C++ Club UK

Gleb Dolgich

2019-02-14

Avoid unsigned

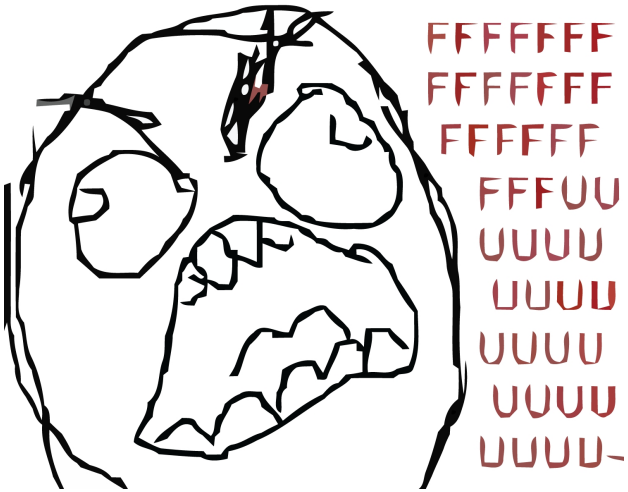
- P0330R4 Literal Suffixes for ptrdiff_t and size_t
- P1227R1 Signed ssize() functions, unsigned size() functions
- P1428R0 Subscripts and sizes should be signed

```
1 template <typename C>
2 constexpr int ssize(const C& c)
3 {
4     const size_t size = c.size();
5     assert(size <= static_cast<size_t>(std::numeric_limits<int>::max()));
6     return static_cast<int>(size);
7 }
```

Cpp.chat with Nicolai Josuttis

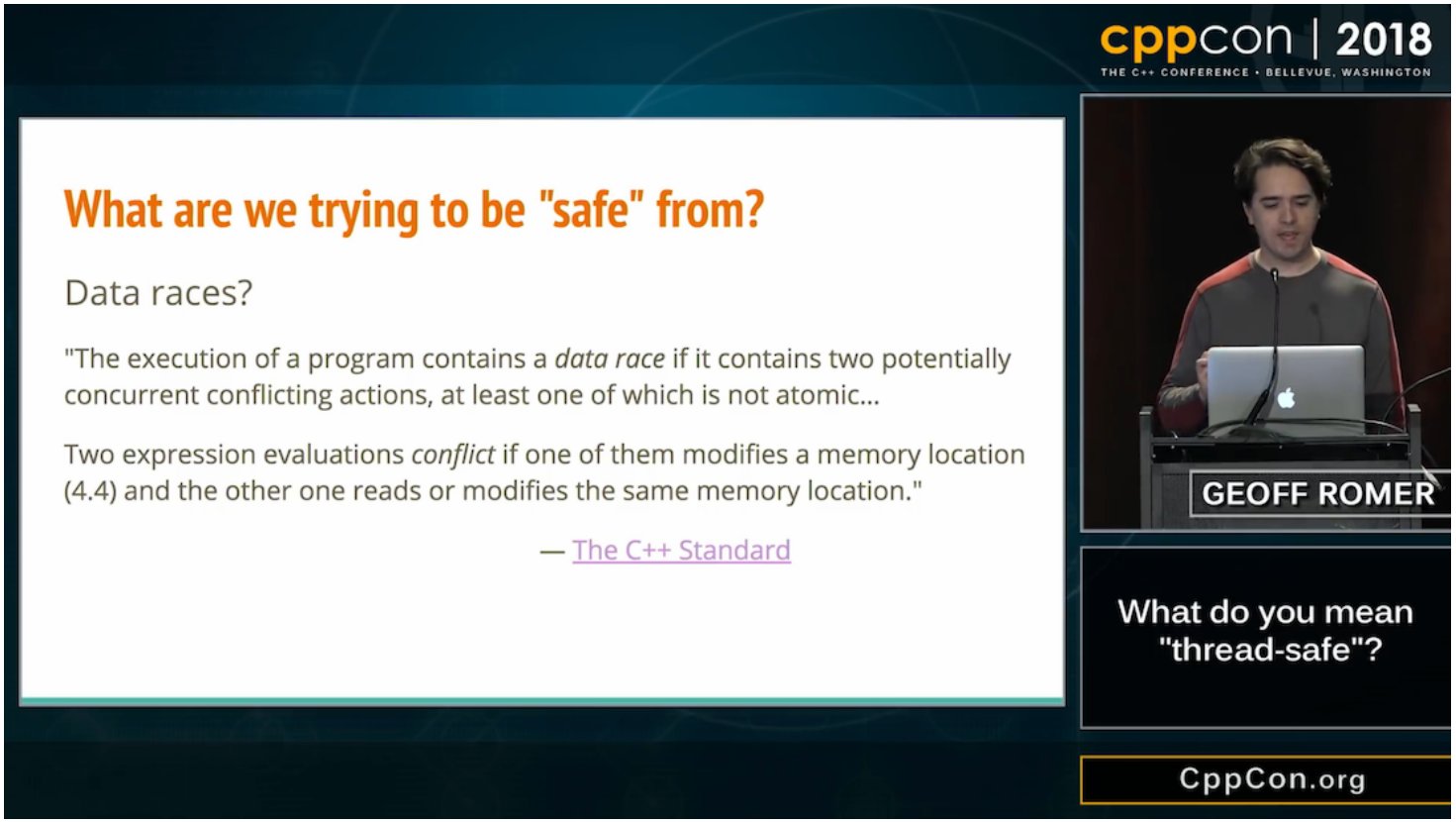
https://youtu.be/_CaP_xwfAFU

- Initialisation is broken!
- Initialiser lists are broken!
- Auto initialisation is broken!



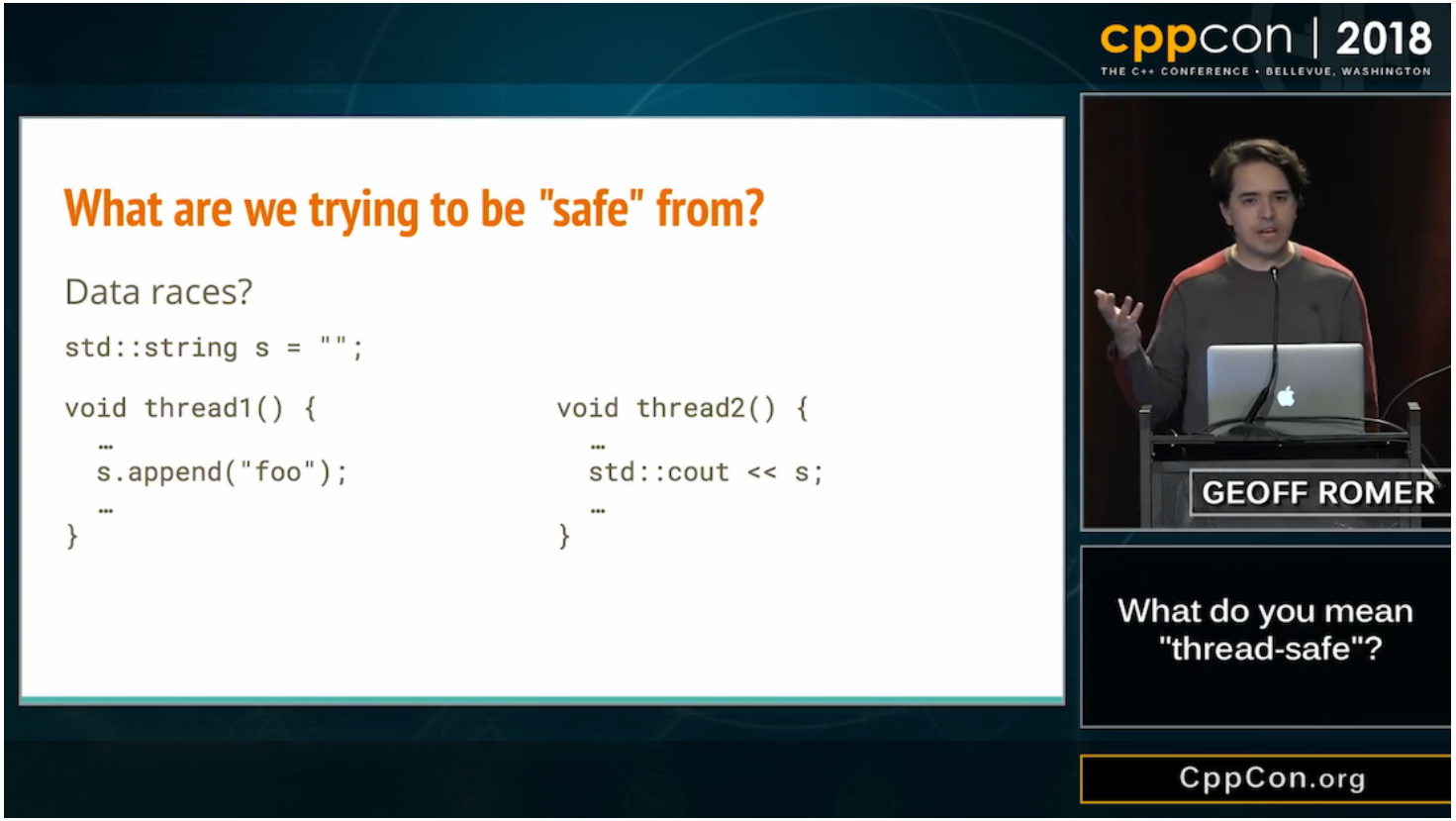
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



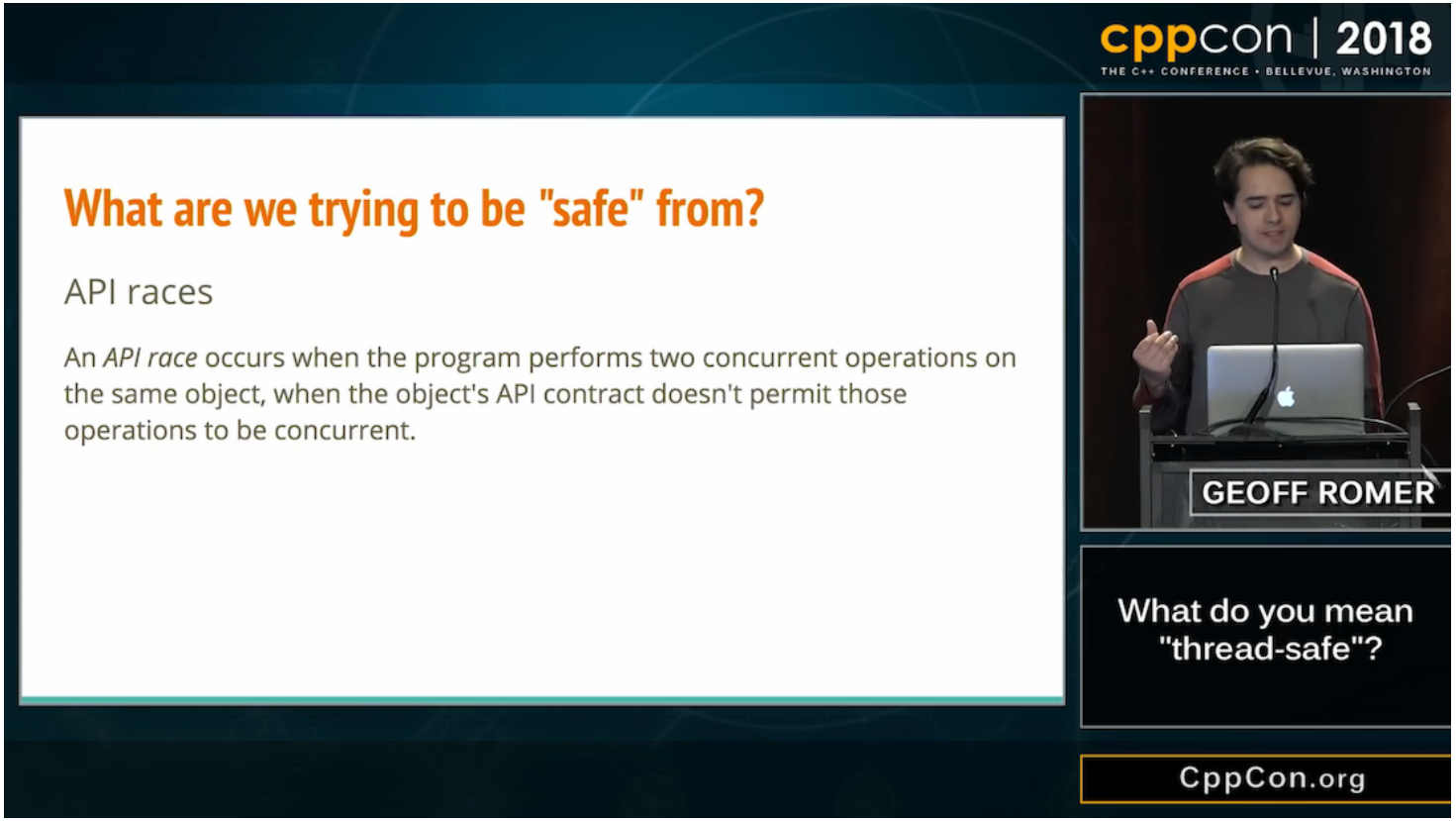
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



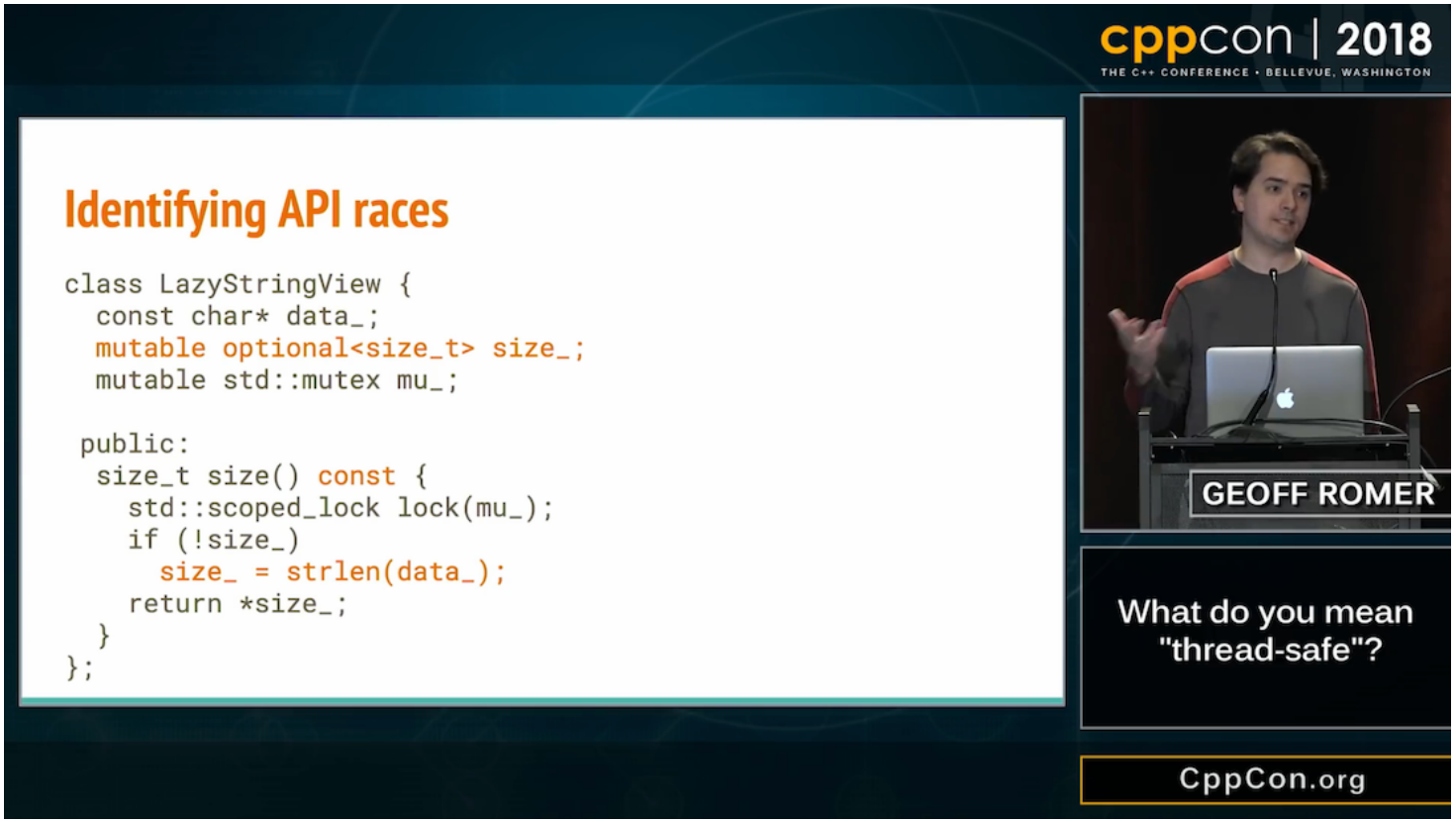
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



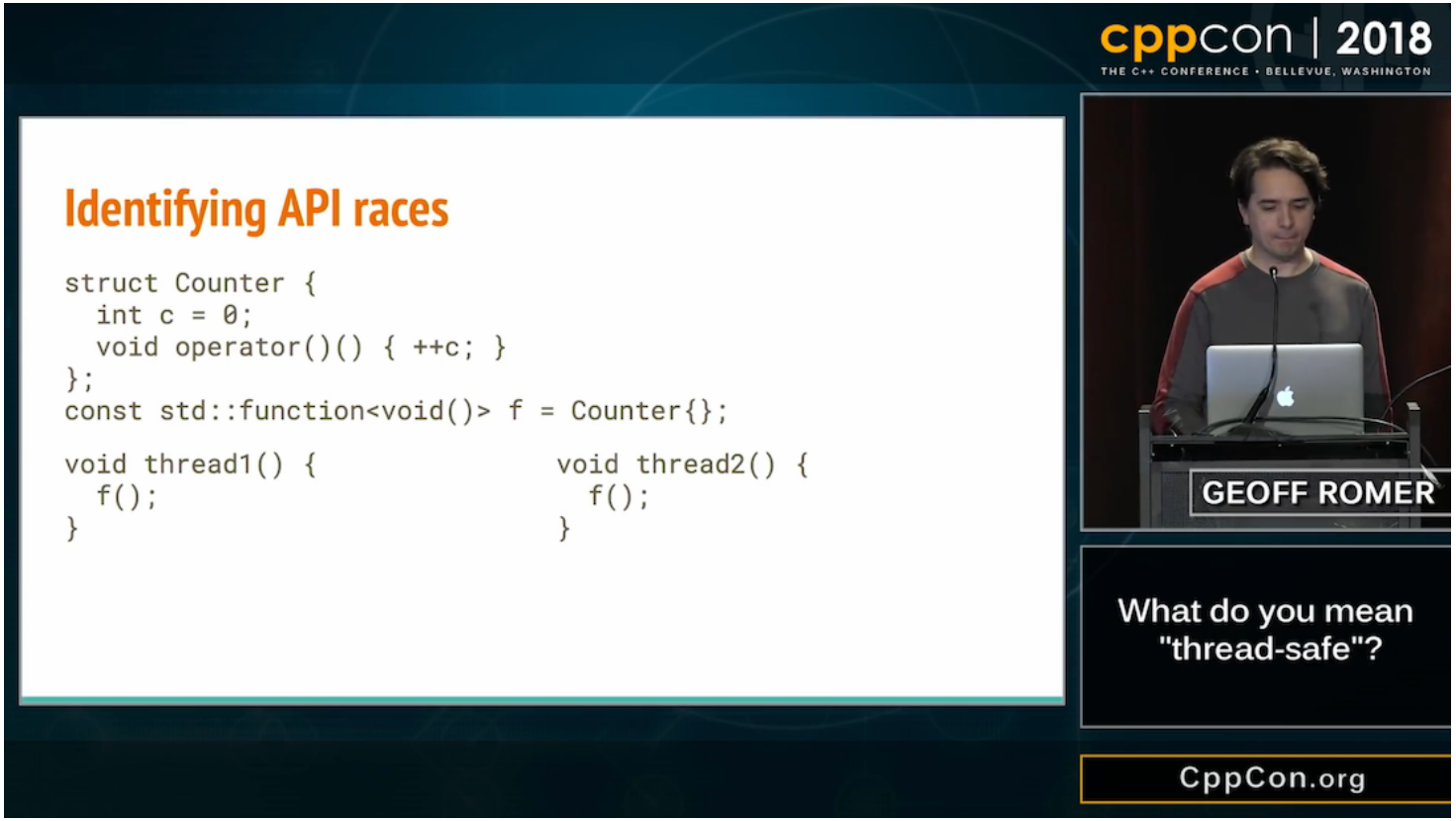
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



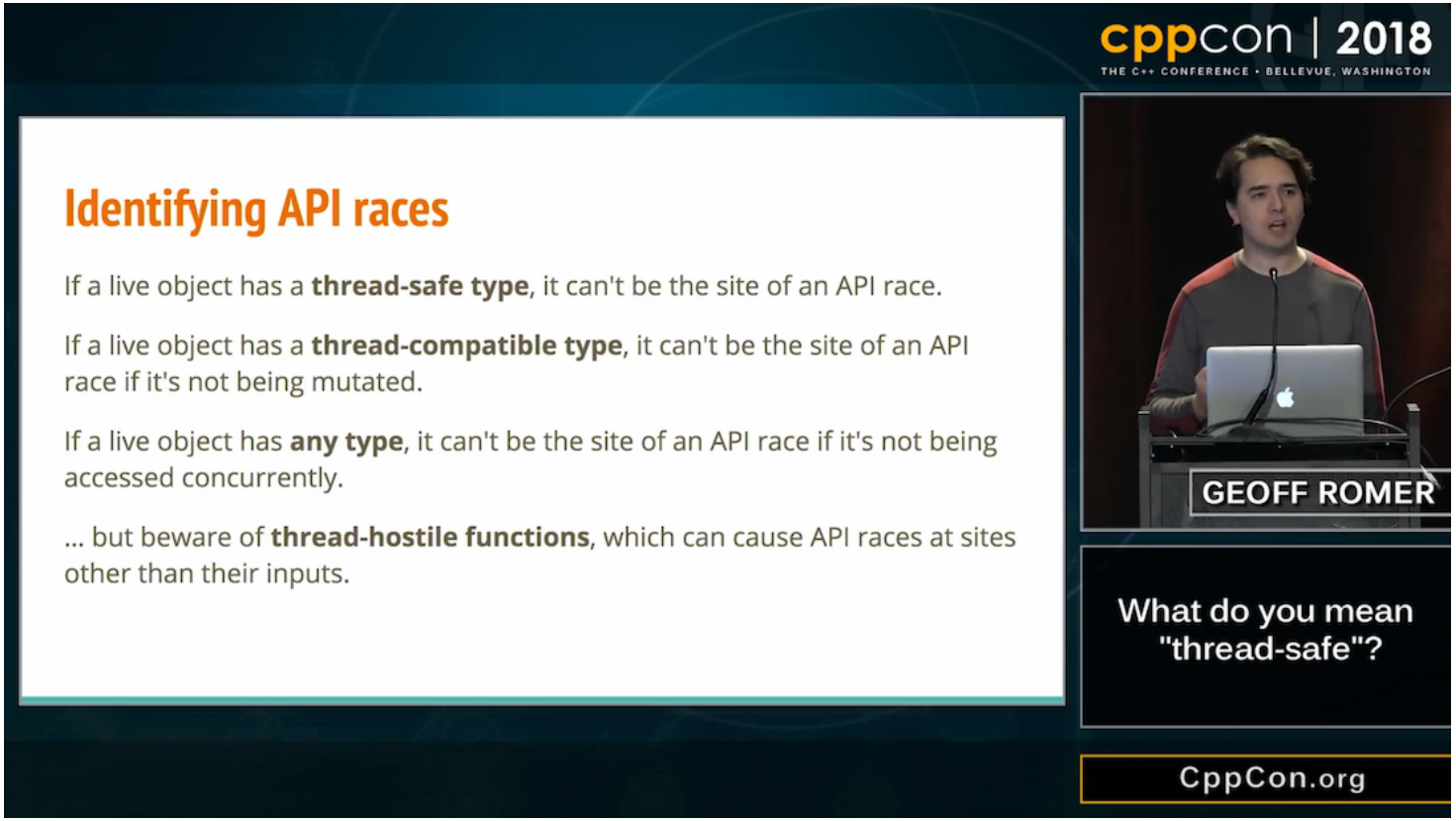
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



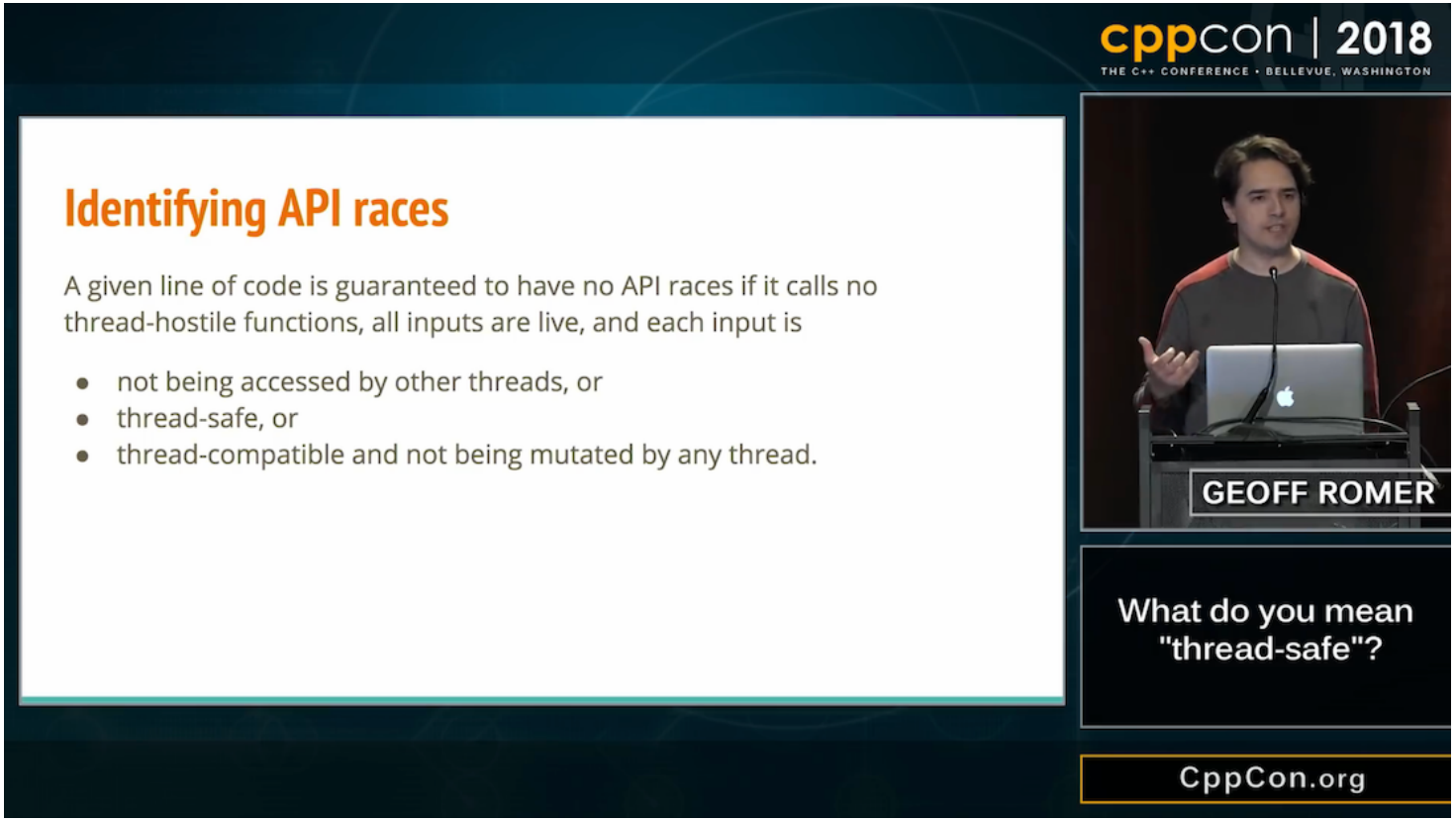
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



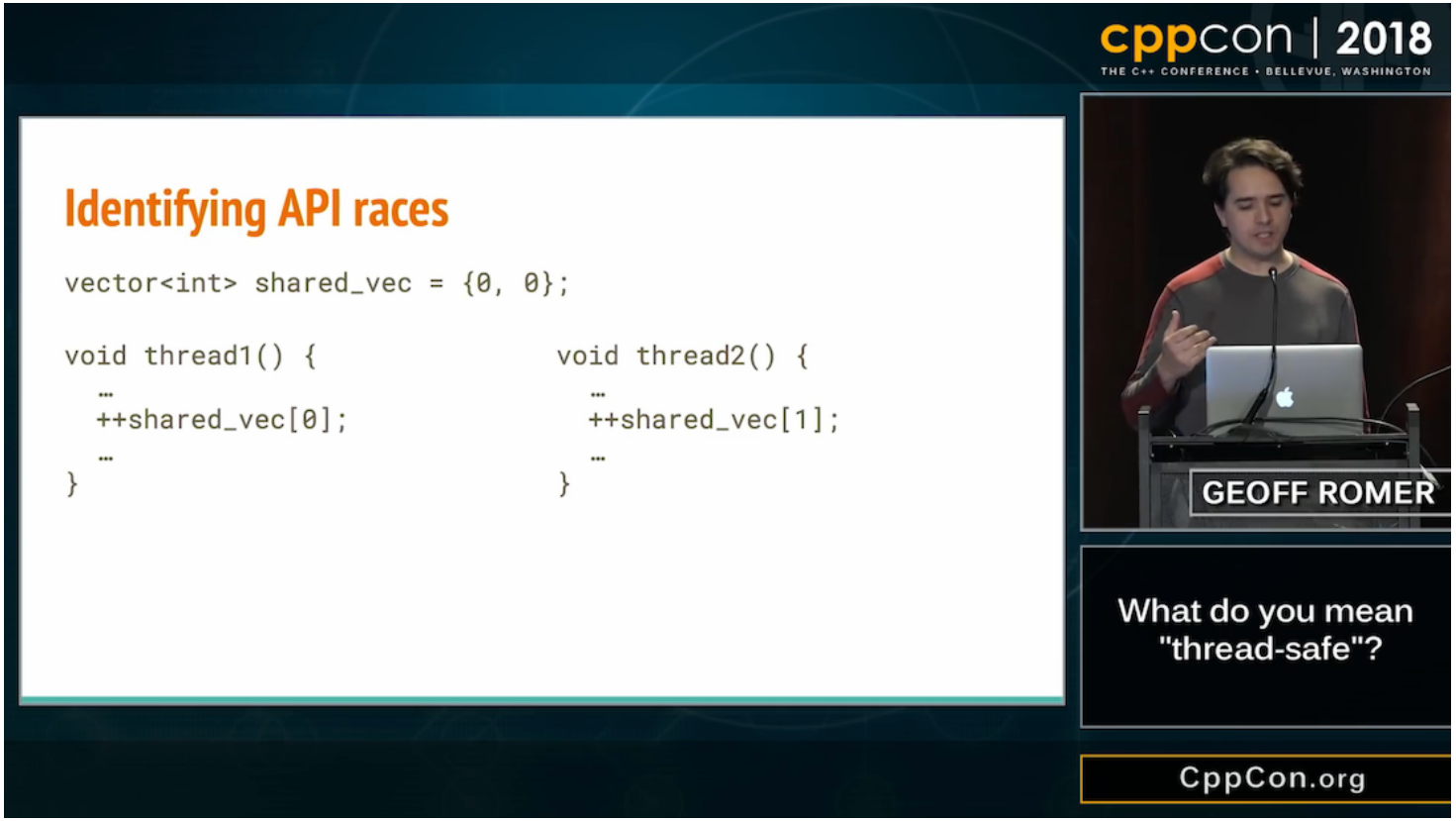
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



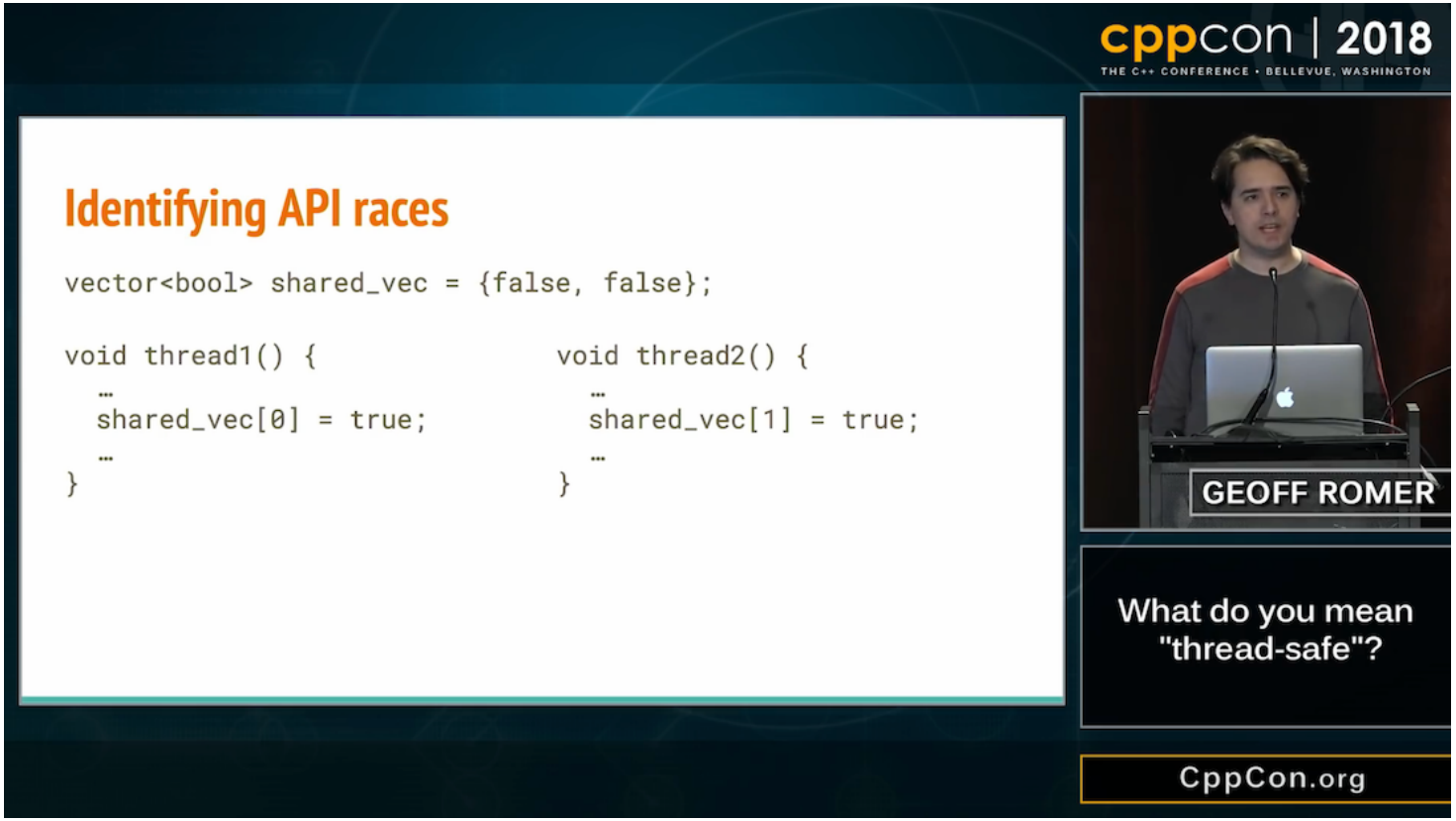
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



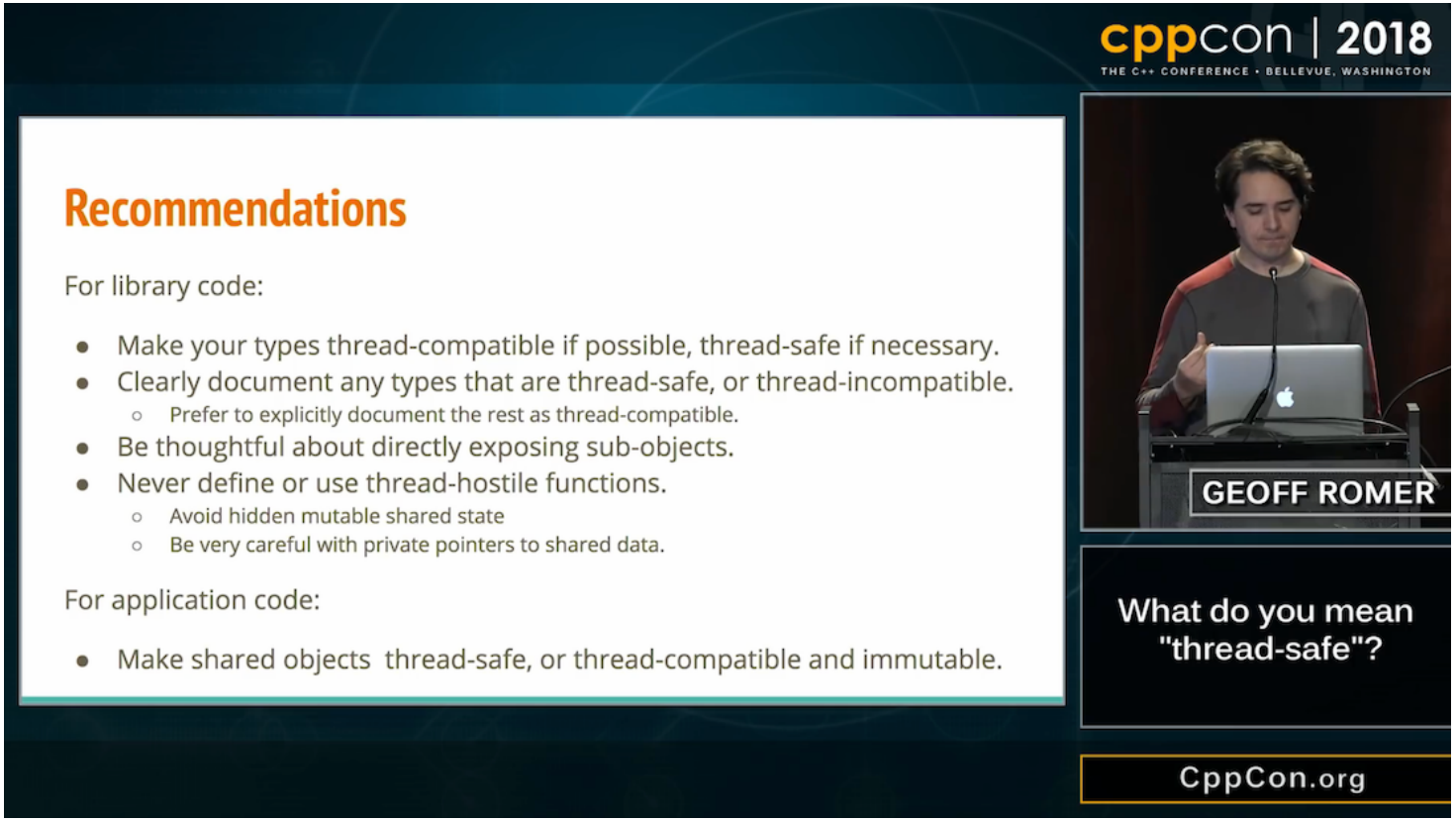
CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



CppCon 2018 - Geoff Romer - What do you mean "thread-safe"?

https://youtu.be/s5PCh_FaMfM



Proper way to iterate backwards in C++

https://www.reddit.com/r/cpp/comments/947a1z/proper_way_to_do_backward_iteration_in_c/

```
for (size_t i = data.size() - 1; i >= 0; --i) { ... } // Nope
for (size_t i = data.size(); i--;) { ... } // The C way

// C++17
std::vector<int> vec;
for (auto [value, idx] : reverse_index_adapter(vec)) {
    // idx = n-1, n-2, ... 0
}

std::for_each(vec.rbegin(), vec.rend(), []() { ... }); // No index

for (auto it = data.rbegin(); it != data.rend(); ++it) {
    auto i = std::distance(it, data.rend()) - 1;
}
```

https://youtu.be/S7I66IZX_zM

Inverse two-phase initialisation

```
class Foo
{
    static expected<construction_token>
    preconstruct(Arg n_arg) noexcept
    {
        construction_token t;
        t.state = make_unique_nothrow(n_arg);
        if (!t.state) return unexpected(my_errc::error);
        return t;
    }

    Foo(construction_token&& t) noexcept
    : m_state(std::move(t.state)) {}
};
```

https://youtu.be/S7I66IZX_zM

Inverse two-phase initialisation: usage

```
// 1
expected<Foo::construction_token> t1 = Foo::preconstruct(args);
if (!t1.has_value()) { /* get out */ }
Foo obj(std::move(*t1));

// 2
auto t2 = Foo::preconstruct(args);
auto obj_ptr = std::make_shared<Foo>(std::move(*t2));

// 3
auto t3 = Foo::preconstruct(args);
std::vector<Foo> objects;
objects.emplace_back(std::move(*t3));
```

https://youtu.be/tsG95Y-C14k

```
// MSVC rejects, GCC accepts, Clang accepts
extern extern "C++" extern "C" extern "C++" int x;

// MSVC accepts, GCC rejects, Clang accepts
extern "C++" extern "C" extern "C++" extern int x;
```

https://youtu.be/tsG95Y-C14k

selection-statement:

`if` `constexpr`$_{opt}$ (*init-statement*$_{opt}$ *condition*) *statement*

```
if (class foo; !ret.second) /* ... */;

if (false; true) /* ... */;

if (; true) /* ... */;
```

https://youtu.be/tsG95Y-C14k

Declare and initialise a variable of type "function pointer":

```
1 auto (*fp)() -> int(&f);
```

https://youtu.be/tsG95Y-C14k

```
1 struct foo;
2 void bar(foo foo);
3 void bar(foo(foo)); // vexing parse
4 void bar(foo((foo))); // more vexing parse
```

https://youtu.be/tsG95Y-C14k

```
class bar {};
int bar;     // OK
bar b;       // error
class bar b; // OK
class std::vector<class bar> bars; // OK

// also acts as a forward declaration
void foo(struct S* x);

// weird scoping rules
class C { void foo(struct S* x); };
S* s;
```

https://youtu.be/tsG95Y-C14k

pseudo-destructor-name

```
int i;
i.~int(); // error: int is not type-name but type-specifier
using foo = int;
i.~foo(); // OK
~int();   // OK (not a destructor!)
```

https://youtu.be/tsG95Y-C14k

Alternative tokens

```
struct Foo
{
    Foo();
    compl Foo();
    Foo(const Foo bitand);
    Foo(Foo and);
};
```

CppCon 2018: Matt Godbolt “The Bits Between the Bits: How We Get to main()”

https://youtu.be/dOfucXtyEsU

https://arne-mertz.de/2019/02/extern-template-reduce-compile-times/

Move smart pointers in and out functions in modern C++

- Move smart pointers in and out functions in modern C++
  - Reddit: https://www.reddit.com/r/cpp/comments/aaux96/move_smart_pointers_in_and_out_functions_in/

Twitter

Lynn
@chordbug

🌶️ HOT JAVASCRIPT TIP: 🌶️

to increment some counter on the page,

```
node.innerText += 1
```

doesn't work (0 → 01 → 011 → ⋯), but

```
node.innerText -= -1
```

works fine (0 → 1 → 2 → ⋯)

4,127 Likes	**1,490** Retweets
5 Feb 2019 at 16:36	via **Twitter Web Client**