

深耕 Flutter Compiler 体系结构

——从理论到实践为底层赋能

刘丰恺 (Flutter Infra)

目录

1. Compiler Infra 基础知识

1. 前、中、后端的基础分类
2. 各部分能做哪些事情、具体优化手段

2. Dart Compiler 编译体系介绍

1. Dart Compiler 包含哪些组成部分
2. Dart Compiler 处理流程
3. 各组成部分的优化手段

3. 现有案例分享

1. Auto Jsonify
2. Dart Version Control

4. For Future 未来规划

Compiler Infra 基础知识



Compiler 基本概念

- 什么是 Compiler ?
- 为什么研究 Compiler ?
- Compiler 的基本原则?

什么是 Compiler

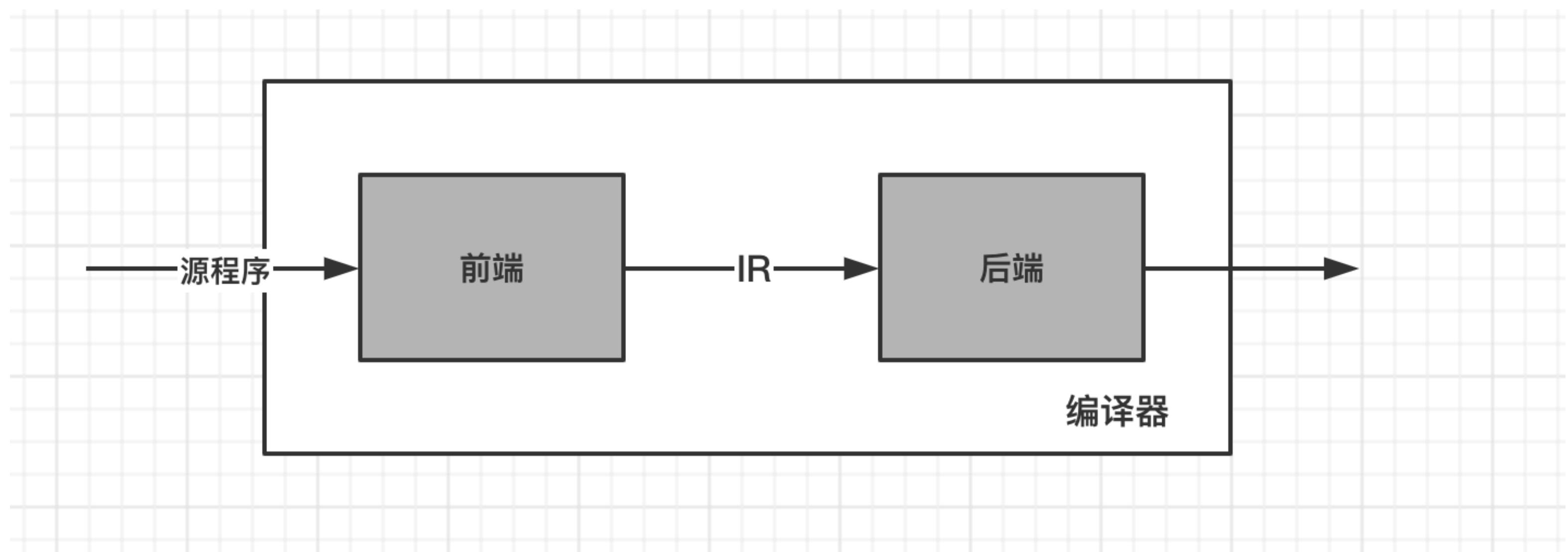
- 经典概念
- 源到源的转换器 | LLVM
- 解释器 or 混合体系

Compiler 结构

1. 早期：单盒结构、引入 IR

IR: intermediate representation

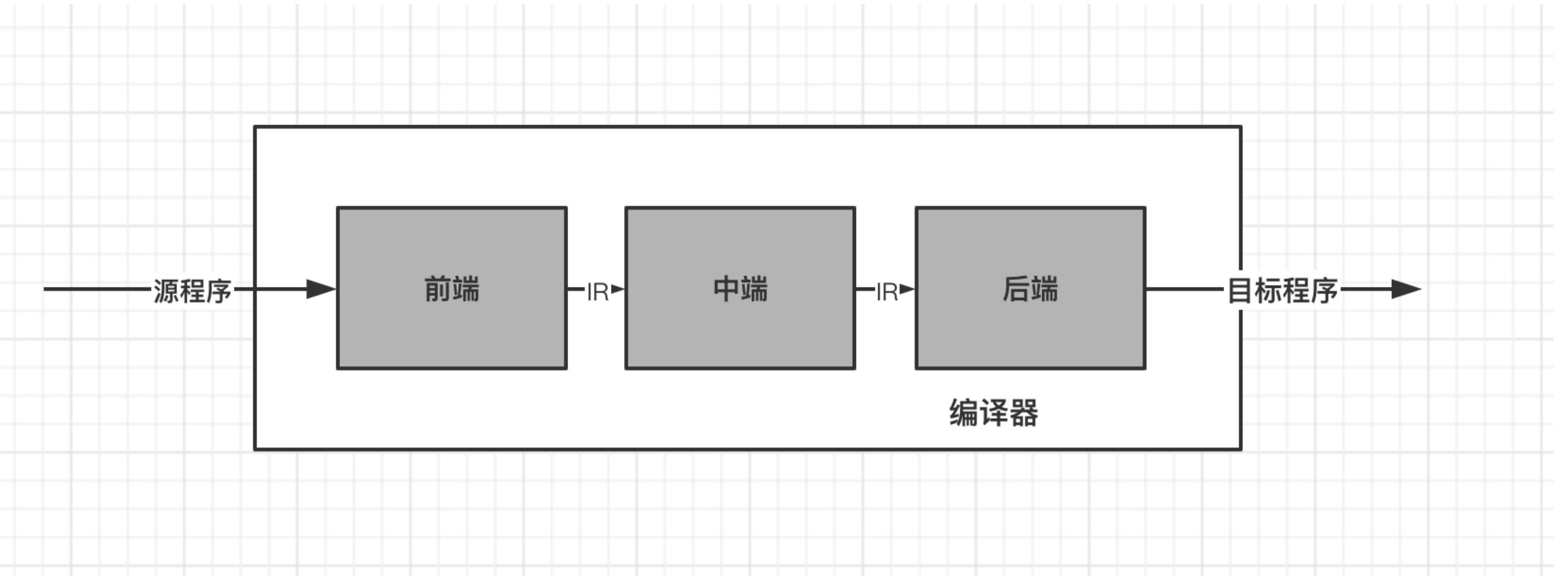
1. 根据需求不同产生的前端概念
2. 产生了 IR 相关概念



Compiler 结构

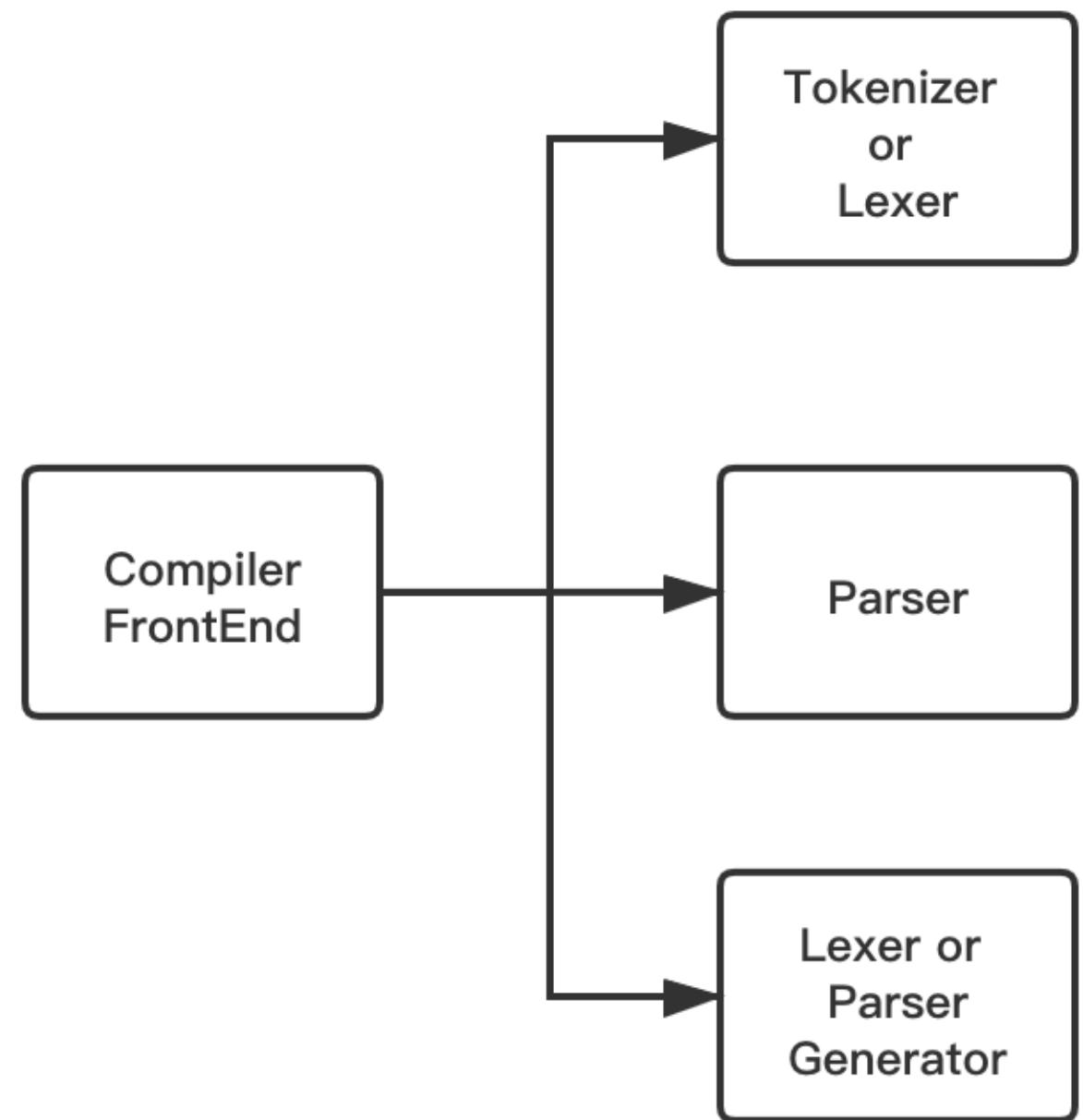
2. 经典：三段式结构、MultiPhase IR Pass

1. 优化着重于 IR 提出中端
2. 中端实现多 Phase 优化
3. IR 可能出现多次转换



1. 前端：读取源程序生成 IR
2. 中端：IR 转换、优化
3. 后端：CodeGen 到其他支持方案、直接运行

Compiler FrontEnd



1. Write Lexer Manually
2. Write Parser Manually
3. Lexer Parser CodeGen

Parser 语法分析 | 定义语法

Context-Free Grammar, Extend - BNF Rule

Context-Sensitive Grammar

1. 产生式

2. 终结符

3. 非终结符

1. := 产生式

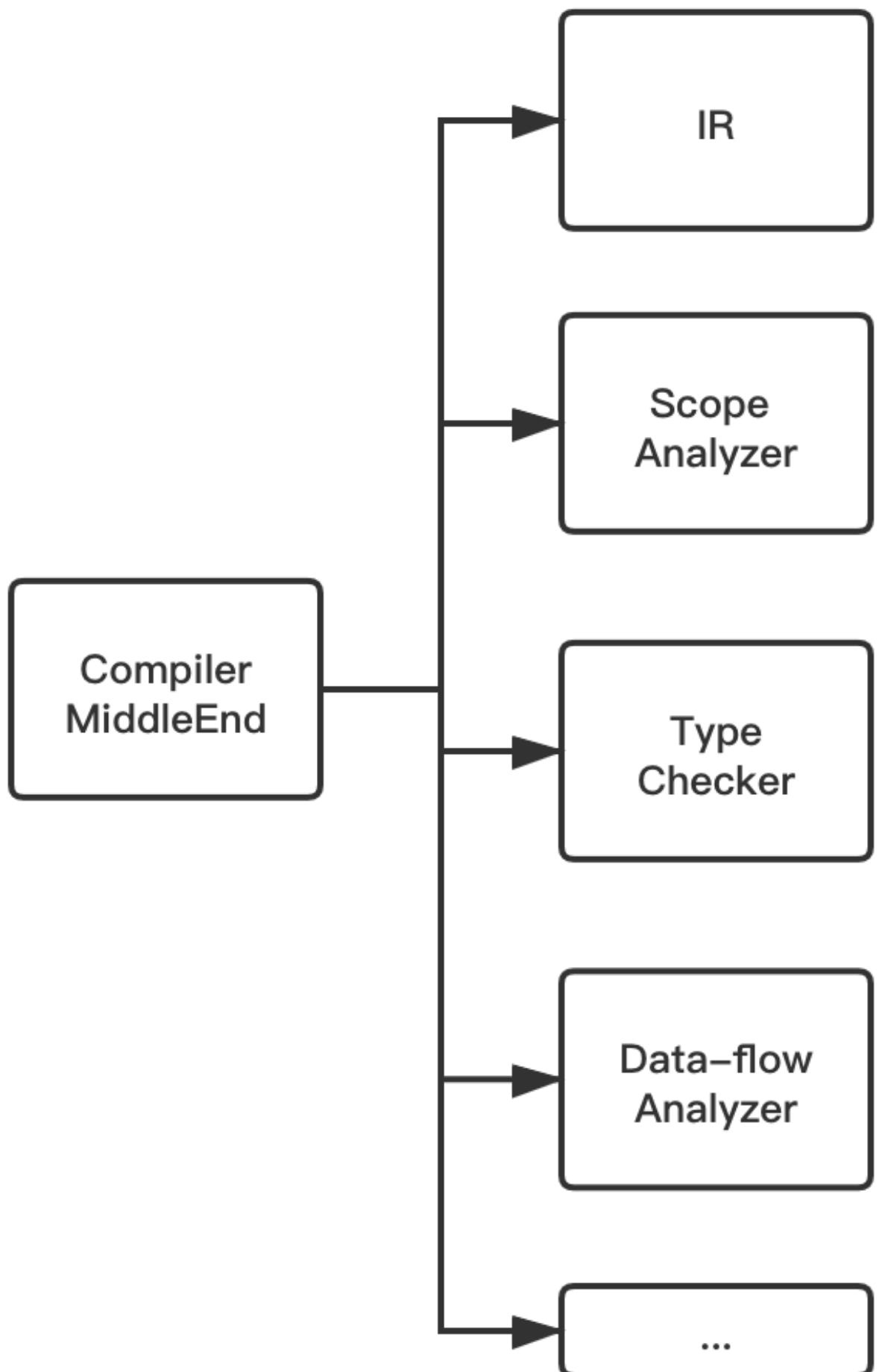
2. ' ' Literal 字面量

3. | 代表并列逻辑

4. * Kleene 闭包

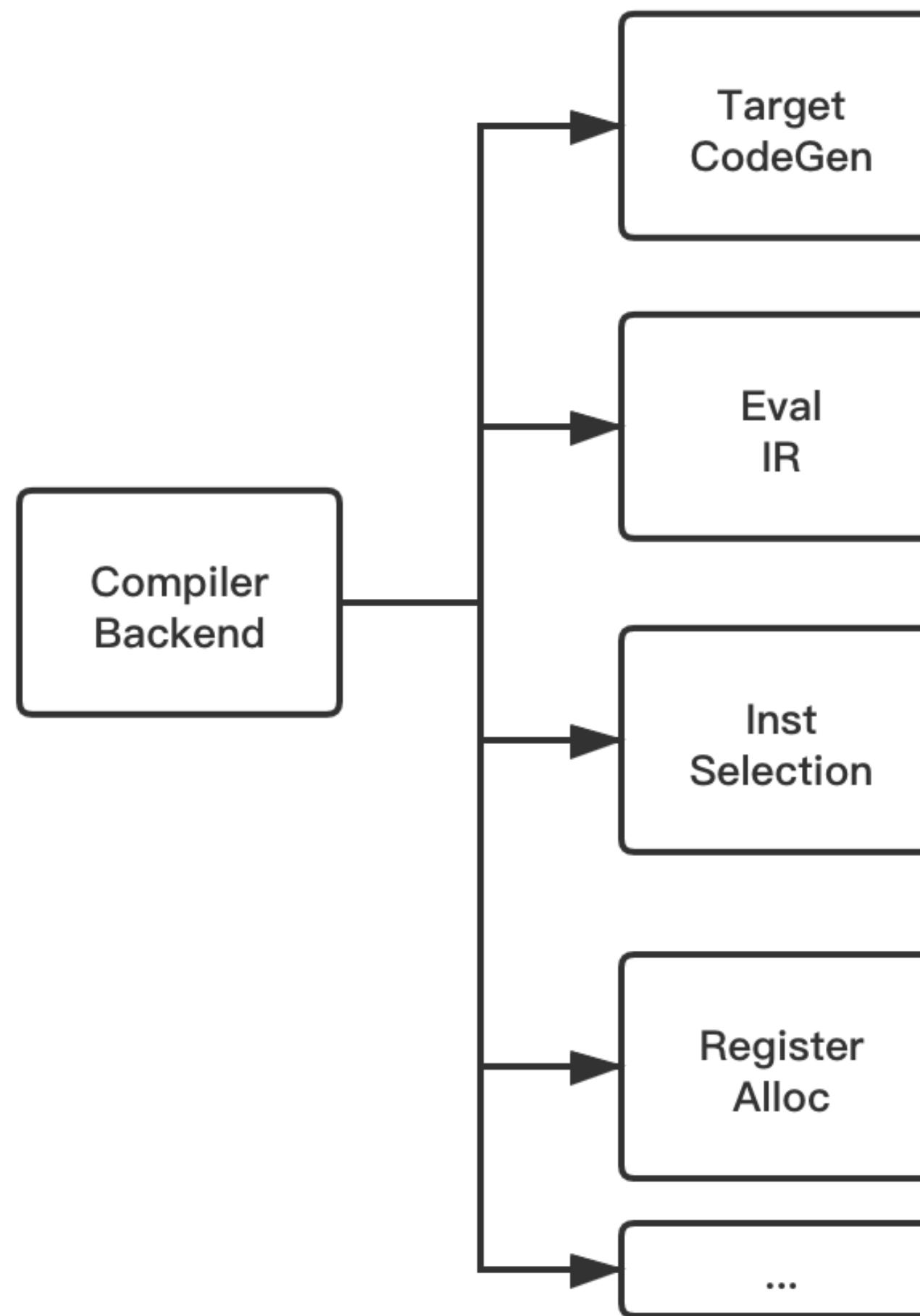
```
primary ::= '(' expr ')' | id | number | string | bool | null
factor  ::= primary | '-' primary | '!' primary # unary operators
expr    ::= factor operator factor # binary operators
simple   ::= expr ';'
block    ::= '{' simple* '}'
if       ::= 'if' '(' expr ')' block [ 'else' block ]
while   ::= 'while' '(' expr ')' block
stmt     ::= if | while | simple
program ::= stmt*
```

Compiler MidEnd



1. IR Generator: AST、DAG
2. Scope Analyzer : Free、Bound、Global、Cell
3. Type Checker: Type Inference
4. Data-Flow Analyzer: TFA

Compiler BackEnd



1. Eval IR: Evaluator

2. Backend CodeGen:

1. String template

2. To Other Lang: dart2js, todot

3. To Target: LLVM、JVM

字符串模板示例 <https://sourcegraph.com/github.com/lfkdk/JJust-Evaluator@adc92bd9650057b757a97cdc45eff725ac3dfa5/-/blob/src/main/java/com/lfkdk/justel/ast/function/OperatorExpr.java#L35:23>

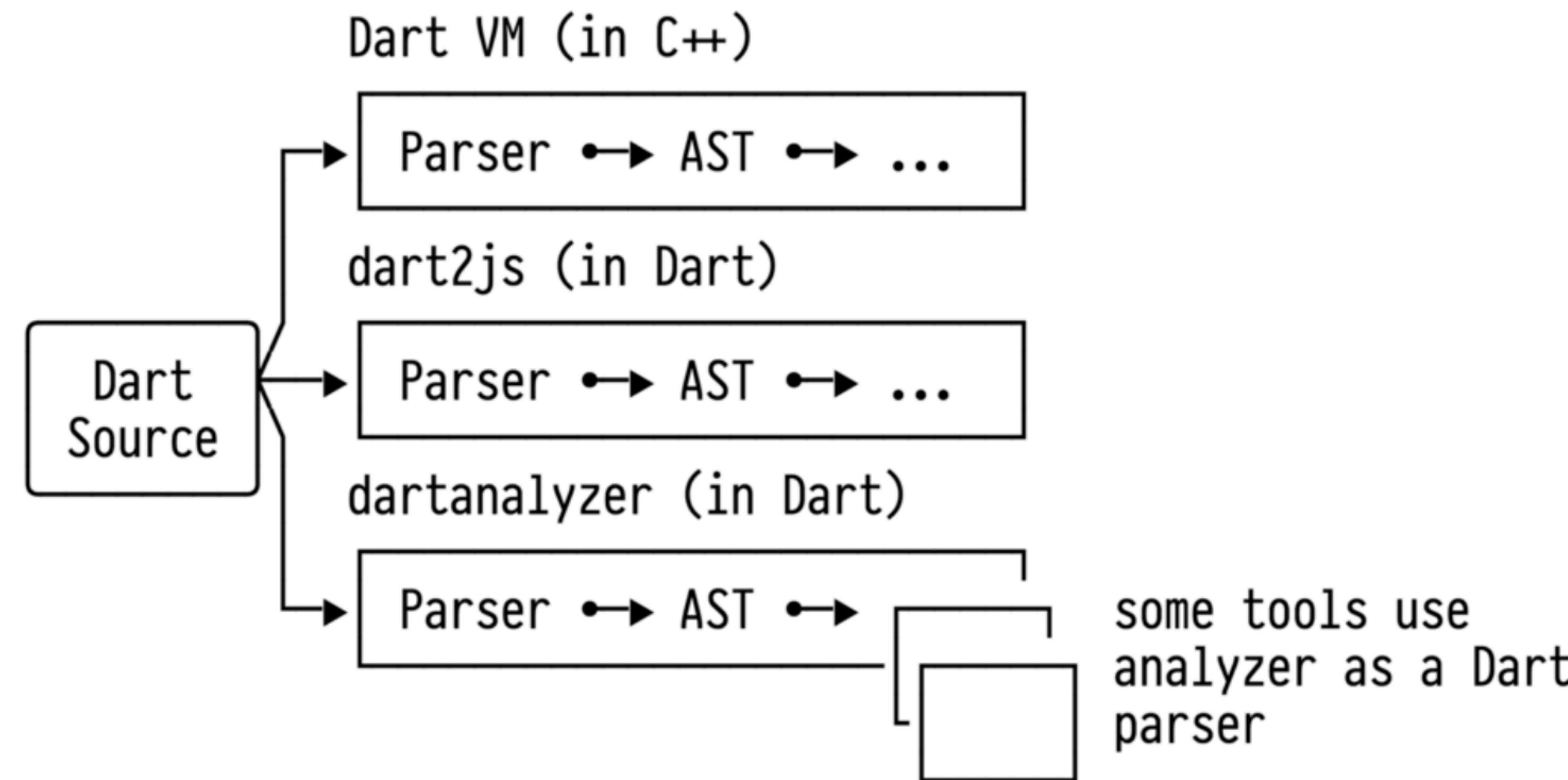
LLVM CodeGen https://github.com/lfkdk/HobbyScript/blob/lfkdk/add-llvm-backend/llvm-runner/src/visitor/llvm_gen_visitor.cpp#L69

Dot Codegen <https://github.com/lfkdk/HobbyScript/blob/lfkdk/add-llvm-backend/src/main/java/hobbyscript/Utils/PrintUtils.java#L101>

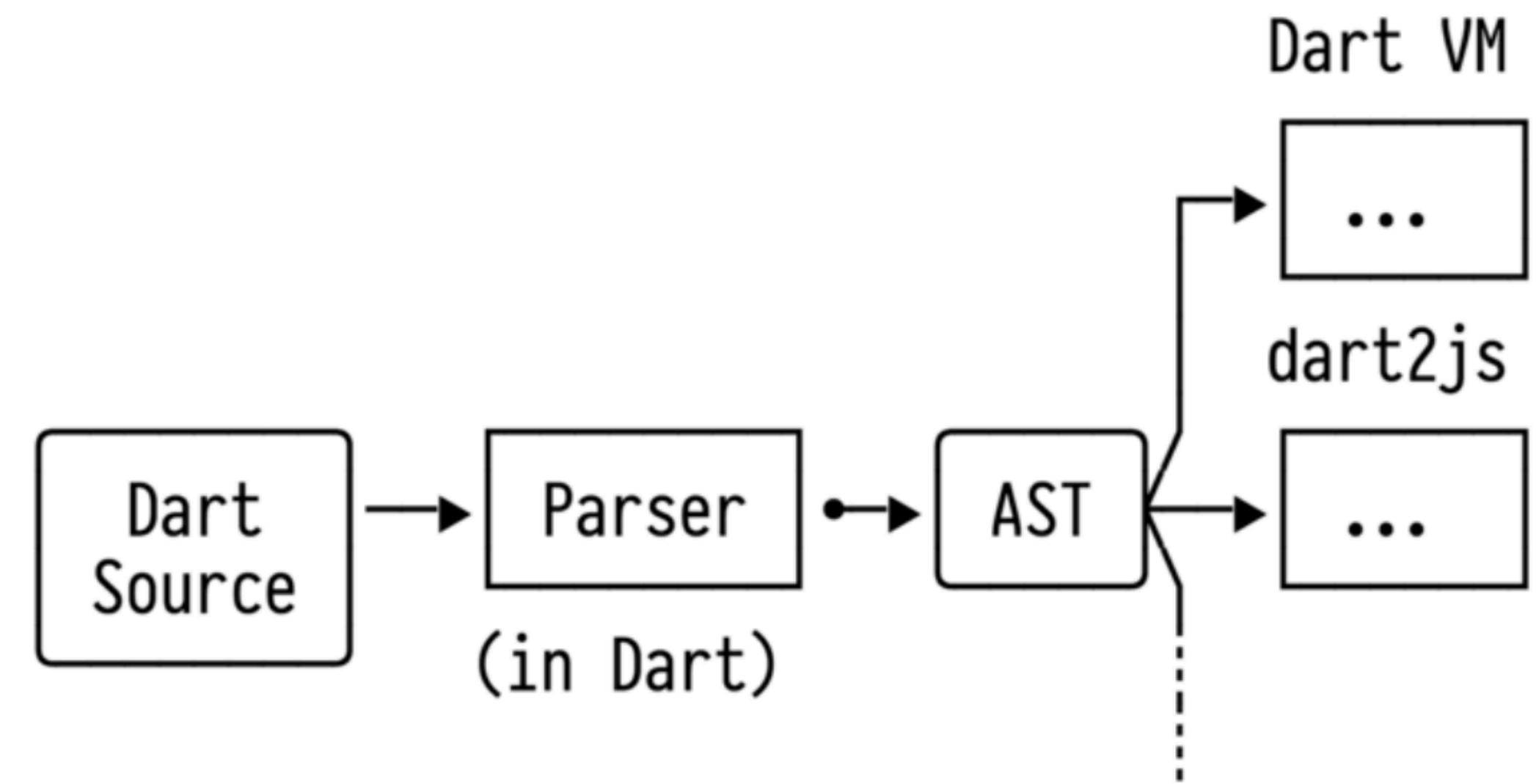
Dart Compiler 体系介绍



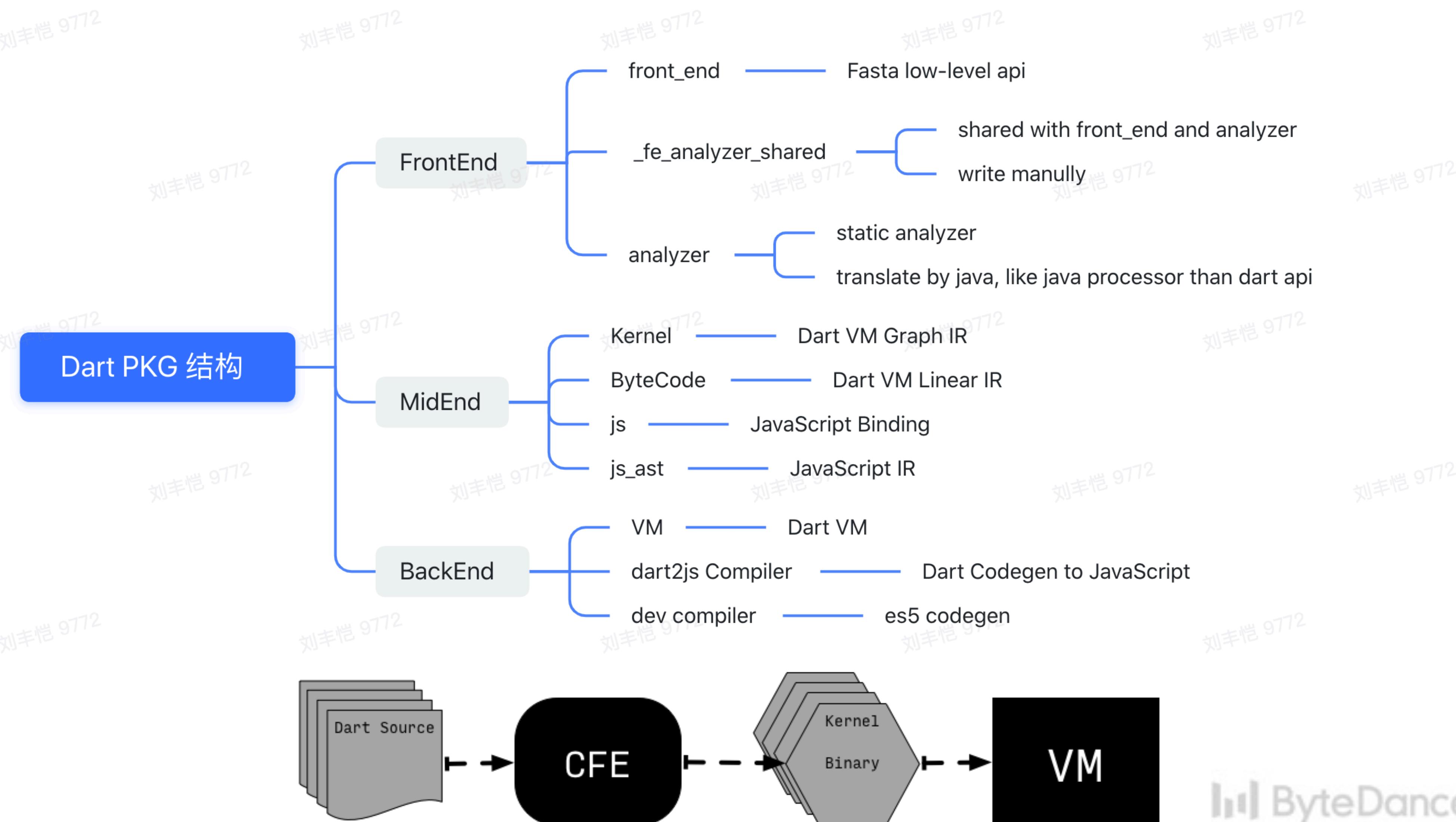
Dart Old Version (2016)



Dart



Dart pkg



Dart ParserGen

1. ANTLR: Java, JavaScript... multi-binding.

2. Old version: front_end, analyzer.

3. Parser Generate Usage, Spec Parser

[Dart Spec Parser https://github.com/dart-lang/sdk/blob/master/tools/spec_parser/Dart.g](https://github.com/dart-lang/sdk/blob/master/tools/spec_parser/Dart.g)

```
// ----- Grammar rules.

libraryDefinition
:   FEFF? SCRIPT_TAG?
|   libraryName?
|   importOrExport*
|   partDirective*
|   (metadata topLevelDefinition)*
|   EOF
;

topLevelDefinition
:   classDeclaration
|   mixinDeclaration
|   extensionDeclaration
|   enumType
|   typeAlias
|   EXTERNAL functionSignature ';'
|   EXTERNAL getterSignature ';'
|   EXTERNAL setterSignature ';'
|   getterSignature functionBody
|   setterSignature functionBody
|   functionSignature functionBody
|   (FINAL | CONST) type? staticFinalDeclarationList ';'
|   LATE FINAL type? initializedIdentifierList ';'
|   topLevelVariableDeclaration ';'
;

topLevelVariableDeclaration
:   LATE? varOrType identifier ('=' expression)?
|   (',' initializedIdentifier)*
;
```

Dart Scanner or Parser

1. Write Scanner & Parser Manually 😅

2. Multi-Impl Dispatch

Scanner <https://sourcegraph.com/github.com/dart-lang/sdk@84ca34bf8b1e04c29e3adfacc7529fee46d4ef40/-/blob/pkg/scanner/scanner.dart>

Parser <https://sourcegraph.com/github.com/dart-lang/sdk@84ca34bf8b1e04c29e3adfacc7529fee46d4ef40/-/blob/pkg/parser/parser.dart>

```
/// Parse a compilation unit.  
///  
/// This method is only invoked from outside the parser. As a result, this  
/// method takes the next token to be consumed rather than the last consumed  
/// token and returns the token after the last consumed token rather than the  
/// last consumed token.  
///  
/// ...  
/// libraryDefinition:  
///   scriptTag?  
///   libraryName?  
///   importOrExport*  
///   partDirective*  
///   topLevelDefinition*  
/// ;  
///  
/// partDeclaration:  
///   partHeader topLevelDefinition*  
/// ;  
/// ...  
Token parseUnit(Token token) {  
  // Skip over error tokens and report them at the end  
  // so that the parser has the chance to adjust the error location.  
  Token errorToken = token;  
  token = skipErrorTokens(errorToken);  
  
  listener.beginCompilationUnit(token);  
  int count = 0;  
  DirectiveContext directiveState = new DirectiveContext();  
  token = syntheticPreviousToken(token);  
  if (identical(token.next.type, TokenType.SCRIPT_TAG)) {  
    directiveState?.checkScriptTag(this, token.next);  
    token = parseScript(token);  
  }  
  while (!token.next.isEof) {  
    final Token start = token.next;  
    token = parseTopLevelDeclarationImpl(token, directiveState);  
    listener.endTopLevelDeclaration(token.next);  
    count++;  
    if (start == token.next) {  
      // Recovery:  
      // If progress has not been made reaching the end of the token stream,  
      // then report an error and skip the current token.  
      token = token.next;  
      listener.beginMetadataStar(token);  
      listener.endMetadataStar(0);  
      reportRecoverableErrorWithToken(  
        "Expected end of file but found $token");  
    }  
  }  
}
```

节跳动

Dart Scanner or Parser

3. Fasta, Analyzer Listener

```
front_end > lib > src > fasta > kernel > body_builder.dart > BodyBuilder > endBinaryExpression
1590     debugEvent("endBinaryExpression");
1591     Expression expression = popForValue();
1592     constantContext = pop();
1593     super.push(expression);
1594 }
1595
1596 @override
1597 void beginBinaryExpression(Token token) {
1598     if (optional("&&", token) || optional("||", token)) {
1599         Expression lhs = popForValue();
1600         typePromoter?.enterLogicalExpression(lhs, token.stringValue);
1601         push(lhs);
1602     }
1603 }
1604
1605 @override
1606 void endBinaryExpression(Token token) {
1607     debugEvent("BinaryExpression");
1608     if (optional(".", token) ||
1609         optional(.., token) ||
1610         optional(?., token)) {
1611         doDotOrCascadeExpression(token);
1612     } else if (optional("&&", token) || optional("||", token)) {
1613         doLogicalExpression(token);
1614     } else if (optional("??", token)) {
1615         doIfNull(token);
1616     } else if (optional(?., token)) {
1617         doIfNotNull(token);
1618     } else {
1619         doBinaryExpression(token);    Johnni Winther, 7 months ago • [cfe] Hand
1620     }
1621 }
1622
1623
1624
1625
1626
1627
1628
1629
1630 }
```

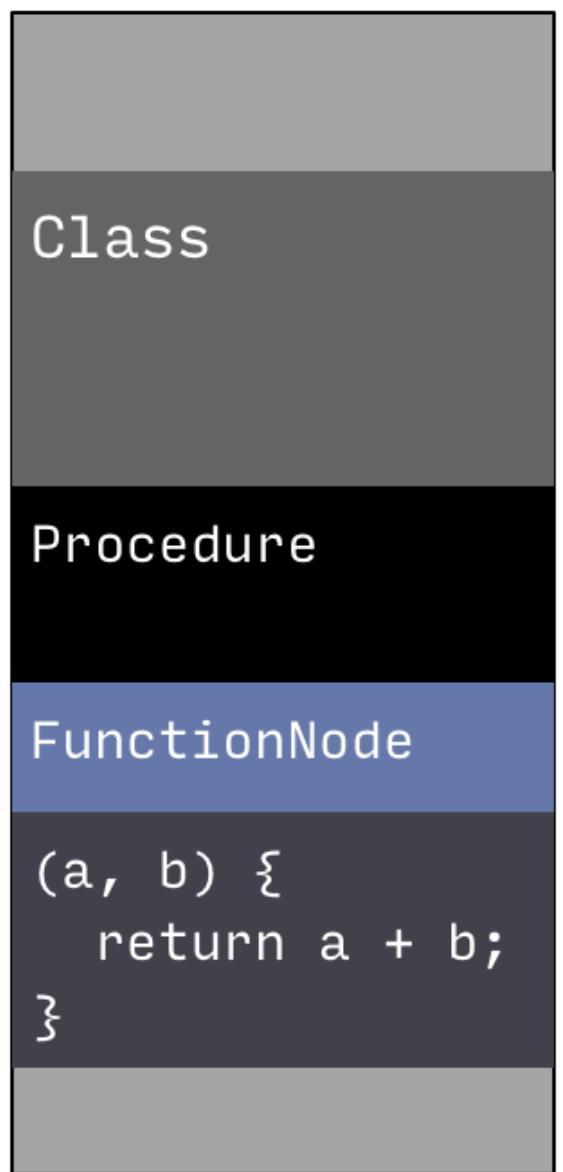
```
analyzer > lib > src > fasta > ast_builder.dart > AstBuilder > endBinaryExpression
583     assert(optional('await', awaitKeyword));
584     debugEvent("AwaitExpression");
585
586     push(ast.awaitExpression(awaitKeyword, pop()));
587 }
588
589 @override
590 void endBinaryExpression(Token operatorToken) {
591     assert(operatorToken.isOperator ||
592            optional('.', operatorToken) ||
593            optional('..', operatorToken) ||
594            optional('...', operatorToken) ||
595            optional('==', operatorToken) ||
596            optional('!=', operatorToken)); Brian Wilkerson, 5 months ago • Han
597     debugEvent("BinaryExpression");
598
599     if (identical(".", operatorToken.stringValue) ||
600         identical(.., operatorToken.stringValue) ||
601         identical(..., operatorToken.stringValue) ||
602         identical(?., operatorToken.stringValue)) {
603         doDotExpression(operatorToken);
604     } else {
605         Expression right = pop();
606         Expression left = pop();
607         reportErrorIfSuper(right);
608         push(ast.binaryExpression(left, operatorToken, right));
609         if (!enableTripleShift & operatorToken.type == TokenType.GT_GT_GT) {
610             handleRecoverableError(
611                 templateExperimentNotEnabled
612                     .withArguments(EnableString.triple_shift),
613                     operatorToken,
614                     operatorToken);
615     }
616 }
617 }
618 }
619 }
```

Dart JIT

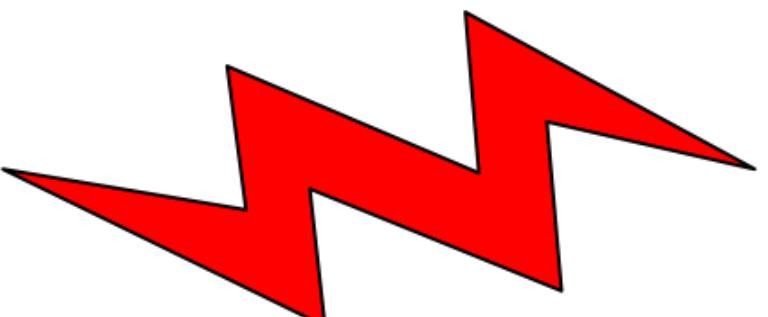
1. Kernel: Graph IR

2. ByteCode, IL : Linear IR

KERNEL BINARY



Graph construction phase constructs a stack based *intermediate language* (**IL**) directly from Kernel binary



LoadLocal('a')
PushArgument
LoadLocal('b')
PushArgument
InstanceCall('+')
Return

The IL is later directly, without any optimization passes, lowered to machine code

pushq [rbp+...]
pushq [rbp+...]
callq InlineCacheStub
retq

ByteCode Codegen: https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/pkg/vm/lib/bytocode/gen_bytecode.dart

ByteCode Interpreter: <https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/runtime/vm/interpreter.cc>

IL : <https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/runtime/vm/compiler/backend/il.h>

Compilers: <https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/runtime/vm/isolate.h#L1405:3>

JIT Call Site Spec

1. Call Site Inline Caching

2. AOT: Like JVM invoke dynamic

```
class Dog {  
    toFace() => '🐶';  
}  
  
class Cat {  
    toFace() => '🐱';  
}  
  
sameFace(animal, face) =>  
    [animal.toFace()] == face;  
  
sameFace(Dog(), '🦊');  
sameFace(Dog(), '🐵');  
sameFace(Cat(), '🐷');
```

RawICData (inline cache data)

```
// {class: (method, frequency)}  
{  
    Dog: (Dog.toFace, 2)  
    Cat: (Cat.toFace, 1)  
}
```

InlineCacheStub

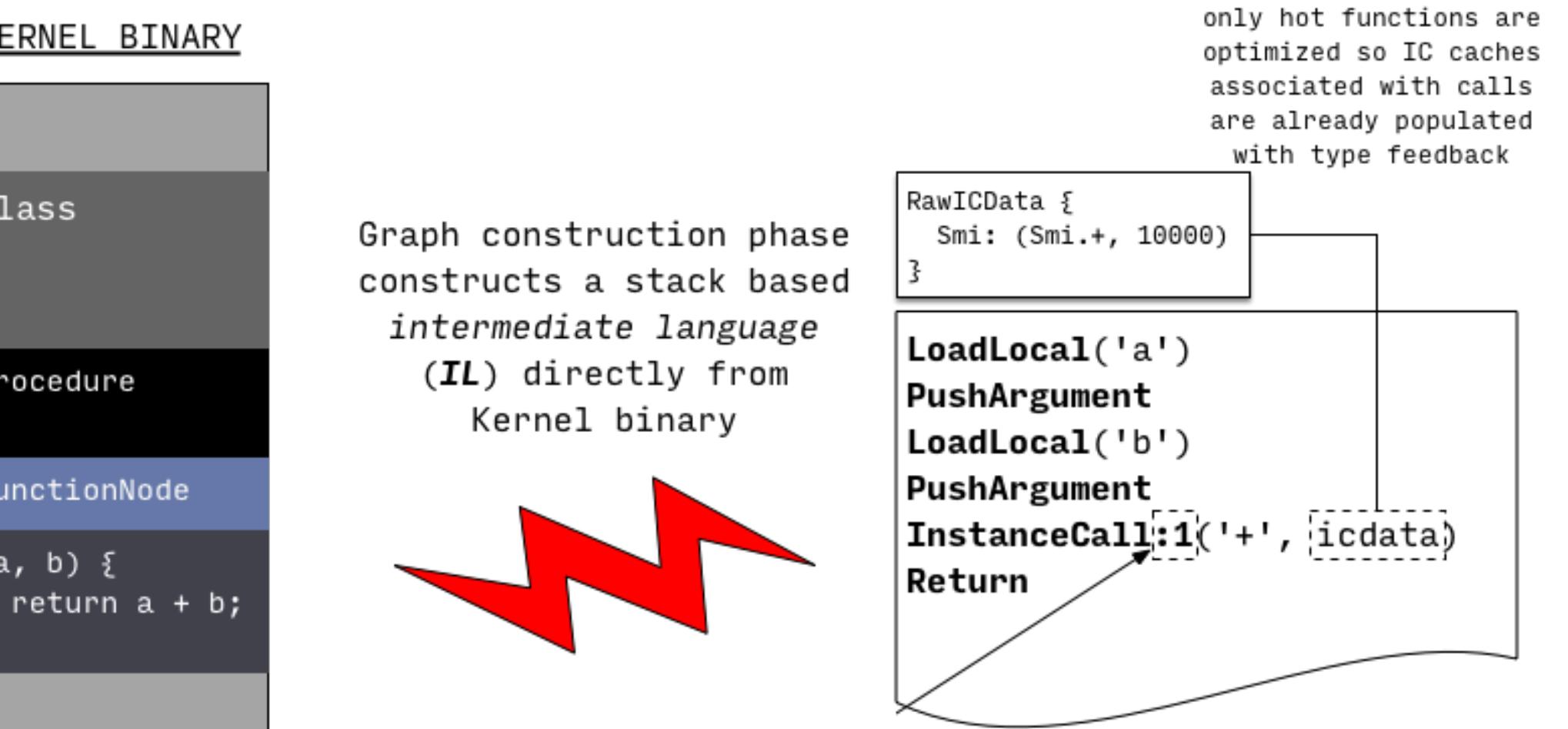
```
entry = cache.lookup(classOf(receiver))  
if (entry != null) {  
    // Success - use cached entry.  
    entry.frequency++;  
    return entry.method(receiver, ...);  
}  
// Failure - call runtime helper to handle  
// actual lookup or throw NoSuchMethodError.  
// If runtime succeeds - it would update  
// the cache.  
return InlineCacheMiss(cache, ...)
```

DartVM SSA Optimizer

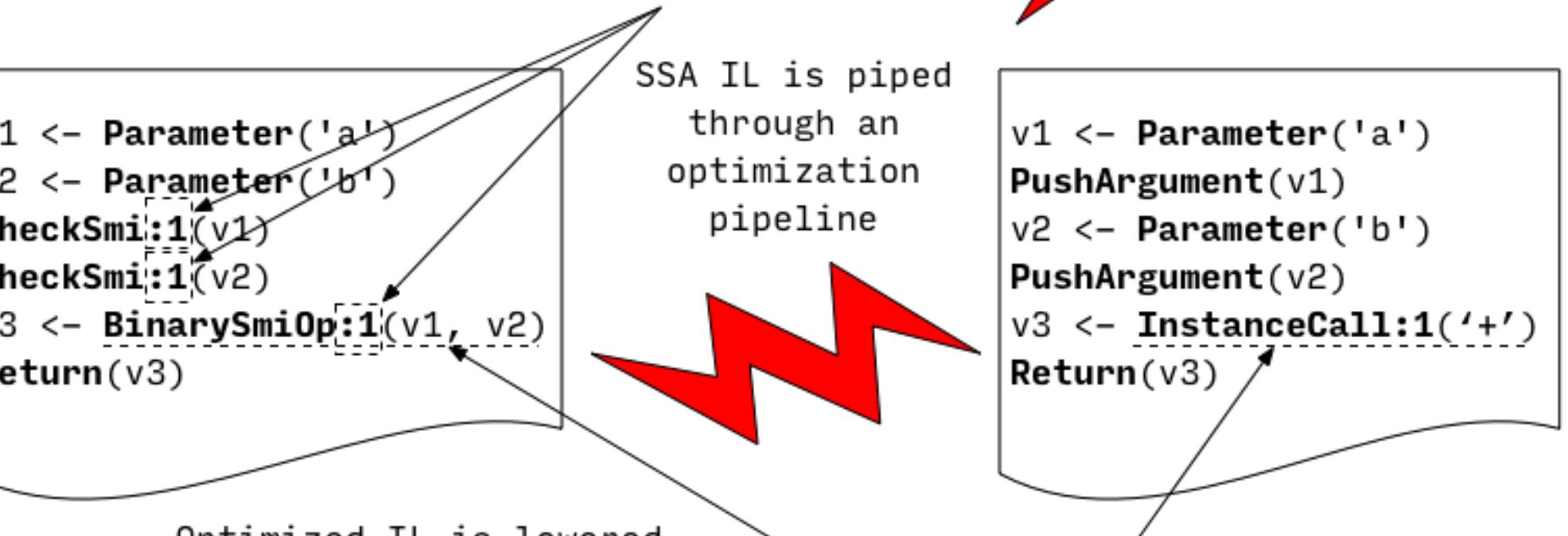
1. SSA: Single Static Assignment.

2. IL -> SSA IL -> Opt IL

3. Example: Diamond Control Flow

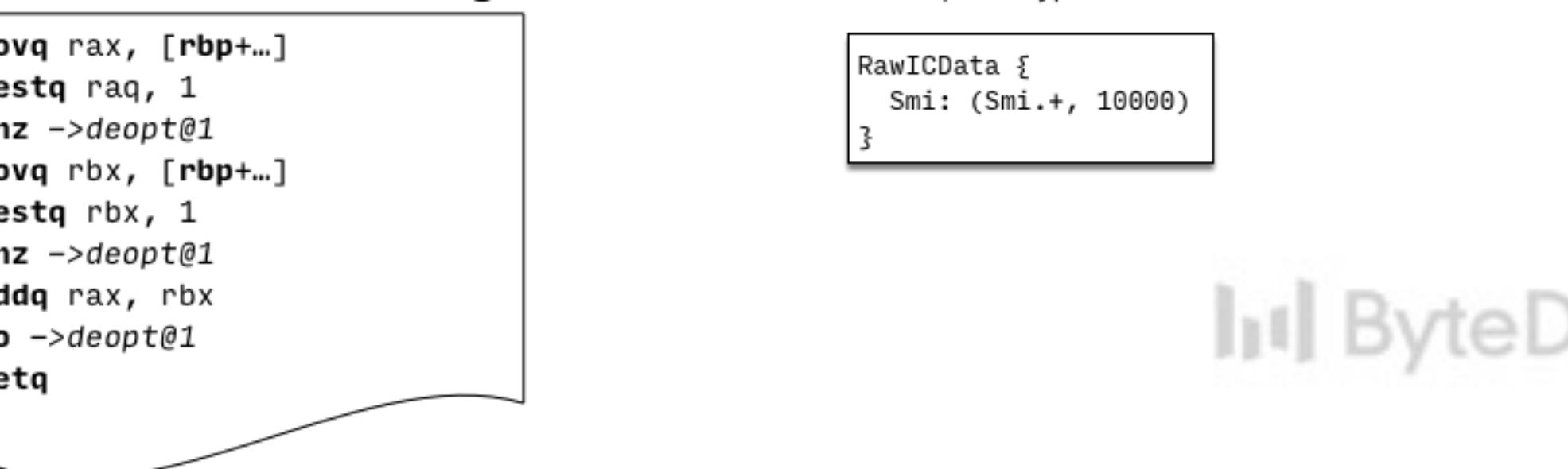


deoptimization ids are assigned to IL instructions when unoptimized IL is built. These ids are used when deoptimizing, i.e. switching from optimized back to unoptimized code.

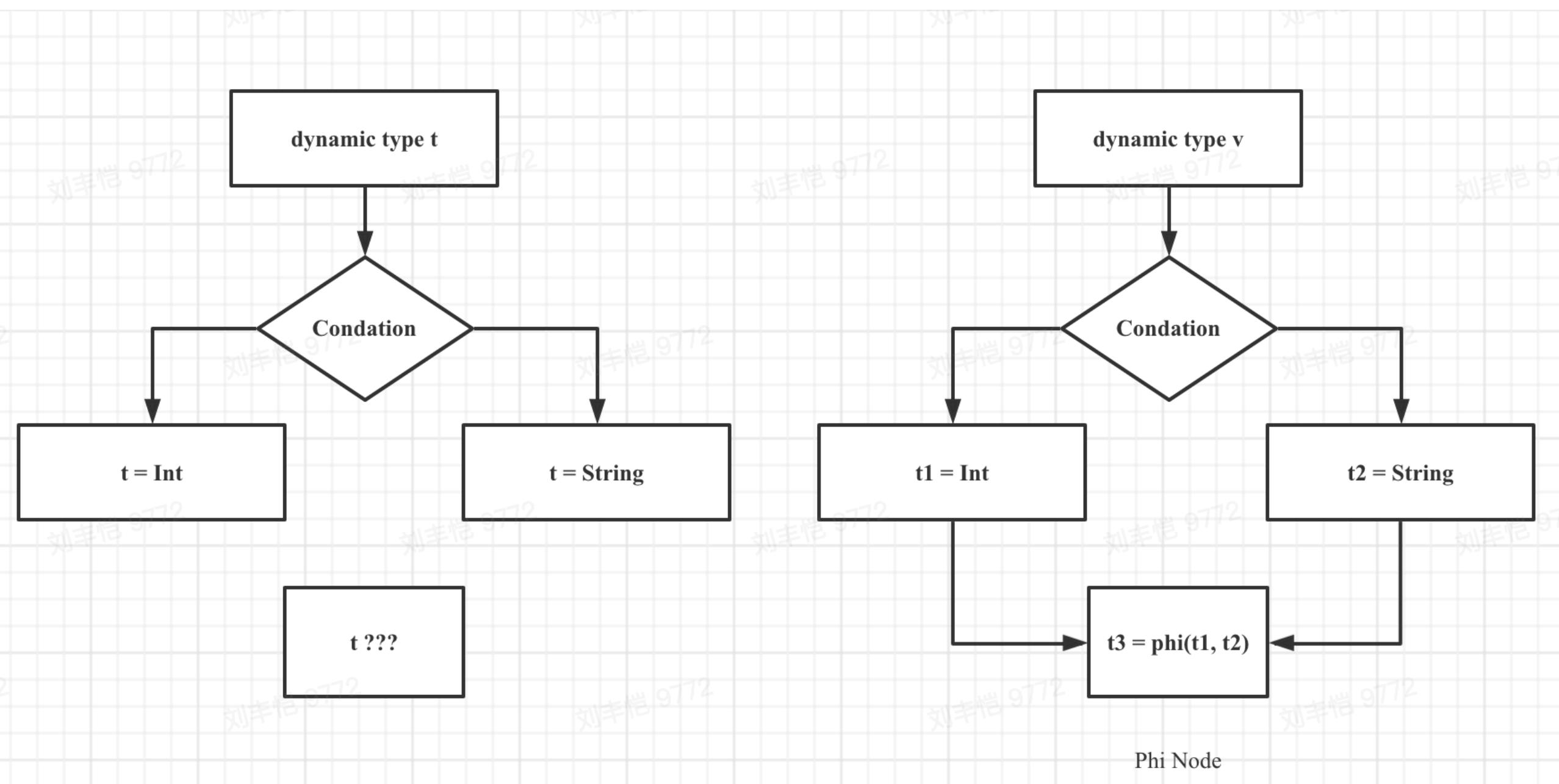


Optimized IL is lowered to machine code using linear scan register allocator and one-to-many IL instruction lowering.

optimization pipeline uses type feedback collected by inline caches to speculatively specialize code: in this example Smi.+ is speculatively inlined based on the monomorphic type feedback



Example: Diamond Control Flow



```

void IfConverter::Simplify(FlowGraph* flow_graph) {
    Zone* zone = flow_graph->zone();
    bool changed = false;

    const GrowableArray<BlockEntryInstr*>& postorder = flow_graph->postorder();
    for (BlockIterator it(postorder); !it.Done(); it.Advance()) {
        BlockEntryInstr* block = it.Current();
        JoinEntryInstr* join = block->AsJoinEntry();

        // Detect diamond control flow pattern which materializes a value depending
        // on the result of the comparison:
        // Florian Schneider, 4 years ago • VM: Move branch optimizations into a
        // B_pred:
        // ...
        // Branch if COMP goto (B_pred1, B_pred2)
        // B_pred1: -- trivial block that contains at most one definition
        //   v1 = Constant(...)
        //   goto B_block
        // B_pred2: -- trivial block that contains at most one definition
        //   v2 = Constant(...)
        //   goto B_block
        // B_block:
        //   v3 = phi(v1, v2) -- single phi
        //
        // and replace it with
        //
        // Ba:
        //   v3 = IfThenElse(COMP ? v1 : v2)
        //
        if ((join != NULL) && (join->phis() != NULL) &&
            (join->phis()->length() == 1) && (block->PredecessorCount() == 2)) {
            BlockEntryInstr* pred1 = block->PredecessorAt(0);
            BlockEntryInstr* pred2 = block->PredecessorAt(1);

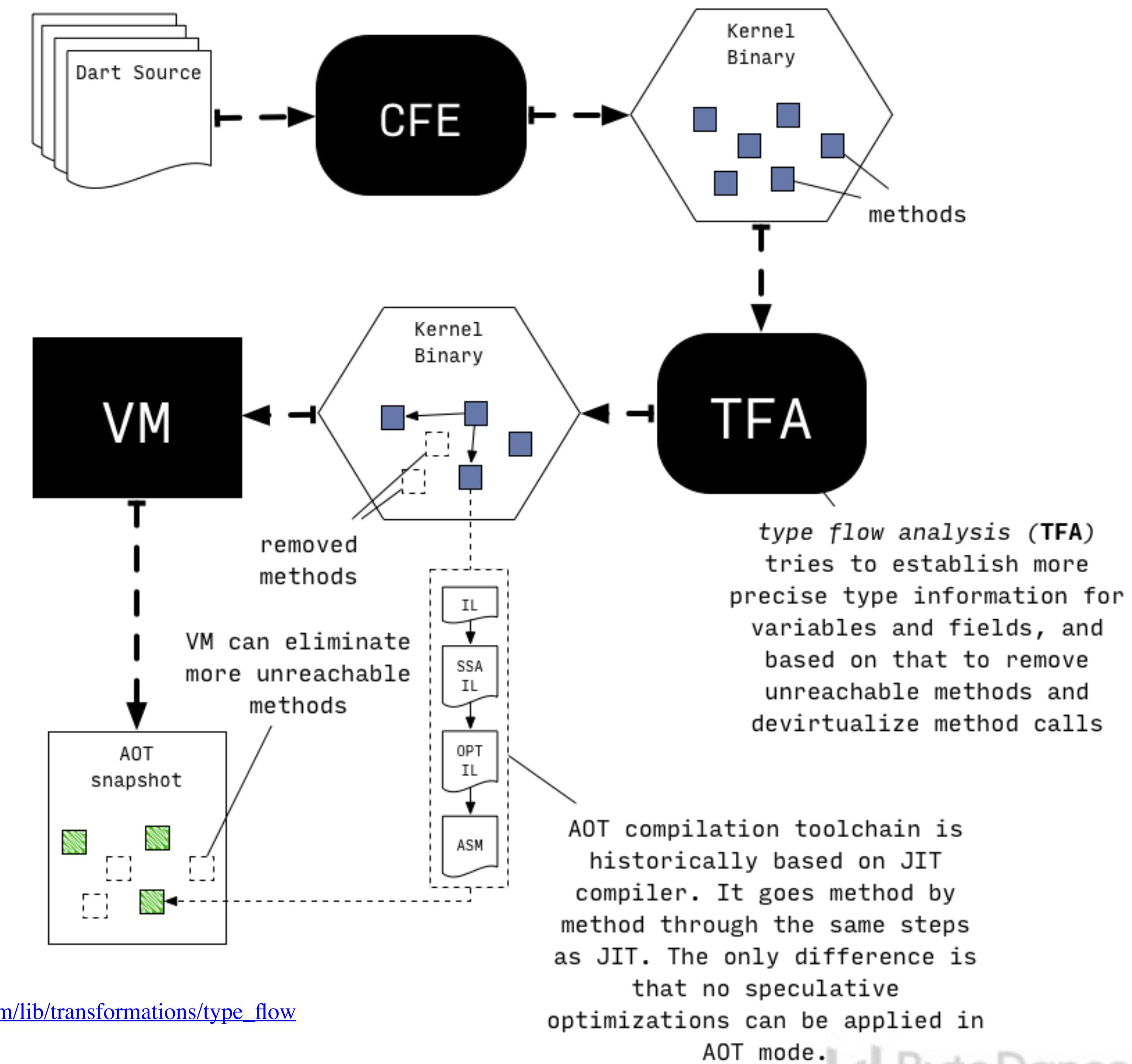
            PhiInstr* phi = (*join->phis())[0];
            Value* v1 = phi->InputAt(0);
  
```

Dart VM SSA Passes: https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/runtime/vm/compiler/compiler_pass.cc#L419

If Converter Pass: https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/runtime/vm/compiler/backend/branch_optimizer.cc#L242

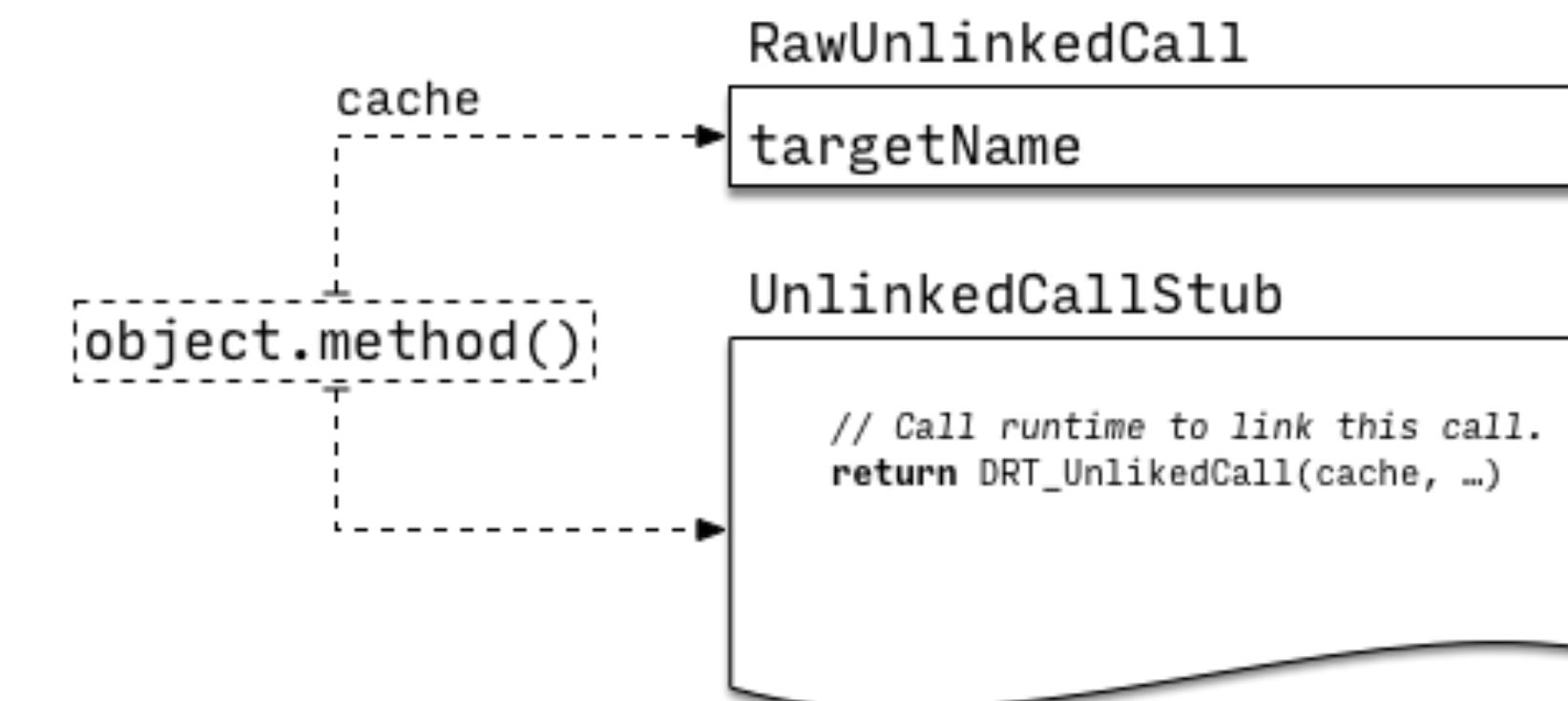
Dart AOT

1. Dead Code Elim: TreeShaking
2. TFA: Type Flow Analysis
3. CFA: Control Flow Analysis
4. TFA = CFA + Type Infr (One Pass)



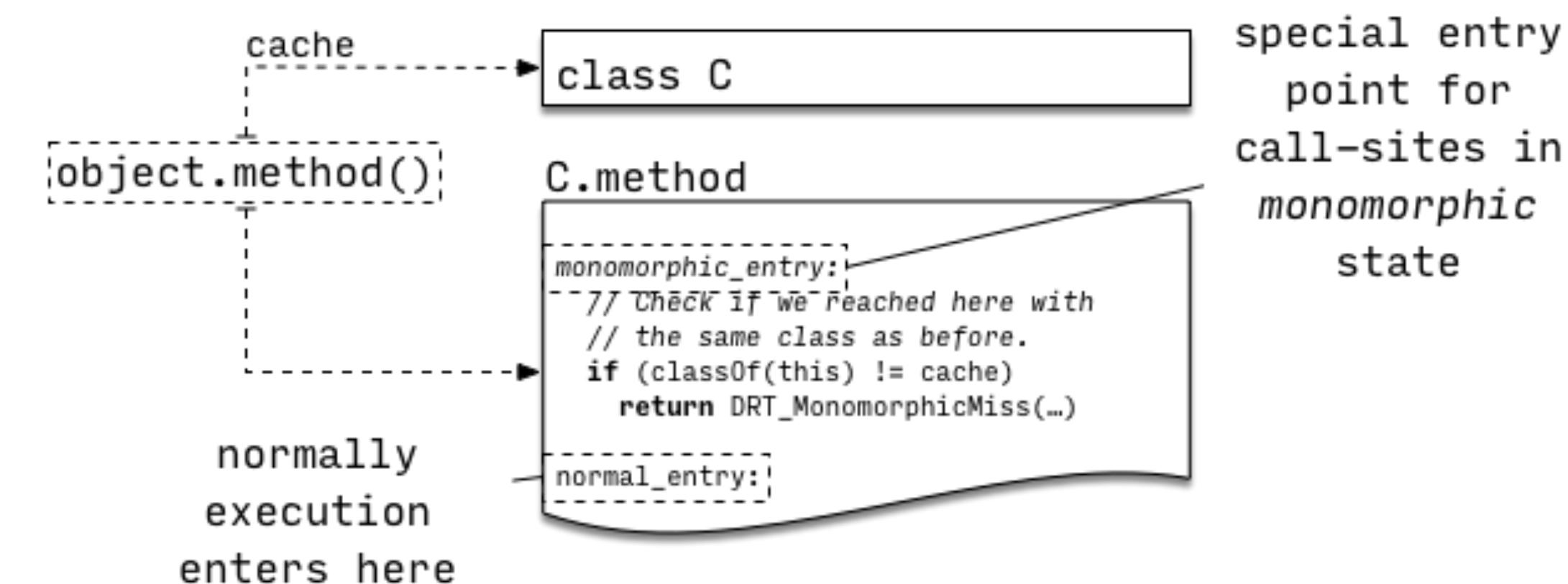
[TFA: https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/tree/pkg/vm/lib/transformations/type_flow](https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/tree/pkg/vm/lib/transformations/type_flow)

AOT Call Site Spec



1. AOT Runtime Dispatch

2. Mono/Poly-Morphic Call Site



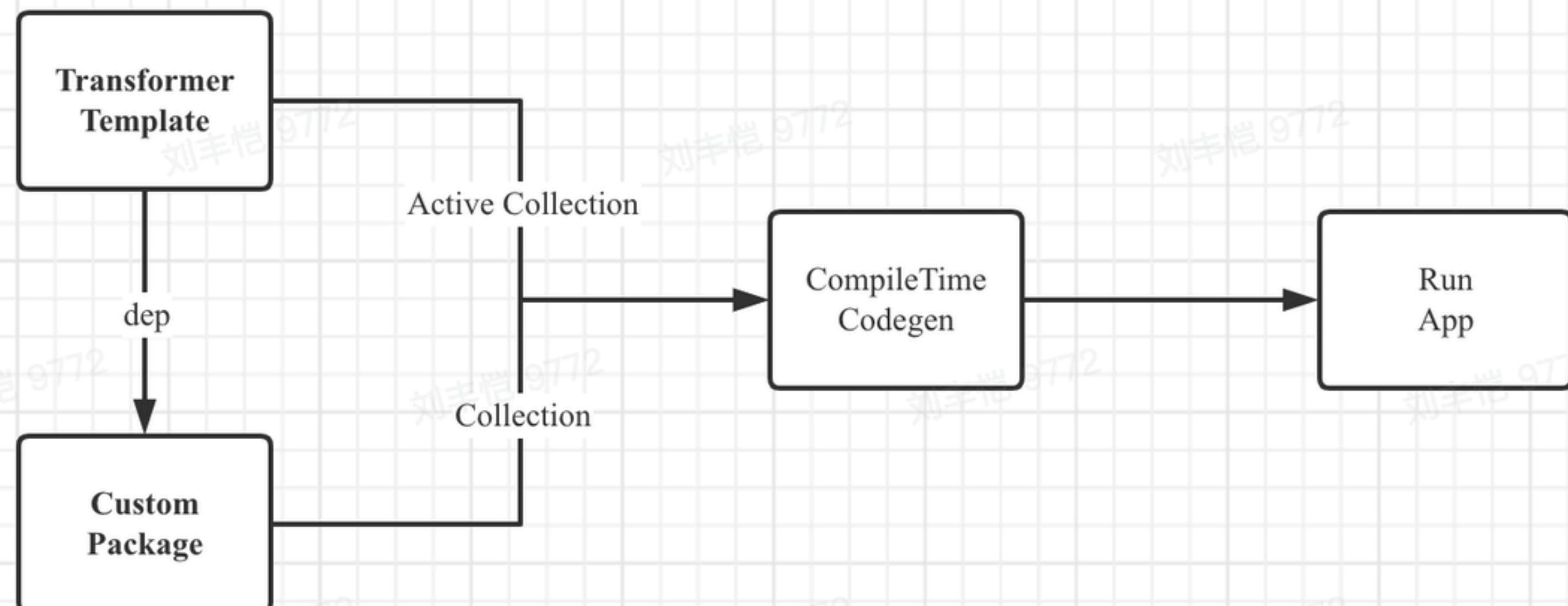
AOT Call Site: https://sourcegraph.com/github.com/dart-lang/sdk@84ca34b/-/blob/runtime/vm/runtime_entry.cc#L1723

具体案例分析



Transformer Template 体系

Compile-Time CodeGen



1. Hook Flutter Compiler
2. Run Transformer Patch

Auto Jsonify

Compile-Time CodeGen

1. Flutter 默认 Json 框架的使用

```
lib > people.dart > Person
1 import 'package:json_annotation/json_annotation.dart';
2
3 part 'people.g.dart';
4
5 @JsonSerializable(nullable: true)
6 class Person {
7   @JsonKey(nullable: true, includeIfNull: false)
8   final String firstName;
9   final String lastName;
10  final DateTime dateOfBirth;
11
12  Person({this.firstName, this.lastName, this.dateOfBirth});
13  factory Person.fromJson(Map<String, dynamic> json) => _$PersonFromJson(json);
14  Map<String, dynamic> toJson() => _$PersonToJson(this);
15 }
```

flutter pub run build_runner build

```
// GENERATED CODE - DO NOT MODIFY BY HAND
part of 'people.dart';

// *****
// JsonSerializerGenerator
// *****

Person _$PersonFromJson(Map<String, dynamic> json) {
  return Person(
    firstName: json['firstName'] as String,
    lastName: json['lastName'] as String,
    dateOfBirth: json['dateOfBirth'] == null
      ? null
      : DateTime.parse(json['dateOfBirth'] as String),
  );
}

Map<String, dynamic> _$PersonToJson(Person instance) {
  final val = <String, dynamic>{};
  void writeNotNull(String key, dynamic value) {
    if (value != null) {
      val[key] = value;
    }
  }

  writeNotNull('firstName', instance.firstName);
  val['lastName'] = instance.lastName;
  val['dateOfBirth'] = instance.dateOfBirth?.toIso8601String();
  return val;
}
```

Auto Jsonify

Compile-Time CodeGen

2. Source Codegen 的问题

- 代码编写复杂、困难
- 代码生成记不记录到 git 系统?
- build_runner 生成不稳定，影响 CI 构建
- json codegen 不同版本生成出的代码不同



```
lib > people.dart > Person
1 import 'package:json_annotation/json_annotation.dart';
2
3 part 'people.g.dart';
4
5 @JsonSerializable(nullable: true)
6 class Person {
7   @JsonKey(nullable: true, includeIfNull: false)
8   final String firstName;
9   final String lastName;
10  final DateTime dateOfBirth;
11
12  Person({this.firstName, this.lastName, this.dateOfBirth});
13  factory Person.fromJson(Map<String, dynamic> json) => _$PersonFromJson(json);
14  Map<String, dynamic> toJson() => _$PersonToJson(this);
15 }
```

Auto Jsonify

Compile-Time CodeGen

```
import 'package:just_jsonify_annotation/jsonify_annotation.dart';

@JsonAble()
class Body with JsonModel<Body> {
    String body;

    Body({this.body});

    @JsonAble()
    class Response with JsonModel<Response> {
        int code;
        @JsonProperty(name: 'msg', nullable: true)
        String message;

        Body body;

        Response({this.code, this.message, this.body});
    }

    void main() {
        print(
            Response(code: 200, message: 'success', body: Body(body: 'data')).toJson(),
        );

        print(
            Response().fromJson({'code': 200, 'message': 'success', 'body': {'body': 'data'}}).code
        );
    }
}
```

1.Compile-Time Codegen

2.Modern Json Framework Usage

3.Convention over configuration

Auto Jsonify

The screenshot shows a Dart project structure and code editor in an IDE. The project tree on the left includes folders for .dart_tool, .idea, android, build, ios, lib, test, web, and various configuration files like pubspec.yaml and README.md. The main.dart file in the lib folder is selected in the editor.

```
transformer-template > jsonify_show > lib > main.dart
Project: transformer-template
1: Project
  - .dart_tool
  - .idea
  - android
  - build
  - ios
  - lib
    - main.dart
  - test
  - web
    - .gitignore
    - .metadata
    - .packages
    - jsonify_example.iml
    - pubspec.lock
    - pubspec.yaml
    - README.md
  - jsonify_show
    - .dart_tool
    - .idea
    - android
    - build
    - ios
    - lib
      - expect_utils.dart
      - main.dart
    - test
      - widget_test.dart
  - web
    - .gitignore
    - .metadata
    - .packages
    - jsonify_show.iml

main.dart
5   @JsonAble()
6   class Body with JsonModel<Body> {
7     String body;
8
9     Body({this.body});
10
11  @JsonAble()
12  class Response with JsonModel<Response> {
13    int code;
14    @JsonProperty(name: 'msg', nullable: true)
15    String message;
16
17    Body body;
18
19    Response({this.code, this.message, this.body});
20
21  void main() {
22    print(
23      Response(code: 200, message: 'success', body: Body(body: 'data')).toJson(),
24    );
25
26    print(
27      Response().fromJson({'code': 200, 'message': 'success', 'body': {'body': 'data'}}).code
28    );
29
30  }
31
```

The terminal at the bottom shows the build process:

```
Running Gradle task 'assembleDebug'...
Running Gradle task 'assembleDebug'... Done 5.8s
✓ Built build/app/outputs/apk/debug/app-debug.apk.
Installing build/app/outputs/apk/app.apk... 6.6s
I/flutter (32295): {code: 200, msg: success, body: {body: data}}
I/flutter (32295): 200
Syncing files to device MIX 2... 324ms
```

Observatory information is also displayed:

```
🔥 To hot reload changes while running, press "r". To hot restart (and rebuild state), press "R".
An Observatory debugger and profiler on MIX 2 is available at: http://127.0.0.1:64205/e1oJlwq7f7s=/
For a more detailed help message, press "h". To detach, press "d"; to quit, press "q".
Application finished.
```

Bottom navigation bar:

- TODO
- Dart Analysis
- Git
- Terminal
- Messages

Bottom status bar:

- 00:00
- 结束录制
- 30:2 LF UTF-8 2 spaces
- master
- 2008 of 3964M

Dart Version Control

Compile-Time CodeGen

Dart 语法向下兼容问题

Dart、Flutter 接口稳定性问题

Dart Version Control

Compile-Time CodeGen

```
...  ... @@ -412,11 +412,9 @@ class _SelectionHandleDriverState extends State<SelectionHandleDr:  
412 412     handleSize.width,  
413 413     handleSize.height,  
414 414 );  
415 - // Make sure the GestureDetector is big enough to be easily interactive.  
416 415 final Rect interactiveRect = handleRect.expandToInclude(  
417 416     Rect.fromCircle(  
418 417     center: handleRect.center, radius: kMinInteractiveSize / 2),  
419 417 +     Rect.fromCircle(center: handleRect.center, radius: 48.0 / 2),  
420 418 );  
421 419 final RelativeRect padding = RelativeRect.fromLTRB(  
422 420     math.max((interactiveRect.width - handleRect.width) / 2, 0),  
...  ... @@ -579,15 +577,21 @@ class _SelectionToolbarState extends State<_SelectionToolbar> {  
579 577     block.localToGlobal(block.size.bottomRight(Offset.zero)),  
580 578 );  
581 579  
582 - // final toolbar = widget.controls.buildToolbar(context, editingRegion,  
583 - //     block.preferredLineHeight, midpoint, endpoints, widget.delegate);  
584 580 final toolbar = widget.controls.buildToolbar(  
585 581     context,  
586 582     editingRegion,  
583 +     block.preferredLineHeight,  
587 584     midpoint,  
588 585     endpoints,  
589 586     widget.delegate,  
590 587 );  
588 + // final toolbar = widget.controls.buildToolbar(  
589 + //     context,  
590 + //     editingRegion,  
591 + //     midpoint,  
592 + //     endpoints,  
593 + //     widget.delegate,  
594 + // );  
591 595 return new CompositedTransformFollower(  
592 596     link: block.layerLink,  
593 597     showWhenUnlinked: false,  
...  ...
```

Dart Version Control

Compile-Time CodeGen

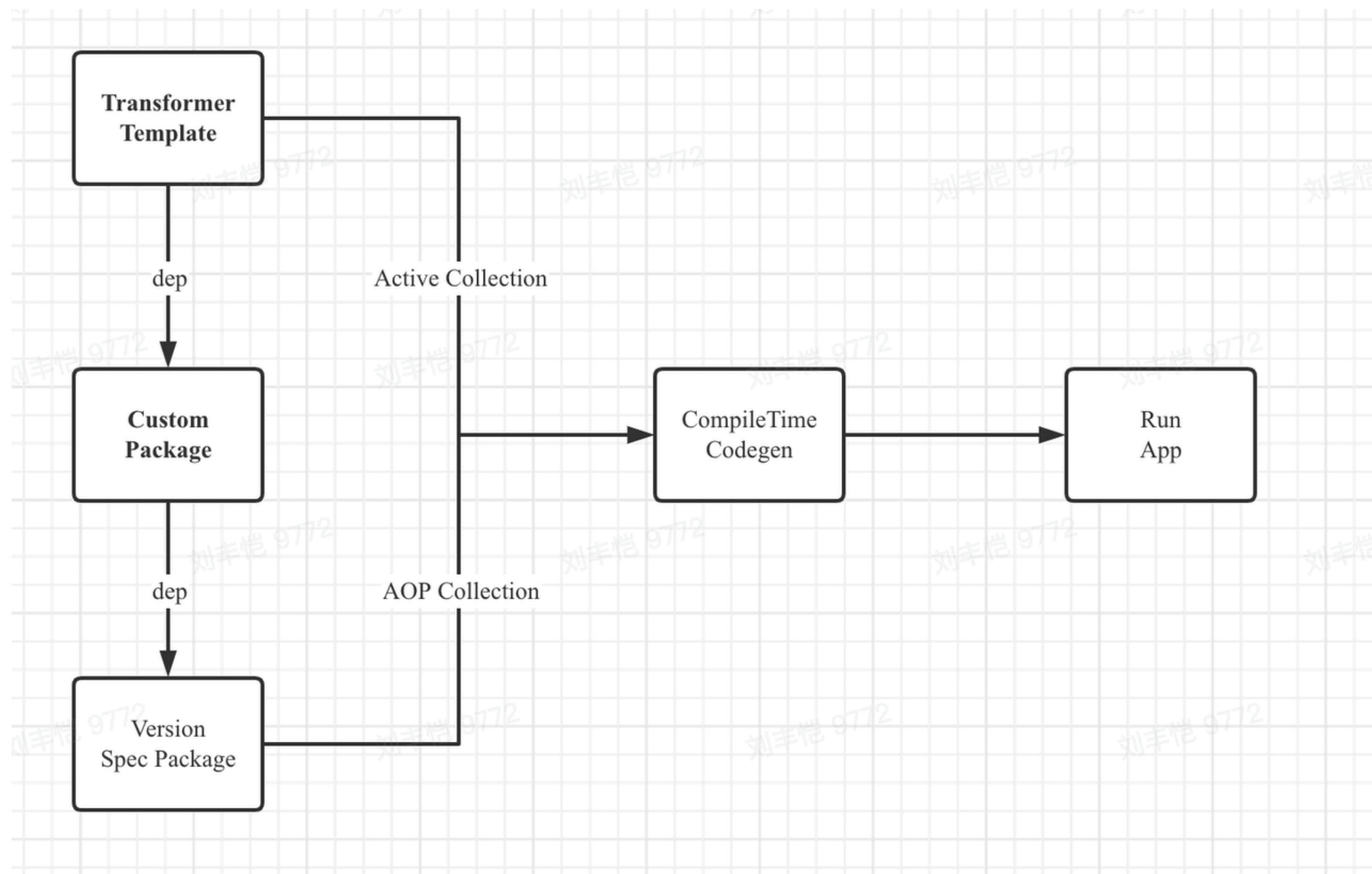
```
// common package.  
@Inject()  
Widget buildToolbar(  
    BuildContext context,  
    Rect globalEditableRegion,  
    double textLineHeight,  
    Offset position,  
    List<TextSelectionPoint> endpoints,  
    TextSelectionDelegate delegate,  
) {  
    return Container();  
}
```

```
// spec package.  
@InjectTo(version: 'bd_154')  
Widget buildToolbar(  
    BuildContext context,  
    Rect globalEditableRegion,  
    double textLineHeight,  
    Offset position,  
    List<TextSelectionPoint> endpoints,  
    TextSelectionDelegate delegate,  
) {  
    // BD 154  
}
```

```
@InjectTo(version: 'bd_112')  
Widget buildToolbar(  
    BuildContext context,  
    Rect globalEditableRegion,  
    double textLineHeight,  
    Offset position,  
    List<TextSelectionPoint> endpoints,  
    TextSelectionDelegate delegate,  
) {  
    // BD 112  
}
```

Dart Version Control

Compile-Time CodeGen



1. Call Site Inject
2. Version Call Site Collection
3. Compile Time Replace

For Future 未来方向



For Future

- 1.General AOP
- 2.Codegen Injection
- 3.SSA Optimizer
- 4.FlutterWeb
- 5.Multi-Backend
- 6.Extend Dart2XXX



THANKS.

