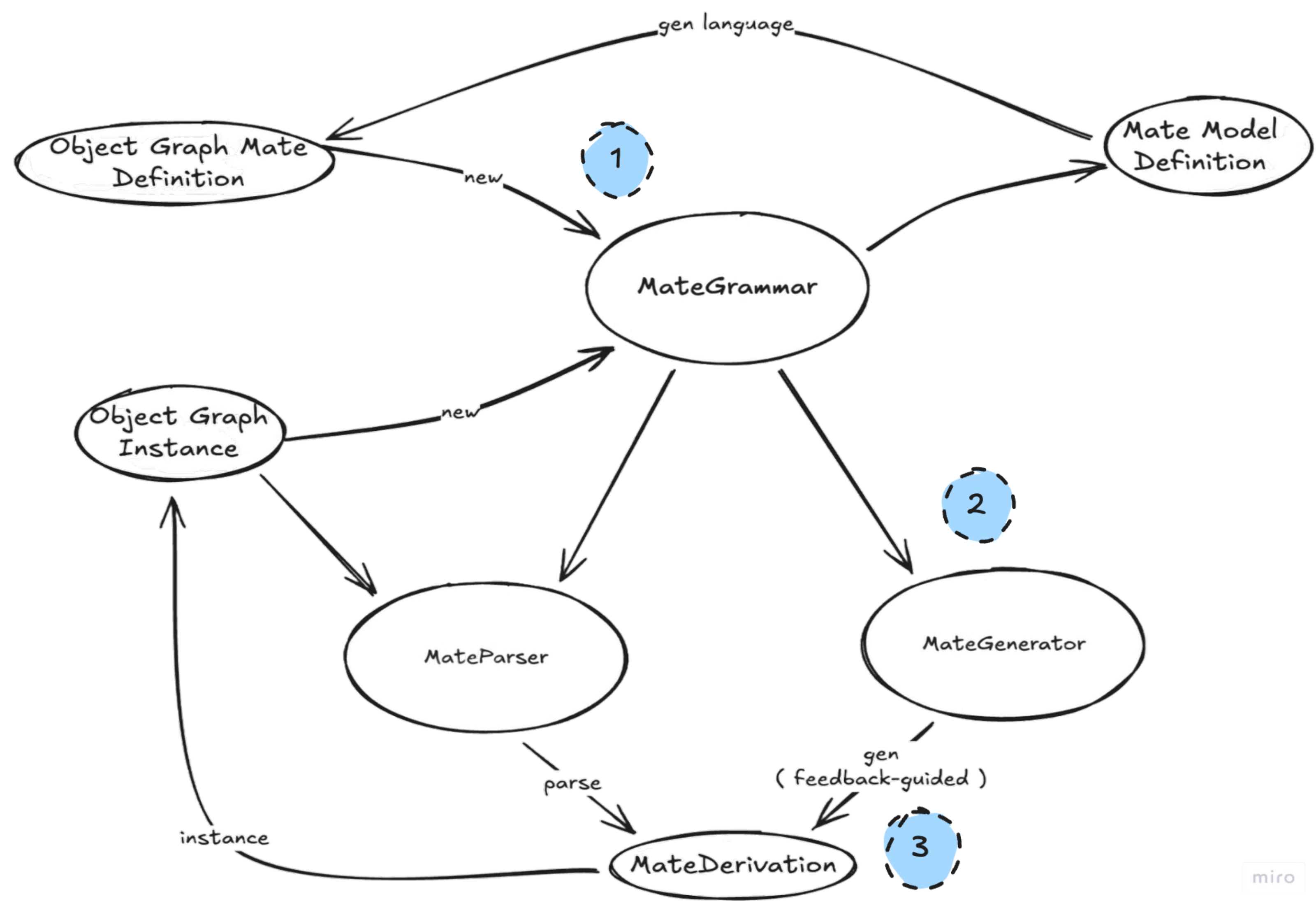
Mate OG Generation



miro



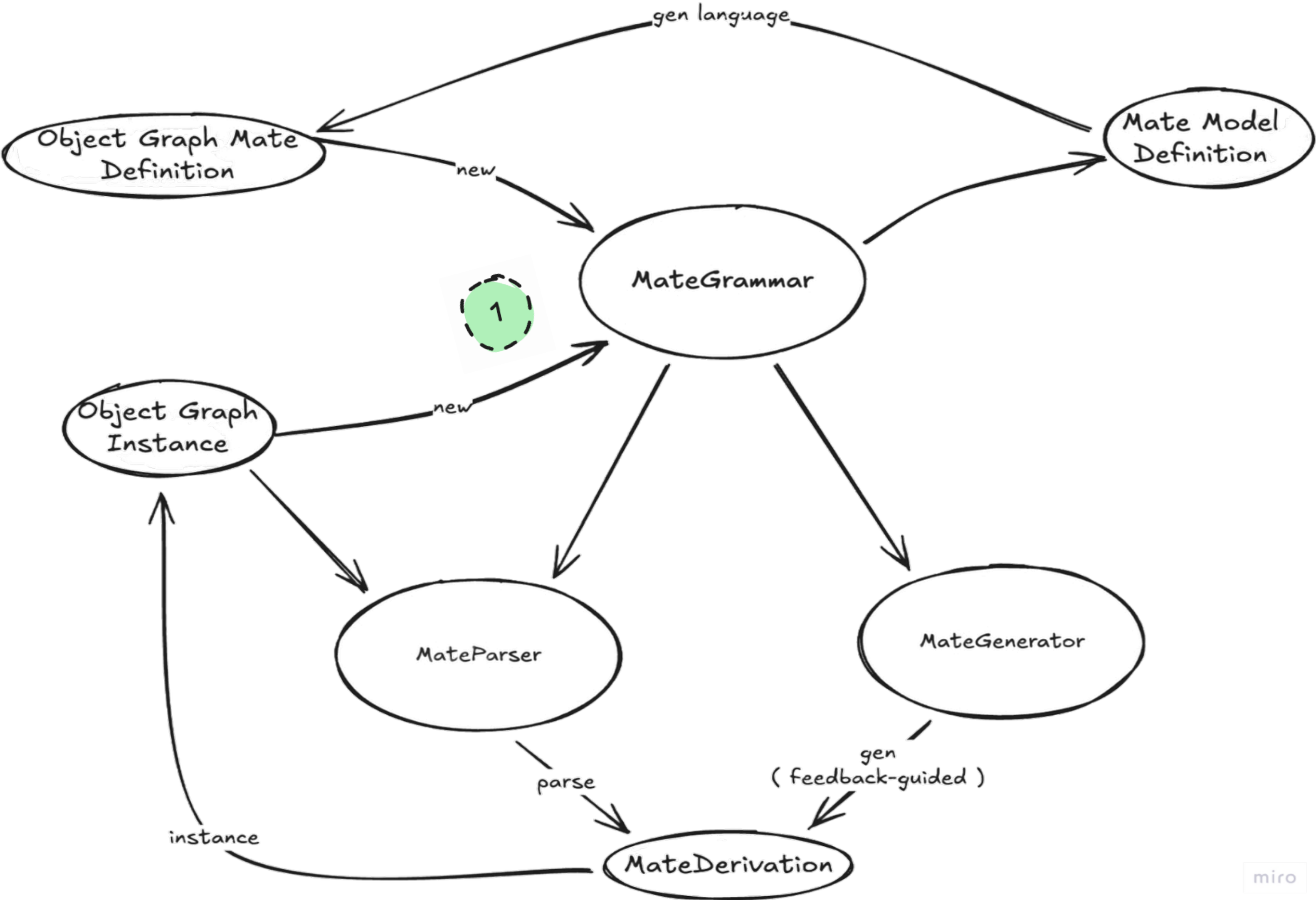
Mate OG Generation



miro



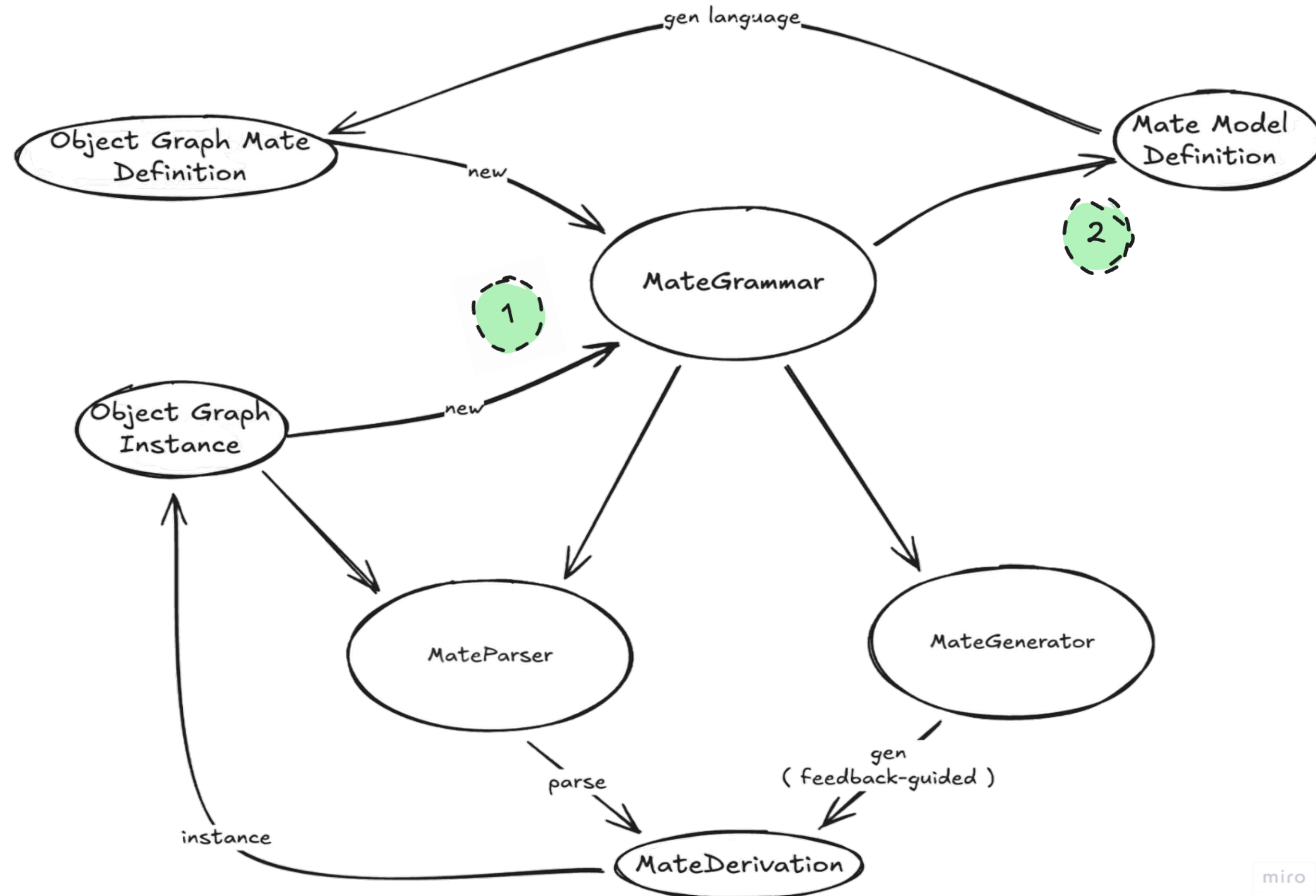
Mate Grammar Generation



miro



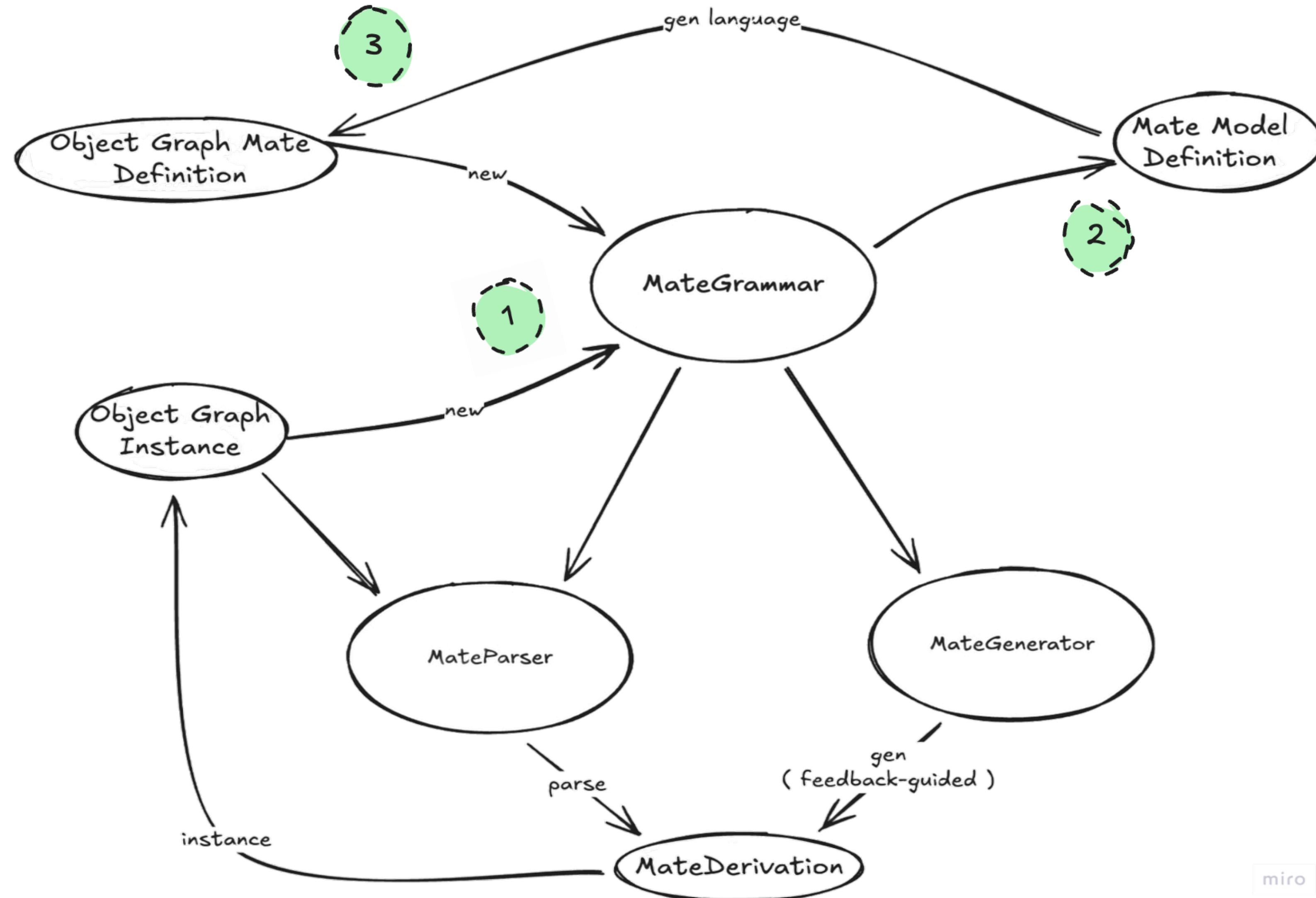
Mate Grammar Generation



miro



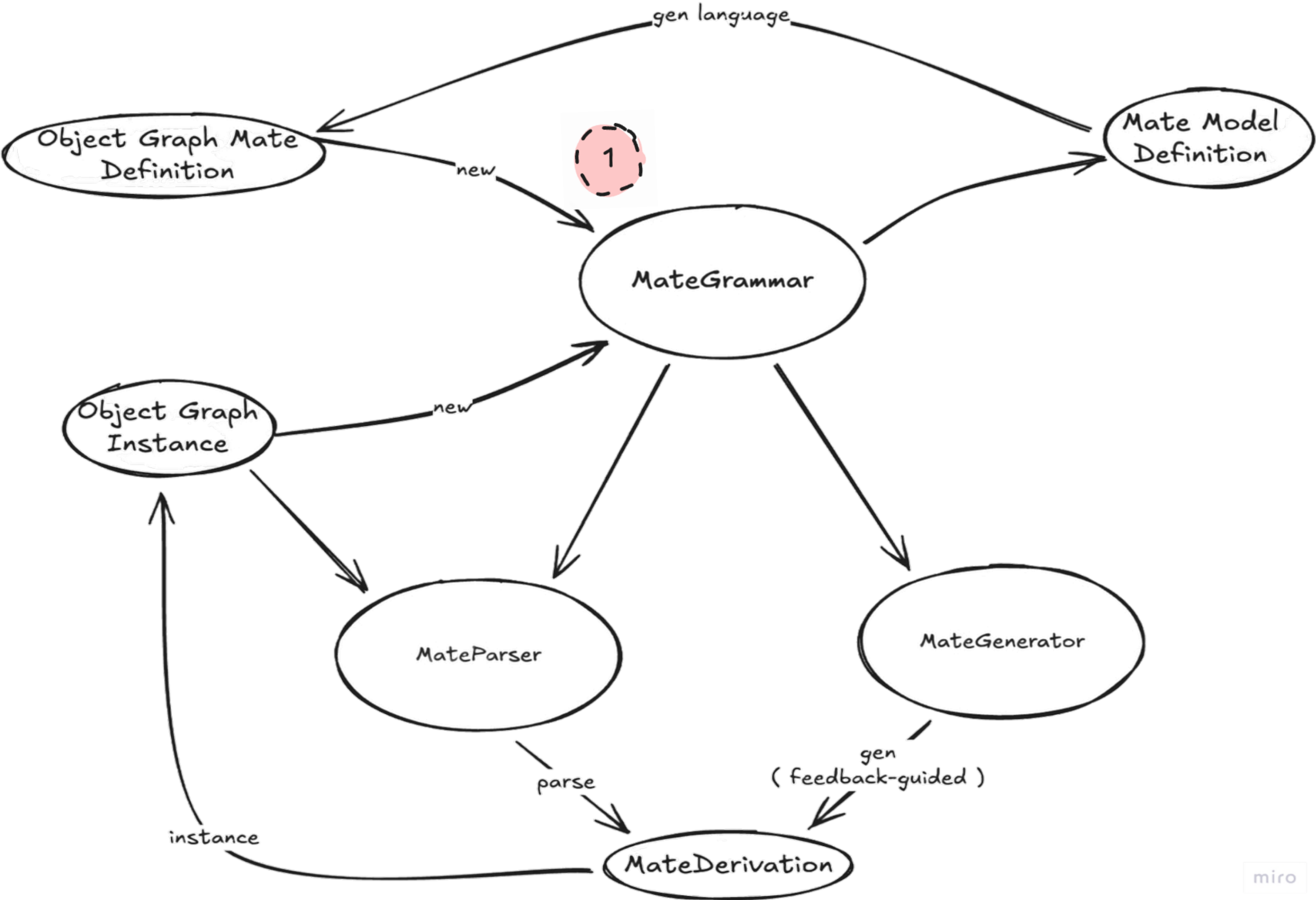
Mate Grammar Generation



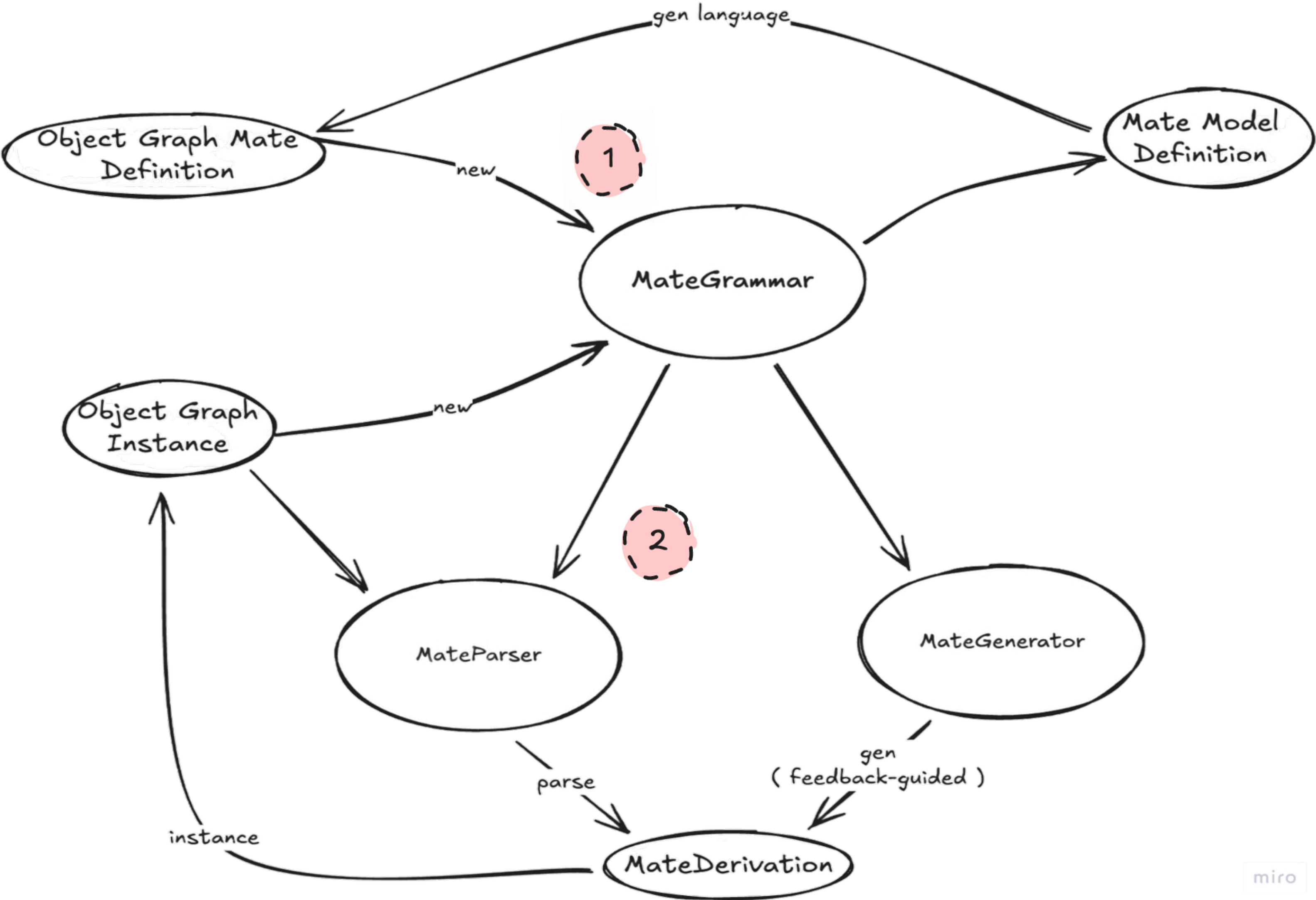
miro



Mate OG Validation



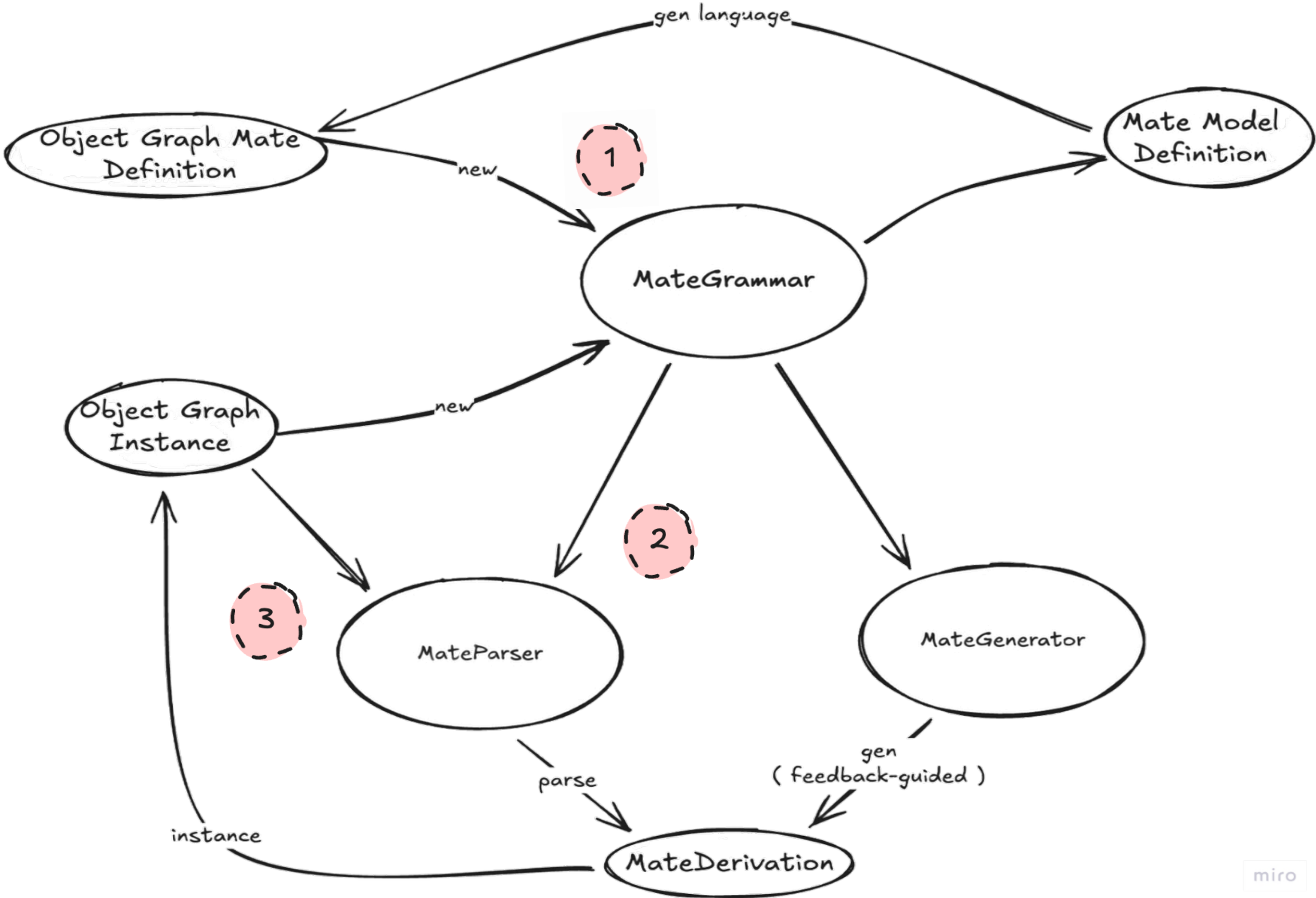
Mate OG Validation



miro



Mate OG Validation



Bloc Grammar

```
BlElement -> {  
    children: BlChildrenArray,  
    visuals: (BlCustomVisuals | BlDefaultVisuals)  
}
```

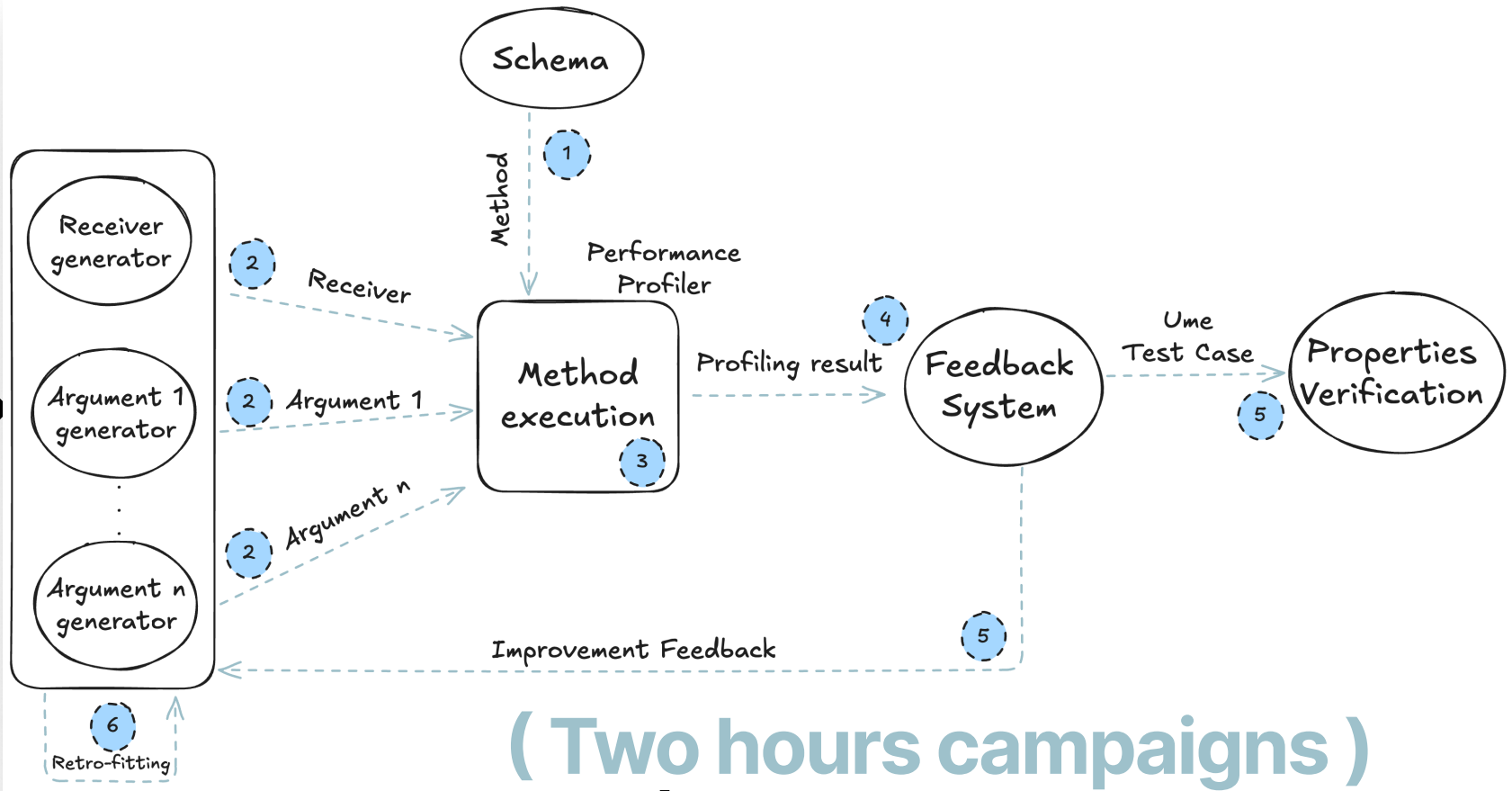
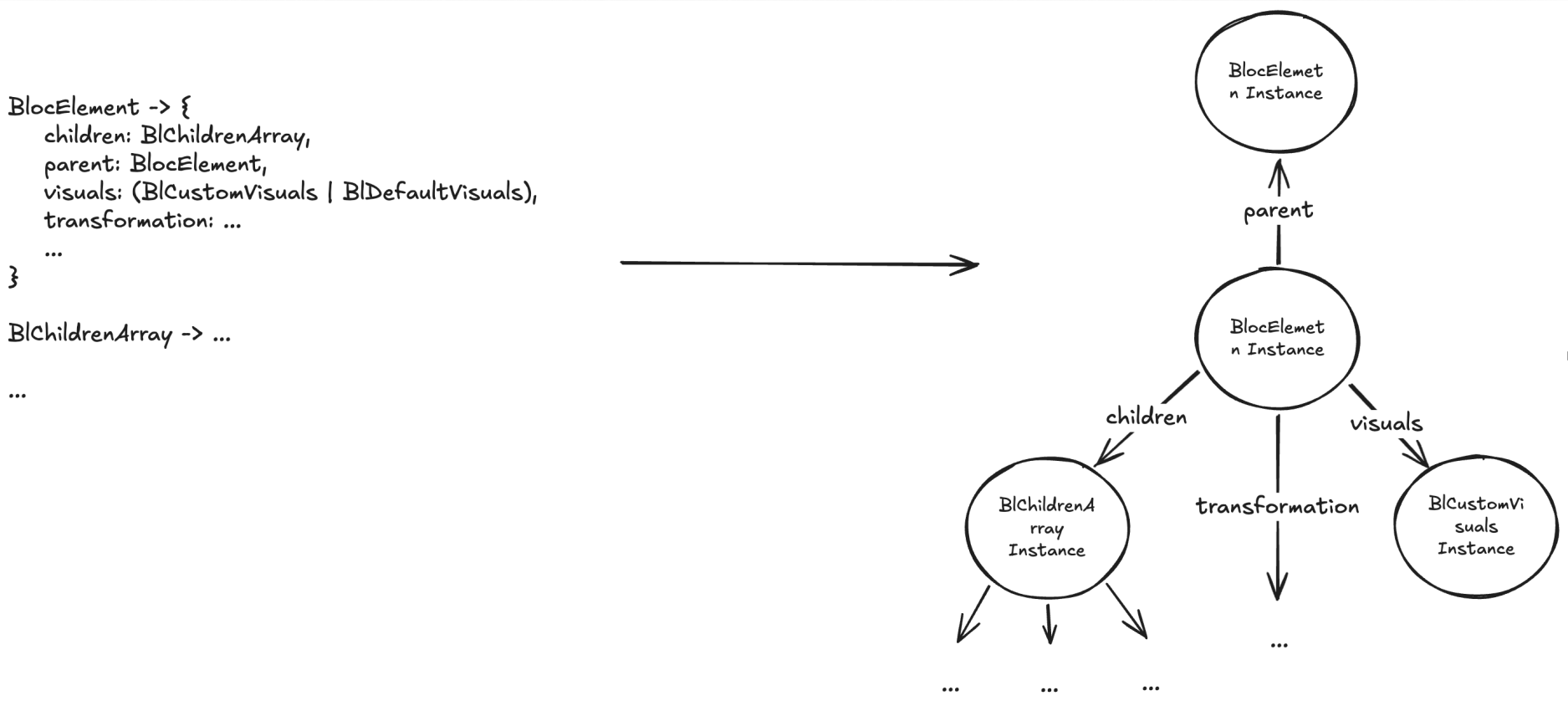
```
BlChildrenArray -> {  
    array: { type: [ BlElement ], sizeBetween: 0 and: 10 }  
}
```

```
BlCustomVisuals -> { ... }
```

```
BlLayoutExactResizer -> {  
    size: { type: SmallFloat64, between: 100 and: 1000 }  
}
```

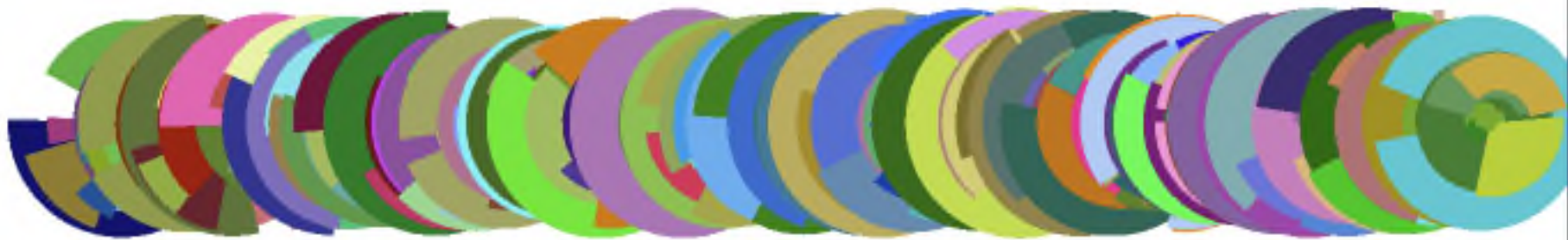
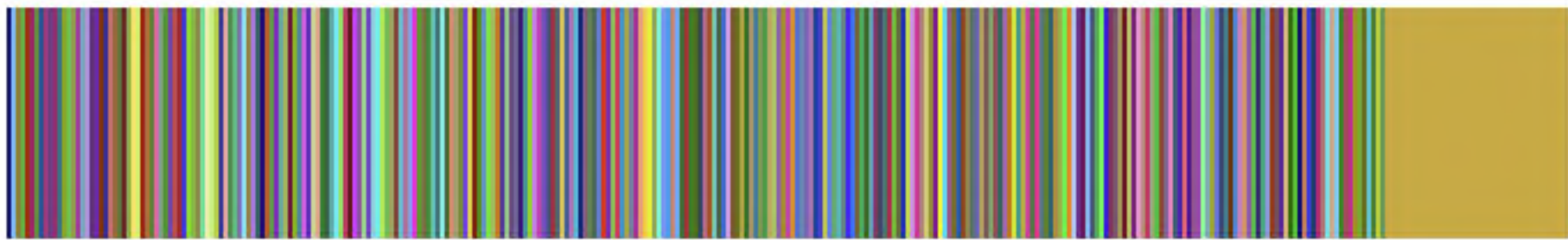
```
BlTriangleGeometry -> { ... }
```

Solution Architecture



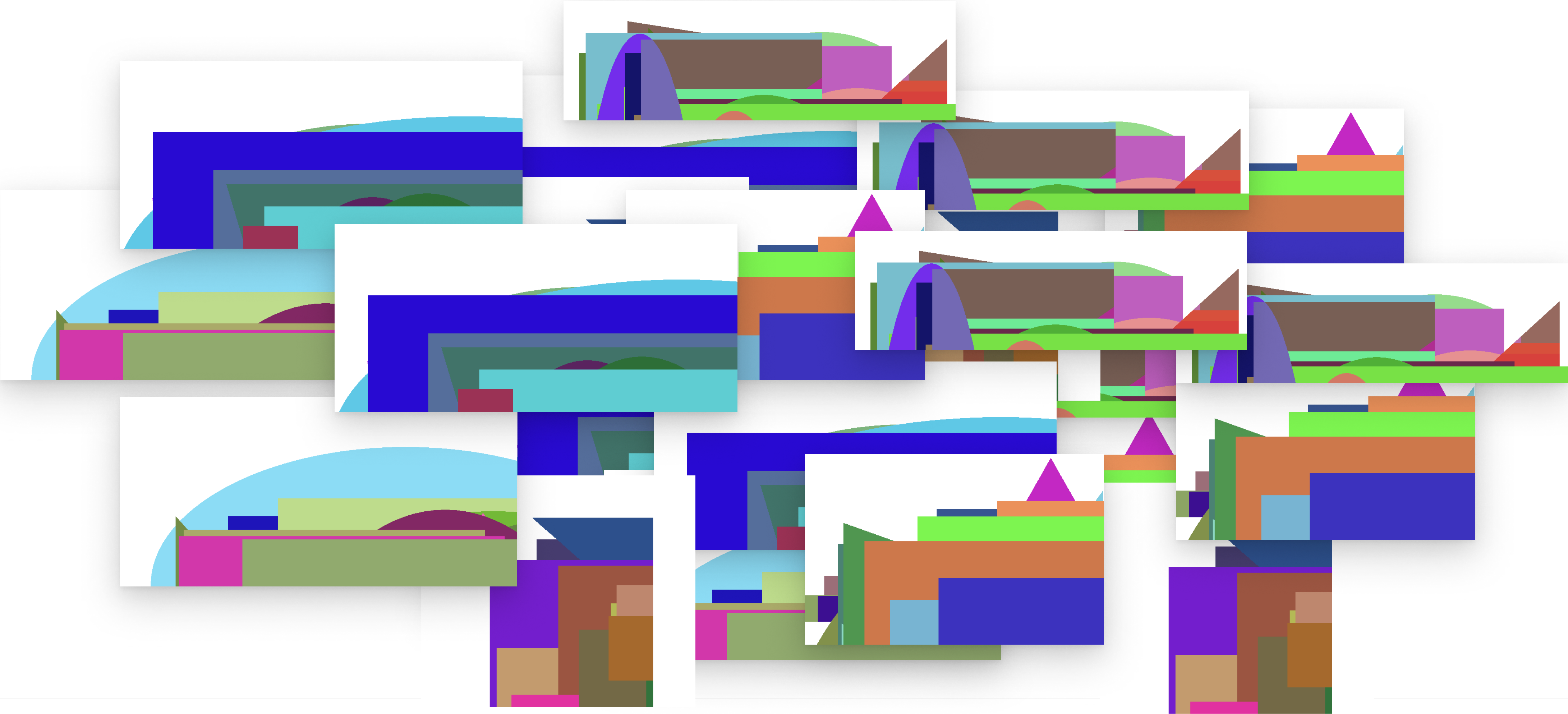
(Two hours campaigns)

Differential Testing



Metric measurement

So...?

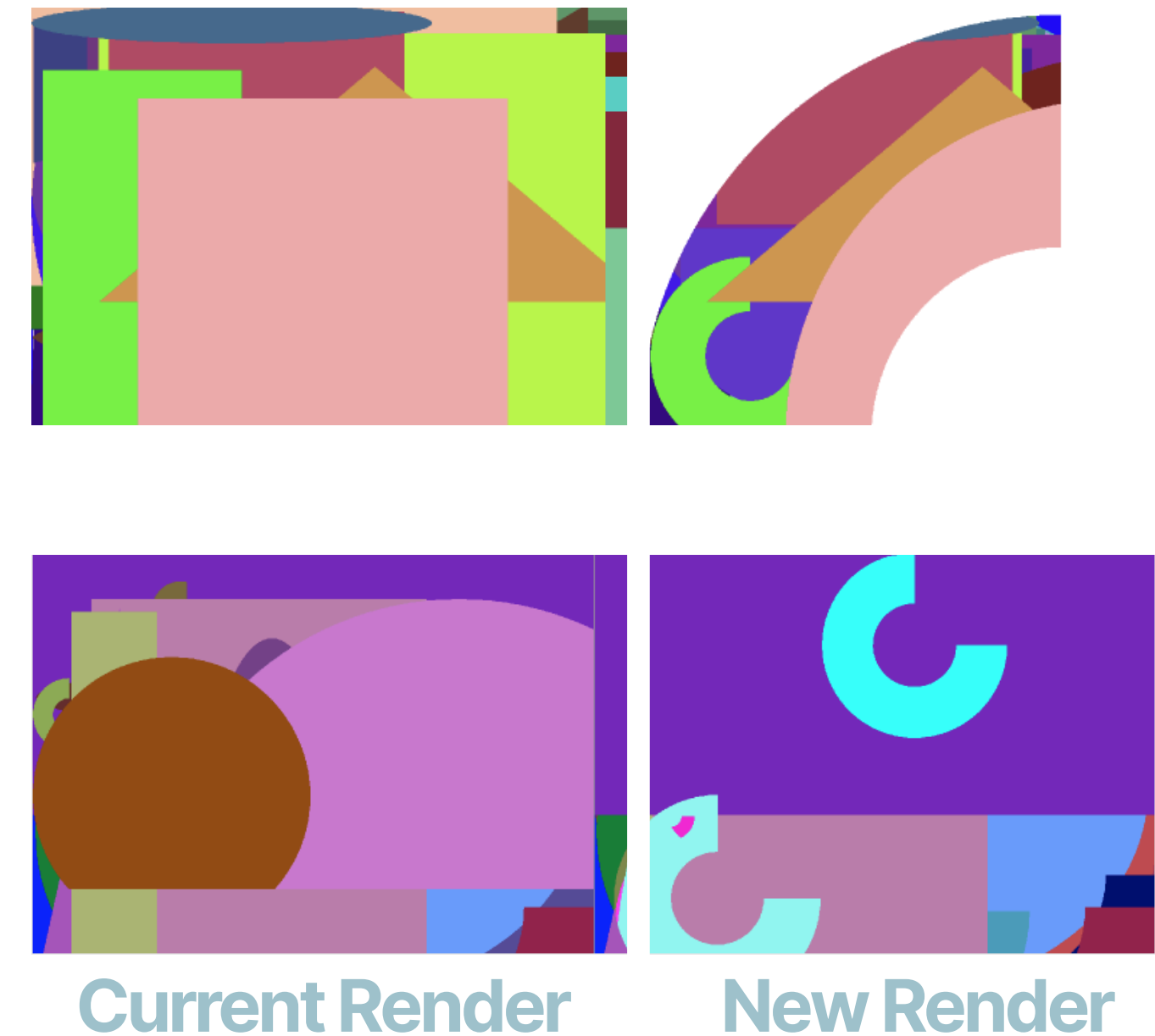


Visual Inconsistencies - Pixel-by-pixel

6,658 Generated DOM graphs

19 Outliers

98% Visual differences



FPS Drops

3,742 Generated DOM graphs

17 Outliers

Reaching frames with **less than 60 pfs**

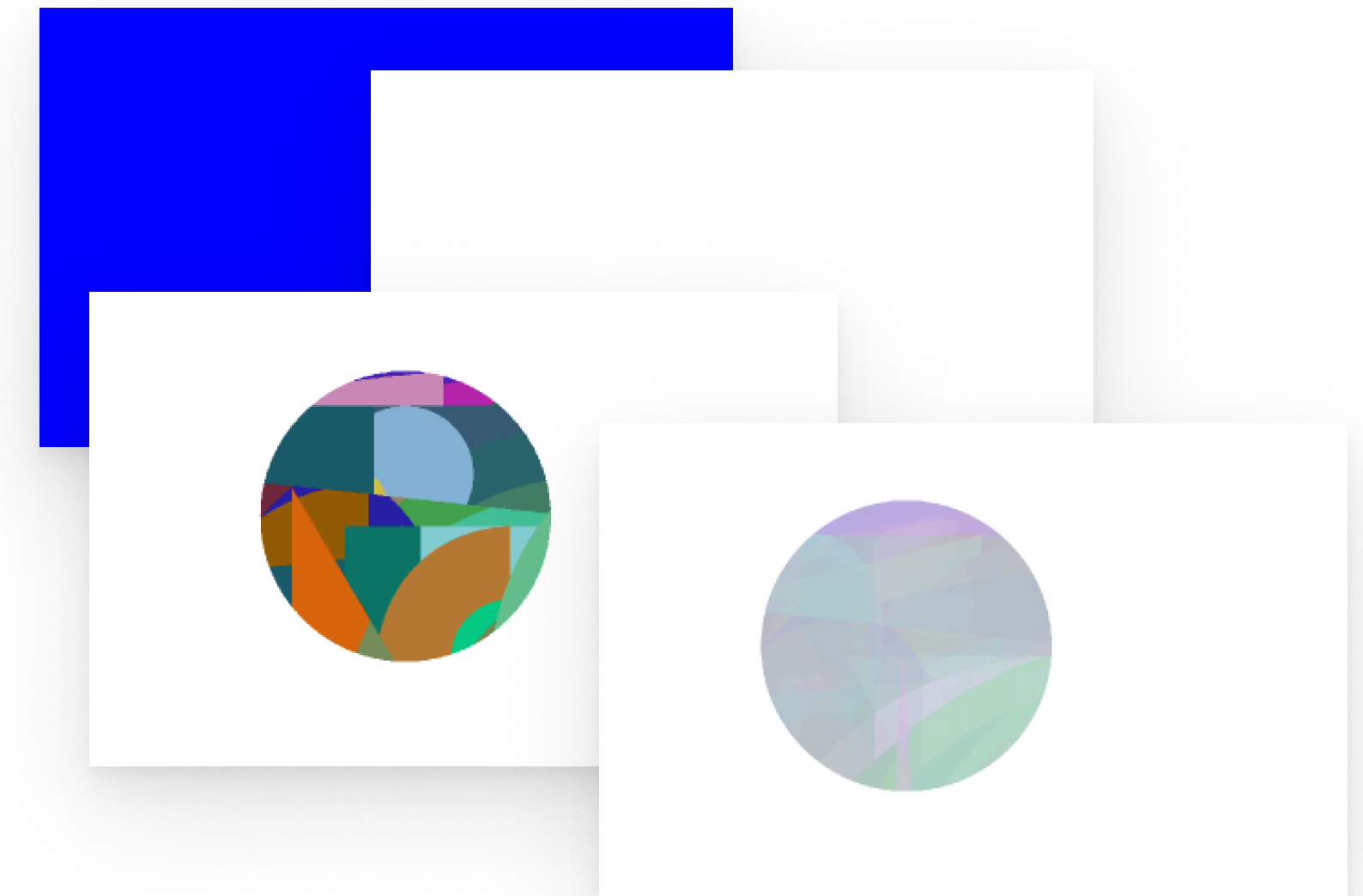


Feature Diversity Errors

62,202 Generated DOM graphs

7 Outliers

Reaching **101** methods and **137** nodes



What we have found so far

- **BIAnnulusSectorGeometry** behavior not implemented
- **Z-order** not handled correctly
- Element **opacity** not handled correctly
- Not handling complex interactions between **layering and z-order**

(We already fixed most of them)

What's next ?

What's next ?

Extending to the remaining challenges

- Memory Usage
- Garbage Collector Calls
- Object life cycles
- Dirty Areas
- Optimization

What's next ?

Extending to the remaining challenges

- Memory Usage
- Garbage Collector Calls
- Object life cycles
- Dirty Areas
- Optimization

Bloc's CI workflow

What's next ?

Extending to the remaining challenges

- Memory Usage
- Garbage Collector Calls
- Object life cycles
- Dirty Areas
- Optimization

Bloc's CI workflow

Experimenting with different platforms and architectures

- Windows
- Linux

Recap

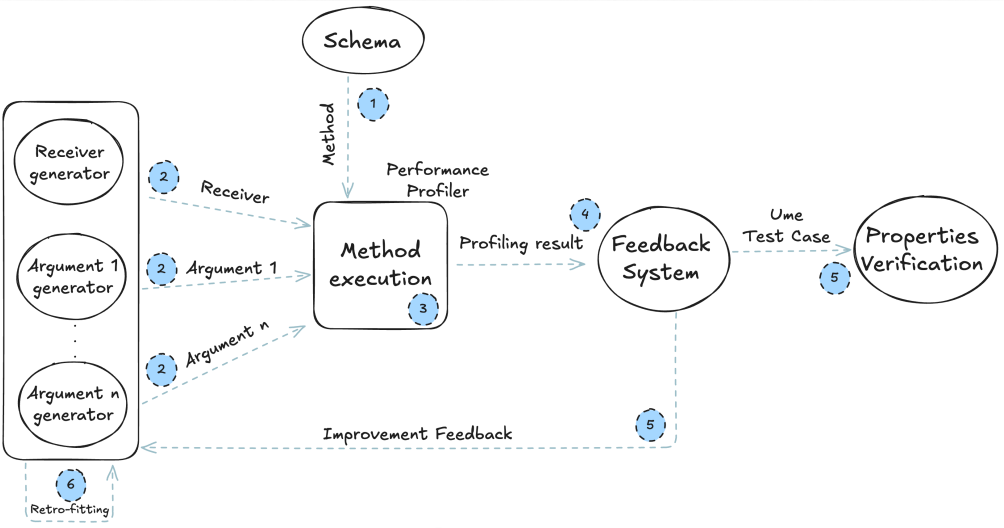
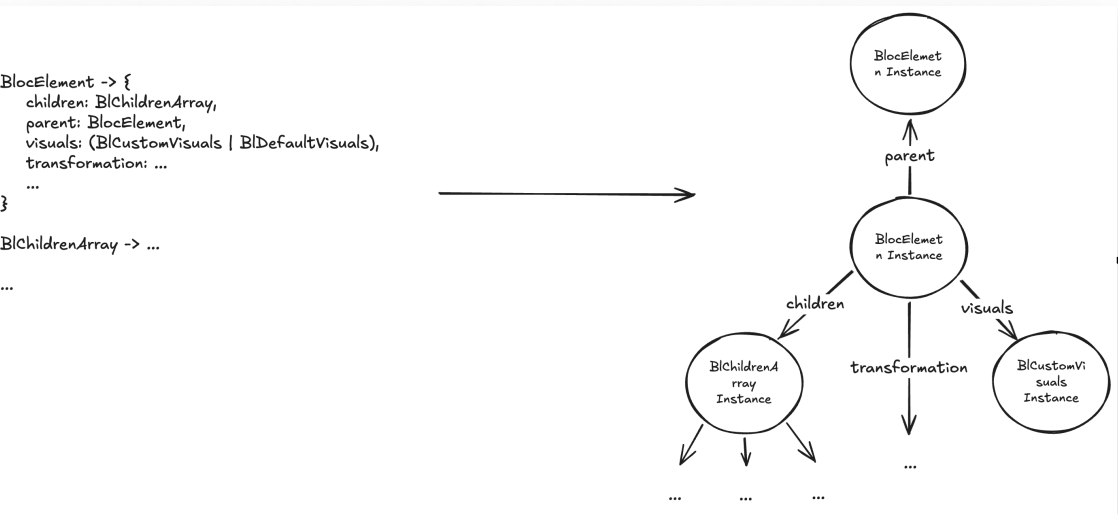
```
BLElement -> {
  children: B1ChildrenArray,
  visuals: (B1CustomVisuals | B1DefaultVisuals)
}

B1ChildrenArray -> {
  array: { type: [ BLElement ], sizeBetween: 0 and: 10 }
}

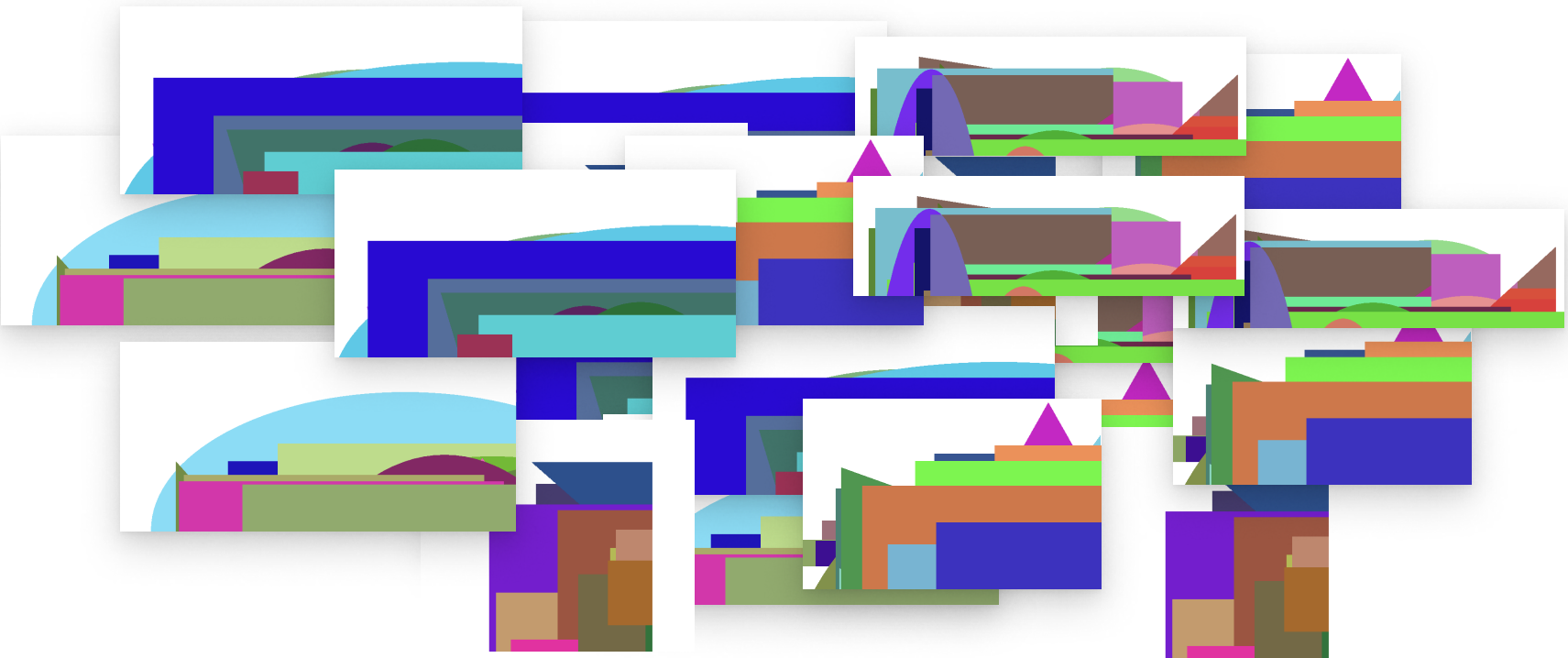
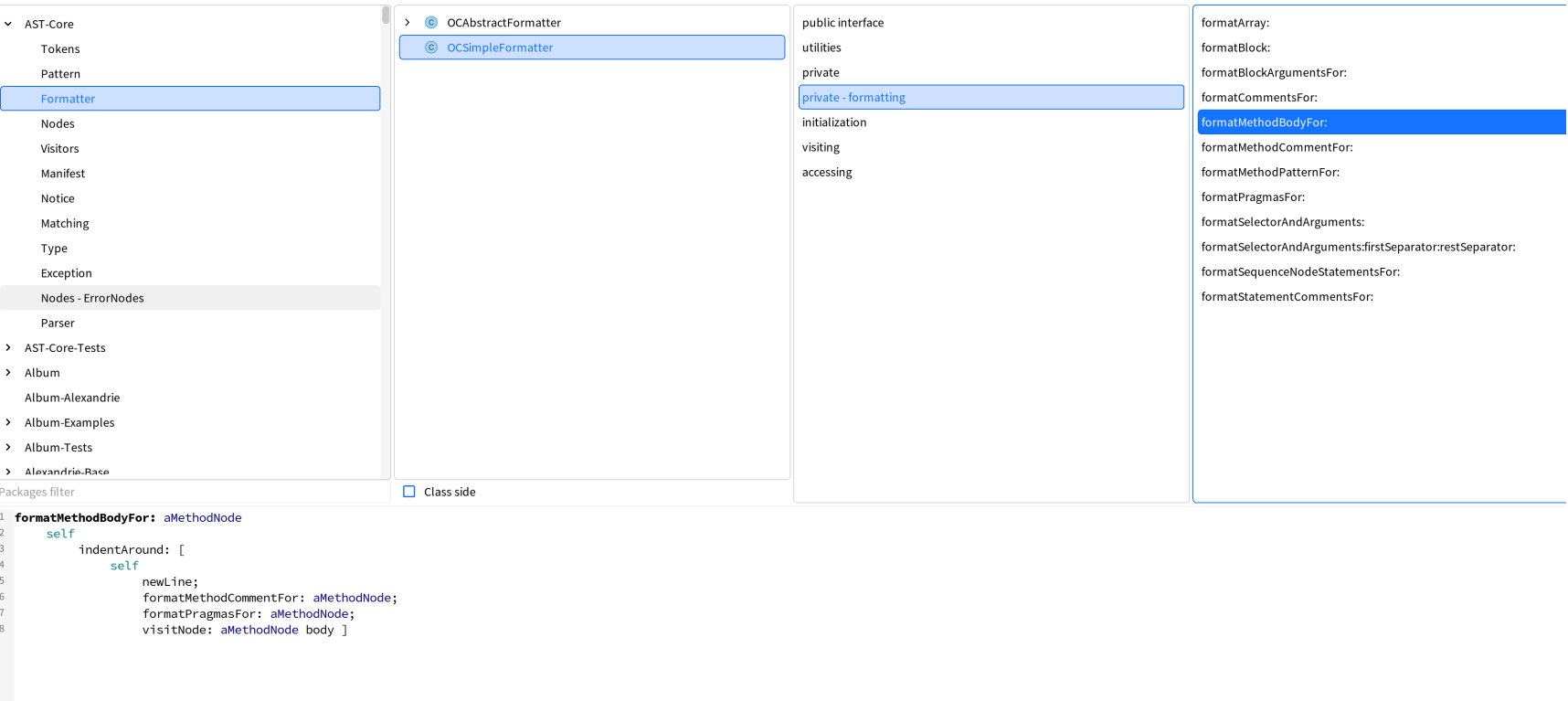
B1CustomVisuals -> { ... }

B1LayoutExactResizer -> {
  size: { type: SmallFloat64, between: 100 and: 1000 }
}

B1TriangleGeometry -> { ... }
```



Differential Testing



federico-javier.lochbaum@inria.fr