C++ Club UK

Gleb Dolgich

2019-10-10

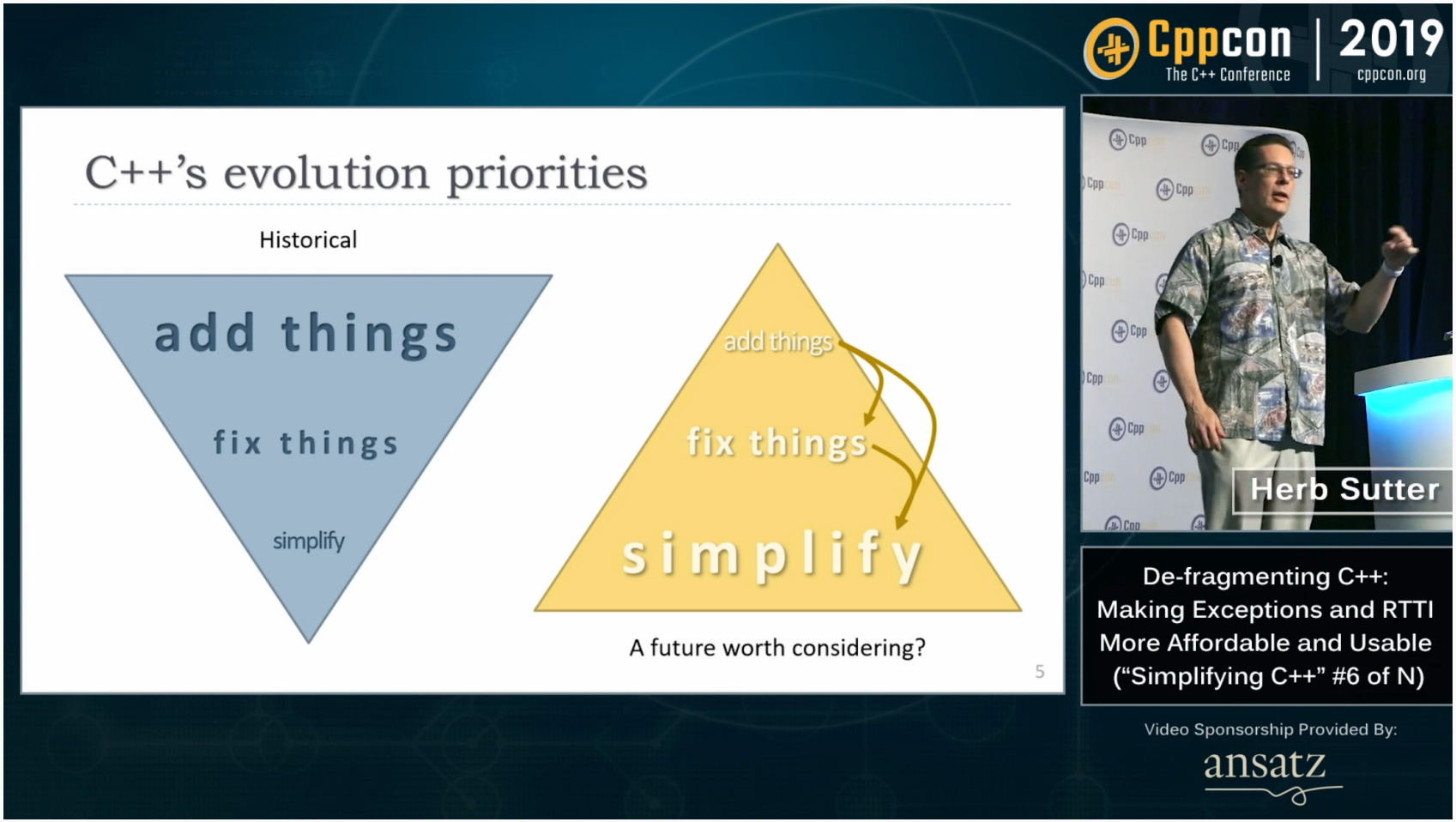
CppCon 2019 Trip Reports

- Matt Godbolt

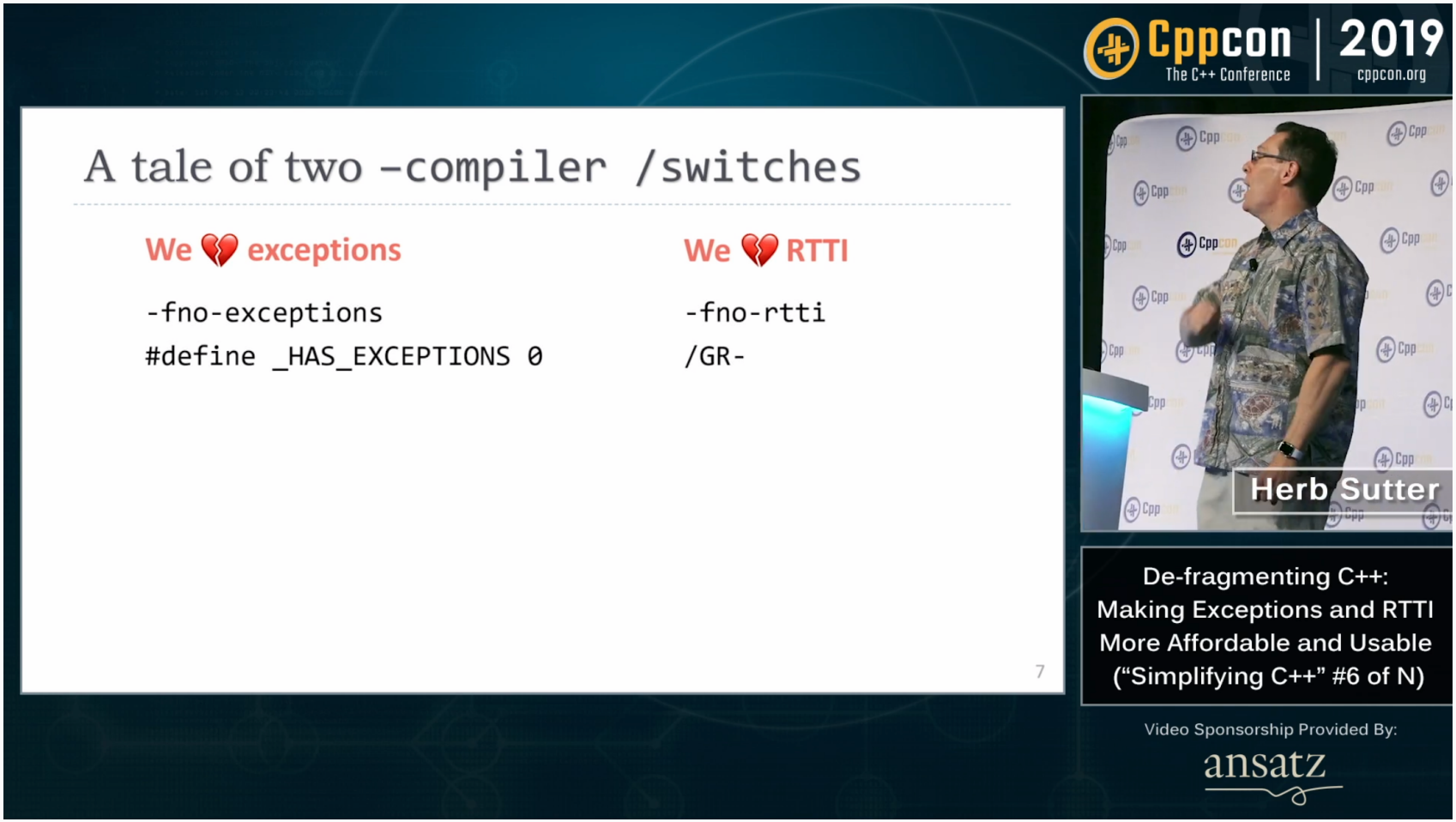
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”

- Video: https://youtu.be/ARYP83yNAWk
  - Reddit: https://www.reddit.com/r/cpp/comments/d87plg/cppcon_2019_herb_sutter_defragmenting_c_making/

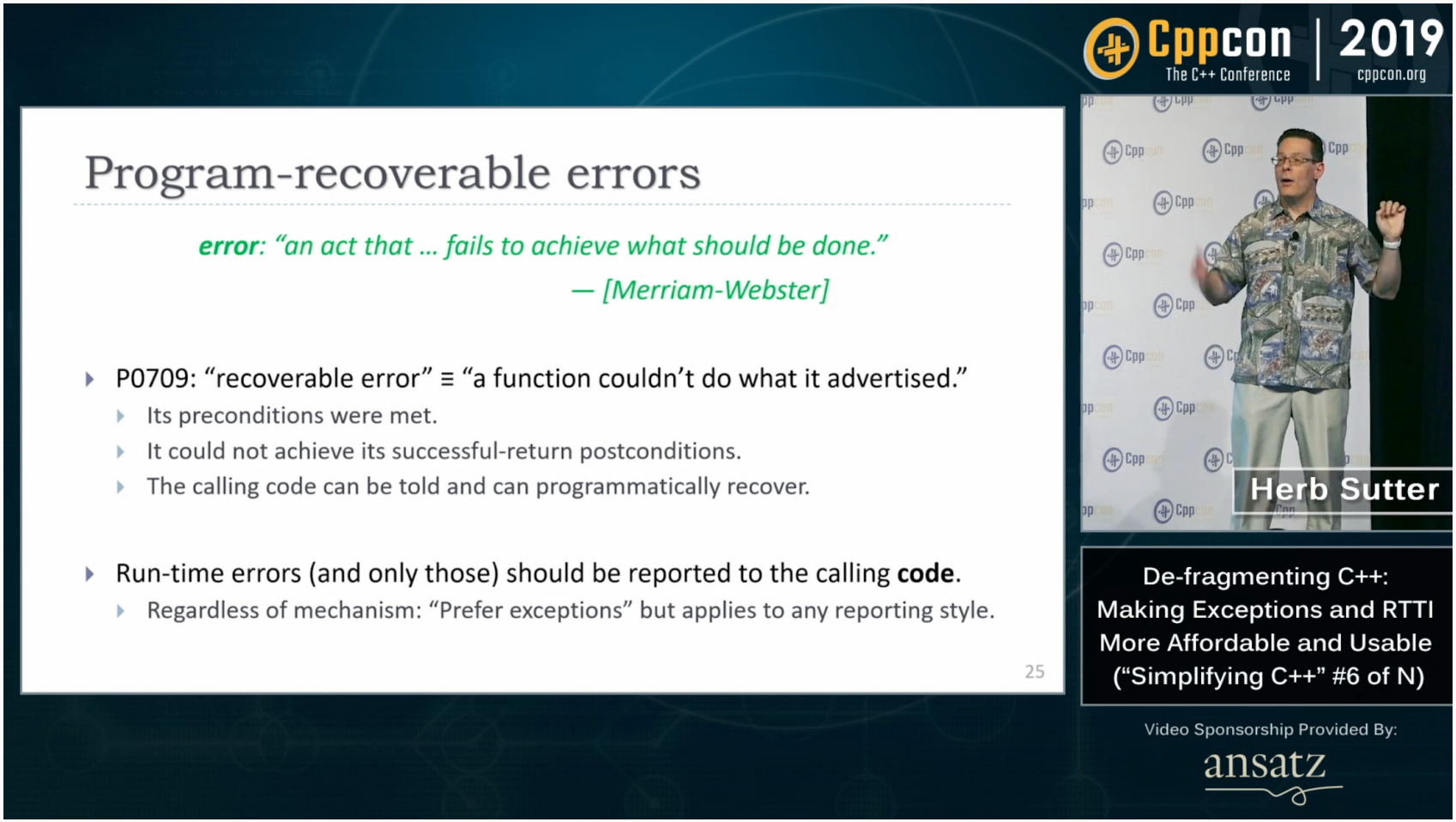
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



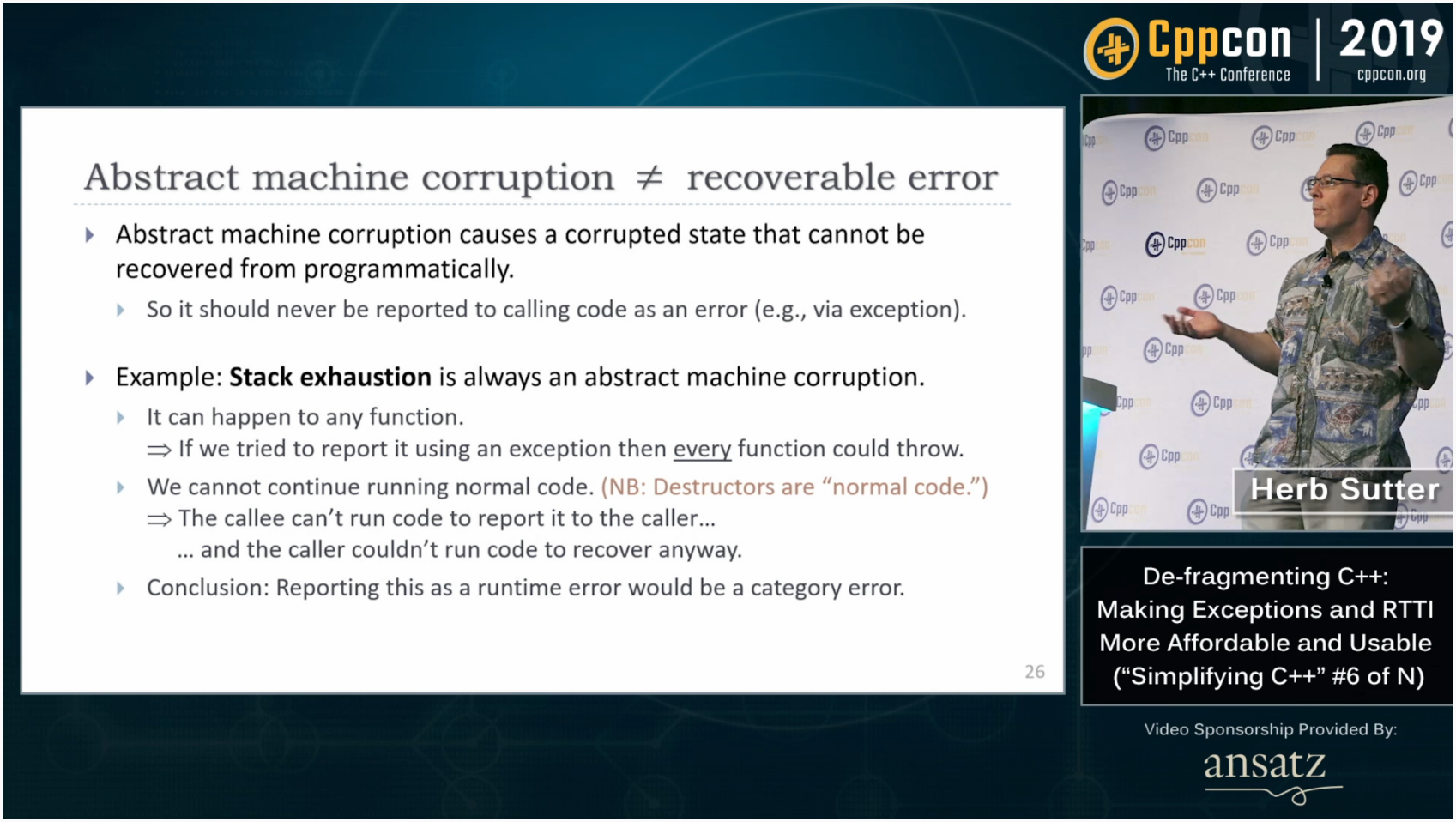
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



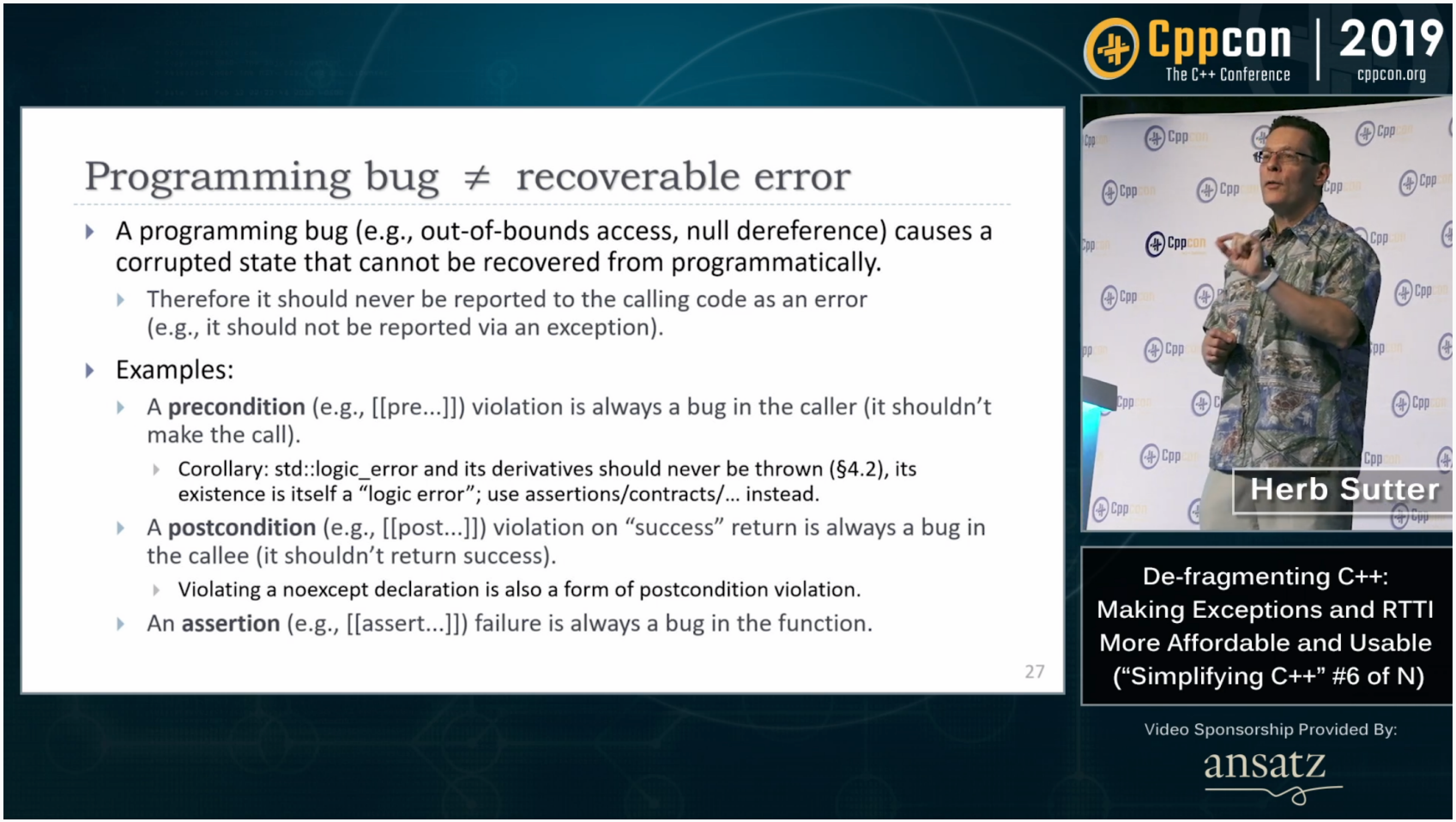
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



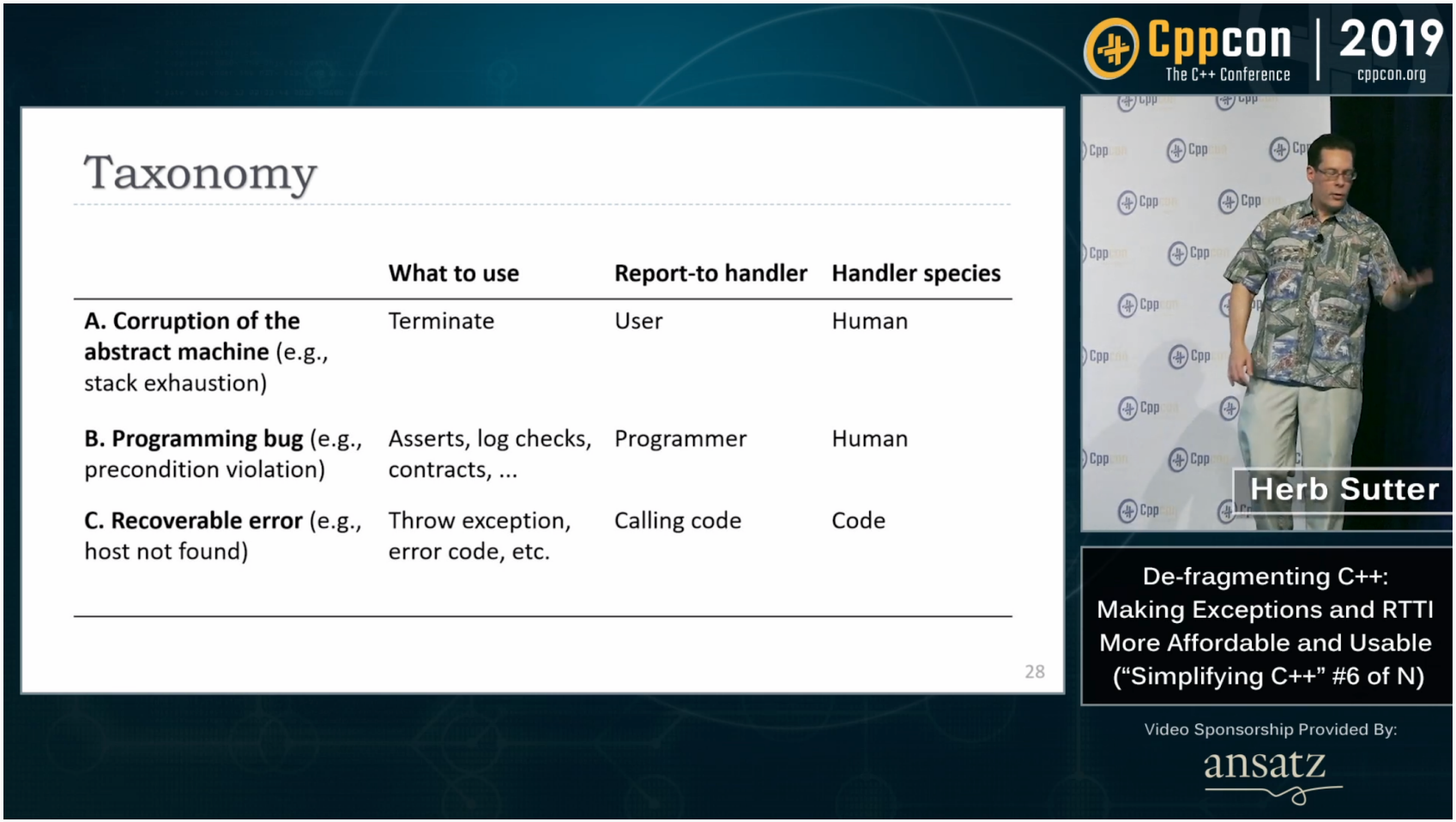
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”

Part 1: Exception Handling (EH)

- Establishing the problem: Today’s EH violates the zero-overhead principle
  - “I can’t afford to enable exception handling” ⇒ paying for what you don’t use
  - “I can’t afford to throw an exception” ⇒ can write it more efficiently by hand
  - Bonus “I can’t throw through this code” ⇒ lack of control, invisible vs. automatic propagation
- Key definition: What is a “recoverable error”?
  - Recoverable error != programming bug != abstract machine corruption
  - Exceptions/codes != pre/post contracts != stack and heap overflow
- Four coordinated proposals
  1. Enable zero-overhead exception handling
  2&3. Throw fewer exceptions (~90% of all exceptions should not be)
  4. Support explicit “try” for visible propagation

29

Cppcon The C++ Conference | 2019 cppcon.org

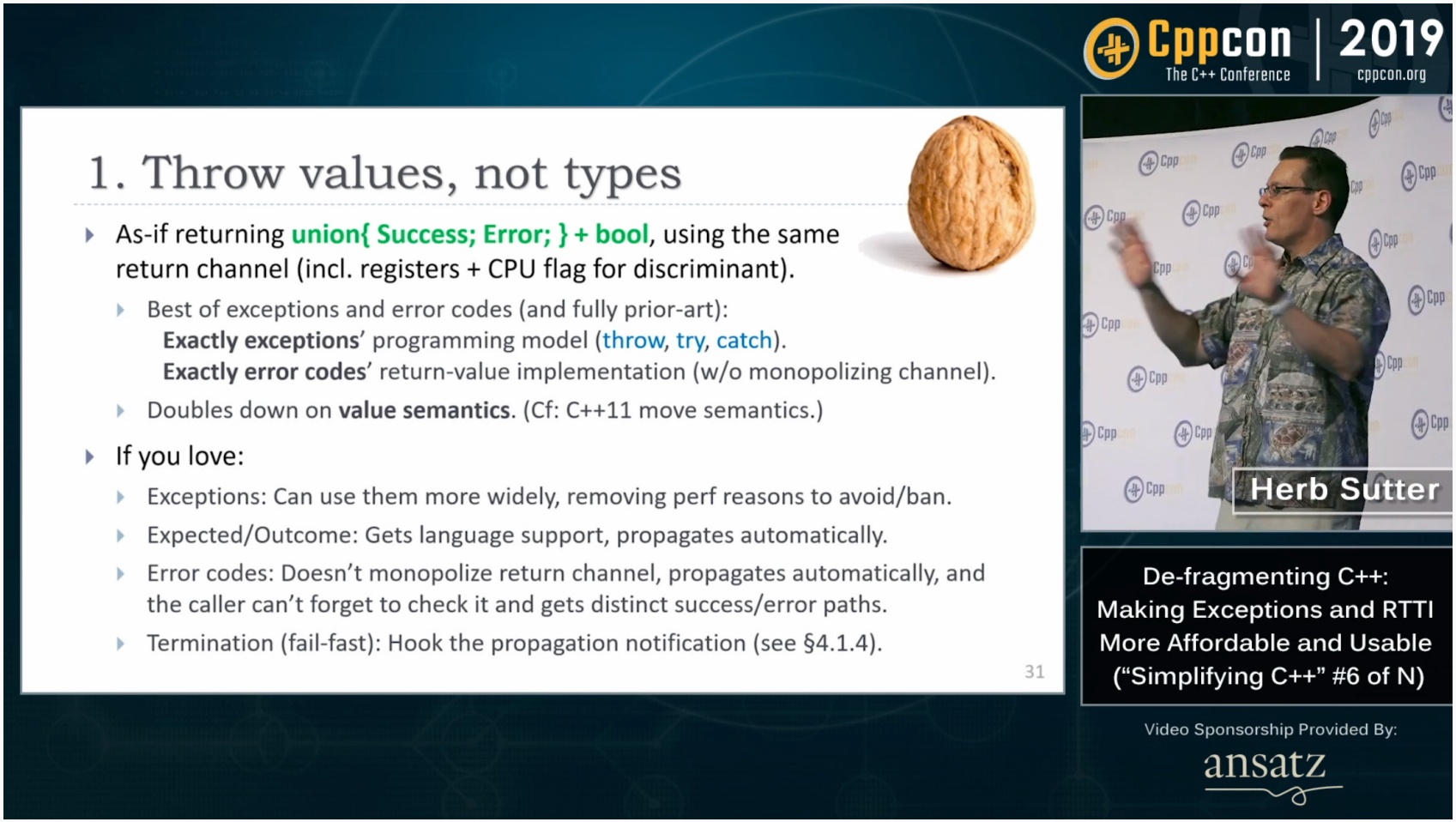
De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable (“Simplifying C++” #6 of N)

Video Sponsorship Provided By:

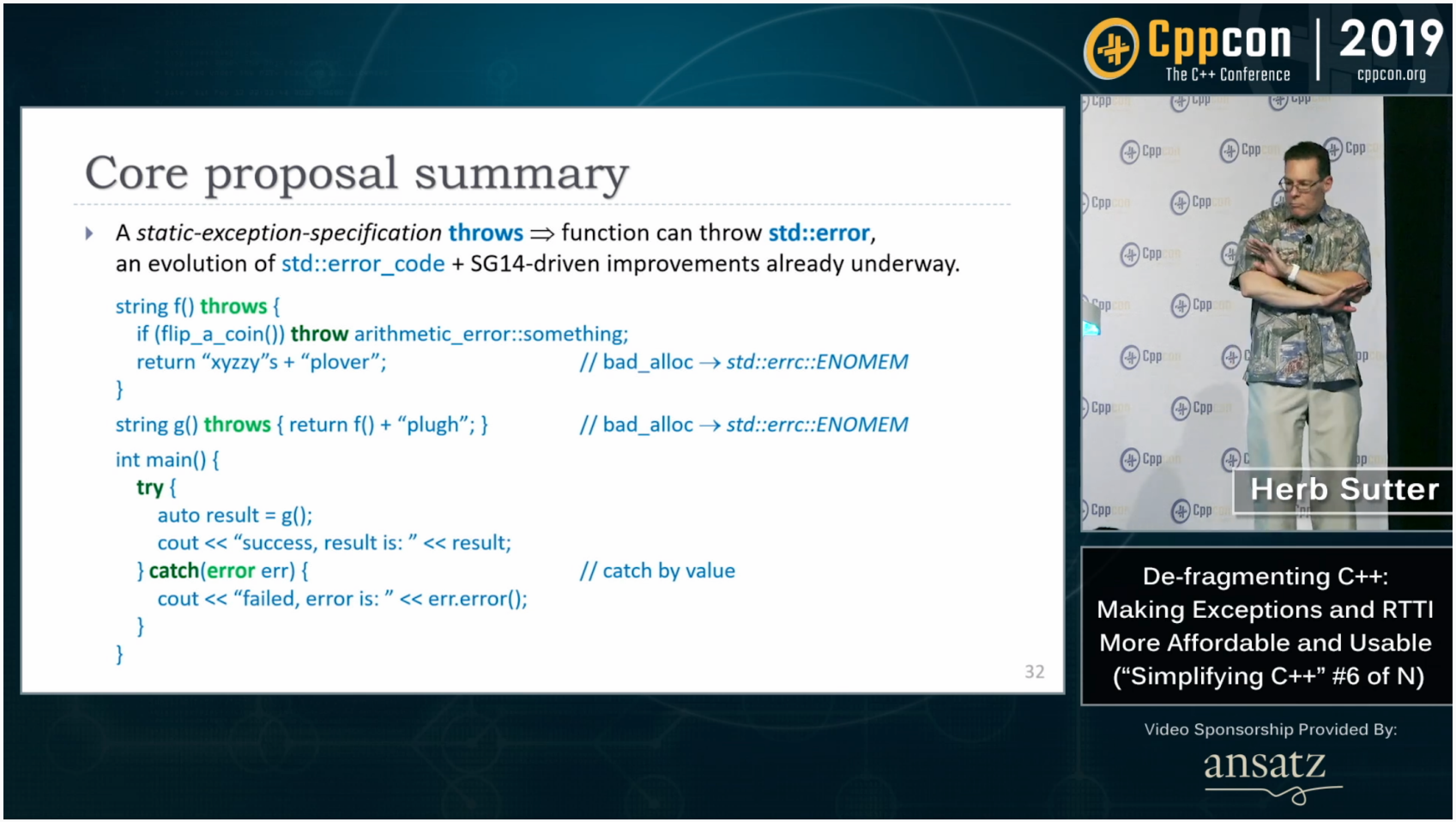
ansatz

CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”

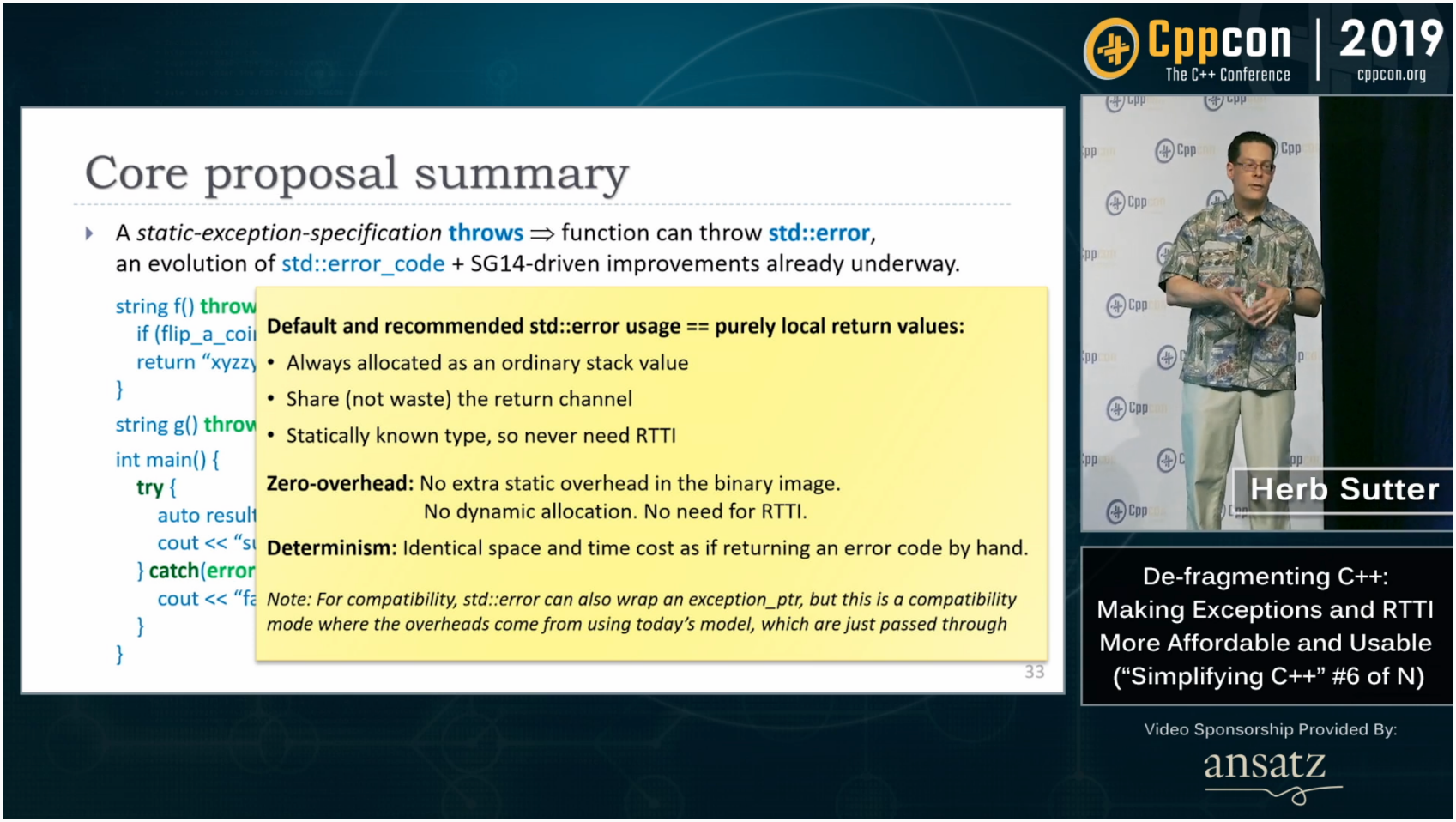
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



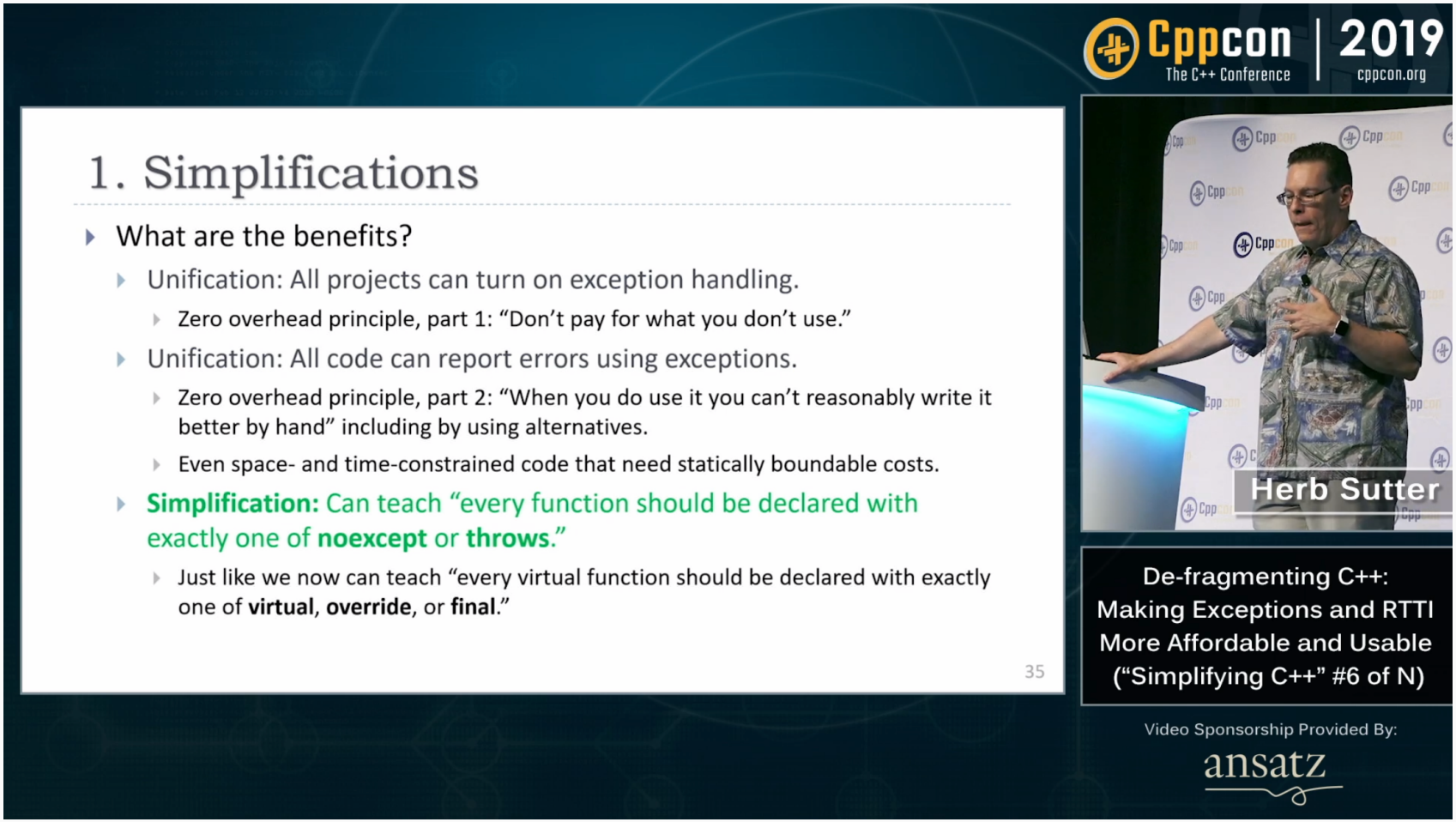
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



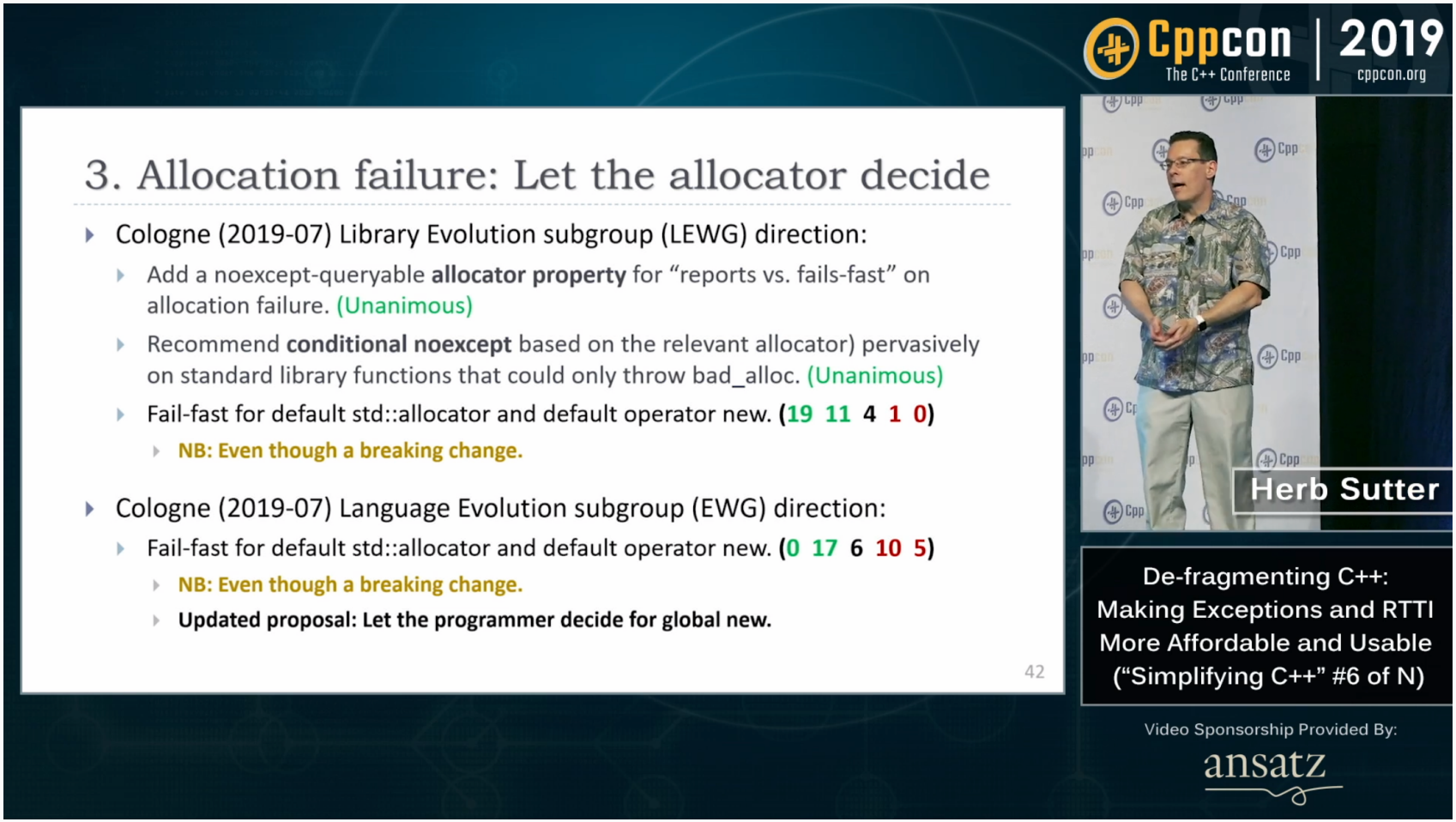
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



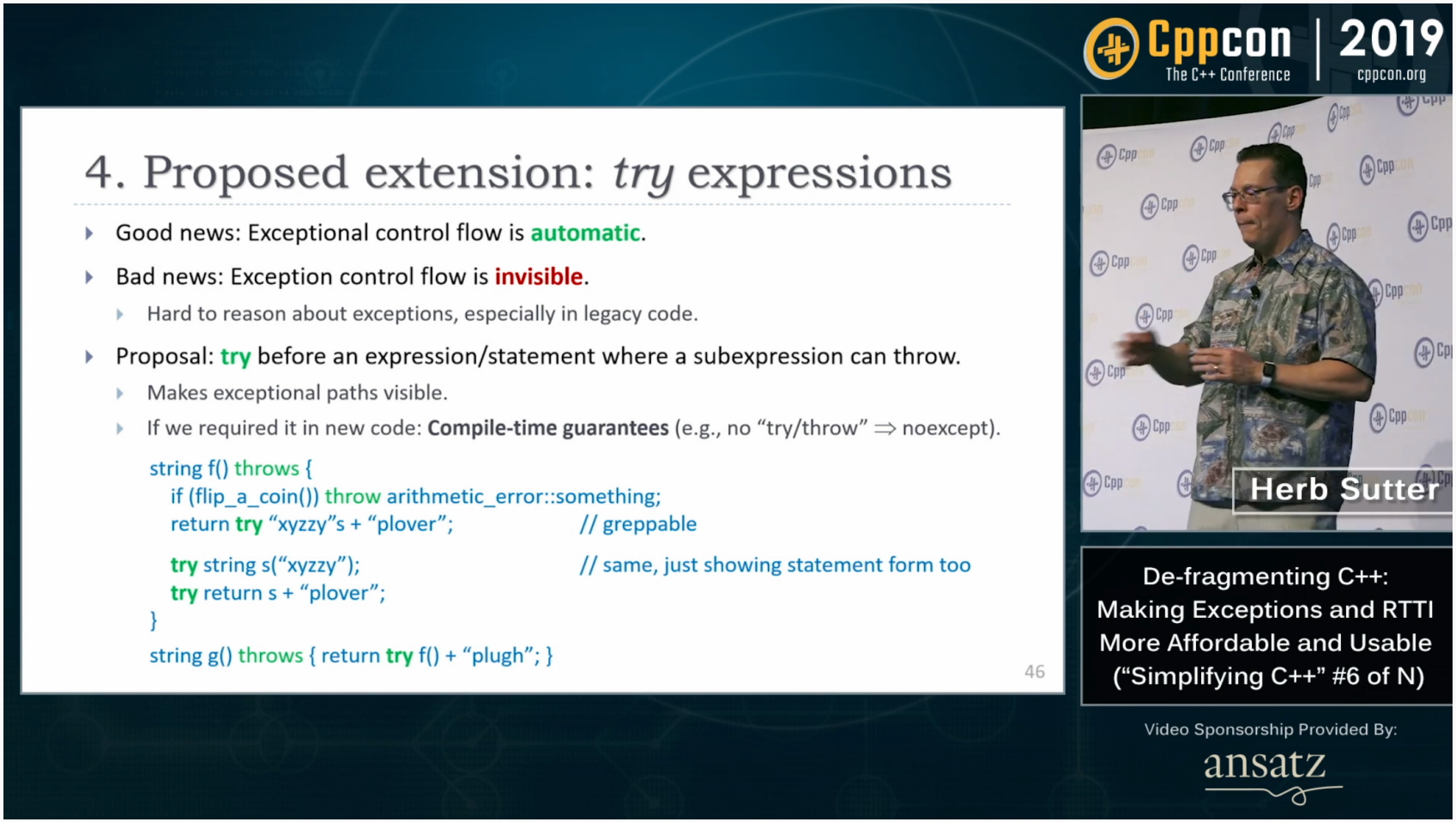
CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”



Arthur O'Dwyer "Thoughts on Herb Sutter's CppCon keynote"

https://quuxplusone.github.io/blog/2019/09/24/thoughts-on-eh/

Reddit: https://www.reddit.com/r/cpp/comments/d8p08f/thoughts_on_herb_sutters_cppcon_keynote/

Tony Van Eerd @tvaneerd

```
try {
  return vec.at(index);
}
catch (std::out_of_range const &) {
  return -1;
}
```

I just saw this in a code review and was going to claim it inefficient, but darn @CompileExplore proved me wrong again. (IIRAC*) All compilers turn it into an if check on the bounds.

*Read ASM

11h • 11/07/2019 • 22:56

Cologne Trip Report

https://www.silexica.com/news/iso_cpp_meeting_2019/

- P1413, Deprecate std::aligned_storage and std::aligned_union
- P1609, C++ Should Support Just-in-Time Compilation
- P1727, Issues with current flat_map proposal
- P0443r10, A Unified Executors Proposal for C++
- P1660r0, A Compromise Executor Design Sketch
- “C++ Tooling Ecosystem” Technical Report
- P1679r0, String Contains function

Quote

Fred Brooks:

> *Much of the essence of building a program is in fact the debugging of the specification.*