Clang Interface Stubs

Syntax Directed Stub Library Generation

Puyan Lotfi
Facebook

Interface Stubs

```
echo "" | clang -shared -fPIC -x c - -o - | llvm-objdump -section-headers

a.out:  file format ELF64-x86-64

Sections:
Idx Name            Size     Address          Type
  0                 00000000 0000000000000000
  1 .dynsym         00000108 00000000000001d0
  2 .dynstr         0000008f 00000000000002d8
  3 .symtab         00000498 0000000000000000
  4 .strtab         00000178 0000000000000000
  5 .shstrtab       000000cc 0000000000000000
  6 .gnu.hash       0000003c 0000000000000190
  7 .gnu.version    00000016 0000000000000368
  8 .gnu.version_r  00000020 0000000000000380
  9 .rela.dyn       000000a8 00000000000003a0
 10 .init           00000017 0000000000000448 TEXT
 11 .plt            00000010 0000000000000460 TEXT
 12 .plt.got        00000008 0000000000000470 TEXT
 13 .text           000000c6 0000000000000480 TEXT
 14 .fini           00000009 0000000000000548 TEXT
 15 .eh_frame_hdr   00000024 0000000000000554 DATA
 16 .eh_frame       0000007c 0000000000000578 DATA
 17 .init_array     00000008 0000000000200e40
 18 .fini_array     00000008 0000000000200e48
 19 .dynamic        00000190 0000000000200e50
 20 .got            00000020 0000000000200fe0 DATA
 21 .got.plt        00000018 0000000000201000 DATA
 22 .data           00000008 0000000000201018 DATA
 23 .bss            00000008 0000000000201020 BSS
 24 .comment        0000008f 0000000000000000
```

Interface Stubs

```
echo "" | clang -shared -fPIC -x c - -o - | llvm-objdump -section-headers

a.out:  file format ELF64-x86-64

Sections:
Idx Name            Size     Address          Type
  0                 00000000 0000000000000000
  1 .dynsym         00000108 00000000000001d0
  2 .dynstr         0000008f 00000000000002d8
  3 .symtab         00000498 0000000000000000
  4 .strtab         00000178 0000000000000000
  5 .shstrtab       000000cc 0000000000000000
  6 .gnu.hash       0000003c 0000000000000190
  7 .gnu.version    00000016 0000000000000368
  8 .gnu.version_r  00000020 0000000000000380
  9 .rela.dyn       000000a8 00000000000003a0
 10 .init           00000017 0000000000000448 TEXT
 11 .plt            00000010 0000000000000460 TEXT
 12 .plt.got        00000008 0000000000000470 TEXT
 13 .text           000000c6 0000000000000480 TEXT
 14 .fini           00000009 0000000000000548 TEXT
 15 .eh_frame_hdr   00000024 0000000000000554 DATA
 16 .eh_frame       0000007c 0000000000000578 DATA
 17 .init_array     00000008 0000000000200e40
 18 .fini_array     00000008 0000000000200e48
 19 .dynamic        00000190 0000000000200e50
 20 .got            00000020 0000000000200fe0 DATA
 21 .got.plt        00000018 0000000000201000 DATA
 22 .data           00000008 0000000000201018 DATA
 23 .bss            00000008 0000000000201020 BSS
 24 .comment        0000008f 0000000000000000
```

Motivation

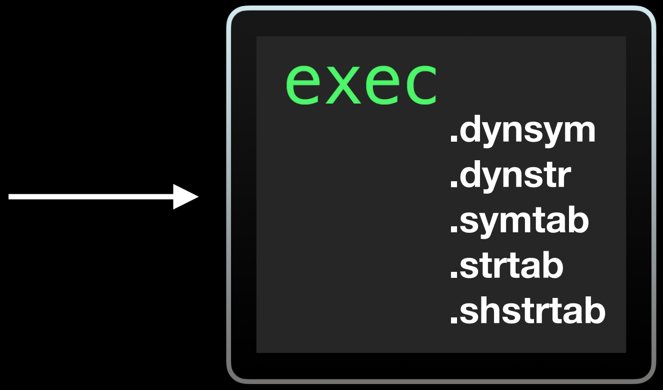
- Generation of lean SDKs
    - No Code
    - Explicit Symbol Exposure
- Code as Source of Truth: Syntax Directed
    - Make use of visibility attributes (ie __attribute__((__visibility__(“hidden”))))

Motivation

- Generation of lean SDKs
  - No Code
  - Explicit Symbol Exposure
- Code as Source of Truth: Syntax Directed
  - Make use of visibility attributes (ie __attribute__((__visibility__("hidden"))))

clang -emit-interface-stubs -o libfoo.so a.cpp c.cpp sq.cpp

```
#define hidden __attribute__(( \
  __visibility__("hidden")))

hidden int b;
int red() { return b; }
hidden void green() { }
hidden void blue() { }
int a;
hidden int c;
```



Prior Art & Approaches

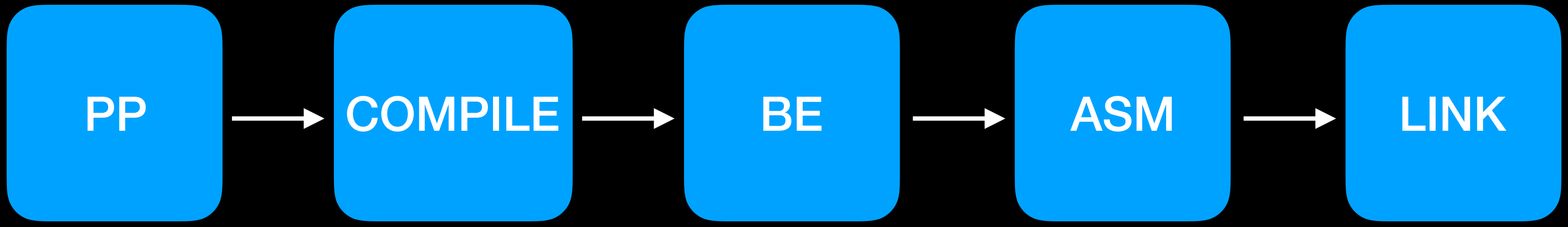
- Microsoft’s Import Libraries
    - Generate stub code from compiler & linker
    - Syntax directed through __declspec(dllimport/dllexport)
- Apple’s TAPI (Text API)
    - Header Scanning & stub generation
    - .so/.dylib/.dll scanning & stripping

Clang Interface Stubs

- Repurposes visibility attribute to direct symbol exposure using code syntax
    - Fine grain control (internal SPI vs external API)
- Aggregates exposure across compilation units via text
- Supports ELF
- Yields smaller SDK and faster link times

Clang Driver Pipeline

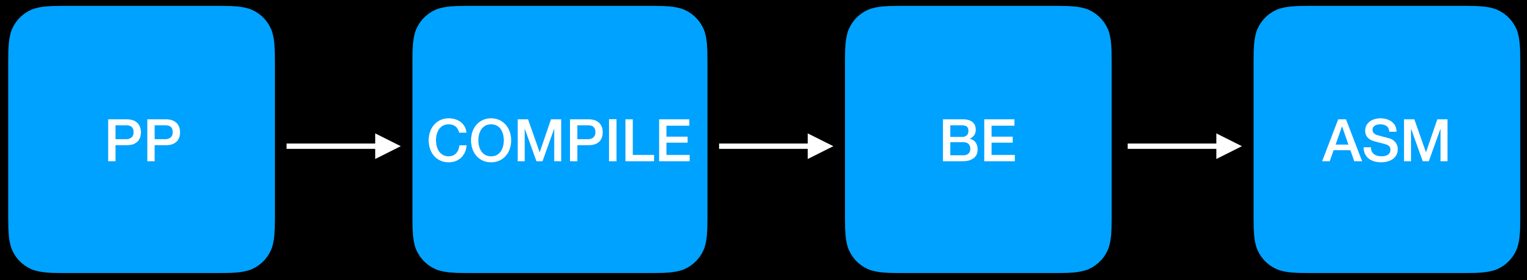
- Traditional Pipeline: Preprocess, Compile, Backend, Assemble, and Link Phases



Clang Driver Pipeline

- Traditional Pipeline: Preprocess, Compile, Backend, Assemble, and Link Phases

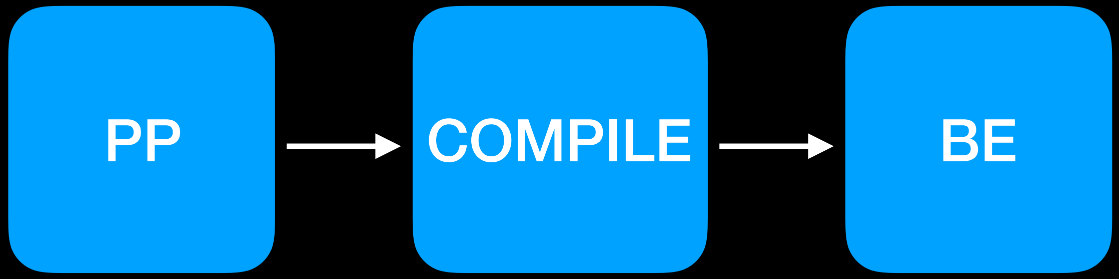
```
clang -c
```



Clang Driver Pipeline

- Traditional Pipeline: Preprocess, Compile, Backend, Assemble, and Link Phases

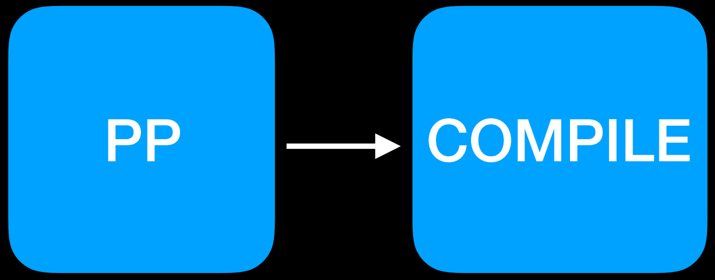
```
clang -S
```



Clang Driver Pipeline

- Traditional Pipeline: Preprocess, Compile, Backend, Assemble, and Link Phases

```
clang -fsyntax_only
```



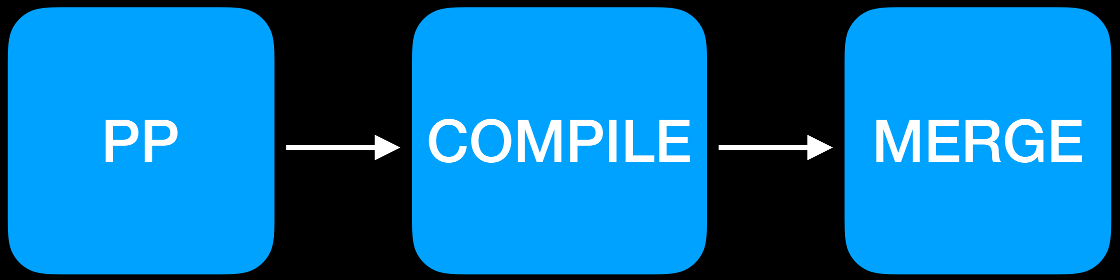
Clang Driver Pipeline

- Traditional Pipeline: Preprocess, Compile, Backend, Assemble, and Link Phases

```
clang -E
```

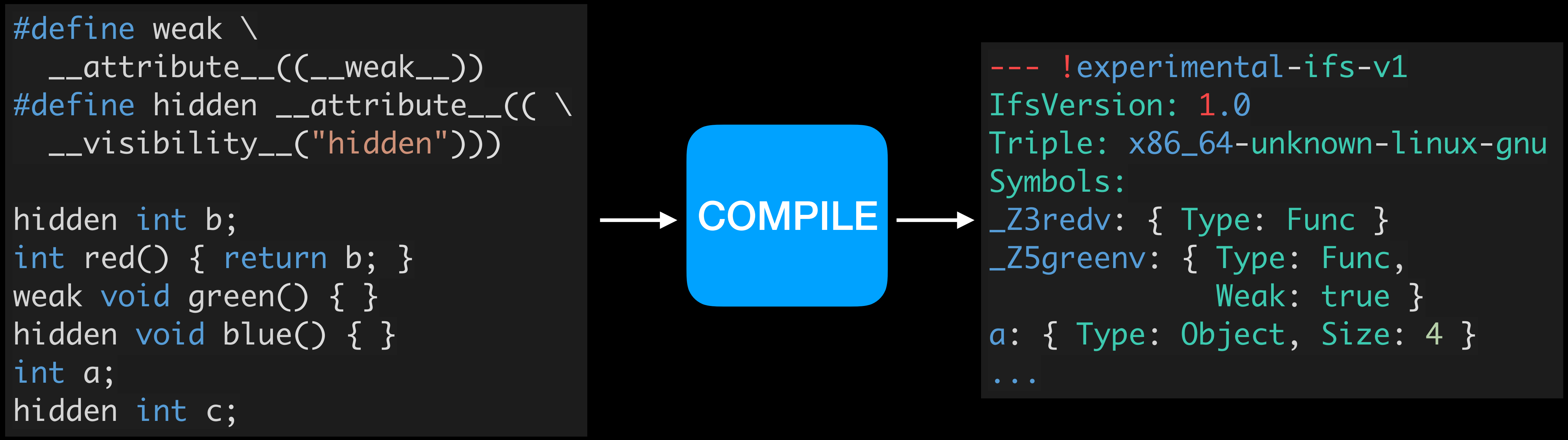
Driver Pipeline

- Clang Interface Stubs Pipeline: Preprocess, Compile, and Merge phases
- Compile Phase: Generates intermediate text (.ifs files)
- Merge Phase: Invokes llvm-ifs to consume & merge .ifs files to produce ELF .so



Driver Pipeline

- Clang Interface Stubs Pipeline: Preprocess, Compile, and Merge phases
- Compile Phase: Generates intermediate text (.ifs files)
- Merge Phase: Invokes llvm-ifs to consume & merge .ifs files to produce ELF .so
- Compile Phase invokes InterfaceStubFunctionsConsumer (clang -cc1)
    - Walks the AST scanning for visible decls



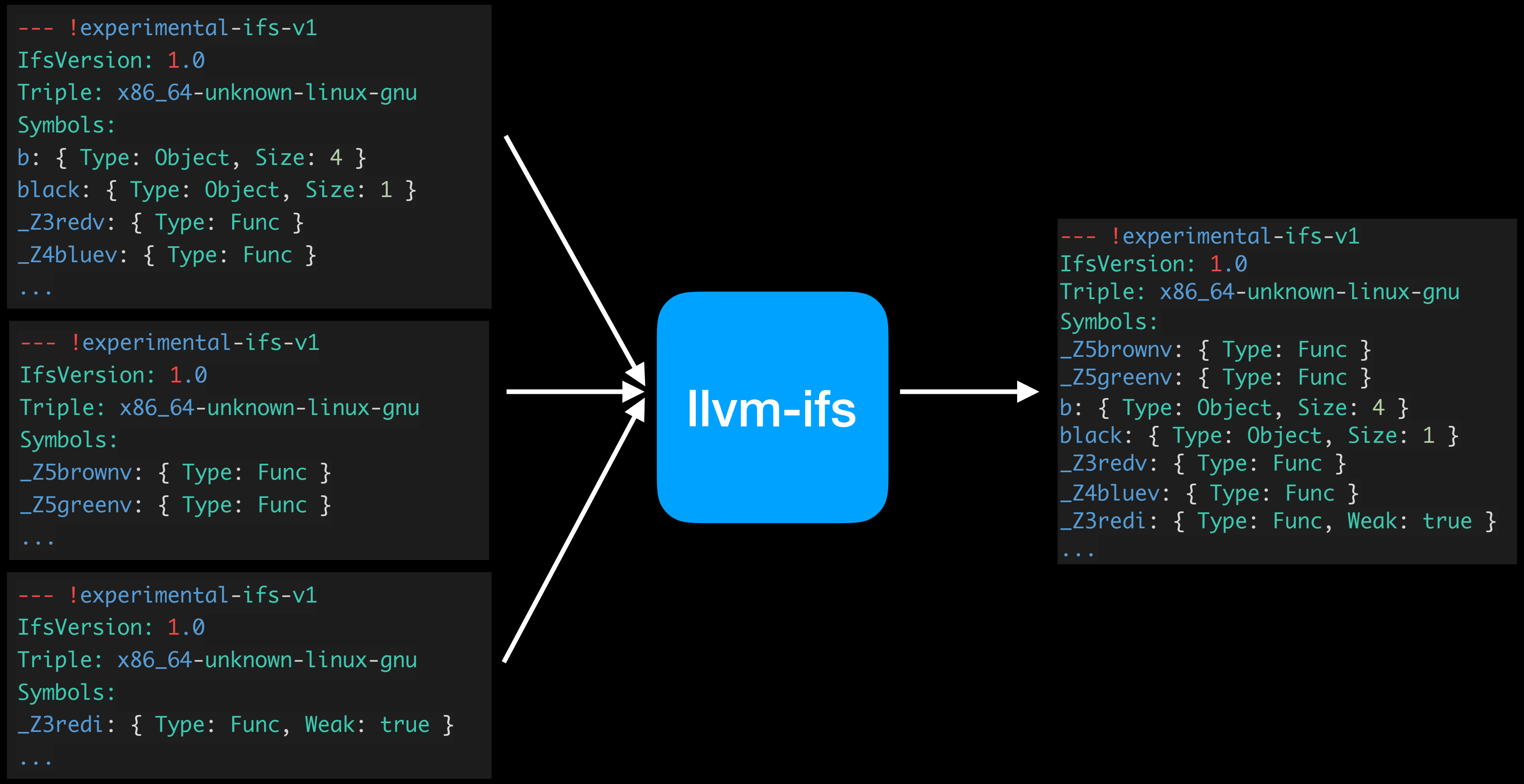
```
#define weak \
  __attribute__((__weak__))
#define hidden __attribute__(( \
  __visibility__("hidden")))

hidden int b;
int red() { return b; }
weak void green() { }
hidden void blue() { }
int a;
hidden int c;
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redv: { Type: Func }
_Z5greenv: { Type: Func,
             Weak: true }
a: { Type: Object, Size: 4 }
...
```

IFS Text Format

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3redv: { Type: Func }
_Z4bluev: { Type: Func }
_Z3redi: { Type: Func, Weak: true }
...
```

IFS Text Format



```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3redv: { Type: Func }
_Z4bluev: { Type: Func }
...
```

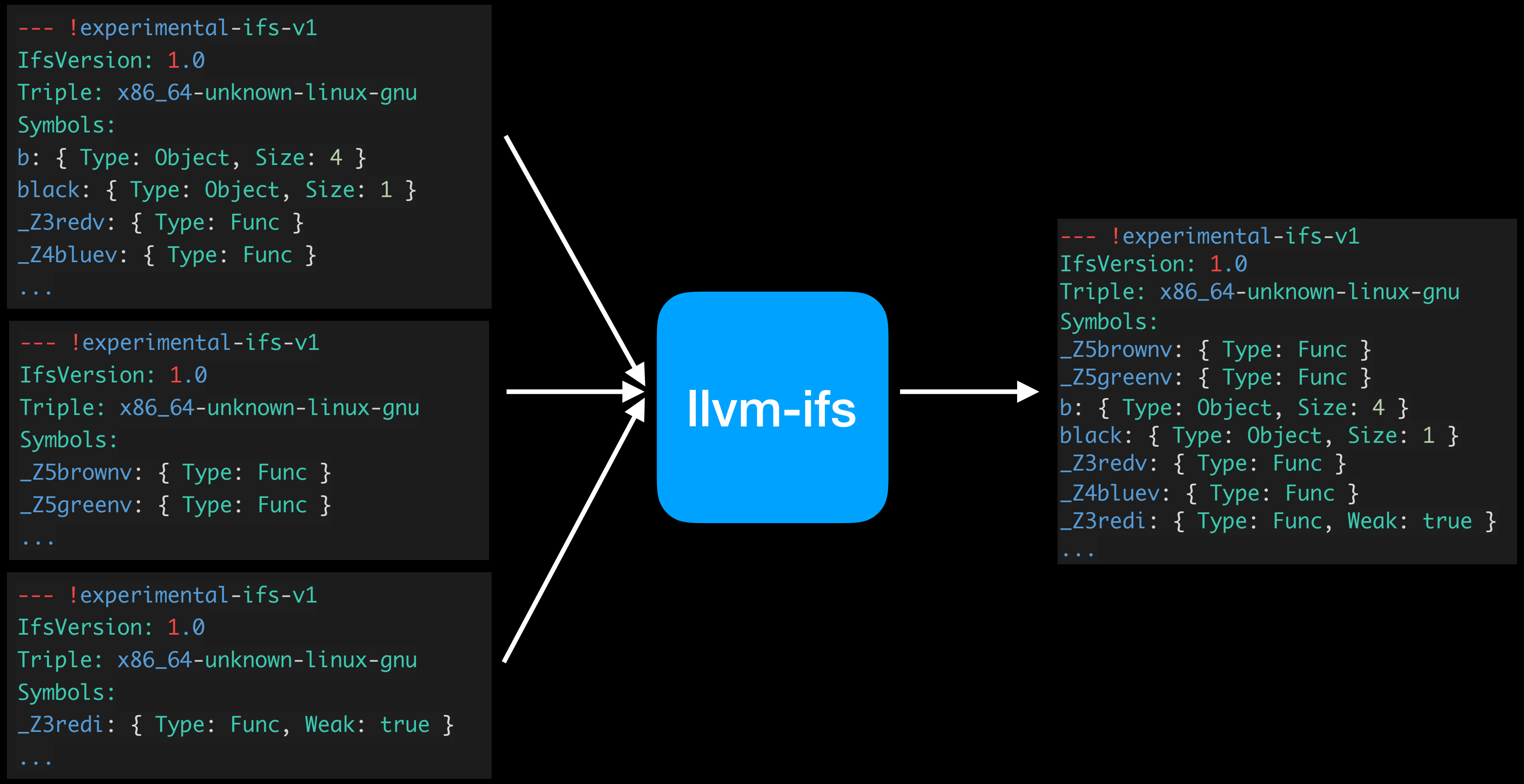
```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redi: { Type: Func, Weak: true }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3redv: { Type: Func }
_Z4bluev: { Type: Func }
_Z3redi: { Type: Func, Weak: true }
...
```

llvm-ifs

- A new llvm tool for consuming IFS files: produces a merged IFS file, or ELF shared object (.dynsym, .dynstr, .symtab only)



```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3redv: { Type: Func }
_Z4bluev: { Type: Func }
...
```

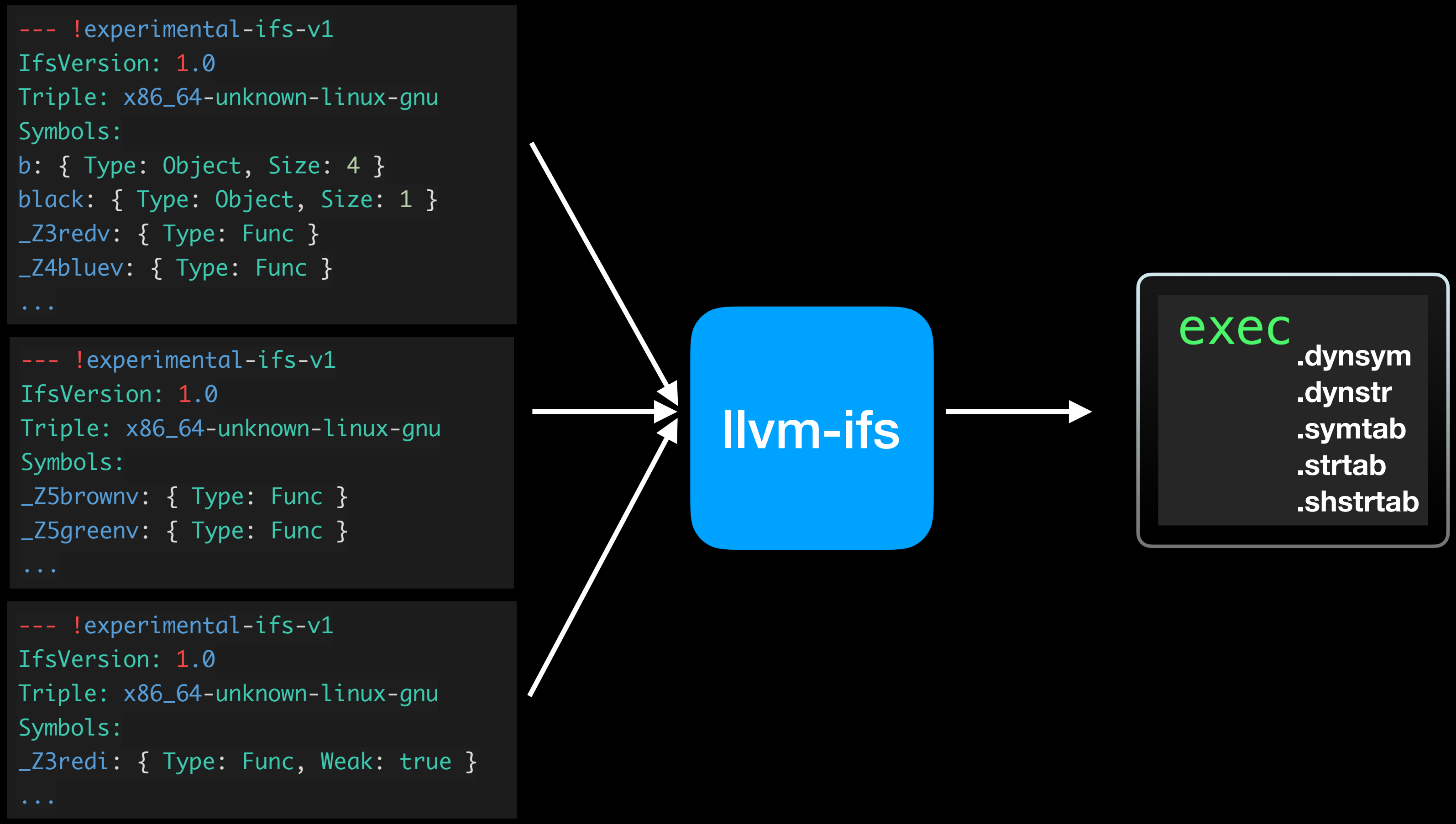
```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redi: { Type: Func, Weak: true }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3redv: { Type: Func }
_Z4bluev: { Type: Func }
_Z3redi: { Type: Func, Weak: true }
...
```

llvm-ifs

- A new llvm tool for consuming IFS files: produces a merged IFS file, or ELF shared object (.dynsym, .dynstr, .symtab only)



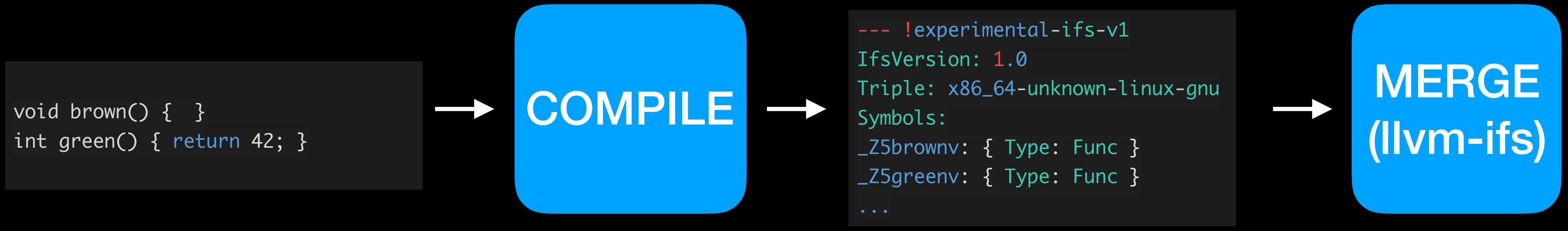
```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3redv: { Type: Func }
_Z4bluev: { Type: Func }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redi: { Type: Func, Weak: true }
...
```

Example

clang -emit-interface-stubs -o libfoo.so a.cpp c.cpp sq.cpp

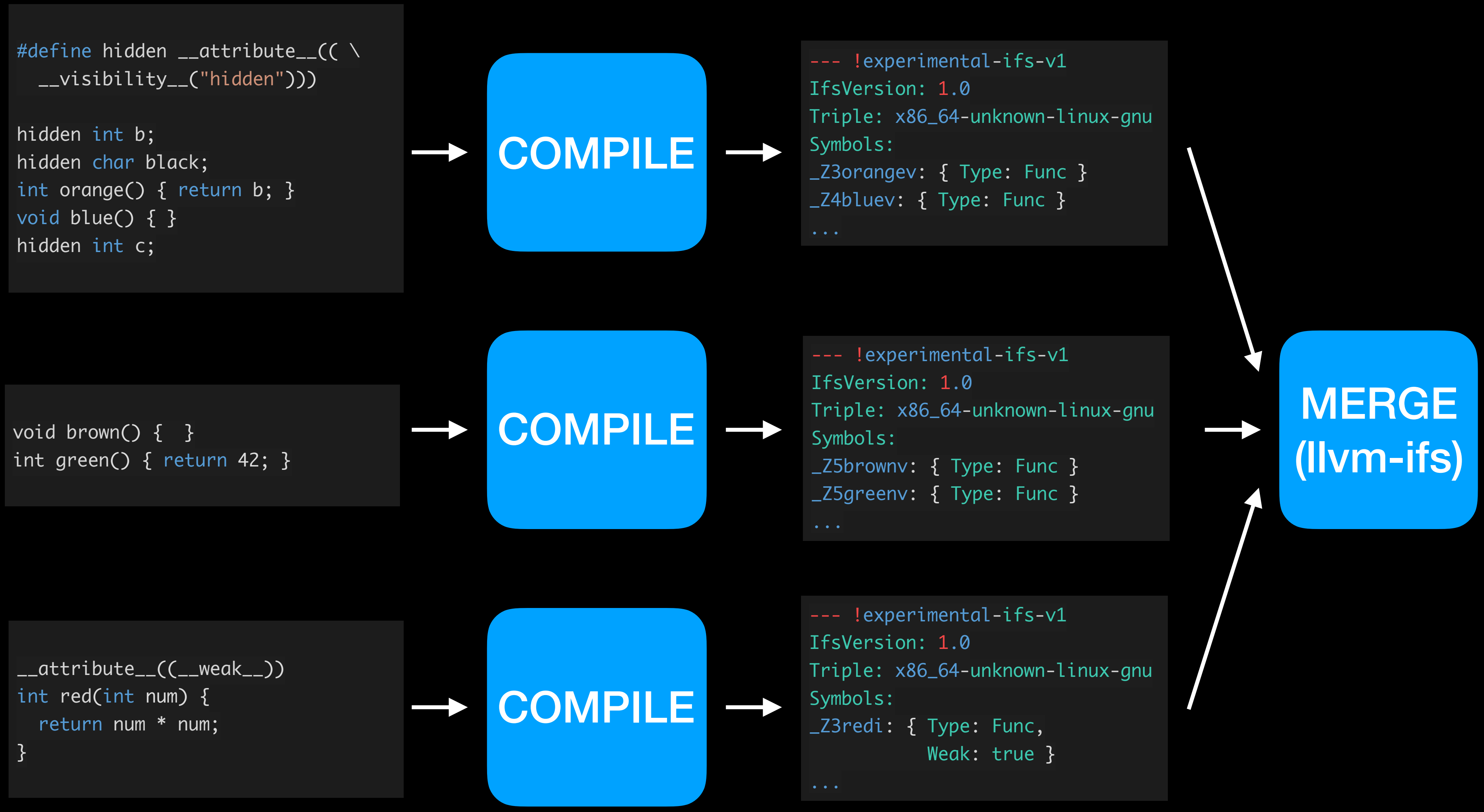


```
void brown() {  }
int green() { return 42; }
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

Example

clang -emit-interface-stubs -o libfoo.so a.cpp c.cpp sq.cpp



```
#define hidden __attribute__(( \
  __visibility__("hidden")))

hidden int b;
hidden char black;
int orange() { return b; }
void blue() { }
hidden int c;
```

```
void brown() {  }
int green() { return 42; }
```

```
__attribute__((__weak__))
int red(int num) {
  return num * num;
}
```

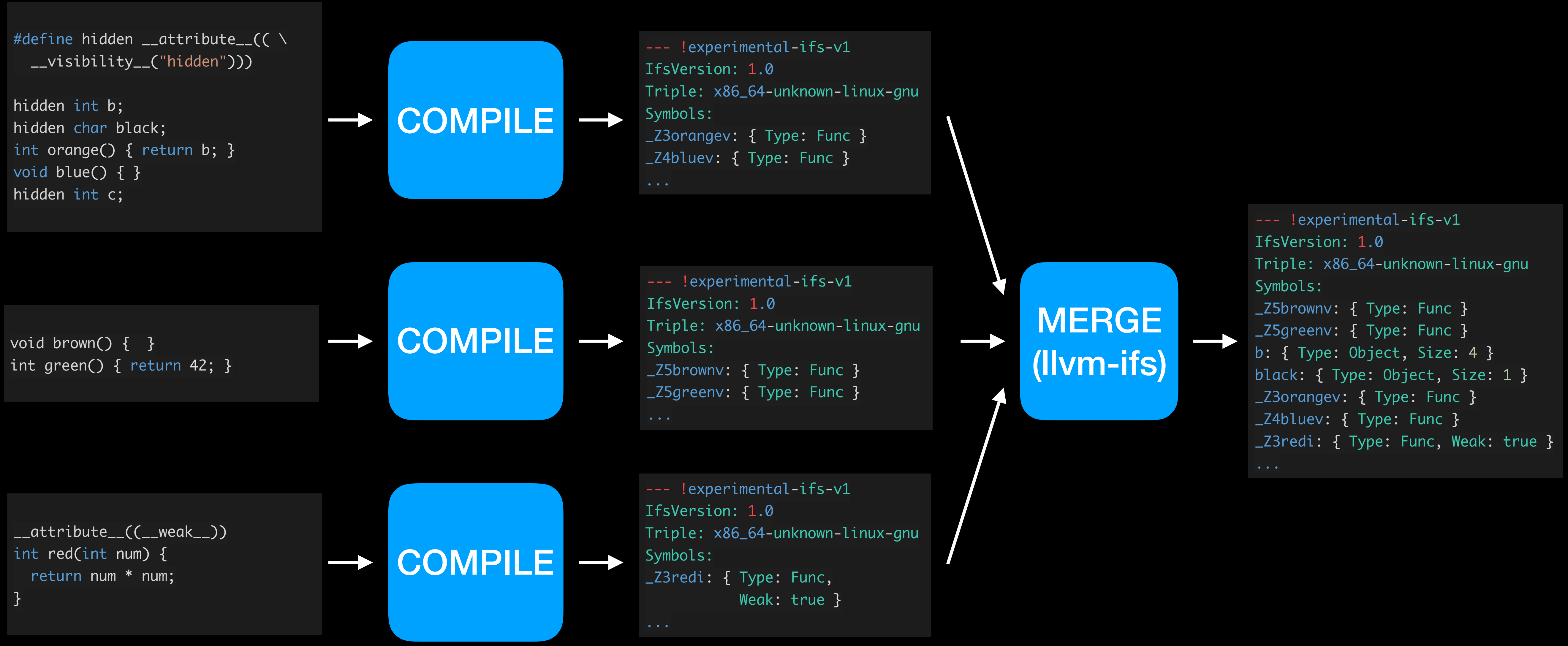
```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3orangev: { Type: Func }
_Z4bluev: { Type: Func }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redi: { Type: Func,
           Weak: true }
...
```

Example

clang -emit-interface-stubs -o libfoo.so a.cpp c.cpp sq.cpp



```
#define hidden __attribute__(( \
  __visibility__("hidden")))

hidden int b;
hidden char black;
int orange() { return b; }
void blue() { }
hidden int c;
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3orangev: { Type: Func }
_Z4bluev: { Type: Func }
...
```

```
void brown() {  }
int green() { return 42; }
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

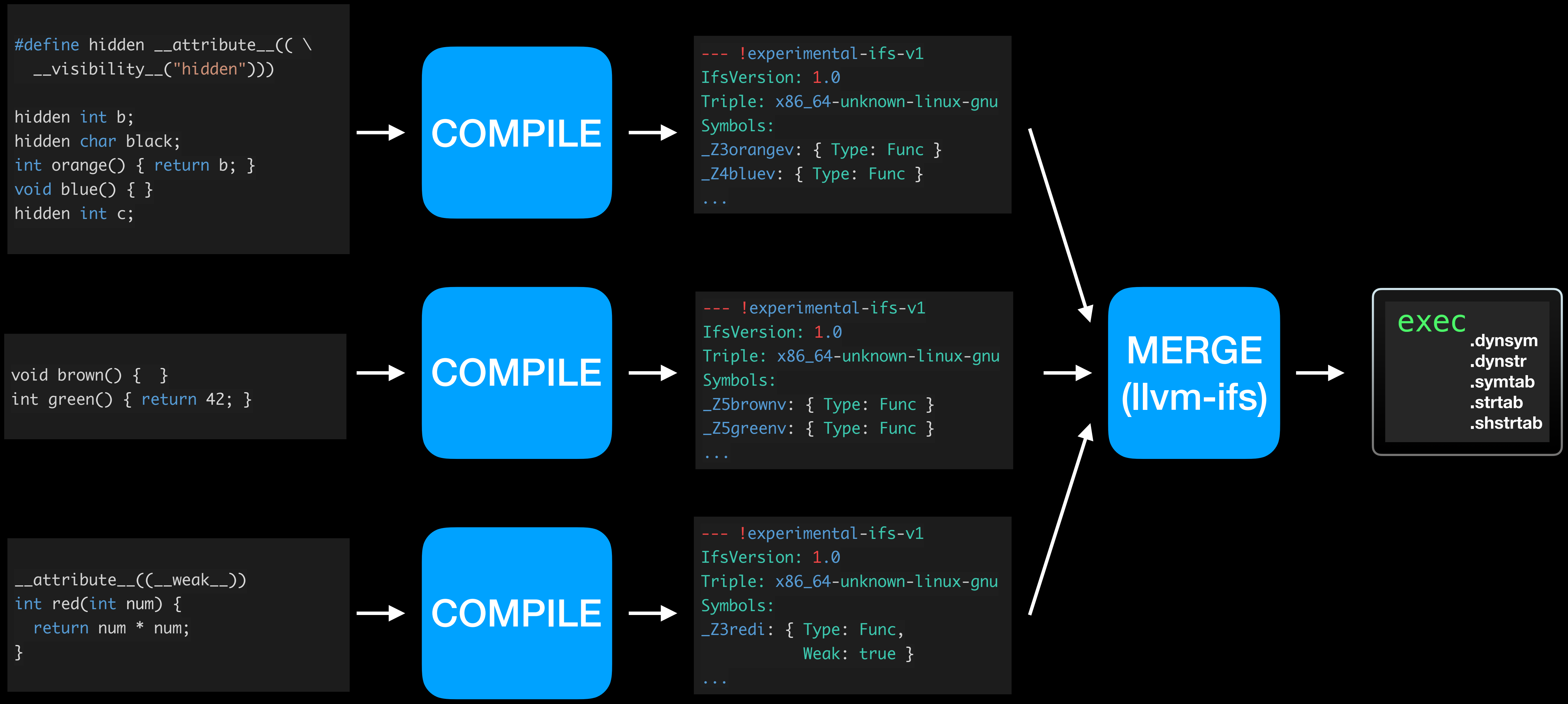
```
__attribute__((__weak__))
int red(int num) {
  return num * num;
}
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redi: { Type: Func,
           Weak: true }
...
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
b: { Type: Object, Size: 4 }
black: { Type: Object, Size: 1 }
_Z3orangev: { Type: Func }
_Z4bluev: { Type: Func }
_Z3redi: { Type: Func, Weak: true }
...
```

Example

clang -emit-interface-stubs -o libfoo.so a.cpp c.cpp sq.cpp



```
#define hidden __attribute__(( \
  __visibility__("hidden")))

hidden int b;
hidden char black;
int orange() { return b; }
void blue() { }
hidden int c;
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3orangev: { Type: Func }
_Z4bluev: { Type: Func }
...
```

```
void brown() {  }
int green() { return 42; }
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z5brownv: { Type: Func }
_Z5greenv: { Type: Func }
...
```

```
__attribute__((__weak__))
int red(int num) {
  return num * num;
}
```

```
--- !experimental-ifs-v1
IfsVersion: 1.0
Triple: x86_64-unknown-linux-gnu
Symbols:
_Z3redi: { Type: Func,
           Weak: true }
...
```

Challenges

- Refactoring for alternate pipeline setup required changes in Driver::BuildActions and getCompilationPhases
- Handling corner cases in the driver setup
- Handling Templates, Specializations, and non-trivial decls
- Converging on the text format required many iterations

Future Work

- Hardening IFS ASTConsumer and llvm-ifs
- Generating interface stubs for libc++ builds
- Inline Assembly
- Support for new formats: MS Import Libs, Darwin TAPI