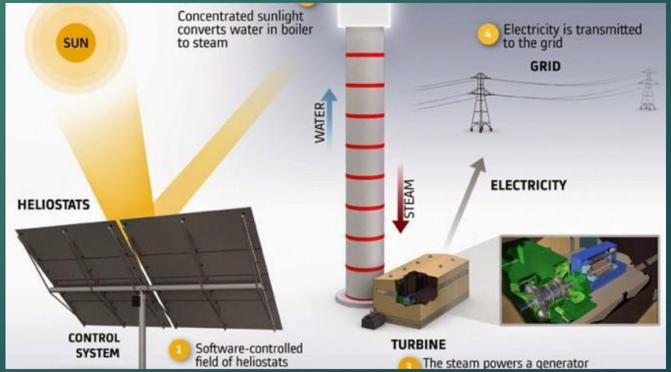
Code to Code

AUTOMATIC CODE GENERATION FROM C++ CODE

ASAF HELFER

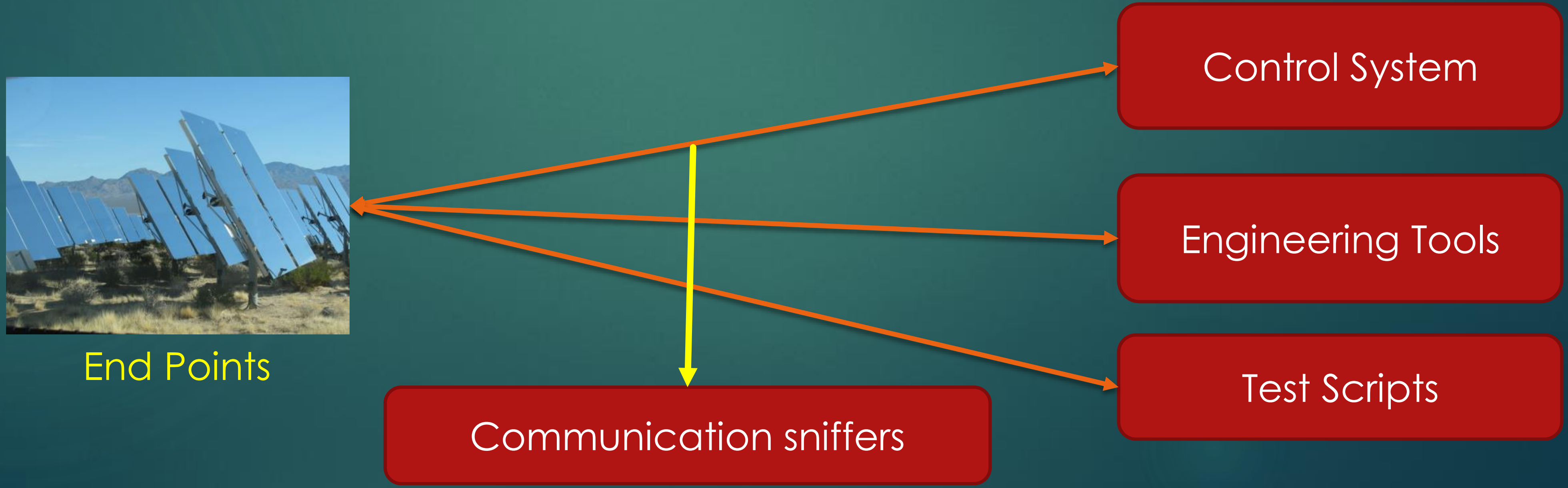
CORE C++ IL LIGHTNING TALKS, JUN 2018

BrightSource


SUN
Concentrated sunlight converts water in boiler to steam
4 Electricity is transmitted to the grid
GRID
WATER
STEAM
ELECTRICITY
HELIOSTATS
CONTROL SYSTEM
1 Software-controlled field of heliostats
TURBINE
3 The steam powers a generator

Problem

- Coordinate protocol of hundreds of commands and data items that can be sent or read from thousands of end points
- Various systems that should know this data
  - Cross platform – embedded devices, servers, test scripts, engineering tools
  - Multiple languages – C++, C#, Python, Lua



End Points

Control System

Engineering Tools

Test Scripts

Communication sniffers

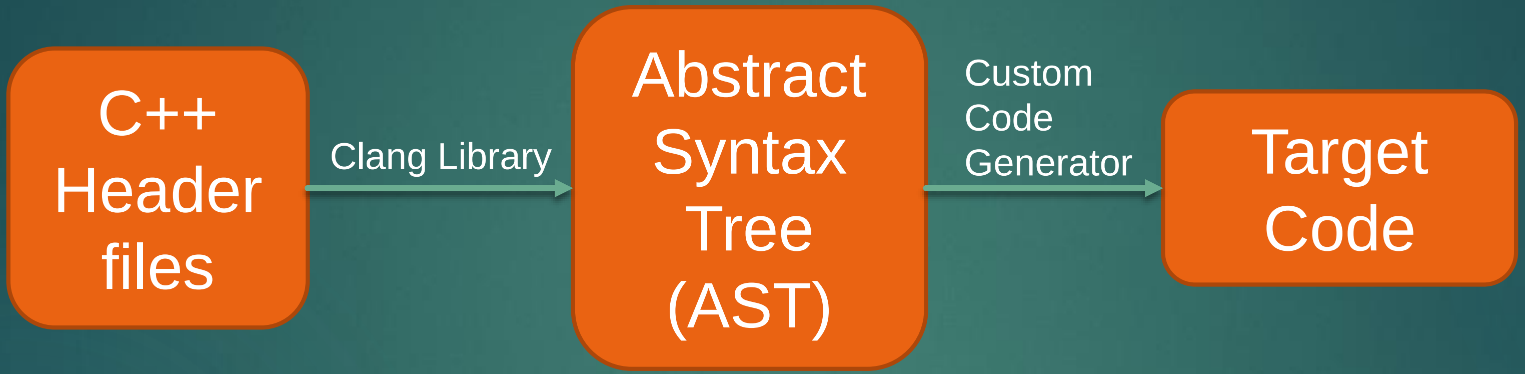
Problem - continued

- Requirements:
  - Need code in multiple languages that can parse and create messages in protocol
  - Protocol is continuously changing – commands and data items are added and modified, so should be done automatically
  - Size should be:
    - Optimized
    - Known at compile time because of communication protocol requirements

Solution: Code Generation
But from what?

- From DB
  - No version control
  - Complicates development
- From configuration/message files
  - For example:
    - Protobuf (Google)
    - Bond (Microsoft)
    - Apache Thrift
  - No sufficient control over serialization
    - Should be fixed sized for our communication protocol
    - Can’t optimize – e.g. bit fields
  - Not completely cross platform

Generate Code From C++

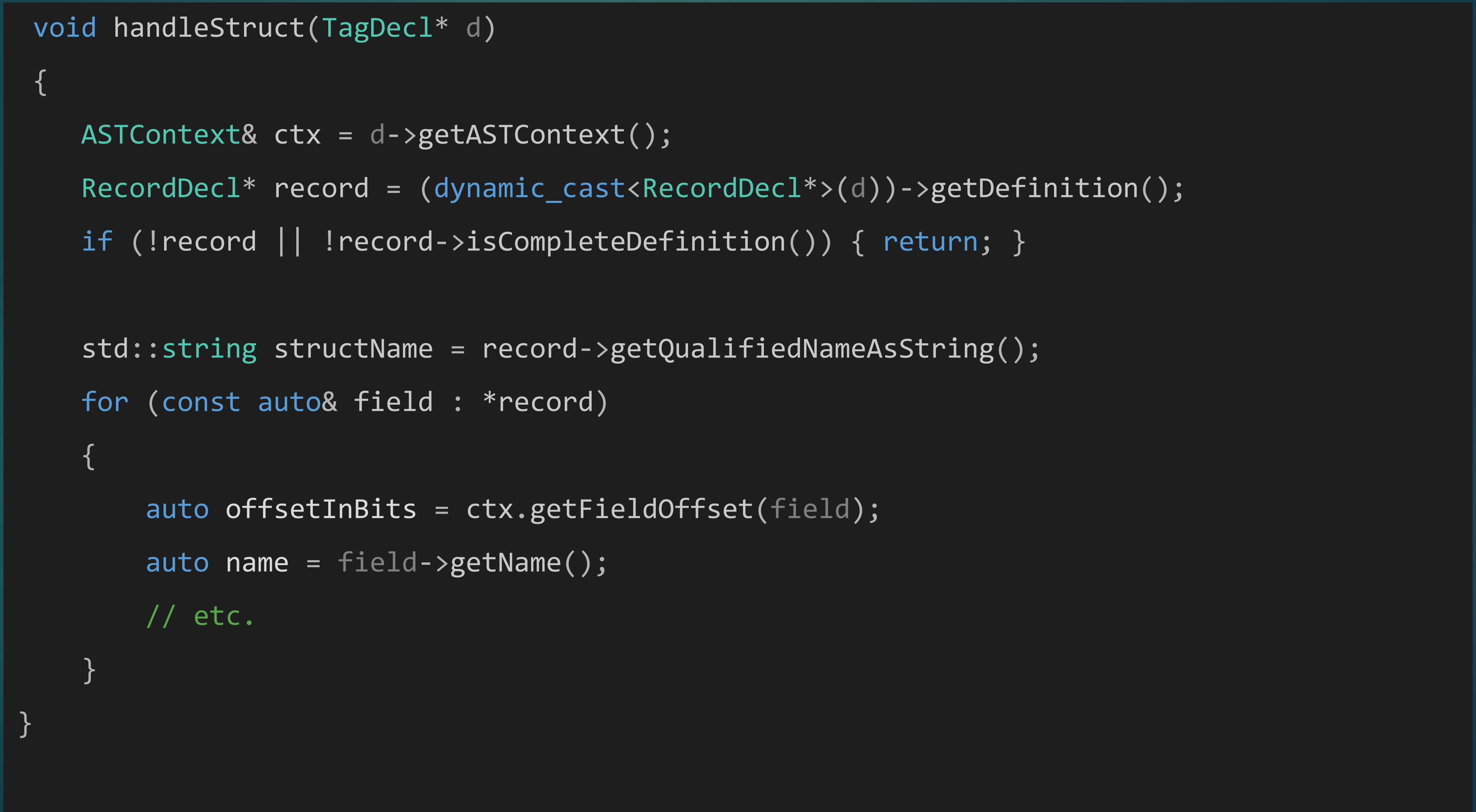


- Complete control over output
- Part of the developer flow (directly editing version-controlled source code files only)
- Cross platform-ness depends on C++ code only

Some Code… Visiting the AST

```
class MyVisitor : public RecursiveASTVisitor < MyVisitor >
{
    virtual bool VisitTagDecl(TagDecl* d) override
    {
        ASTContext& ctx = d->getASTContext();
        if (d->getIdentifier())
        {
            if (d->isStruct() || d->isClass())
            {
                handleStruct(d);
            }
            else if (d->isEnum())
            {
                handleEnum(d);
            }

            return true;
         }
     }
};
```

Some Code… Iterating struct fields



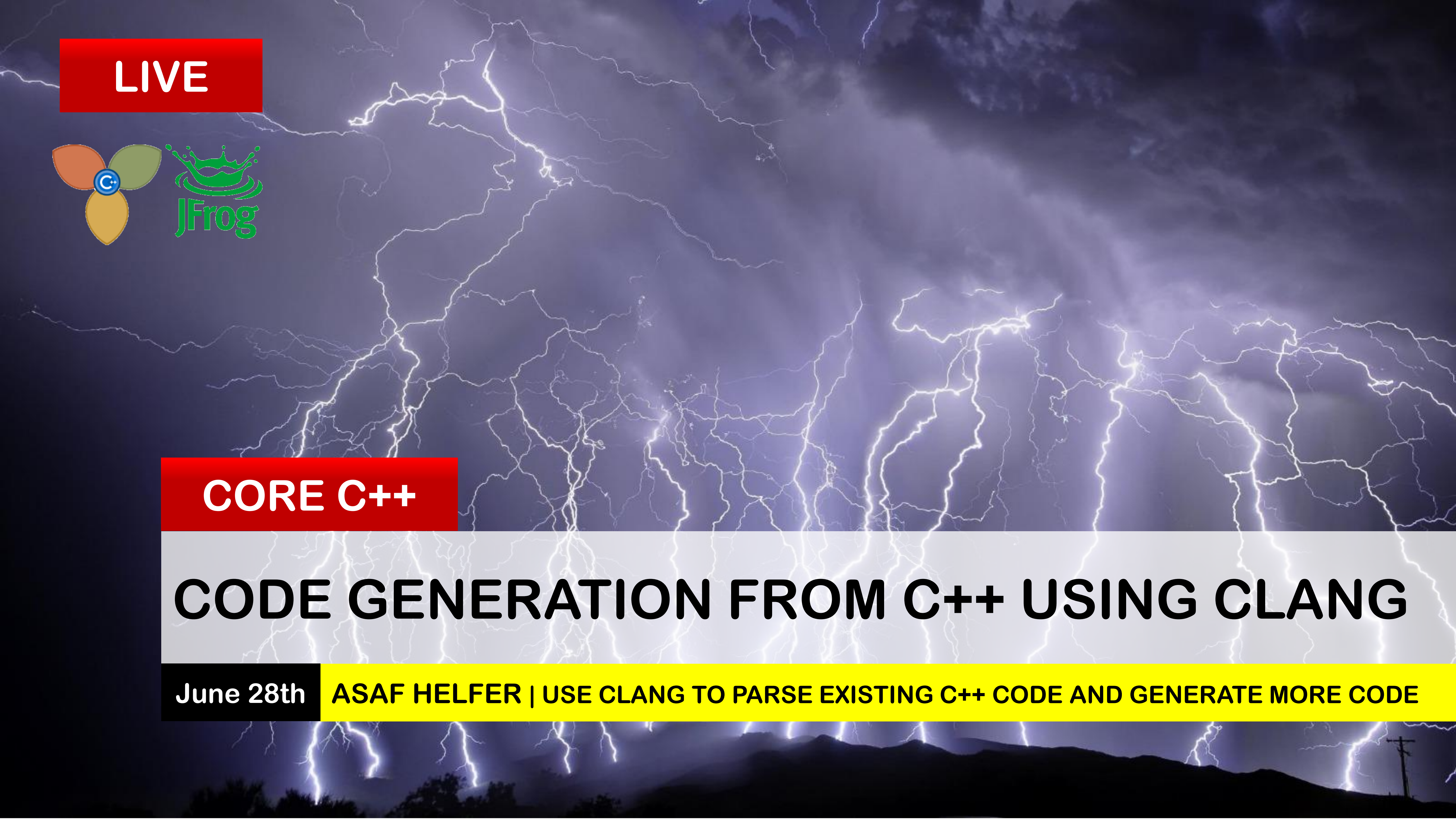
```
 void handleStruct(TagDecl* d)
 {
    ASTContext& ctx = d->getASTContext();
    RecordDecl* record = (dynamic_cast<RecordDecl*>(d))->getDefinition();
    if (!record || !record->isCompleteDefinition()) { return; }

    std::string structName = record->getQualifiedNameAsString();
    for (const auto& field : *record)
    {
        auto offsetInBits = ctx.getFieldOffset(field);
        auto name = field->getName();
        // etc.
    }
}
```

What else can we do with it?

- Automatically generate serialization for complex classes
- Automatically generate metadata for enums (automatic toString()!)

Can also do it with C# (ClangSharp)

```
CXTranslationUnit translationUnit;
var errorCode = clang.parseTranslationUnit2(..., out translationUnit);
if (errorCode == CXErrorCode.CXError_Success)
{
    clang.visitChildren(
        clang.getTranslationUnitCursor(translationUnit),
        structVisitor.Visit,
        new CXClientData(IntPtr.Zero));
}
```



LIVE

CORE C++

CODE GENERATION FROM C++ USING CLANG

June 28th

ASAF HELFER | USE CLANG TO PARSE EXISTING C++ CODE AND GENERATE MORE CODE

FIN