Language-side Foreign Function Interfaces with NativeBoost

IWST 2013

Camillo Bruni, Luc Fabresse, Stéphane Ducasse, Igor Stasenko

Outline

Context

Language

Virtual Machine (VM)

Cairo

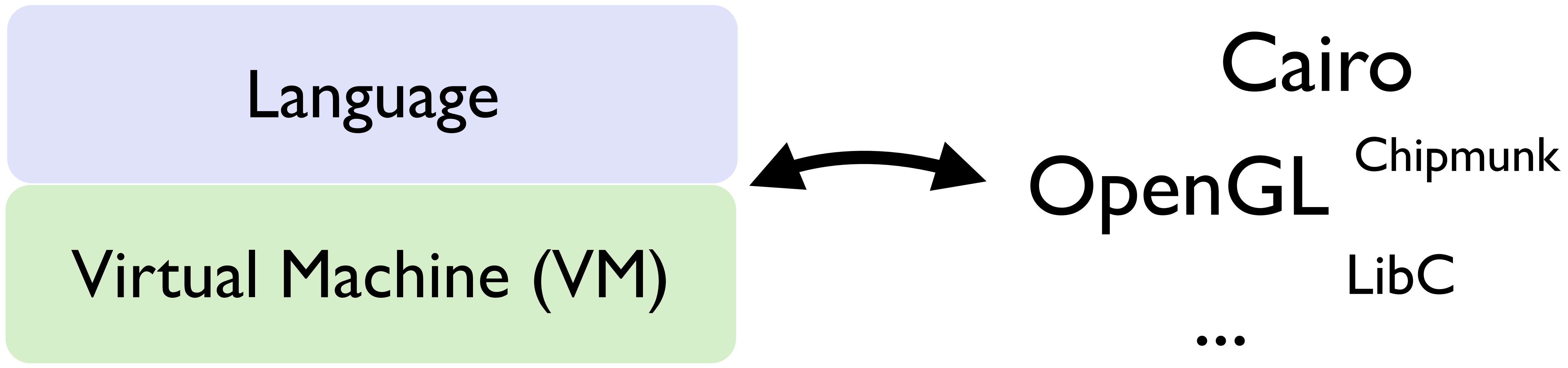
OpenGL

Chipmunk

LibC

...

Context



How to interact with external libraries?

Existing Solutions

- Language-side Library
- VM Extension
- VM Plugin
- Foreign Function Interface
    - VM-level
    - Language-level

Existing Solutions

- ✗ Language-side Library costly
- VM Extension
- VM Plugin
- Foreign Function Interface
  - VM-level
  - Language-level

Existing Solutions

- ✗ Language-side Library
- ✗ VM Extension
- ✗ VM Plugin
- Foreign Function Interface
  - VM-level
  - Language-level

costly

low-level

Existing Solutions

- ✗ Language-side Library — costly
- ✗ VM Extension — low-level
- ✗ VM Plugin
- ✓ Foreign Function Interface
  - ✗ VM-level — fast
  - ✓ Language-level — slow

NativeBoost

A *language-side*

and *fast*

FFI implementation

Language-side

- Extensible
- Easy to use
  - no VM code needed
  - no low-level code (C wrapper) needed

Fast

Transparent

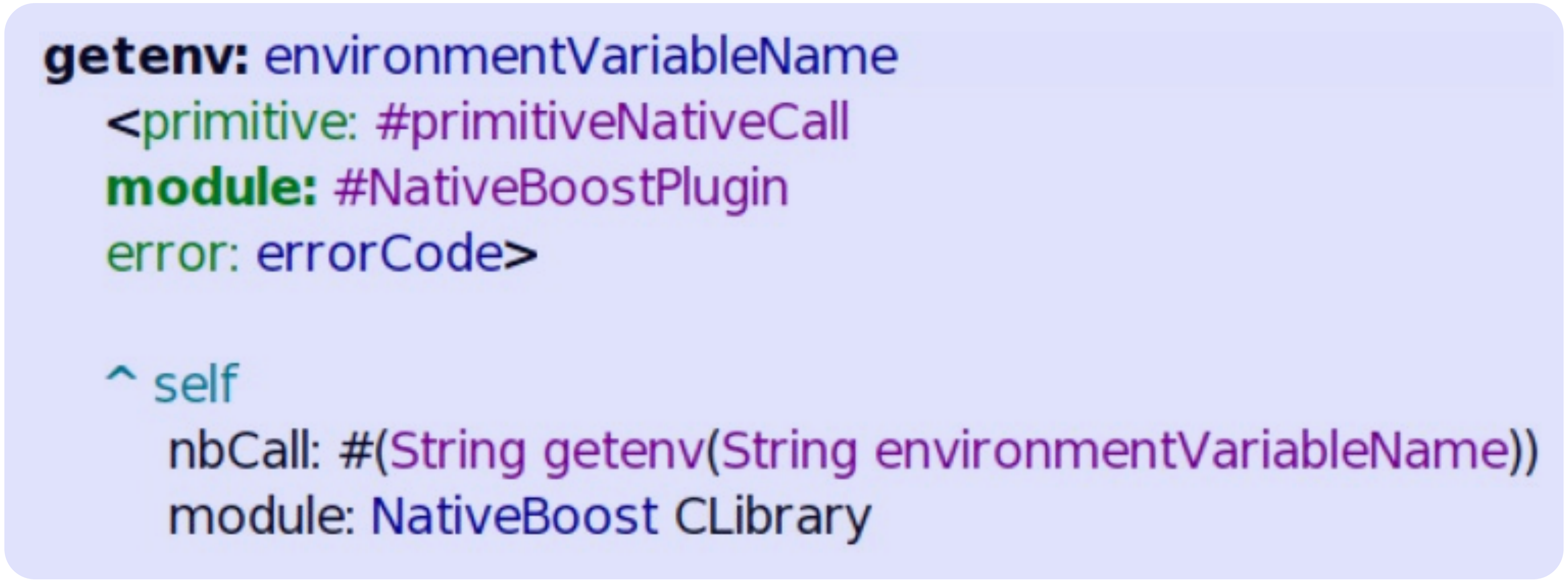
generation of Assembly code from the language-side

NativeBoost Example

```
char* getenv(const char*)
```

NativeBoost Example

```
char* getenv(const char*)
```

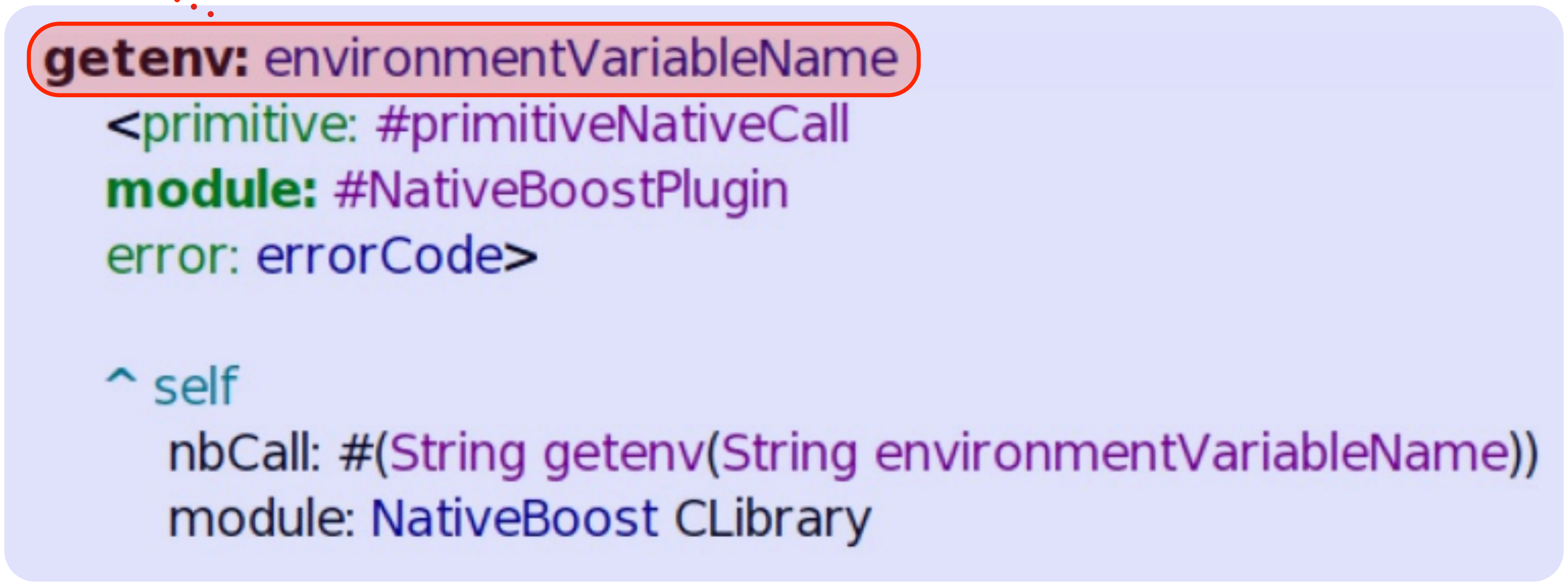


```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

NativeBoost Example

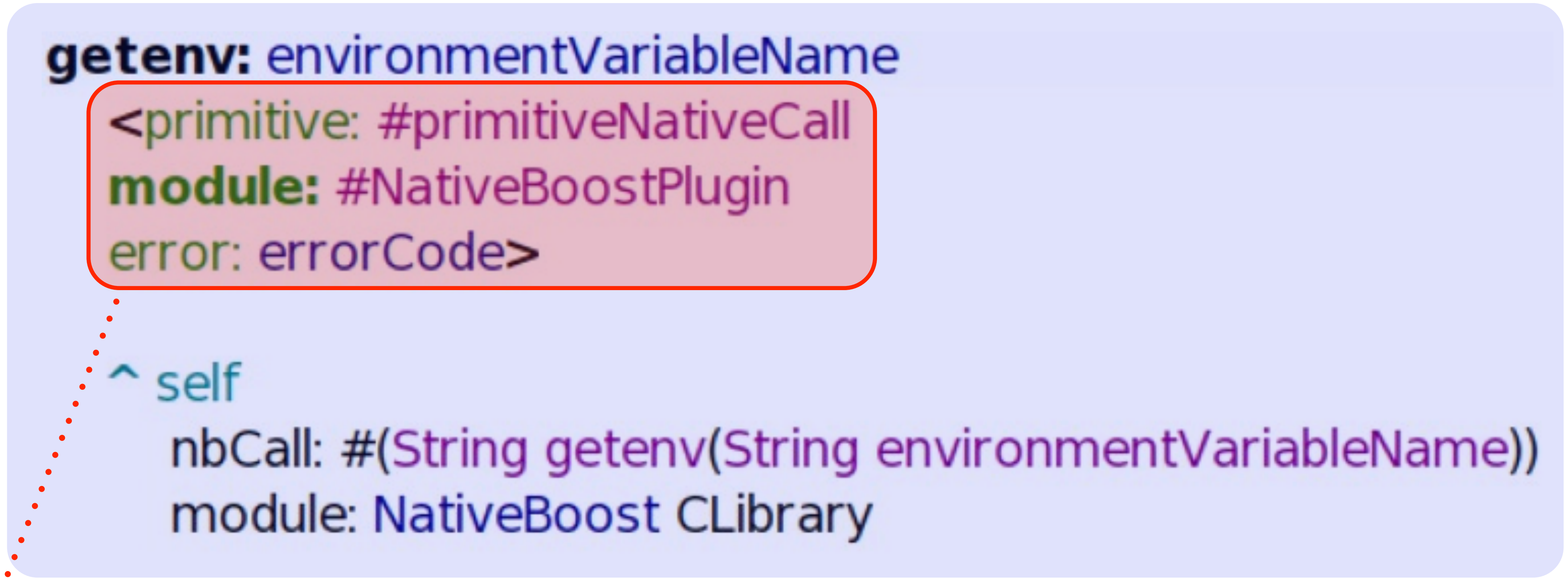
Regular Smalltalk method
with one argument



```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

NativeBoost Example

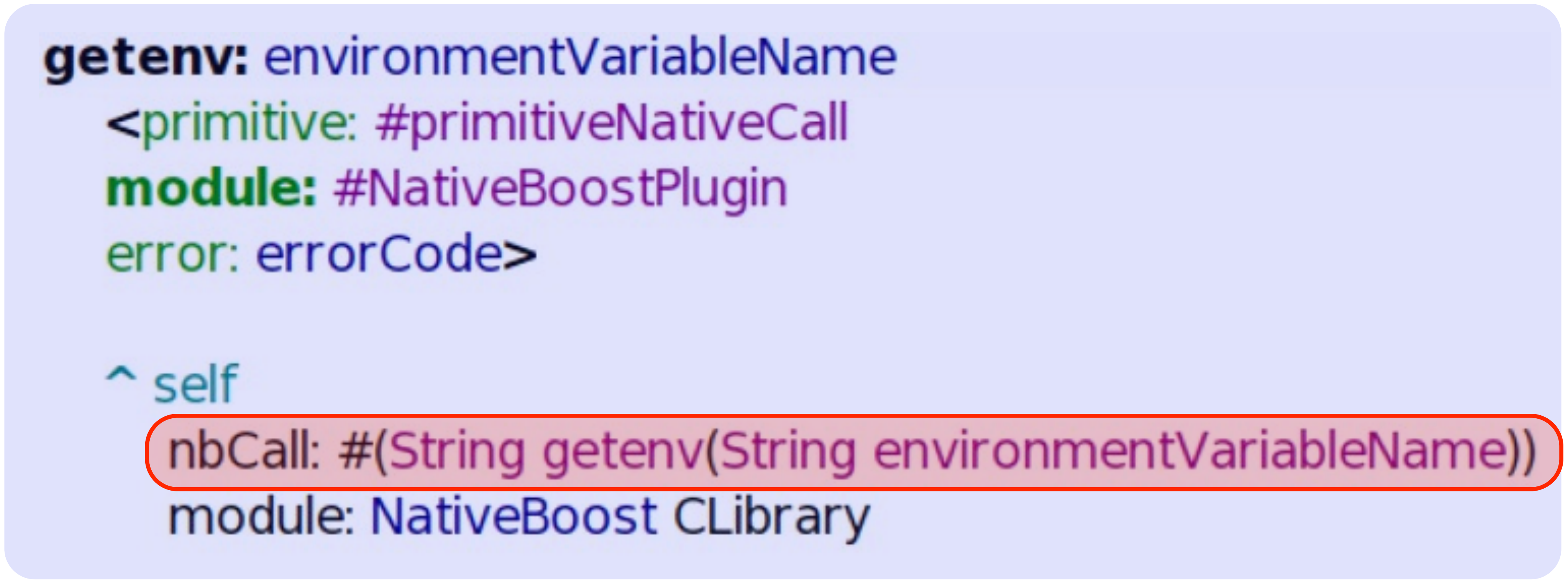


```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

A pragma indicating that #primitiveNativeCall of #NativeBoost plugin should be executed when this method is executed

NativeBoost Example

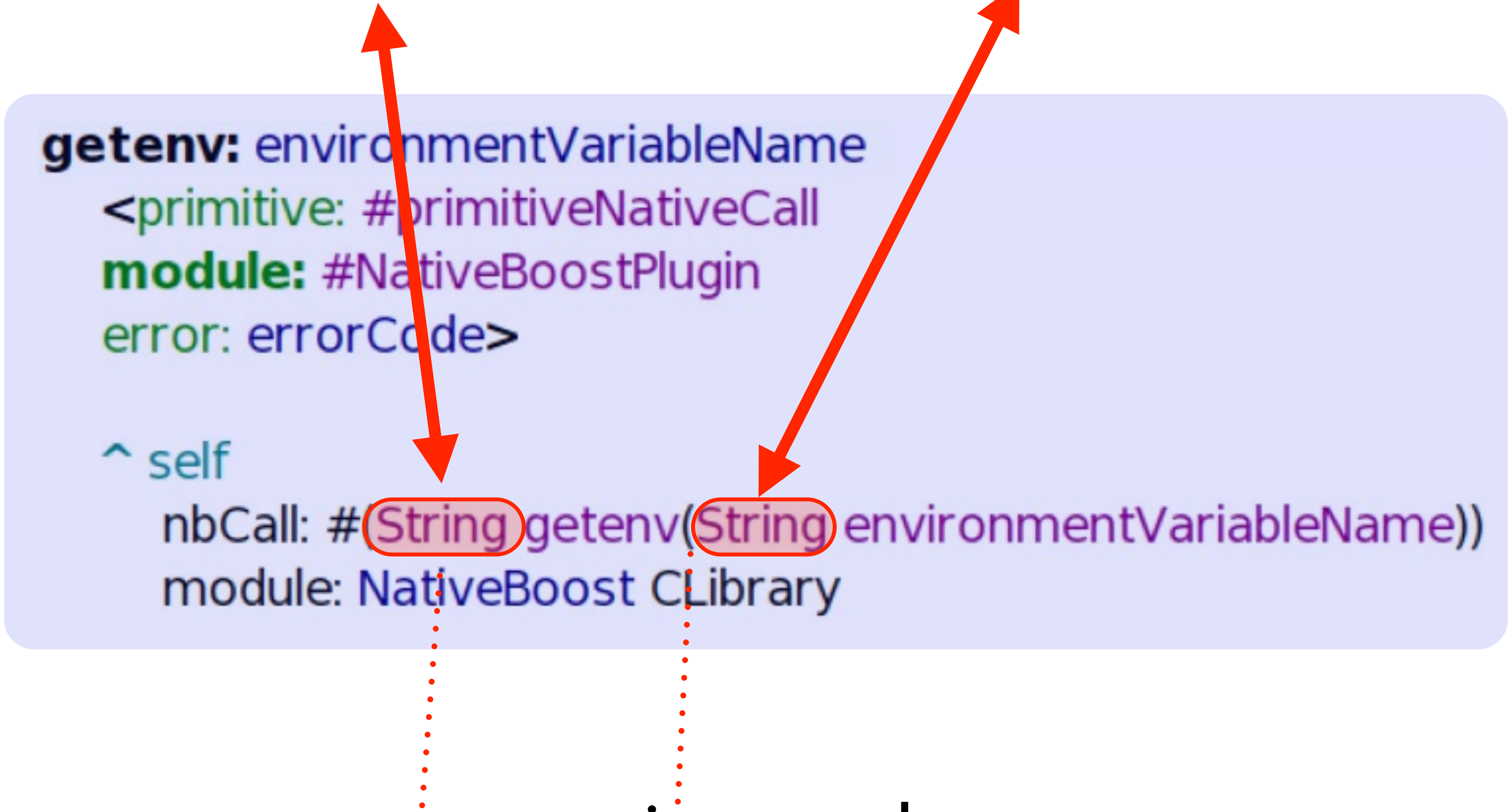


```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

NativeBoost Example

```
char* getenv(const char*)
```



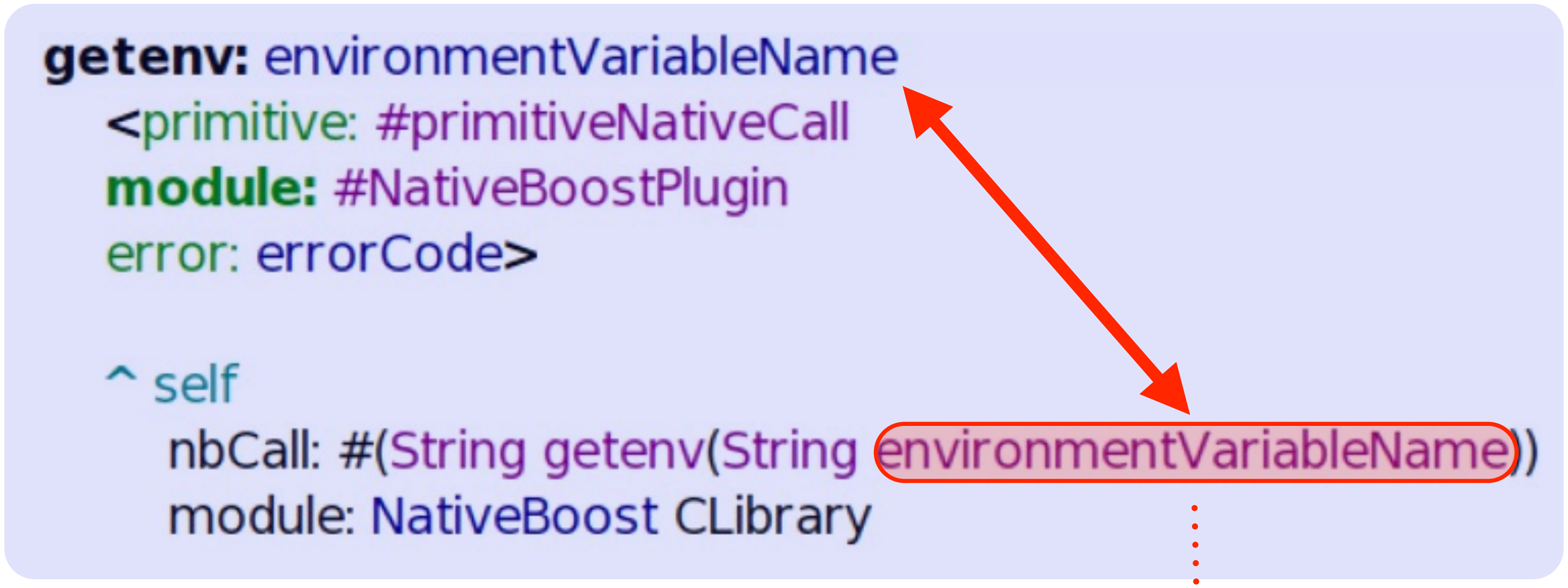
```
getenv: environmentVariableName
   <primitive: #primitiveNativeCall
   module: #NativeBoostPlugin
   error: errorCode>

   ^ self
      nbCall: #(String getenv(String environmentVariableName))
      module: NativeBoost CLibrary
```

types annotation used

to generate **marshalling** code

NativeBoost Example

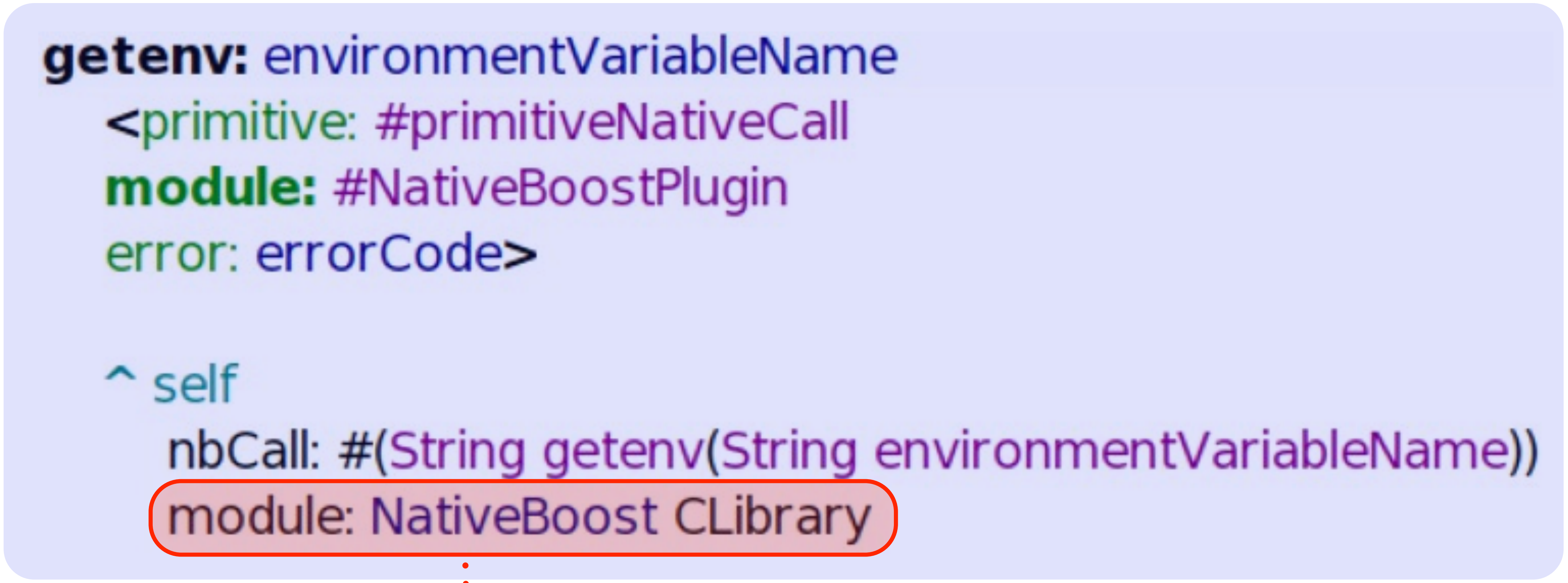


```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

the value to be passed
when calling out

NativeBoost Example



```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

the external library address in
which the function is looked up

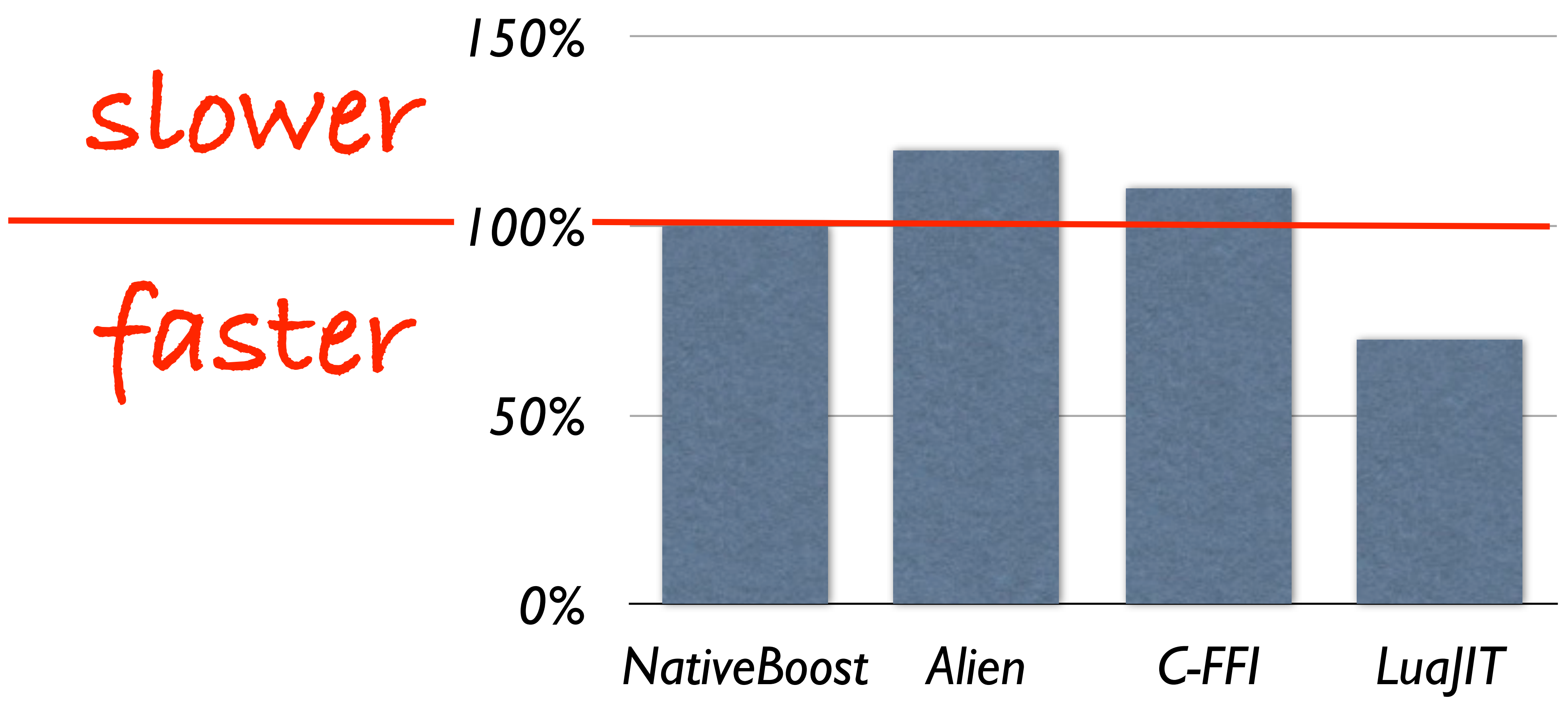
Speed Comparisons

- NativeBoost
- Alien FFI
- C-FFI
- LuaJIT

- Callouts
- Marshalling
- Callbacks

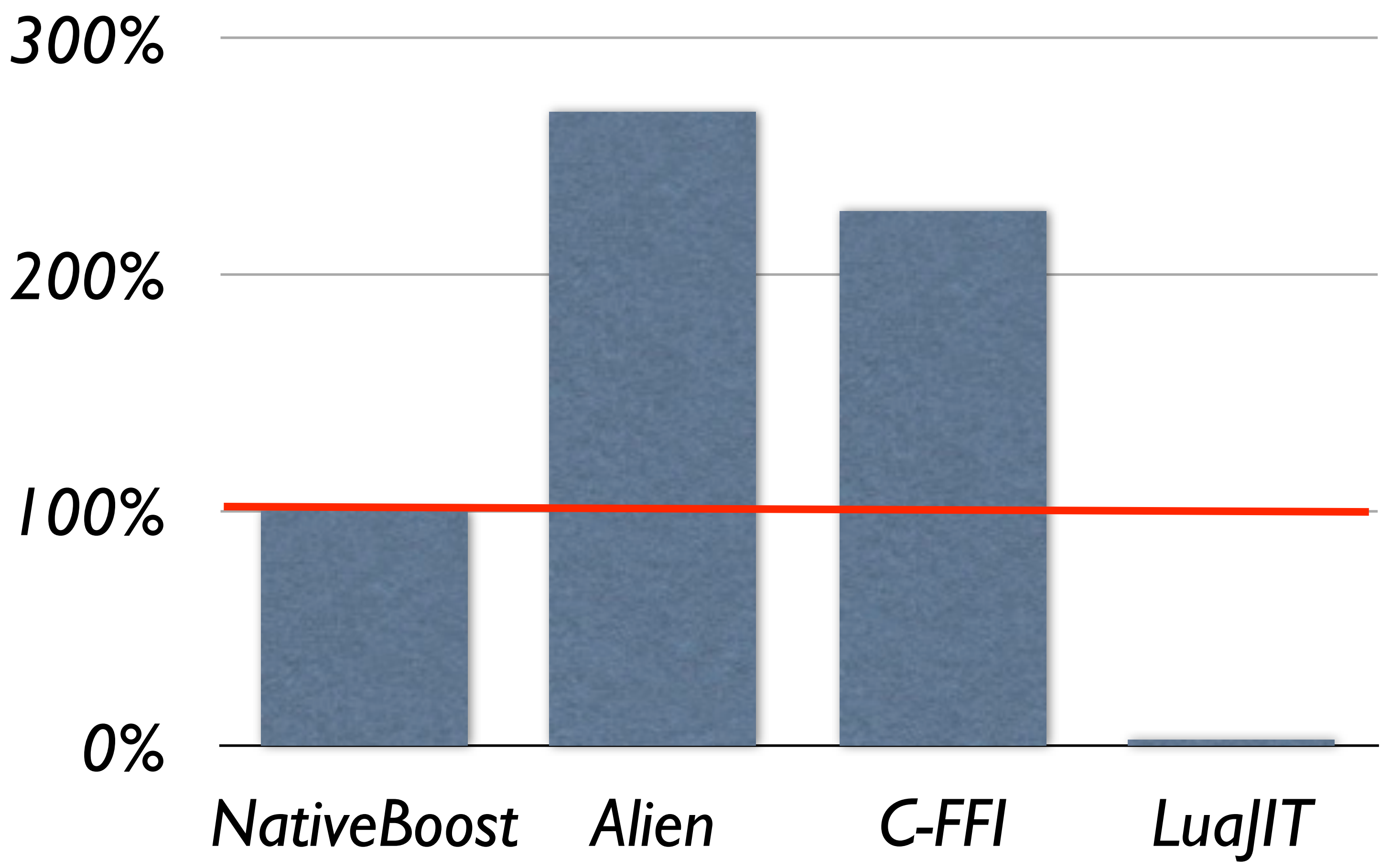
Callout Evaluation

Average time
calling out `uint clock(void)`



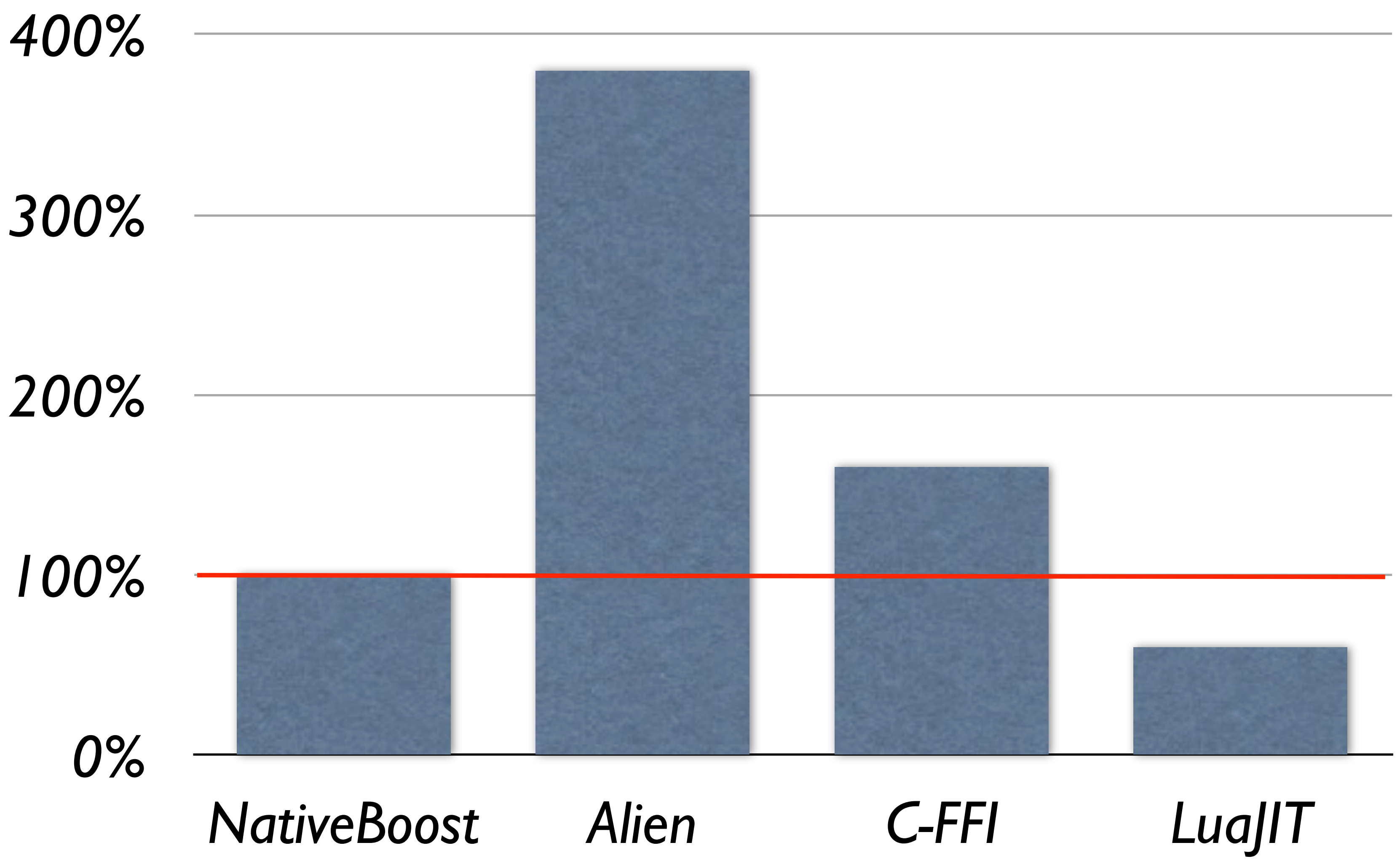
Marshalling int

`int abs(int)`



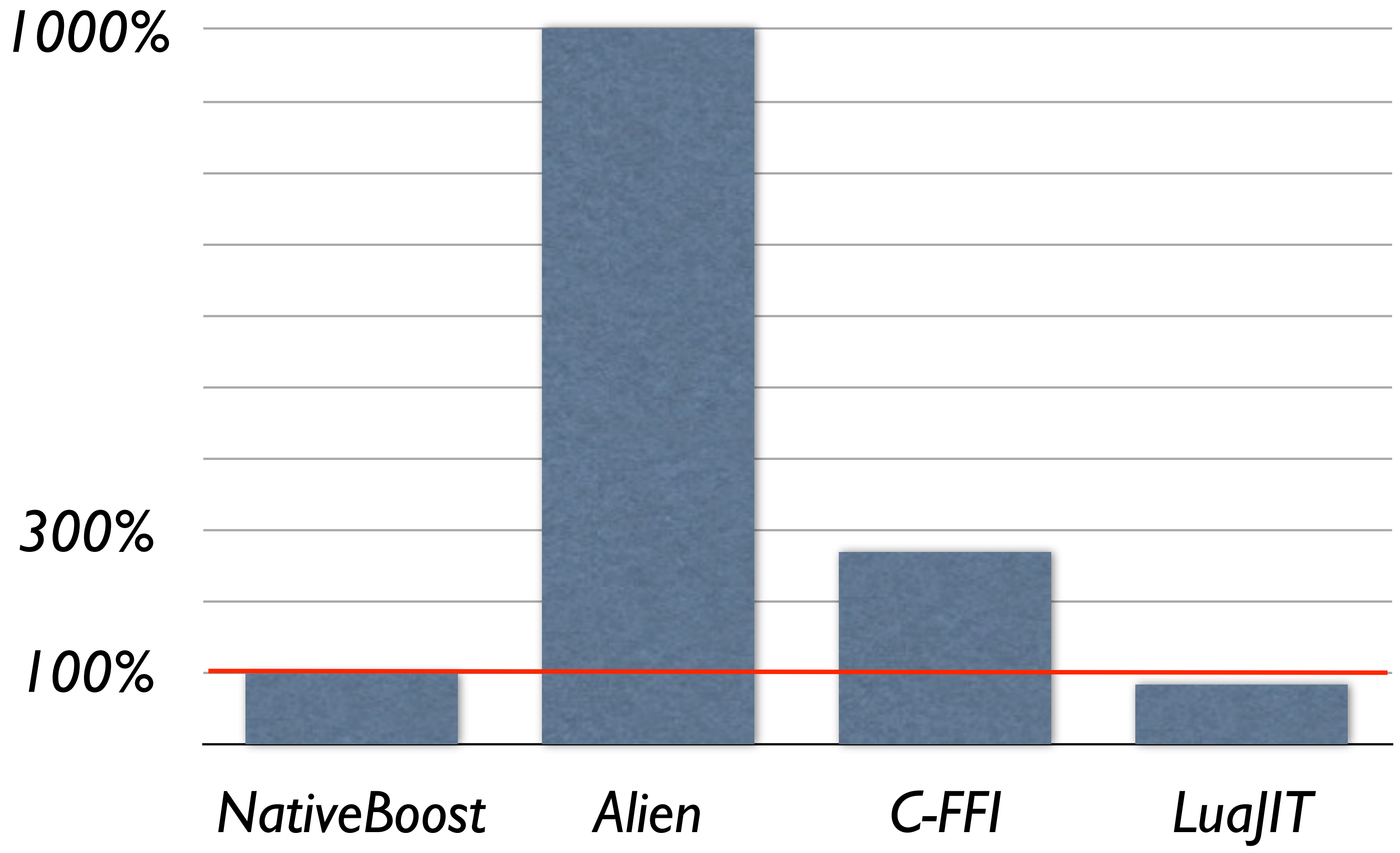
Marshalling char*/String

```
int printf(char*,int,int)
```



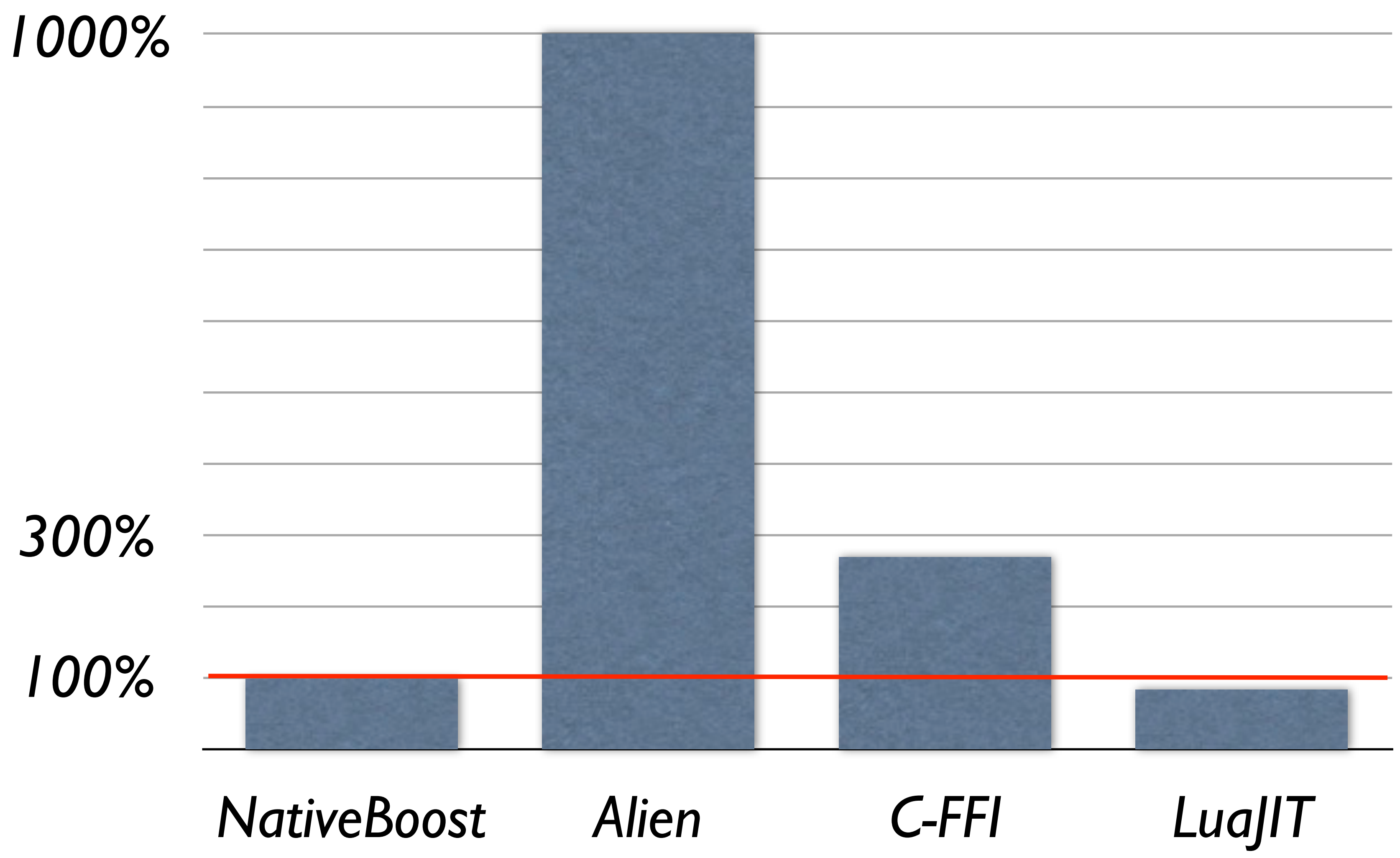
Marshalling char*/String

```
char* getenv( char* )
```



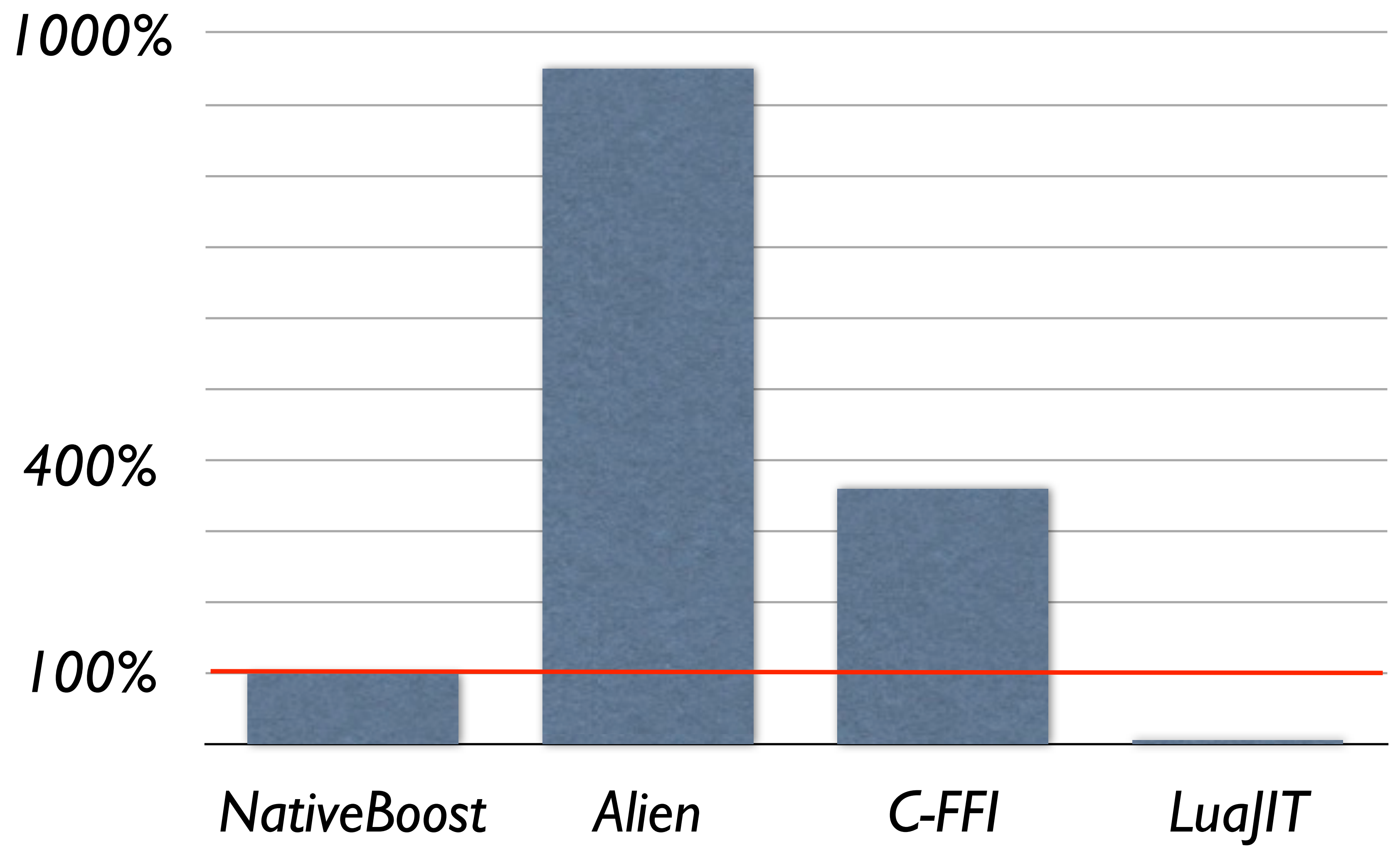
Marshalling char*/String

```
char* getenv( char* )
```



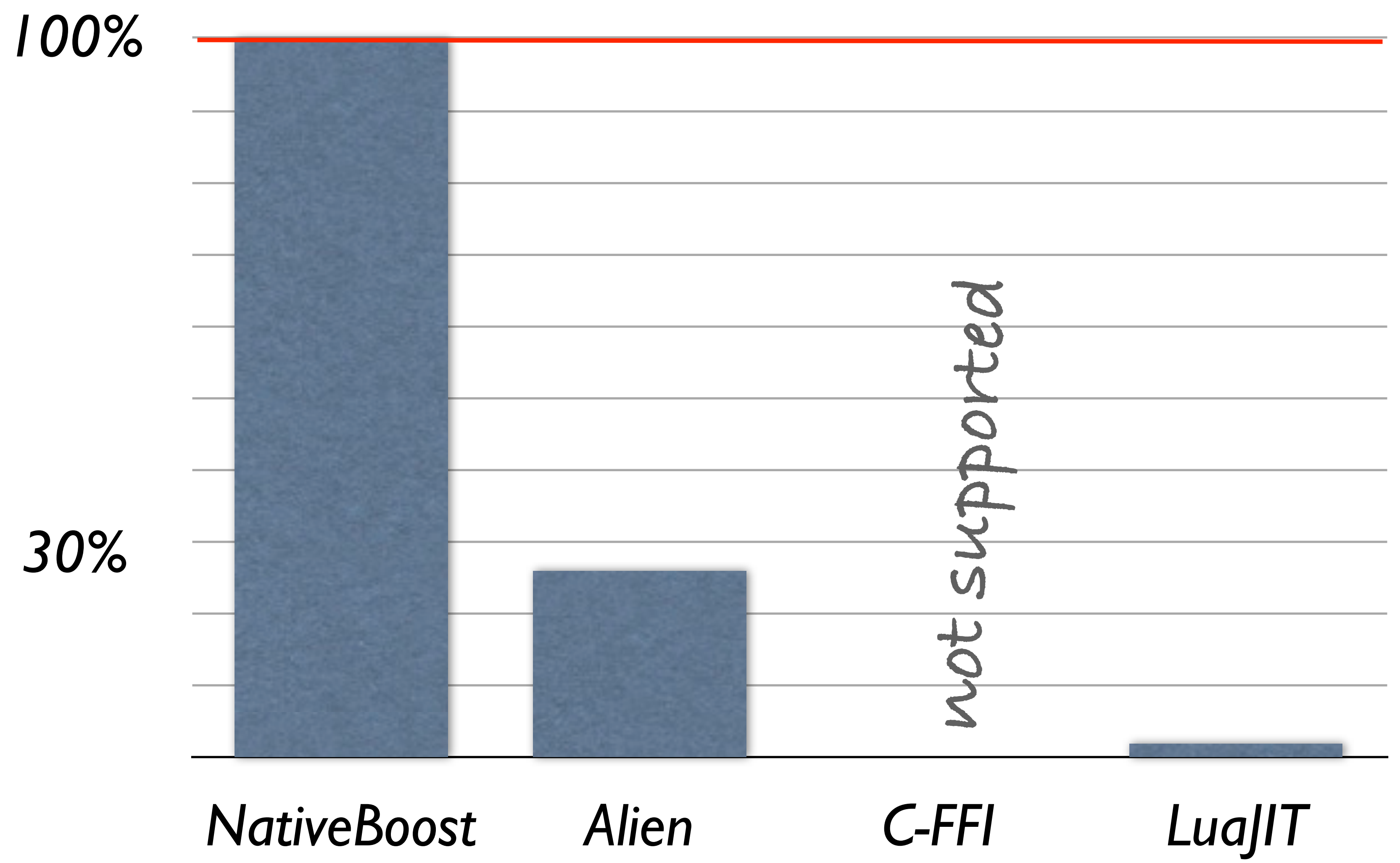
Marshalling structs

```
void
cairo_matrix_multiply (
    cairo_matrix_t *result,
    cairo_matrix_t *a,
    cairo_matrix_t *b)
```



Callbacks Evaluation

```
void qsort (
    void *base,
    size_t nel,
    size_t width,
    int (*compare)(const void*, const void*))
```



Insights into NativeBoost Internals

```
NBExample getenv: 'PATH'
```

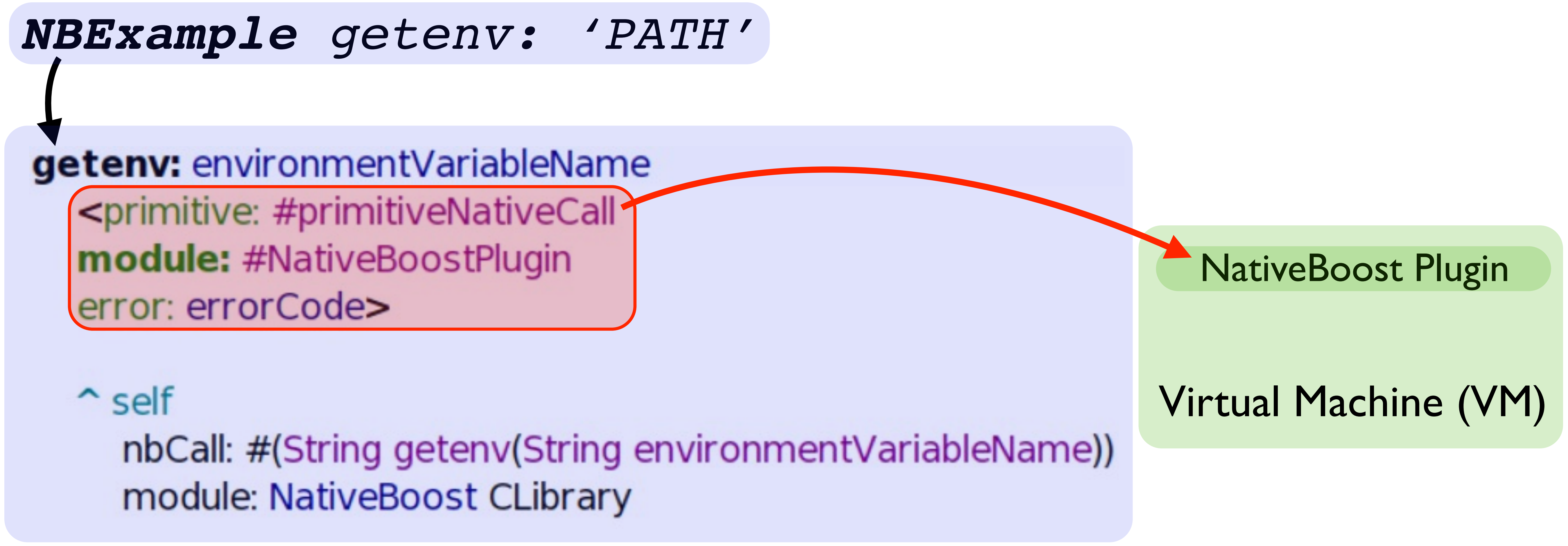
Insights into NativeBoost Internals

```
NBExample getenv: 'PATH'
```

```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

Insights into NativeBoost Internals

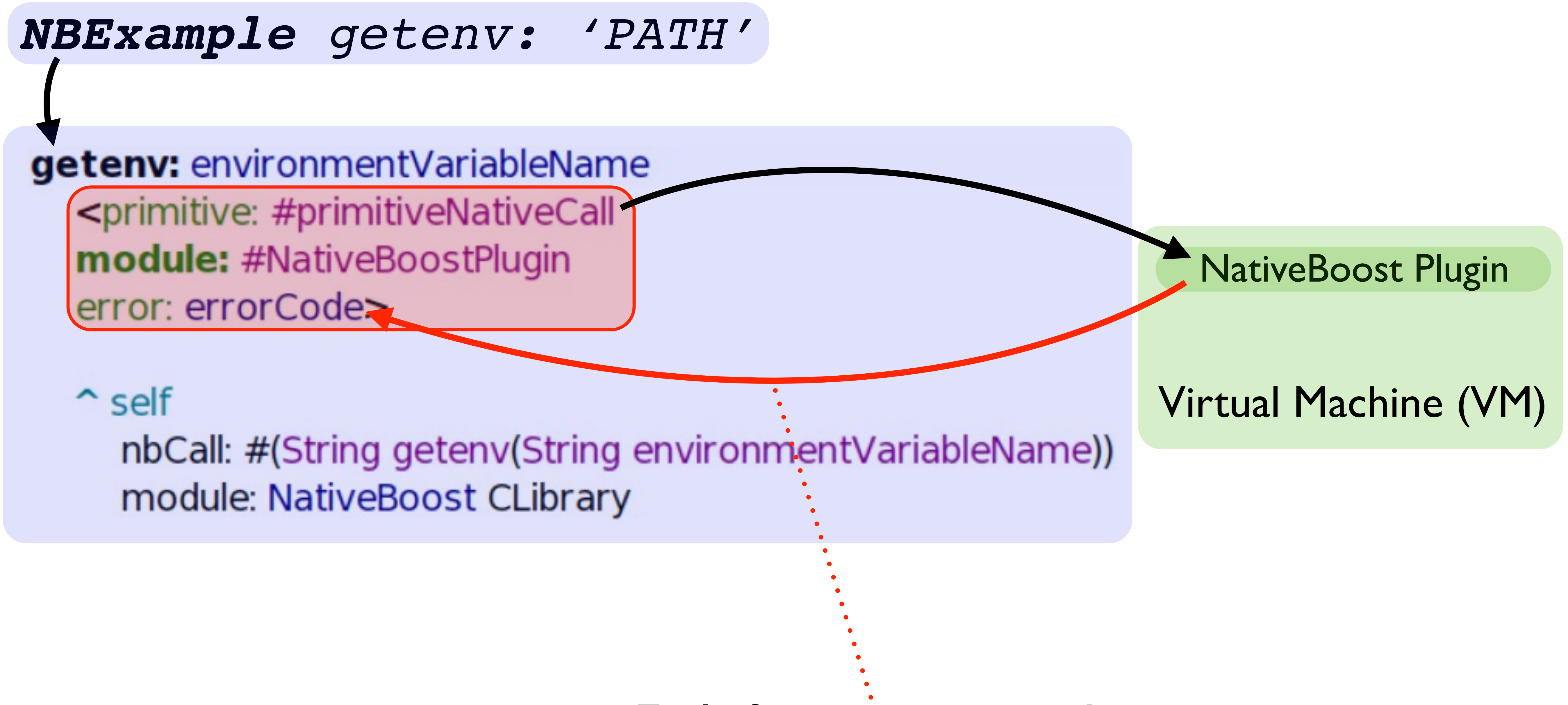


```
NBExample getenv: 'PATH'
```

```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

Insights into NativeBoost Internals



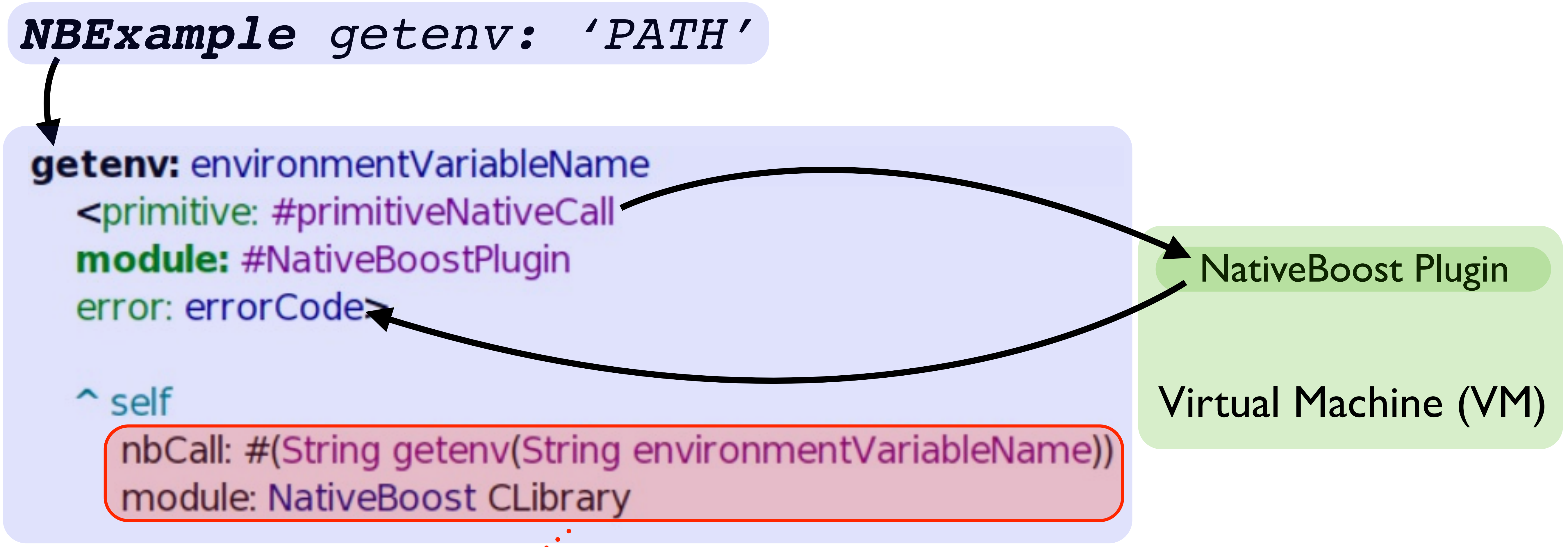
```
NBExample getenv: 'PATH'
```

```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

Fail if no native code associated with #getenv:

Insights into NativeBoost Internals



```
NBExample getenv: 'PATH'
```

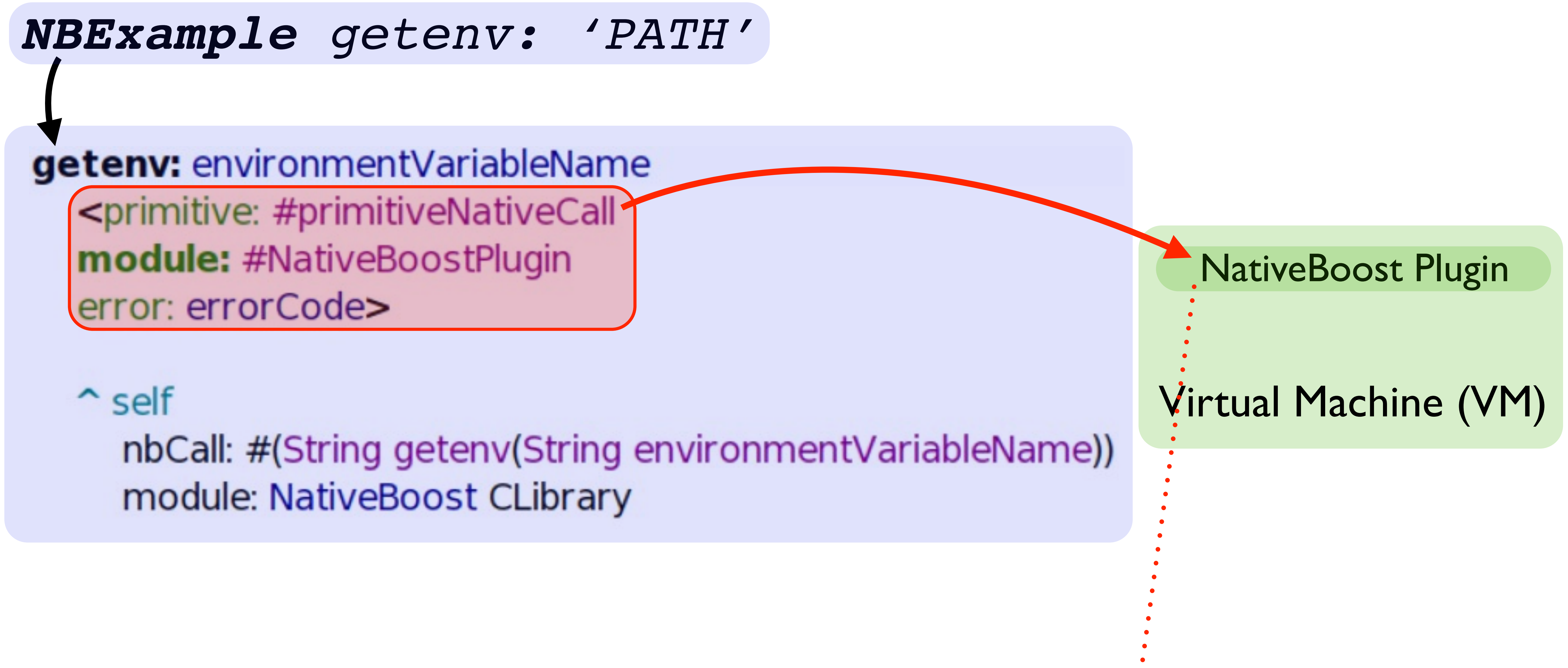
```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
      nbCall: #(String getenv(String environmentVariableName))
      module: NativeBoost CLibrary
```

1. generate native code for marshalling, ...
2. associate it with #getenv:
3. restart the method execution

Insights into NativeBoost Internals

```
NBExample getenv: 'PATH'
```



```
getenv: environmentVariableName
    <primitive: #primitiveNativeCall
    module: #NativeBoostPlugin
    error: errorCode>

    ^ self
        nbCall: #(String getenv(String environmentVariableName))
        module: NativeBoost CLibrary
```

activate the native code associated with #getenv:

Conclusion

- NativeBoost-FFI is:
  - Language-side: extensible, high-level code only
  - Fast compared to other Smalltalk FFI
    - Needs optimizations on Callbacks but that would require strong VM support

Future Work

- Improve NativeBoost Callback performance
    - Reuse Alien's VM Callback support?
- Better integration of NativeBoost with the JIT
    - Do not leave JIT-mode when activating a NB method

Language-side Foreign Function Interfaces with NativeBoost

IWST 2013

Camillo Bruni, Luc Fabresse, Stéphane Ducasse, Igor Stasenko