YarIx: Scalable YARA-based Malware Intelligence

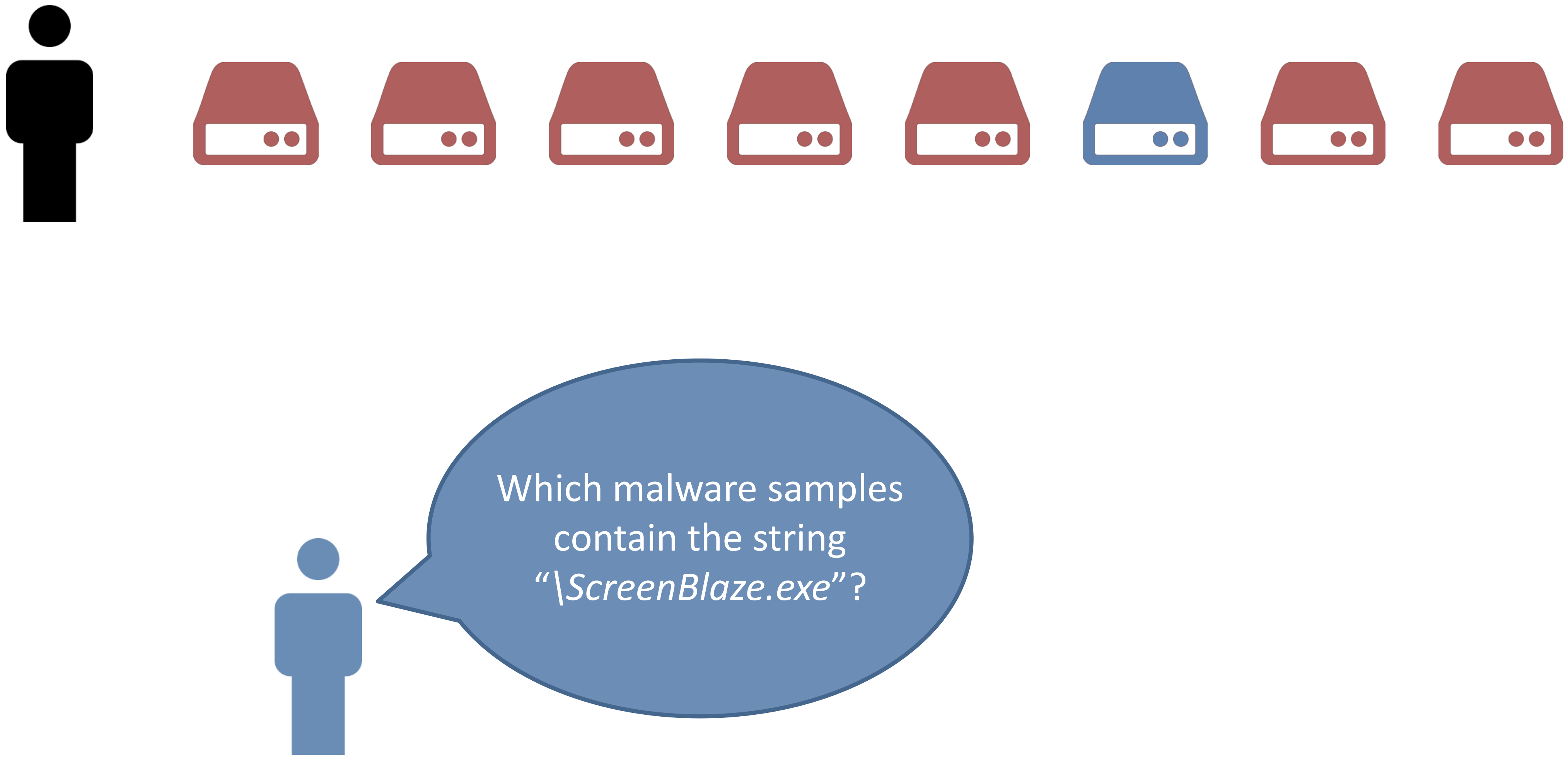
ARTIFACT EVALUATED
usenix ASSOCIATION
PASSED

Michael Brengel, Christian Rossow

CISPA Helmholtz Center for Information Security

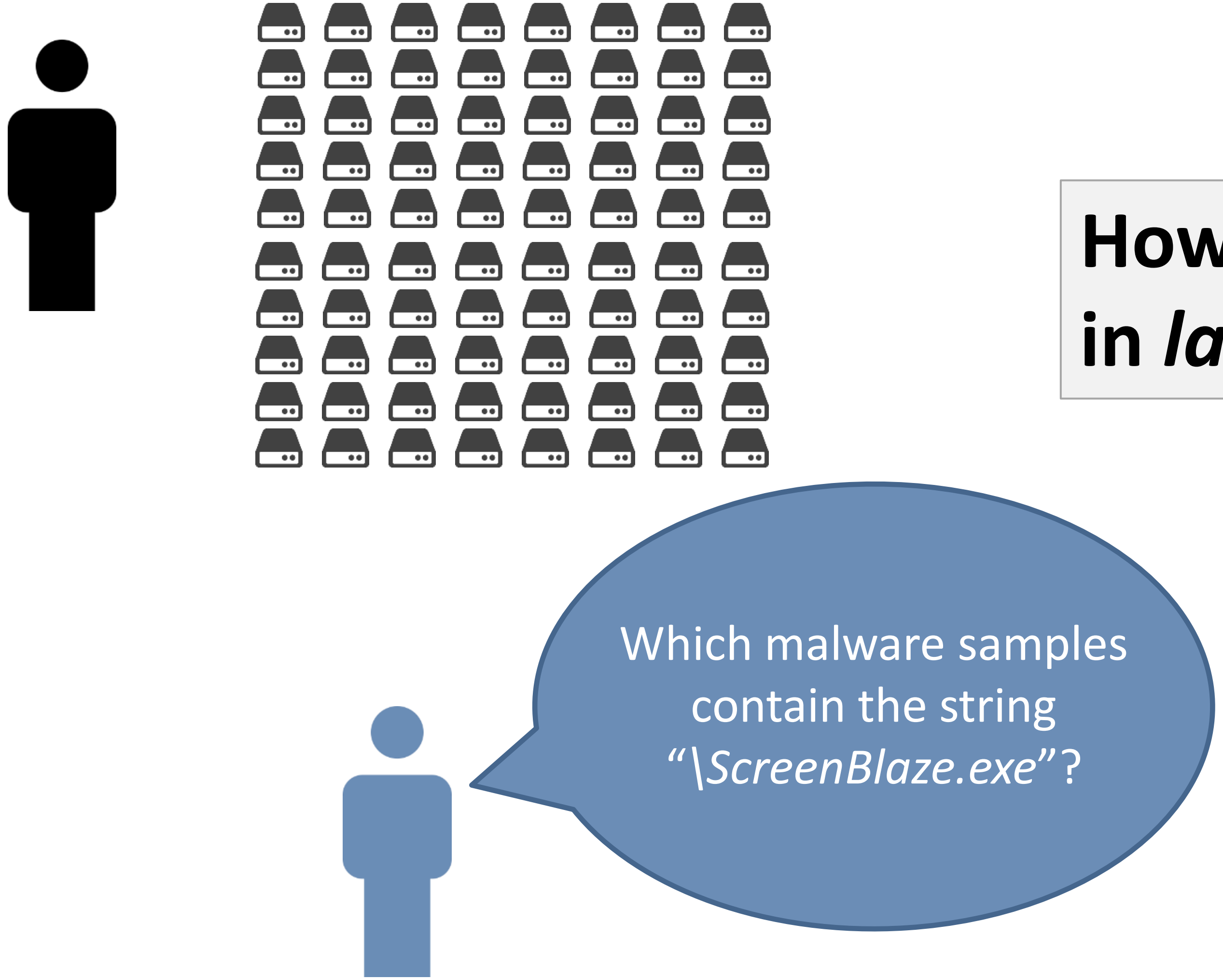
https://github.com/mbrengel/yarix

File Search with YARA Signatures



Which malware samples contain the string “*ScreenBlaze.exe*”?

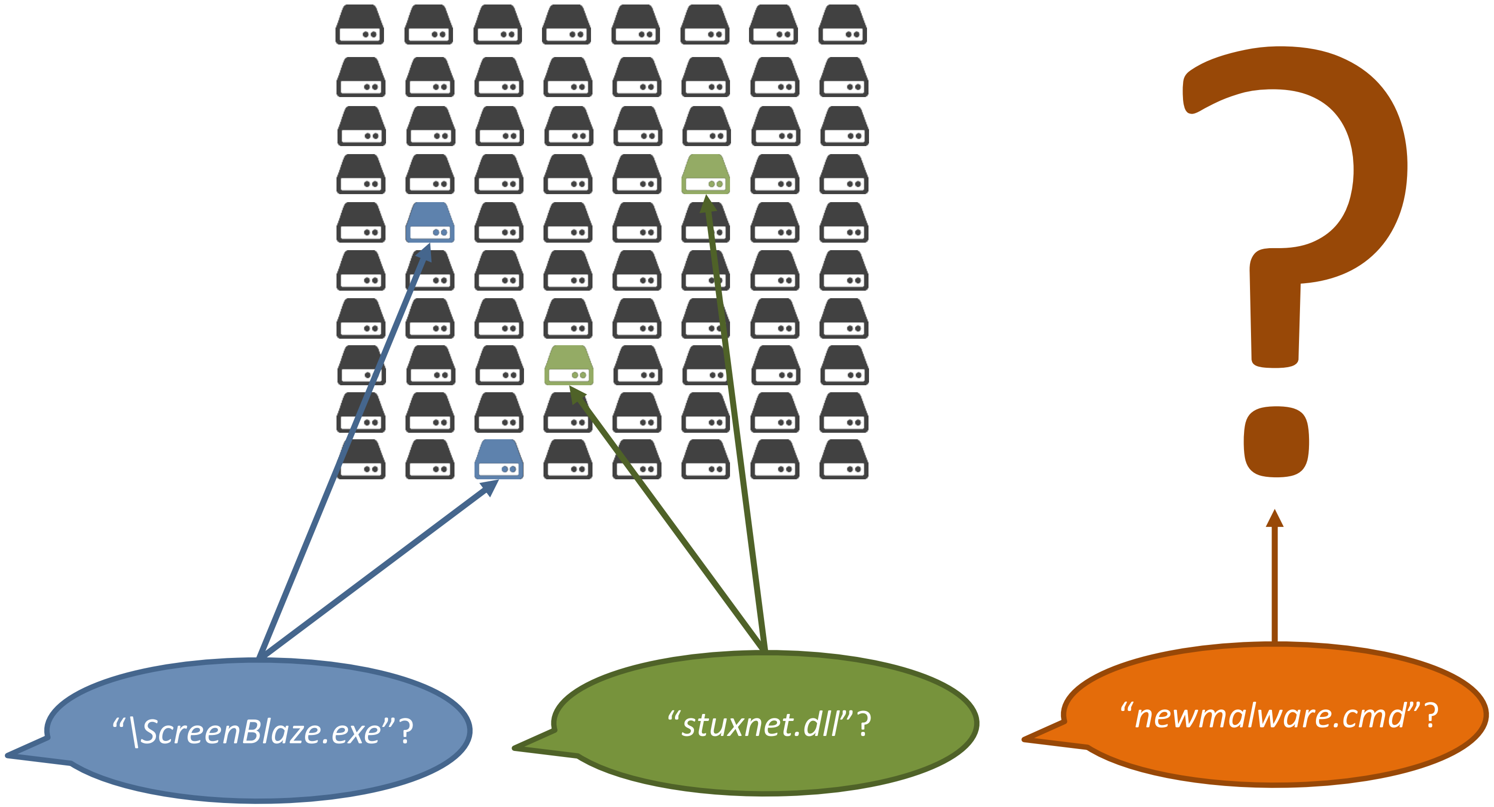
Sequential YARA Scans Do Not Scale



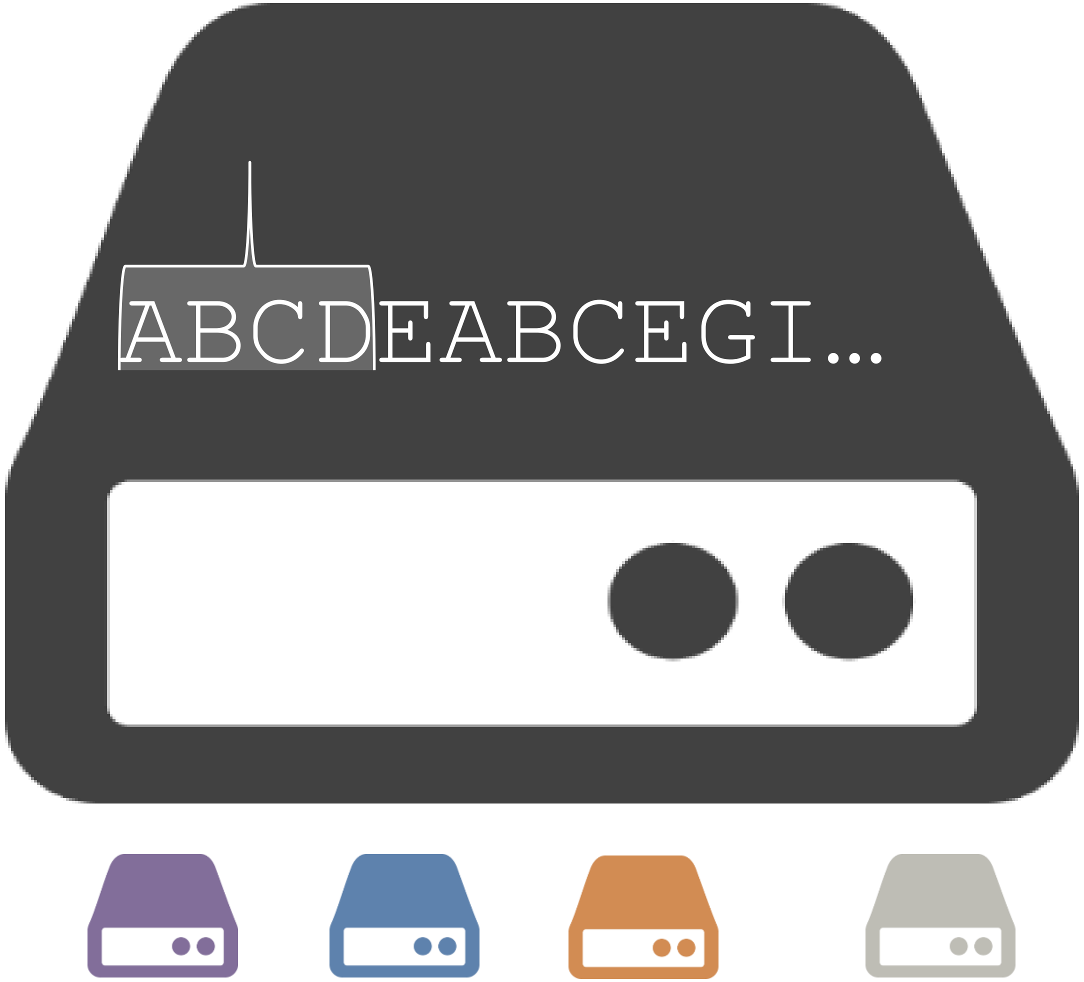
Which malware samples contain the string “*\ScreenBlaze.exe*”?

How can we use YARA *efficiently* in *large* malware databases?

Strawman Solution: Cache Previous Searches

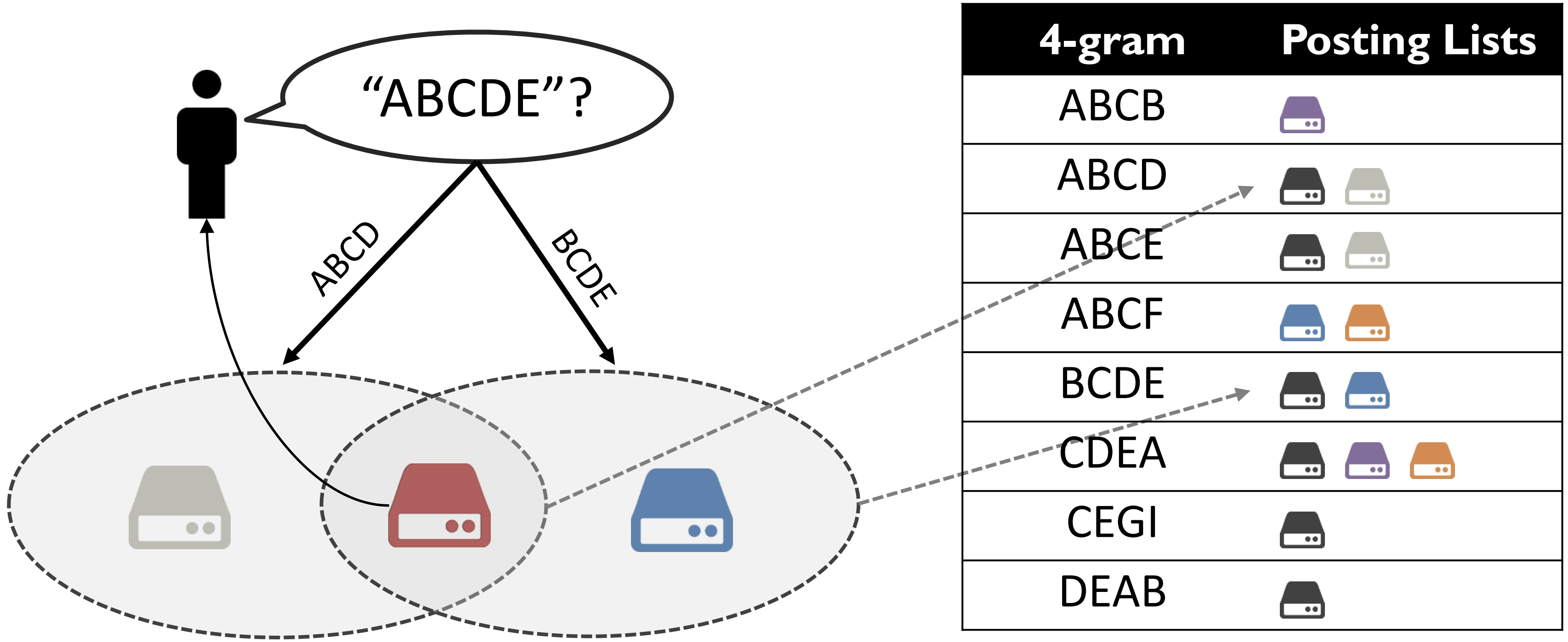


YarIx: Reverse Malware Index



4-gram	Posting Lists
ABCB	🟪
ABCD	⬛ ⬜
ABCE	⬛ ⬜
ABCF	🟦 🟧
BCDE	⬛ 🟦
CDEA	⬛ 🟪 🟧
CEGI	⬛
DEAB	⬛

Yarlx: Reverse Malware Index



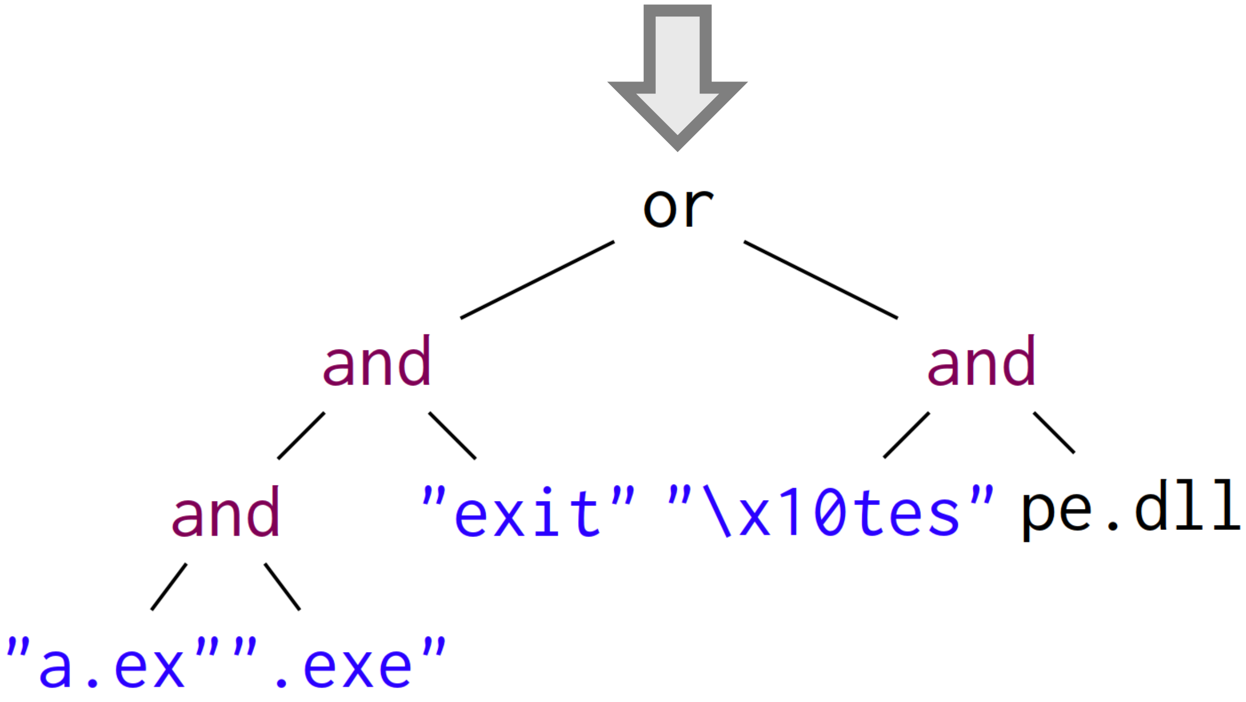
4-gram	Posting Lists
ABCB	🟪
ABCD	⬛ ⬜
ABCE	⬛ ⬜
ABCF	🟦 🟧
BCDE	⬛ 🟦
CDEA	⬛ 🟪 🟧
CEGI	⬛
DEAB	⬛

Extract Search Terms From YARA Signatures

YARA rule

```
      "a.exe" and "exit"
              or
"\\x10tes[0-9]" and pe.dll
```

YarIx Search Terms



or

and

and

and

"exit"

"\x10tes"

pe.dll

"a.ex"

".exe"

YARA Feature Support

✓ Plain Strings (“a.exe”, “exit”, “pe.dll”, …)

✓ Hex Strings (“{ DE AD BE EF }”, “{ CA FE FE BA [2-5] BE FF FF FF }”)

✓ Regular Expressions (“calc[0-9a-z]+\.exe”)

✓ 2 of (“ABCD”, “ABCE”, “ABCF”, “BCDE”)

✓ Condition Logic

“ABCD” and “ABCE” ➔ “ABCD” ∩ “ABCE”

“ABCD” or “ABCE” ➔ “ABCD” ∪ “ABCE”

- **Search terms using unsupported features are (e.g., the `not` keyword) over-approximated**

Storage-Efficient Index Design (1/2)

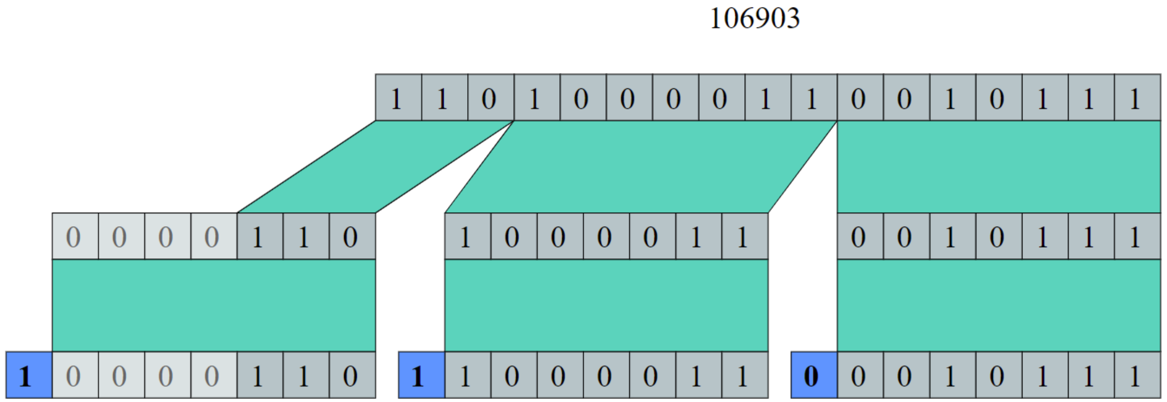
- Offset-free

  🖴, 🖴, 🖴

- Delta Encoding for File IDs

  {1000000, 4, 1, 5, 2, 4, 4 }

- Variable-length 7-bit encoding for File IDs



106903

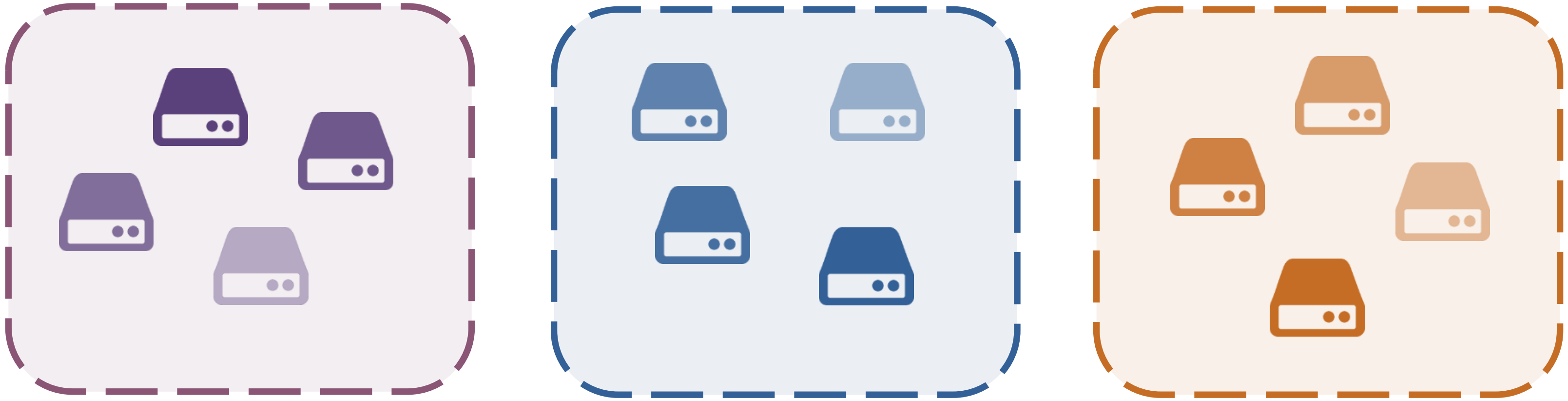
1 1 0 1 0 0 0 0 1 1 0 0 1 0 1 1 1

0 0 0 0 1 1 0 | 1 0 0 0 0 1 1 | 0 0 1 0 1 1 1

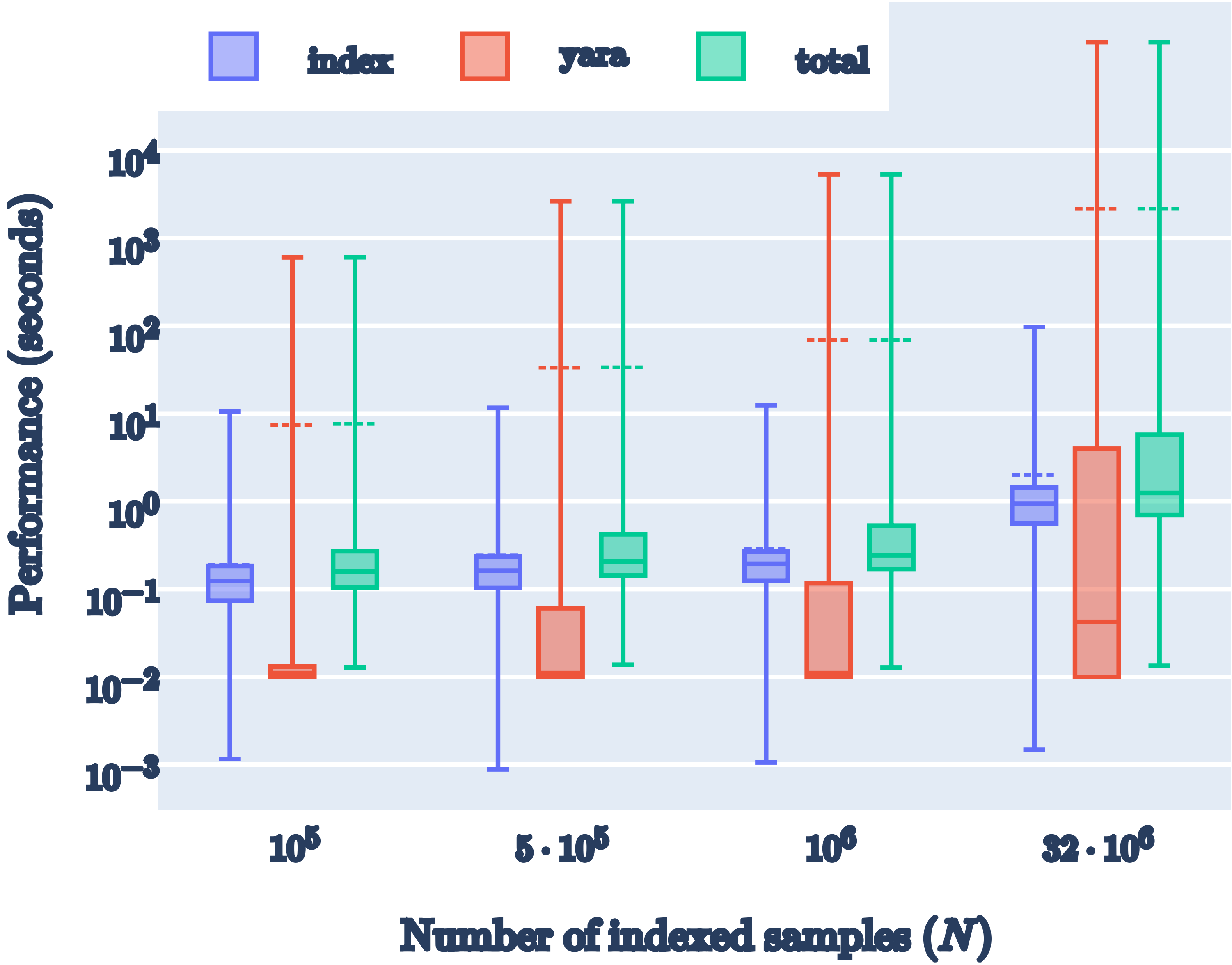
1 0 0 0 0 1 1 0 | 1 1 0 0 0 0 1 1 | 0 0 0 1 0 1 1 1

Storage-Efficient Index Design (2/2)

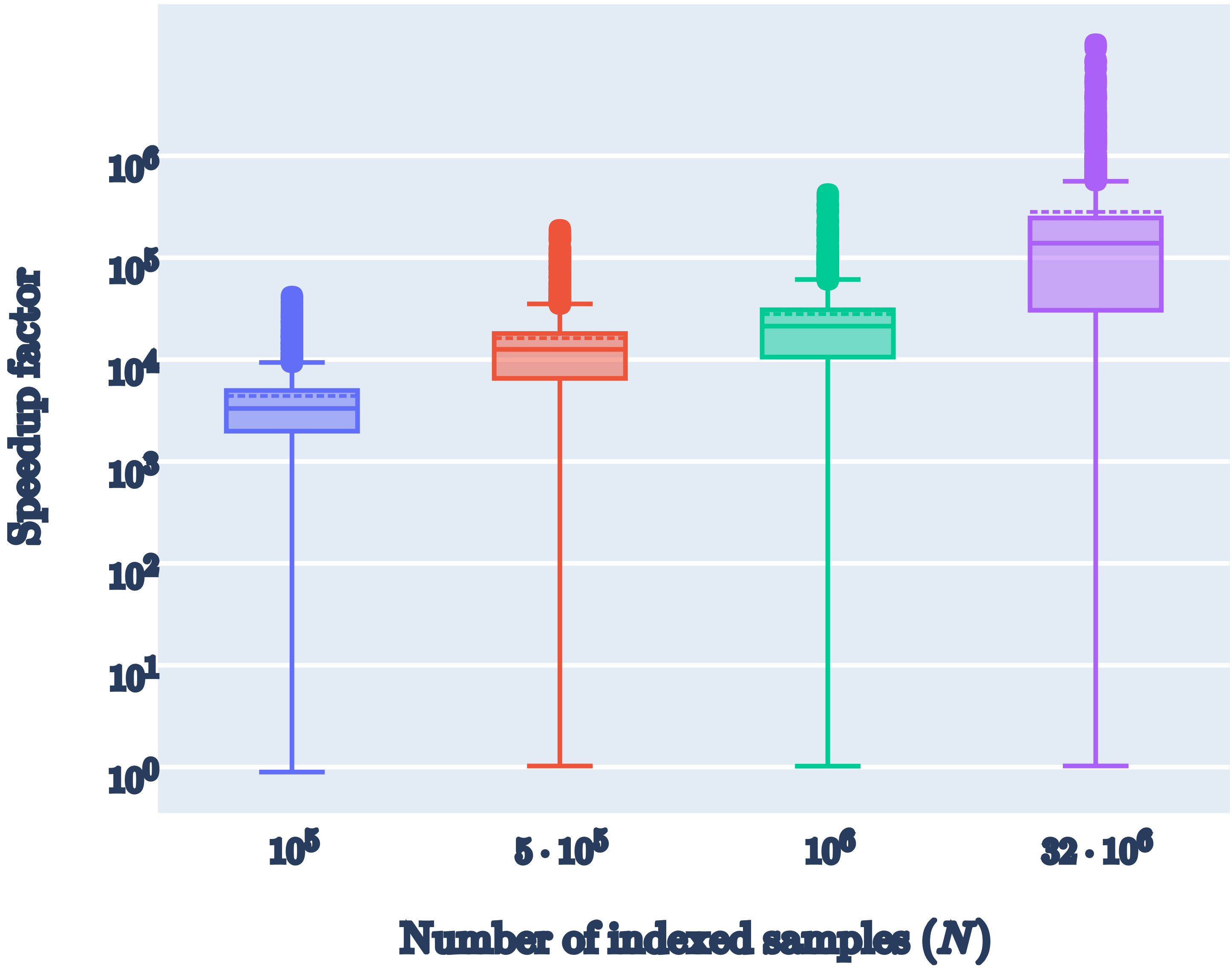
- Grouping



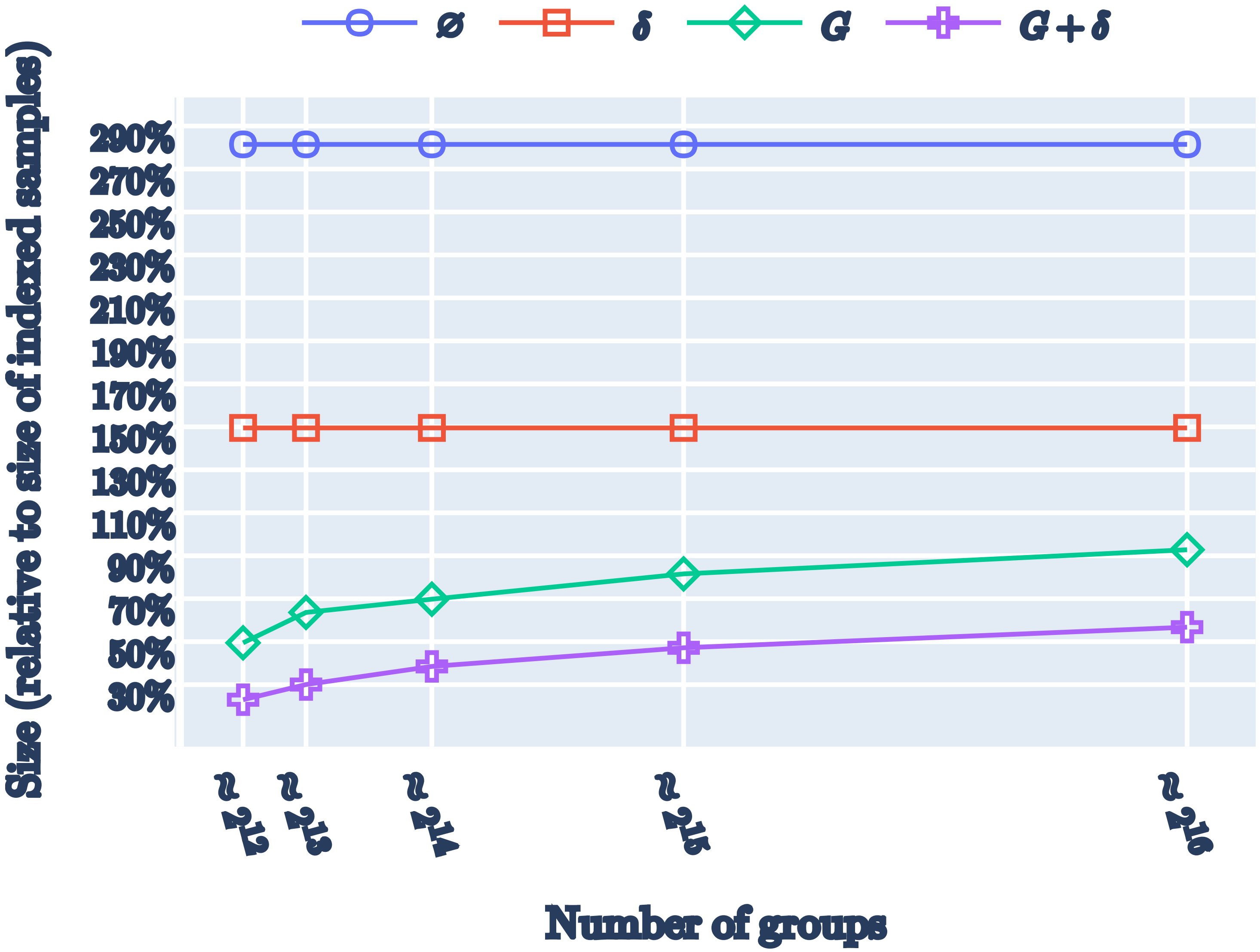
Evaluation (Performance)



Evaluation (Performance)



Evaluation (Space)



Summary

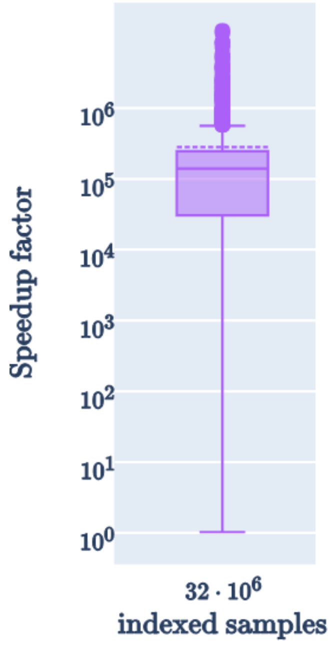
Plain Strings (“a.exe”, “exit”, “pe.dll”, …) ✓

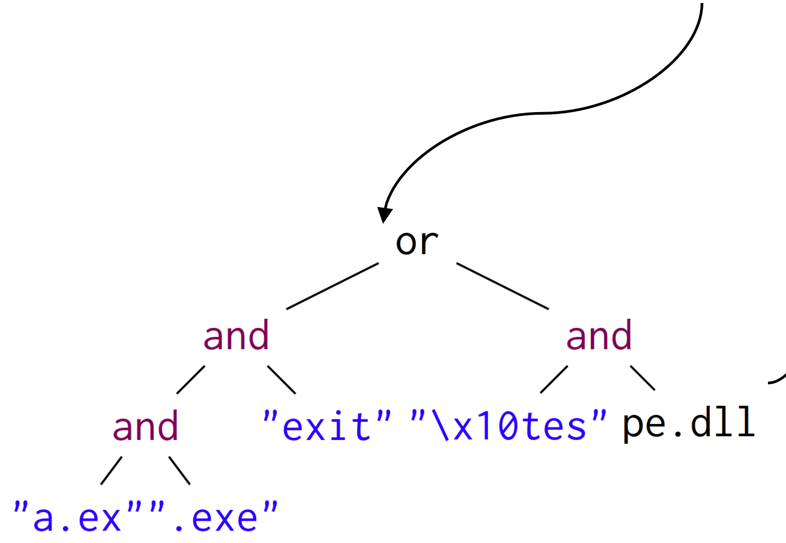
Hex Strings (“{ DE AD BE EF }”, “{ CA FE FE BA [2-5] BE FF FF FF }”) ✓

Regular Expressions (“calc[0-9a-z]+\.exe”) ✓

2 of (“ABCD”, “ABCE”, “ABCF”, “BCDE”) ✓

Condition Logic ✓





4-gram	Posting Lists
ABCB	🟪
ABCD	⬛ ⬜
ABCE	⬛ ⬜
ABCF	🟦 🟧
BCDE	⬛ 🟦
CDEA	⬛ 🟪 🟧
CEGI	⬛
DEAB	⬛

ARTIFACT EVALUATED usenix ASSOCIATION PASSED

https://github.com/mbrengel/yarix