C++ Club Meeting Notes

Gleb Dolgich

2018-11-22

San Diego trip reports

- Ben Craig
- Corentin Jabot
  - Reddit
- René Rivera
- CppCast with Ashley Hedberg

CppCon 2018 Trip Reports

- JeanHeyd Meneide

Better template support and error detection in C++ Modules with MSVC 2017 version 15.9

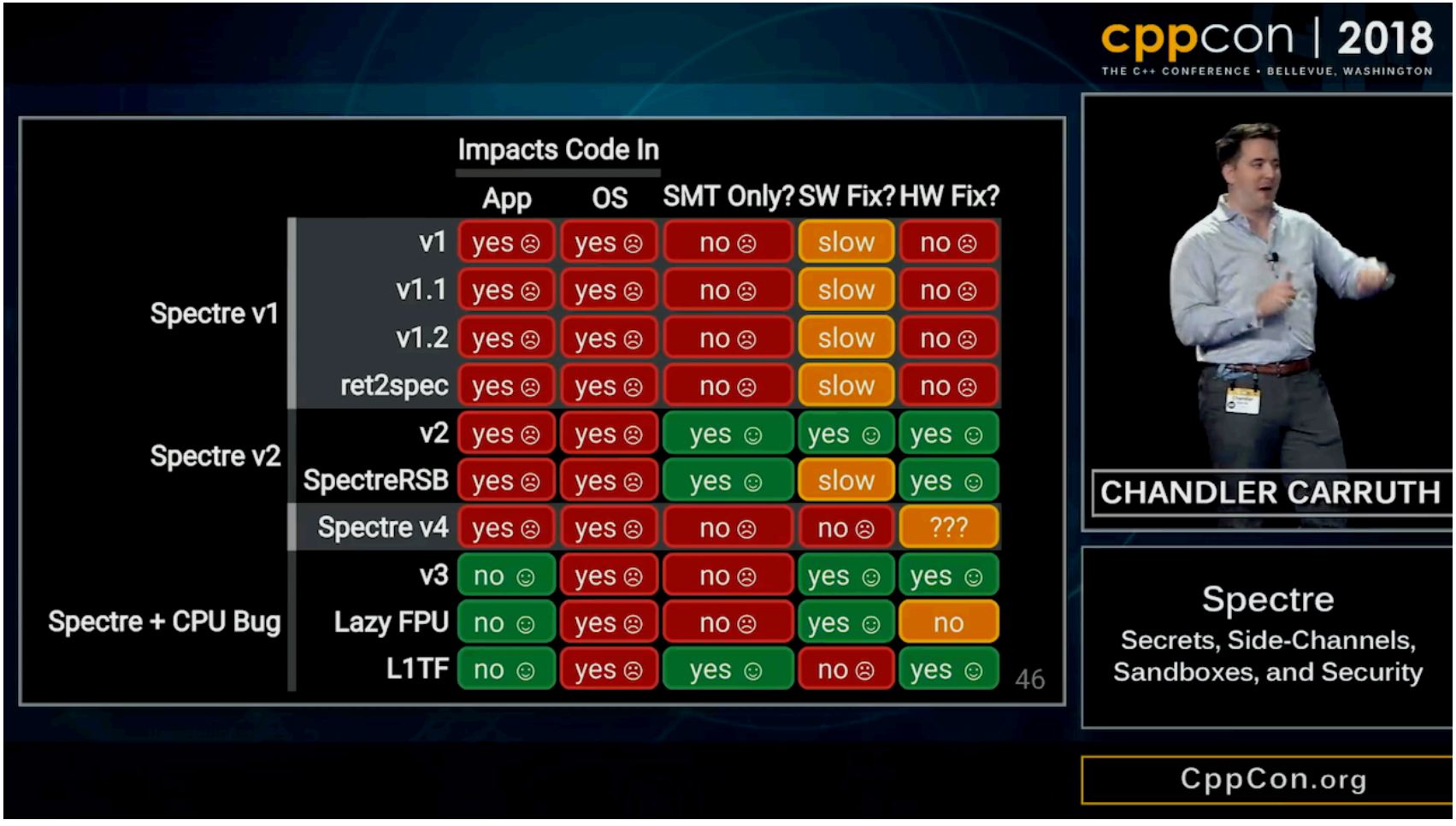
VC Blog

C++17

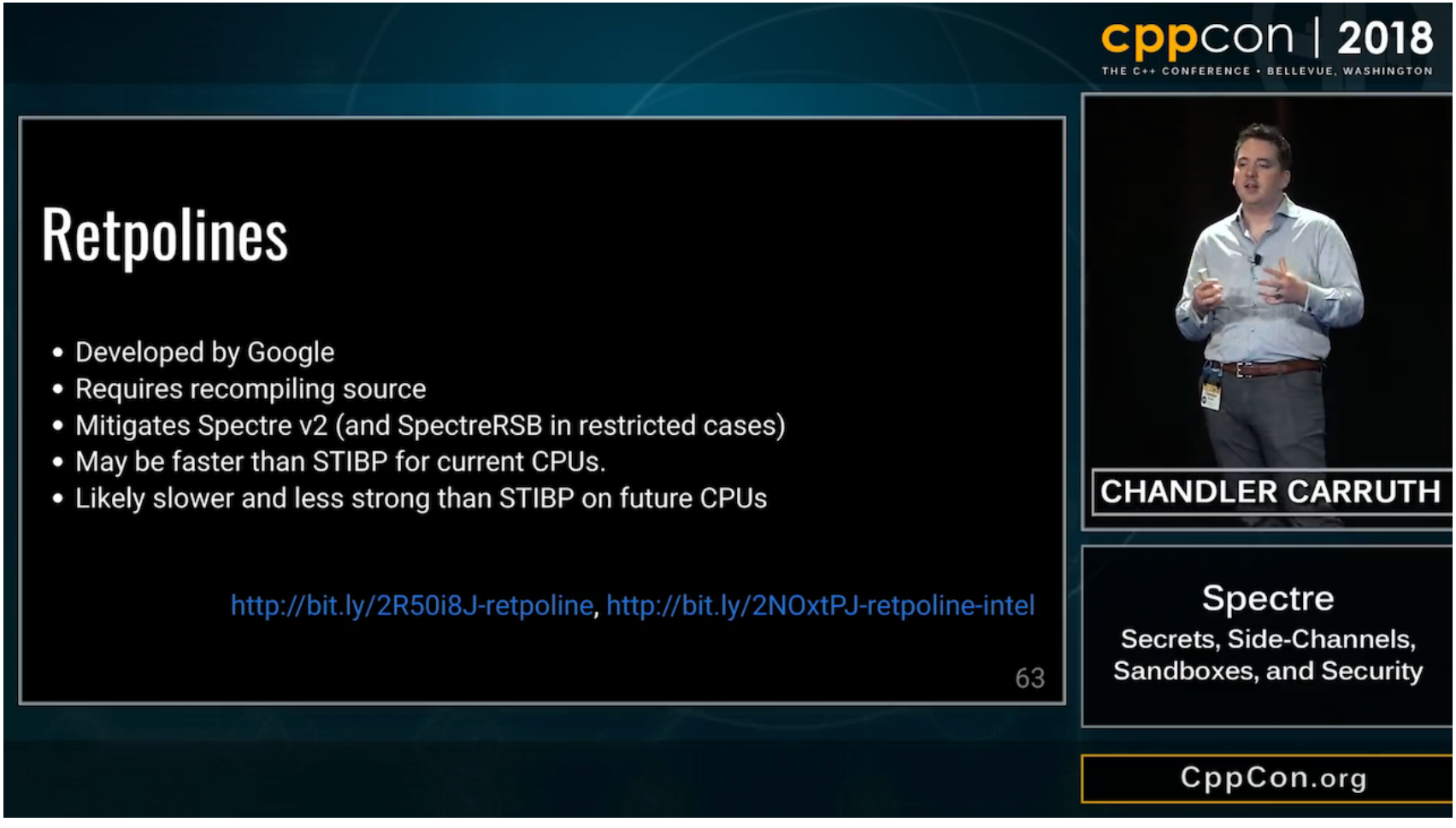
- Viva64: C++17 Refresher
- VCBlog: Standard Library Algorithms: Changes and Additions in C++17

CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (1/9)

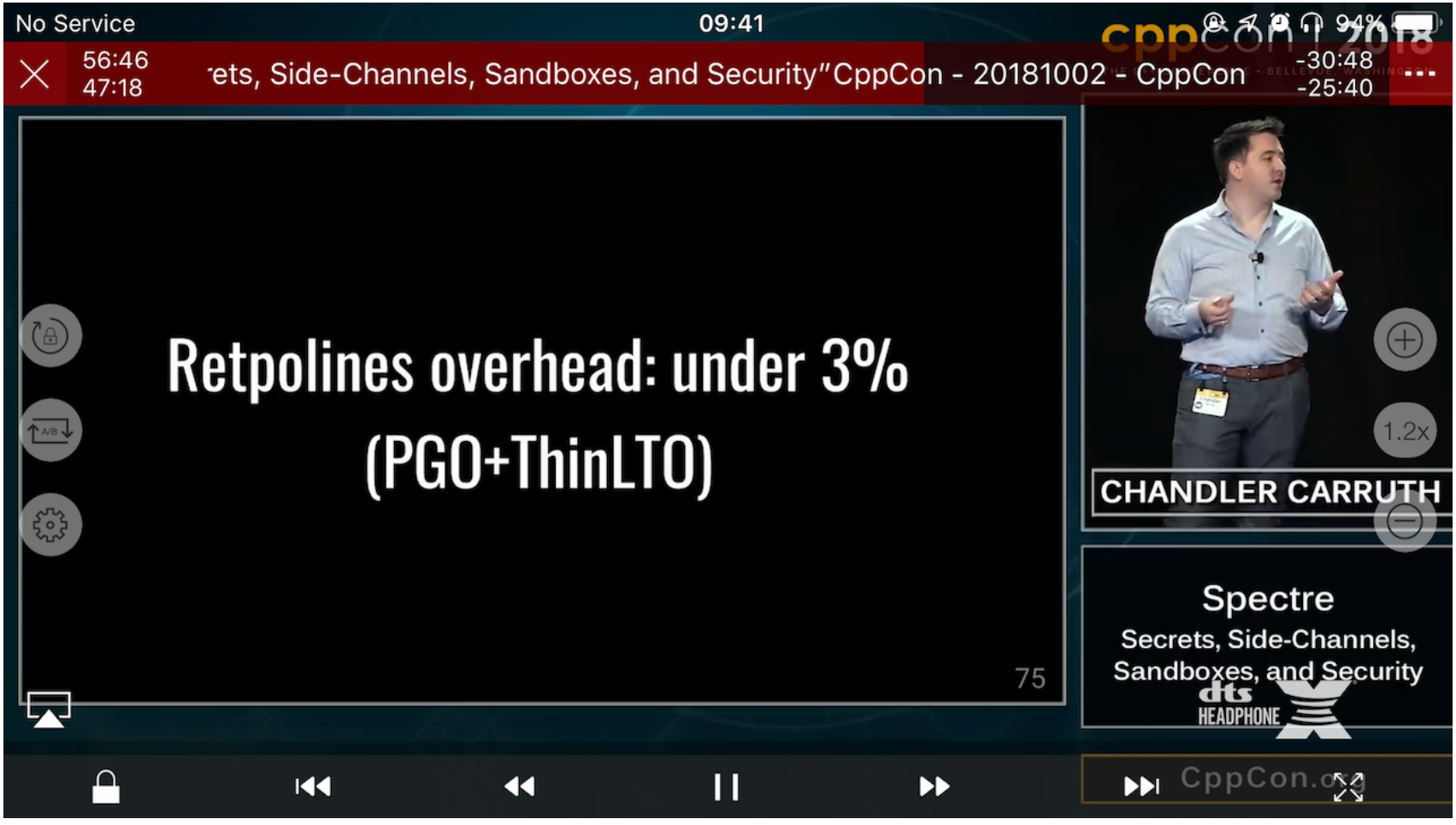
Video



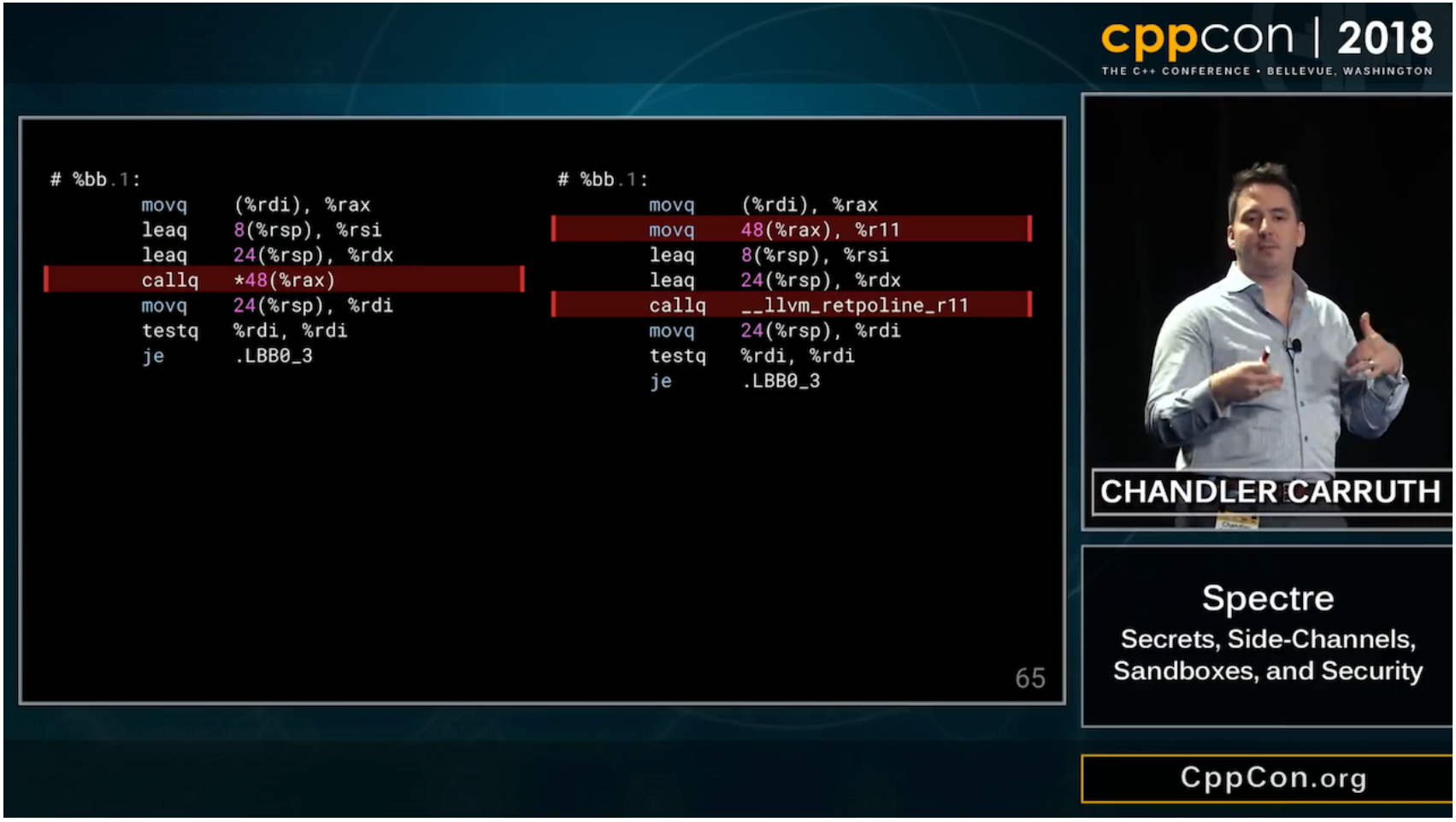
CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (2/9)



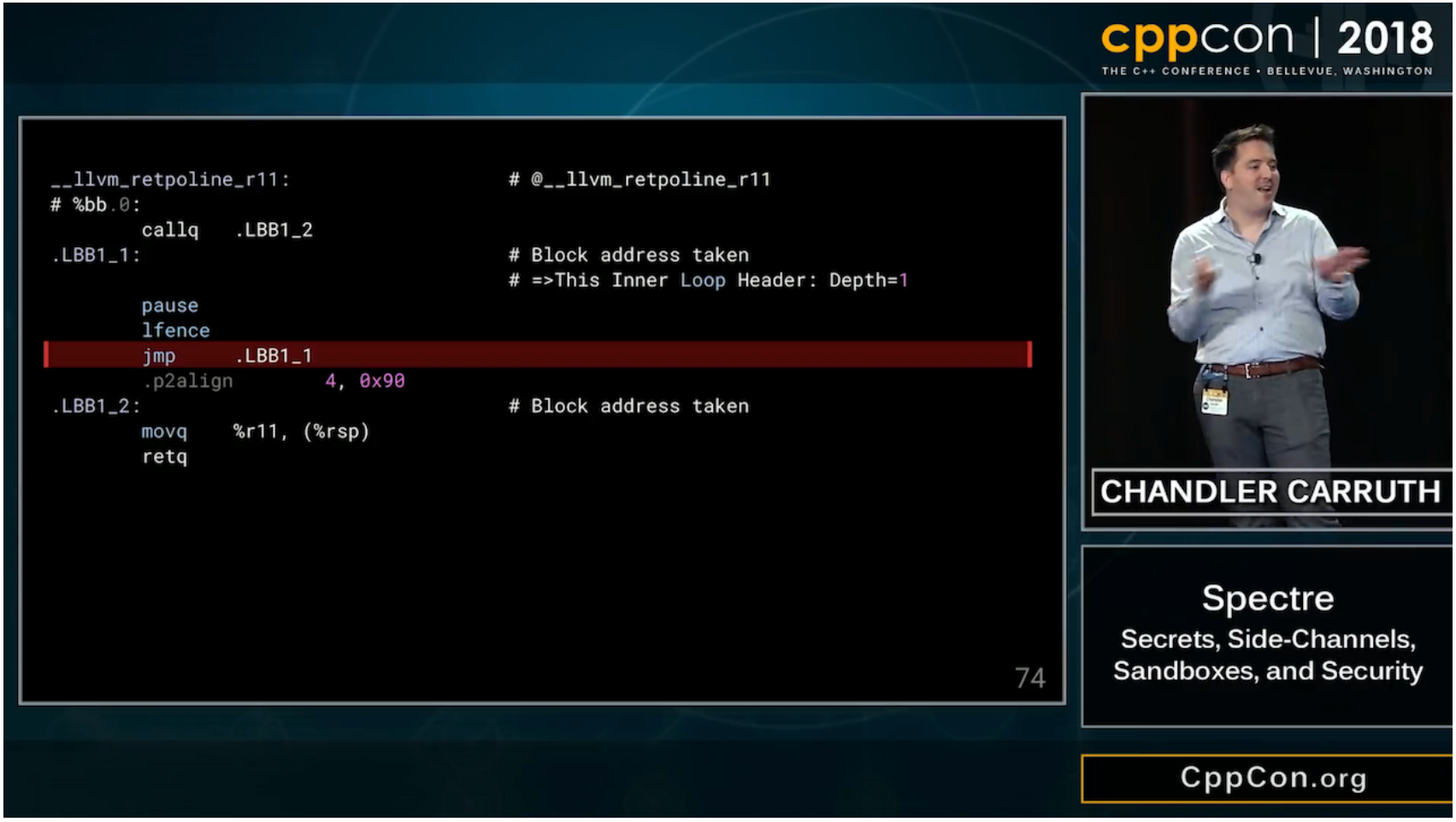
CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (3/9)



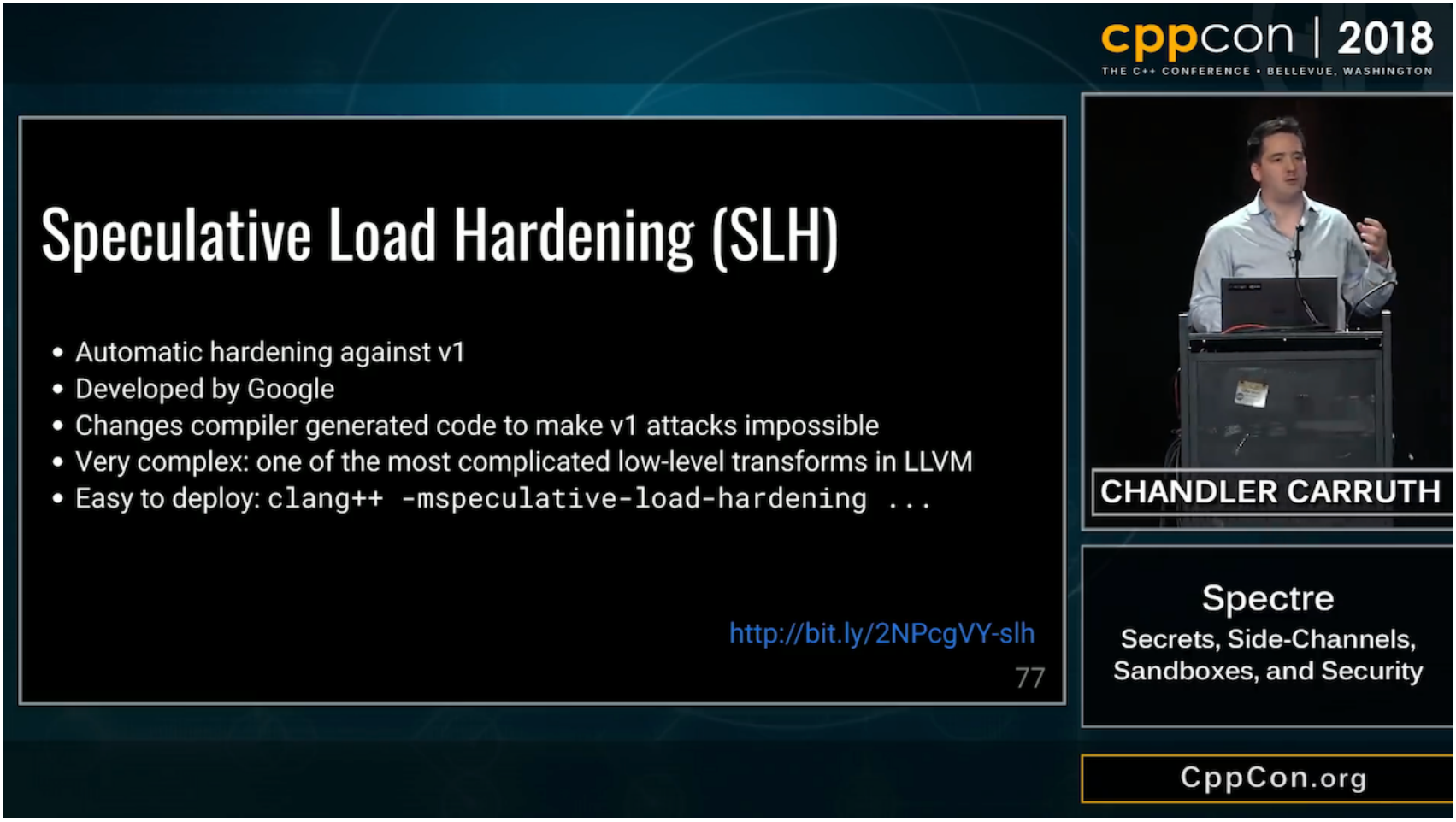
CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (4/9)



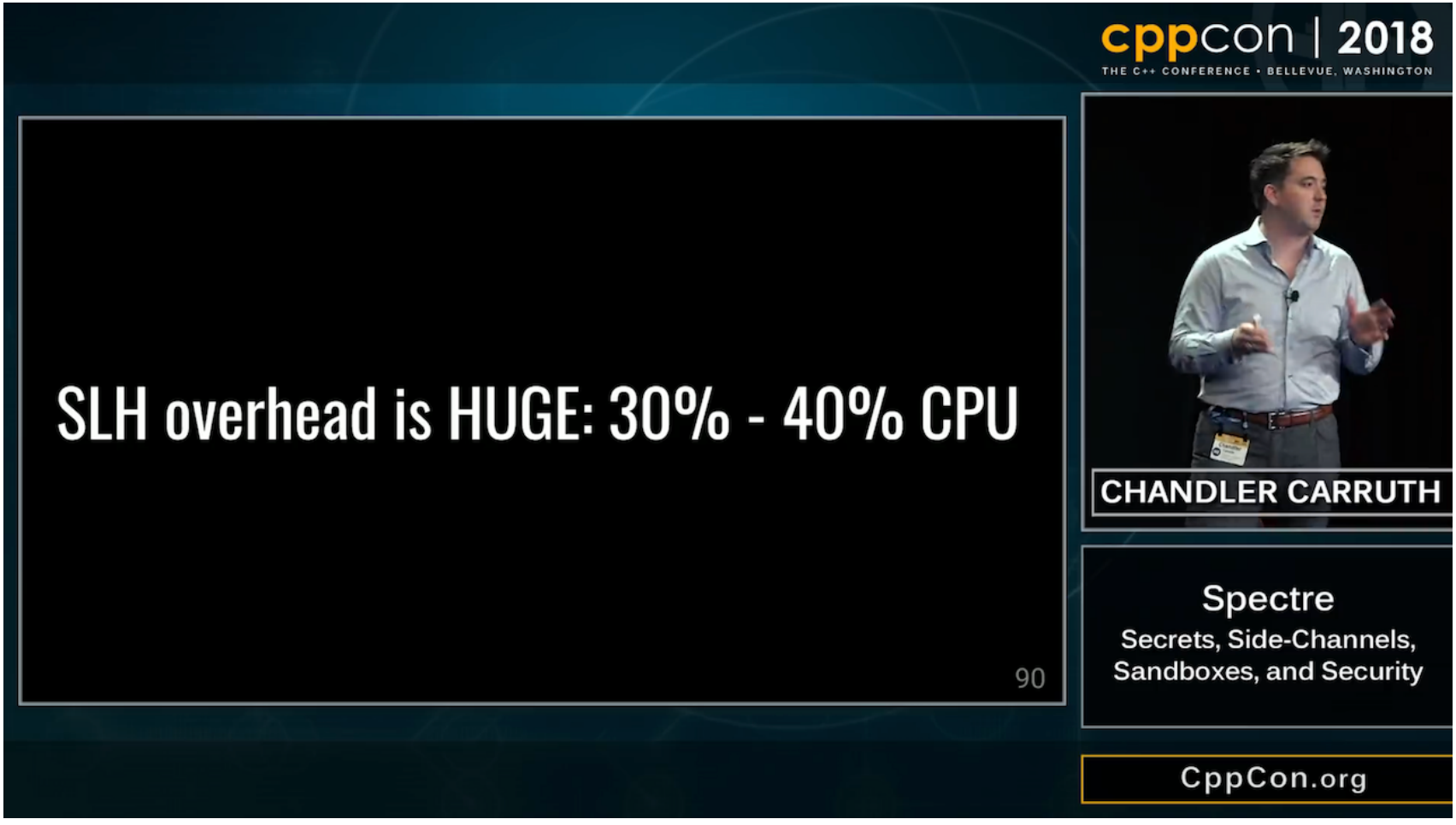
CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (5/9)



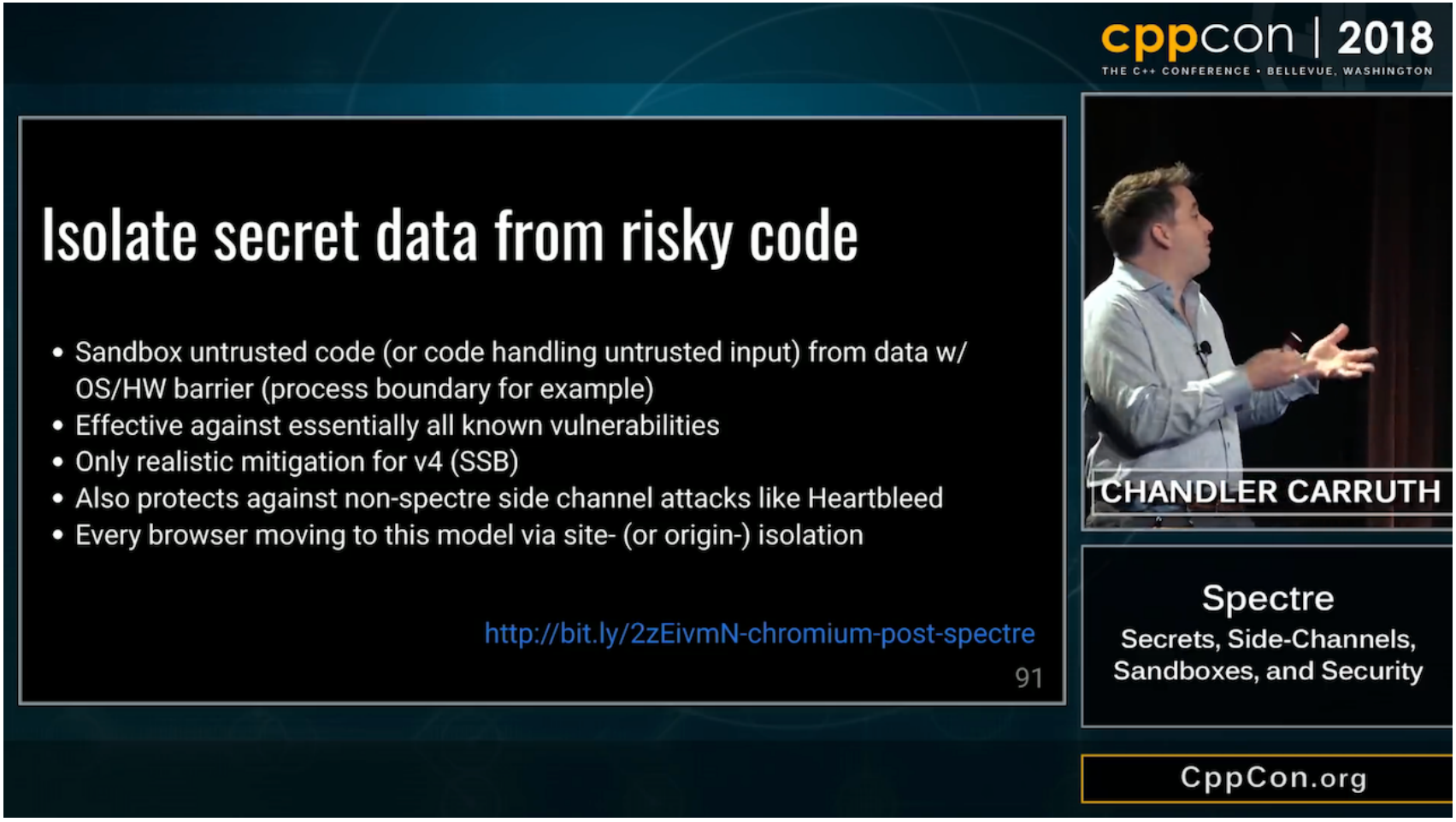
CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (6/9)



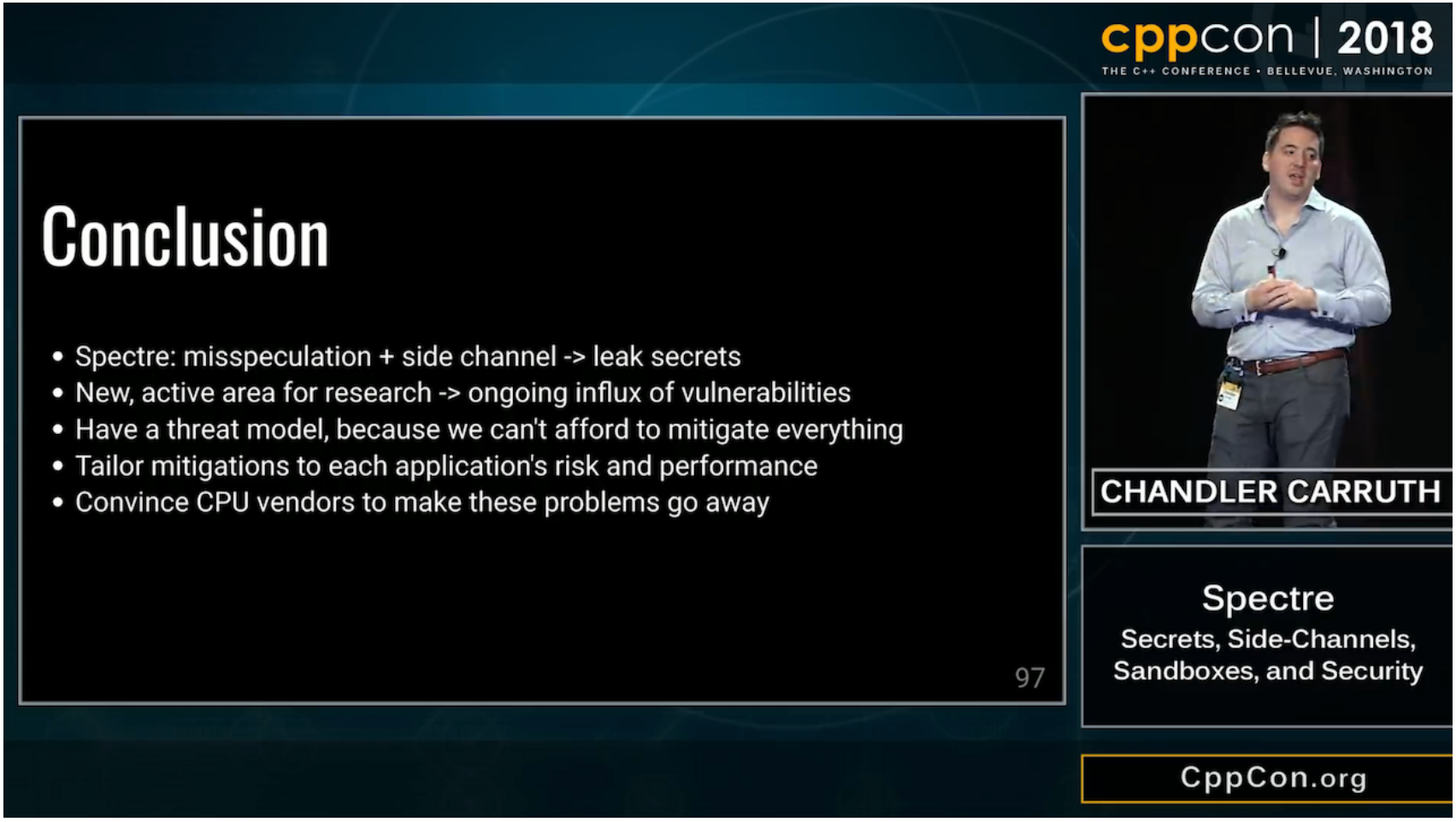
CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (7/9)



CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (8/9)



CppCon 2018: Chandler Carruth “Spectre: Secrets, Side-Channels, Sandboxes, and Security” (9/9)



Meltdown, Spectre etc.

- Matt Klein
- Graham Sutherland via Frank Denneman

Hardware Effects

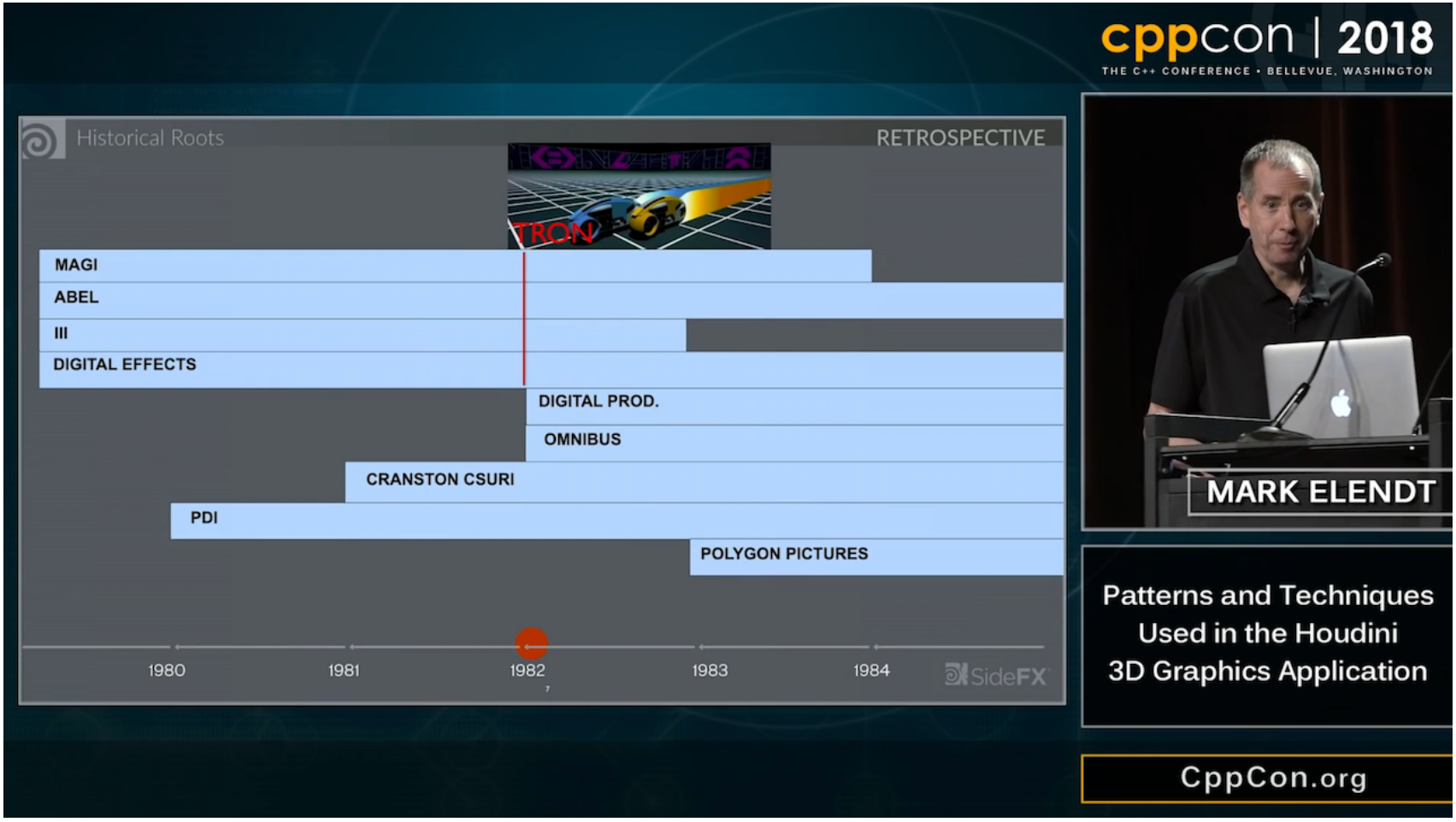
This repository demonstrates various hardware effects that can degrade application performance in surprising ways and that may be very hard to explain without knowledge of the low-level CPU and OS architecture. For each effect I try to create a proof of concept program that is as small as possible so that it can be understood easily.

- GitHub – Reddit
  - Demonstrates: bandwidth saturation, branch misprediction, branch target misprediction, cache aliasing, cache/memory hierarchy bandwidth, data dependencies, denormal floating point numbers, false sharing, hardware prefetching, memory-bound program, non-temporal stores, software prefetching, write combining.

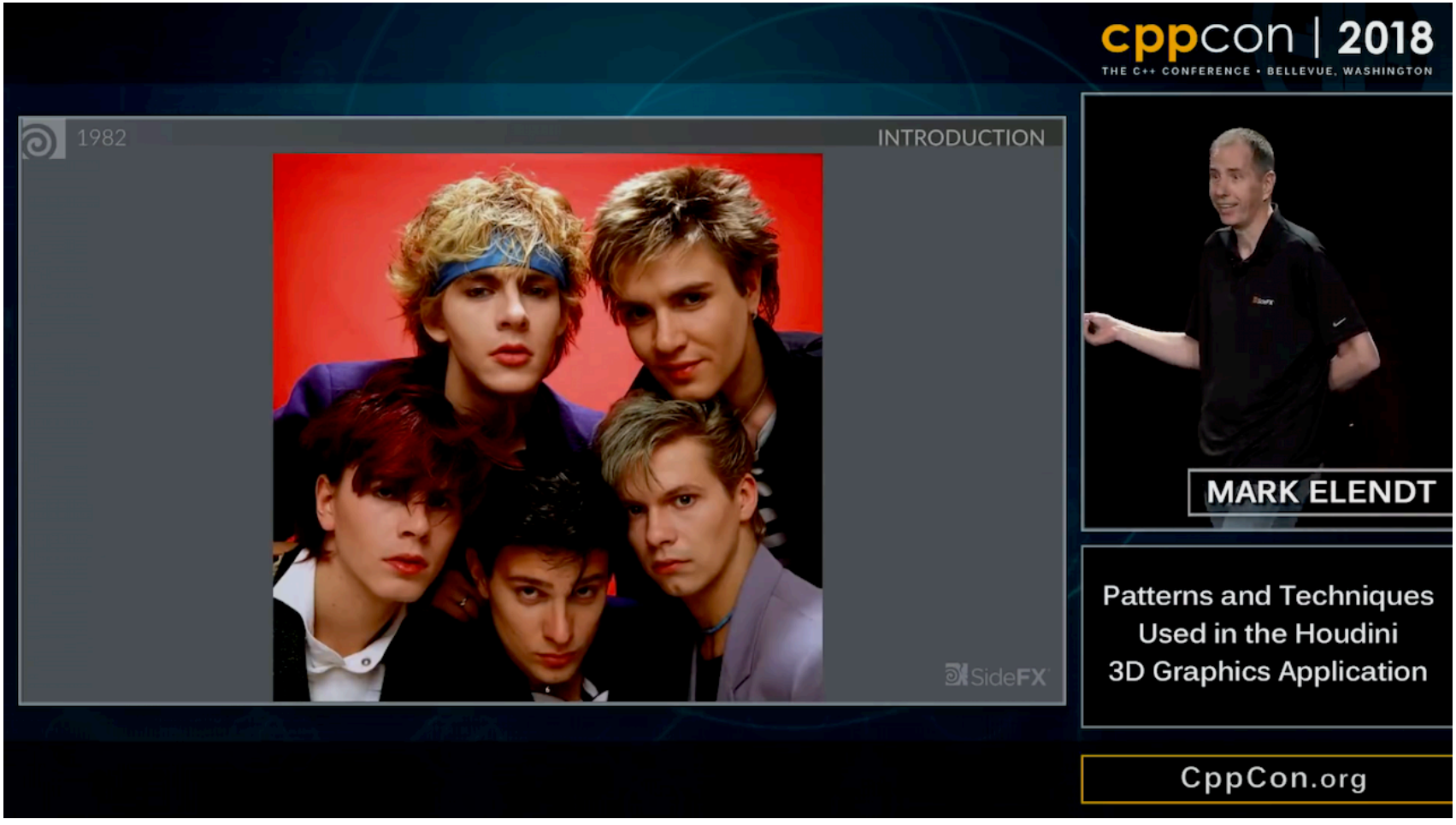
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (1/13)

Video

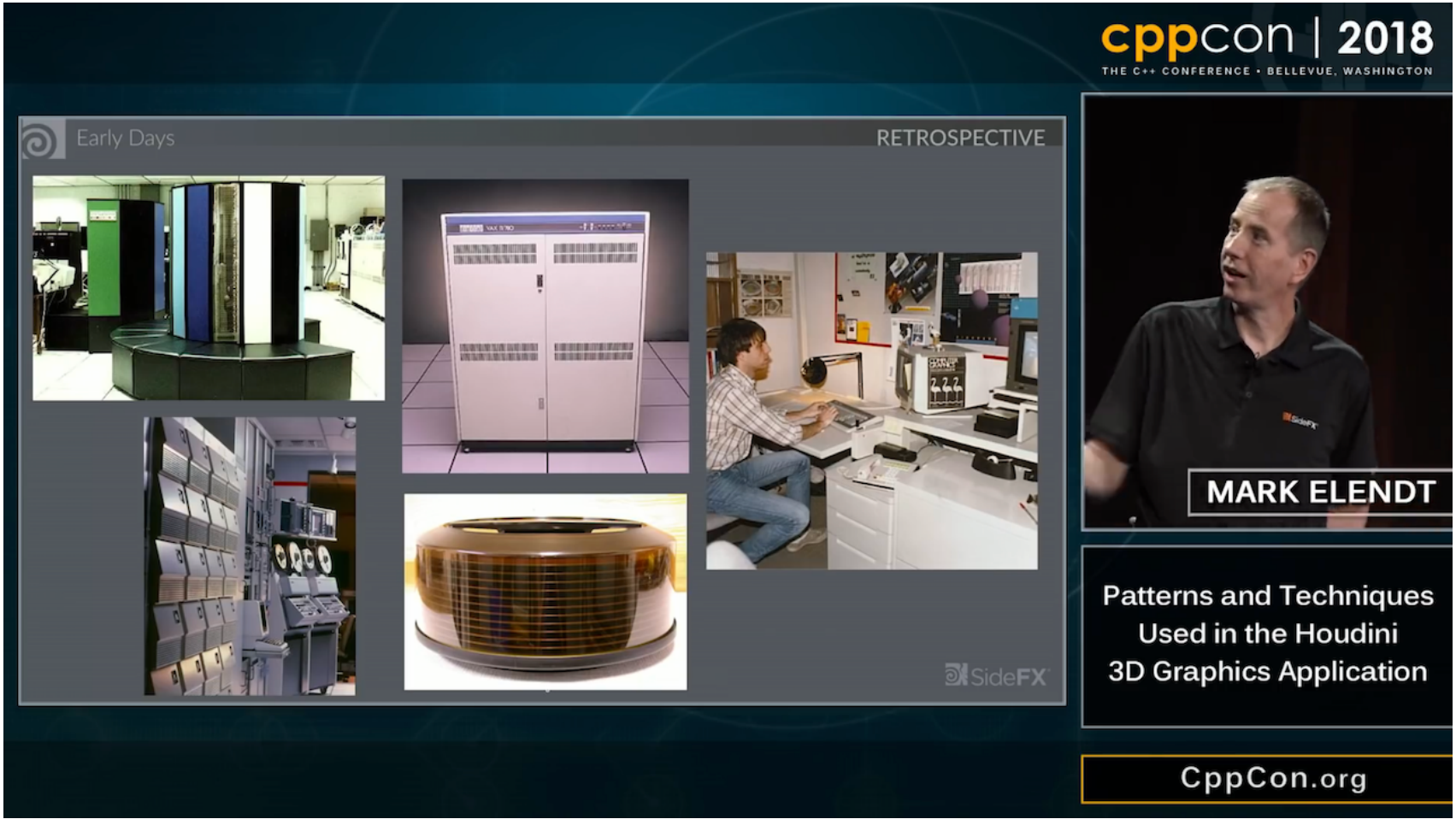
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (2/13)



CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (3/13)



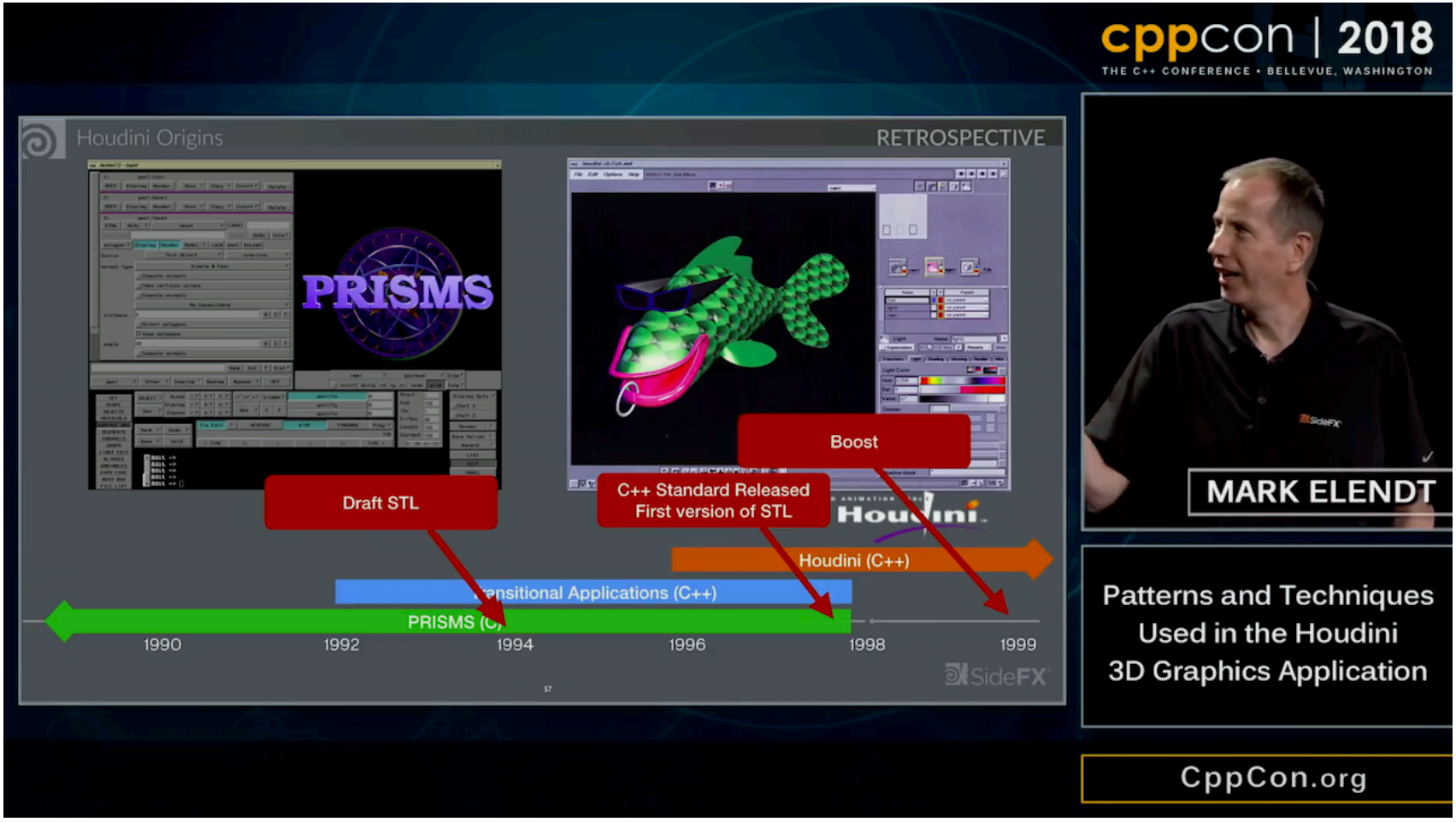
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (4/13)



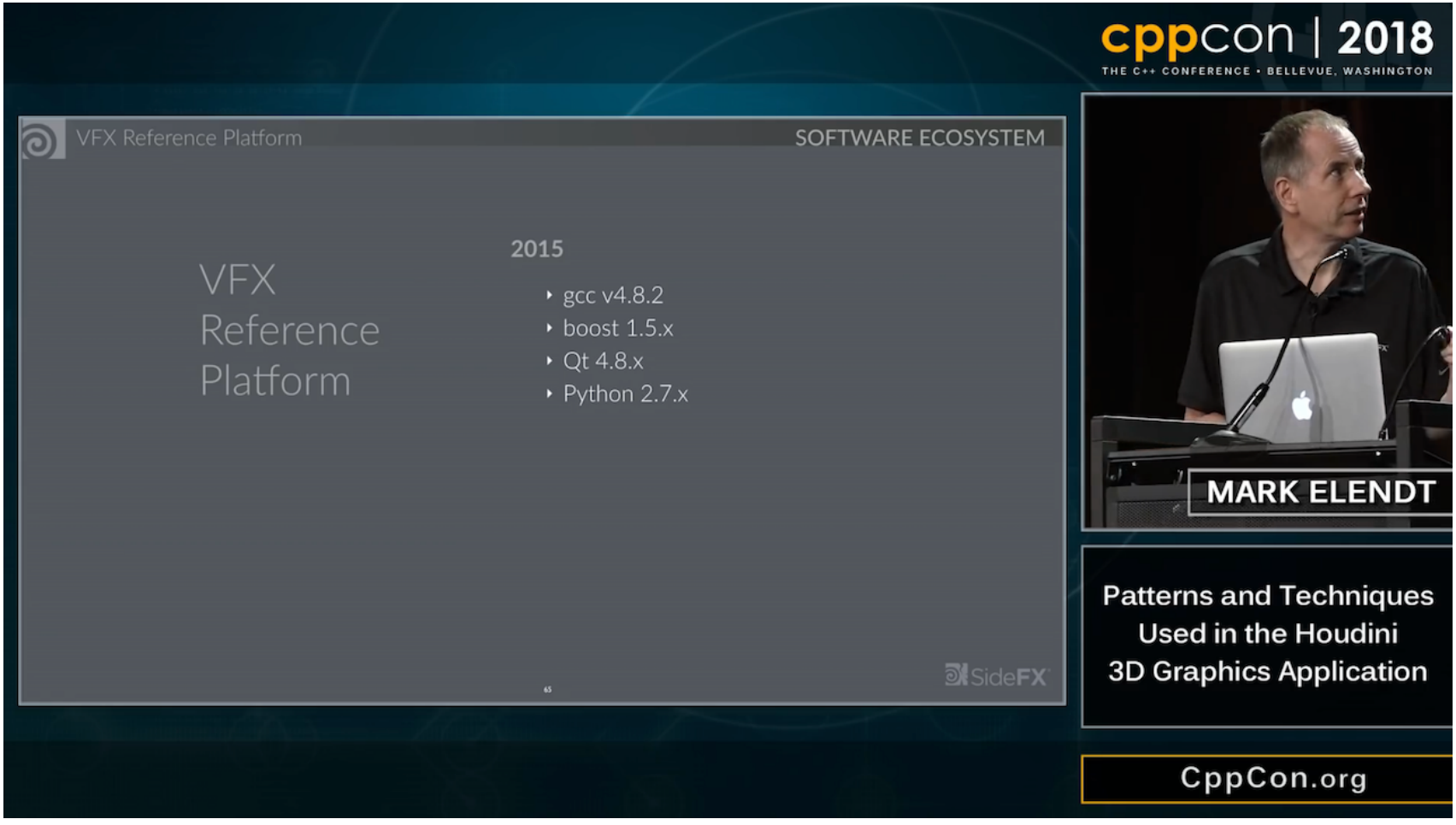
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (5/13)

CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (6/13)

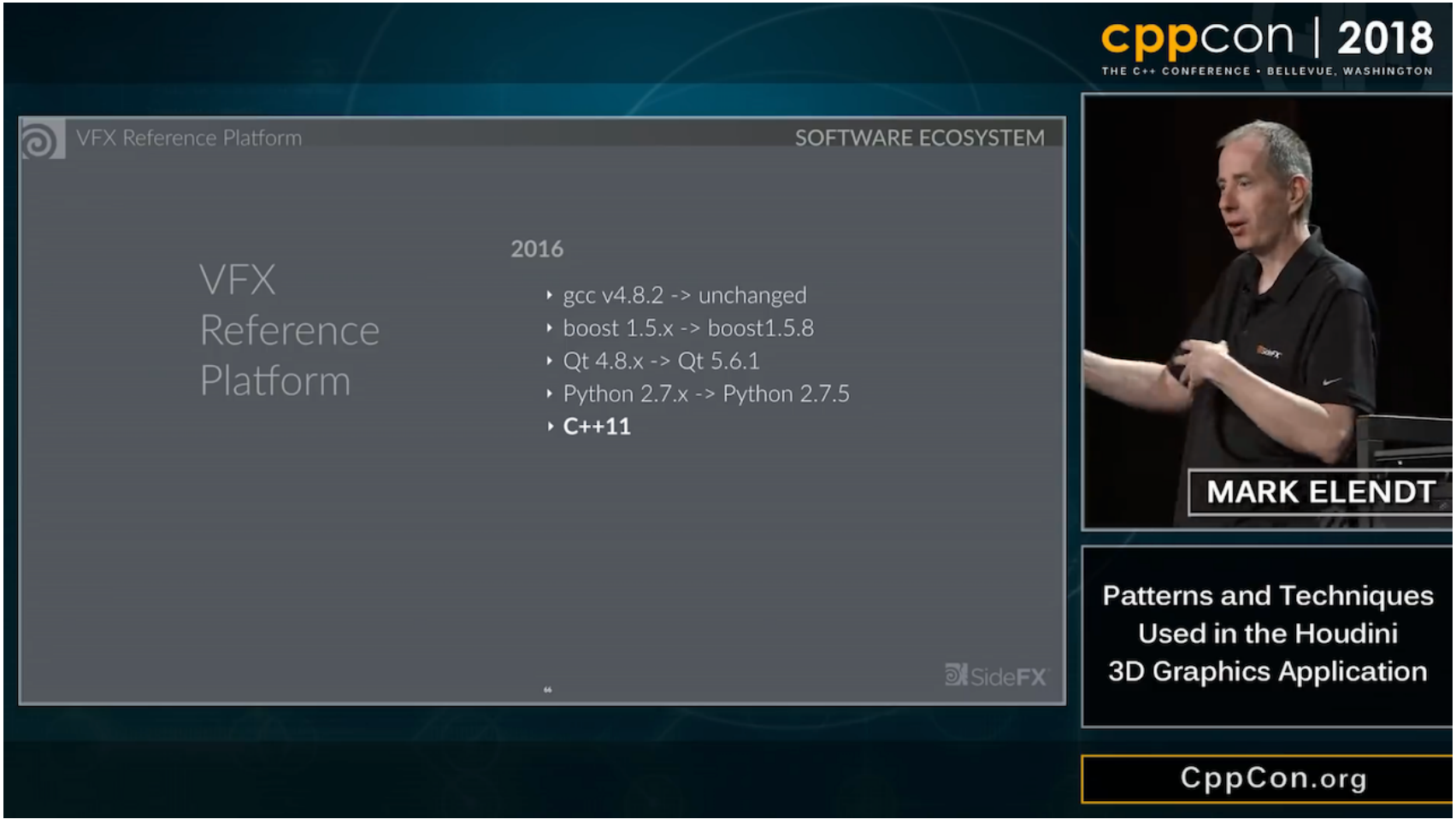
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (7/13)



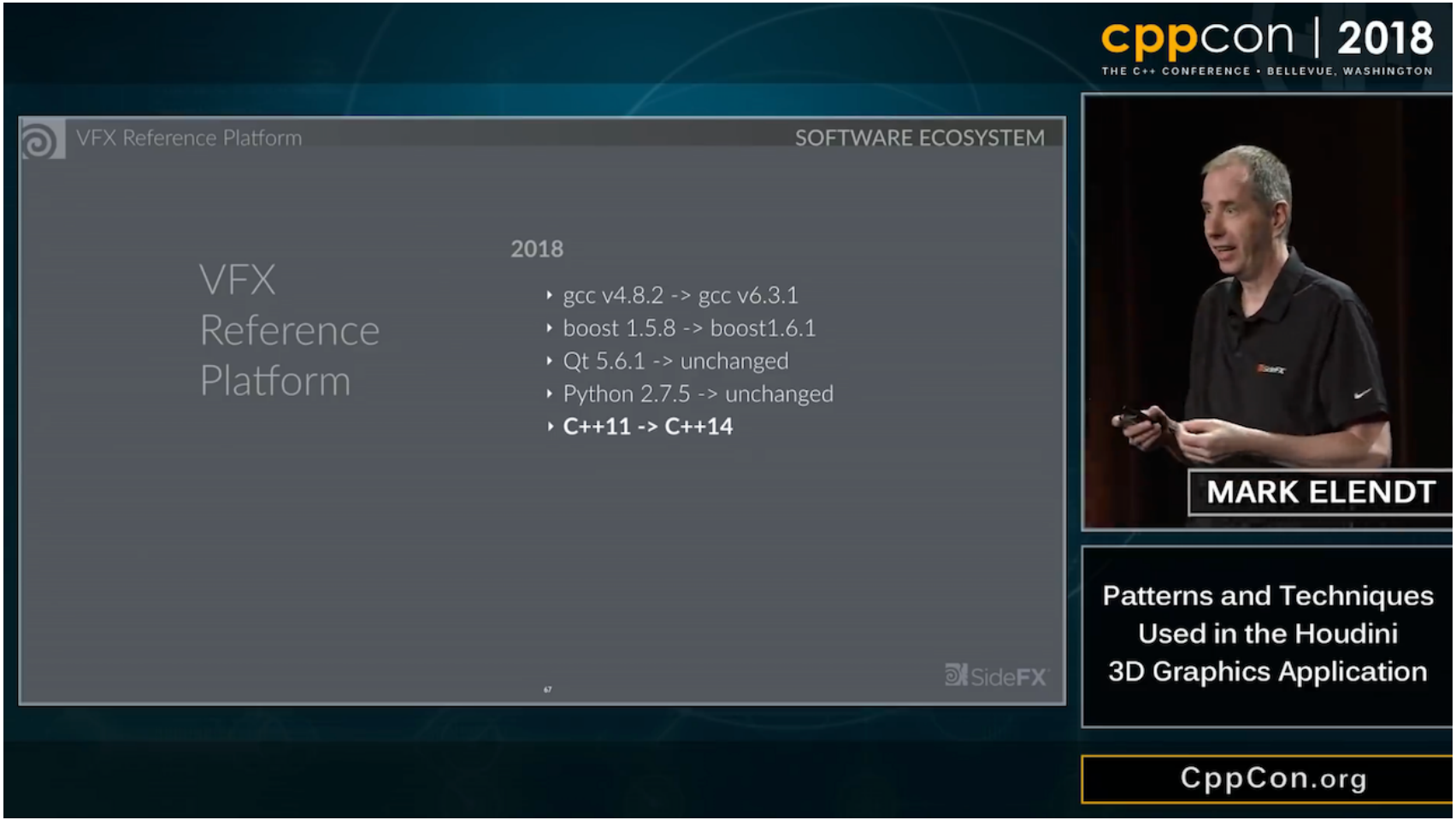
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (8/13)



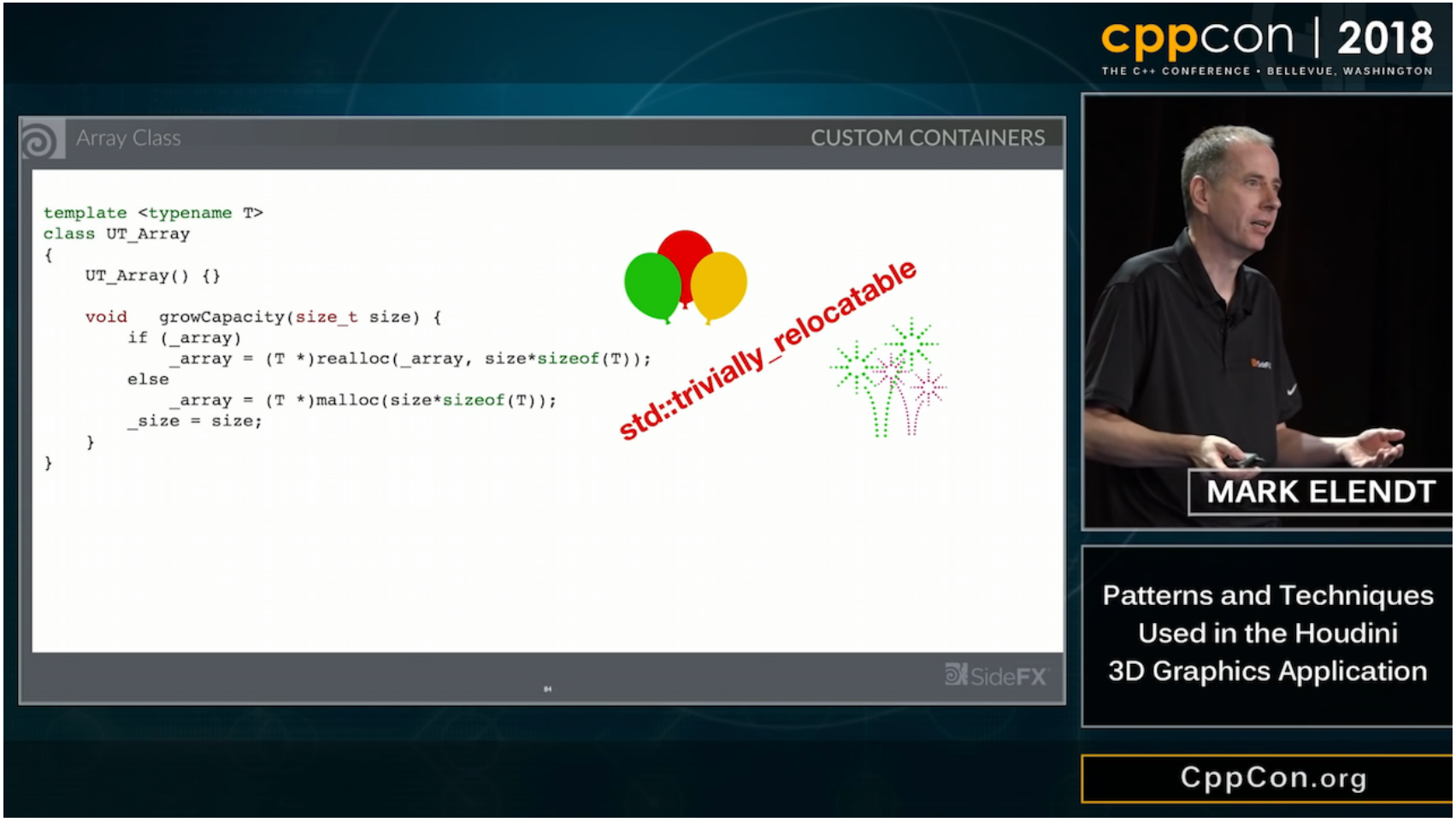
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (9/13)



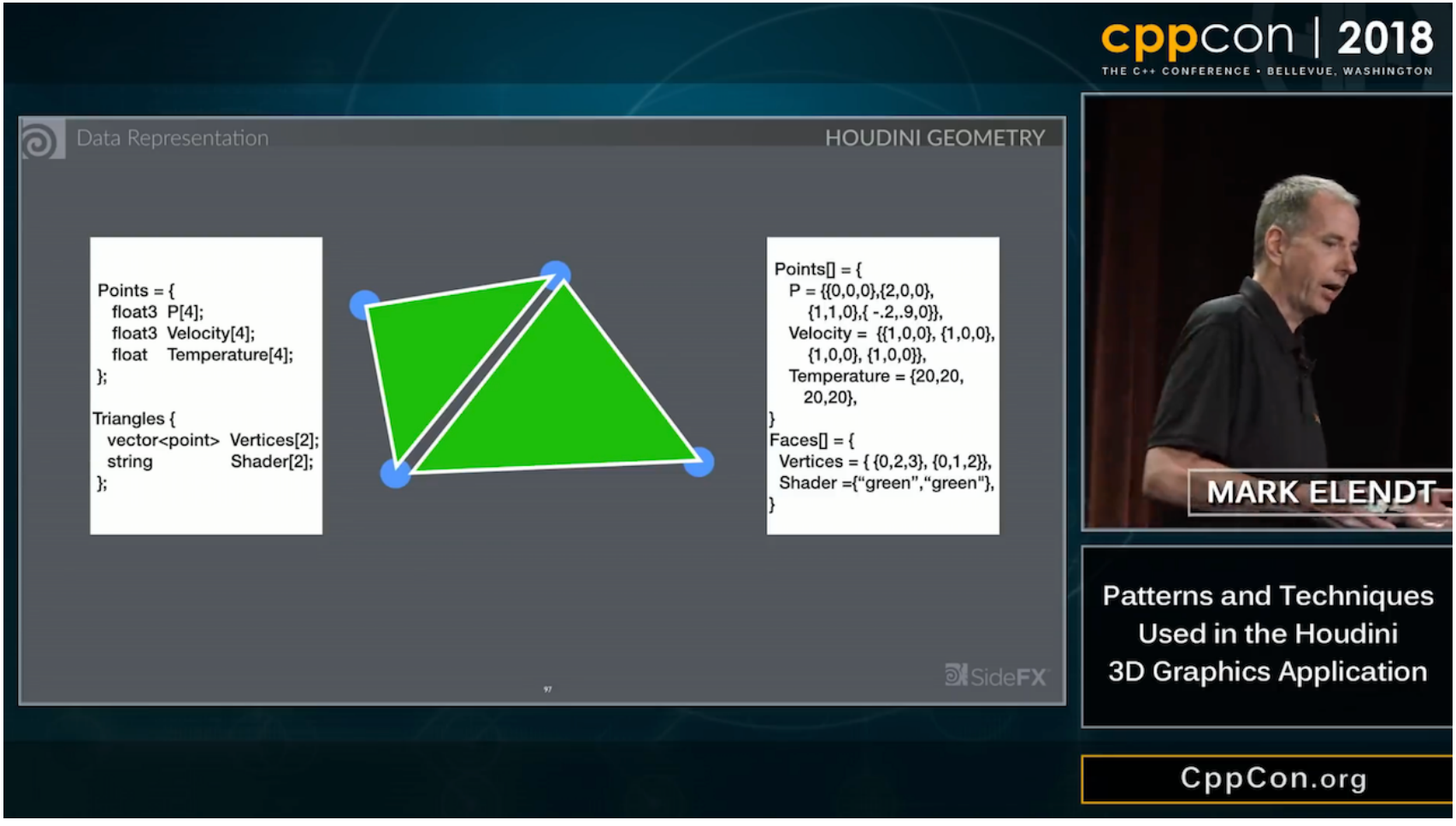
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (10/13)



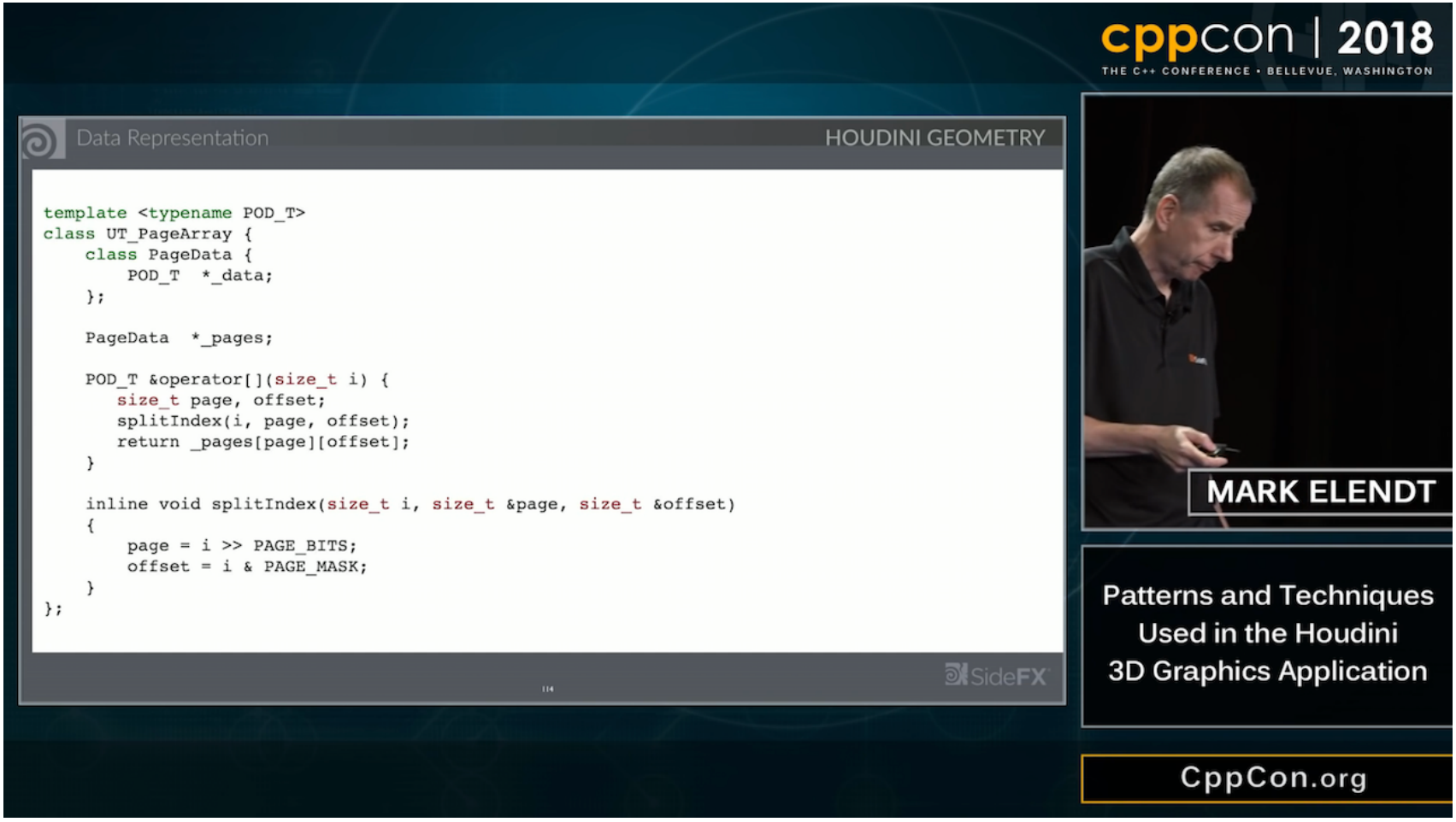
CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (11/13)



CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (12/13)



CppCon 2018: Mark Elendt “Patterns and Techniques Used in the Houdini 3D Graphics Application” (13/13)



Who is STL? I mean the person, not the library

Reddit

> *Hey. I'm Stephan T. Lavavej ("Steh-fin Lah-wah-wade"), and I've worked on MSVC's STL since 2007. I've also worked on several Standard proposals that were accepted (notably the transparent operator functors). I filmed a bunch of videos for MS's Channel 9 years ago, introducing various Core Language and Standard Library topics, and I've given talks at C++Now (formerly BoostCon) and CppCon which have been recorded.*

Real world problems with #pragma once?

Reddit

Pointer-to-member-functions can be tricky

- Post
- Snippet
- Raymond Chen: Pointers to member functions are very strange animals

Prepare thy Pitchforks: A De-facto Standard Project Layout

- Early Reddit post
- Later Reddit post
- Blog post
- GitHub repo

Reimplementing NumPy in C++

- NumCpp
- xtensor

Other linear algebra libraries

- Blaze
- Eigen
  - the official repo
  - docs

Visual C++ Team Blog - std::any: How, when, and why

Post

When you need to store an object of an arbitrary type, pull std::any out of your toolbox. Be aware that there are probably more appropriate tools available when you do know something about the type to be stored.

C++ Best Practices, by Jason Turner

GitHub

Library: SQLite ORM

SQLite ORM light header only library for modern C++

- Code (BSD-2-Clause)
  - C++14
  - No raw string queries
  - CRUD support; pure select query support
  - Custom types binding support
  - Supports: `BLOB` - maps to `std::vector<char>` or a custom type; `FOREIGN KEY`; composite keys; `JOIN`; transactions; `ORDER BY` and `LIMIT`, `OFFSET`, `GROUP BY` / `DISTINCT`, `INDEX`, `COLLATE`
  - Migration functionality
  - The only dependency - **libsqlite3**
  - No undefined behaviour - if something goes wrong lib throws an exception
  - In-memory database support - provide `:memory:` or empty filename

Library: Inja - a template engine for modern C++

- Code
  - Licence: MIT
  - Header-only
  - Uses NLohmann's JSON library
  - Conan wrapper

```
1 json data;
2 data["name"] = "world";
3 inja::render("Hello {{ name }}!", data); // Returns "Hello world!"
```

Library: C++ REST SDK (formerly Casablanca) by Microsoft

- Code
  - Licence: MIT
  - C++11
  - Supports Windows, Linux, macOS, iOS, Android

The C++ REST SDK is a Microsoft project for cloud-based client-server communication in native code using a modern asynchronous C++ API design. This project aims to help C++ developers connect to and interact with services.

Library: Caffe2 - A New Lightweight, Modular, and Scalable Deep Learning Framework

- Website
- Code
  - Licence: Apache-2.0

Tool: Superluminal profiler for Windows

Website

- Combines sampling and instrumentation
- Visualizes thread communication flow
- Kernel-level callstacks
- Dynamic filtering of areas of interest
- High frequency sampling (8 KHz)
- Timeline view, call graph, source view
- 7-day free trial, then EUR 99/149/289

Conan, vcpkg or build2?

Reddit

- Pragmatic choice: vcpkg or Conan (they work today and are complete enough)
- Pragmatic no-brainer choice: vcpkg (it's the simplest and it have more packages ready)
- Pragmatic but need finer control choice: Conan (it gives more options)
- (Very) Long term choice: Build2 (shows great promises because it uses a coherent model...)
- Ideal choice (from the future): help SG15 (the group reflecting on tools vs C++) define interfaces for build systems and dependency managers so that your choice is not impacted by your dependencies choices.

Improving C++ Builds with Split DWARF

Article

```
$ g++ -c -g -gsplit-dwarf main.cpp -o main.o
$ g++ main.o -o app
```

Having some fun with higher-order functions

- Article by Barry Revzin
- Boost.HOF

Compile-time raytracer by Tristan Brindle

- Code
  - Reddit

Iterators: What Must Be Done?

- Article

Quote

Andrey Mokhov (@andreymokhov) via Twitter:

> *Inside every large program there is a small build system struggling to get out.*

Twitter

chrisk5000
@chrisk5000

Almost made fun of an electrician today like "is it an electrician's guild rule that you have to perform 5 minutes of ritual complaining about What Was The Previous Electrician Thinking before you're allowed to fix anything?" but then I remembered I'm a software engineer

8,883 Likes	**2,464** Retweets
15 Nov 2018 at 17:25	via **Twitter Web Client**