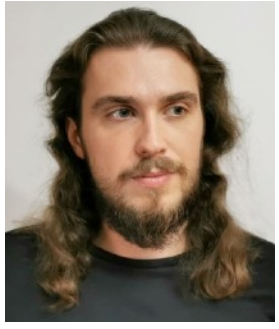
Smart Greybox Fuzzing (TSE'19)

ICSE 2020 - Journal First Track

Thuan Pham

Marcel Böhme

Andrew E. Santosa



Alexandru R. C.

Abhik Roychoudhury

THE UNIVERSITY OF MELBOURNE

MONASH University

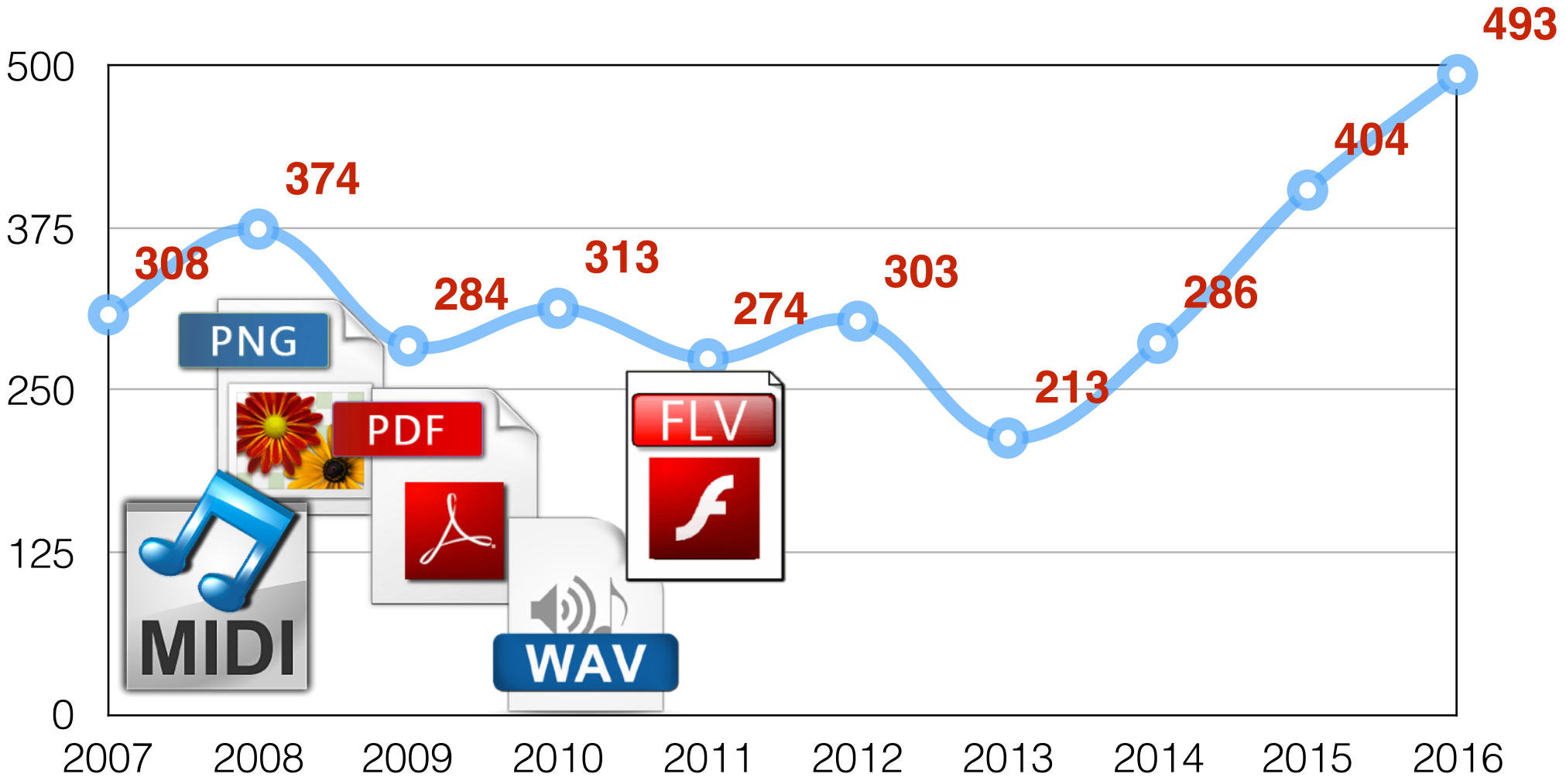
VERACODE

UiPath™

NUS National University of Singapore

Structure-Aware Fuzzing For *Chunk-based Formats*

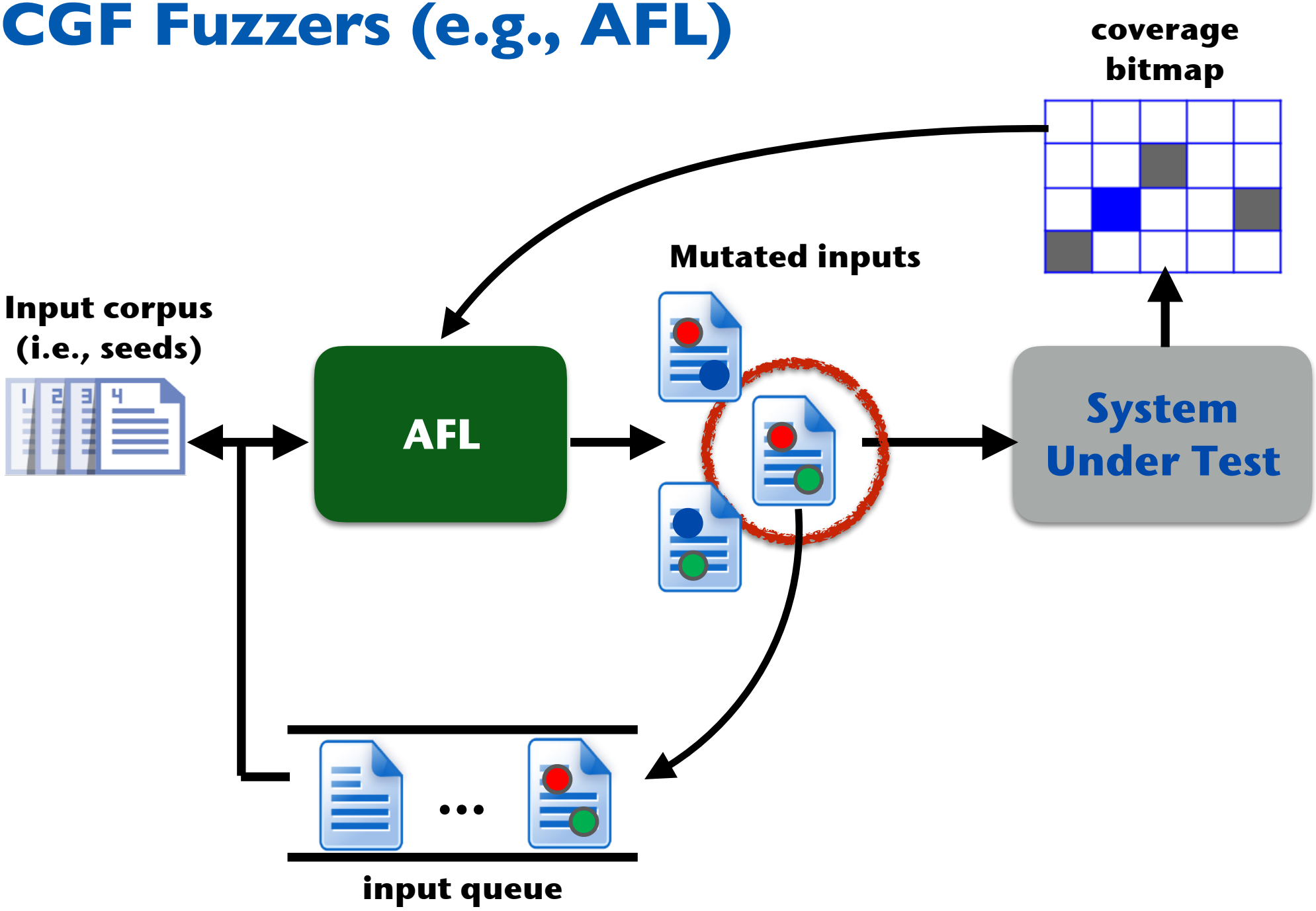
#CVE-assigned vulnerabilities in chunk-based file-processing programs by year



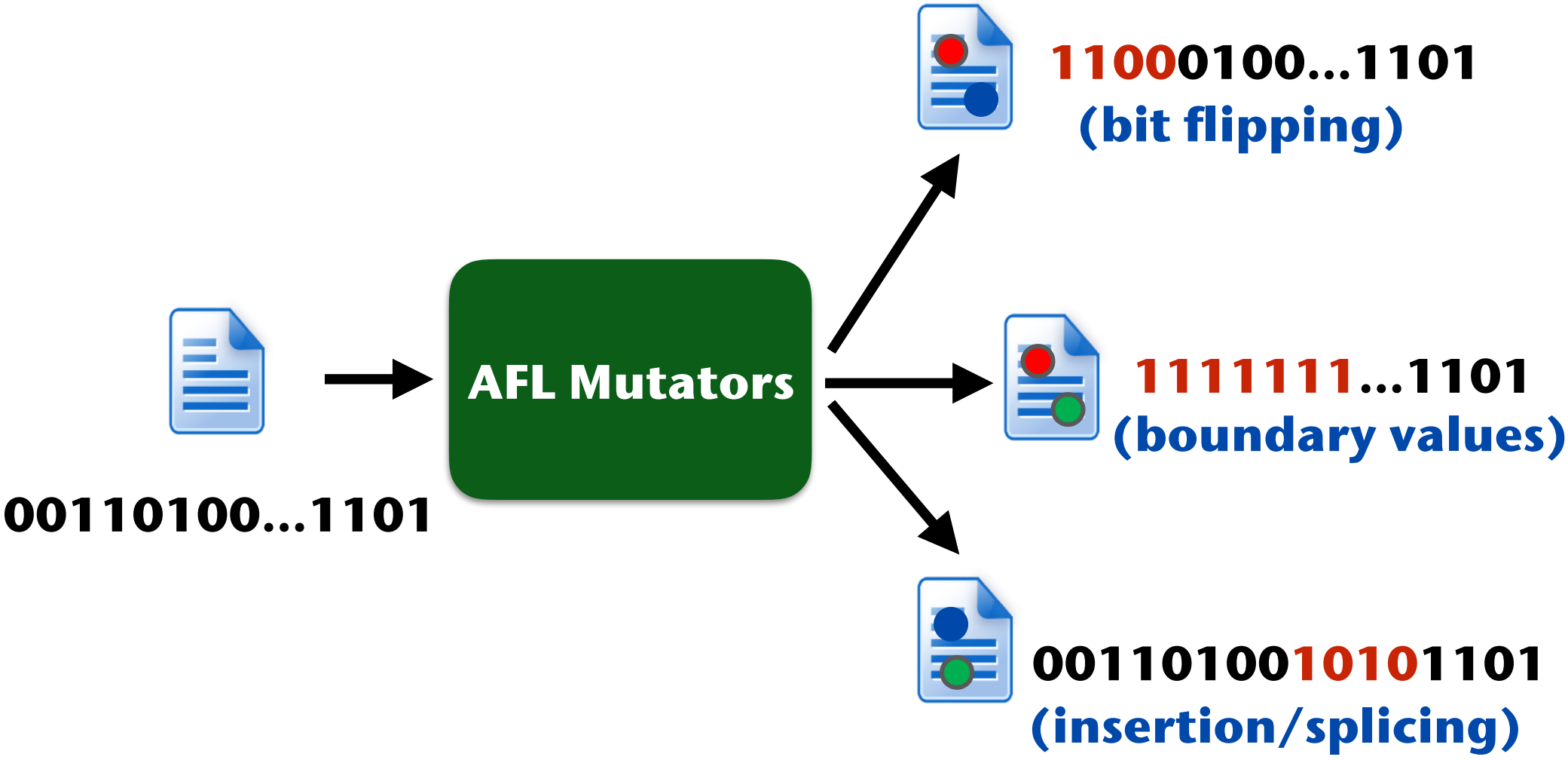
Outline

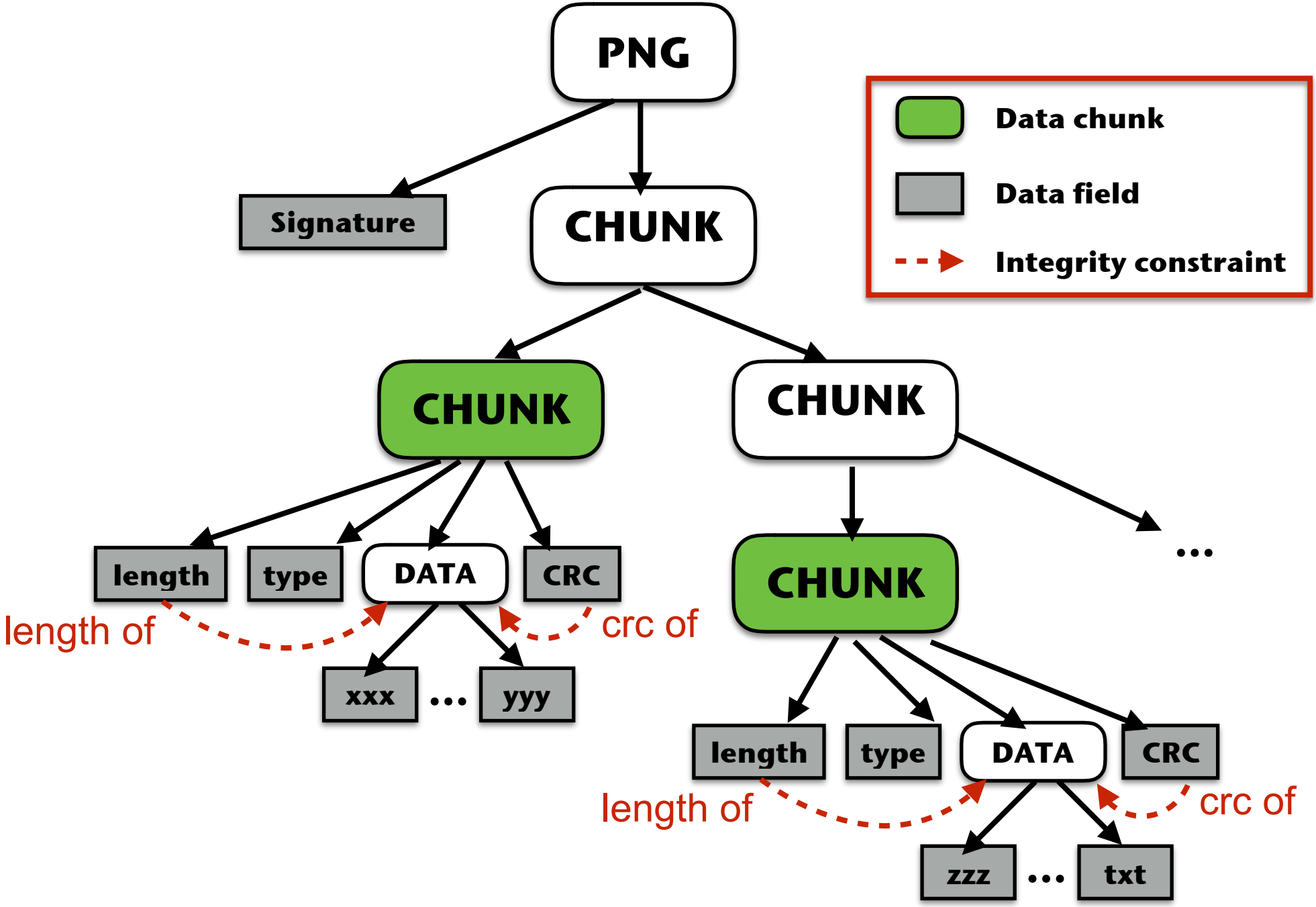
- Limitation of vanilla coverage-guided greybox fuzzing (CGF) fuzzers (e.g., American Fuzzy Lop - AFL) in handling chunk-based file formats
- AFLSmart - Smart Greybox Fuzzing
- Experimental results

CGF Fuzzers (e.g., AFL)



Limitation


AFL Mutators
00110100...1101
11000100...1101
(bit flipping)
1111111...1101
(boundary values)
0011010010101101
(insertion/splicing)


PNG
Signature
CHUNK
Data chunk
Data field
Integrity constraint
CHUNK
CHUNK
length
type
DATA
CRC
length of
crc of
xxx
...
yyy
CHUNK
...
length
type
DATA
CRC
length of
crc of
zzz
...
txt

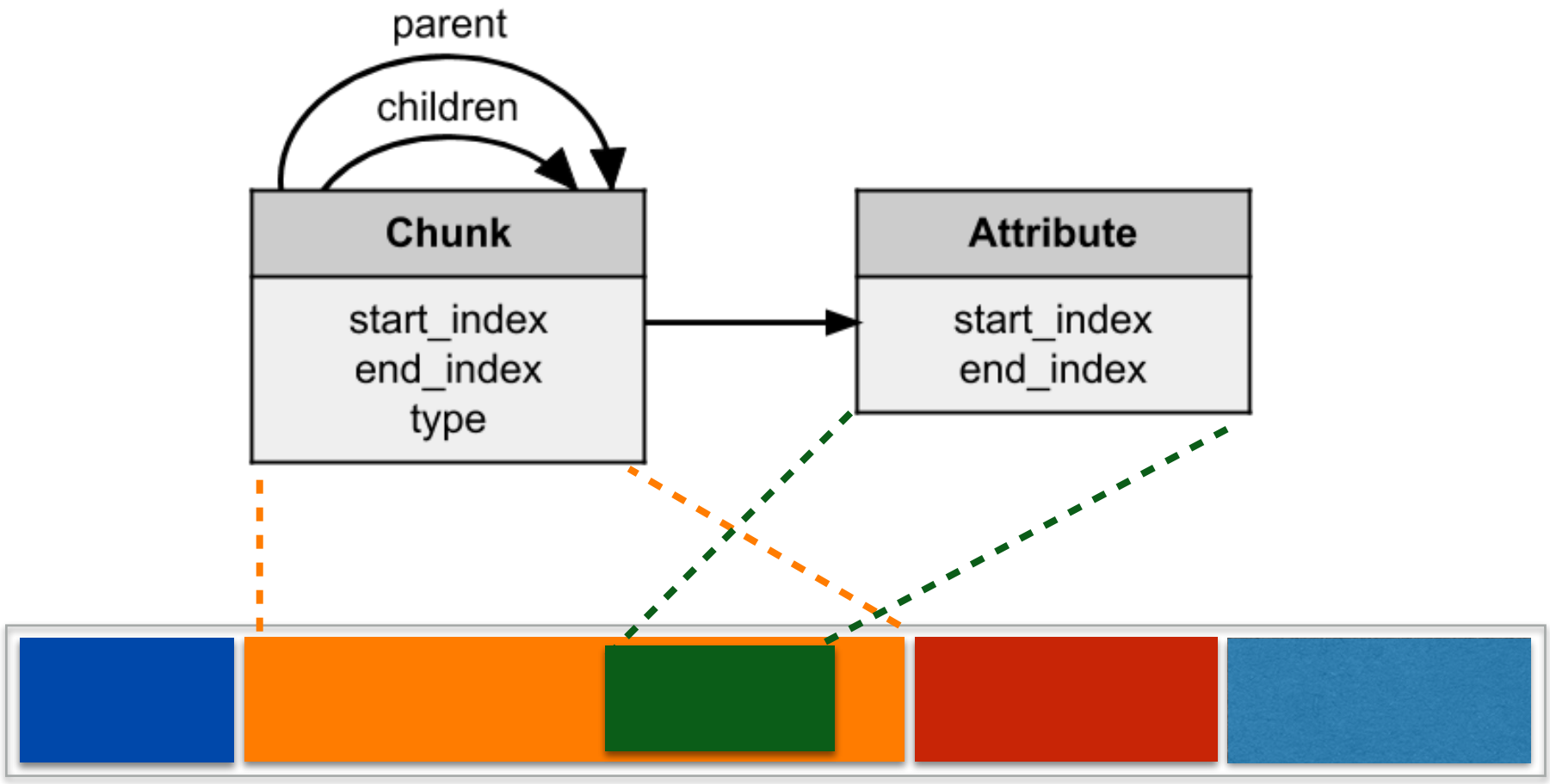
Smart Greybox Fuzzing

Make Greybox Fuzzing input-structure aware by

1. Changing the input *representation*
   - Use tree-like representation instead of bit string
2. Adding new *mutation operators*
   - working at chunk level (e.g., chunk deletion, insertion and splicing)
3. Prioritizing *more valid seed* inputs
   - More valid seeds are assigned higher fuzzing "energy"
4. Applying *optimisations* to retain fuzzing efficiency

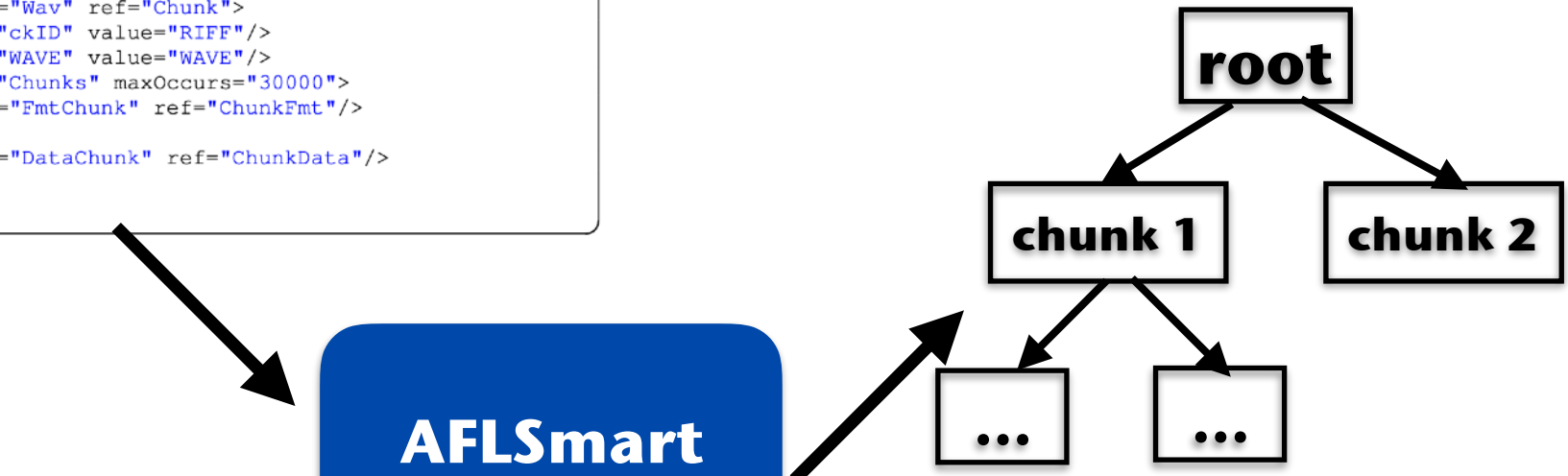
High-level structural representation

virtual file structure



```
<DataModel name="Chunk">
  <String name="ckID" length="4"/>
  <Number name="cksize" size="32" >
    <Relation type="size" of="Data"/>
  </Number>
  <Blob name="Data"/>
  <Padding alignment="16"/>
</DataModel>
<DataModel name="ChunkFmt" ref="Chunk">
   <String name="ckID" value="fmt "/>
   <Block name="Data">
      <Number name="wFormatTag" size="16"/>
      <Number name="nChannels" size="16"/>
      <Number name="nSampleRate" size="32"/>
      <Number name="nAvgBytesPerSec" size="32"/>
      <Number name="nBlockAlign" size="16" />
      <Number name="nBitsPerSample" size="16"/>
   </Block>
</DataModel>
...
<DataModel name="Wav" ref="Chunk">
  <String name="ckID" value="RIFF"/>
  <String name="WAVE" value="WAVE"/>
  <Choice name="Chunks" maxOccurs="30000">
    <Block name="FmtChunk" ref="ChunkFmt"/>
    ...
    <Block name="DataChunk" ref="ChunkData"/>
  </Choice>
</DataModel>
```

XML-based input model. One input model for each file format. (e.g., Peach pits)



AFLSmart File Cracker

Seed input

validity score (0->100)
100: the whole seed is fully cracked/parsed

Guideline to write input model (see the paper)

Data model for a generic data chunk

```
<DataModel name="Chunk">
    <Number name="Length" size="32" endian="big">
        <Relation type="size" of="Data" />
    </Number>
    <Block name="TypeData">
        <String name="Type" length="4" />
        <Blob name="Data" />
    </Block>
    <Number size="32" endian="big">
        <Fixup class="Crc32Fixup">
            <Param name="ref" value="TypeData"/>
        </Fixup>
    </Number>
</DataModel>
```

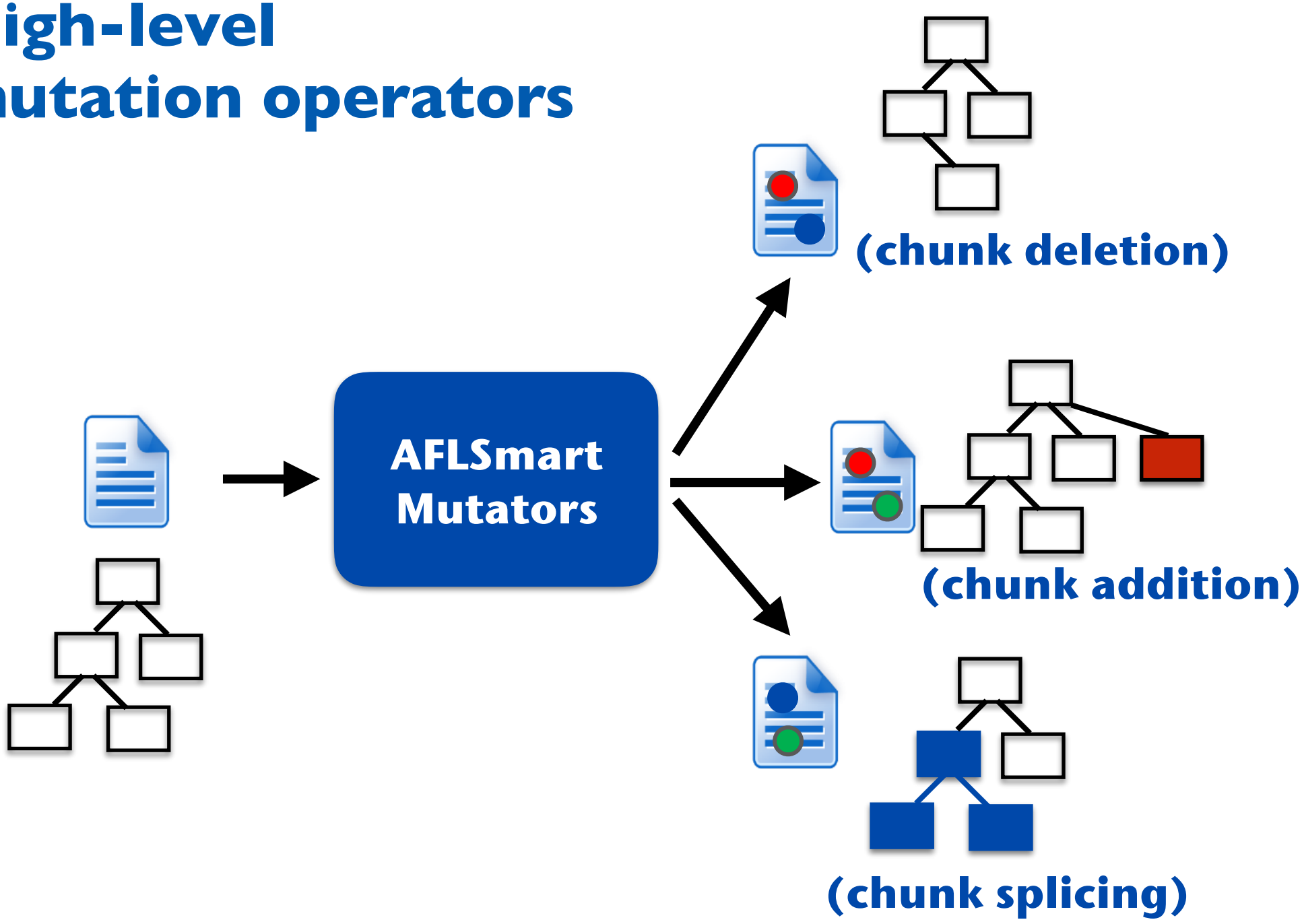
inherits common data fields & relationships

```
<DataModel name="Chunk_IHDR" ref="Chunk">
    <Block name="TypeData">
        <String name="Type" value="IHDR" length="4"
        token="true"/>
        <Block name="Data">
            <Number name="width" size="32" />
            <Number name="height" size="32" />
            <Number name="bits" size="8" />
            <Number name="color_type" size="8" />
            <Number name="compression" size="8" />
            <Number name="filter" size="8" />
            <Number name="interlace" size="8" />
        </Block>
    </Block>
</DataModel>
```

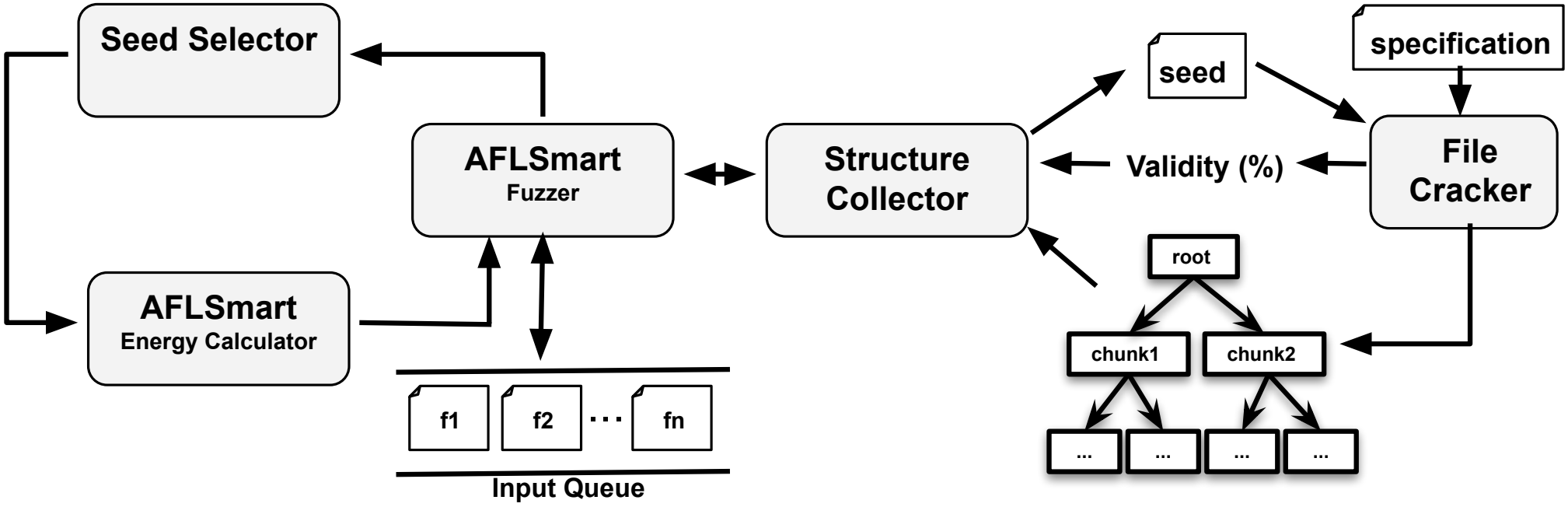
Data model for PNG image files

```
<DataModel name="Png">
    <Number name="Signature" valueType="hex"
    value="89504e470d0a1a0a" size="64"
    token="true" />
    <Choice maxOccurs="30000">
        <Block ref="Chunk_IHDR"/>
        <Block ref="Chunk_PLTE"/>
        <Block ref="Chunk_IDAT"/>
        ...
        <Block ref="Chunk_IEND"/>
        <Block ref="Chunk"/>
    </Choice>
</DataModel>
```

High-level mutation operators



AFLSmart - Architecture



Experimental evaluation

AFLSmart vs AFL/AFLFast, Vuzzer and Peach

15 real-world programs

10 different file formats

Program	Description	Size (LOC)	Test driver	Format
Binutils	Binary analysis utilities	3700 K	`readelf`	**ELF**
Binutils	Binary analysis utilities	3700 K	`nm-new`	**ELF**
LibPNG	Image processing	111 K	`pngimage`	**PNG**
ImageMagick	Image processing	385 K	`magick`	**PNG**
LibJPEG-turbo	Image processing	87 K	`djpeg`	**JPEG**
LibJasper	Image processing	33 K	`imginfo`	**JPEG**
FFmpeg	Video/Audio/Image processing	1100 K	`ffmpeg`	**AVI**
LibAV	Video/Audio/Image processing	670 K	`avconv`	**AVI**
LibAV	Video/Audio/Image processing	670 K	`avconv`	**WAV**
WavPack	Lossless Wave file compressor	47 K	`wavpack`	**WAV**
OpenJPEG	Image processing	115 K	`decompress`	**JP2**
LibJasper	Image processing	33 K	`jasper`	**JP2**
mpg321	Command line MP3 player	5 K	`mpg321`	**MP3**
gif2png+libpng	Image converter	36 K	`gif2png`	**GIF**
pdf2svg+libpoppler	PDF to SVG converter	92 K	`pdf2svg`	**PDF**
tcpdump+libpcap	Network traffic analysis	102 K	`tcpdump`	**PCAP**
tcptrace+libpcap	TCP connection analysis	55 K	`tcptrace`	**PCAP**
djpeg+libjpeg	Image processing	37 K	`djpeg`	**JPEG**

Branch coverage improvement

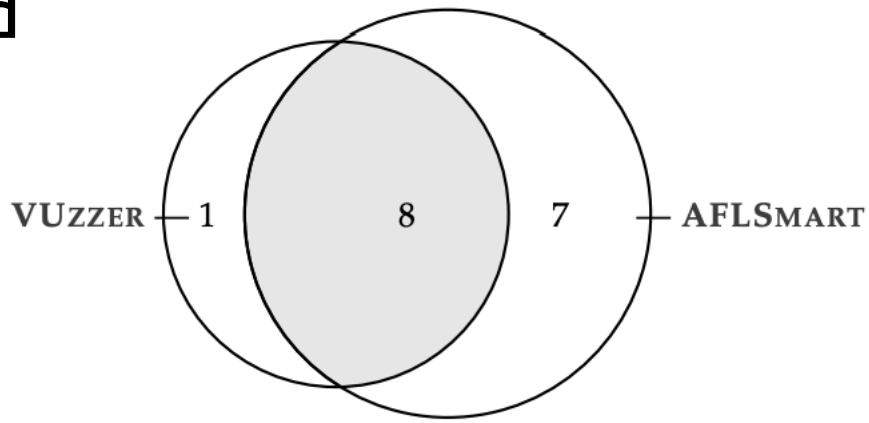
- AFLSmart vs AFL/AFLFast (Vanilla Grey-box Fuzzers)
  - On average: **14.40%**, up to **86.9%**
- AFLSmart vs Peach Fuzzer (Smart Black-box Fuzzer)
  - On average: **133.95%**

Bug finding

- AFLSmart doubled #bugs found

42 zero-day bugs found
23 CVEs assigned

9 CVEs in FFmpeg



AFLSmart vs Vuzzer on Vuzzer's benchmark

aflsmart / aflsmart

Unwatch 42 | Star 317 | Fork 61

Code | Issues 4 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights

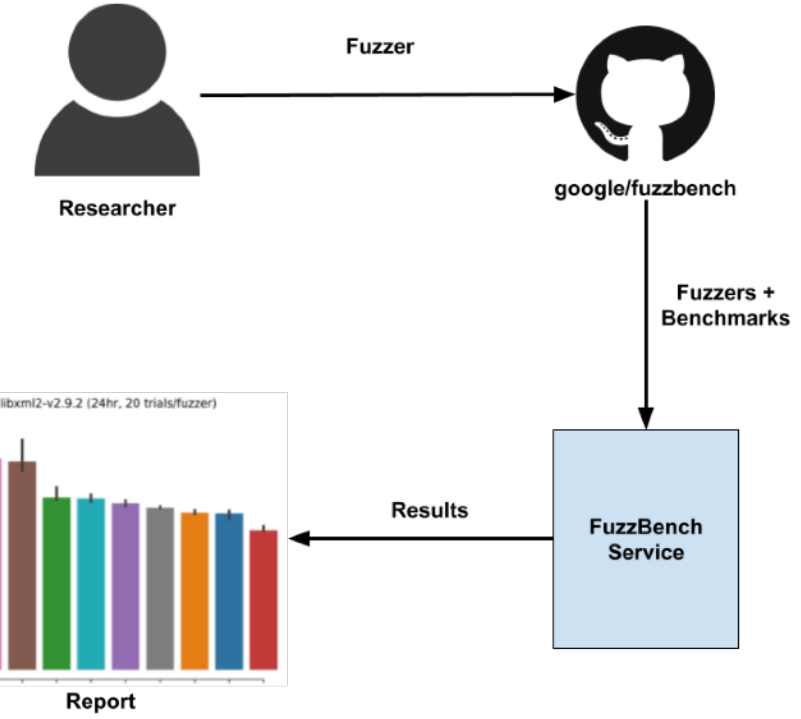
Smart Greybox Fuzzing (https://thuanpv.github.io/publications/TSE19_aflsmart.pdf)

247 commits | 1 branch | 0 packages | 0 releases | 9 contributors

Hot fuzz: Bug detectives whip up smarter version of classic ...
https://www.theregister.co.uk › 2018/11/28 › better_fuzzer_aflsmart
Nov 28, 2018 - Known as **AFLSmart**, this fuzzing software is built on the powerful American ... We're told **AFLSmart** is pretty good at testing applications for common .... The Register - Independent **news** and views for the tech community.

AFLSmart | Latest AFLSmart News, Articles and Updates
https://cyware.com › tags › aflsmart
AFLSmart - Check out latest **news** and articles about **AFLSmart** on Cyware.com. We provide machine learning based curation engine brings you the top and ...

Researchers Introduce Smart Greybox Fuzzing | SecurityWeek ...
https://www.securityweek.com › researchers-introduce-smart-greybox-fuzz...
Nov 29, 2018 - Information Security **News**, IT Security **News** and Cybersecurity Insights: ... According to the experts, **AFLsmart** is highly efficient in analyzing ...



Google FuzzBench

Smart Greybox Fuzzing

https://github.com/aflsmart/aflsmart

17 input models are available

PDF, MP4, MP3, AVI, WAV, PNG, JPEG
JPEG2000, GIF, PCAP, ELF, WEBP, ELF, ZIP, TTF
OTF, OGG