

phpers

SUMMIT

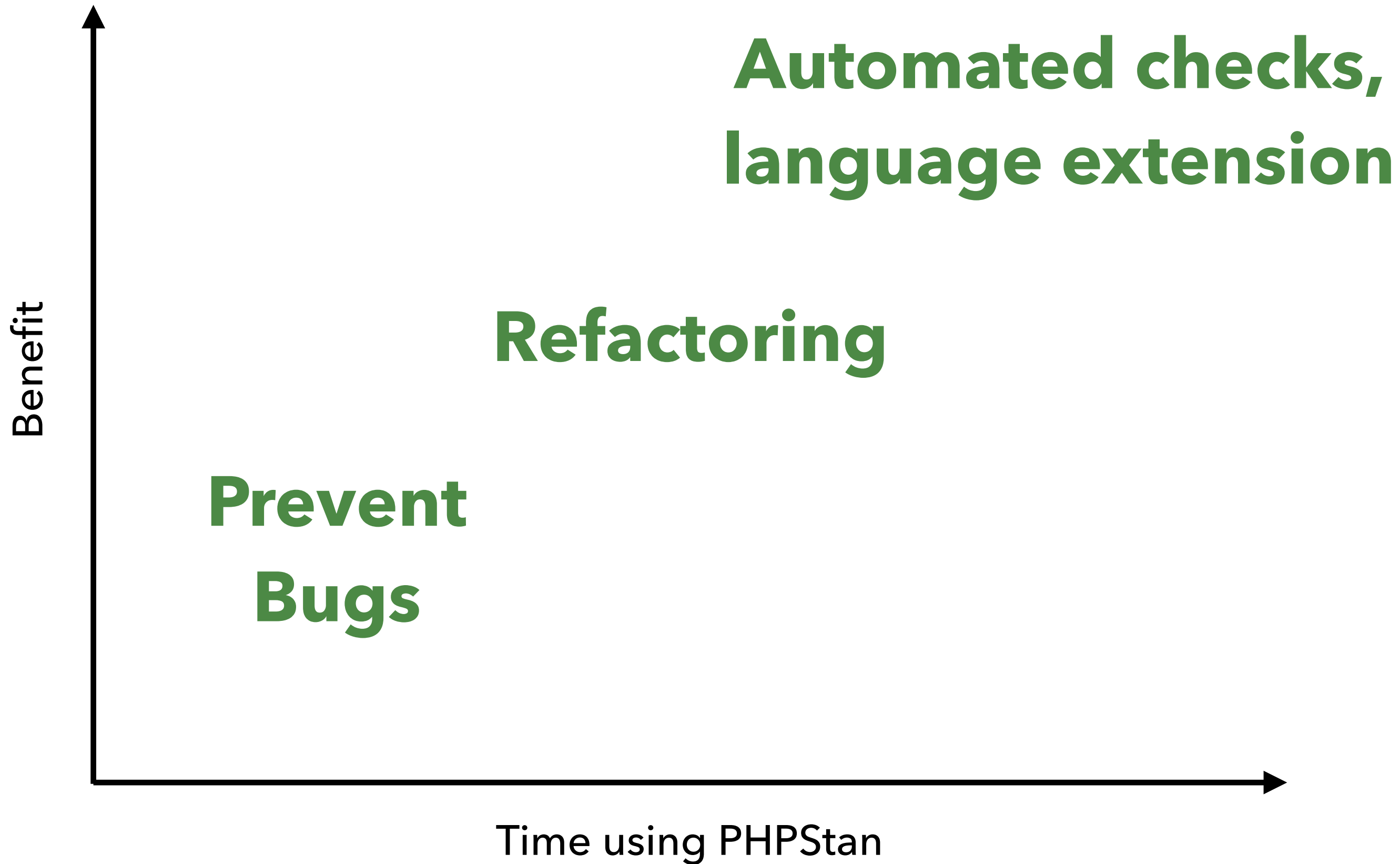
2023

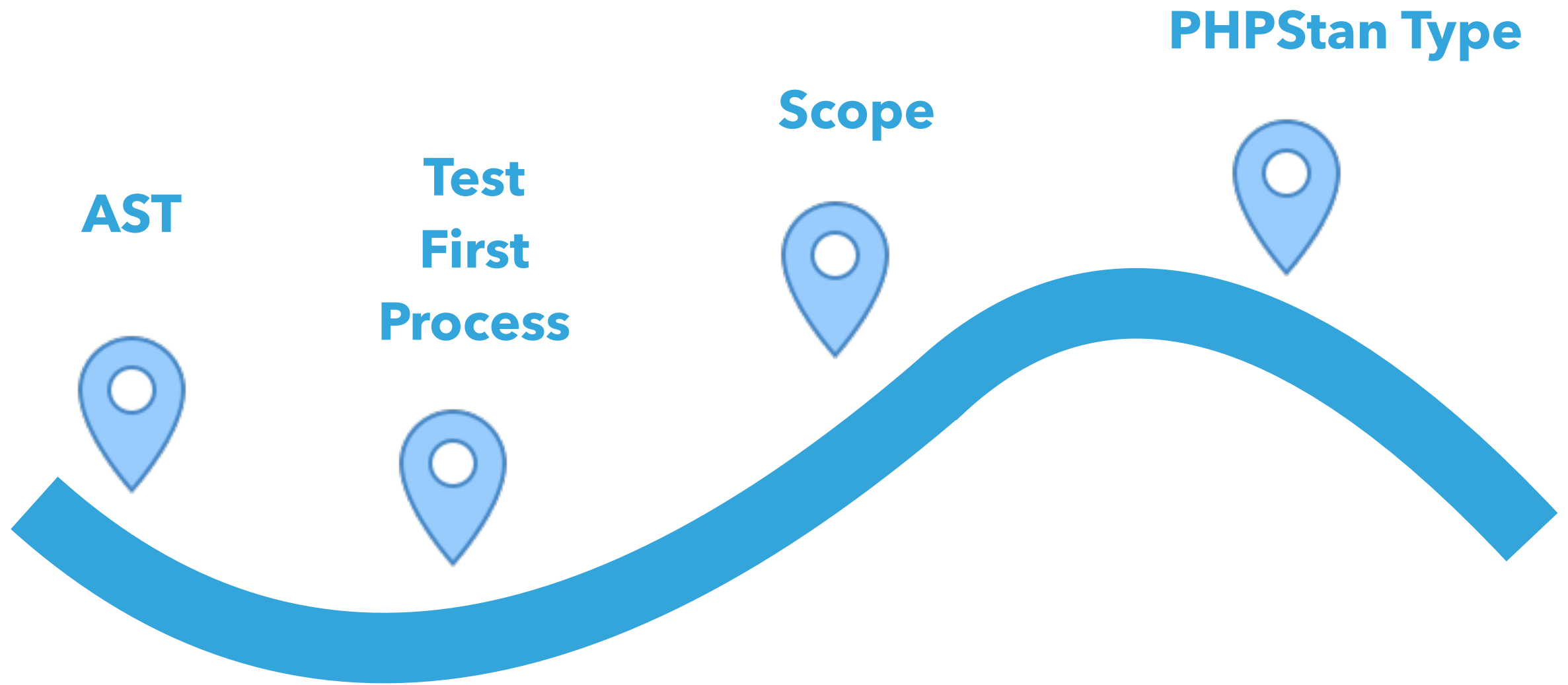
Writing your own custom PHPStan rules

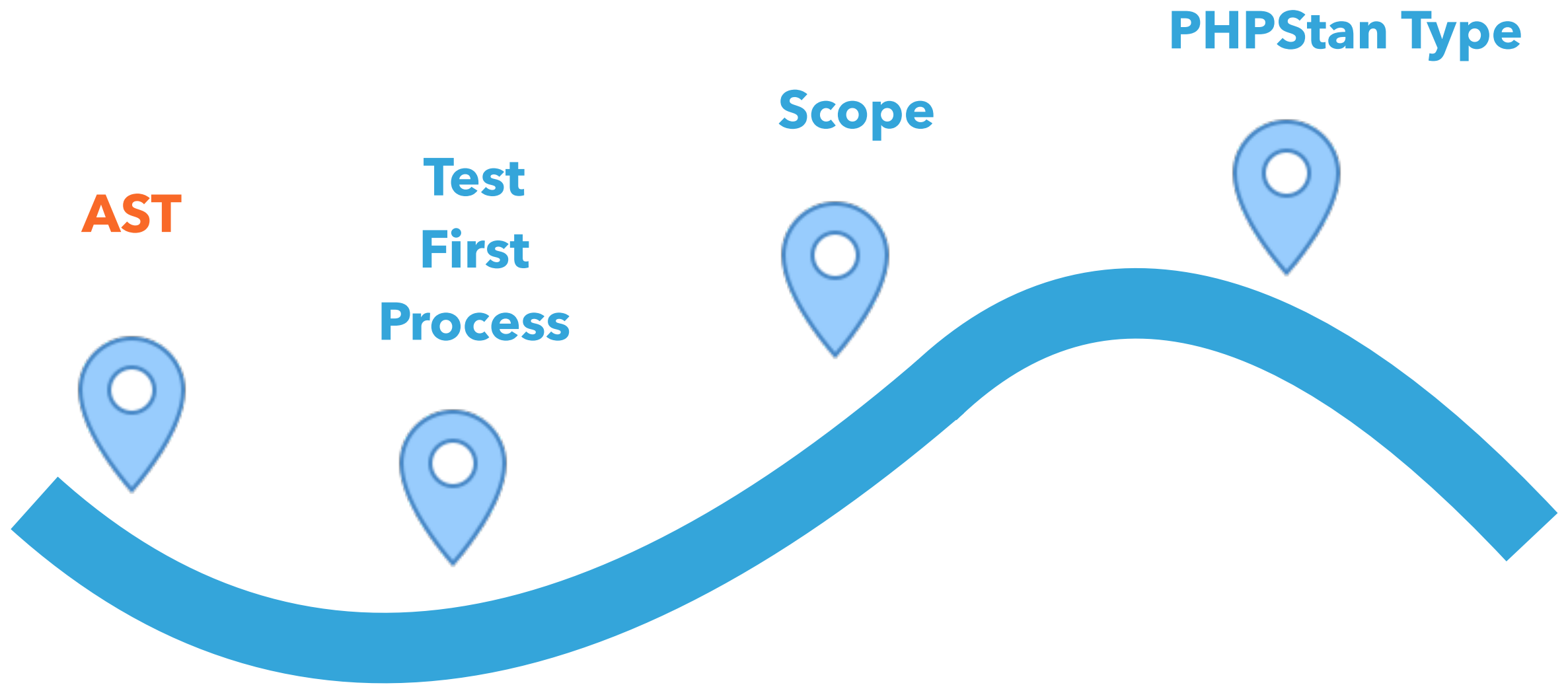
Dave Liddament

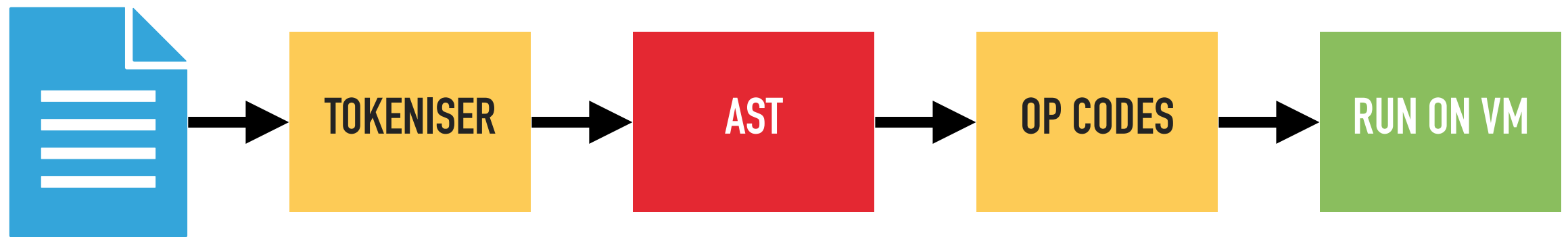
@DaveLiddament

github.com/DaveLiddament/phpstan-rules-tutorial









```
class PersonNotifier
```

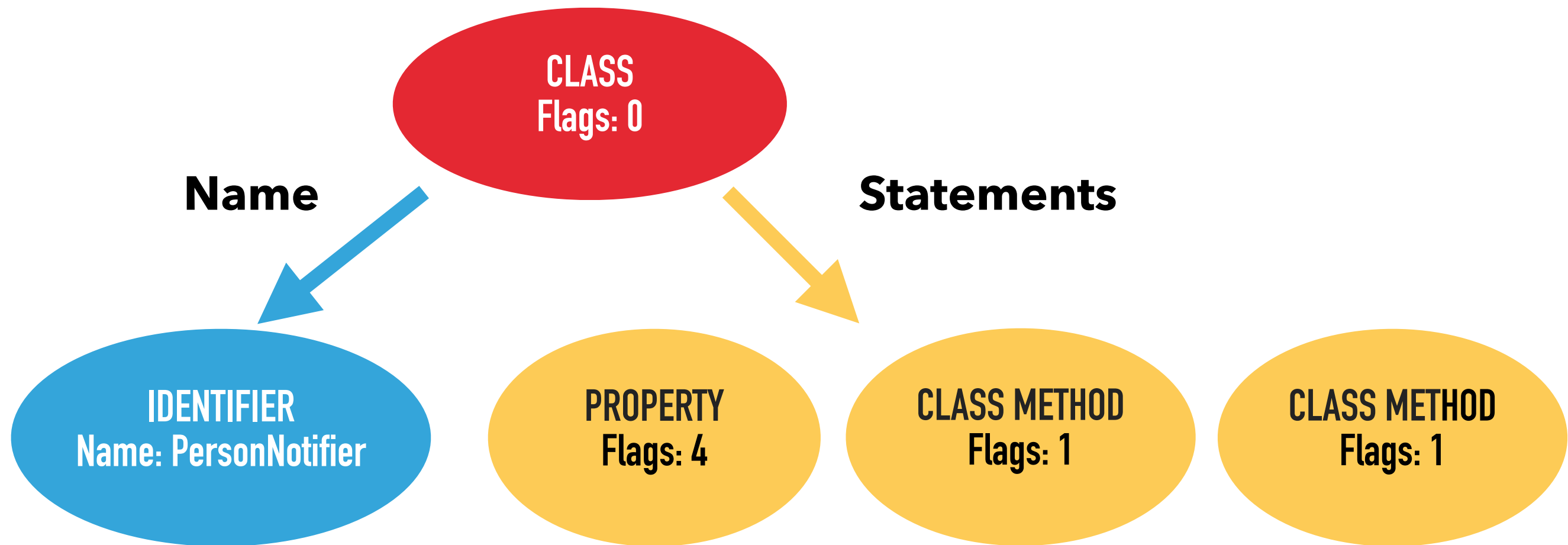
```
{
```

```
private TextMessageSender $sender;
```

```
public function __construct() {...}
```

```
public function notifyPlayer() {...}
```

```
}
```

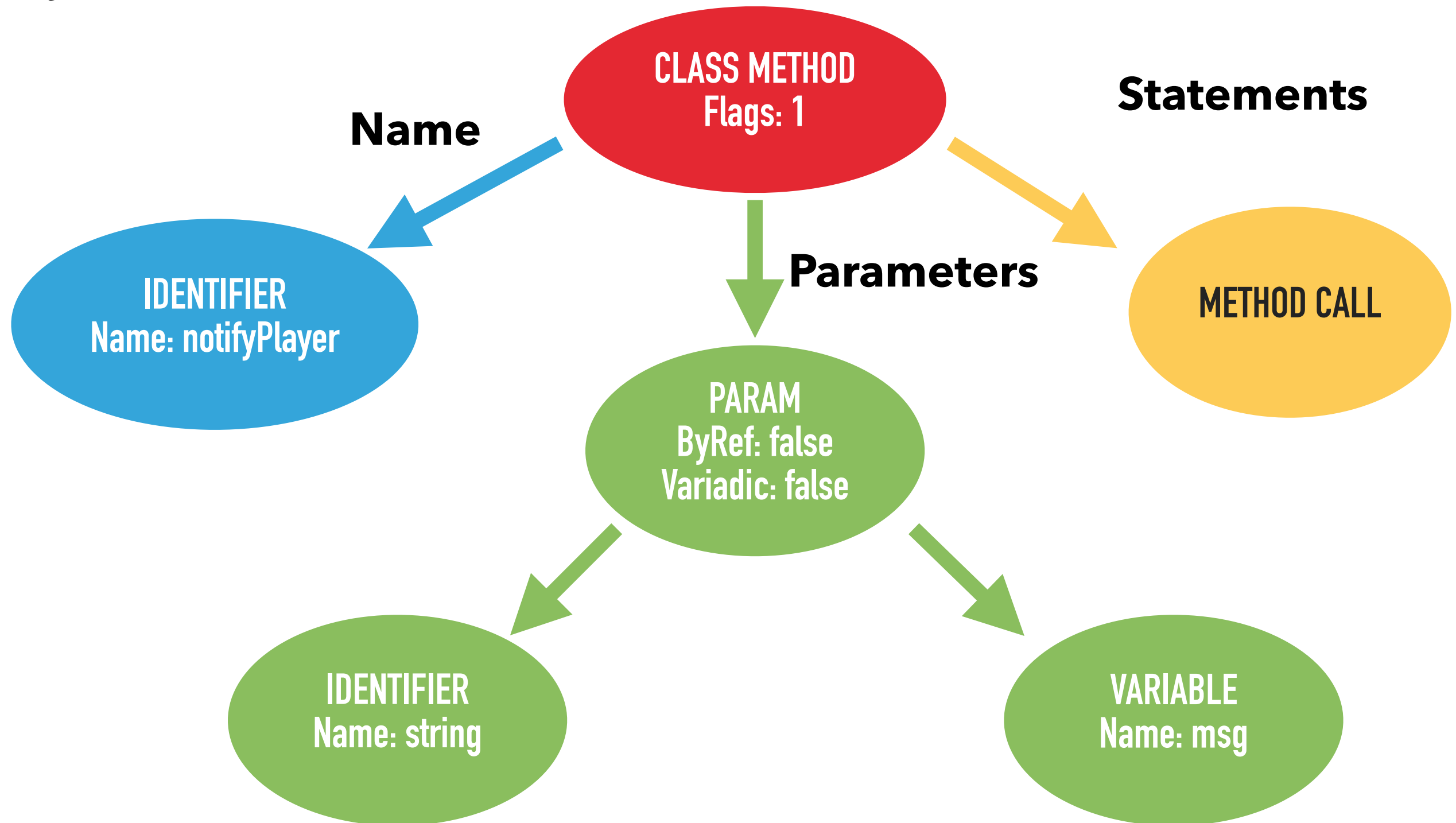


```
public function notifyPlayer (string $msg)
```

```
{
```

```
$this->sender->sendMessage ($msg);
```

```
}
```



https://github.com/nikic/PHP-Parser

nikic / PHP-Parser Public Watch 232 Fork 891

[Code](#) [Issues 44](#) [Pull requests 9](#) [Actions](#) [Wiki](#) [Security](#) [Insights](#)

master 9 branches 80 tags Go to file Add file Code

nikic Bail out on PHP tags in removed code ... ✓ b0edd4c 2 hours ago 🕒 1,526 commits

📁 .github/workflows	Test PHP 8.2 in CI	3 days ago
📁 bin	Add --version flag to php-parse	17 days ago
📁 doc	Fix pretty printing example	17 days ago
📁 grammar	Support readonly before DNF type	3 days ago
📁 lib/PhpParser	Bail out on PHP tags in removed code	2 hours ago
📁 test	Bail out on PHP tags in removed code	2 hours ago
📁 test_old	Avoid repeatedly downloading archive in run-php-src.sh	2 months ago
📁 tools	Add tools/ directory	10 days ago
📄 .editorconfig	[PHP 8.1] Add support for enums (#758)	17 months ago
📄 .gitattributes	Add CONTRIBUTING.md	23 days ago
📄 .gitignore	gitignore: add phpunit test cache	3 years ago
📄 .php-cs-fixer.dist.php	Also format the grammar directory	23 days ago
📄 CHANGELOG.md	Release PHP-Parser 5.0.0-alpha1	17 days ago
📄 CONTRIBUTING.md	Add CONTRIBUTING.md	23 days ago
📄 LICENSE	Corrected license text	2 years ago
📄 README.md	Partial documentation update	17 days ago
📄 UPGRADE-1.0.md	Fix typos	8 years ago

About

A PHP parser written in PHP

[php](#) [parser](#) [static-analysis](#) [ast](#)

[Readme](#)
[BSD-3-Clause license](#)
★ 15.7k stars
👁 232 watching
🍴 891 forks

Releases 76


📦 **PHP-Parser 4.15.1** Latest
18 days ago

[+ 75 releases](#)

Packages

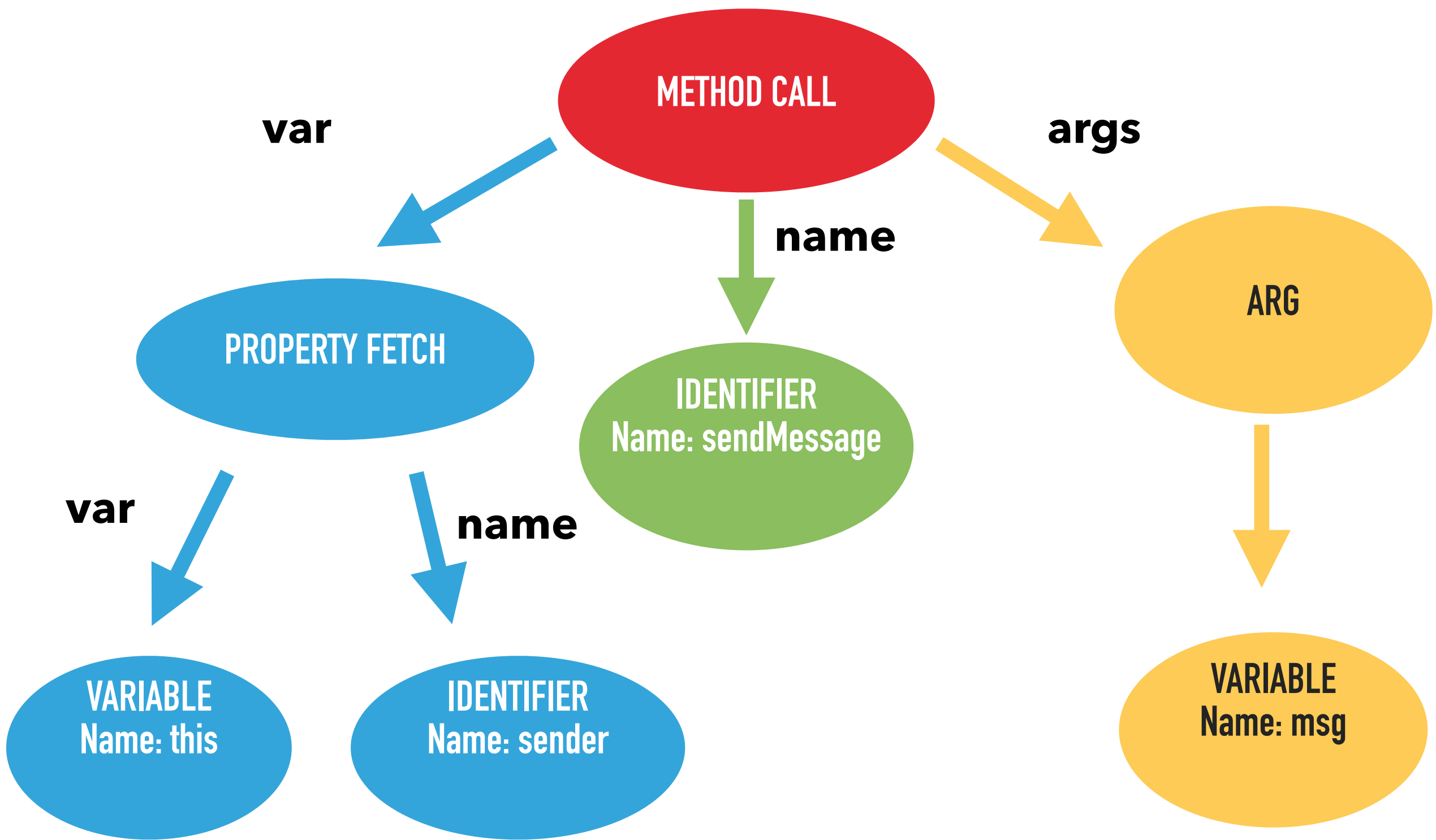
No packages published

Used by 1.5m

 + 1,486,143

Contributors 123


```
$this->sender -> sendMessage ($msg) ;
```



```
class MethodCall extends \PhpParser\Node\Expr\CallLike
{
```

```
/** @var Expr Variable holding object */
public $var;
```

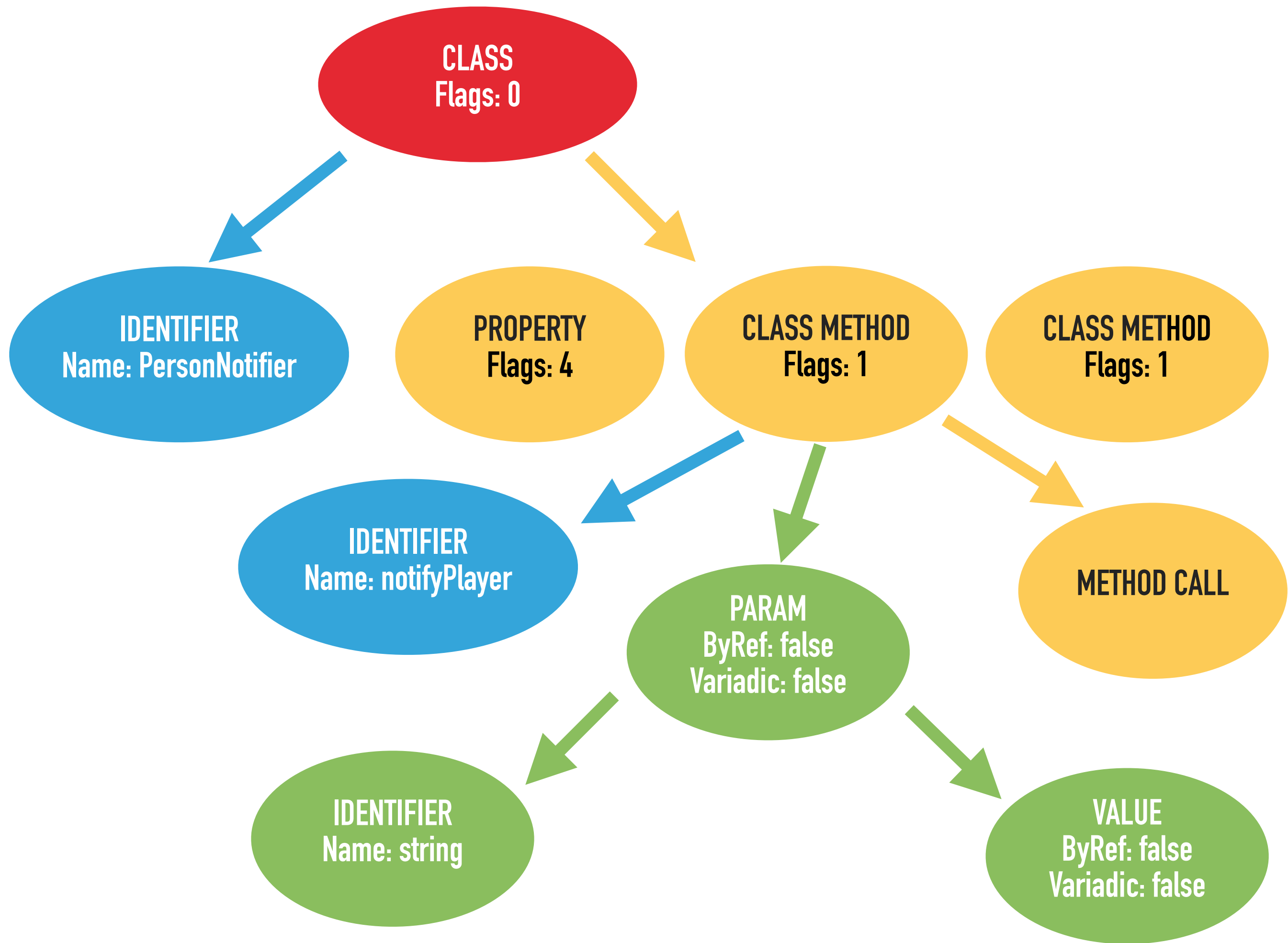
```
/** @var Identifier|Expr Method name */
public $name;
```

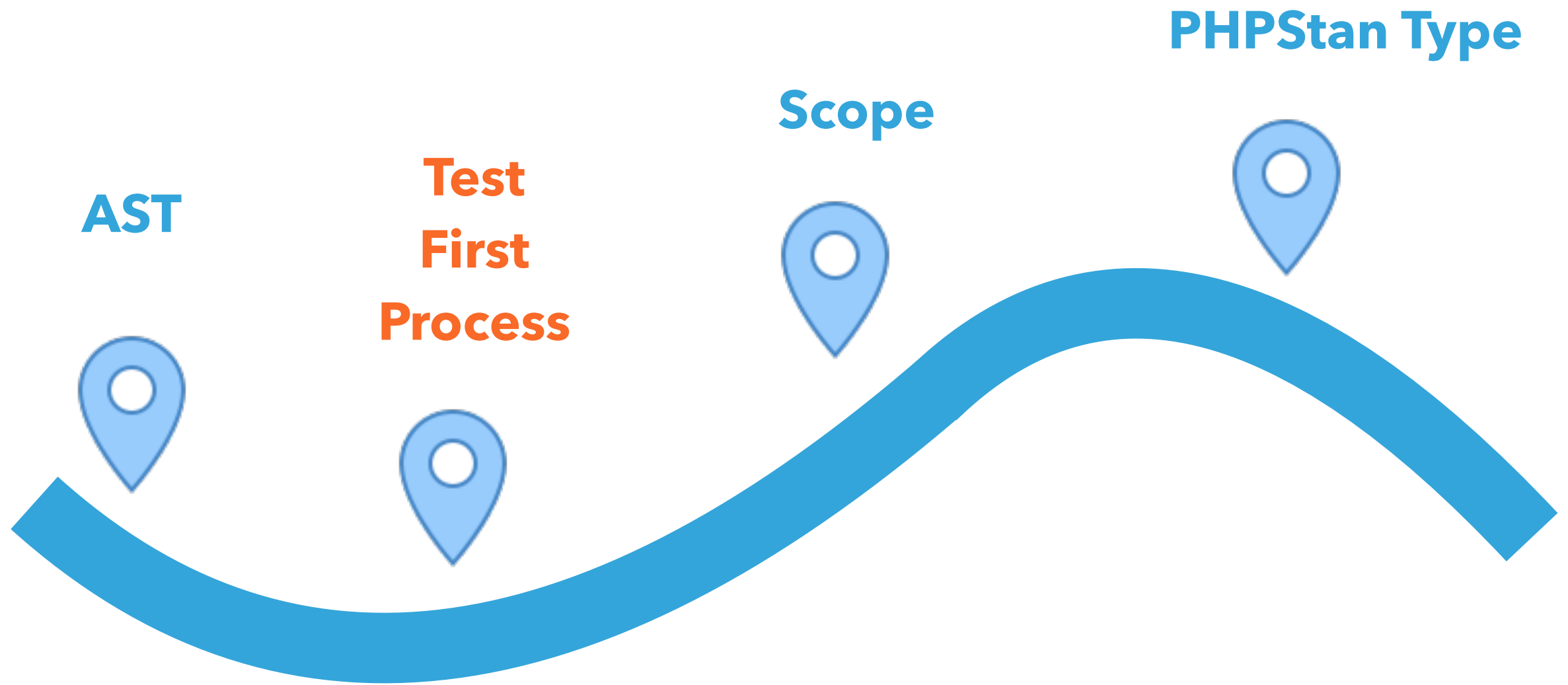
```
/** @var array<Arg|VariadicPlaceholder> Arguments */
public $args;
```

```
// Rest of class ...
```

```
$this->sender -> sendMessage ($msg);
```

- ▶ PHP code can be represented by an AST
- ▶ Different types of Node
- ▶ Nodes contain information
- ▶ Each type of node has different information



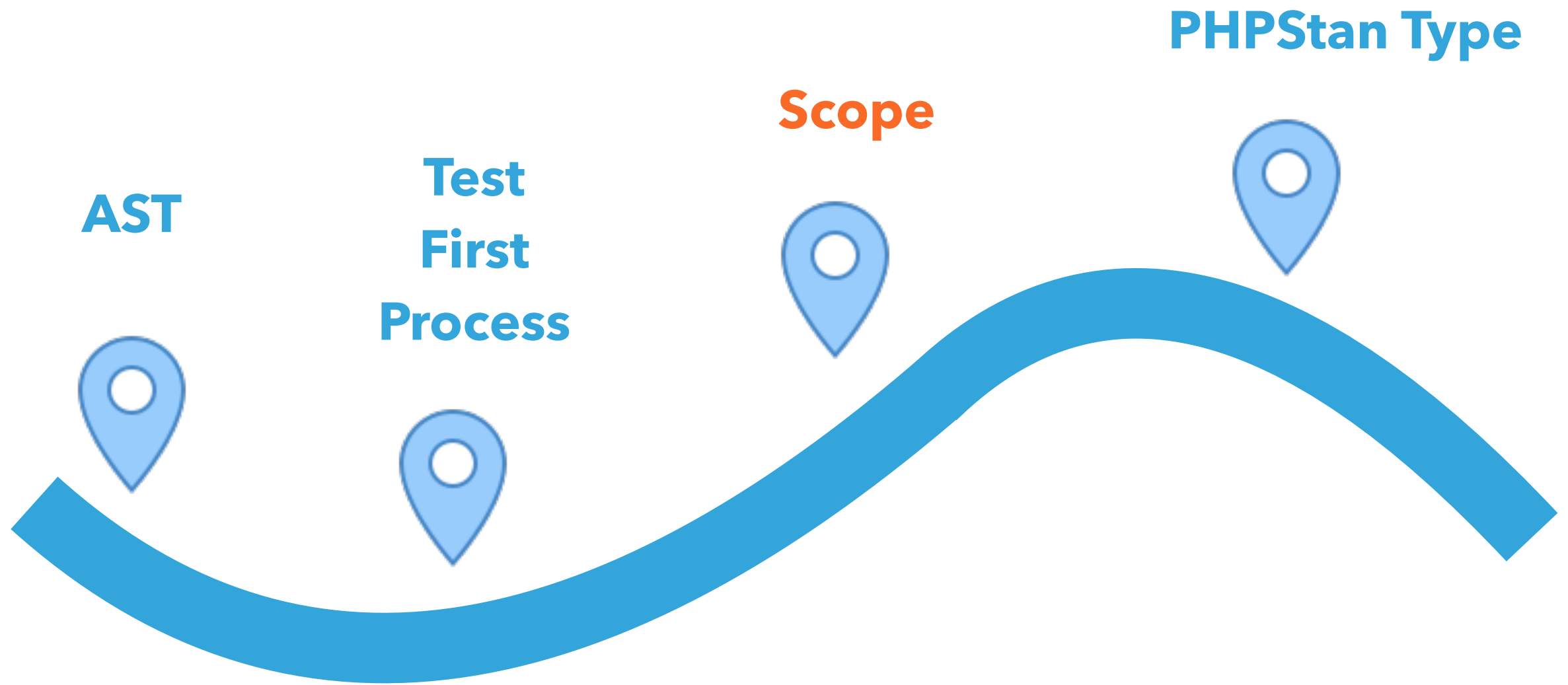


```
interface Rule
{

    public function getNodeTypes() : string;

    /**
     * @return (string|RuleError)[] errors
     */
    public function processNode(
        \PhpParser\Node $node,
        \PHPStan\Analyser\Scope $scope
) : array;

}
```



Information for current AST node

\$scope

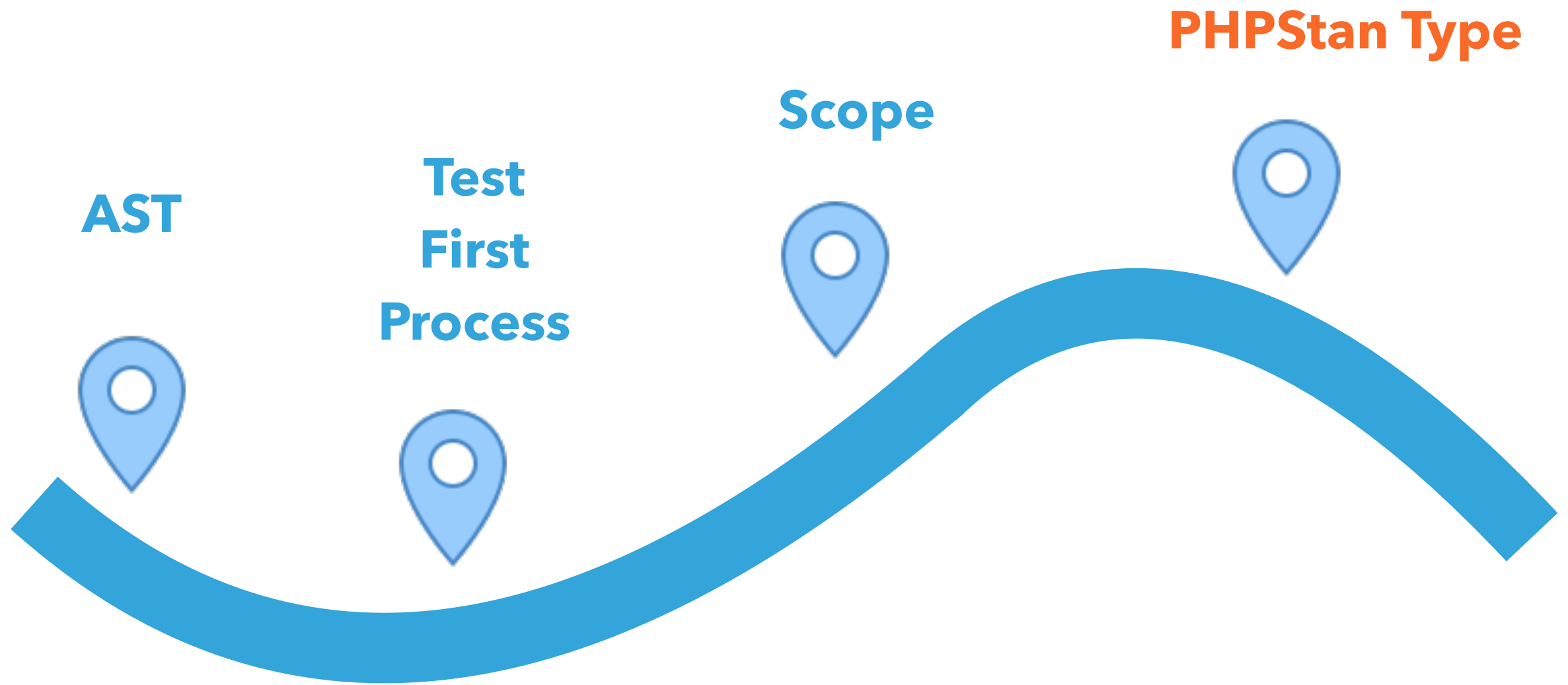
->getFile()

->getNamespace()

->getClassReflection()

->getFunction()

->getType()



```
$scope->getType (  
    Expr $expression,  
) : PHPStan\Type\Type
```

Type offers methods including:

\$type

->getReferencedClasses(): array;

->isBoolean(): TrinaryLogic

->isTrue(): TrinaryLogic

->isString(): TrinaryLogic

->hasMethod(): TrinaryLogic

->isSuperTypeOf(\$type): TrinaryLogic

...

TrinaryLogic

```
$isString = $type->isString();  
  
if ($isString->yes()) {  
    // $type is a string  
}  
else if ($isString->maybe()) {  
    // $type might be string  
    // E.g. $type could be mixed, or null|string  
}  
else {  
    // $type definitely not a string.  
    // E.g. $type could be int  
}
```

```
function takesValue(mixed $value): void
{
    process($value);    PHPStan\Type\MixedType

    if (!is_bool($value)) return;

    process($value);    PHPStan\Type\BooleanType

    if ($value === false) return;

    process($value);    PHPStan\Type\ConstantBooleanType
}
```

```
function takesValue(mixed $value): void
{
```

```
    process($value);    Type::isBoolean()    maybe
                       Type::isFalse()    maybe
```

```
    if (!is_bool($value)) return;
```

```
    process($value);    Type::isBoolean()    yes
                       Type::isFalse()    maybe
```

```
    if ($value === false) return;
```

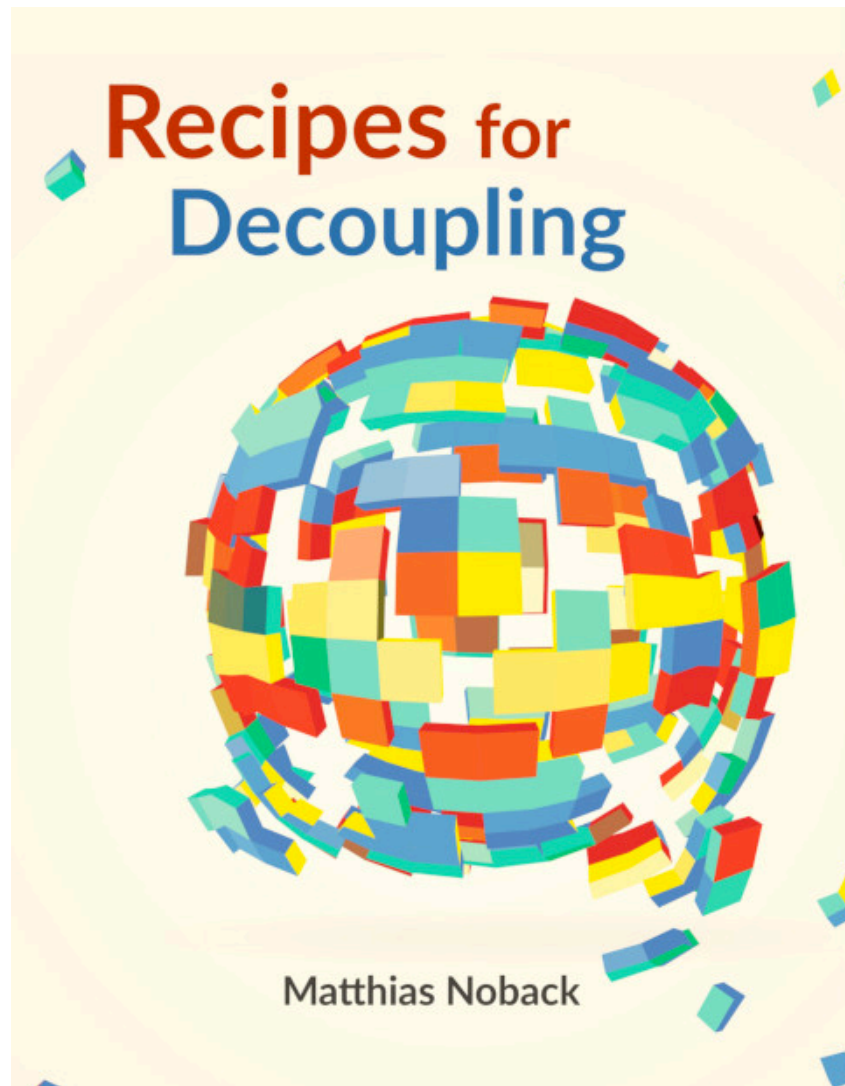
```
    process($value);    Type::isBoolean()    yes
                       Type::isFalse()    no
}
```

Does a type represent an object?

```
$testCase = new ObjectType(TestCase::class);  
  
if ($testCase->isSuperTypeOf($otherType)->yes())  
{  
    // $otherType is of type TestCase  
}
```

Further information

<https://phpstan.org/developing-extensions/rules>

This is a screenshot of a YouTube video player. The video title is 'PHP UK Conference 2018 - James Titcumb - Climbing the Abstract Syntax Tree'. The video content shows a presentation slide titled 'The PHP Parser' with the code `zend_language_parser.y` highlighted. The slide also features a red play button icon, a Twitter handle '@asgrim', and the name 'Roave'. In the bottom right corner of the video frame, the speaker's name 'James Titcumb' and the video title 'Climbing the Abstract Syntax Tree' are displayed. The video player interface includes 'Watch later' and 'Share' buttons in the top right corner and a 'Watch on YouTube' button in the bottom left corner.