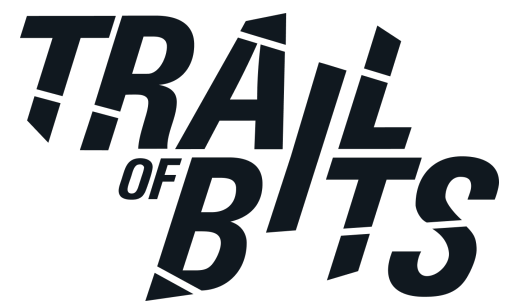


PolyTracker

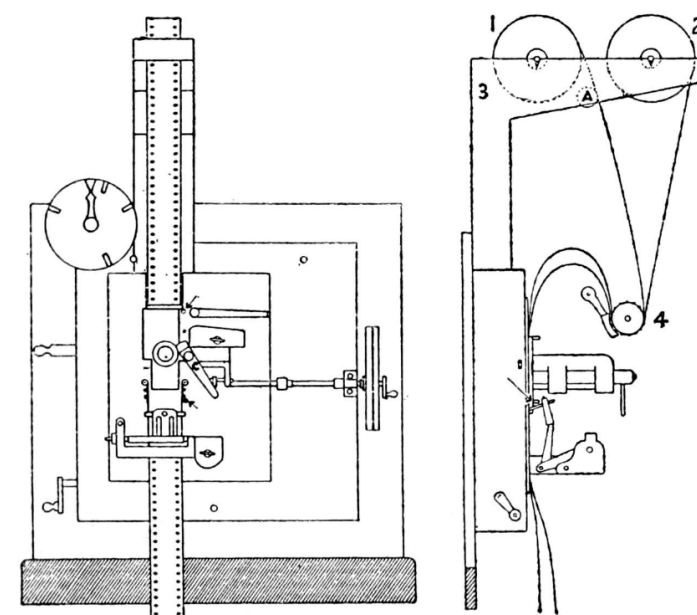
Whole-Input Dynamic Information Flow Tracing

Evan Sultanik, Marek Surovič, Henrik Brodin, **Kelly Kaoudis**,
Facundo Tiesca, Carson Harmon, Lisa Overall,
Joseph Sweeney, Bradford Larsen



PolyTracker

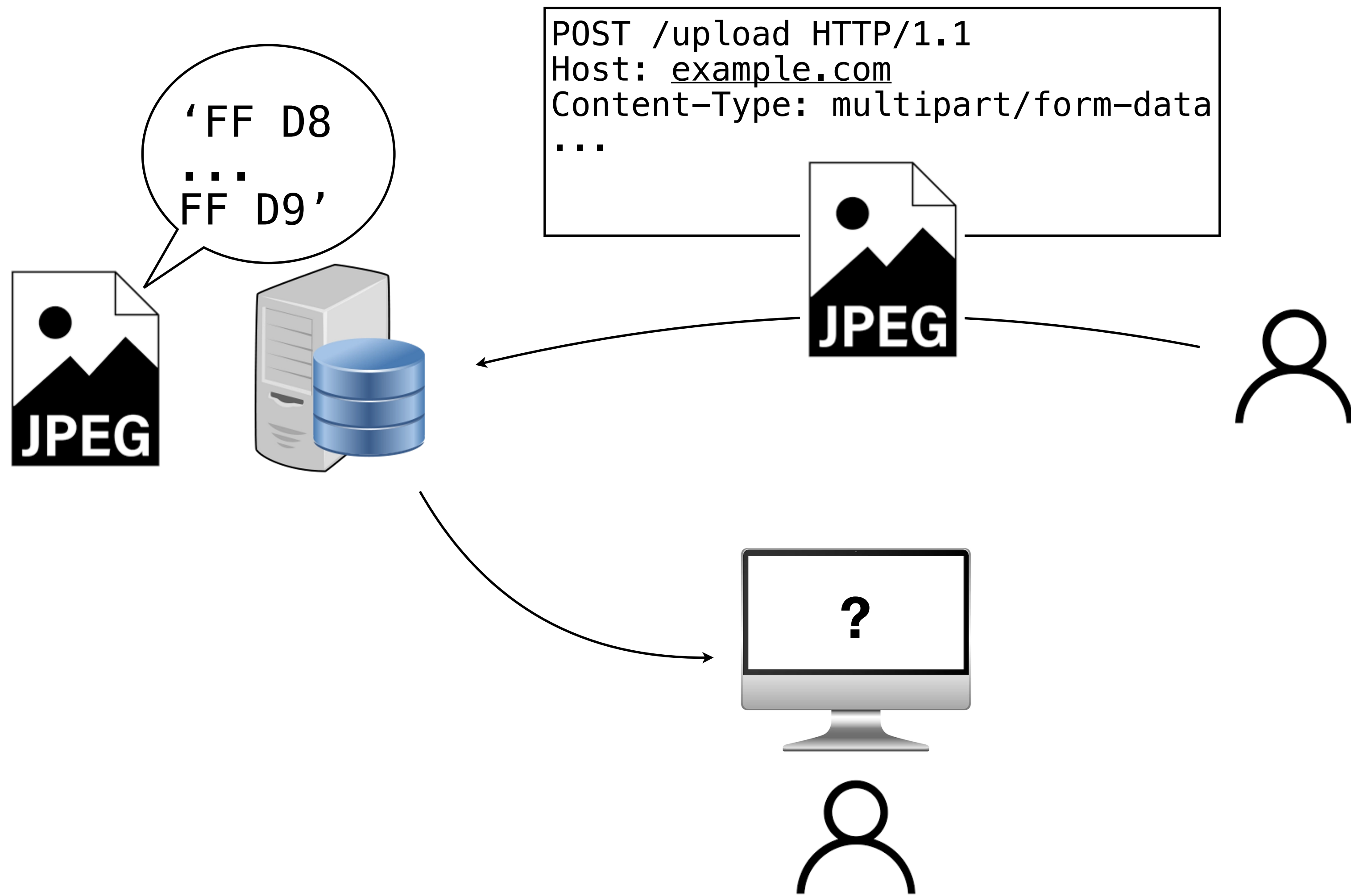
- Runtime (*dynamic* information flow tracing)
- Every input byte
- Record parts of CFG that data flow enters for later call stack reconstruction
- Intermediate labels' provenance relations to source *and* sink bytes
- Novel trace representation: the tainted directed acyclic graph

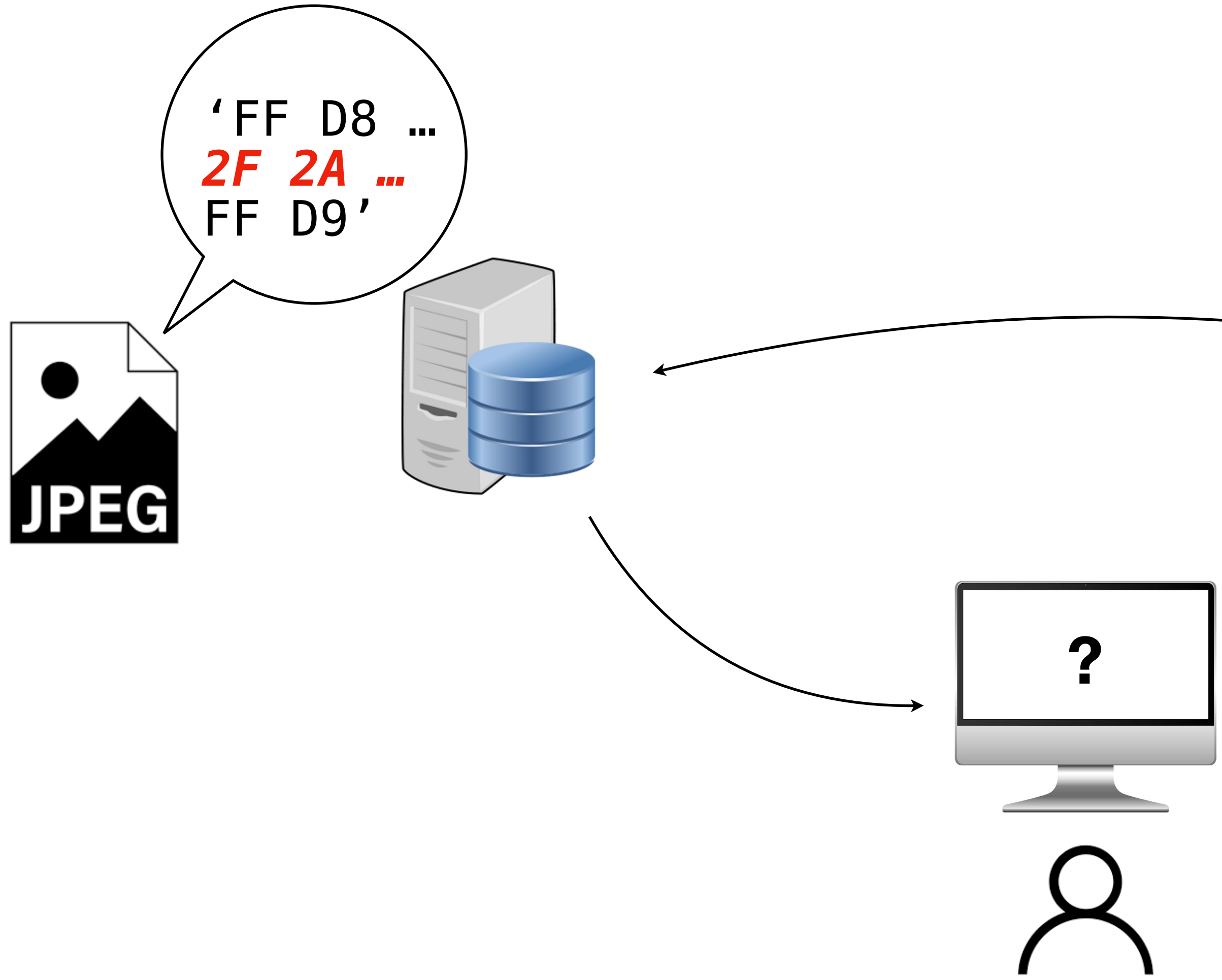






Malicious inputs?





<https://infosecwriteups.com/exploiting-xss-with-javascript-jpeg-polyglot-4cff06f8201a>
https://bugzilla.mozilla.org/show_bug.cgi?id=1288361

```

test.jpg - GHex
File Edit View Windows Help
00000000 ff 08 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 ff e2 01 d4 49 43 43 5f 50 52 4f .....JFIF.....ICC_PRO
0000001f 46 49 4c 45 00 01 01 00 00 01 c4 6c 63 60 73 02 10 00 00 6d 6e 74 72 52 47 42 20 58 59 5a 20 FILE.....lcms.....mnrRGB_XYZ
0000003e 07 e1 00 02 00 01 00 12 00 01 00 00 61 63 73 70 41 50 50 4c 00 00 00 00 00 00 00 00 00 00 .....acspAPPL.....
0000005d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....lcms..
0000007c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....desc.....wpt
0000009a 00 00 01 20 00 00 00 00 00 00 00 00 09 64 65 73 63 00 00 00 00 00 00 00 00 00 00 00 00 .....bkpt...4...XYZ...H...
000000d9 14 67 58 59 5a 00 00 01 5c 00 00 00 14 62 58 59 5a 00 00 01 70 00 00 00 14 72 54 52 43 00 00 .....pXYZ.....bXYZ...p...rTRC...
000000f8 01 84 00 00 00 40 67 54 52 43 00 00 01 84 00 00 00 40 62 54 52 43 00 00 01 84 00 00 00 40 64 .....@gTRC.....@bTRC.....@d
00000117 65 73 63 00 00 00 00 00 00 14 73 52 47 42 20 49 6d 61 67 65 4f 70 74 69 6d 2e 63 6f 6d 00 esc.....sRGB ImageOptim.com.
00000136 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 58 59 5a 20 00 00 00 00 00 00 00 00 .....XYZ.....XYZ
00000155 00 00 00 d3 20 58 59 5a 20 00 00 00 00 00 00 03 15 00 00 03 33 00 00 02 a4 58 59 5a 20 00 00 .....XYZ.....3...XYZ
00000174 00 00 00 00 6f a2 00 00 38 f5 00 00 03 90 58 59 5a 20 00 00 00 00 00 62 99 00 00 00 b7 85 00 .....@...8...XYZ.....b...
00000193 00 18 da 58 59 5a 20 00 00 00 00 24 a0 00 00 0f 84 00 00 06 cf 63 75 72 76 00 00 00 00 .....XYZ.....$.....curv...
000001b2 00 00 00 1a 00 00 00 c8 01 c9 03 63 05 92 08 68 08 f6 10 3f 15 51 1b 34 21 f1 29 90 32 18 38 .....c...k...?Q4!...)2;
000001d1 92 46 05 51 77 5d ed 68 70 7a 05 89 b1 9a 7c ac 69 bf 7d d3 c3 e9 30 ff ff ff db 00 84 00 06 .F.Qw].kpz...|.i)...0
00001f0 06 06 06 06 07 07 07 09 0a 09 0a 09 0e 0c 0b 0b 0c 0e 15 0f 10 0f 10 0f 15 1f 13 17 13 .....!2''''290-09E>>EWSMrr
000020f 13 17 13 1f 1c 21 1b 19 1b 21 1c 32 27 22 22 27 32 39 3b 20 38 39 45 3e 45 57 53 57 72 72 .....!...!2''''290-09E>>EWSMrr
000022e 99 01 08 89 89 8a 0a 0a 0f 10 10 0f 14 16 14 1e 1b 19 19 1b 1e 20 20 23 20 23 20 2d 44 .....# # #D
000024d 2b 32 2b 2b 32 2b 44 3c 49 3c 38 3c 49 3c 6d 55 4c 4c 55 6d 7e 6a 64 6a 7e 98 88 88 98 c0 b6 +--+2+d-I<8-I<mULLUm-ldl-.....
  
```

Research questions

Differences when handling benign and malicious input?

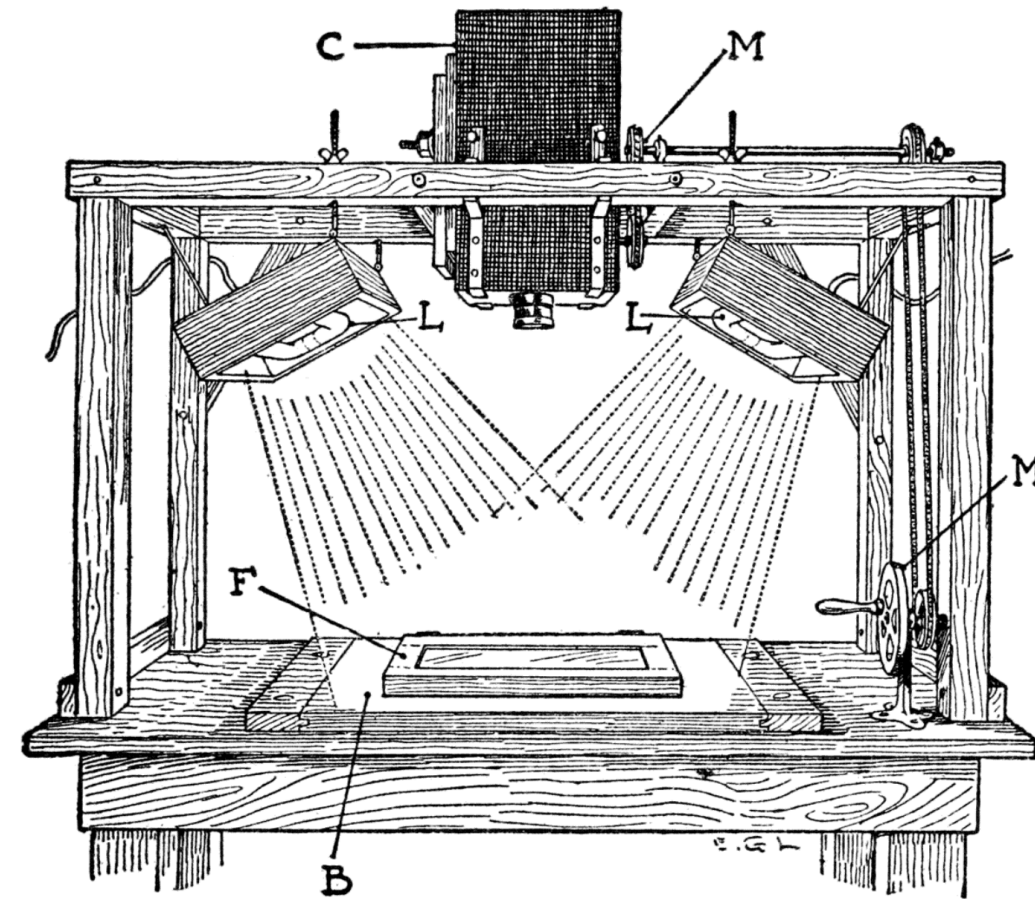
What program logic fails on unexpected input?

What differs between parsers that *should* implement the same input specification?

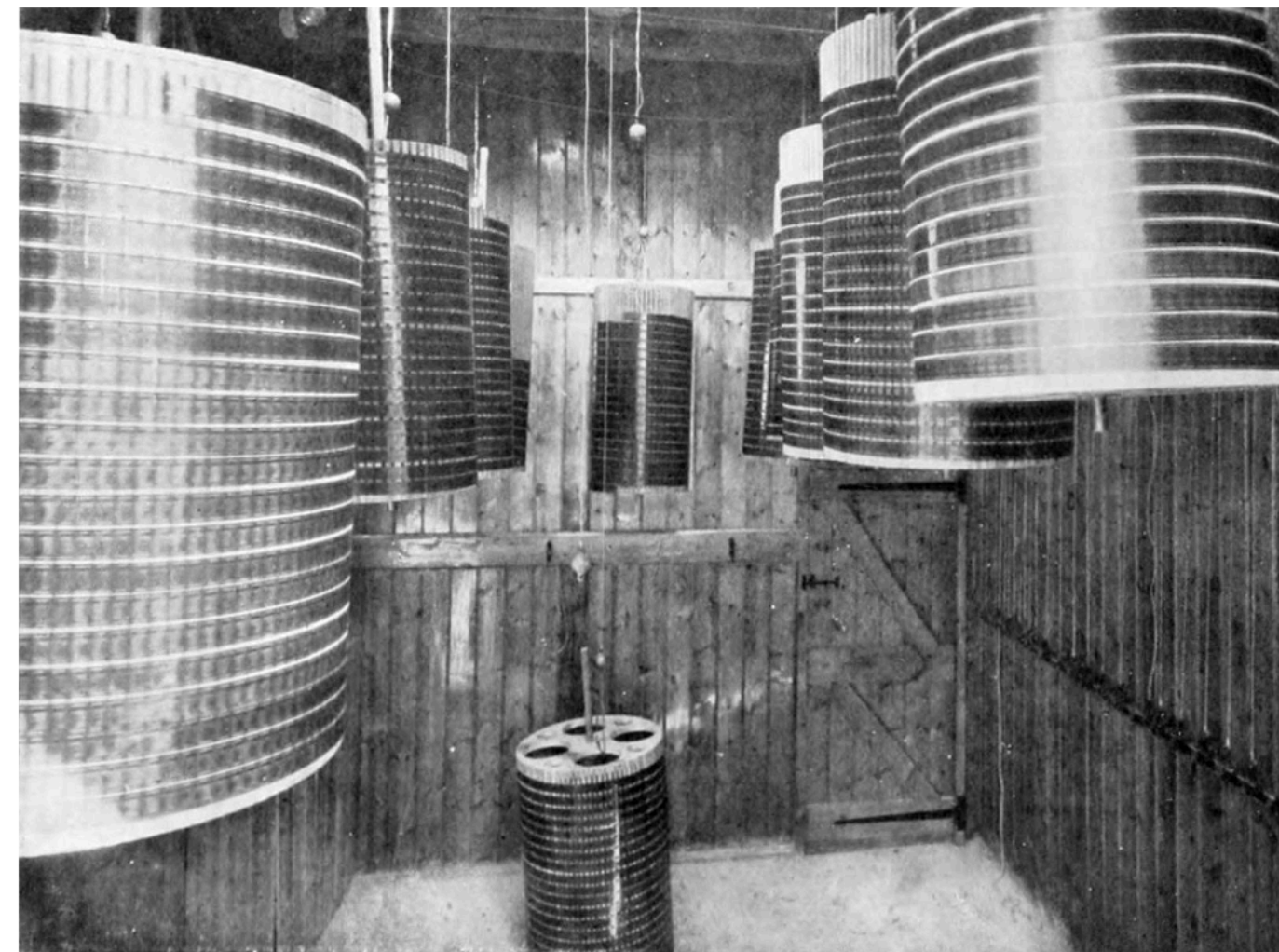
What parts of a specification enable creating polyglot files?

Format *ground truth*?

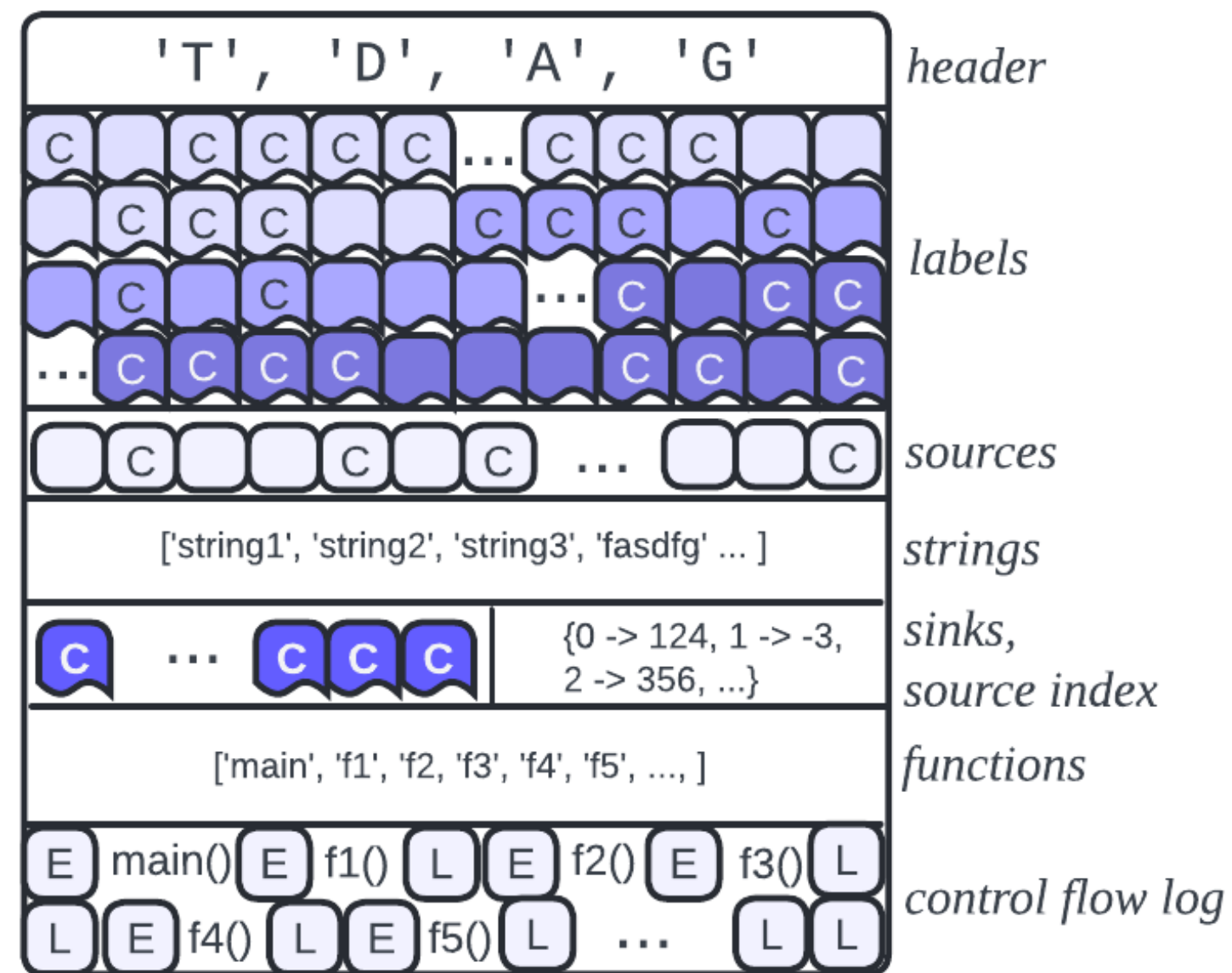
Challenge: manual instrumentation doesn't scale



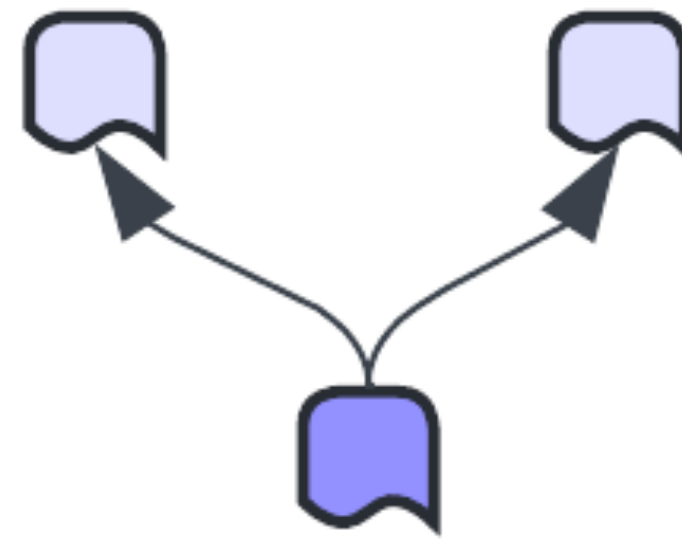
Challenge: *much* data



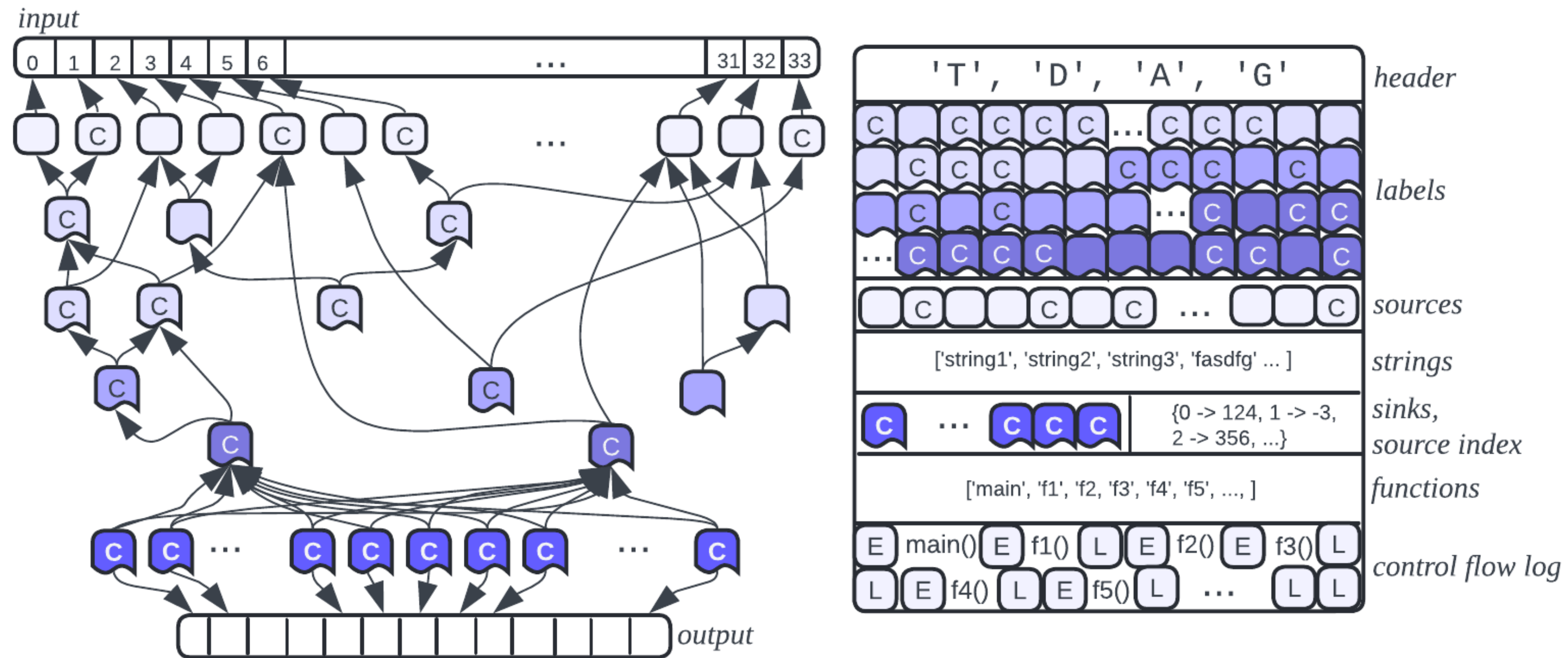
Tainted Directed Acyclic Graph (TDAG)



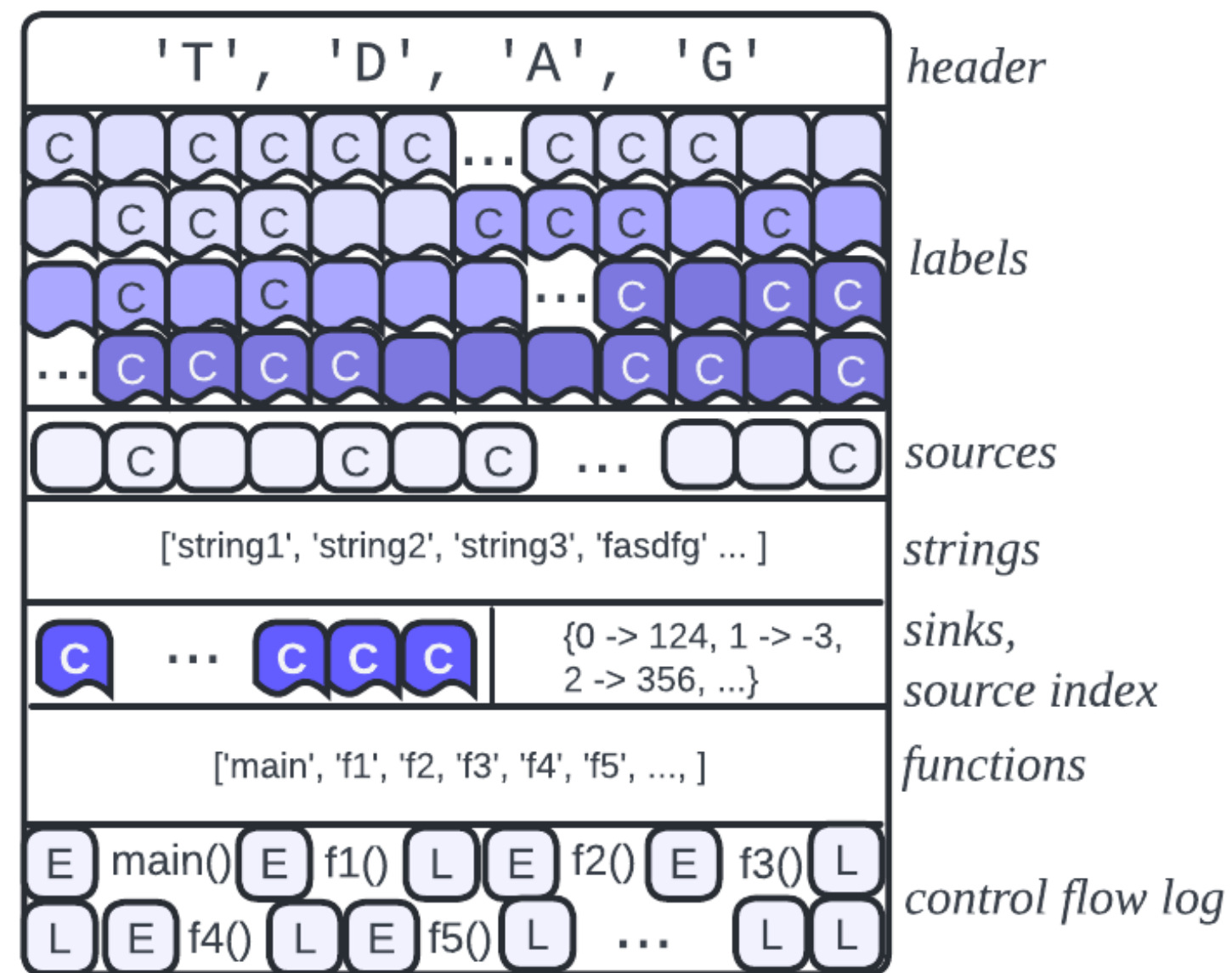
Tainted Directed Acyclic Graph (TDAG)

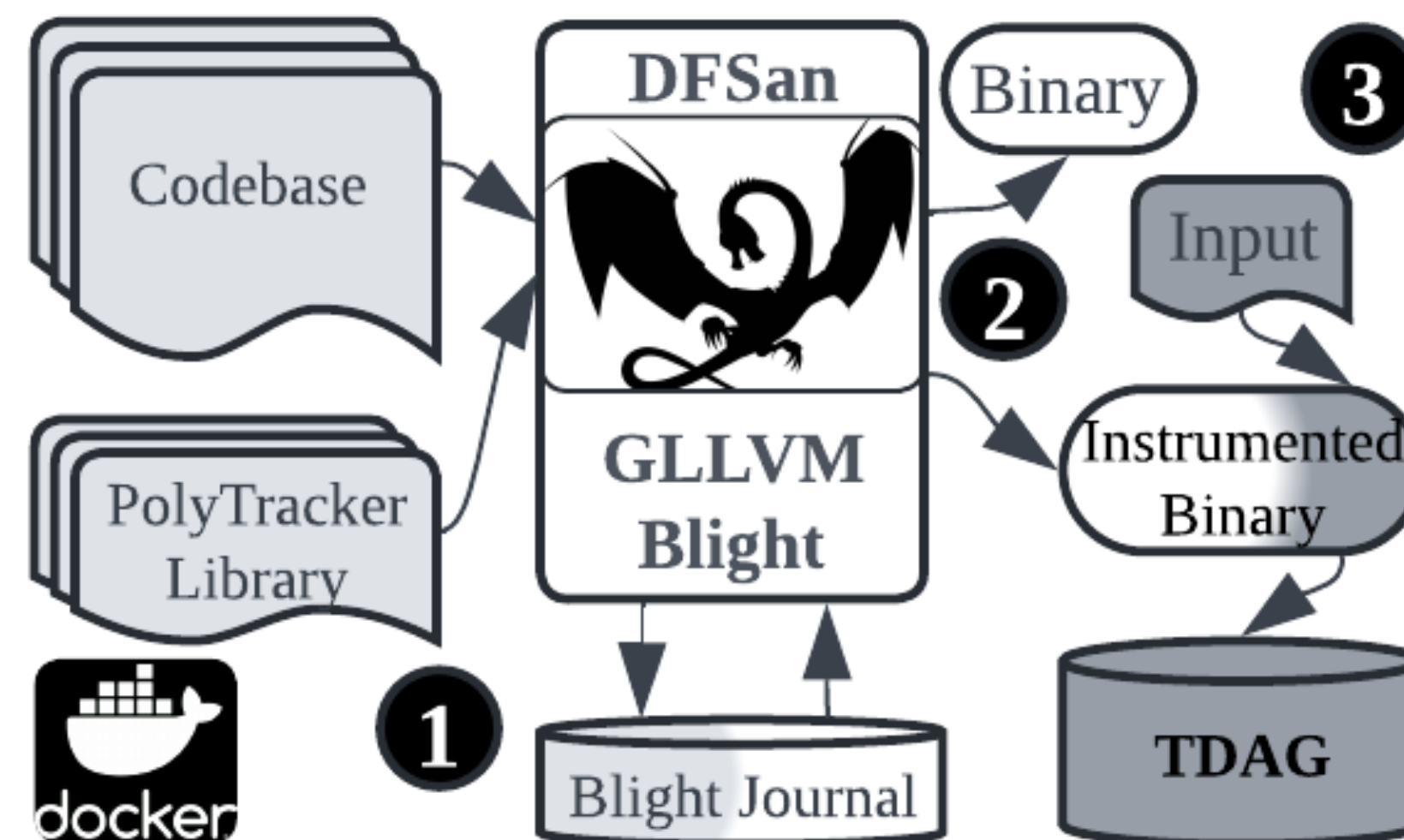


Tainted Directed Acyclic Graph (TDAG)



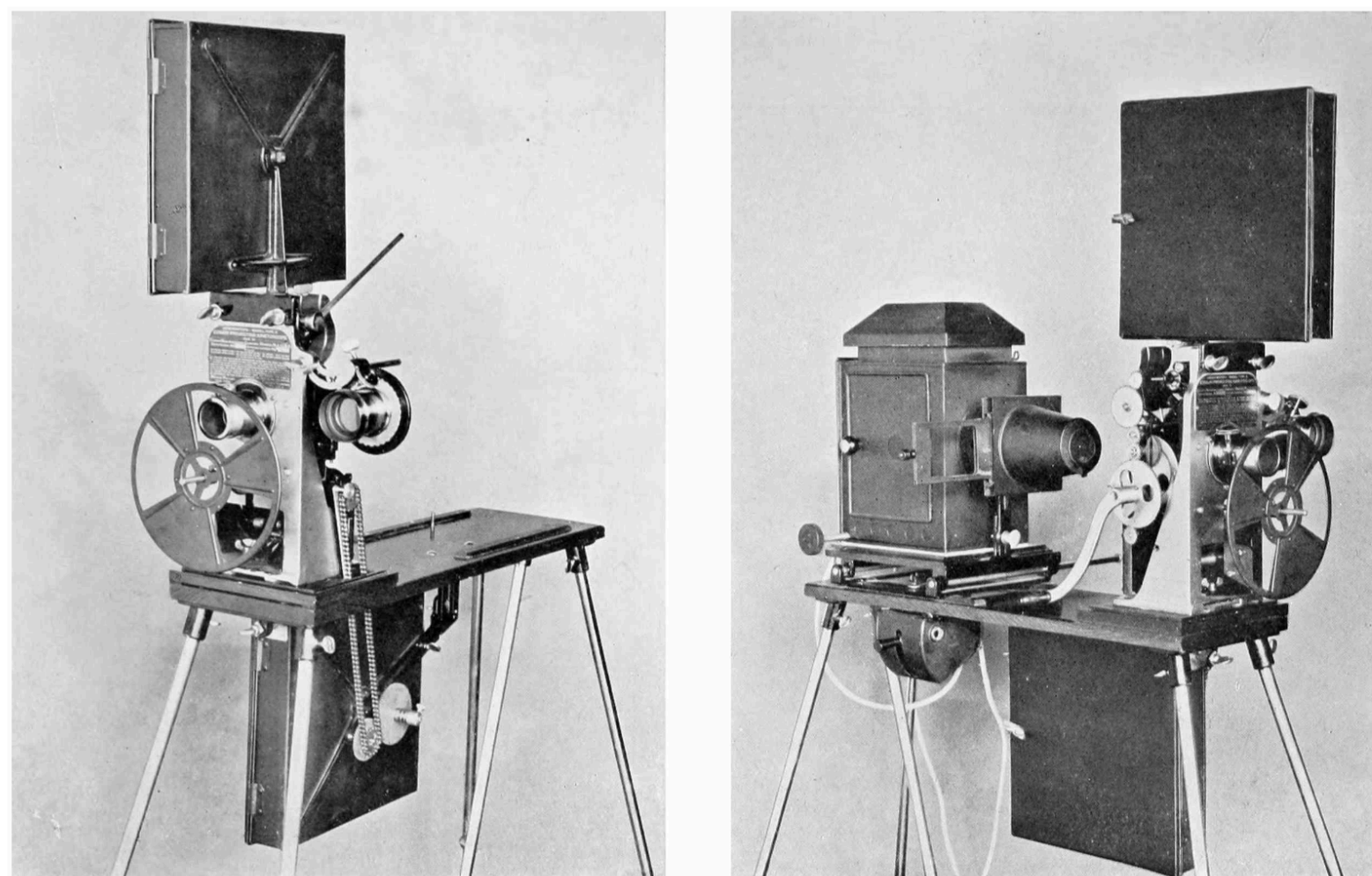
Tainted Directed Acyclic Graph (TDAG)





Write side implementation

```
root@a60fd5d03bf3:/polytracker/the_klondike/poppler# ls
AUTHORS          README-XPDF      config.h.cmake  hooks           poppler          qt6
CMakeLists.txt  README.contributors  cpp            make-glib-api-docs  poppler-cpp.pc.cmake  regtest
COPYING         README.md        do-the-gnupg-2.4-dance.sh  pdftops.instrumented  poppler-glib.pc.cmake  splash
COPYING3       _clang-format   fofi          pdftops.instrumented.bc  poppler-qt5.pc.cmake  test
ConfigureChecks.cmake  blight_journal.jsonl  glib         pdftotext.instrumented  poppler-qt6.pc.cmake  utils
INSTALL        build           goo          pdftotext.instrumented.bc  poppler.pc.cmake
NEWS          cmake          gtkdoc.py    polytracker.tdag        qt5
root@a60fd5d03bf3:/polytracker/the_klondike/poppler#
```

ISSTA Tool Demonstrations | 18 September 2024 | kelly.kaoudis@trailofbits.com

<https://github.com/trailofbits/polytracker>

**TRAIL
OF
BITS**