BlazeHtml

Design of a blazingly fast html combinator library

Hello!

My name is jasper

Studying BSc CS @ UGent

I like to make things

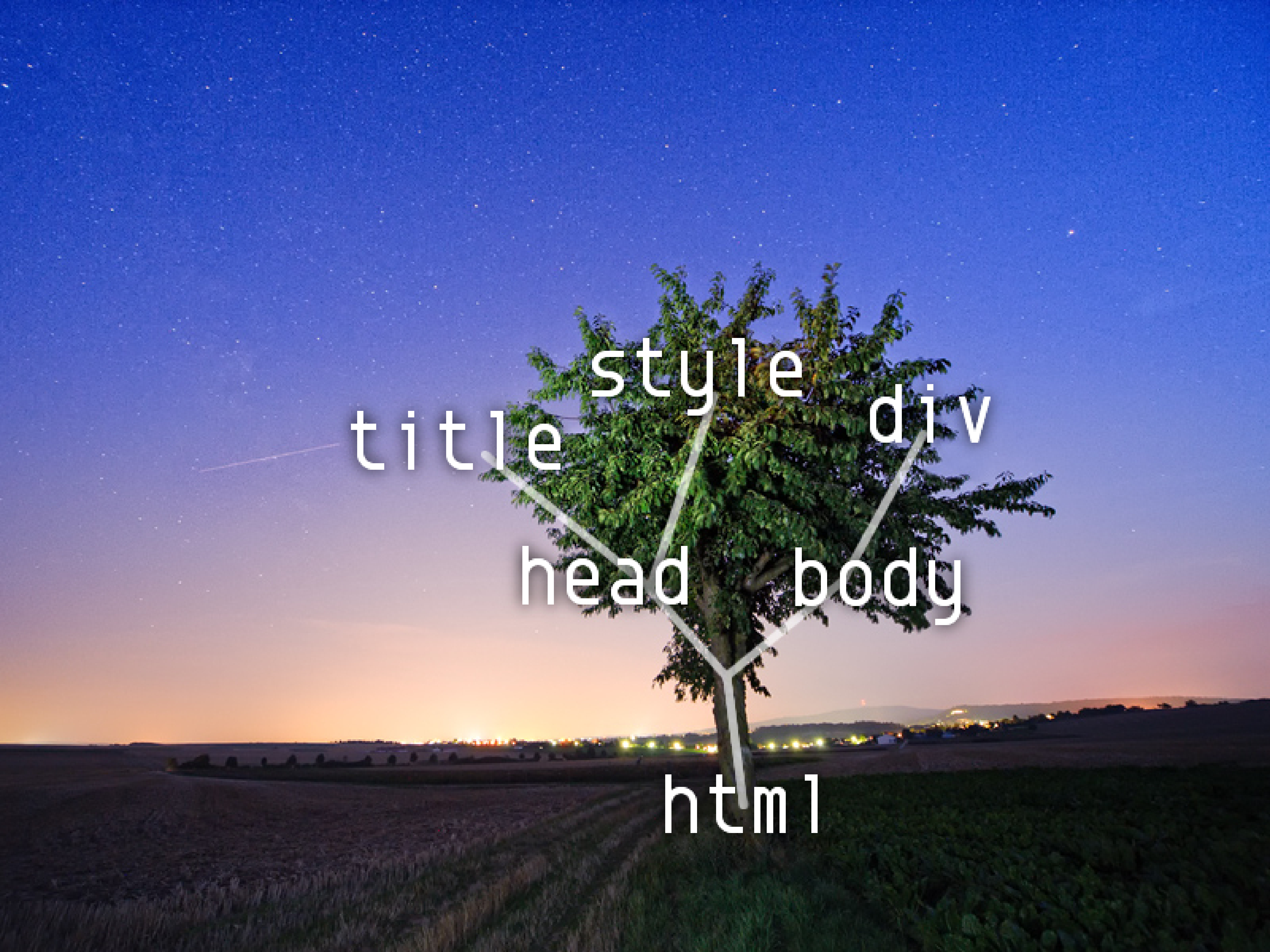
@jaspervdj

jaspervdj.be

Introduction

A web app usually has 3 important layers:

- web application server
- data storage layer
- html generation layer


style
title
div
head
body
html

Trees in Haskell

```
> data Tree
>     = Node Tree Tree
>     | Empty
```

Html is a tree

This makes writing an Html generation library trivial in Haskell.

Le't write a library!

```
> type Attribute =
>       (String, String)
> type Tag = String
```

Le't write a library!

```
> data Html
>   = Node Tag [Attribute]
>             Html
>   | Leaf Tag [Attribute]
>   | Concat [Html]
>   | Text String
>   deriving (Show)
```

Lazy cat
Fixed your library

Hackage quick look

- html
- xhtml
- xhtml-combinators
- moe
- xhtml1

We bravely started on ZuriHac

we need to be

FASTER

s/String/Data.Text/g

- A first good step

s/String/Data.Text/g

- A first good step

```
text1 `append` text2
```

s/String/Data.Text/g

- A first good step

```
text1 `append` text2
```

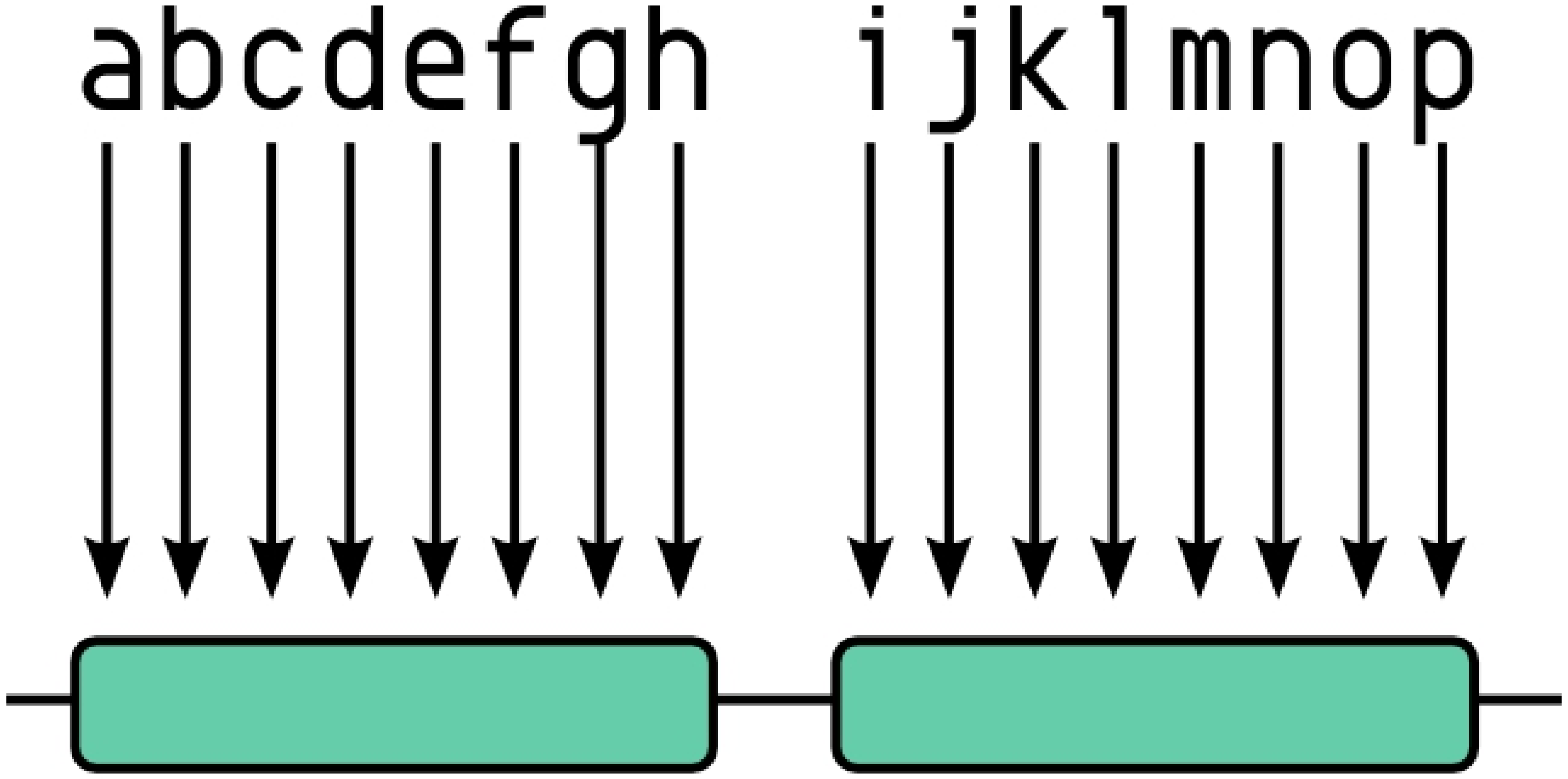
BAD BAD BAD

No inspiration?

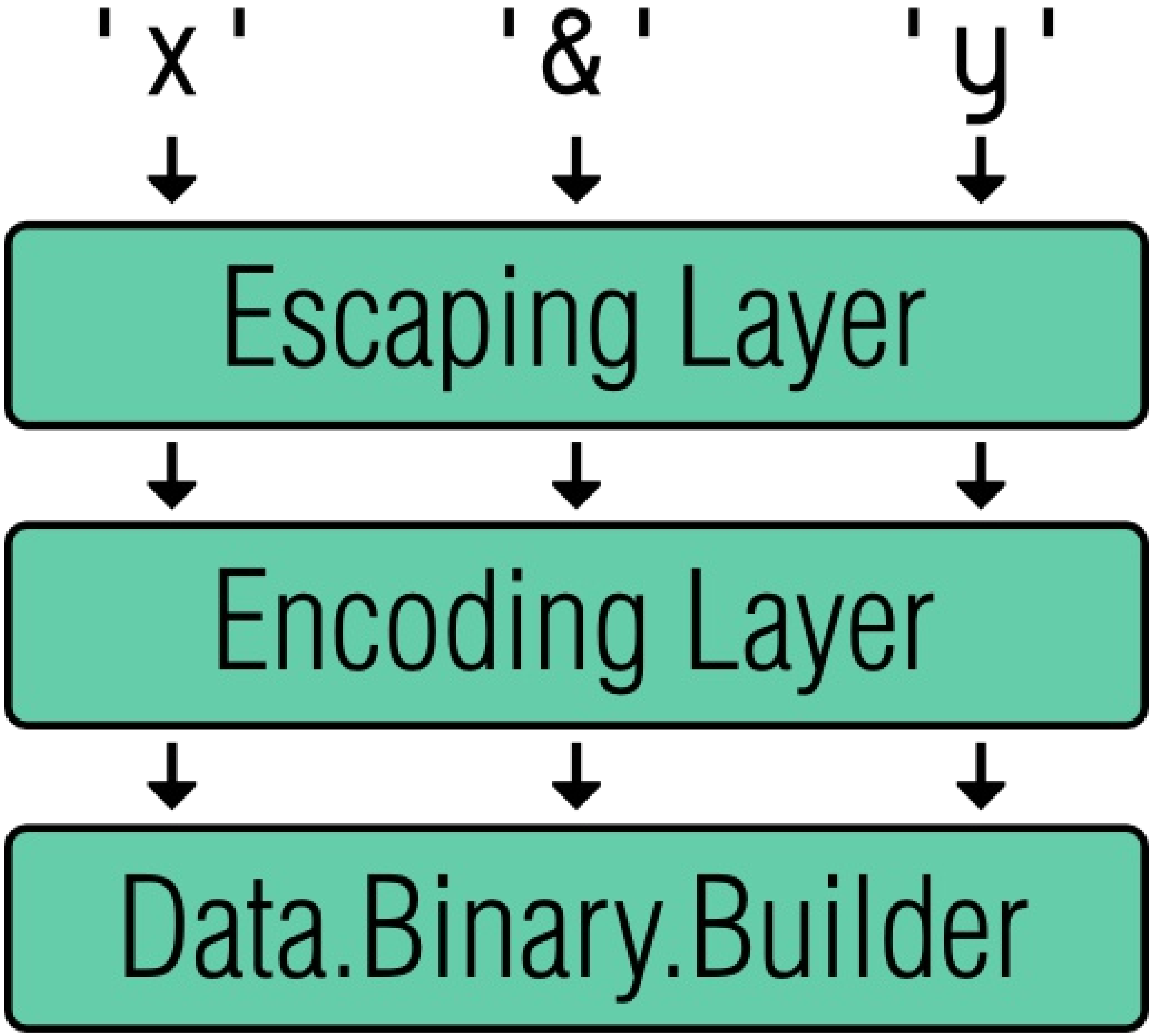
Just shamelessly steal ideas from other Haskell projects.

Ninja time! Let's steal Data.Binary.Builder

Builder Monoid



3-layer approach



Too much overhead makes performance kitteh sad

Time to fork Data.Binary.Builder

Builder Fork

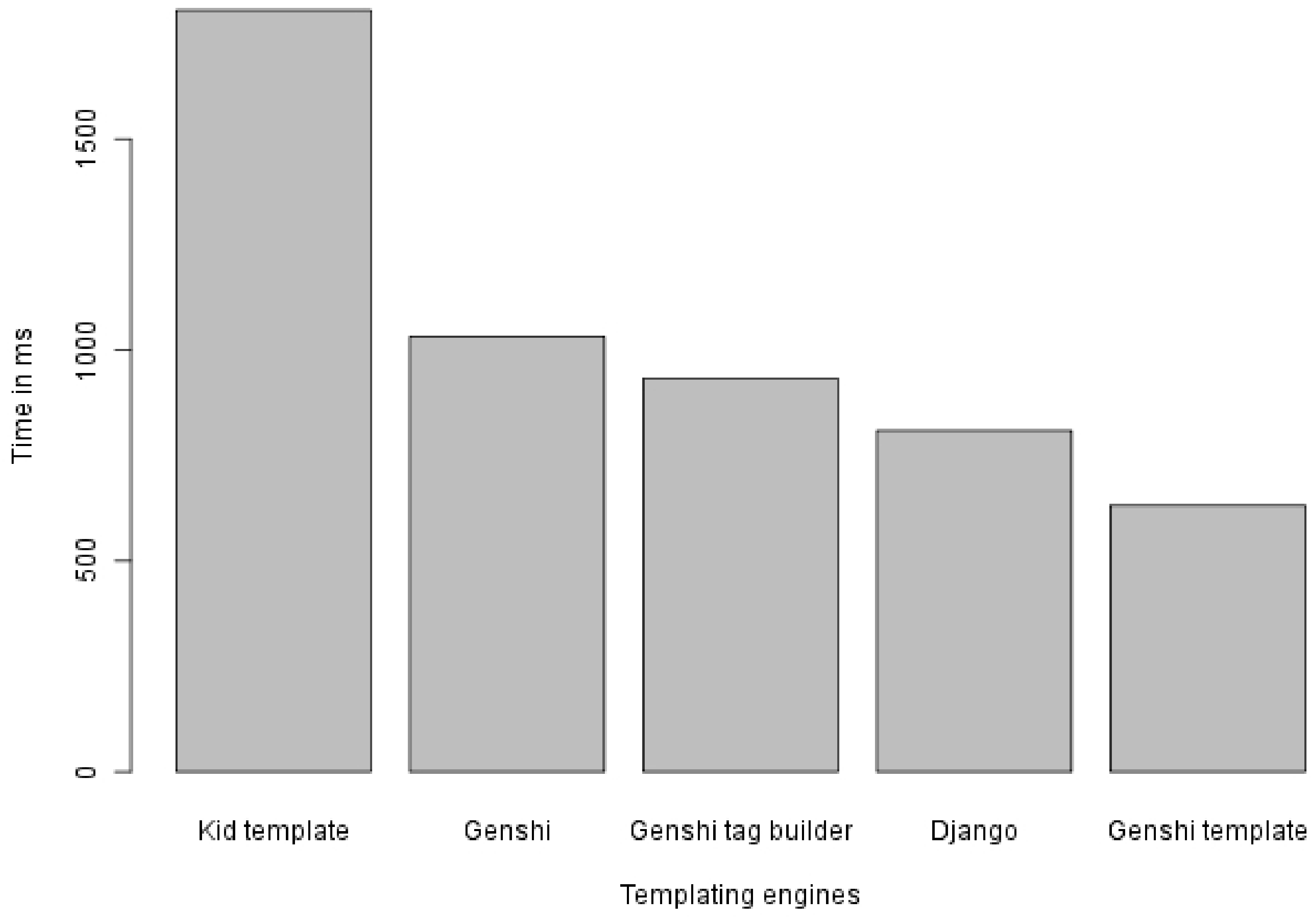
```
fromText :: Text -> Builder
fromEscapedText :: Text
                -> Builder
fromShow :: Show a
         => a -> Builder


fromRawAscii7Char :: Char
                  -> Builder
```

Builder Fork

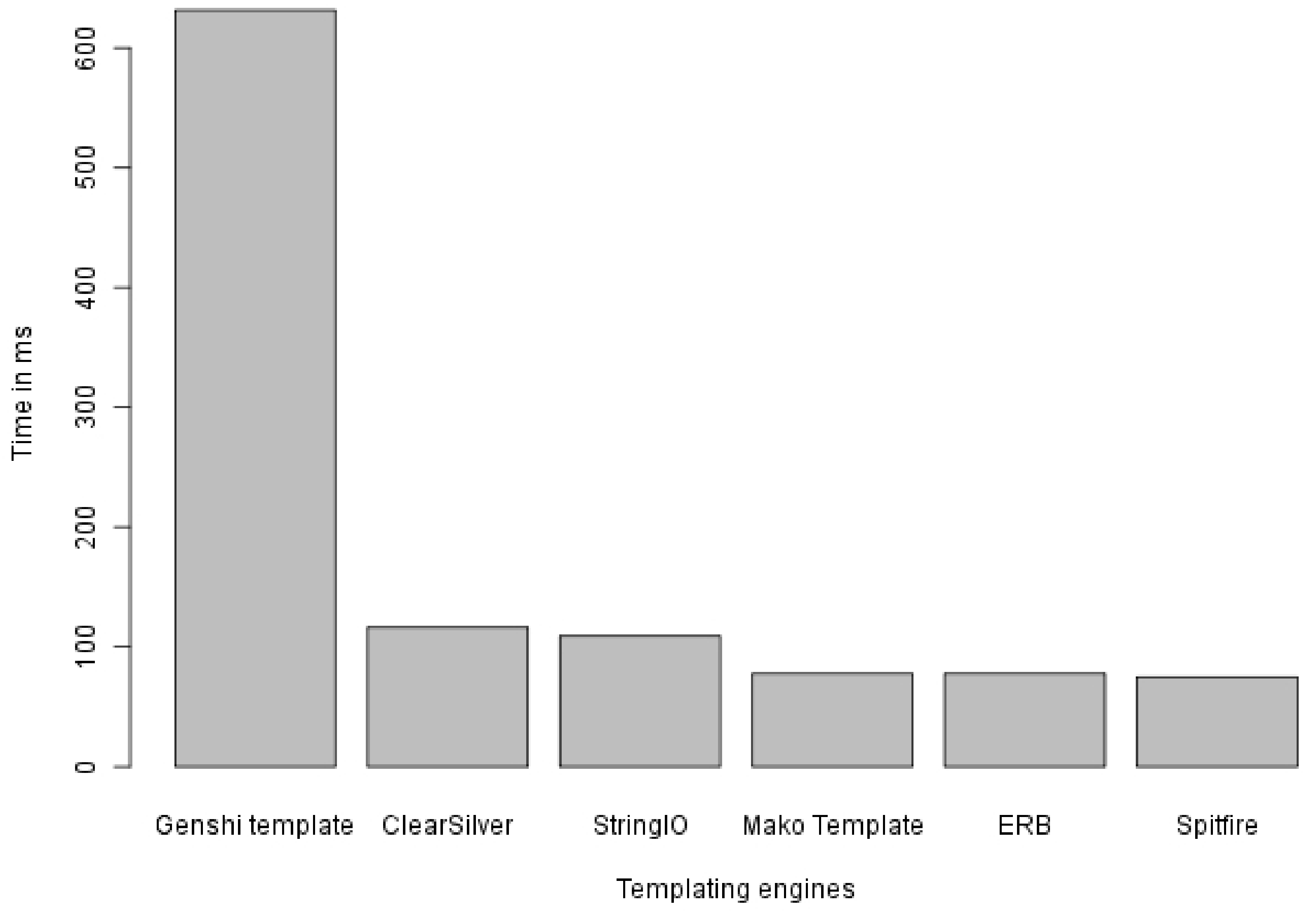
Additional function:

```
> fromUnsafeWrite
>     :: Int
>     -> (Ptr Word8
>           -> IO ())
>     -> Builder
```

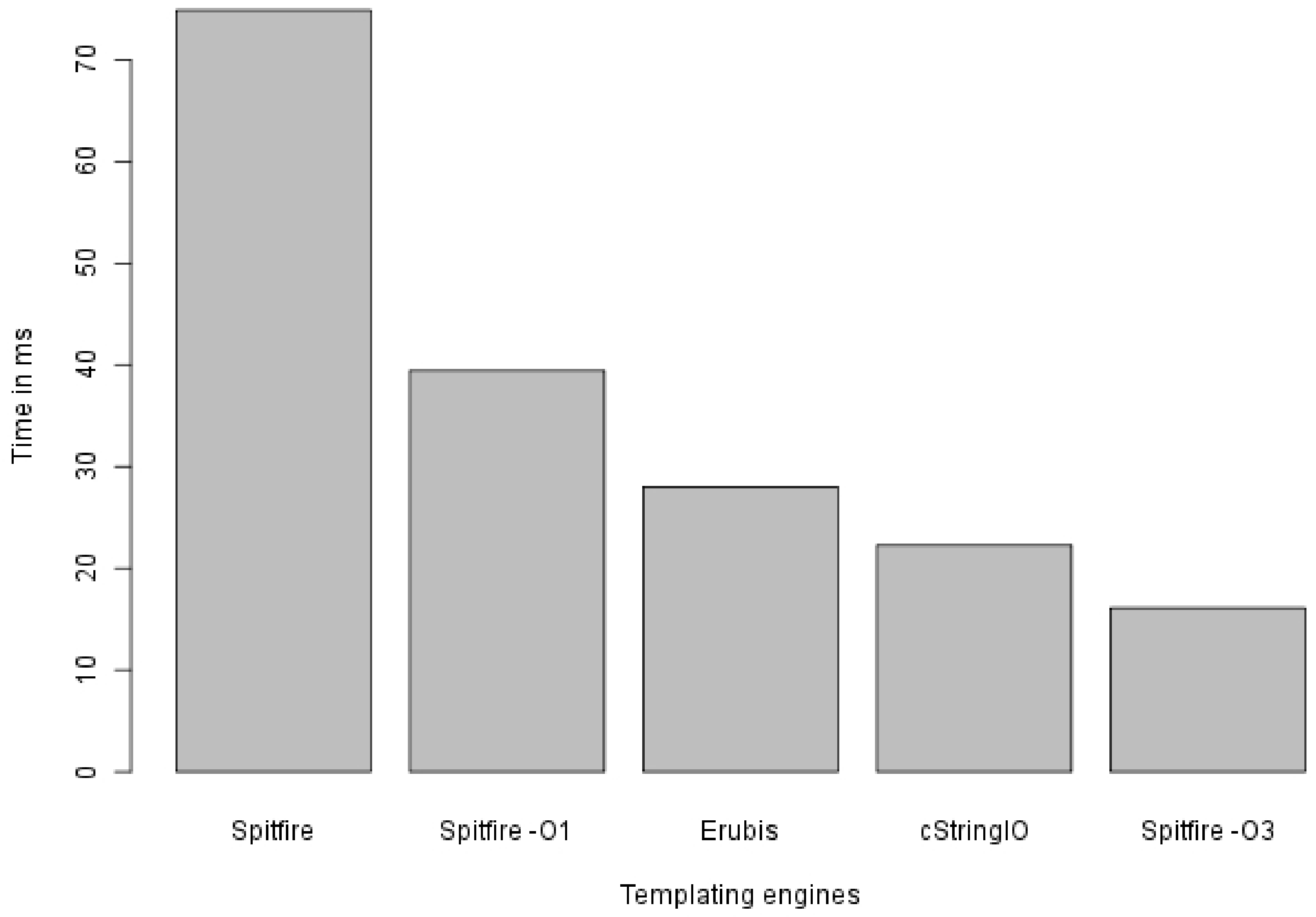
Benchmarks


Time in ms
0
500
1000
1500
Kid template
Genshi
Genshi tag builder
Django
Genshi template
Templating engines

Benchmarks

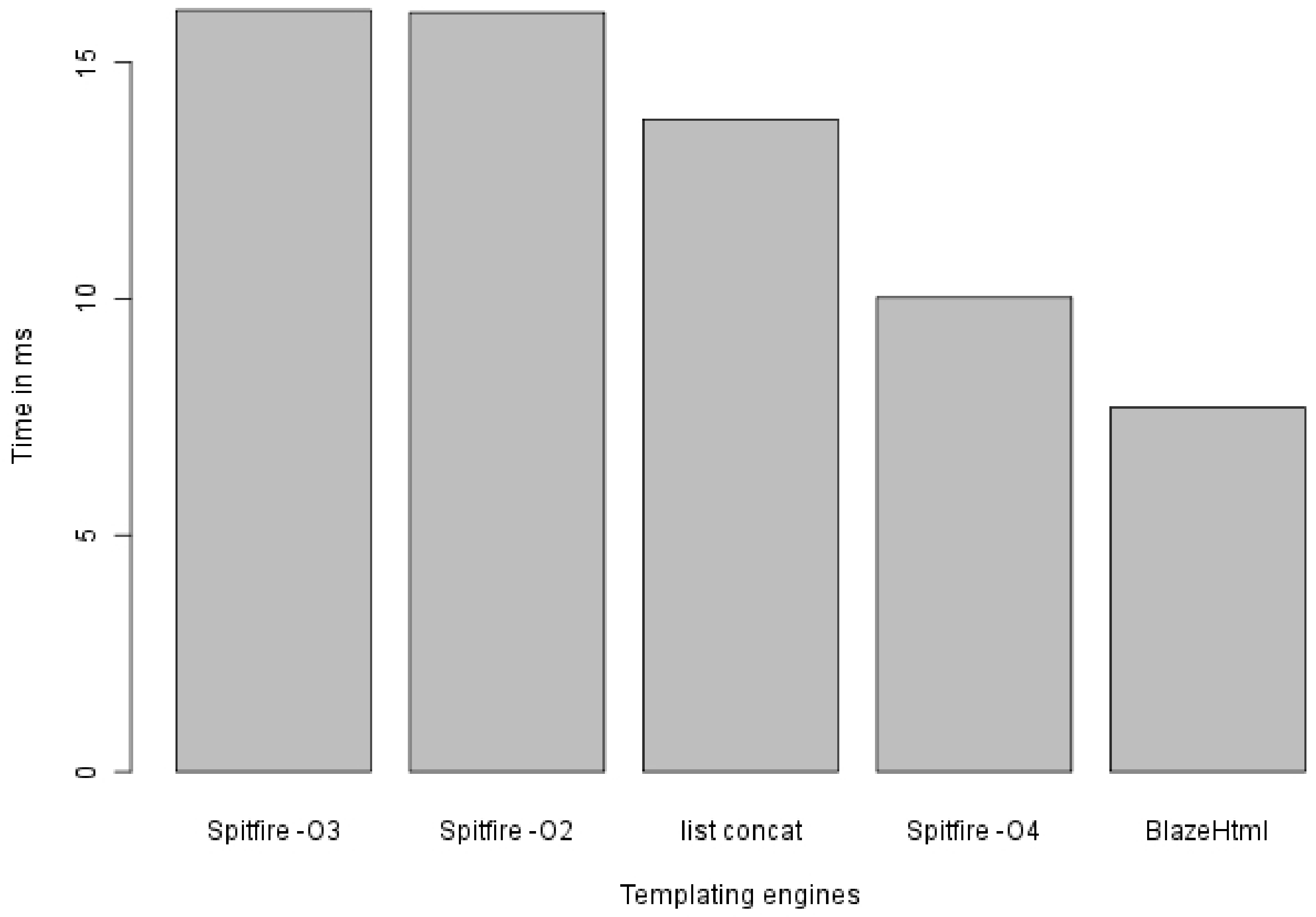

Time in ms
0
100
200
300
400
500
600
Genshi template
ClearSilver
StringIO
Mako Template
ERB
Spitfire
Templating engines

Benchmarks



Time in ms
0
10
20
30
40
50
60
70
Spitfire
Spitfire -O1
Erubis
cStringIO
Spitfire -O3
Templating engines

Benchmarks



Time in ms
0
5
10
15
Spitfire -O3
Spitfire -O2
list concat
Spitfire -O4
BlazeHtml
Templating engines

The future

Lots still to do until we have a stable, fast, awesome version.

For the curious:
`github.com/jaspervdj/BlazeHtml`

Questions?