C++ Club UK

Gleb Dolgich

2019-02-07

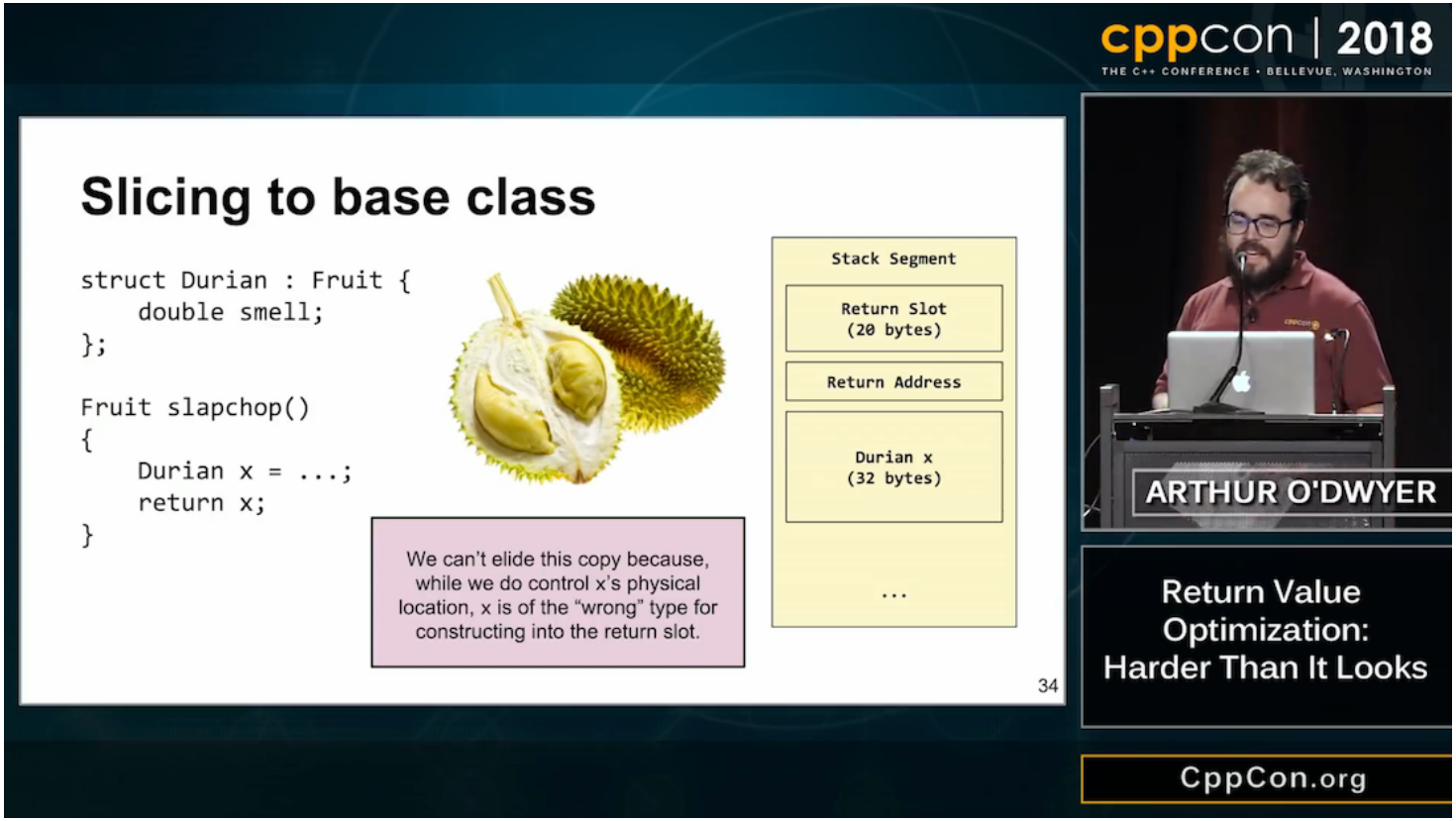
Celebrating Honoris Causa Doctorate at UC3M by Bjarne Stroustrup

The continuing evolution of C++

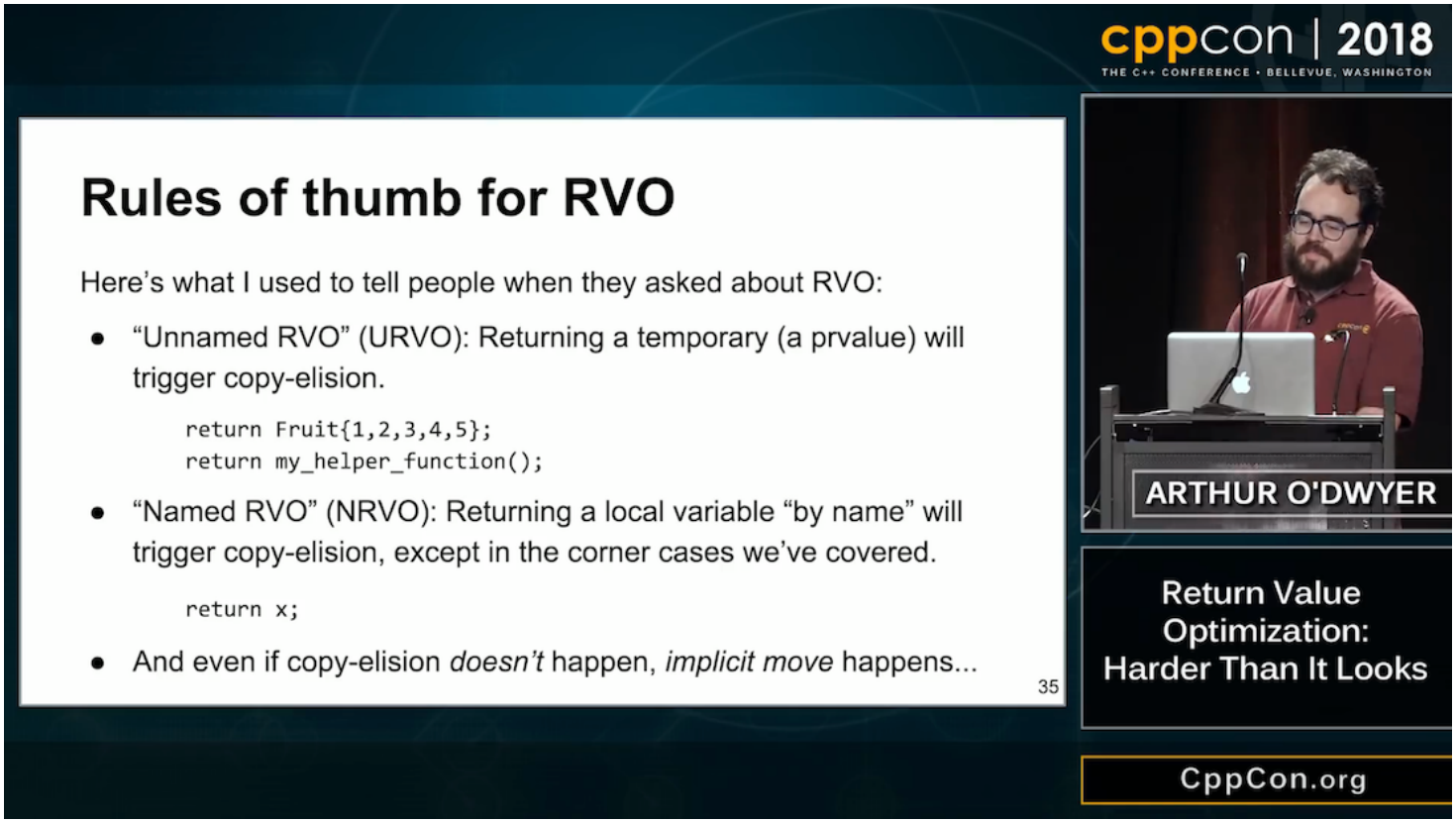
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"

- CppCon 2018 - Arthur O'Dwyer - RVO: Harder Than It Looks
- Trivially Relocatable FAQ

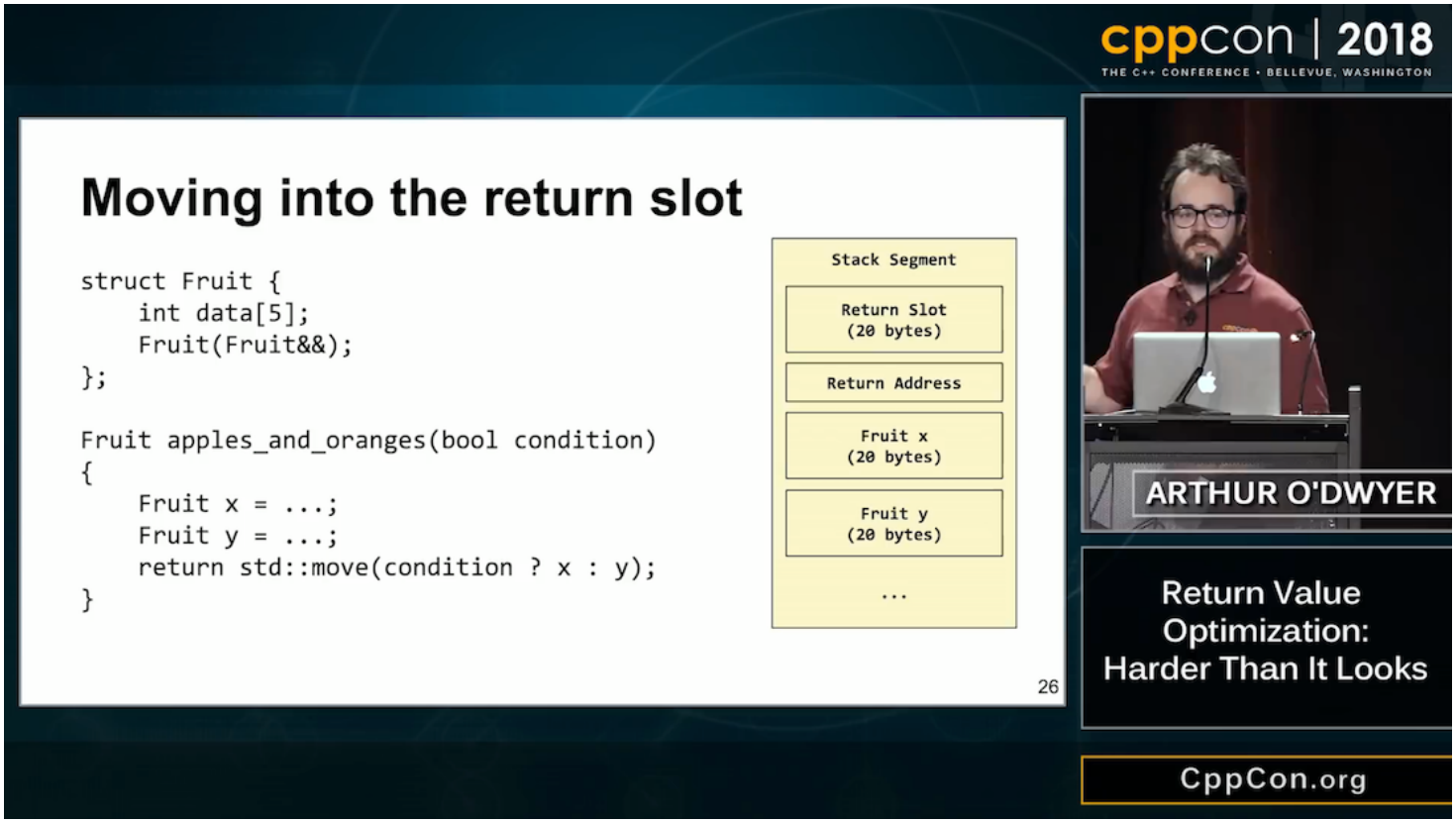
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



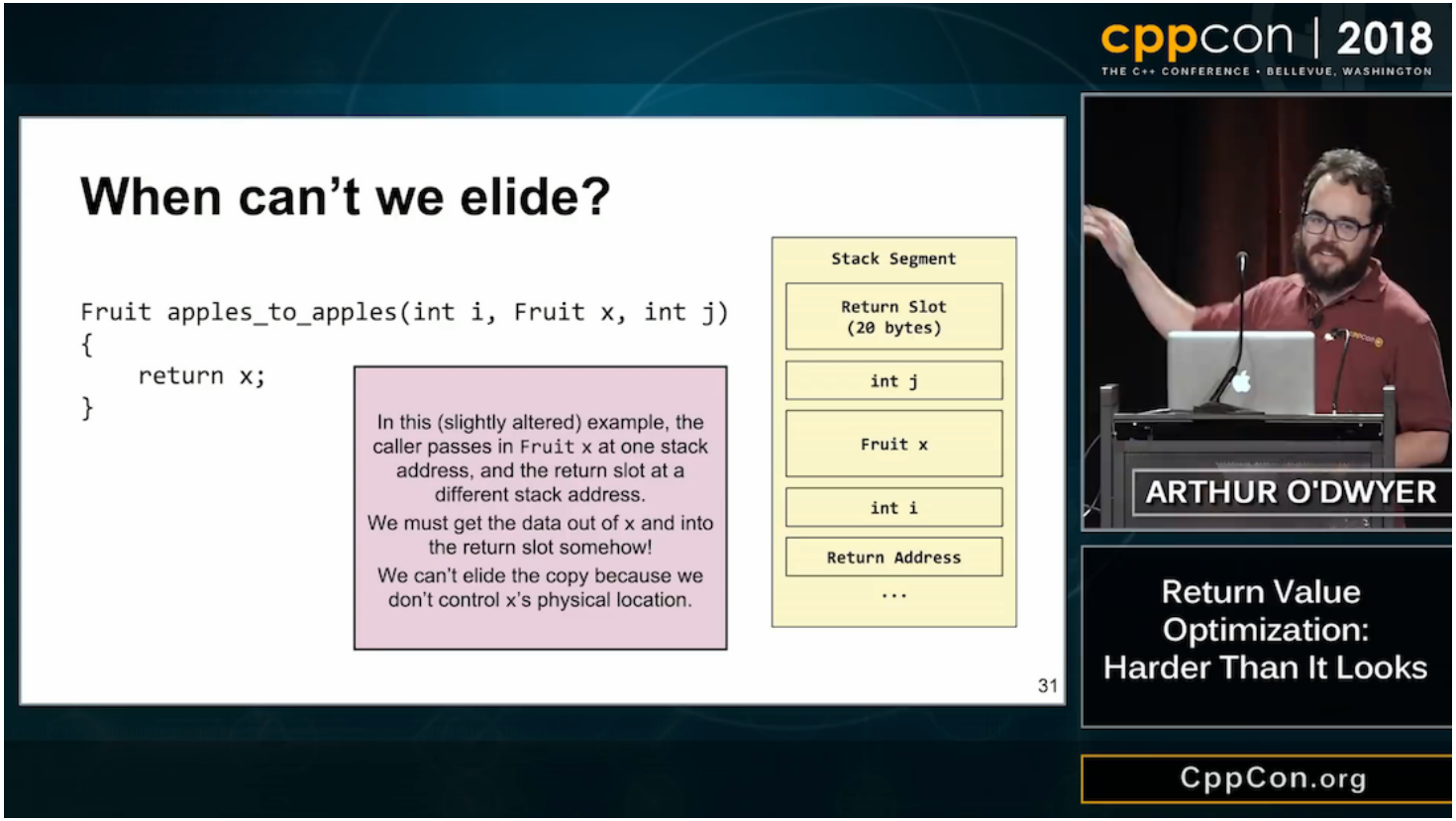
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



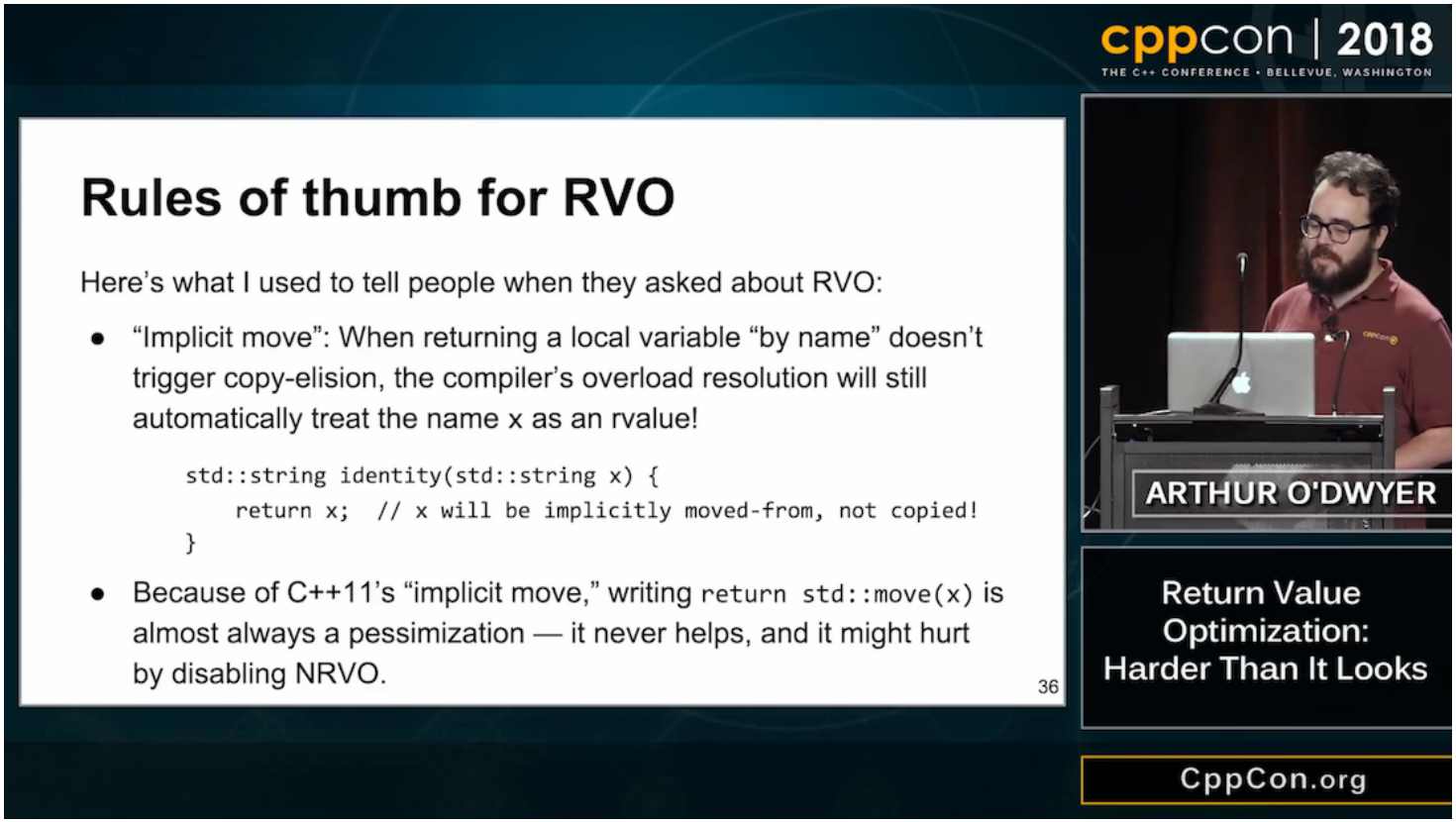
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



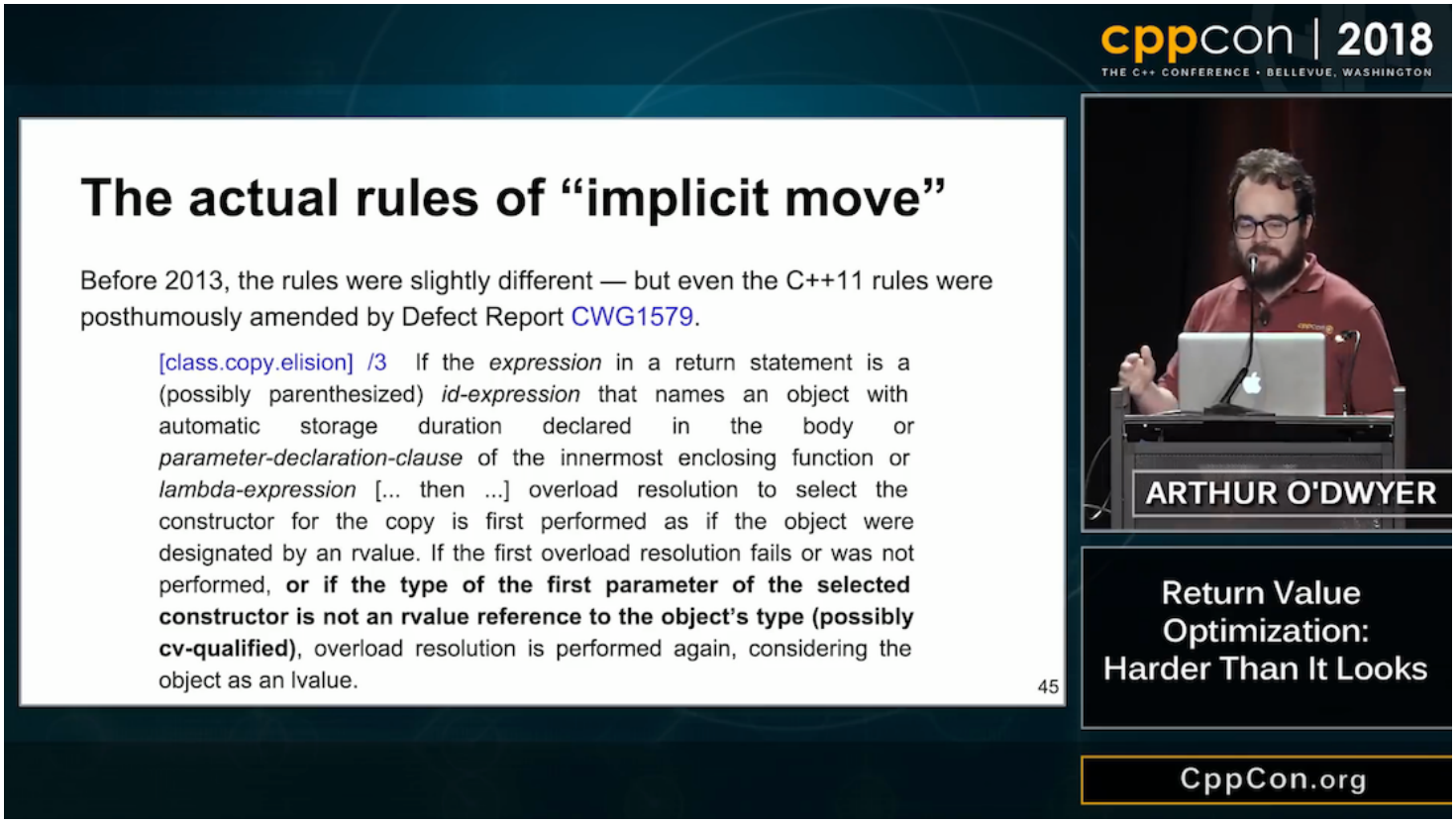
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



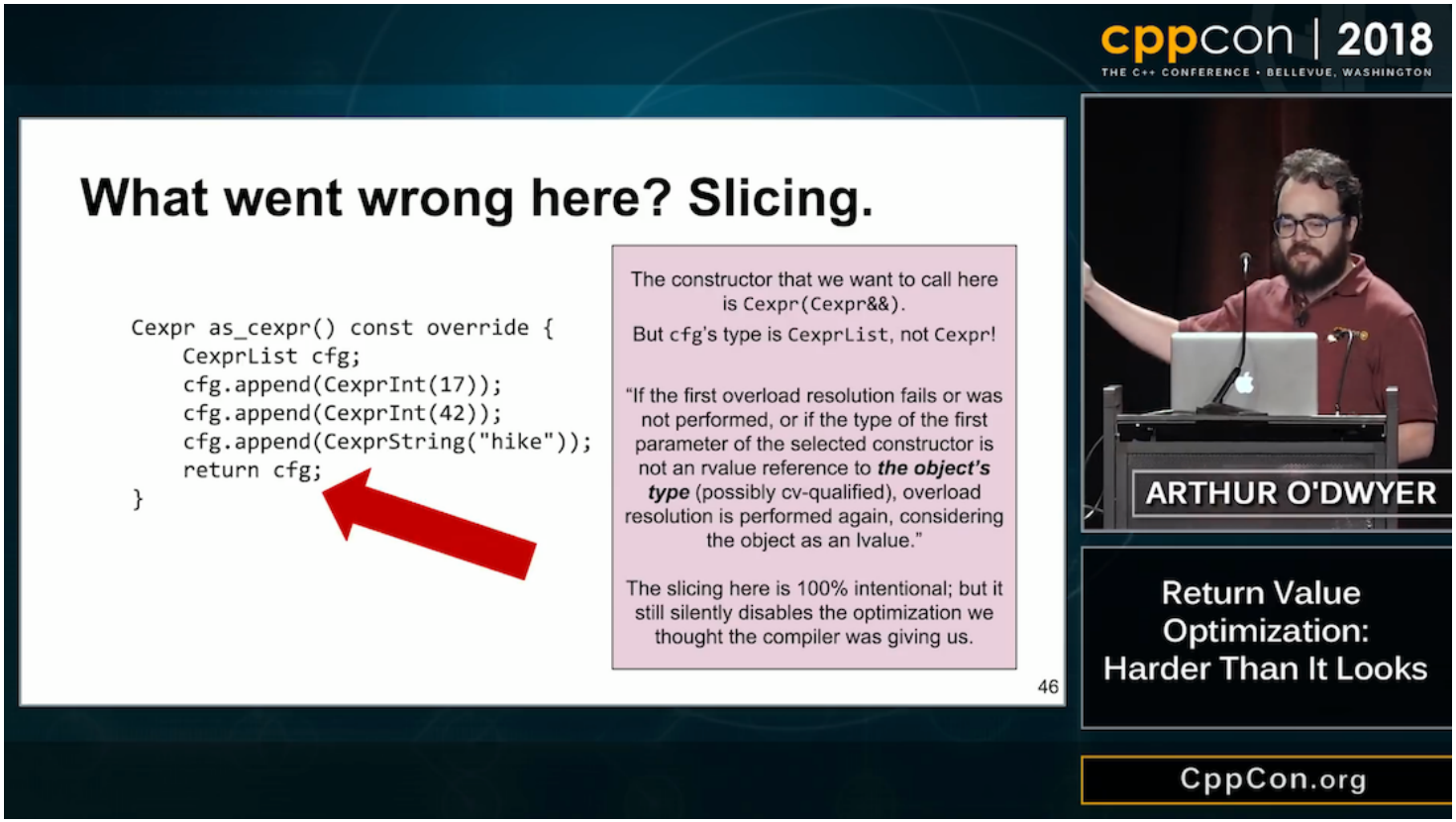
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



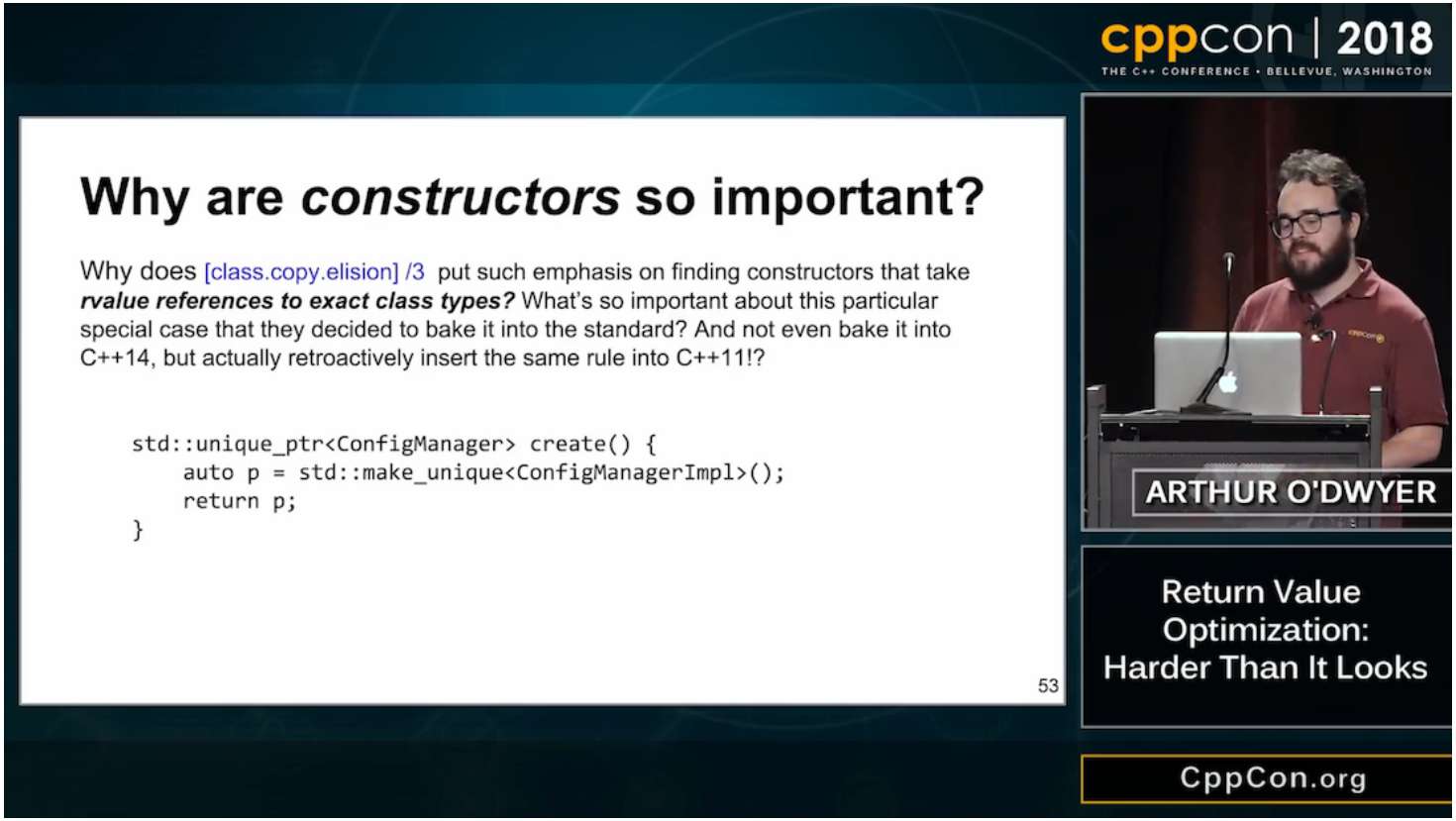
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



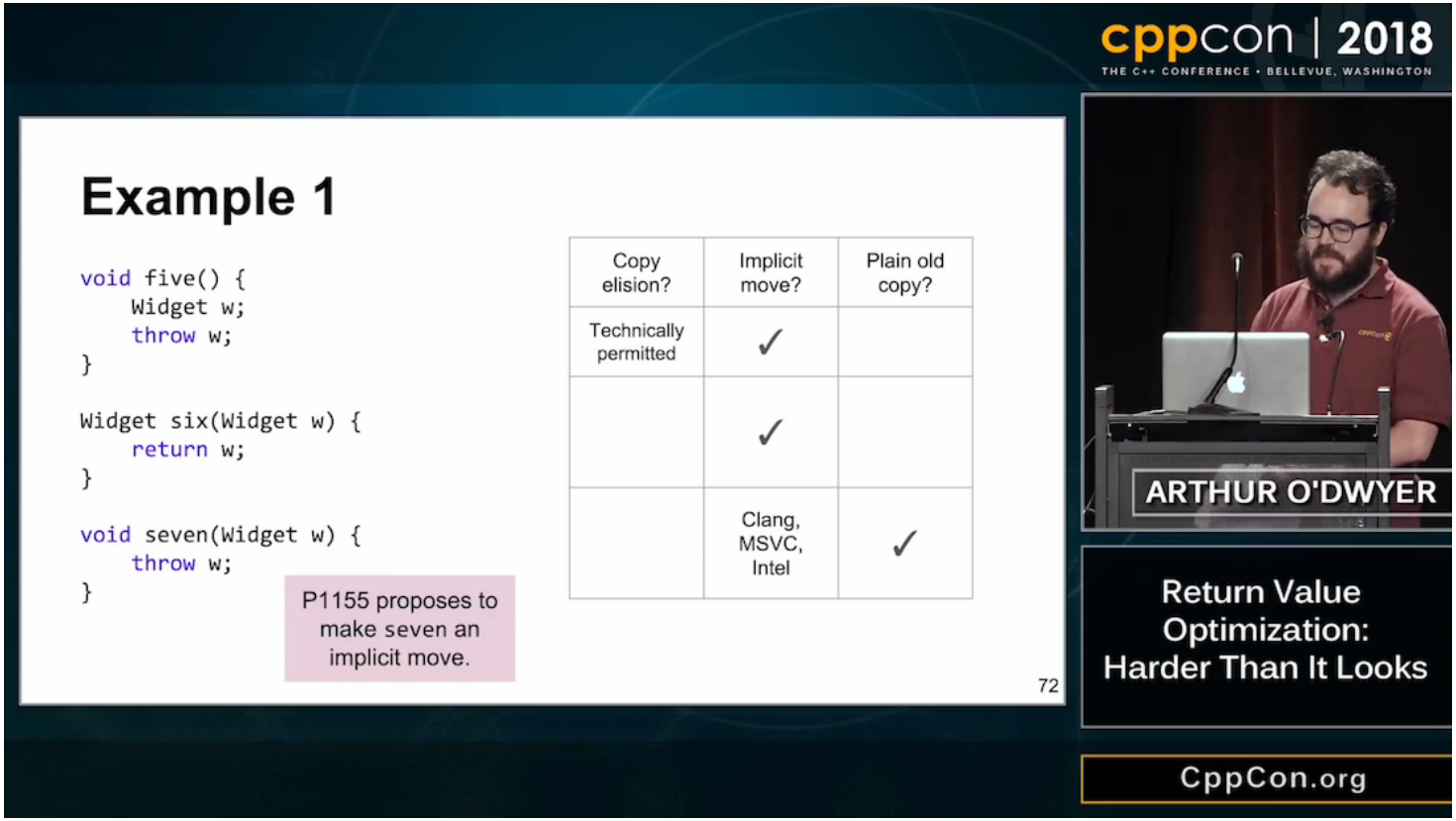
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



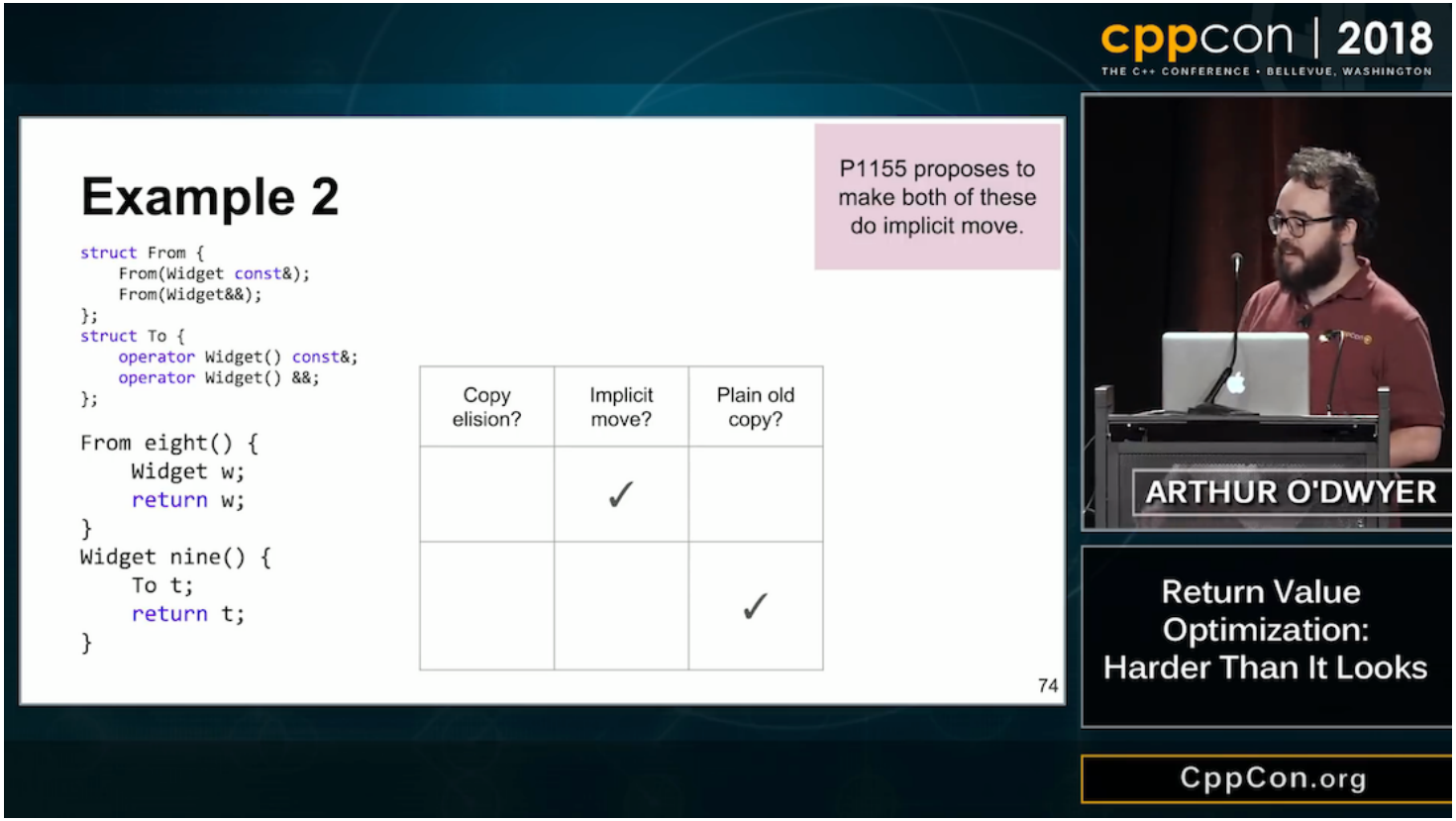
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



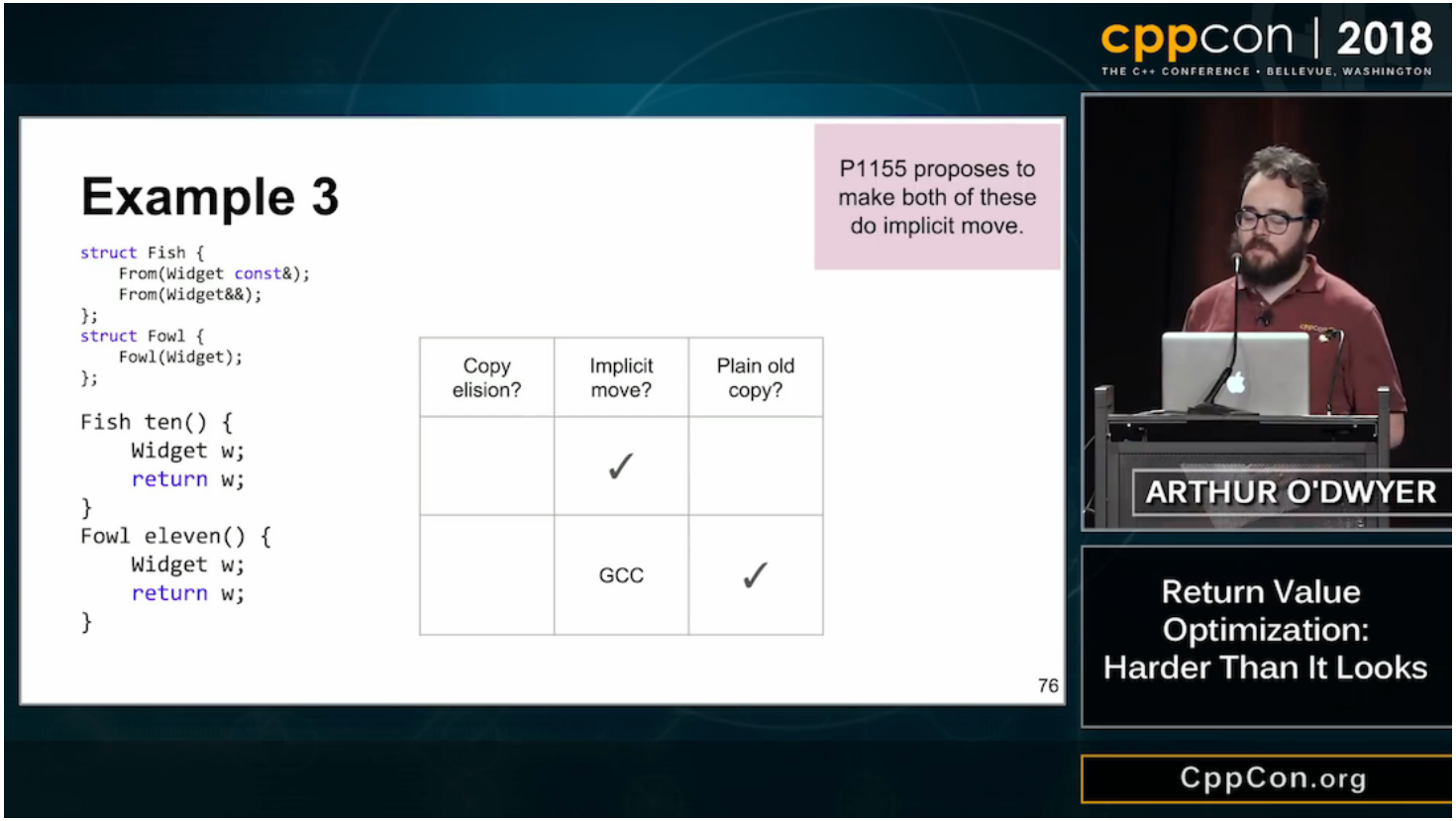
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



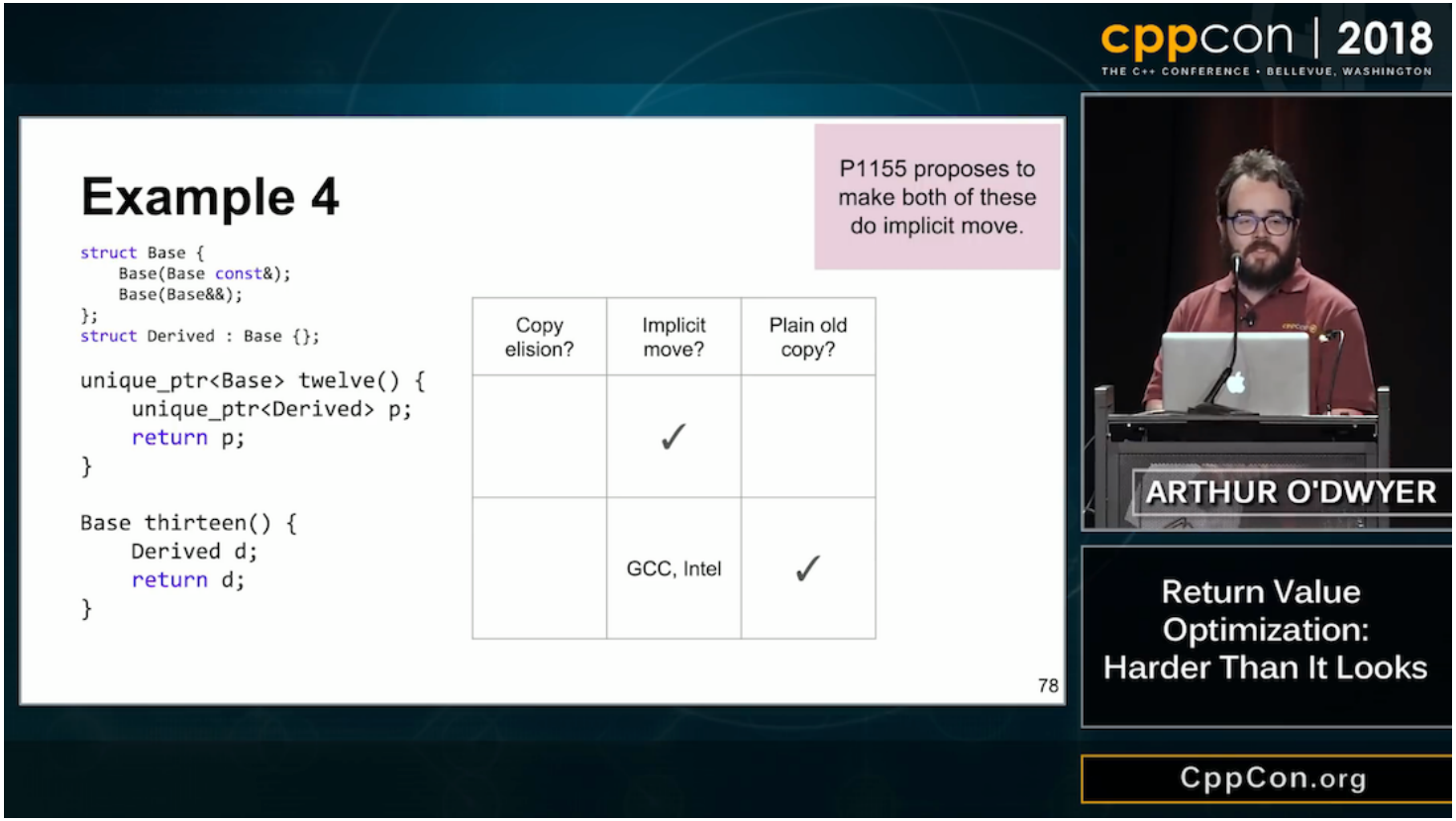
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



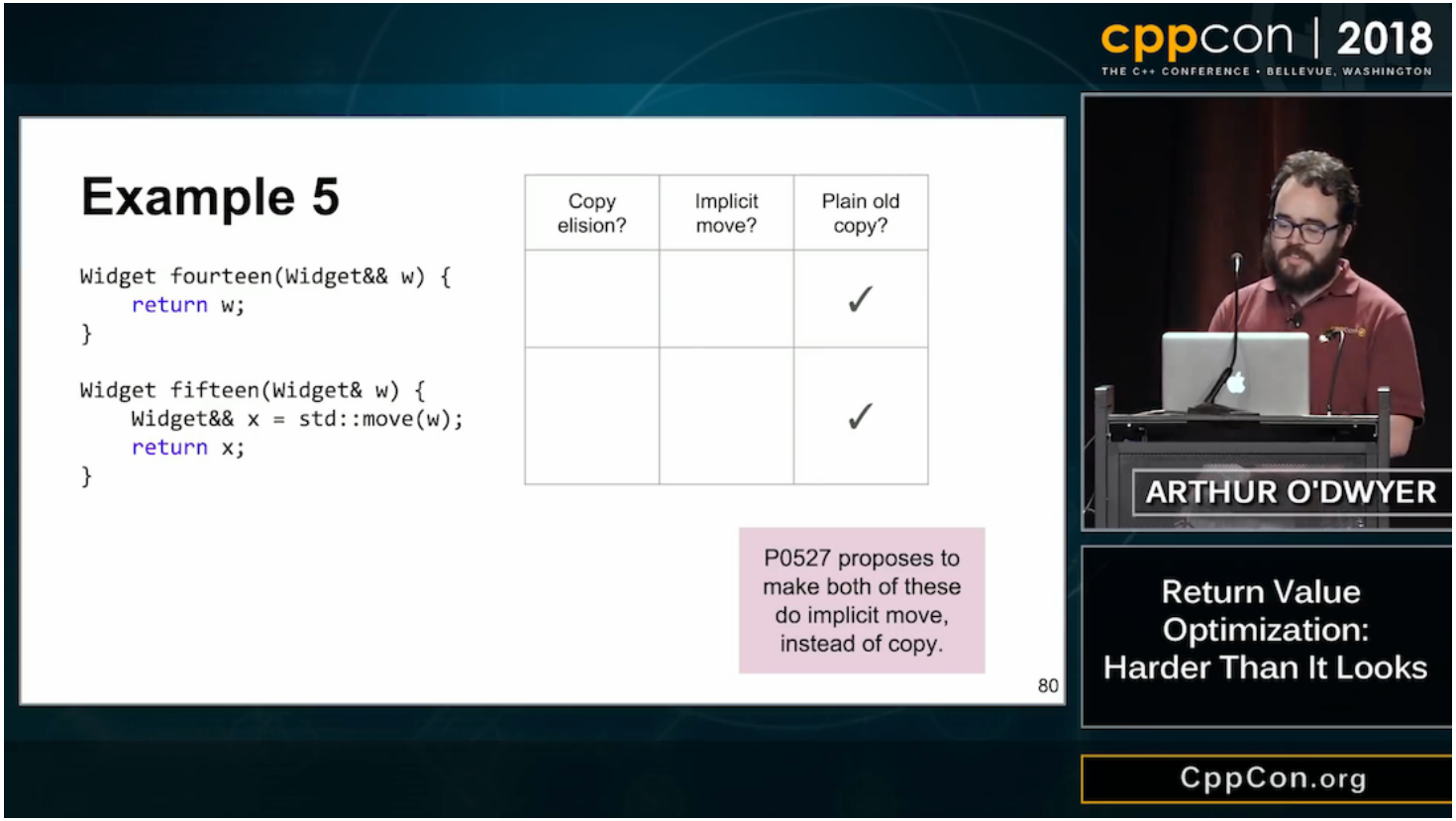
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



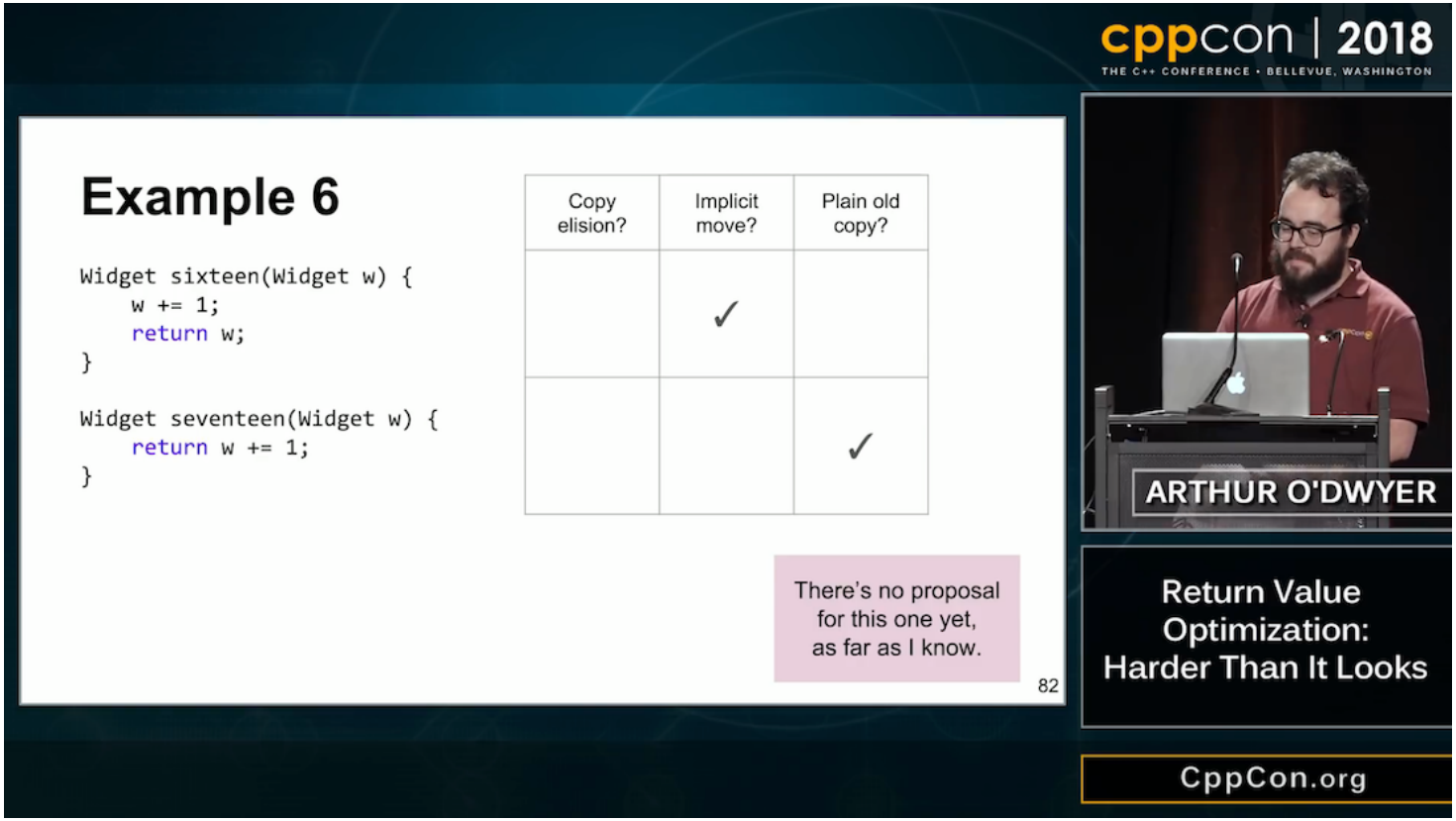
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



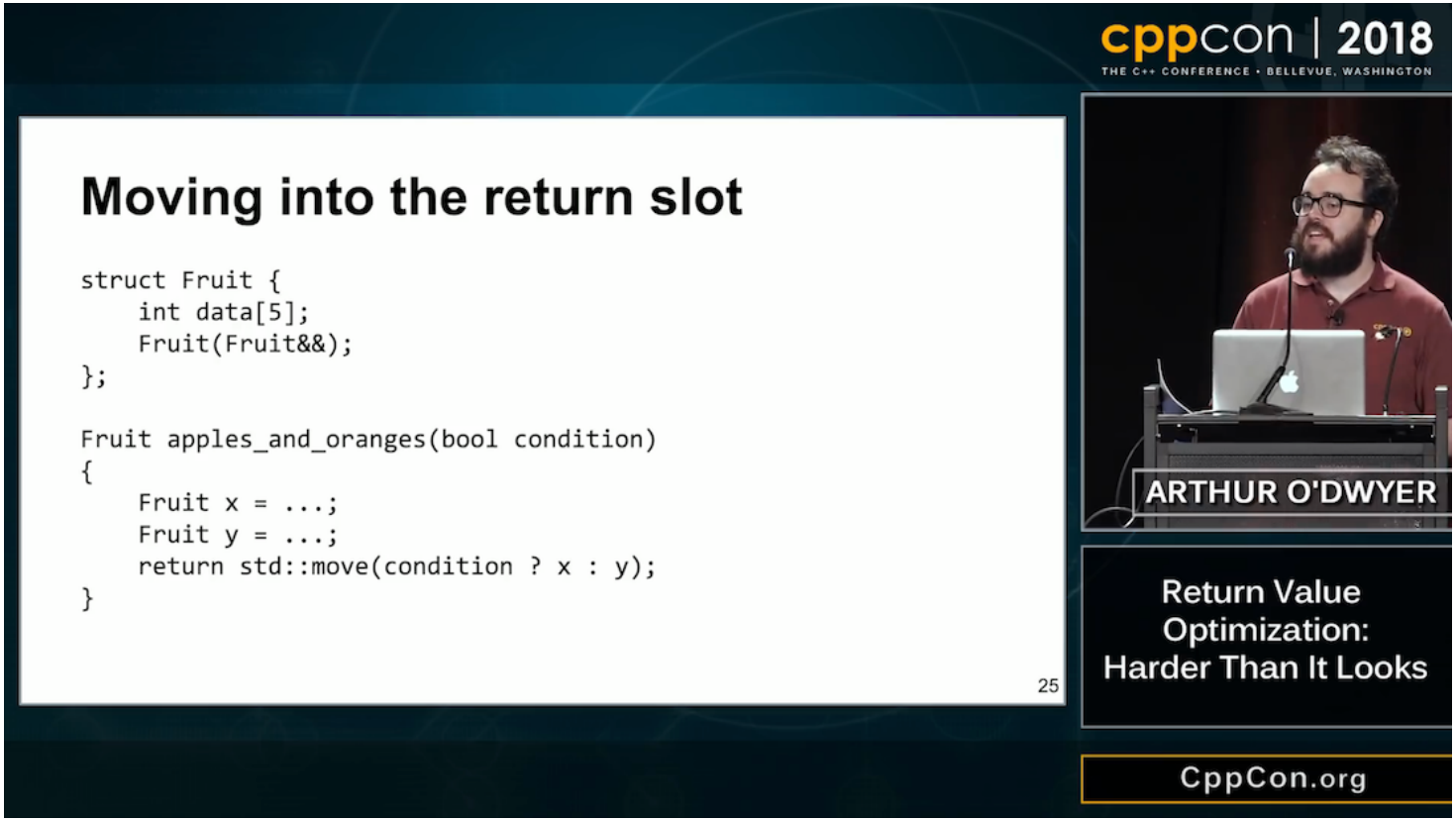
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"

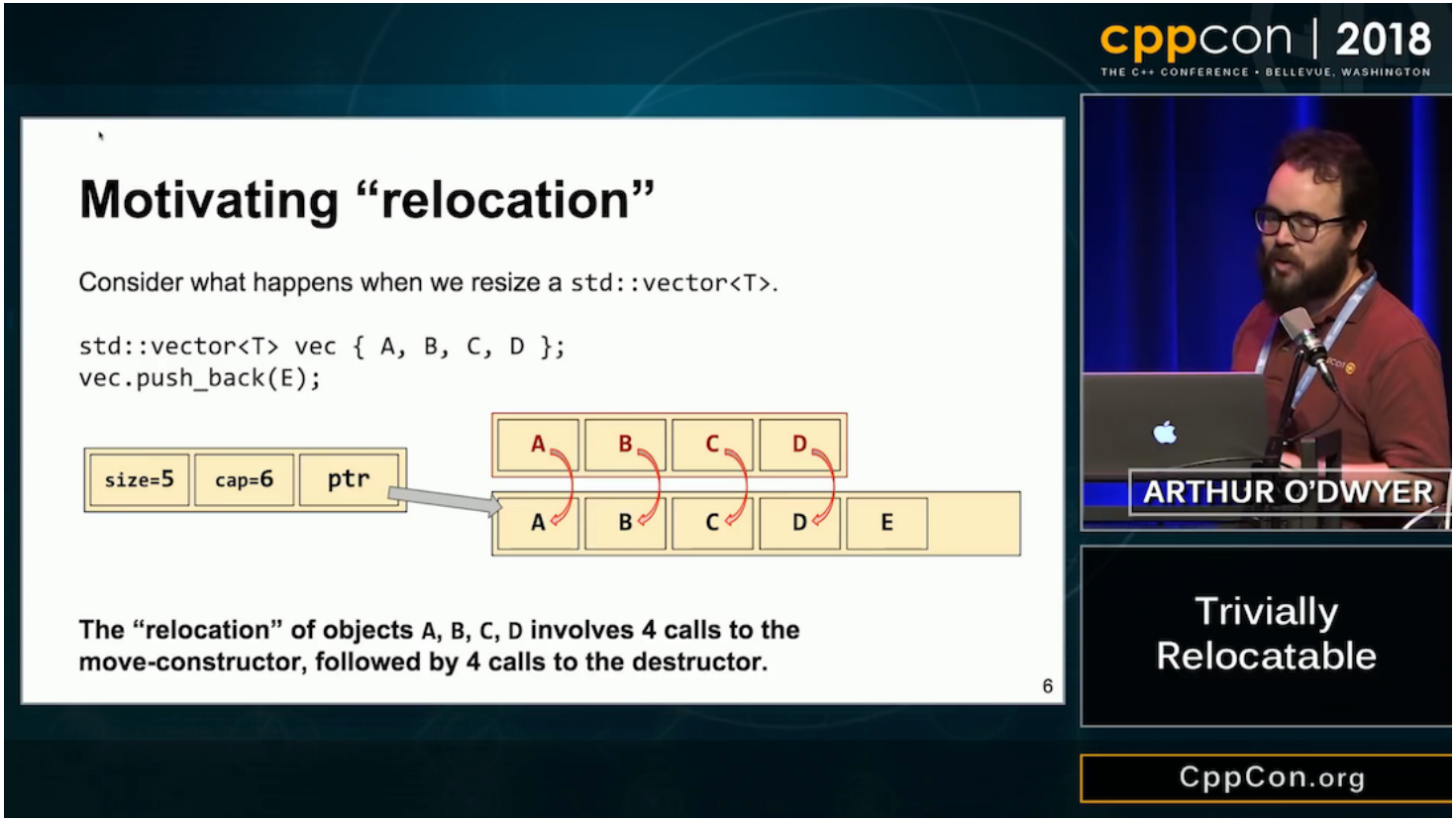


CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"

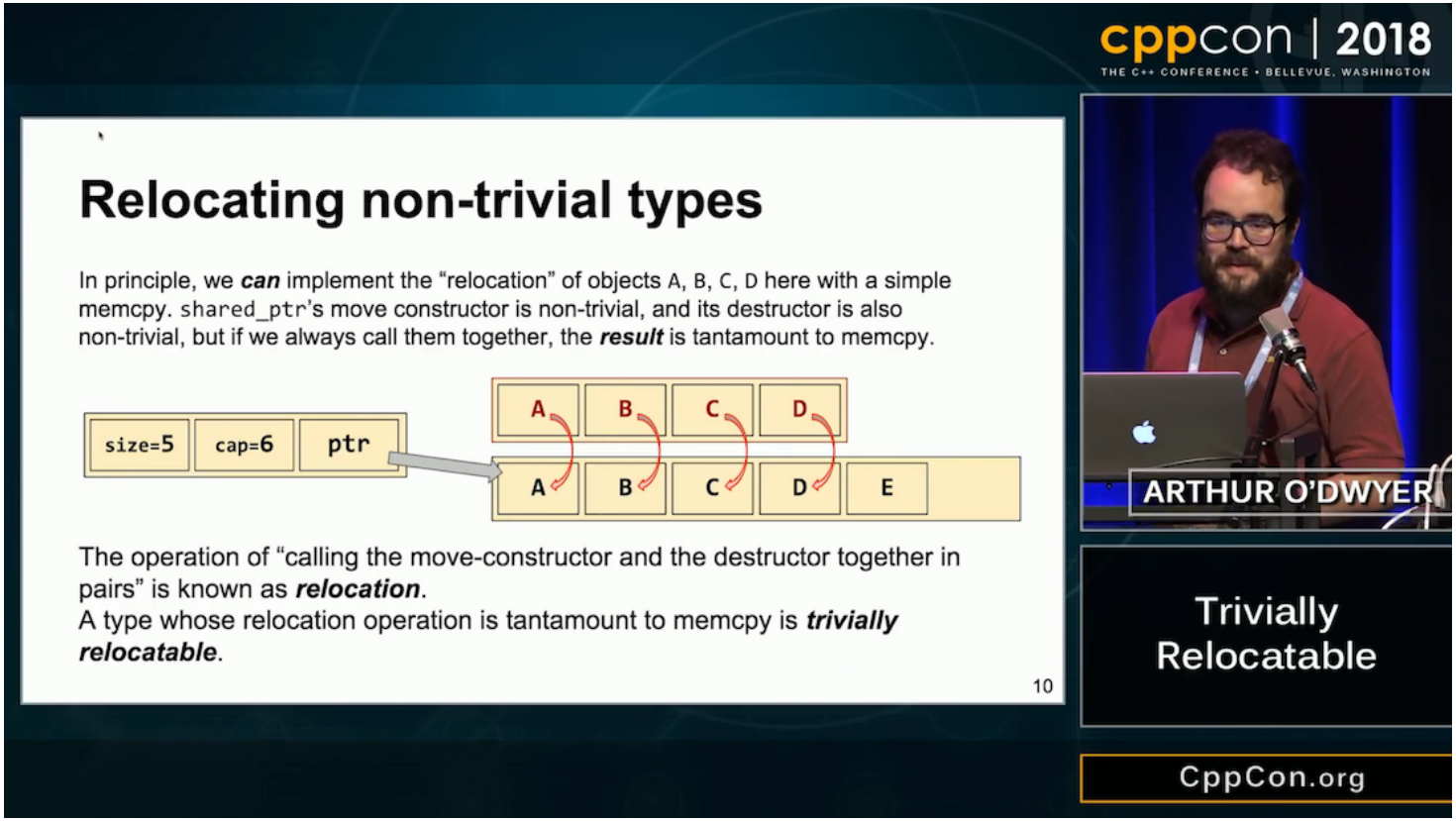


CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"




cppcon | 2018
THE C++ CONFERENCE • BELLEVUE, WASHINGTON
Motivating "relocation"
Consider what happens when we resize a std::vector<T>.
std::vector<T> vec { A, B, C, D };
vec.push_back(E);
size=5
cap=6
ptr
A
B
C
D
A
B
C
D
E
The "relocation" of objects A, B, C, D involves 4 calls to the move-constructor, followed by 4 calls to the destructor.
6
ARTHUR O'DWYER
Trivially Relocatable
CppCon.org

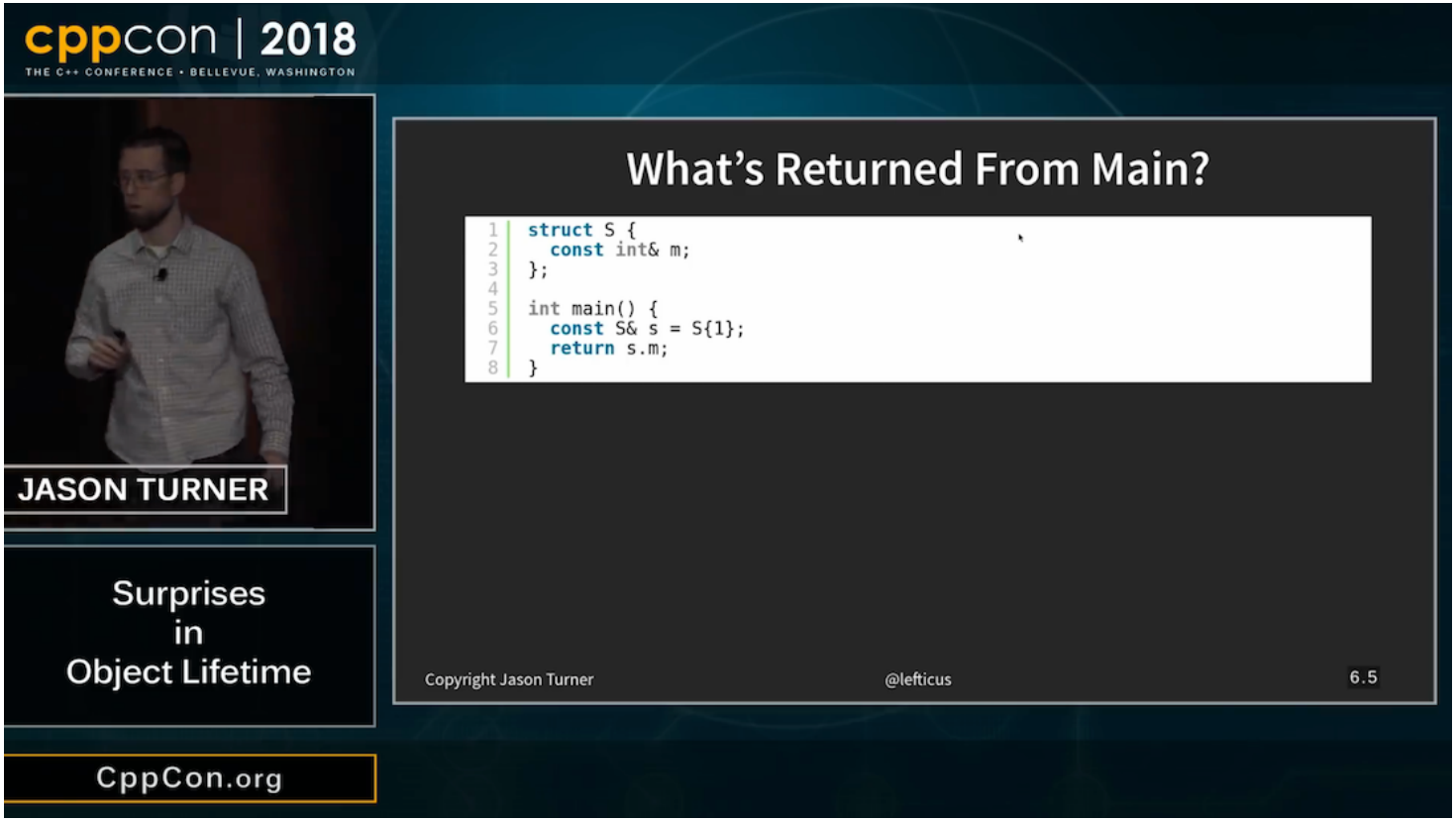
CppCon 2018: Arthur O'Dwyer "RVO: Harder Than It Looks"



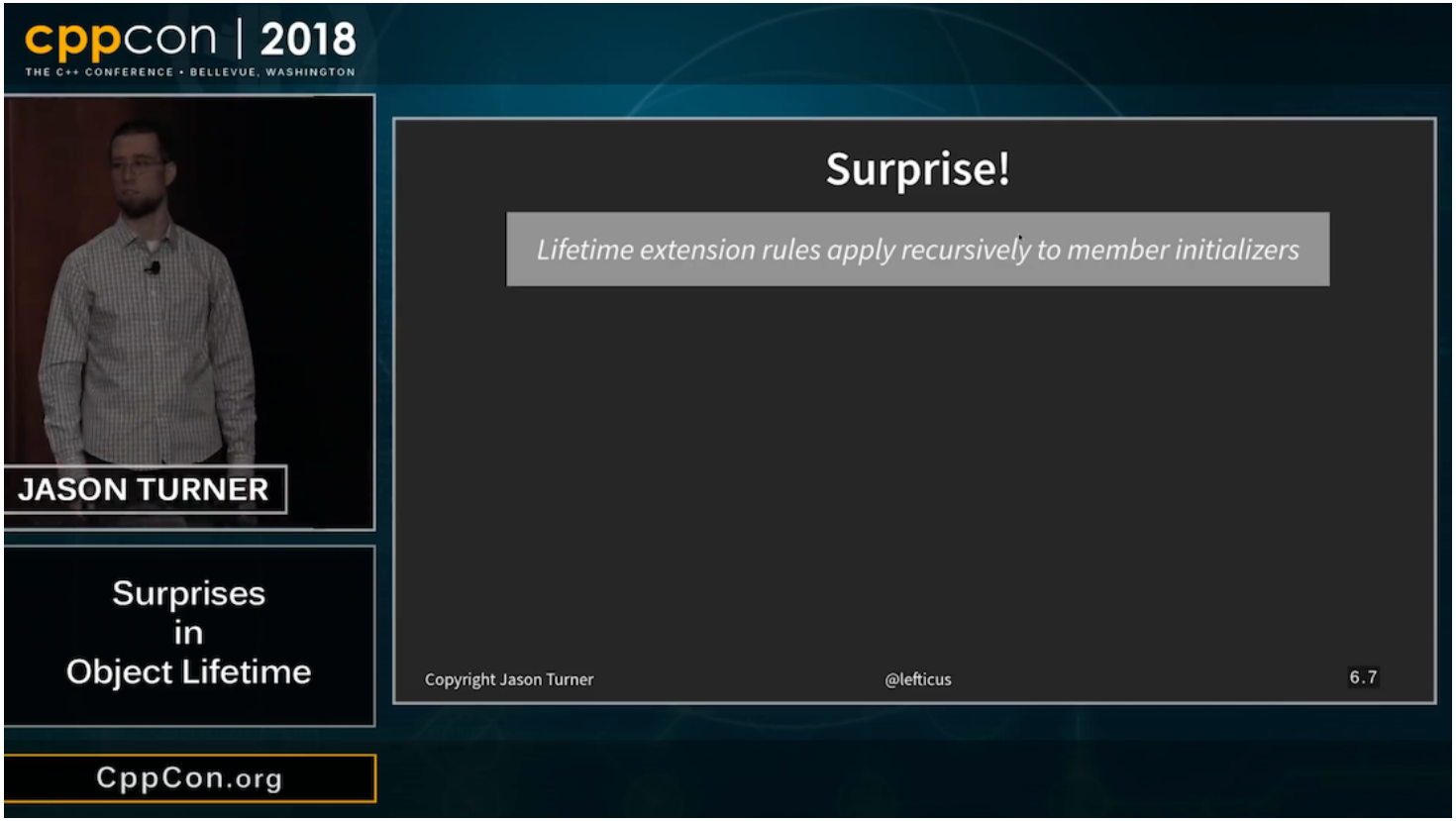
CppCon 2018: Jason Turner "Surprises in Object Lifetime"

CppCon 2018 - Jason Turner - Surprises in Object Lifetime

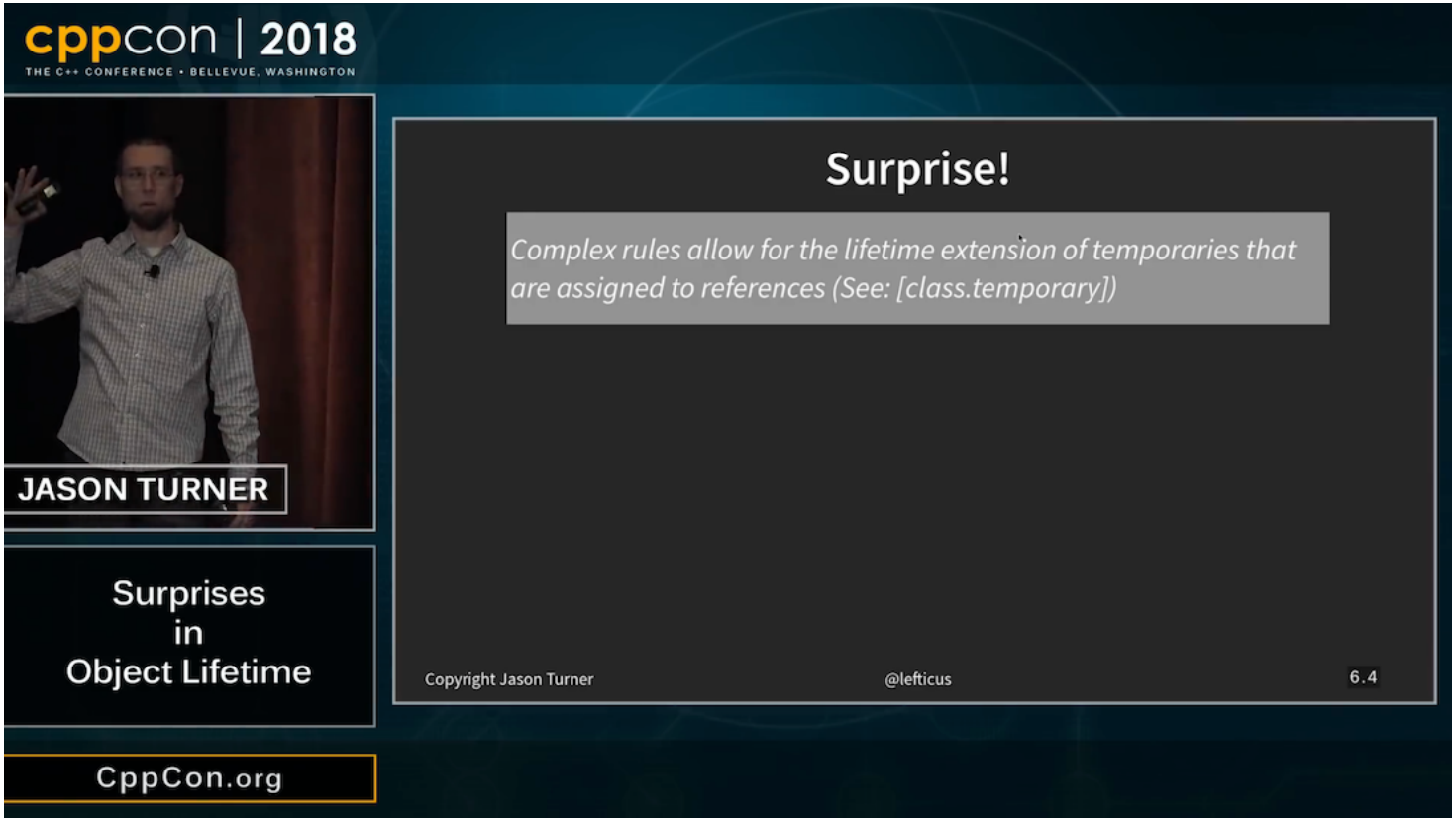
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



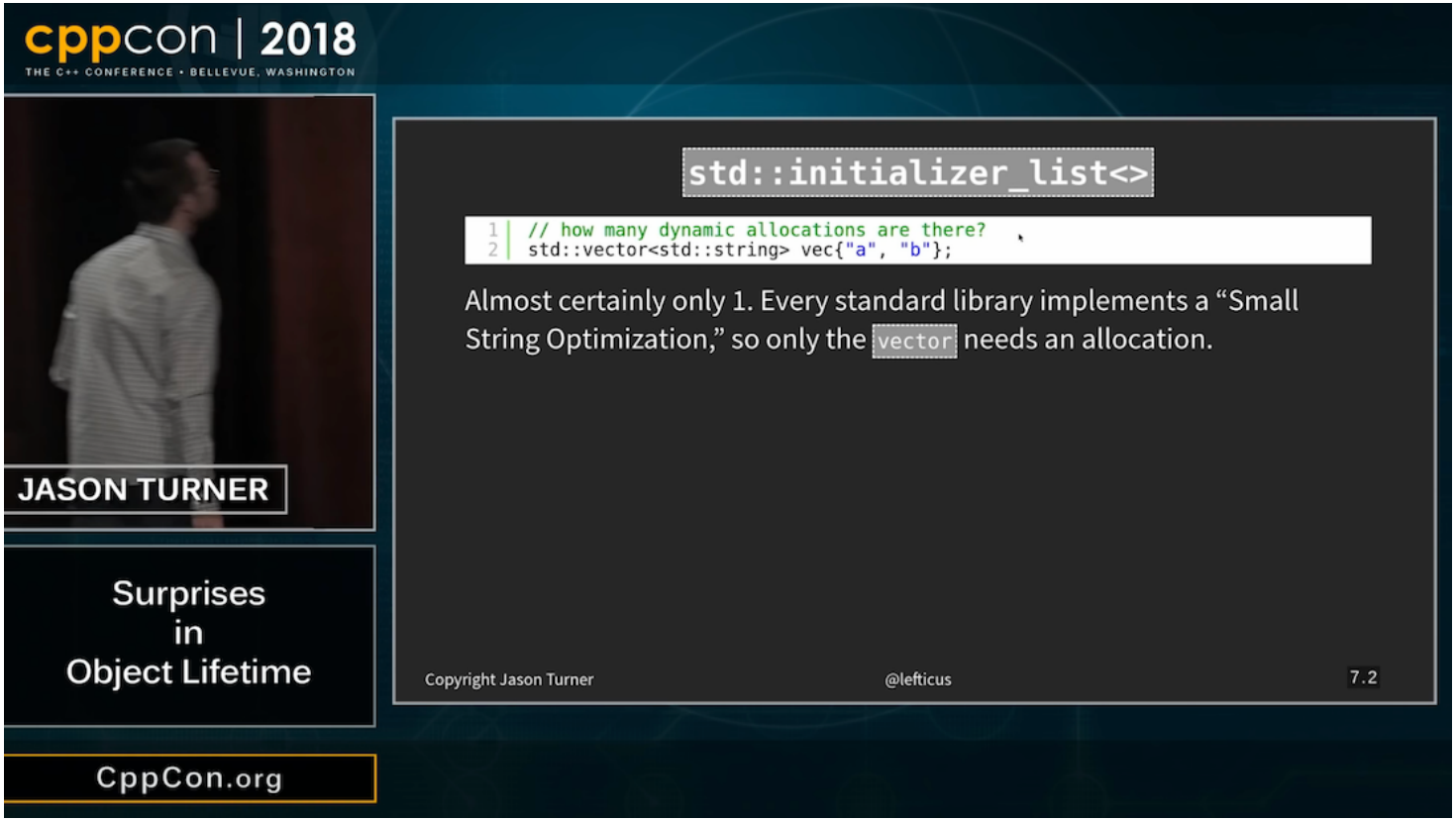
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



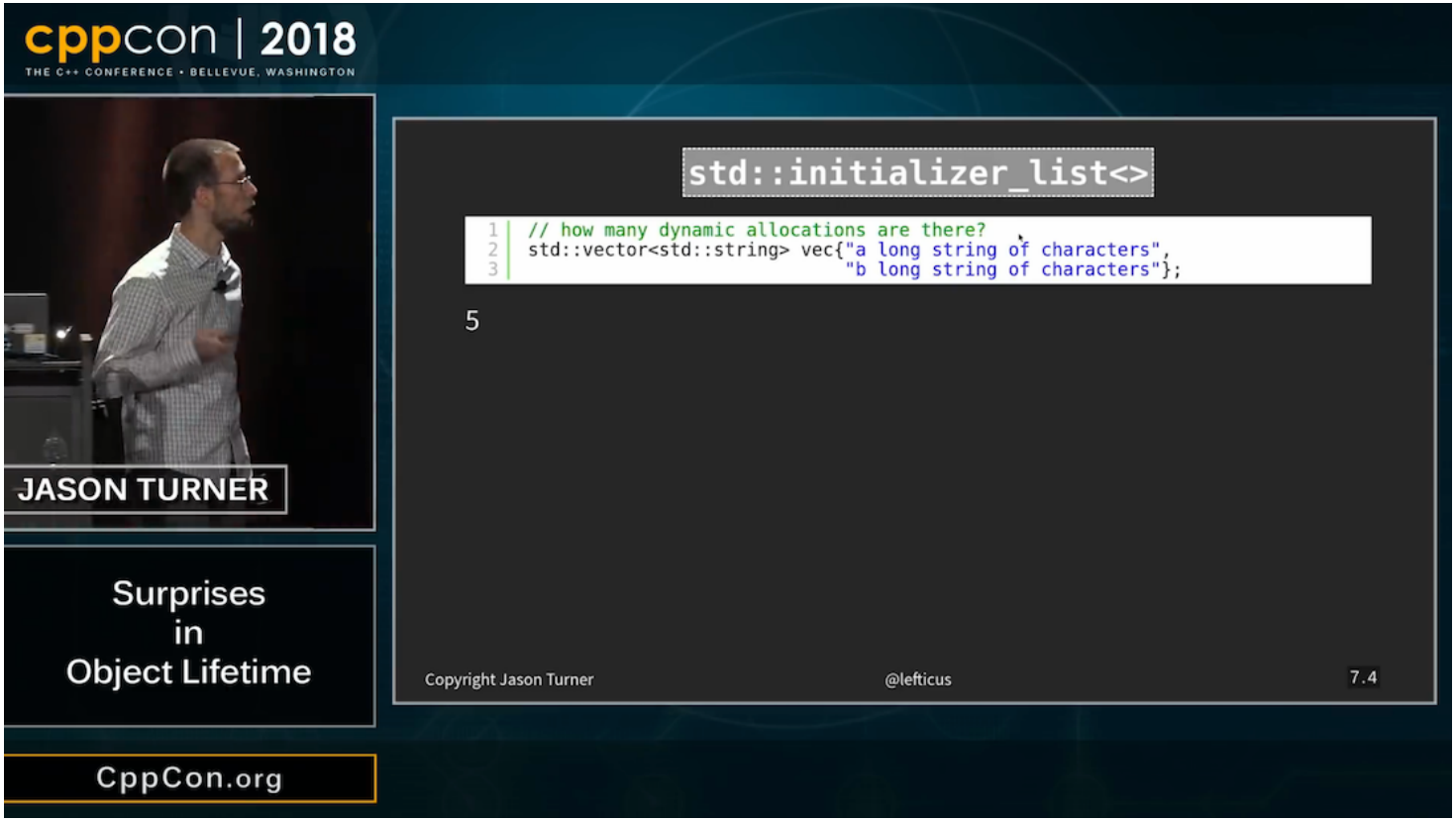
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



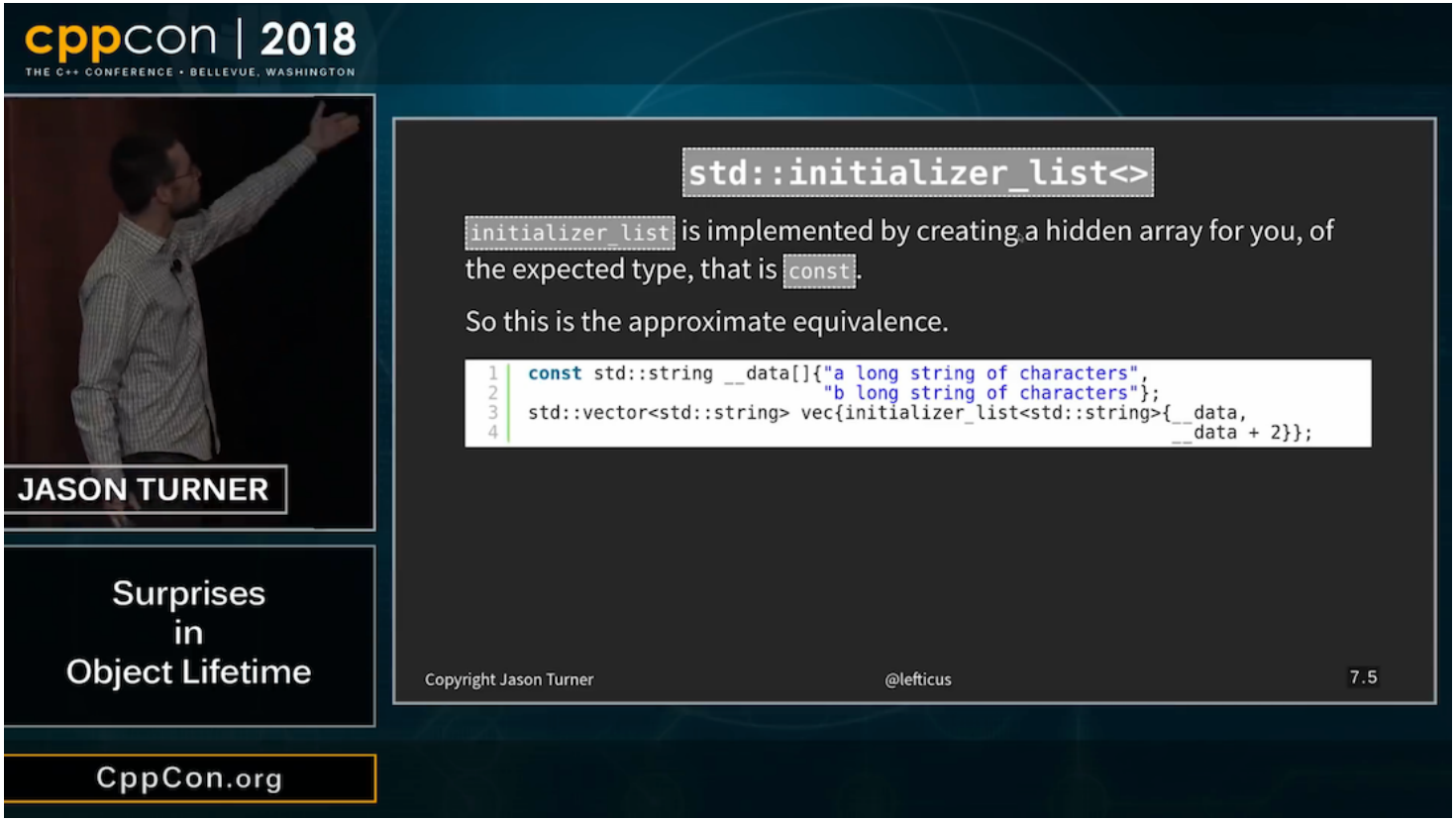
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



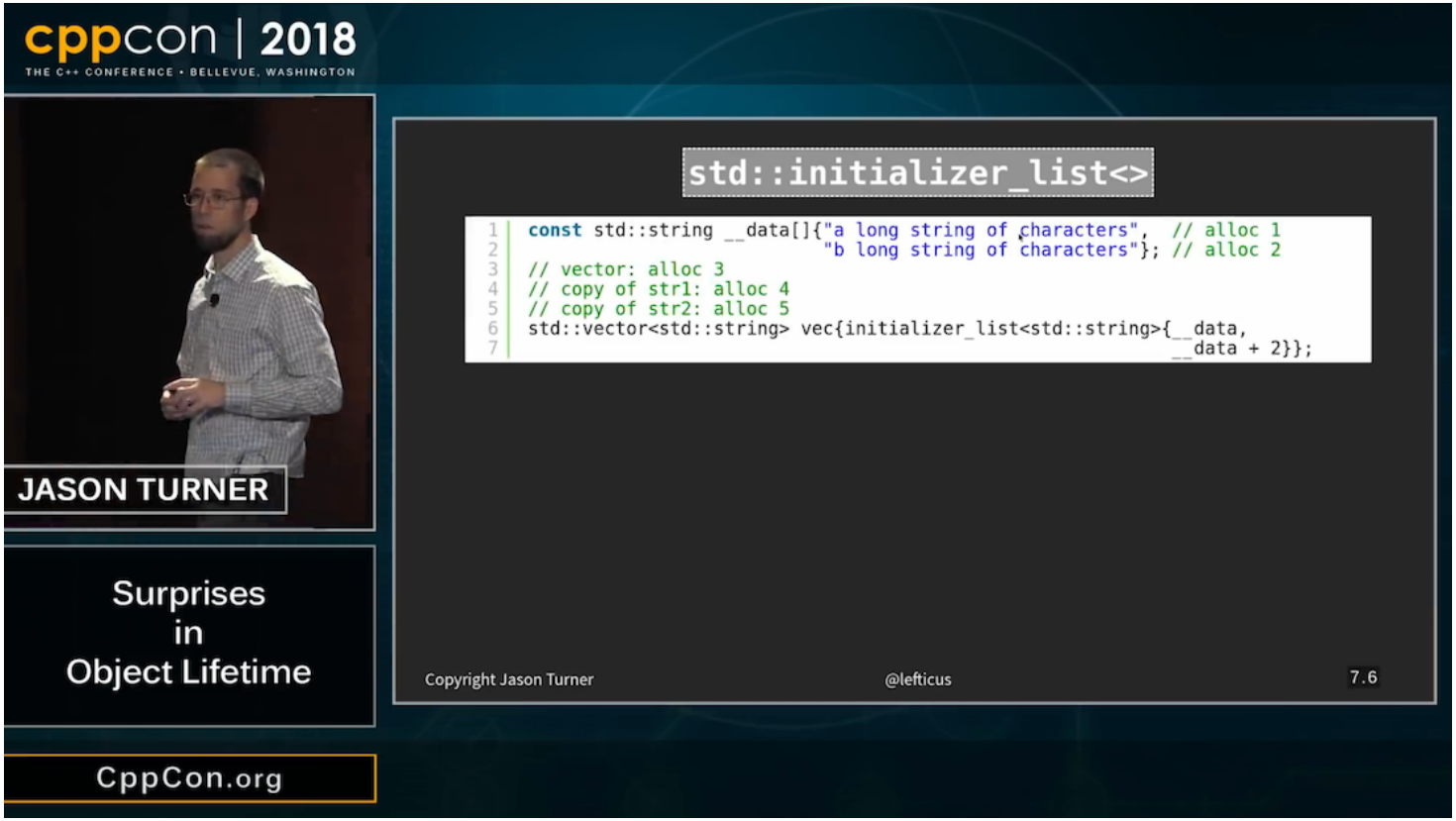
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



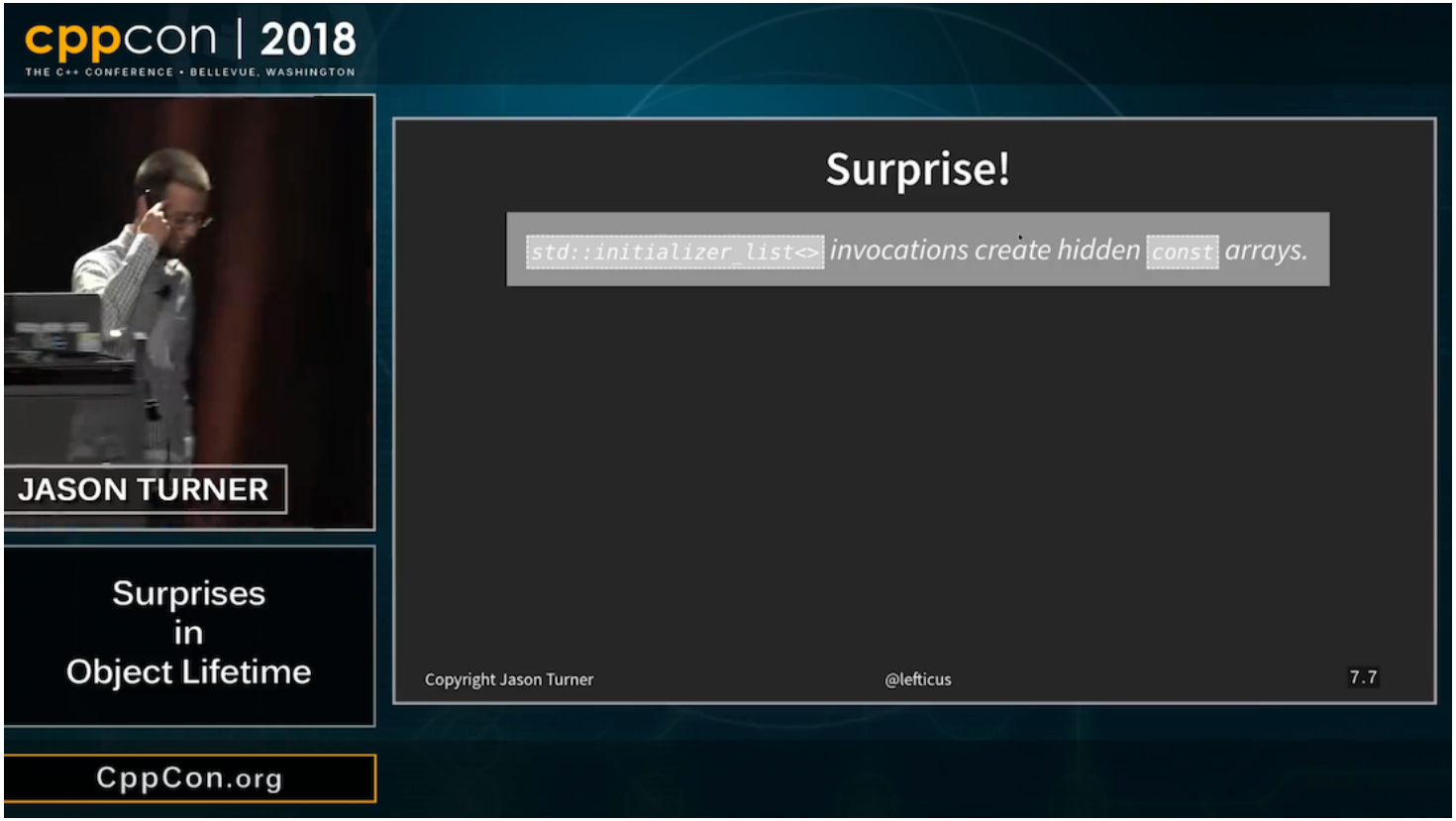
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



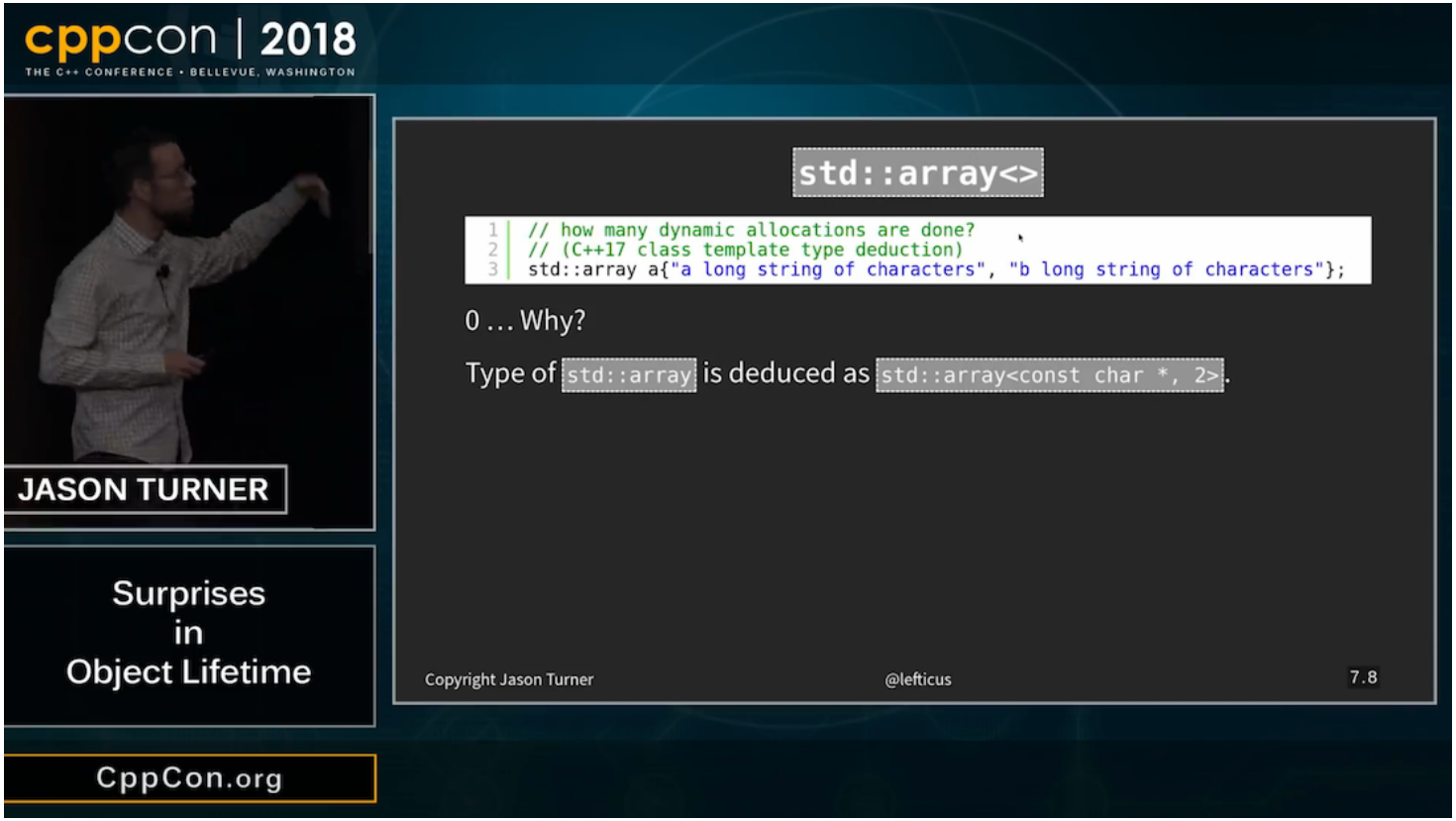
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



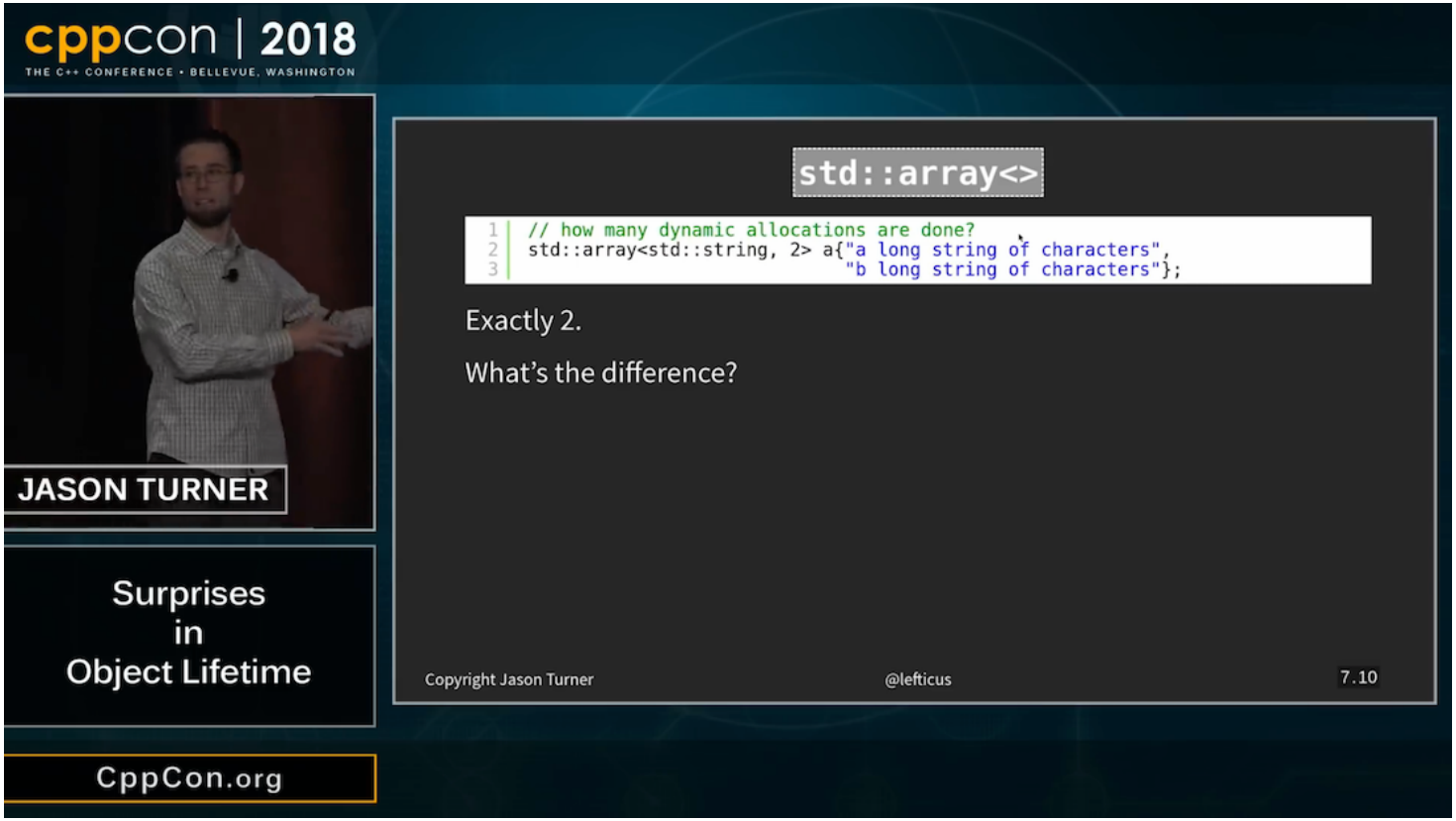
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



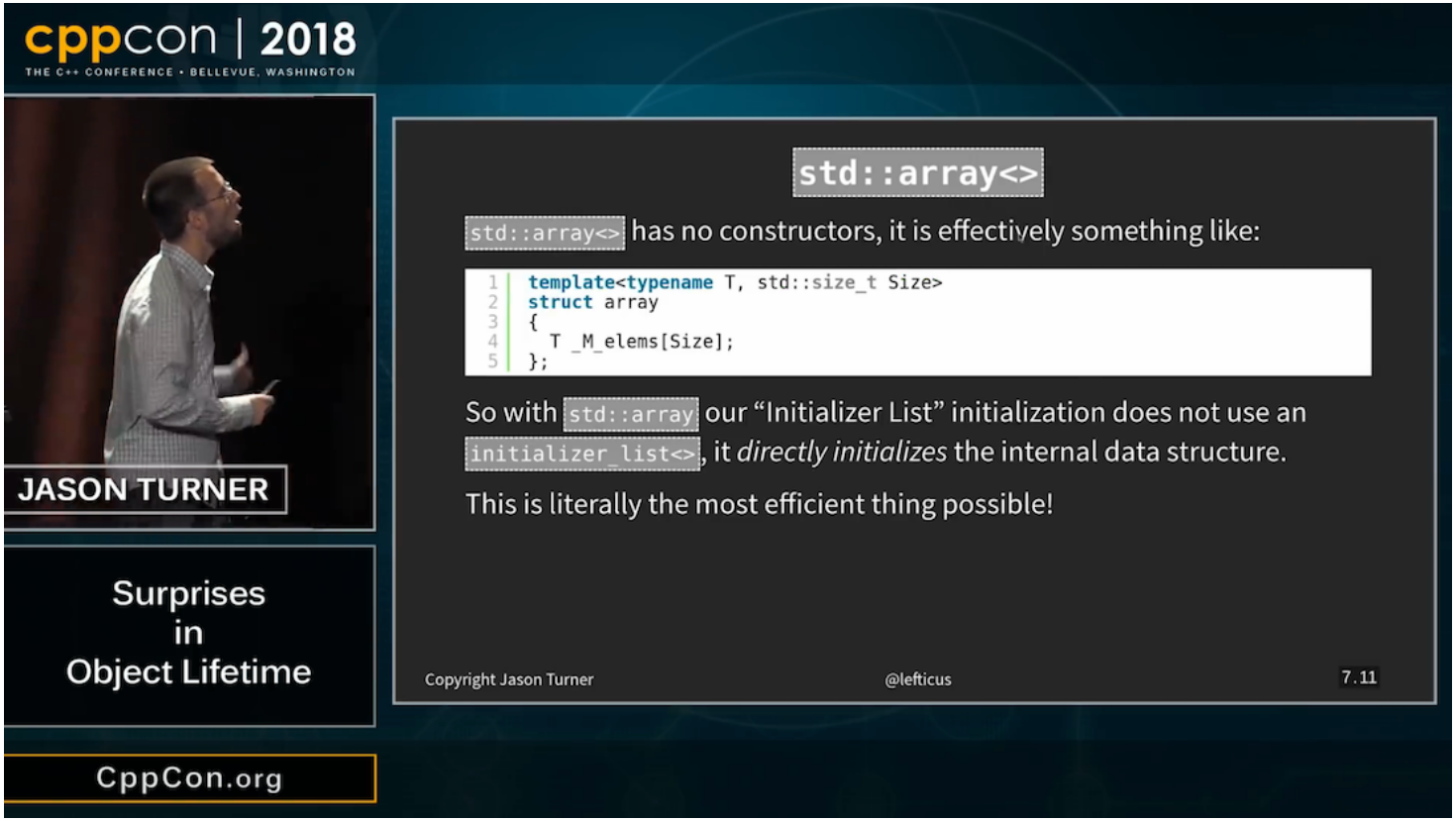
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



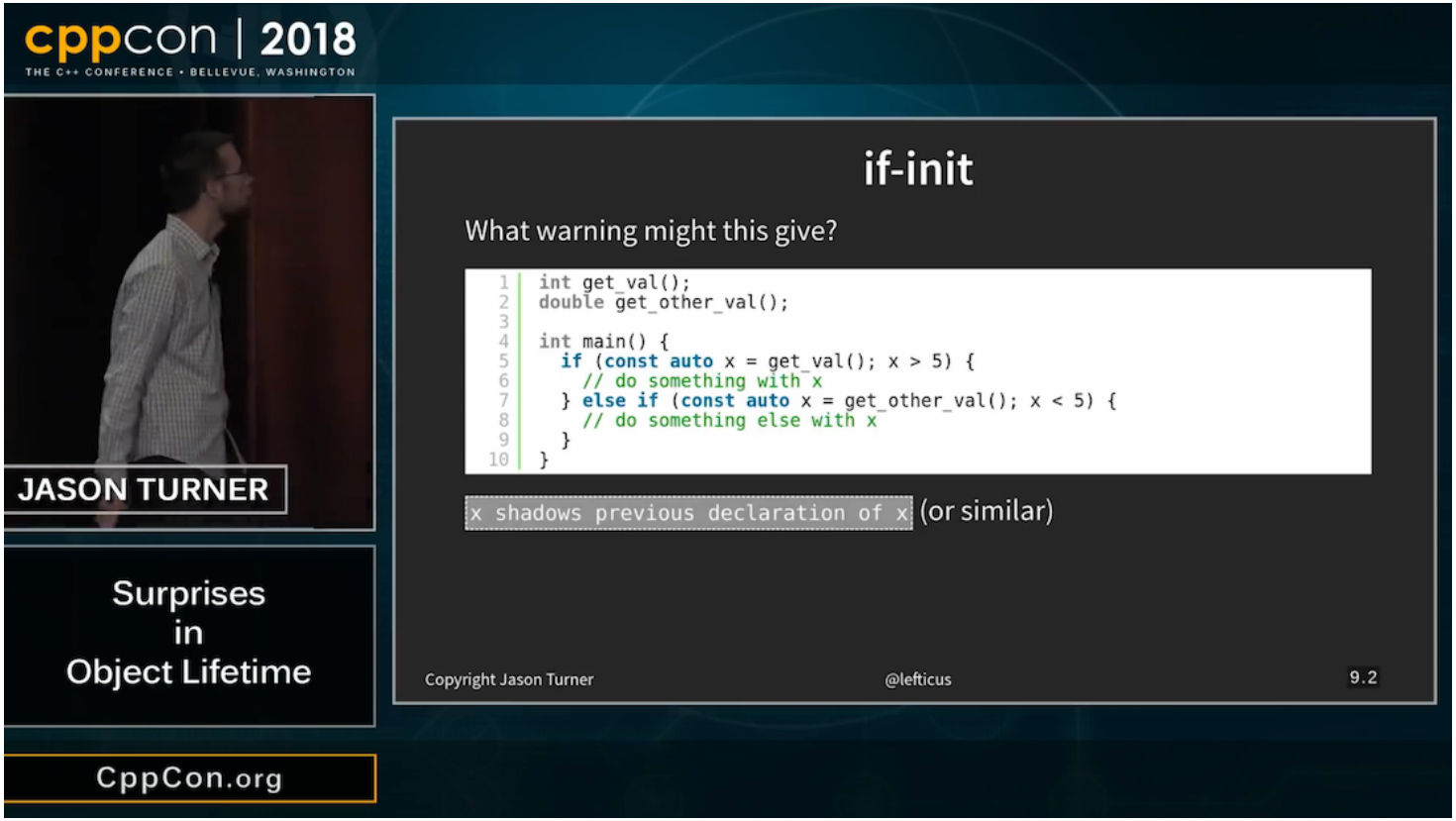
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



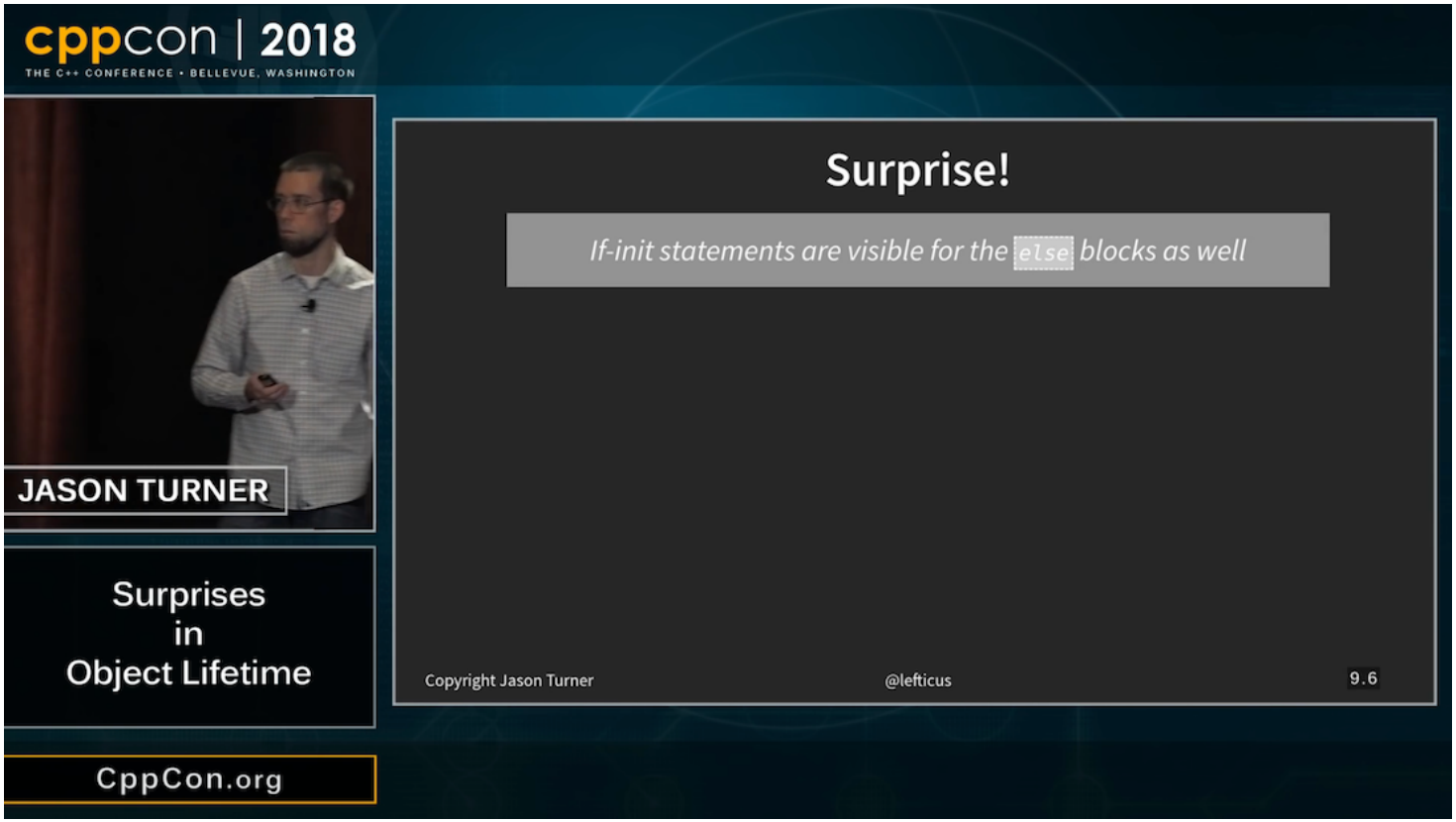
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



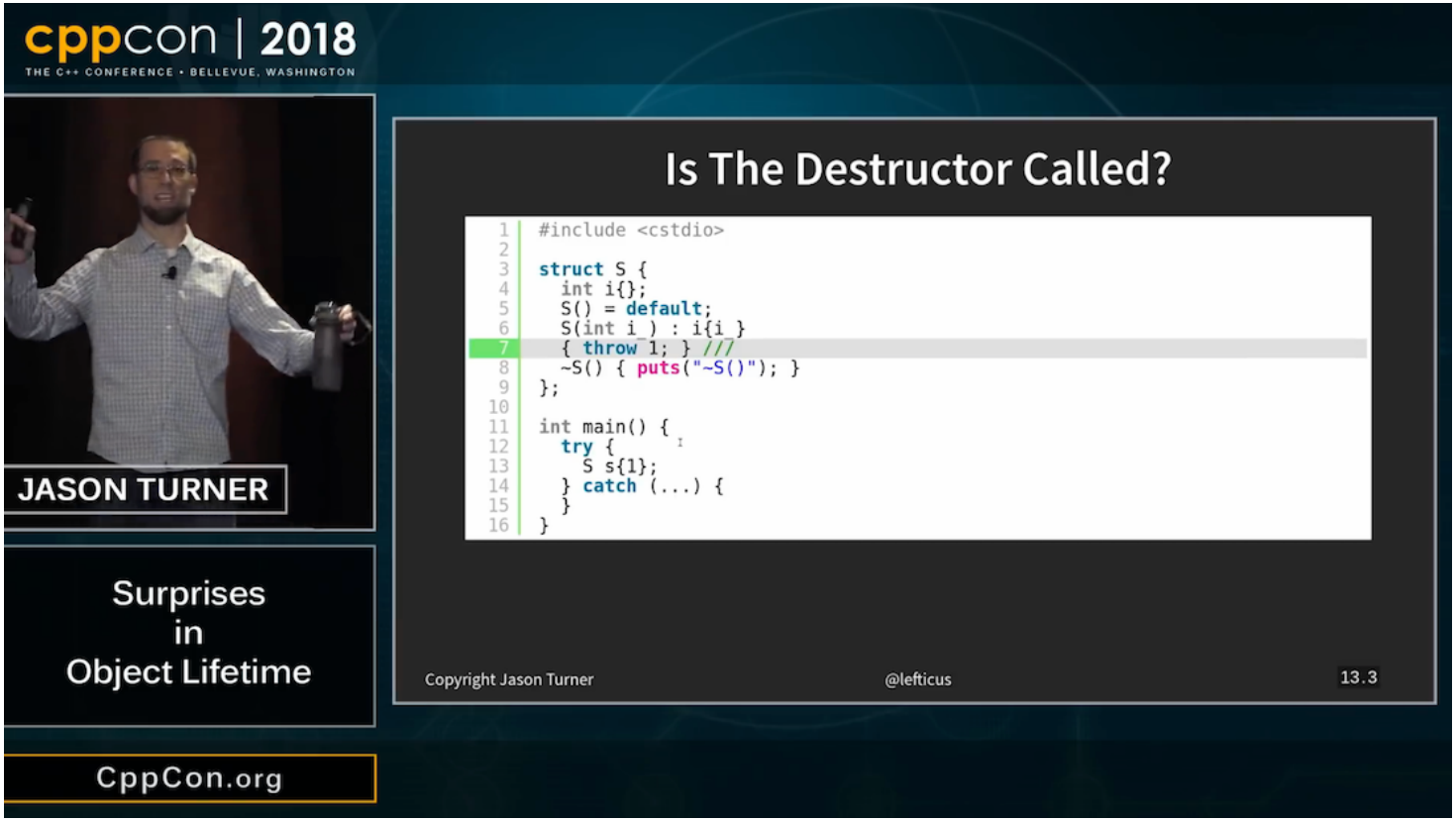
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



CppCon 2018: Jason Turner "Surprises in Object Lifetime"

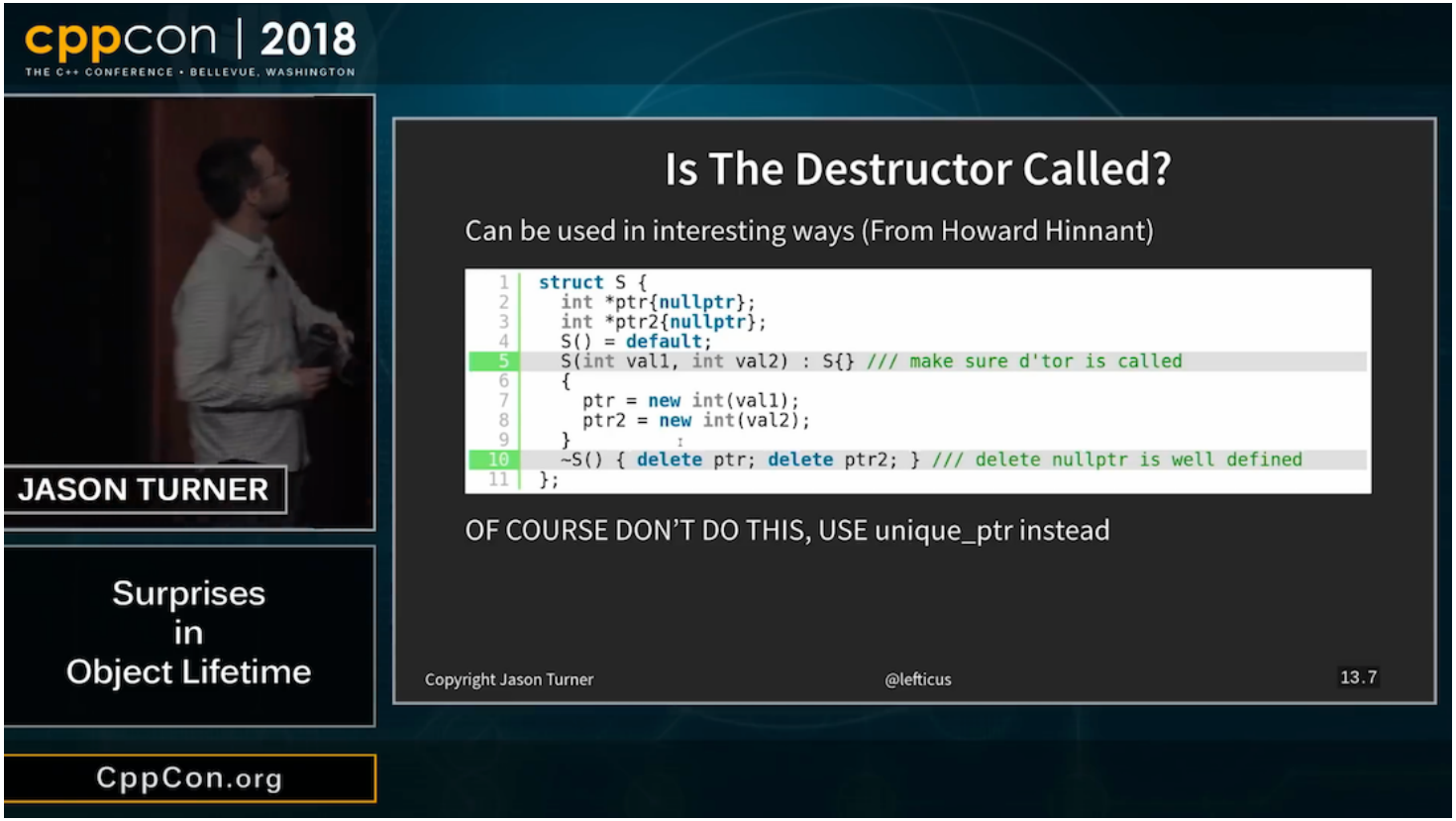


CppCon 2018: Jason Turner "Surprises in Object Lifetime"

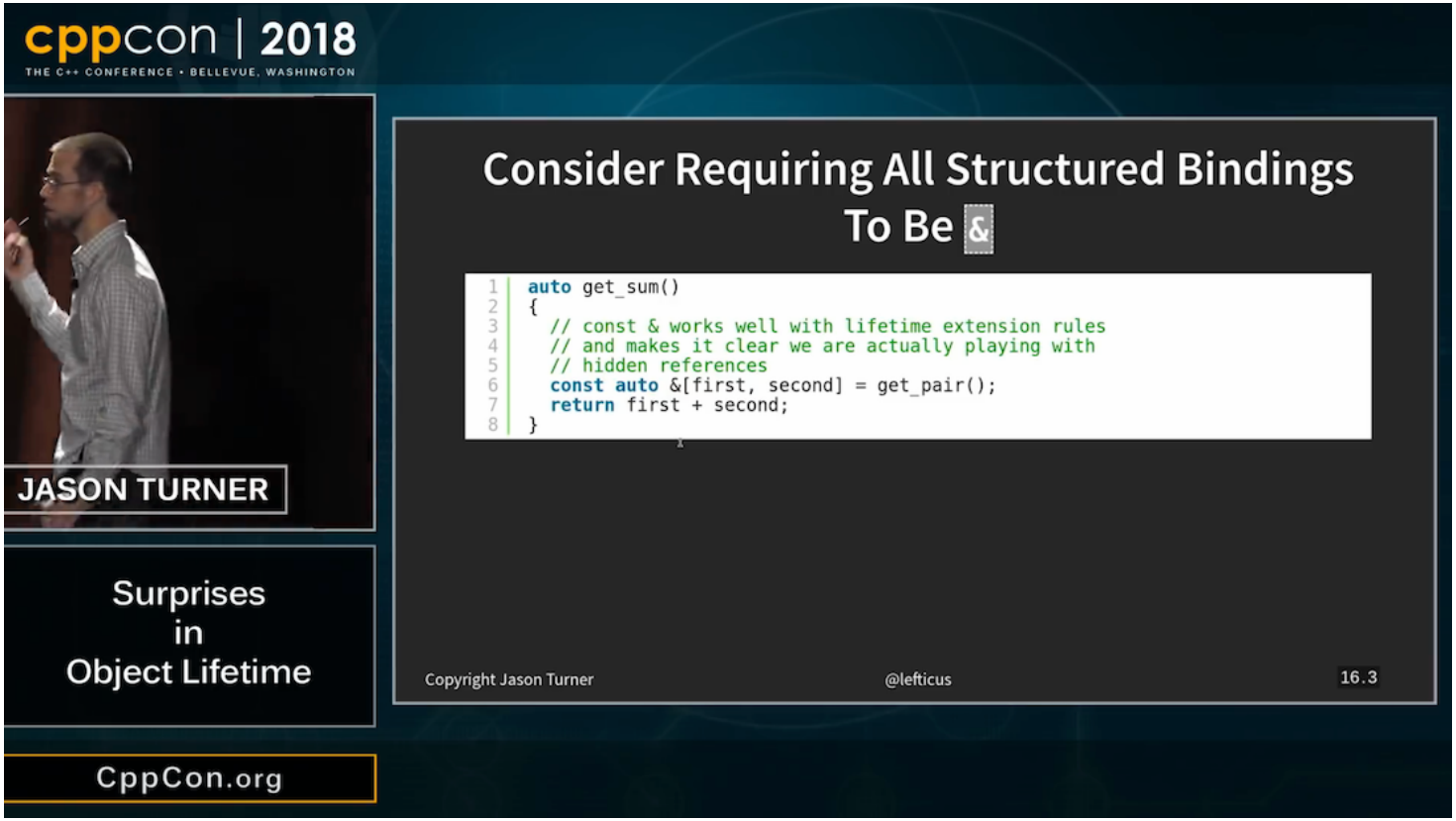


CppCon 2018: Jason Turner "Surprises in Object Lifetime"

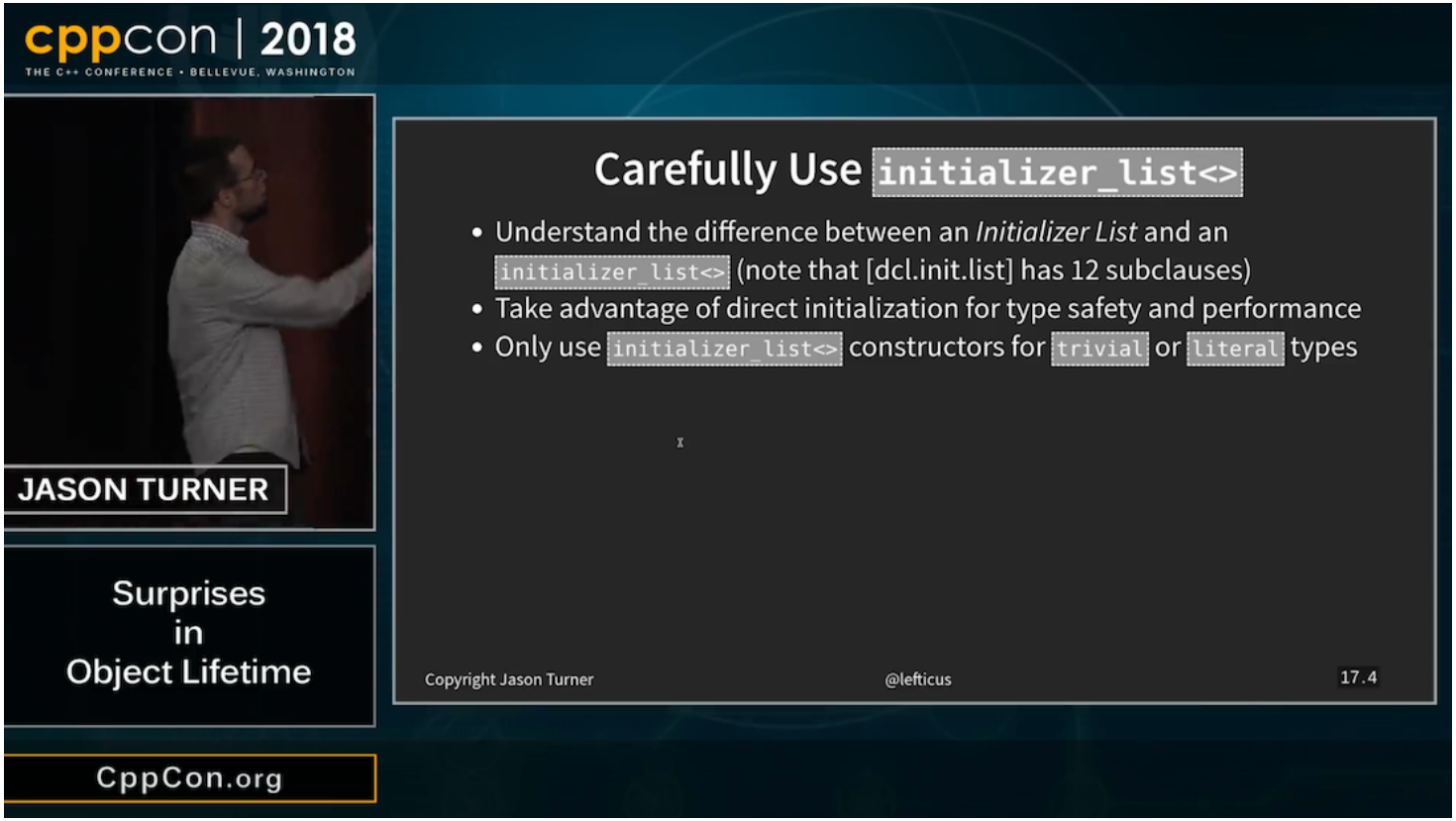
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



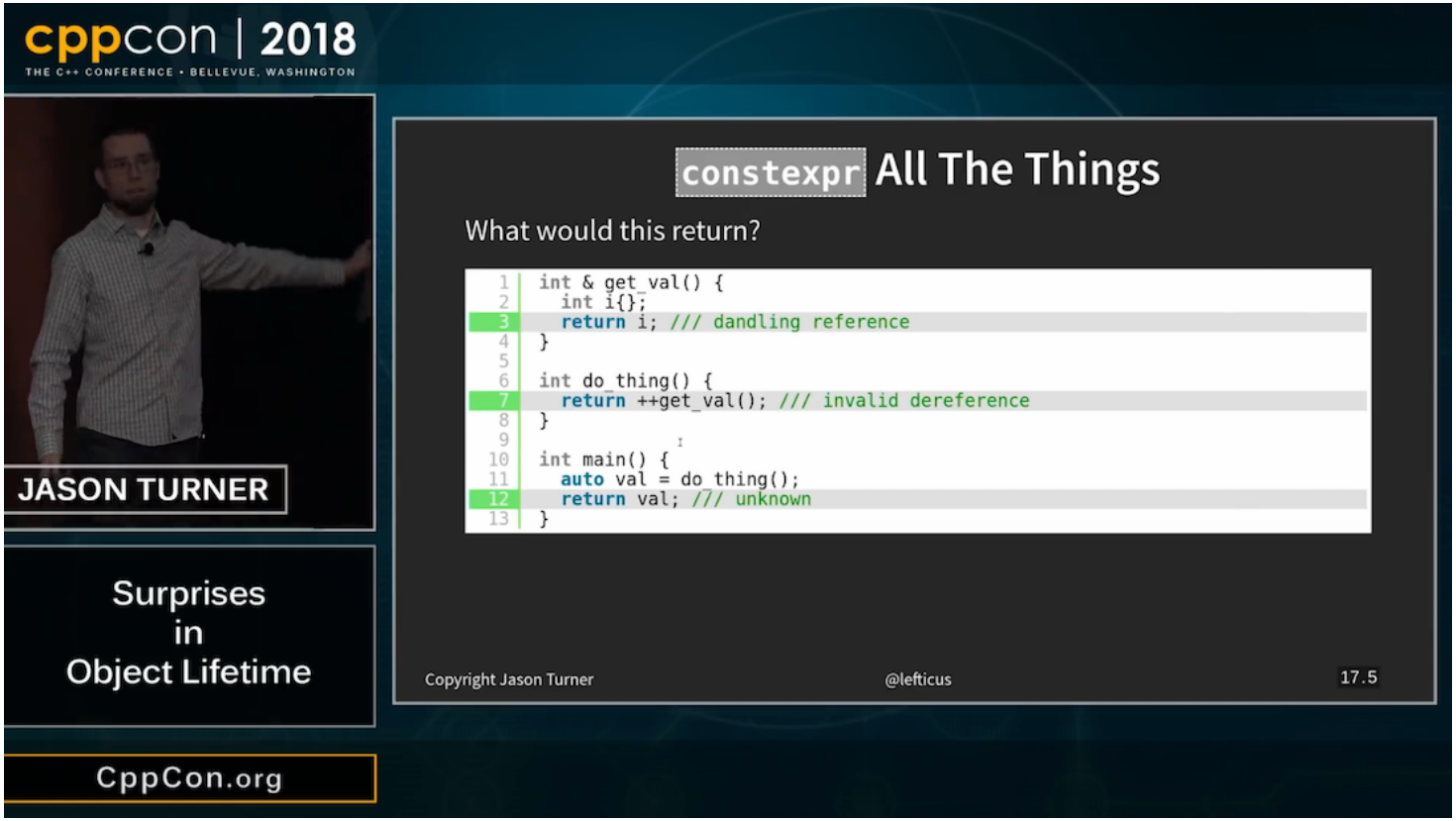
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



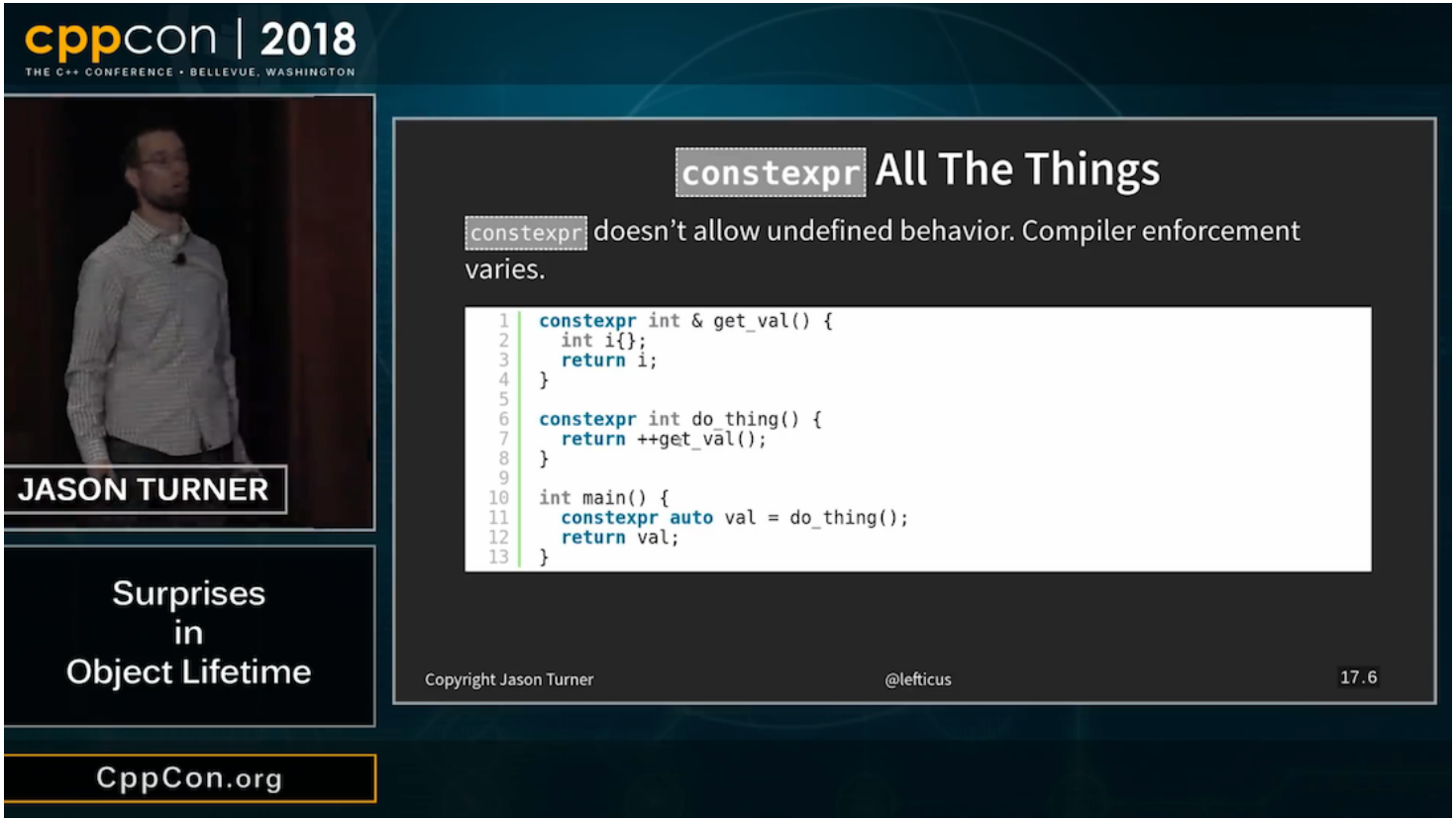
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



CppCon 2018: Jason Turner "Surprises in Object Lifetime"



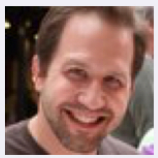
CppCon 2018: Jason Turner "Surprises in Object Lifetime"



Arthur O'Dwyer: Feature that Always Works

- for (auto&& elt : range) Always Works
- for (auto&& elt : range) Still Always Works

```
for (auto&& elt : range) {
    do_something_with(elt);
}
```



Scott Hanselman
@shanselman

2 Replies, 42 Quotes

If you're starting a sentence with "Why don't you just..." then it's very likely you don't understand the complexity of the problem.

26/09/2017, 18:25 (Yesterday)
Twitter Web Client

850 Likes | 403 Retweets | Thread >

Ahmed Khalaf @ahmedkhalaf_92

And “WHYNOTJUST.md” for justifying unintuitive solutions that have intuitive alternatives that don’t work/impractical.

> 
>
> **Joe Groff** @jckarter
>
> Ideas often get copied from well-known established projects into new projects with the background behind those decisions lost. Projects should have a "MISTAKES.md" to lay out regretful design choices that can't be reversed for time, compatibility, etc. reasons
>
> 13h • 14/01/2019 • 20:36

13h • 14/01/2019 • 20:46