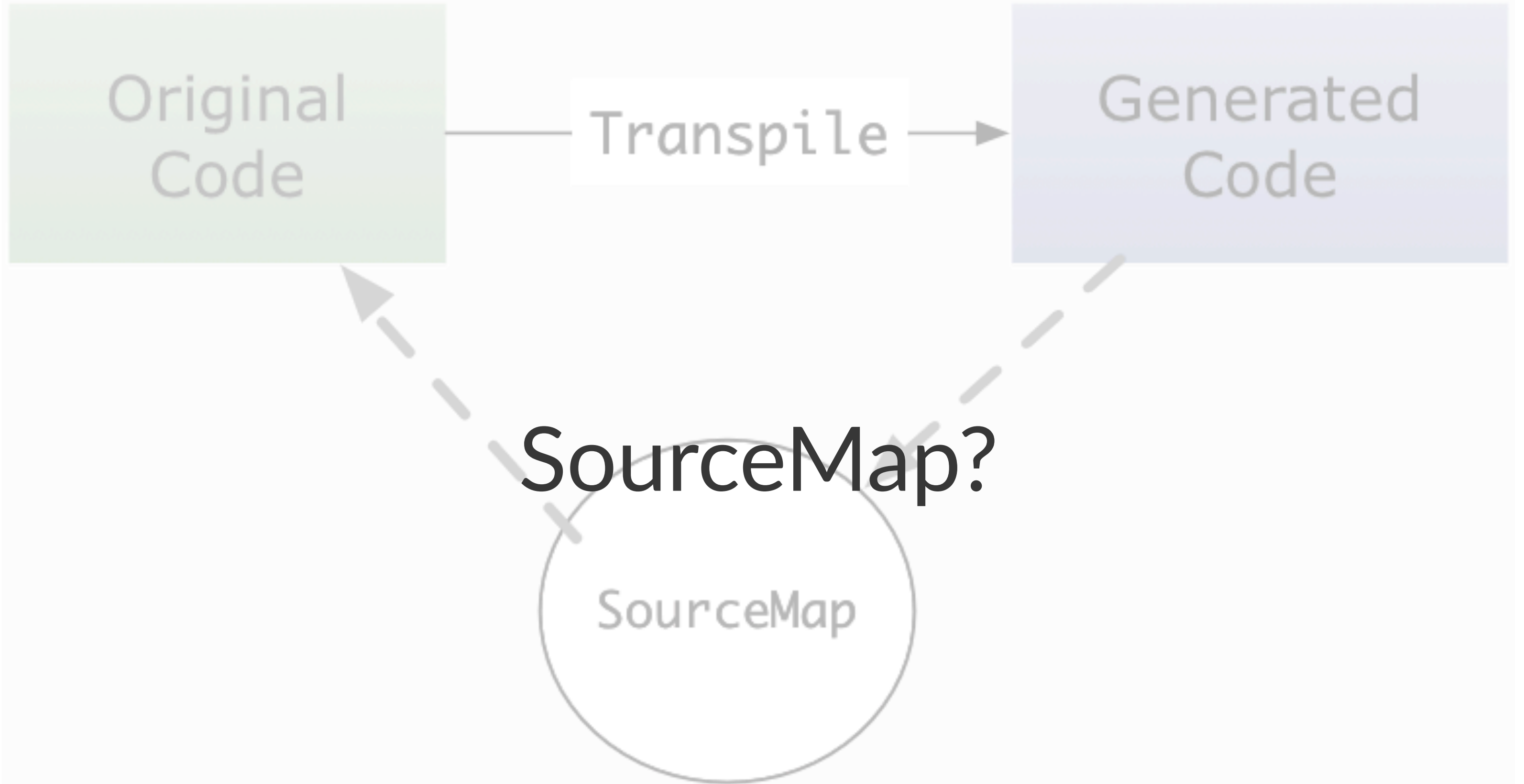


多段SourceMapと統一的なインターフェース

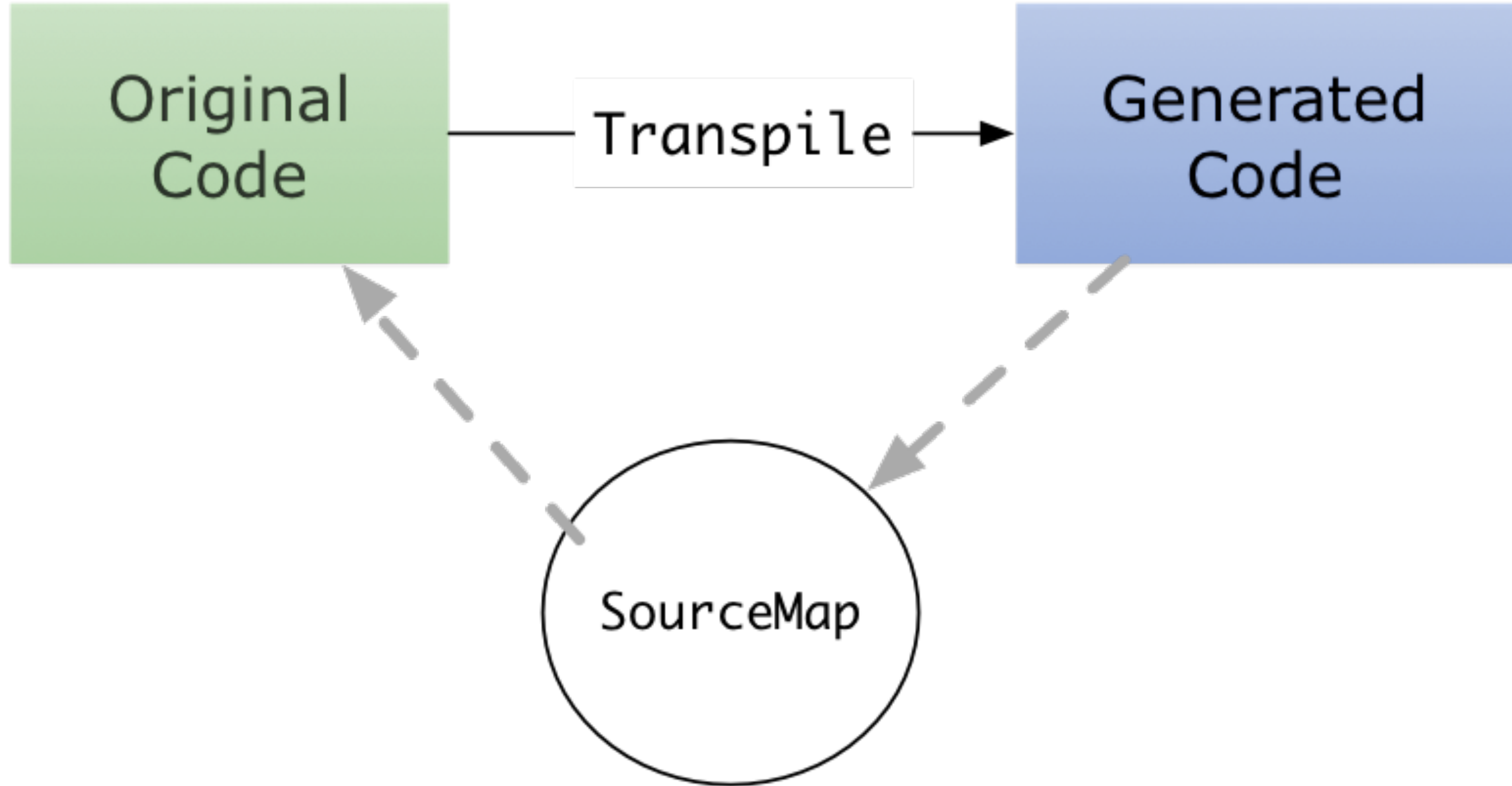
自己紹介

- Name : **azu**
- Twitter : [@azu_re](#)
- Website: [Web scratch](#), [JSer.info](#)





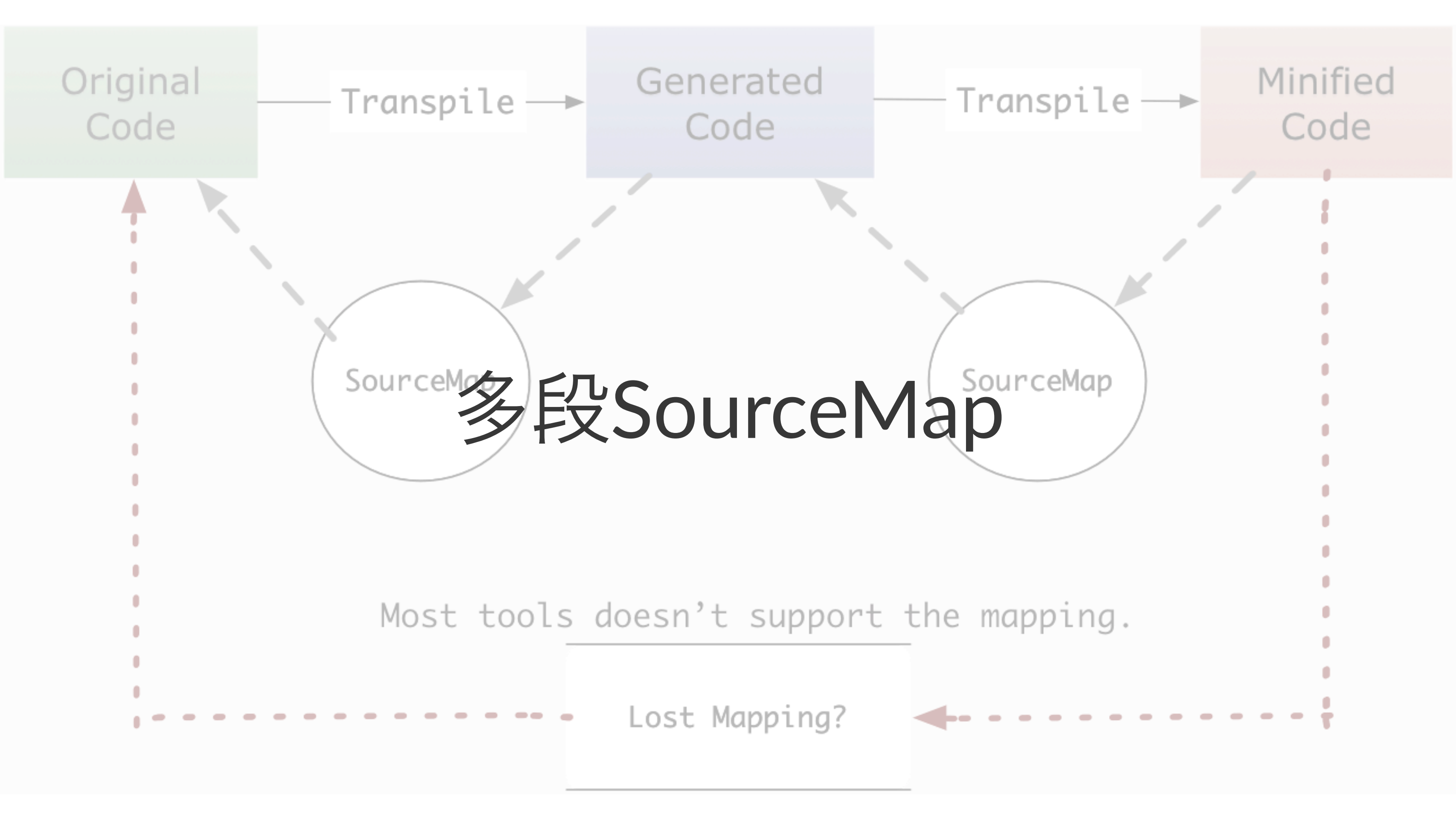
Mapping: line-number, column, filePath, name

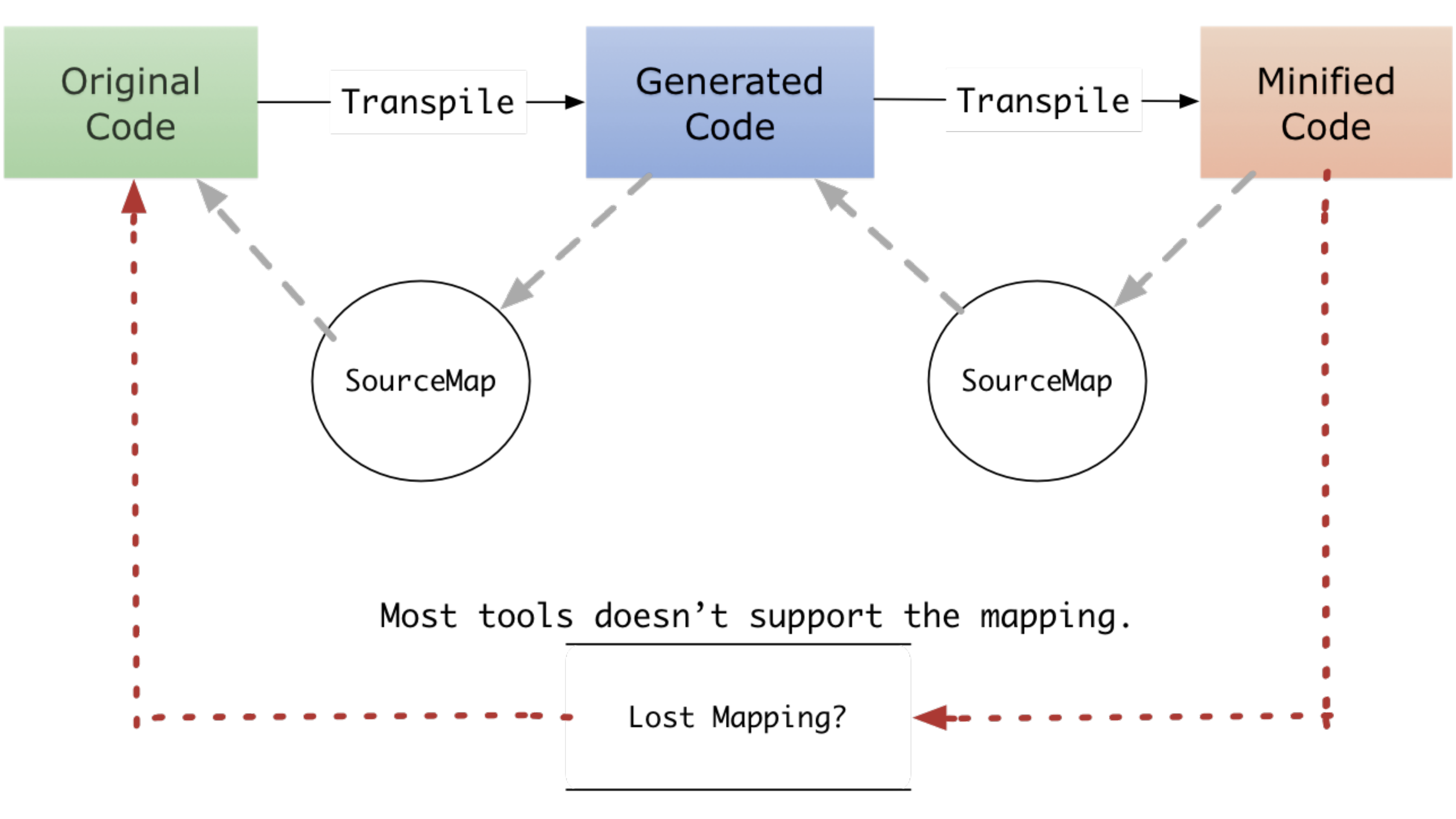


Mapping: `line-number, column, filePath, name`

1対1のSourceMap

- 問題ない
- 便利



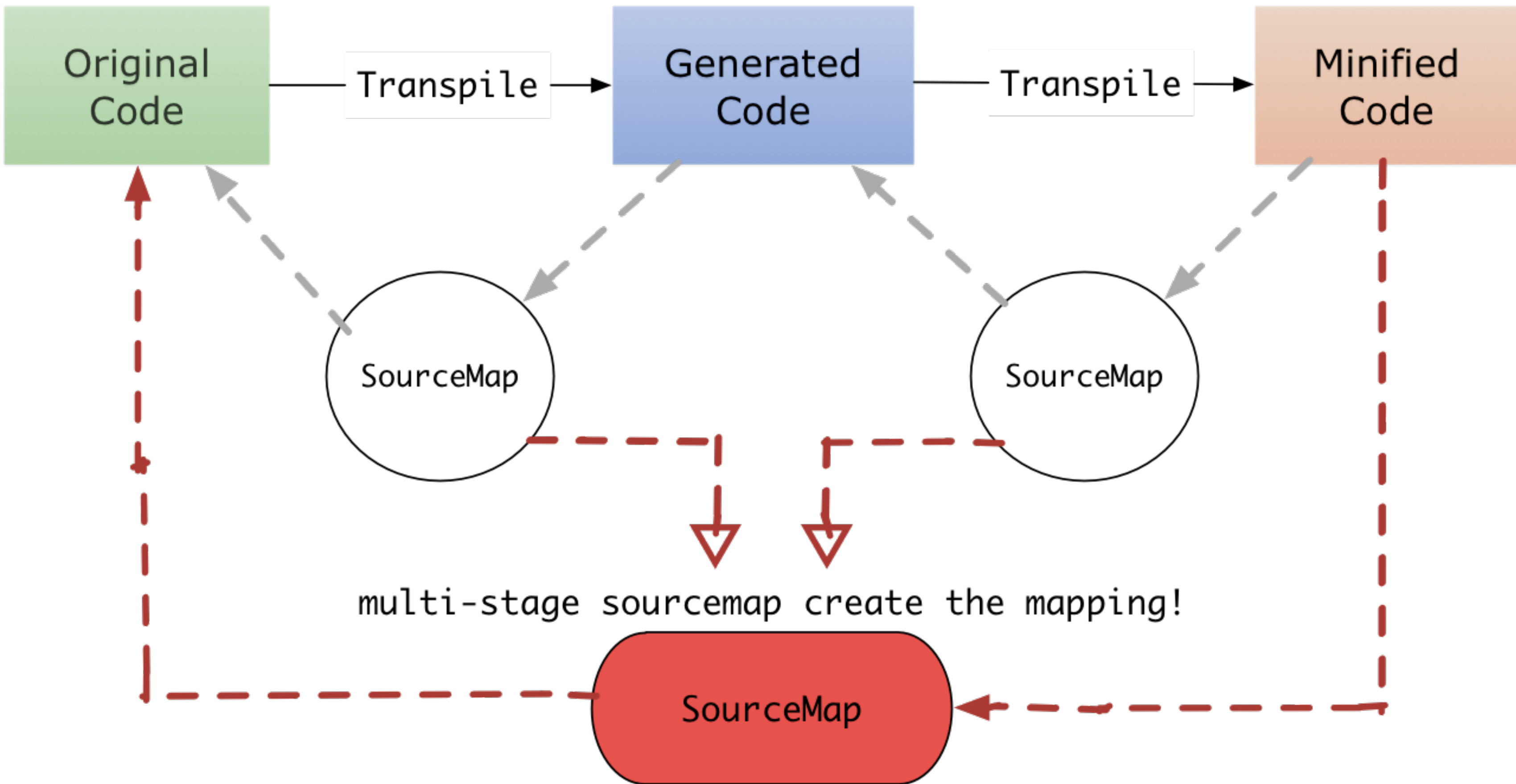


変換後から元のファイルを辿れない

AltJS -> JS -> 圧縮等で起きる問題

解決する方法を考えた

multi-stage-sourcemap



Original Code

Transpile

multi-stage-sourcemap

Generated Code

Transpile

Minified Code

- 中間のSourceMap自体はある(という前提)
- 中間のSourceMap同士を繋いで新たなSourceMapを作成
- => 最初と最後のソースを繋ぐSourceMapを作る

multi-stage sourcemap create the mapping!

SourceMap

仕様書に載ってた

The easy but lossy way is to ignore the intermediate steps in the process for the purposes of debugging, the source location information from the translation is either ignored or the source location information is carried through.

– *Source Map Revision 3 Proposal*

実例

- [twada/power-assert](#)
 - [Release v0.9.0: Release 0.9.0 \(2014-09-02\) multi-stage sourcemaps · twada/power-assert](#)
 - [power-assert 多段 SourceMap 対応の方針](#)
 - [twada/battlefield-sourcemaps](#)
- [mishoo/UglifyJS2](#)
 - [Multi-level Source maps | The CSS Ninja - All things CSS,](#)

spy-js

-- <https://twitter.com/SpyDashJs/status/512503730625732608>

Pull Requestもきたし多分載るのでは

dea and awesome library

b.com/azu/multi-stag ..., thanks

re, will be trying to use it to improv

s

多段SourceMapは戦場

- SourceMapの仕様には**多段で変換された**という情報がない
 - A -> B(a-b.js.map) -> C(b-c.js.map)
 - b-c.js.mapにはAに関する情報は存在しない
- GruntやGulpなどプラグインの責務で変換を行う場合に問題が起きる
 - [twada/battlefield-sourcemaps](#)

Grunt

grunt では上流 SourceMap がどう来るか調べる必要がある。

- js と同一ディレクトリに外部 .map ファイルのパターン
- js ファイル末尾に base64 コメントで付いているパターン

— *power-assert* 多段 SourceMap 対応の方針

Gulp

[floridoo/gulp-sourcemaps](https://github.com/floridoo/gulp-sourcemaps) に対応してればOK

Browserify

独自のチェーンを持っているのでそれに対応する

世代を持つSourceMapを扱う統一イ
ンターフェースがない

SourceMapのまとめ

- 多段SourceMapの対応は最初と最後を繋いだSourceMapを作ることで対応
- 中間結果は捨ててしまう
- 将来的に中間のバージョンも持つような仕組みが仕様に欲しい
- [Beyond Source Maps](#)

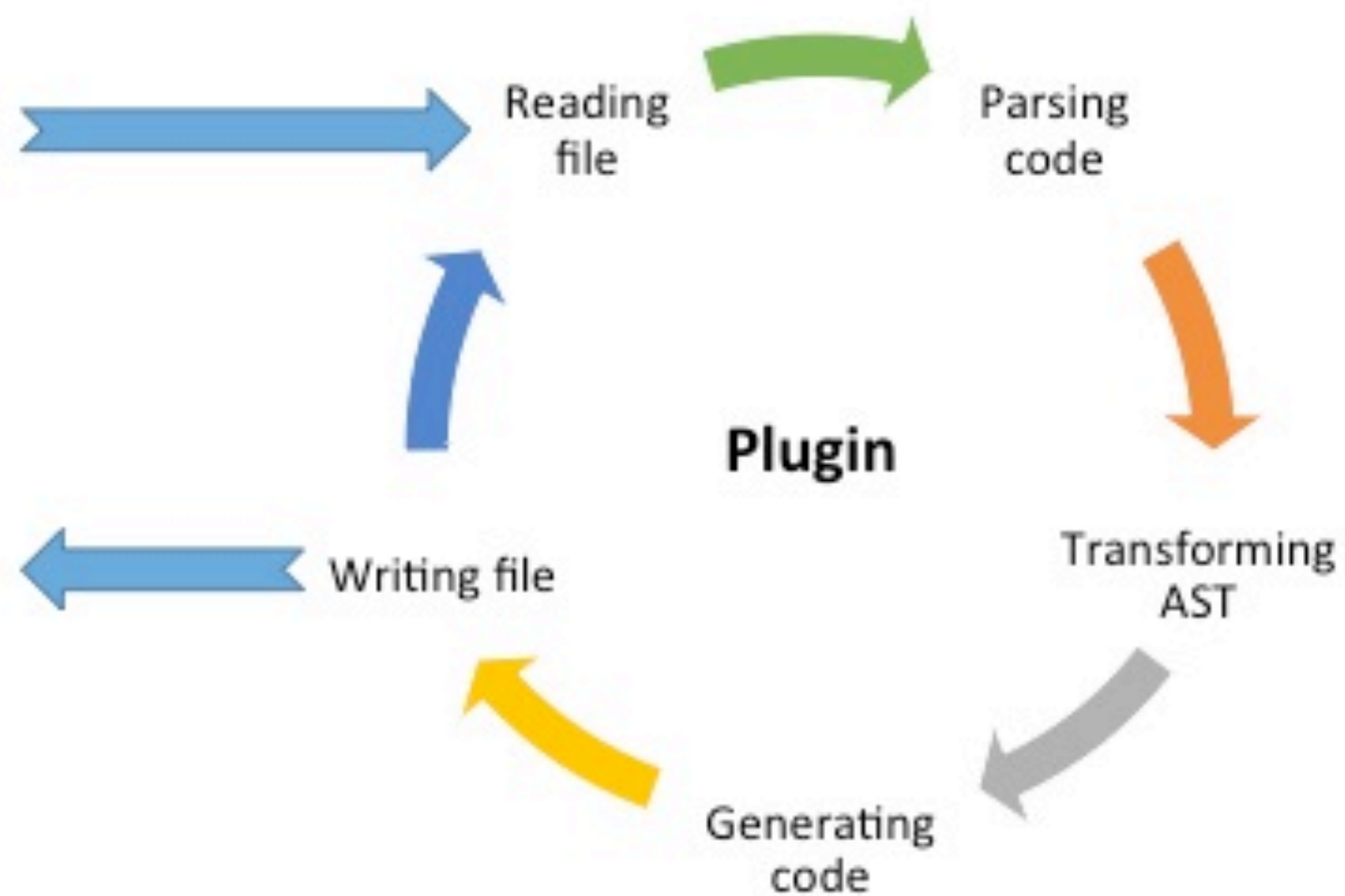
ここまで前置き

統一的なインターフェース(AST)

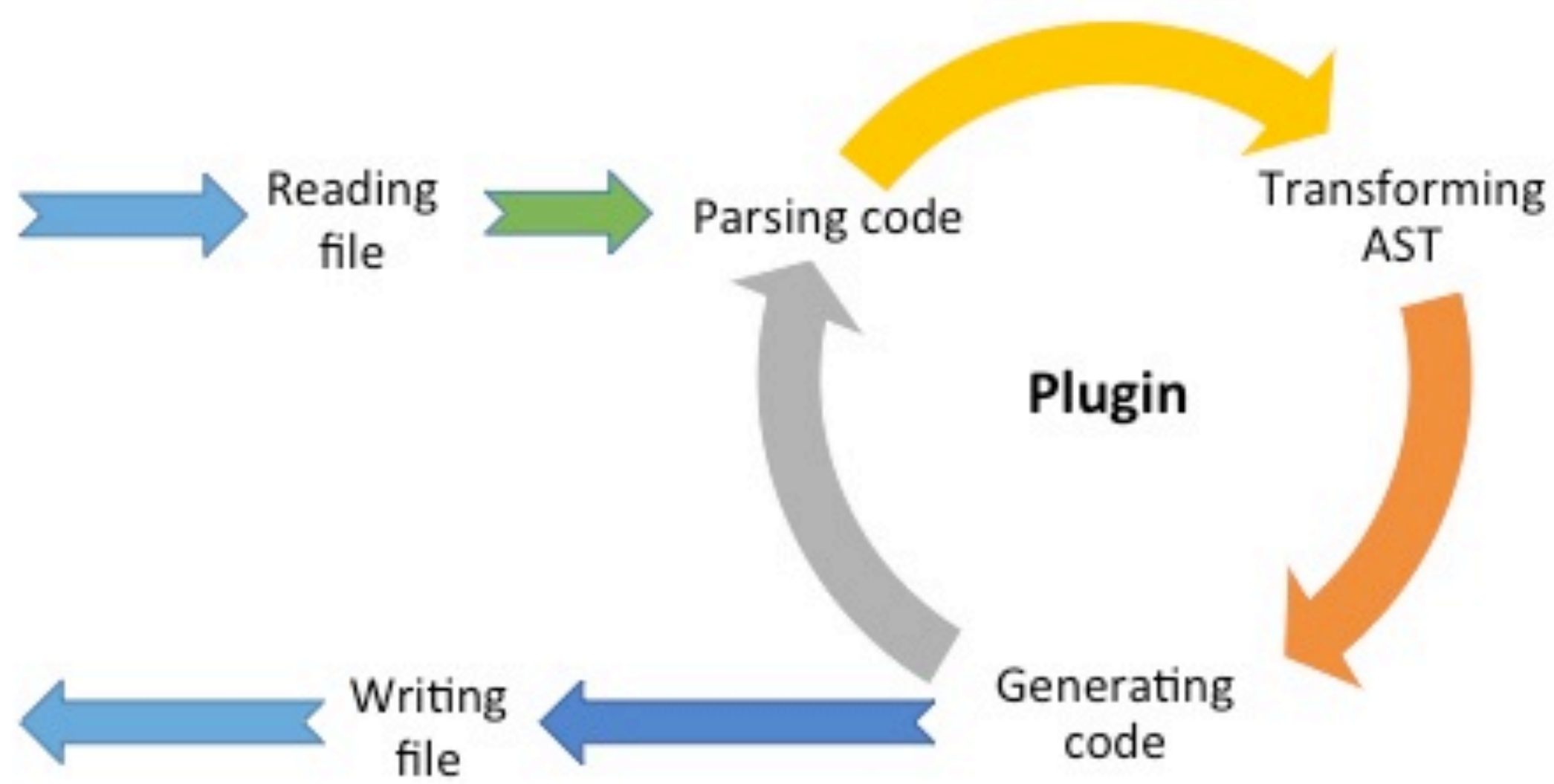
aster - AST-based code builder

- ASTを使ったツールの問題を解決しようとしている
- それぞれのプラグイン(変換)ごとにパースと生成の処理が入っている問題
- ParseとGenerateはとても重たい処理

File-based builders (Grunt)



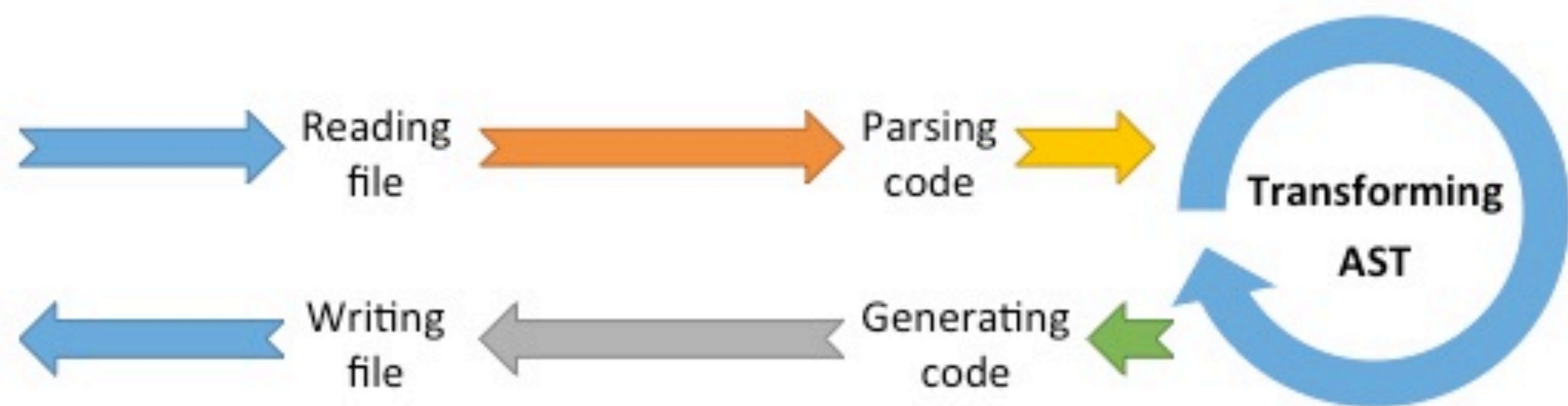
Streaming builders (Gulp, Browserify)



ASTを受け取りASTを返す

- `aster`はASTをInput/Outputするプラグインを使う
- ASTを変換するプラグインはASTを返すインターフェースを持つ
- 効率的なASTの変換が行える

Next logical step



現実

- ASTを使うモジュールがコードを受け取ってコードを返す
- ASTを受け取ってASTを返すAPIをPublicにしてないものが多い
- SourceMapと同じように扱う形式(SourceMap/AST)は決まっている
- それらを連続的に扱う方法については決まってないので自由に破壊されるケースが多い

まとめ:我々の扱う入力/出力

- ソースコード
 - ファイルのコンテンツ
- SourceMap
 - ファイルのコンテンツ と ファイルパス
- AST
 - SpiderMonkey AST

そのた

- [ASTを扱うツールの作り方・Issue #19・efcl/efcl.github.io](#)
- [多段SourceMapの対応方法とライブラリ | Web Scratch](#)