Introducing brigand

EDOUARD ALLIGAND – QUASARDB

JOEL FALCOU - NUMSCALE

What is brigand?

A **brigand** is a person who usually lives in a gang and lives by pillage and robbery.

A lightweight C++ 11 meta-programming library

Inspiration

Boost.MPL by *Aleksey Gurtovoy and David Abrahams*

Simple C++ 11 Metaprogramming Article by *Peter Dimov*

Tiny Metaprogramming Library by *Eric Niebler*

Goals

Instant-compile time

Sustainable code base

Wide support

Use cases

Compile-time checks

Constant computations

Protocol generator

Memory data layout (NT²)

And many more!

Features

Algorithms	Functions	Sequences
• Count	• Apply	• List
• Find	• Arithmetic	• Integral_list
• Fold	• Comparisons	• Map
• For_each	• Logical	• Set
• Remove	• Repeat	• Pair
• Reverse		• Range
• Select		
• Transform		
• Wrap		

99% of Boost.MPL is there!
Works on Visual Studio 2013 (Update 4)!
Macro-free!

Type manipulation

```
// this is my list, there are many like it but this one is mine
using my_list = brigand::list<char, int, bool, void>; // yes void is supported
using my_ptr_list = brigand::transform<my_list, std::add_pointer<brigand::_1>>;
using my_ptr_tuple = brigand::as_tuple<my_ptr_list>;


// everything is a type list
using my_list = std::variant<char, int, bool>;
using my_ptr_list = brigand::transform<my_list, std::add_pointer<brigand::_1>>;
using my_ptr_tuple = brigand::as_tuple<my_ptr_list>;
```

Range based interface

```
// find returns a range, not an iterator
using my_list = brigand::list<char, float, void *, int, bool>;


using found = brigand::find<my_list, std::is_same<int, brigand::_1>>;
using found_alt = brigand::find_element<my_list, int>;


// the result is brigand::list<int, bool>, you can search again
using found2 = brigand::find<my_list, std::is_same<int, brigand::_1>>;
```

Go for it!

[HTTPS://GITHUB.COM/EDOUARDA/BRIGAND](https://github.com/edouarda/brigand)

BOOST SOFTWARE LICENSED