RAIN: Refinable Attack Investigation with On-demand Inter-process Information Flow Tracking

Yang Ji, Sangho Lee, Evan Downing, Weiren Wang, Mattia Fazzini, Taesoo Kim, Alessandro Orso, and Wenke Lee

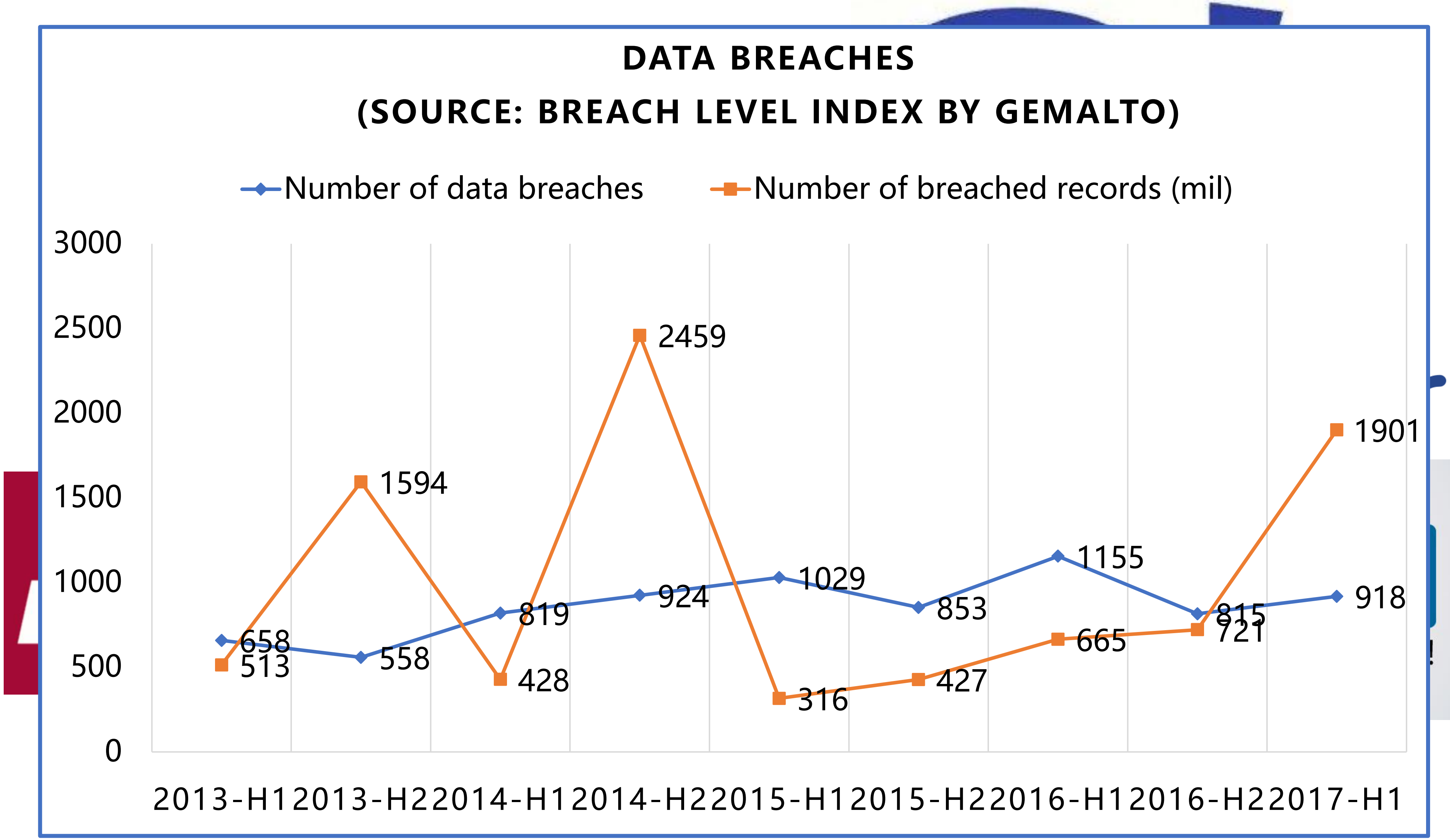
ACM CCS 2017
Oct 31, 2017

Georgia Tech

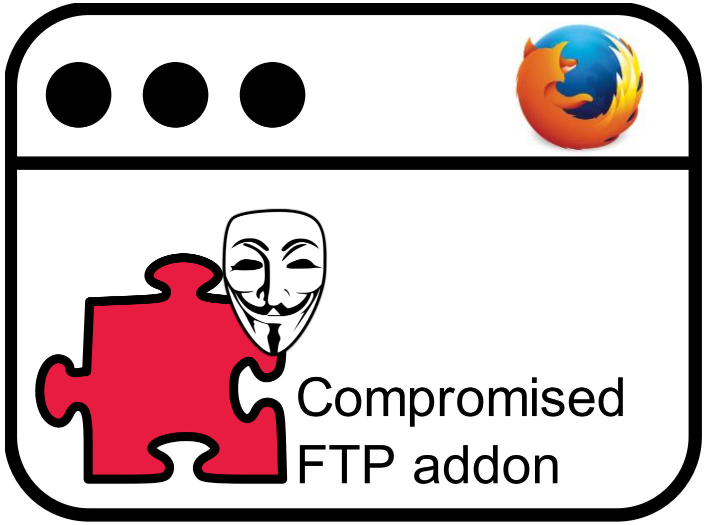
More and more data breaches

SONY
Y!
YAHOO
BREACH
experian™
Mega Data Breach
EQUIFAX
Exposes 143 Million U.S. Consumers' Data
myspace
167 Million
Linked in
Hacked accounts on SALE!

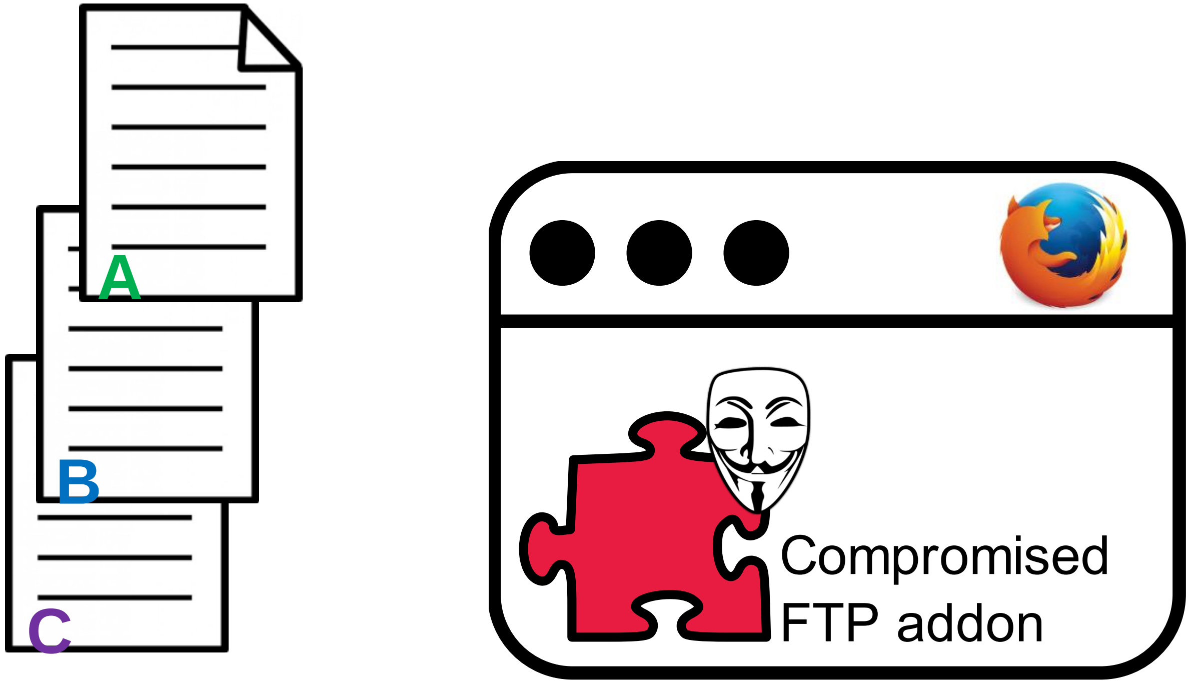
More and more data breaches



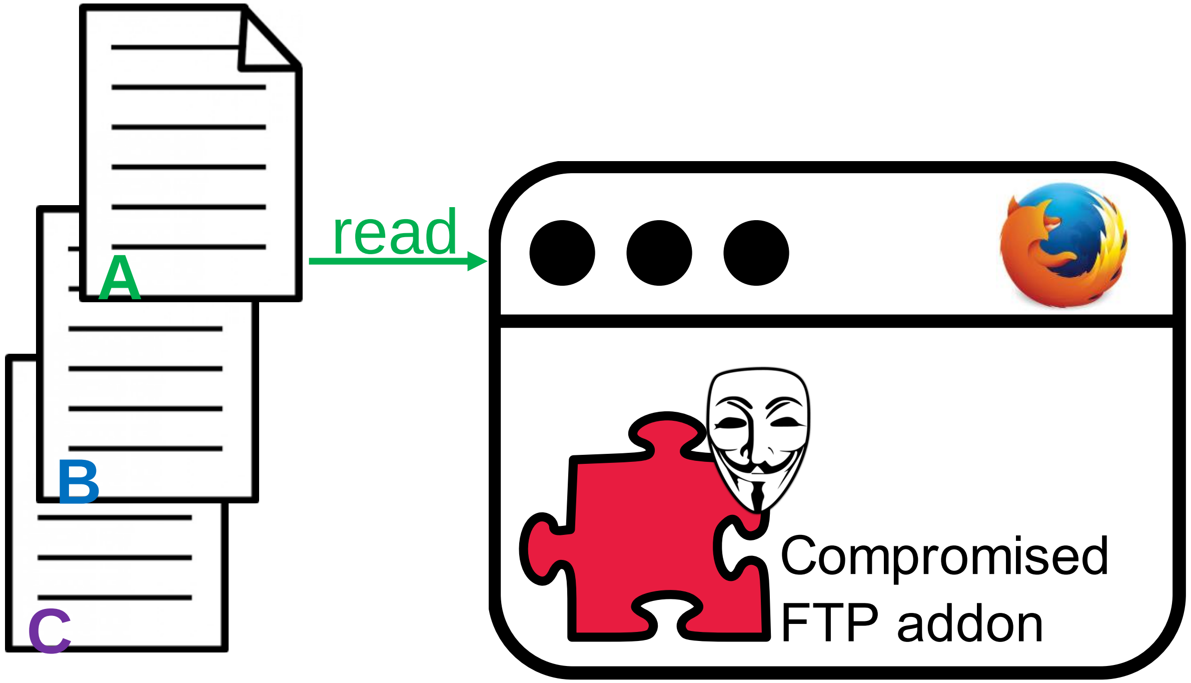
Is attack investigation accurate?



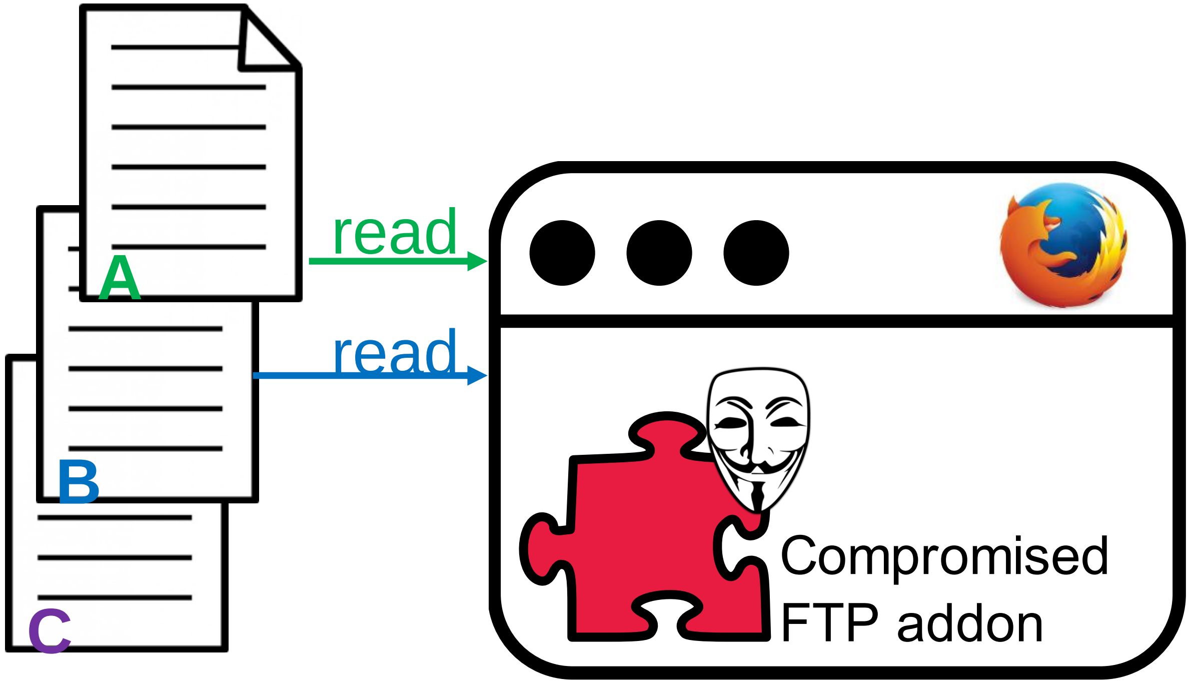
Is attack investigation accurate?



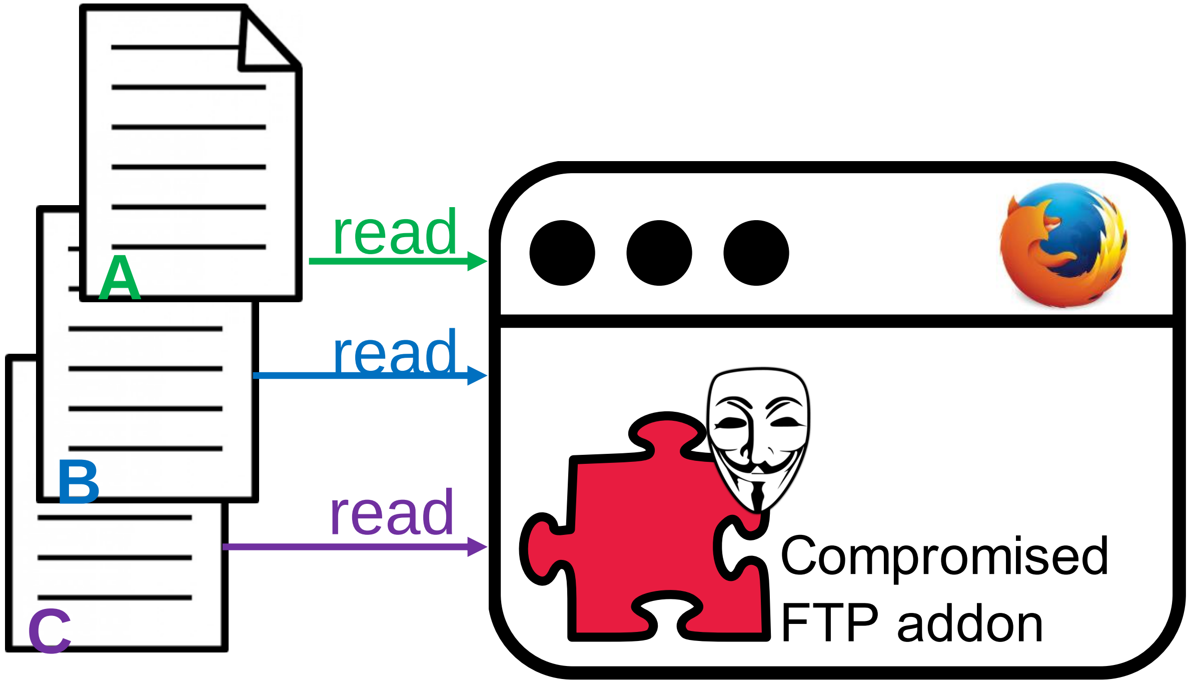
Is attack investigation accurate?



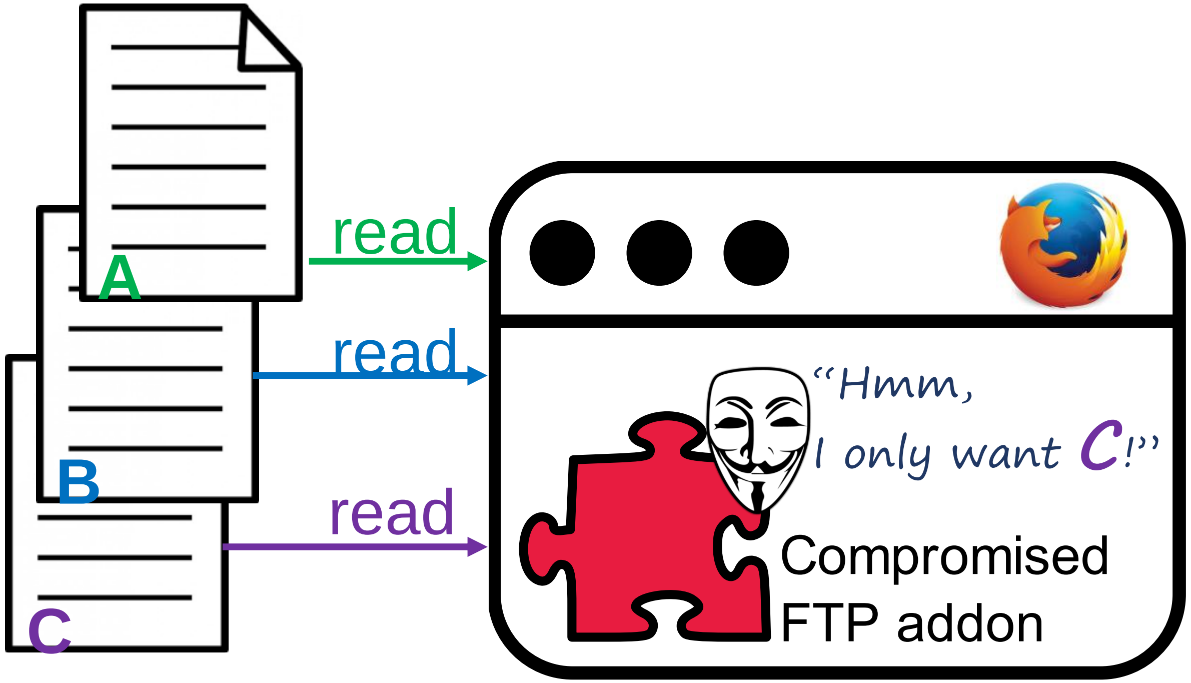
Is attack investigation accurate?



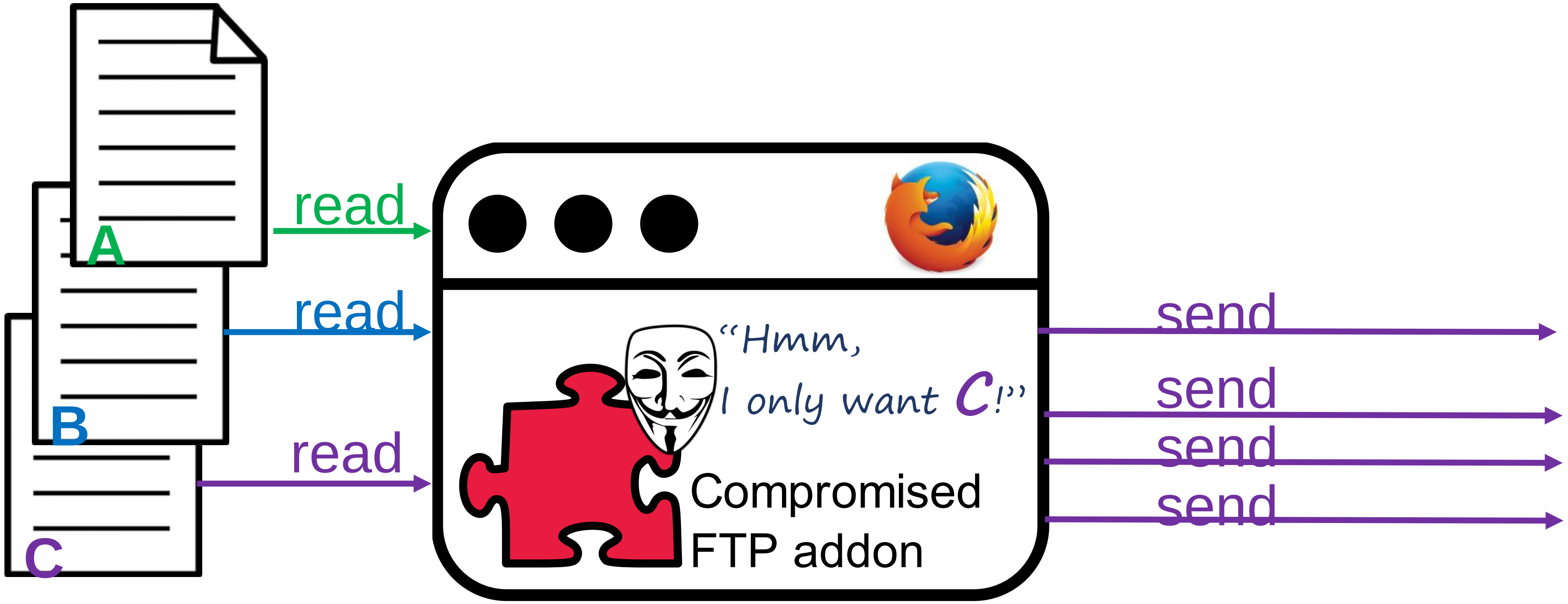
Is attack investigation accurate?



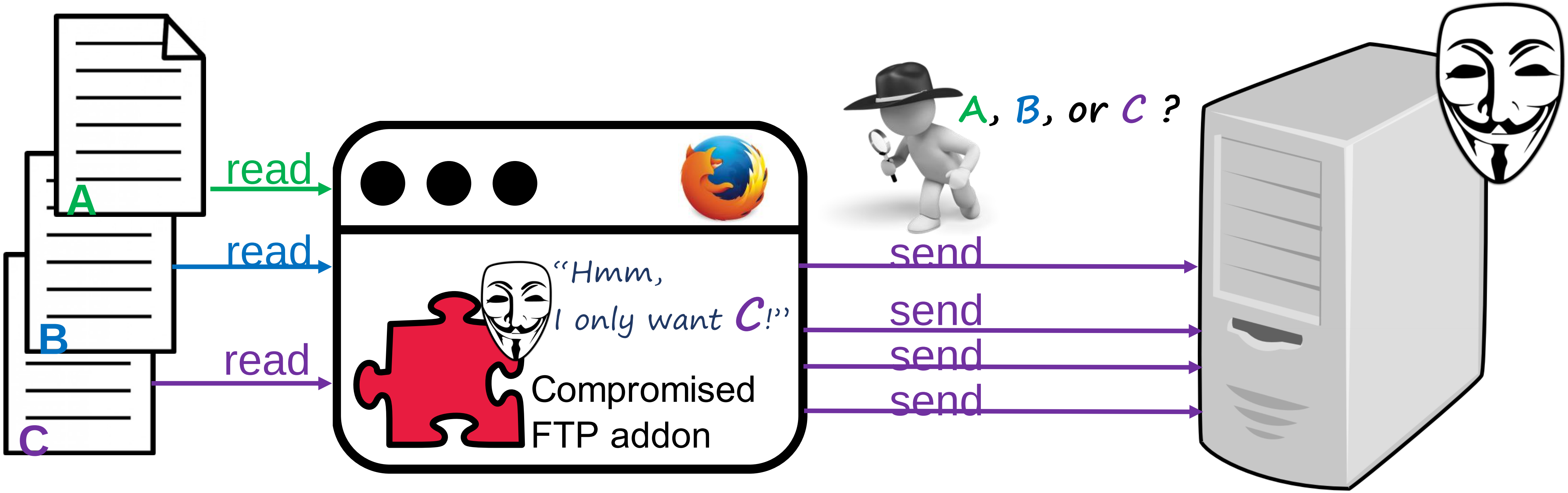
Is attack investigation accurate?



Is attack investigation accurate?

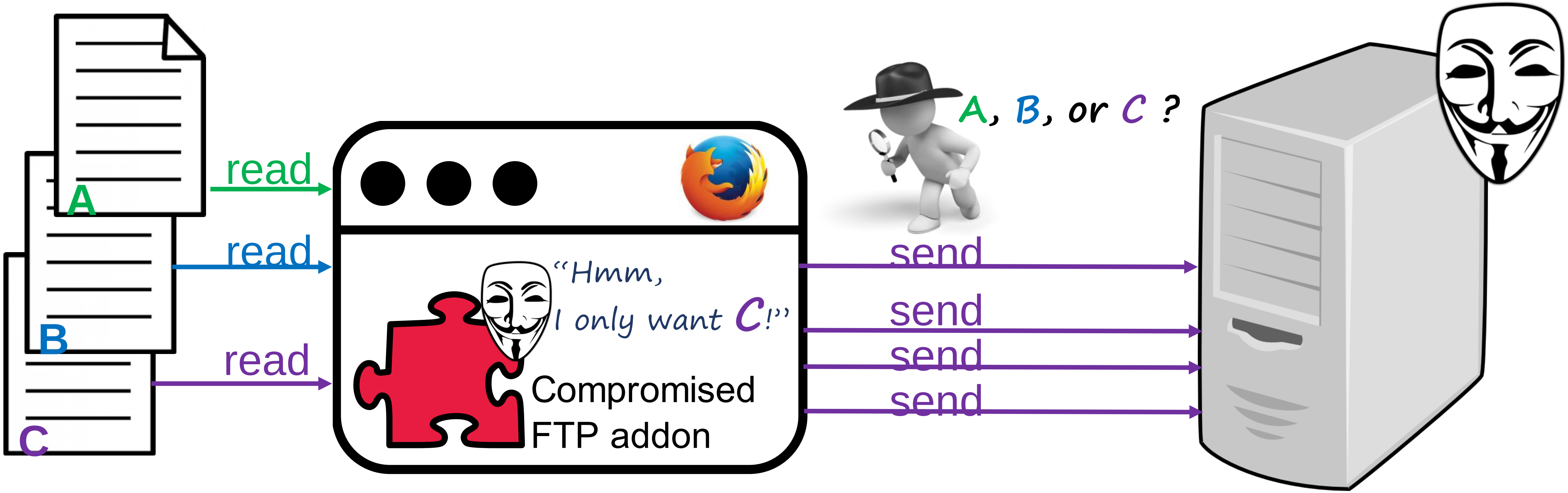


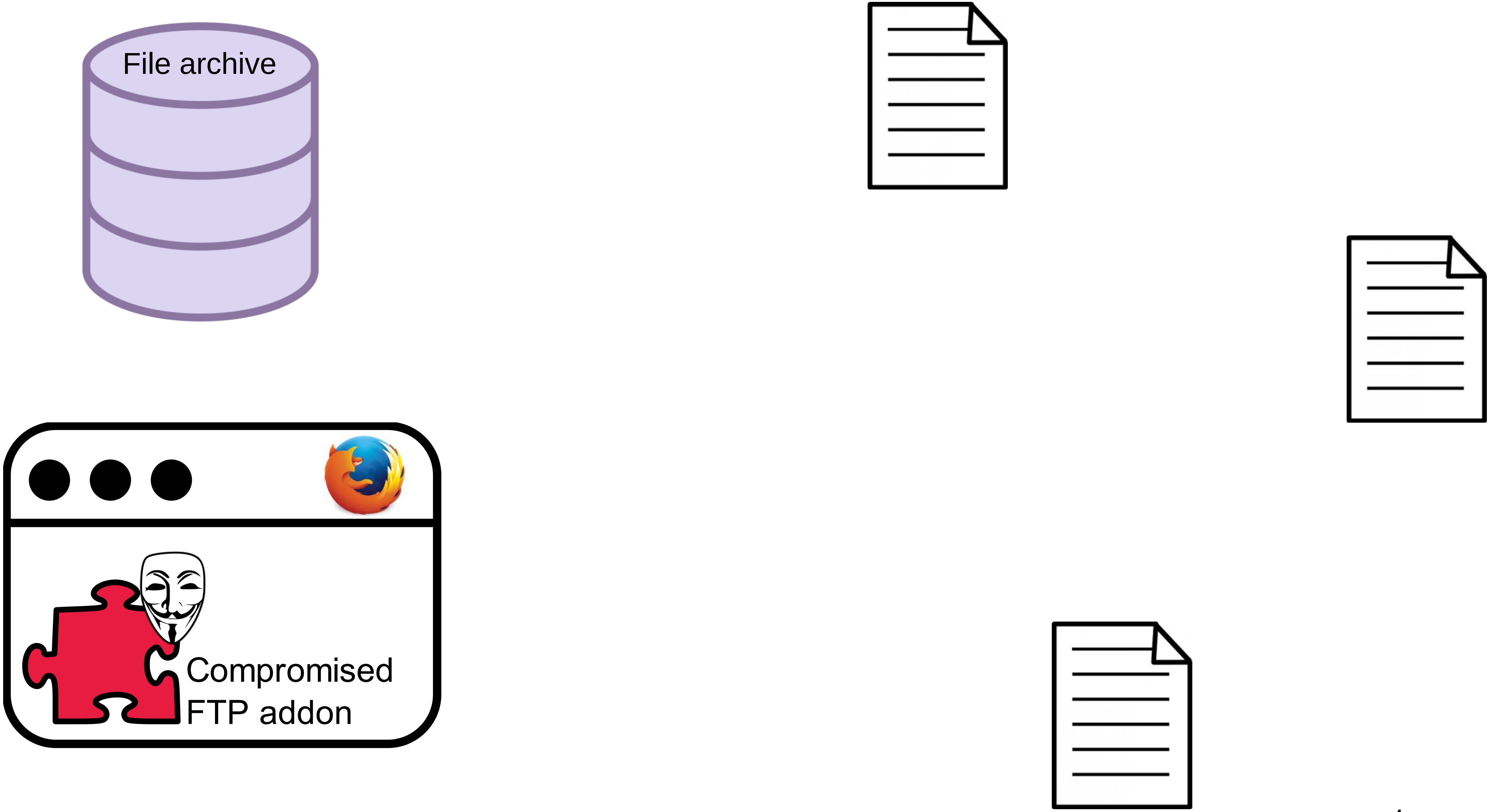
Is attack investigation accurate?



Is attack investigation accurate?

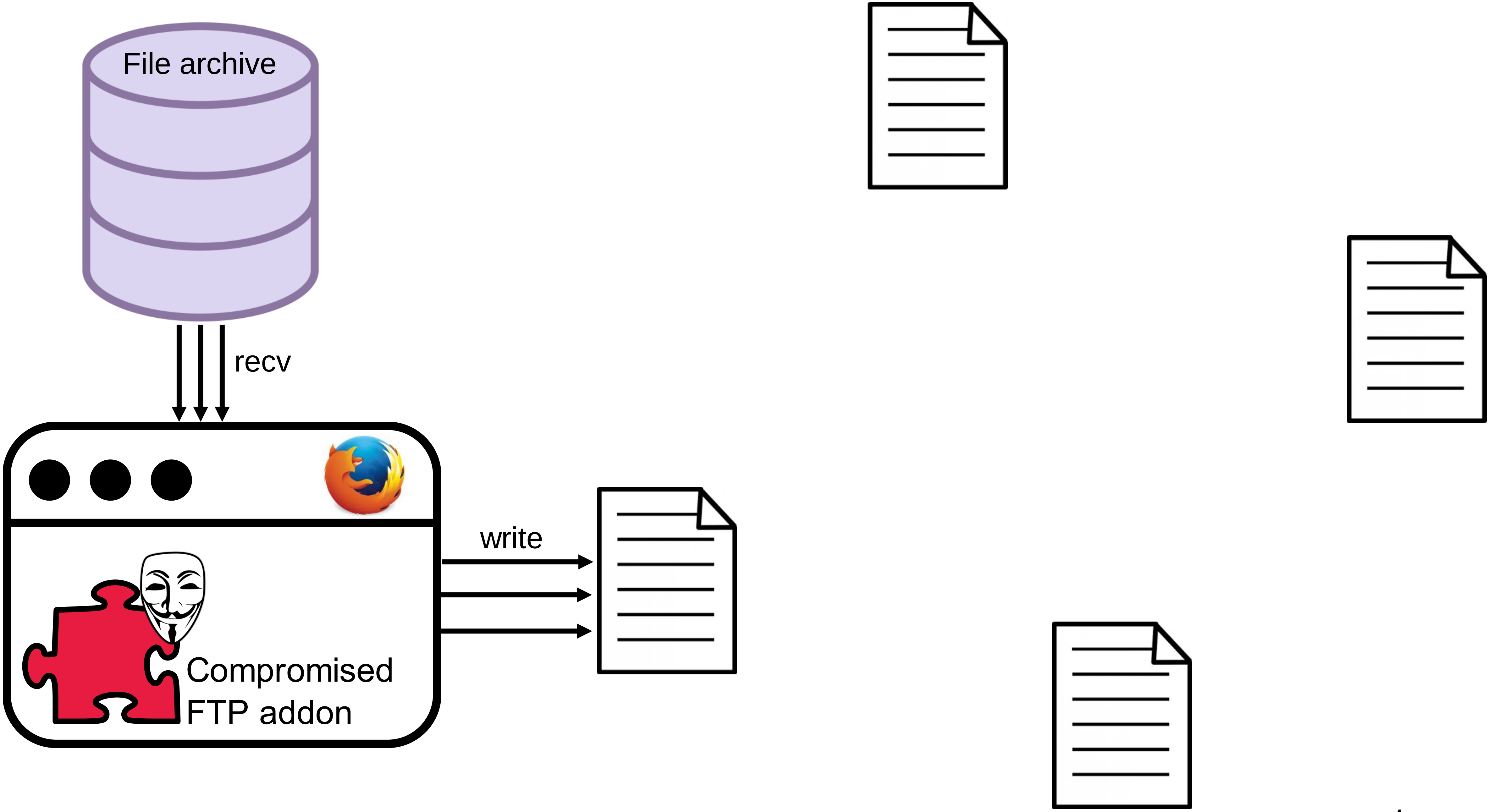
Dependency confusion!





File archive

Compromised FTP addon

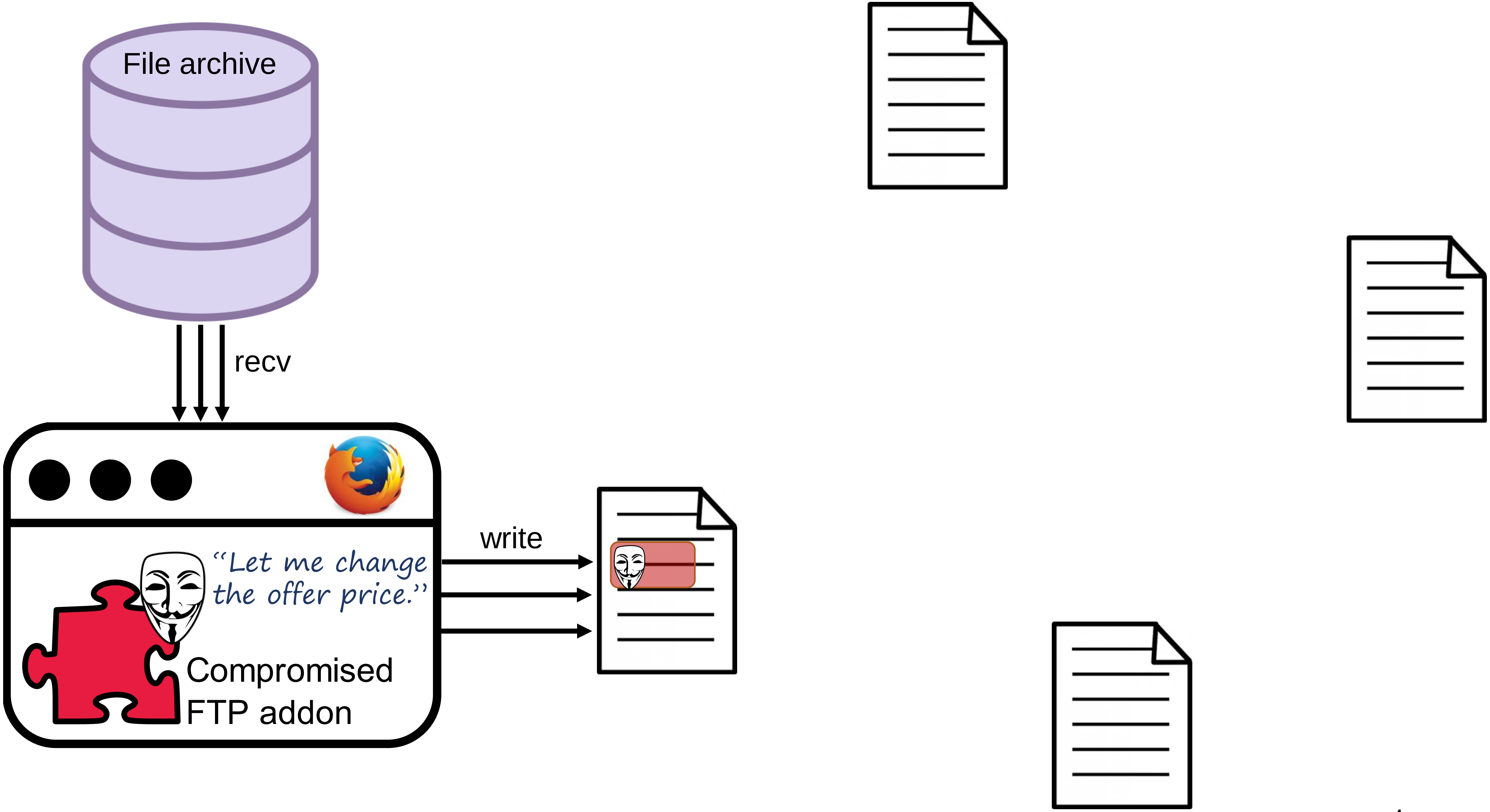


File archive

recv

Compromised FTP addon

write



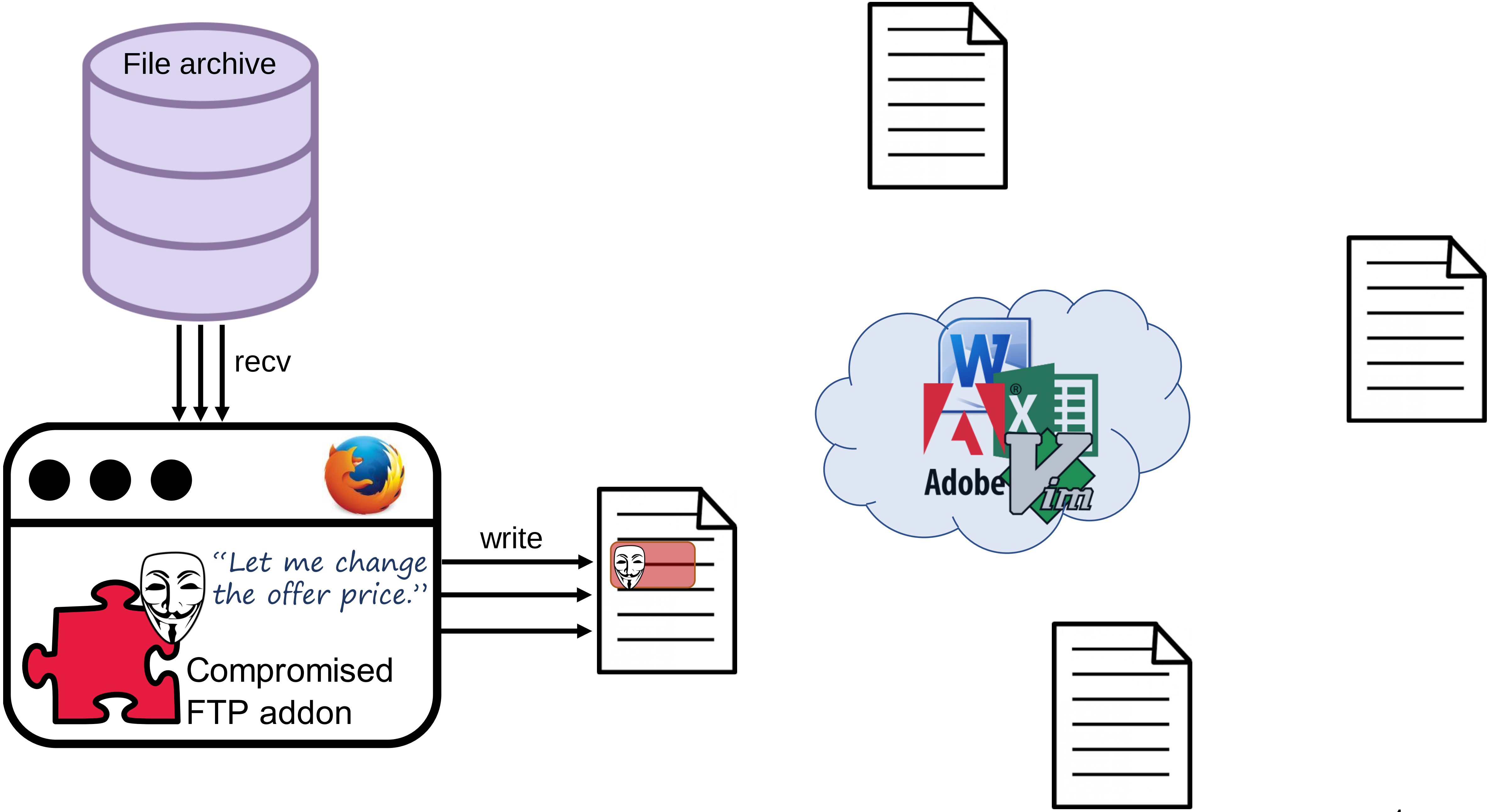
File archive

recv

“Let me change the offer price.”

Compromised FTP addon

write



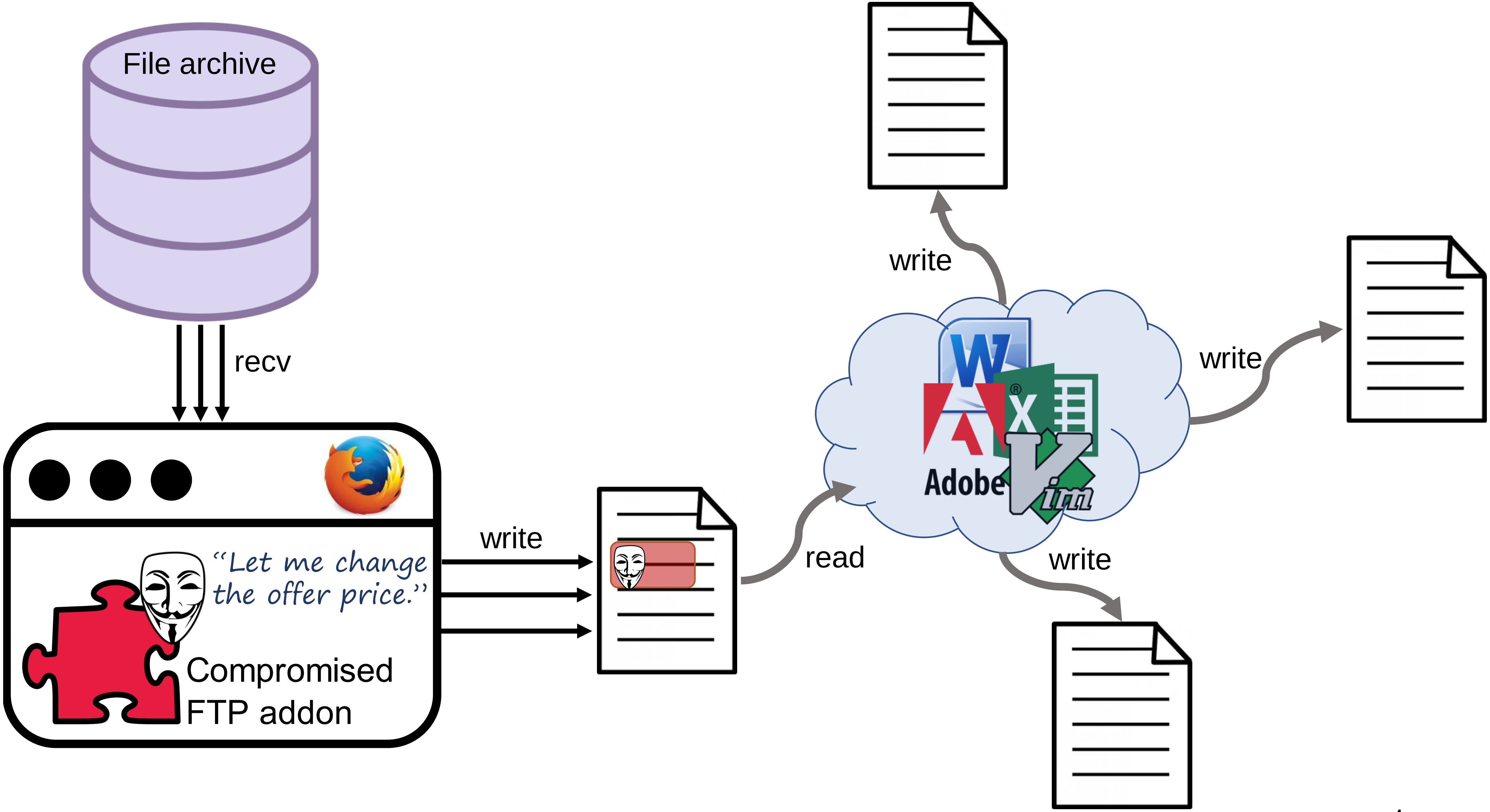
File archive

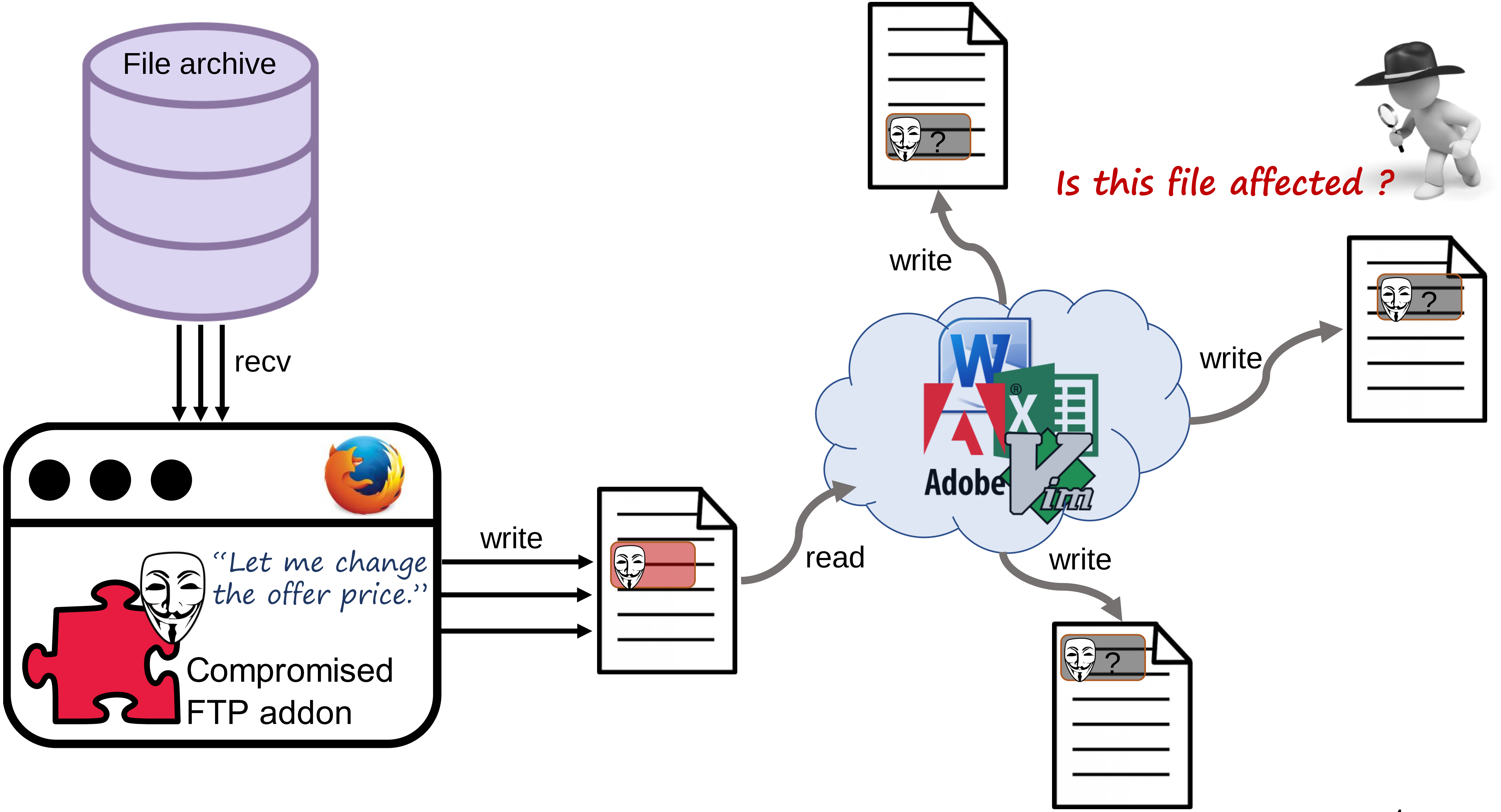
recv

“Let me change the offer price.”

Compromised FTP addon

write


File archive
recv
"Let me change the offer price."
Compromised FTP addon
write
read
write
write
write
Adobe



File archive

recv

Compromised FTP addon

"Let me change the offer price."

write

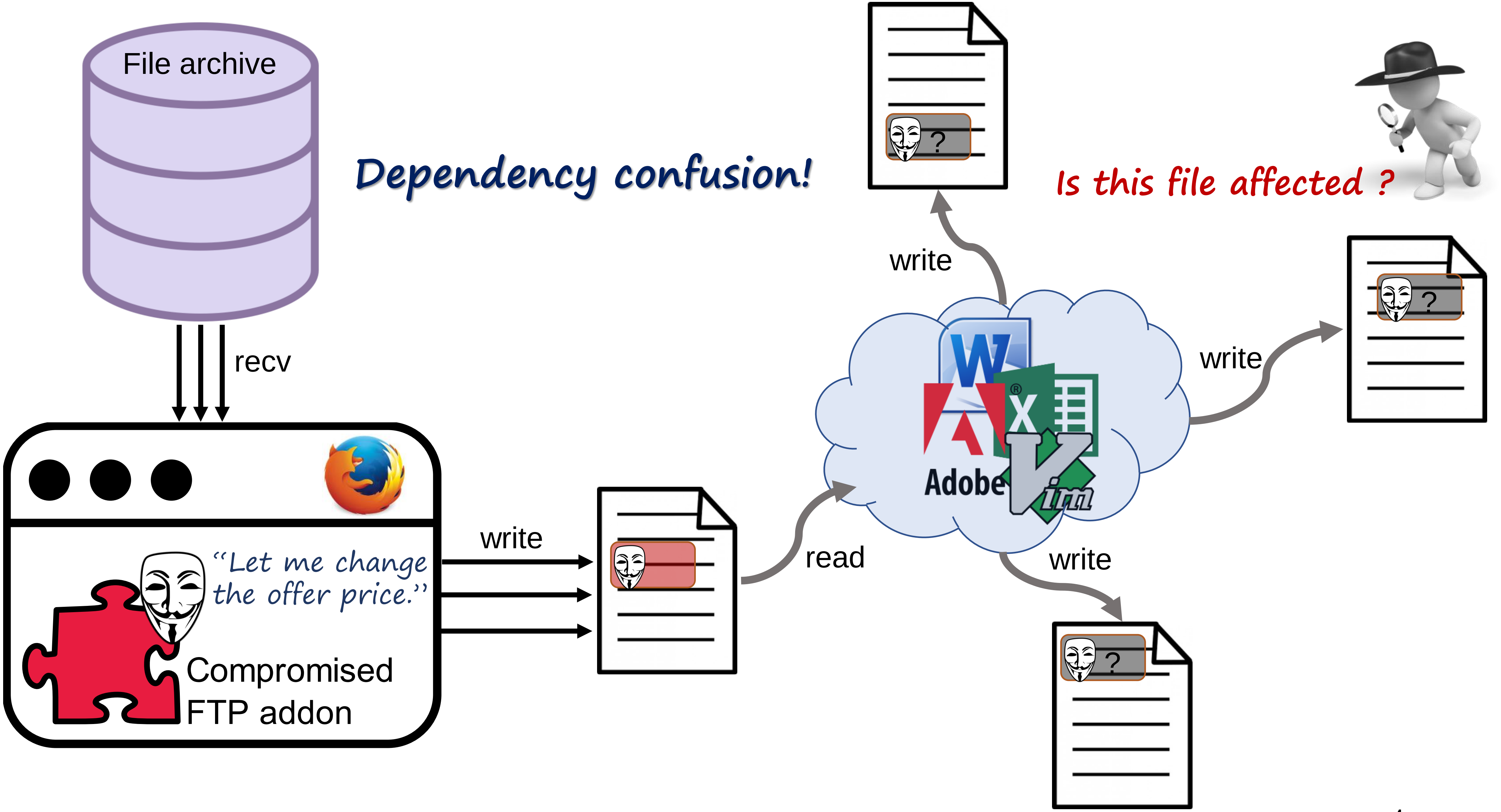
read

write

write

write

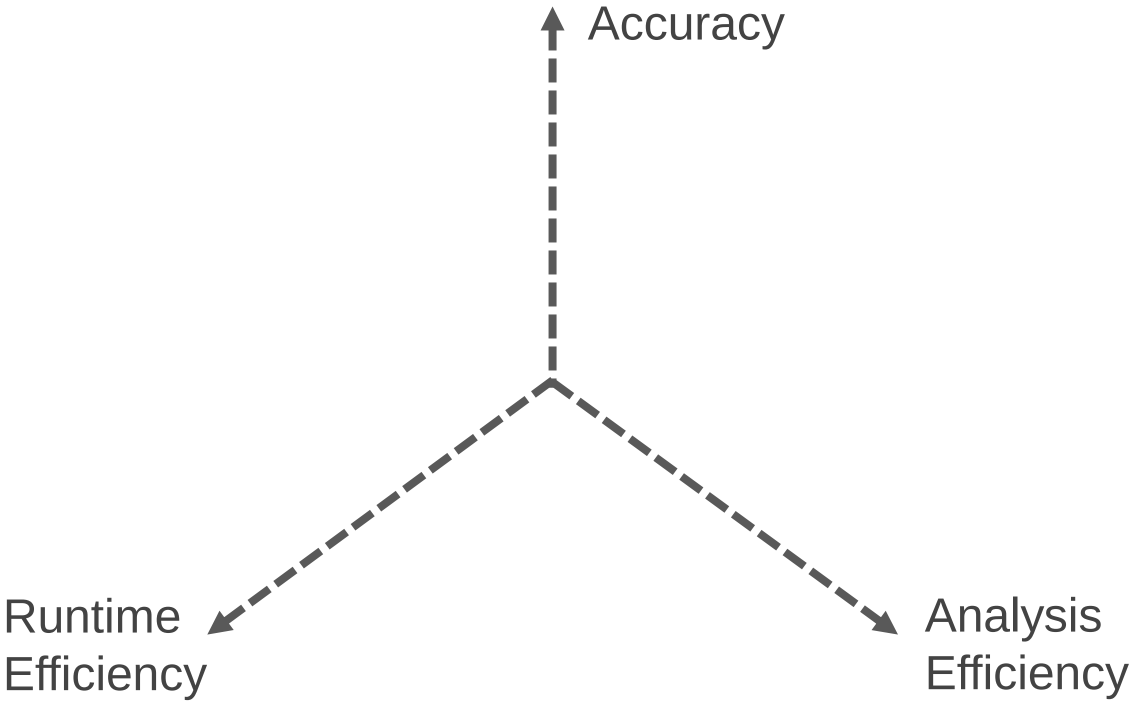
Is this file affected ?



Dependency confusion!

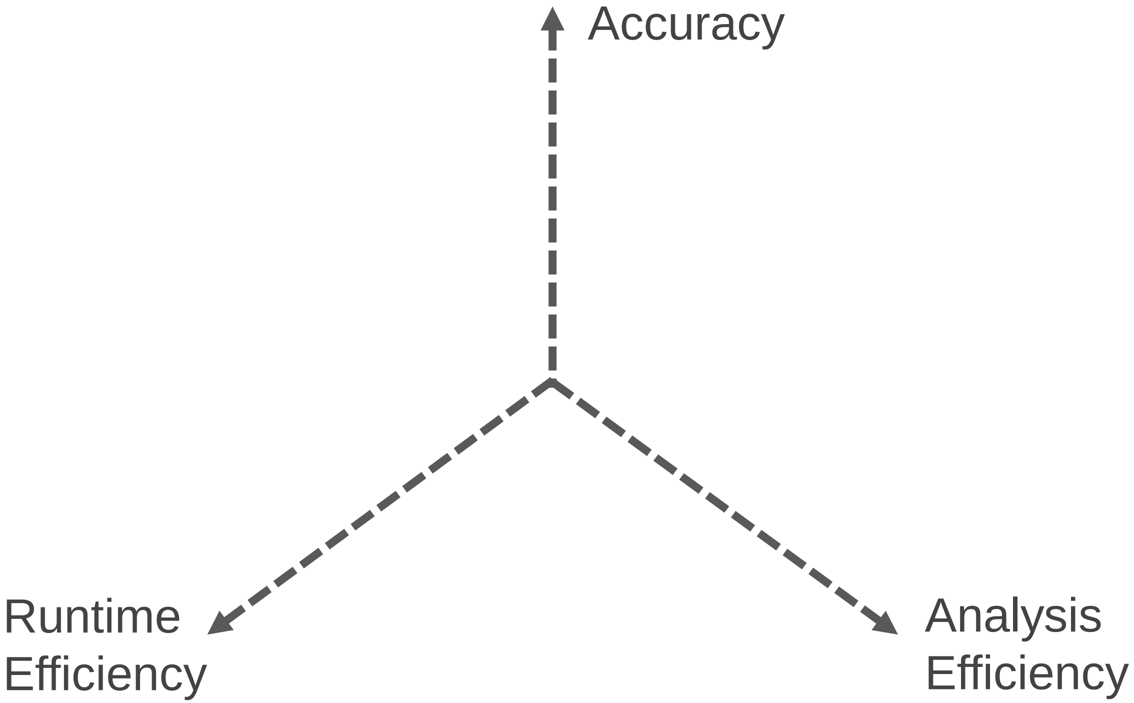
Is this file affected ?

Related work



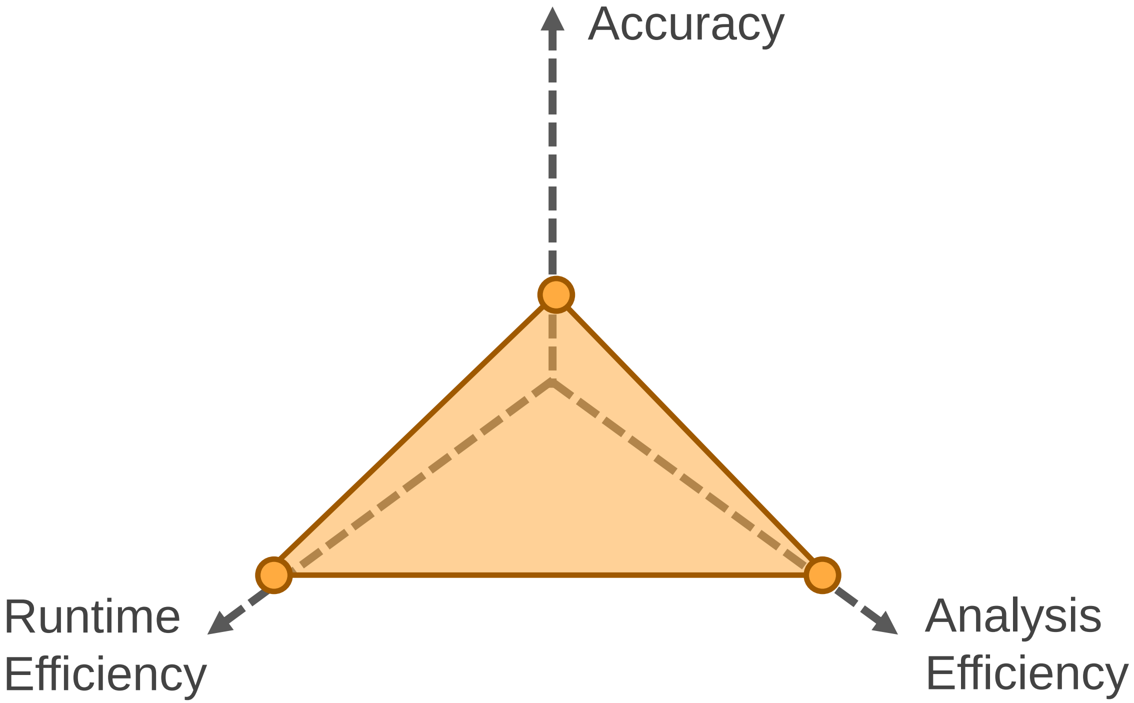
Related work

- System-call-based
  - DTrace, Protracer, LSM, Hi-Fi



Related work

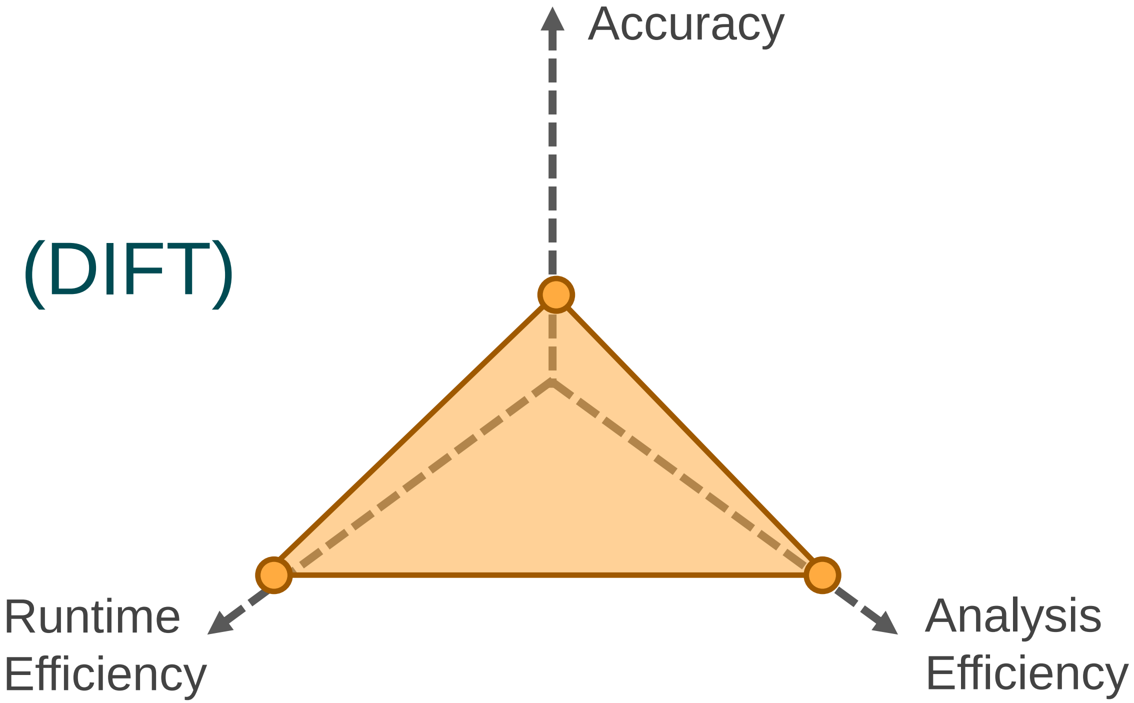
- System-call-based
    - DTrace, Protracer, LSM, Hi-Fi



Related work

- System-call-based
    - DTrace, Protracer, LSM, Hi-Fi

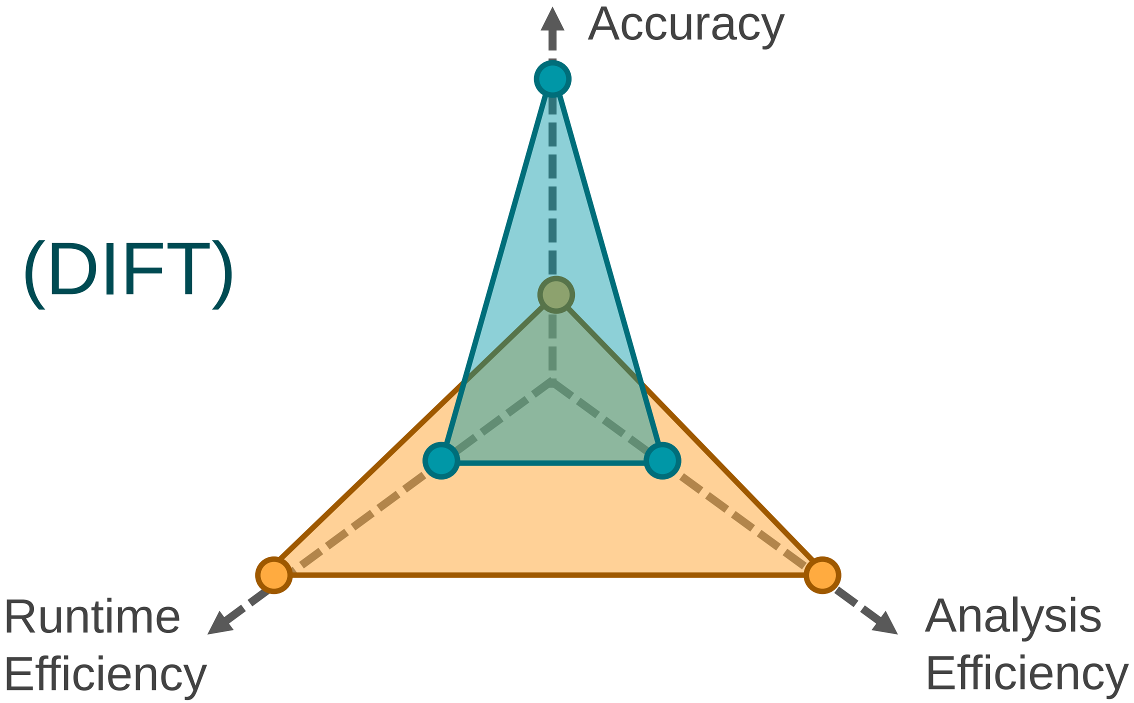
- Dynamic Information Flow Tracking (DIFT)
    - Panorama, Dtracker



Related work

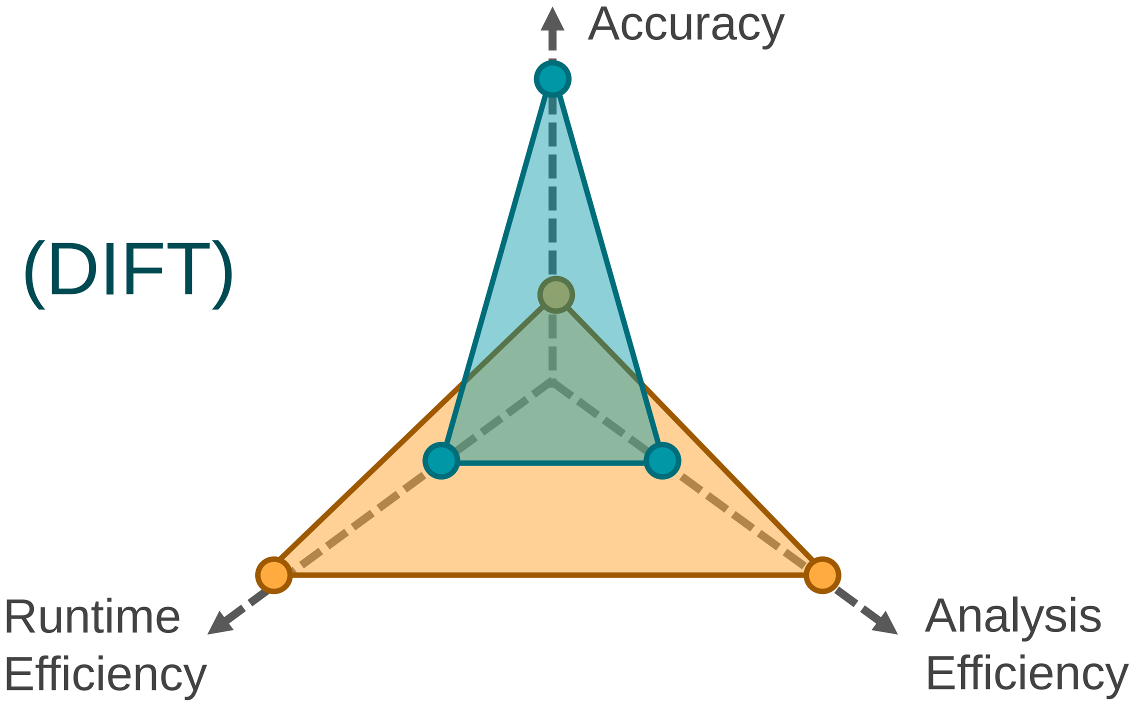
- System-call-based
  - DTrace, Protracer, LSM, Hi-Fi

- Dynamic Information Flow Tracking (DIFT)
  - Panorama, Dtracker



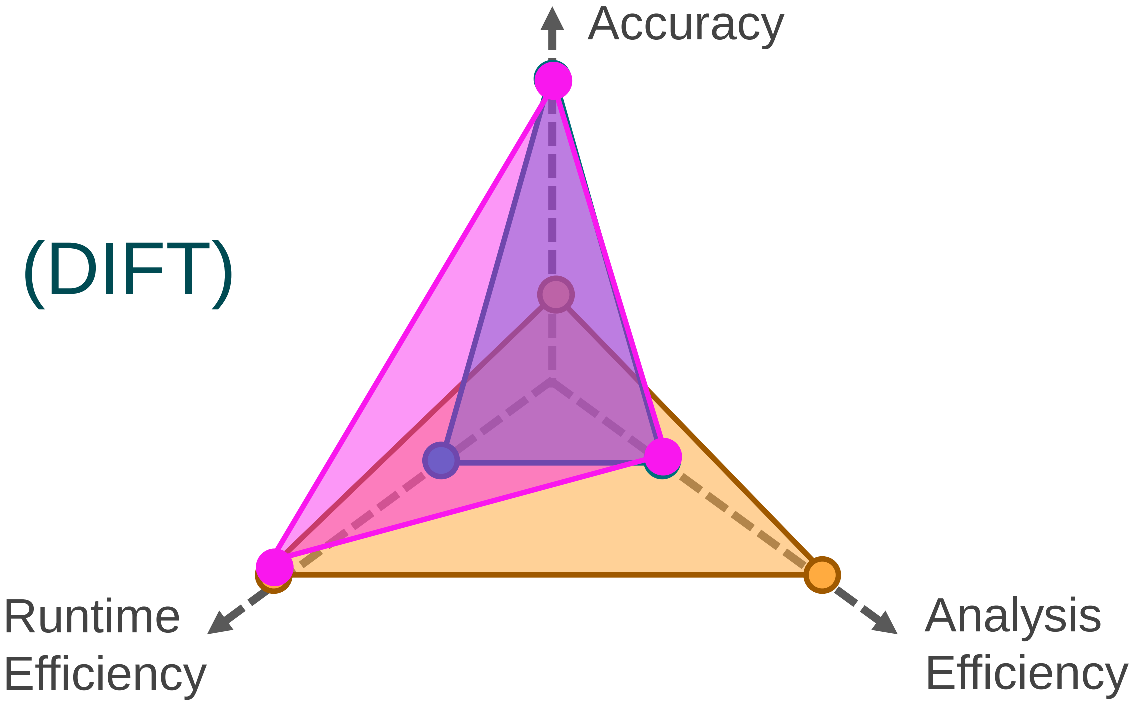
Related work

- System-call-based
  - DTrace, Protracer, LSM, Hi-Fi
- Dynamic Information Flow Tracking (DIFT)
  - Panorama, Dtracker
- DIFT + Record replay
  - Arnold

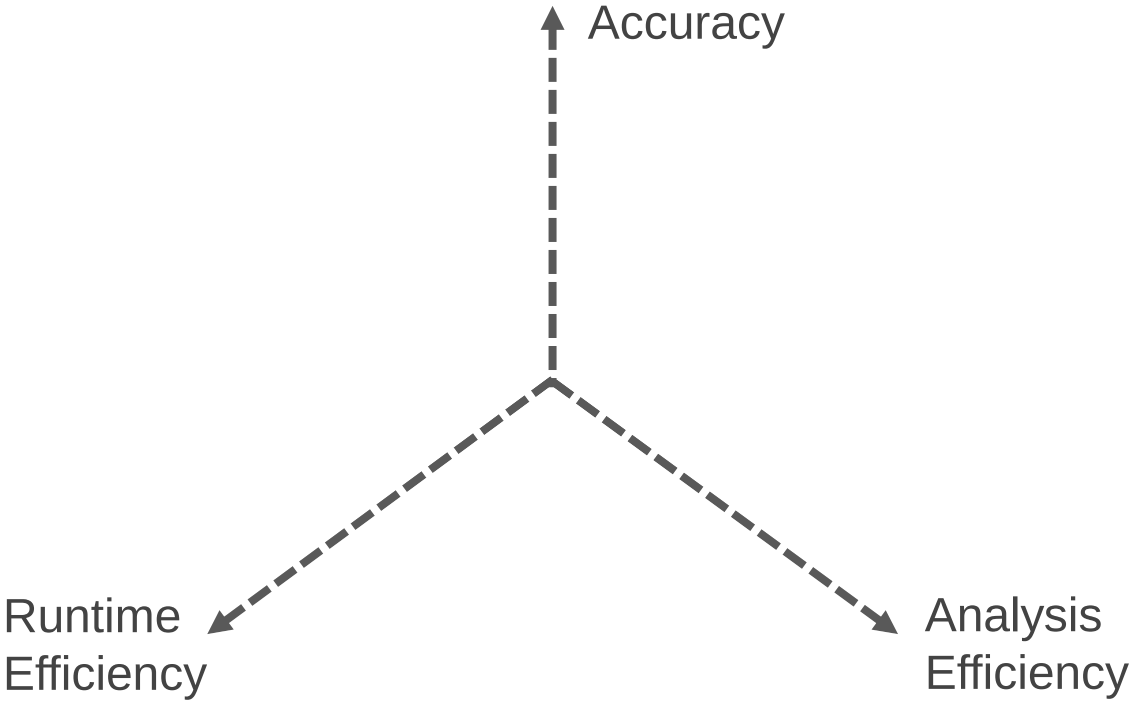


Related work

- System-call-based
  - DTrace, Protracer, LSM, Hi-Fi
- Dynamic Information Flow Tracking (DIFT)
  - Panorama, Dtracker
- DIFT + Record replay
  - Arnold

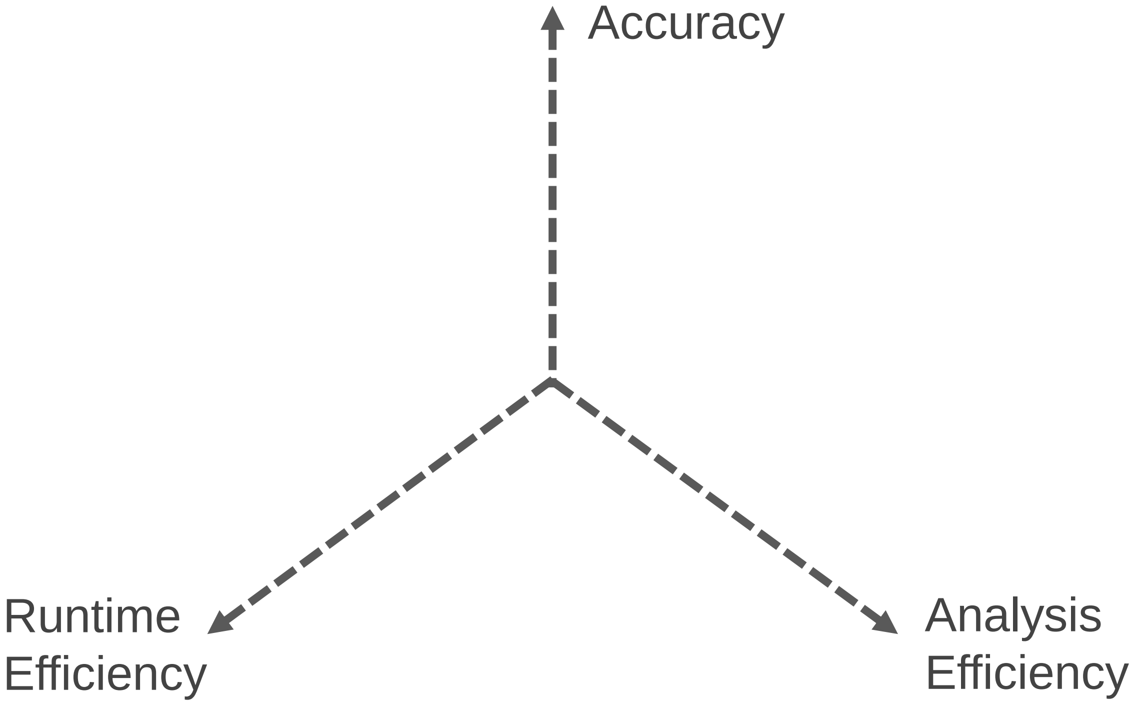


RAIN



RAIN

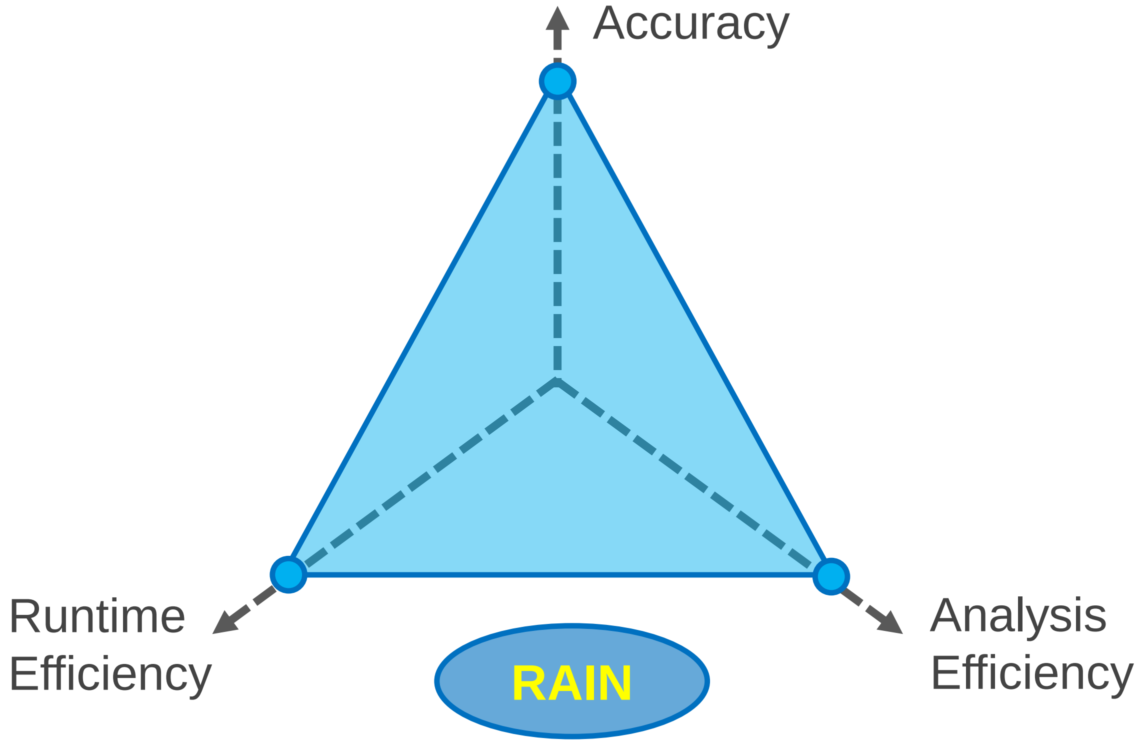
- We use
  - Record replay
  - Graph-based pruning
  - Selective DIFT



RAIN

- We use
  - Record replay
  - Graph-based pruning
  - Selective DIFT

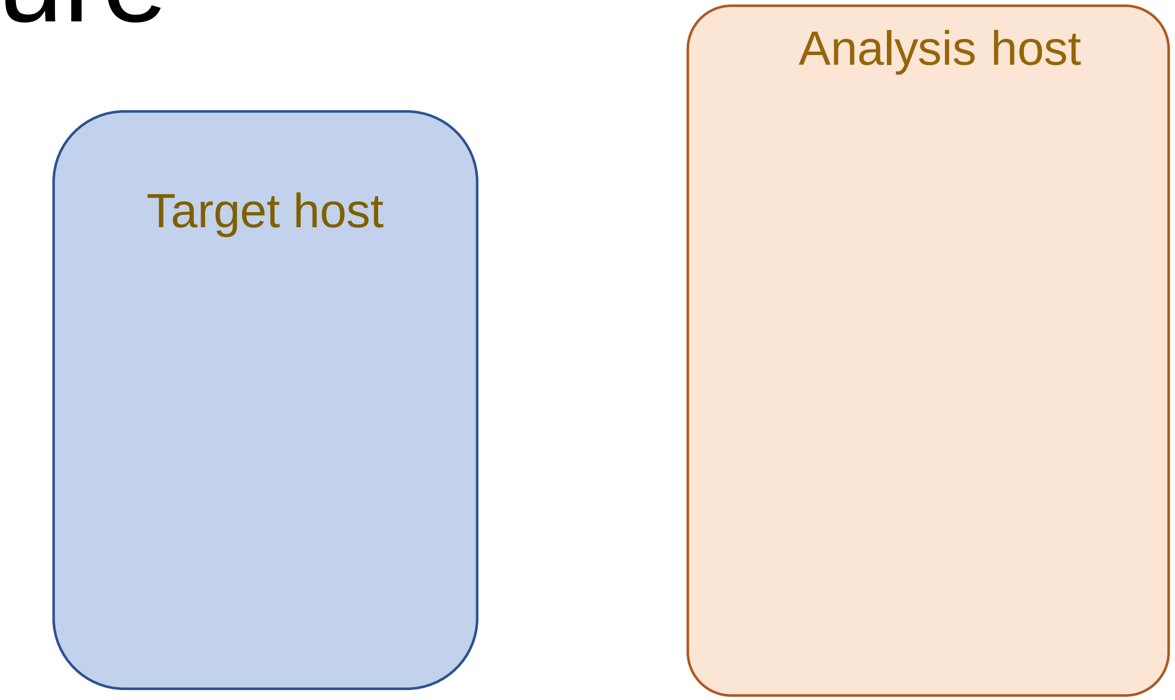
- We achieve
  - High accuracy
  - Runtime efficiency
  - Highly improved analysis efficiency



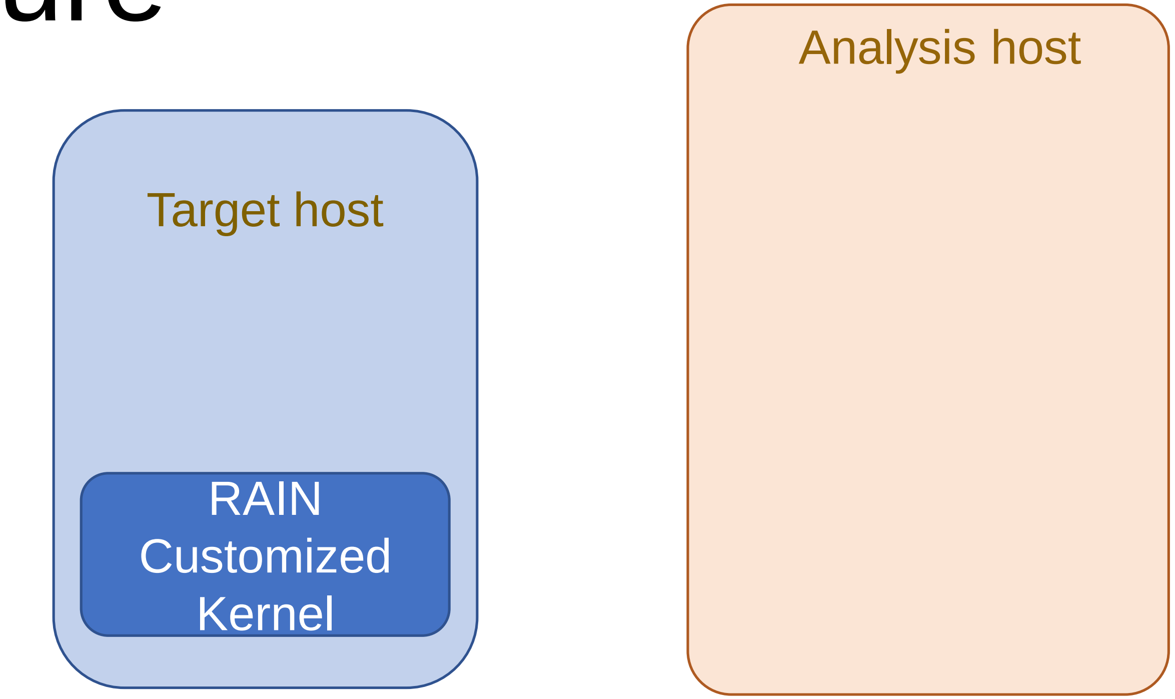
Threat model

- Trusts the OS
    - RAIN tracks user-level attacks.
- Tracks explicit channels
    - Side or covert channel is out of scope.
- Records all attacks from their inception
    - Hardware trojans or OS backdoor is out of scope.

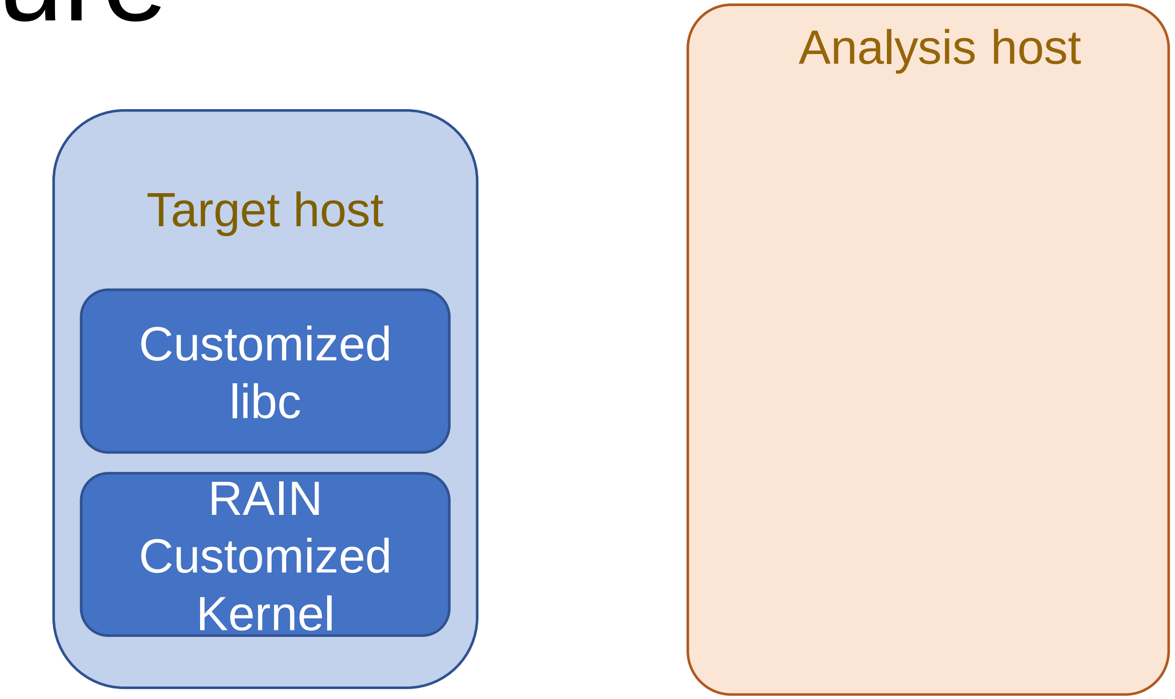
Architecture



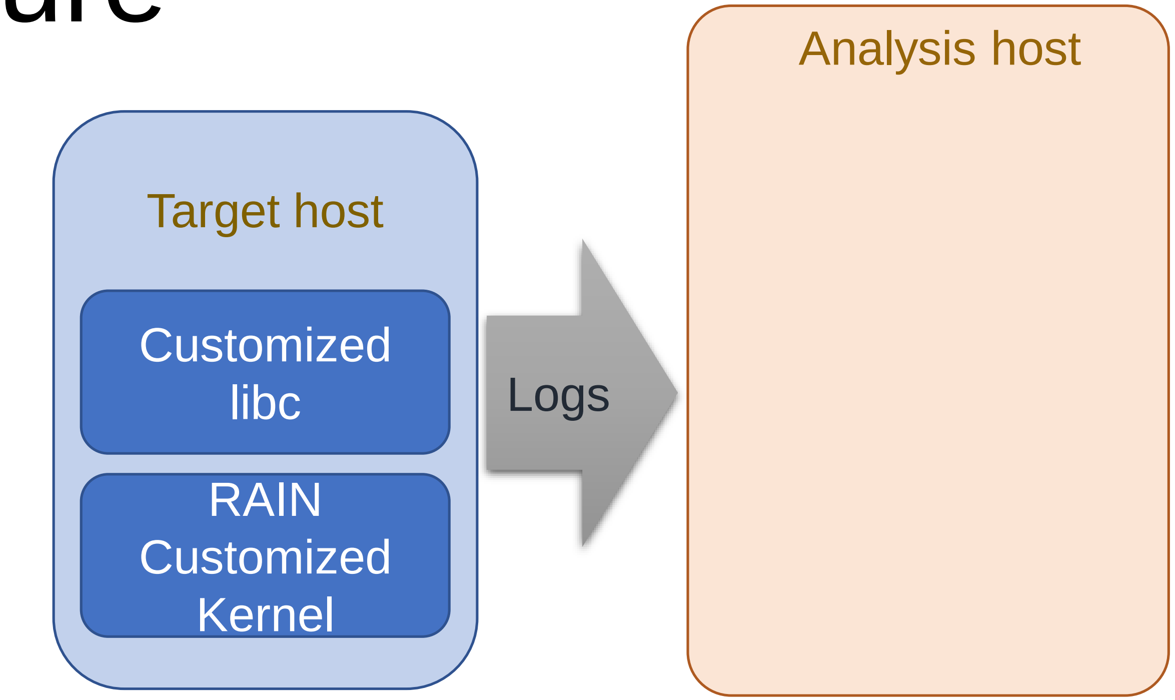
Architecture



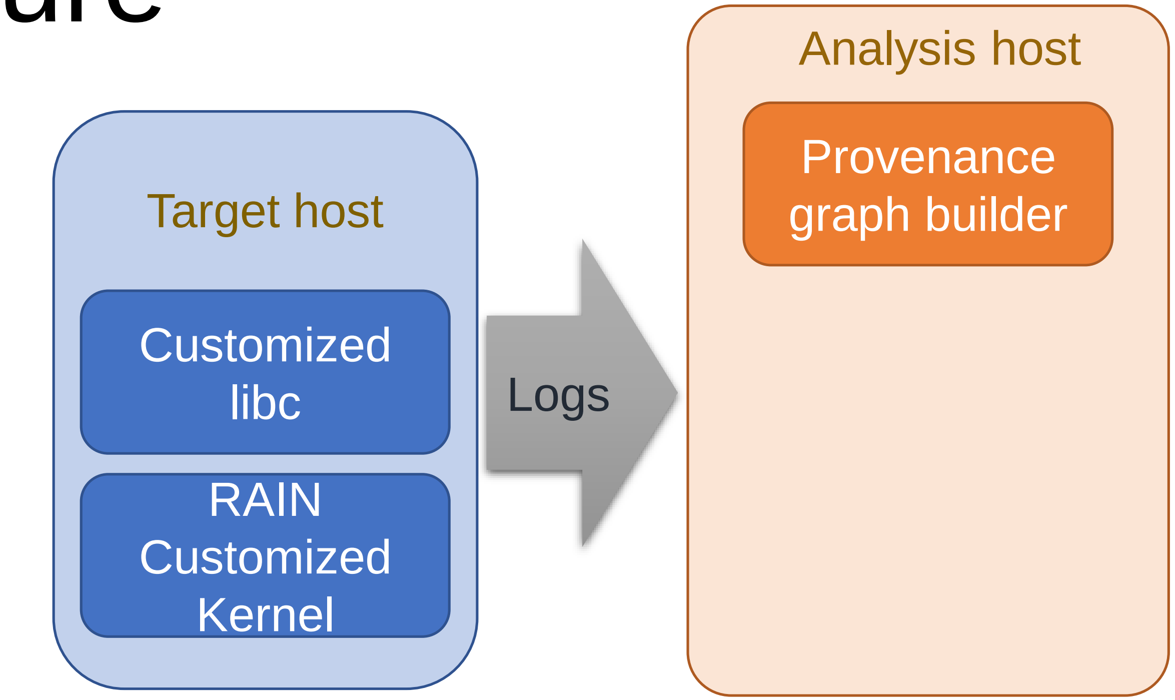
Architecture



Architecture



Architecture



Target host

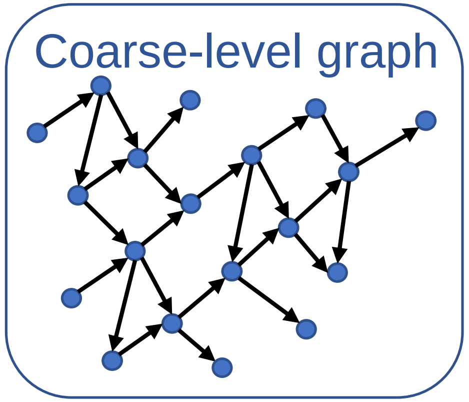
Customized libc

RAIN Customized Kernel

Logs

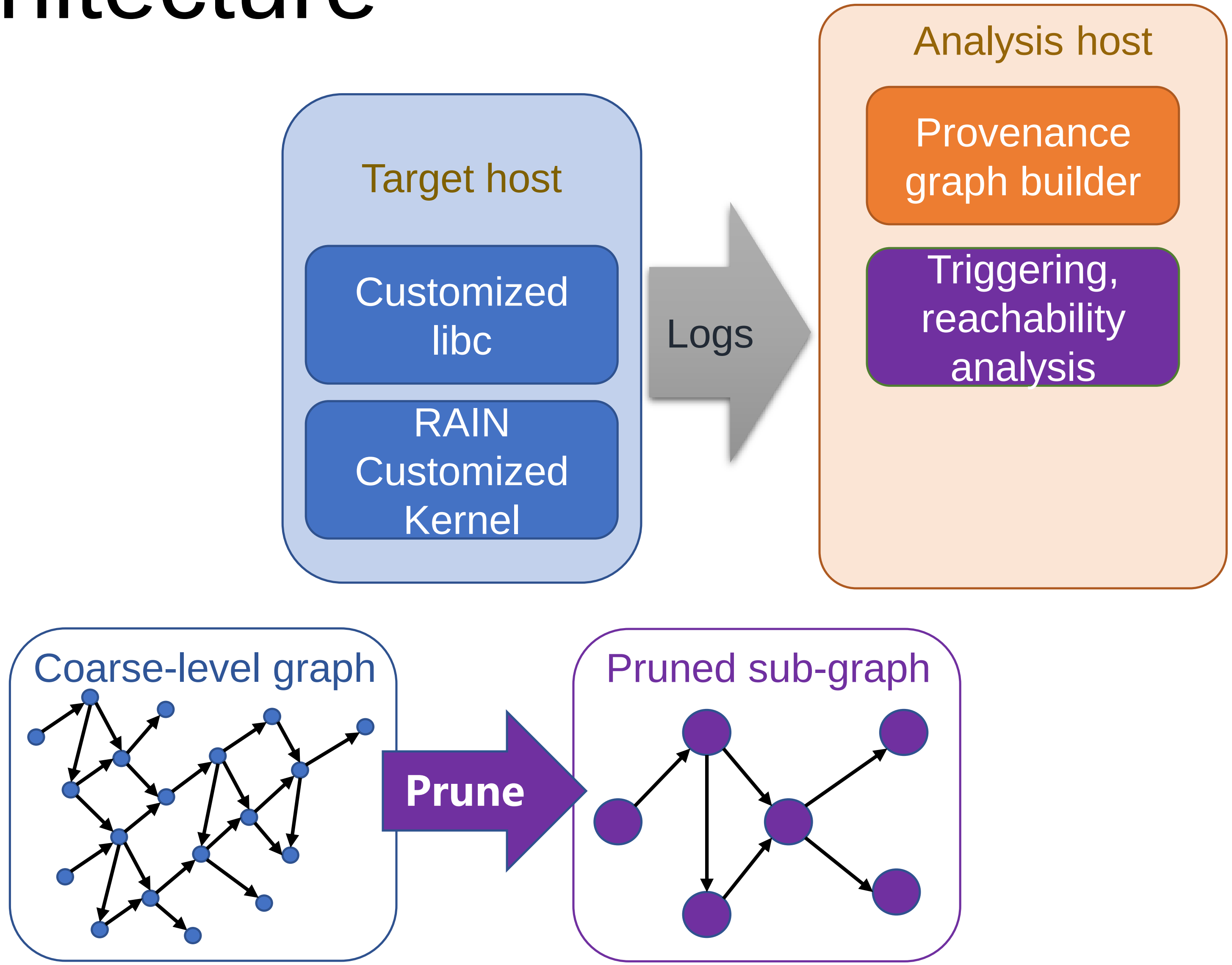
Analysis host

Provenance graph builder



Coarse-level graph

Architecture



Target host

Customized libc

RAIN Customized Kernel

Logs

Analysis host

Provenance graph builder

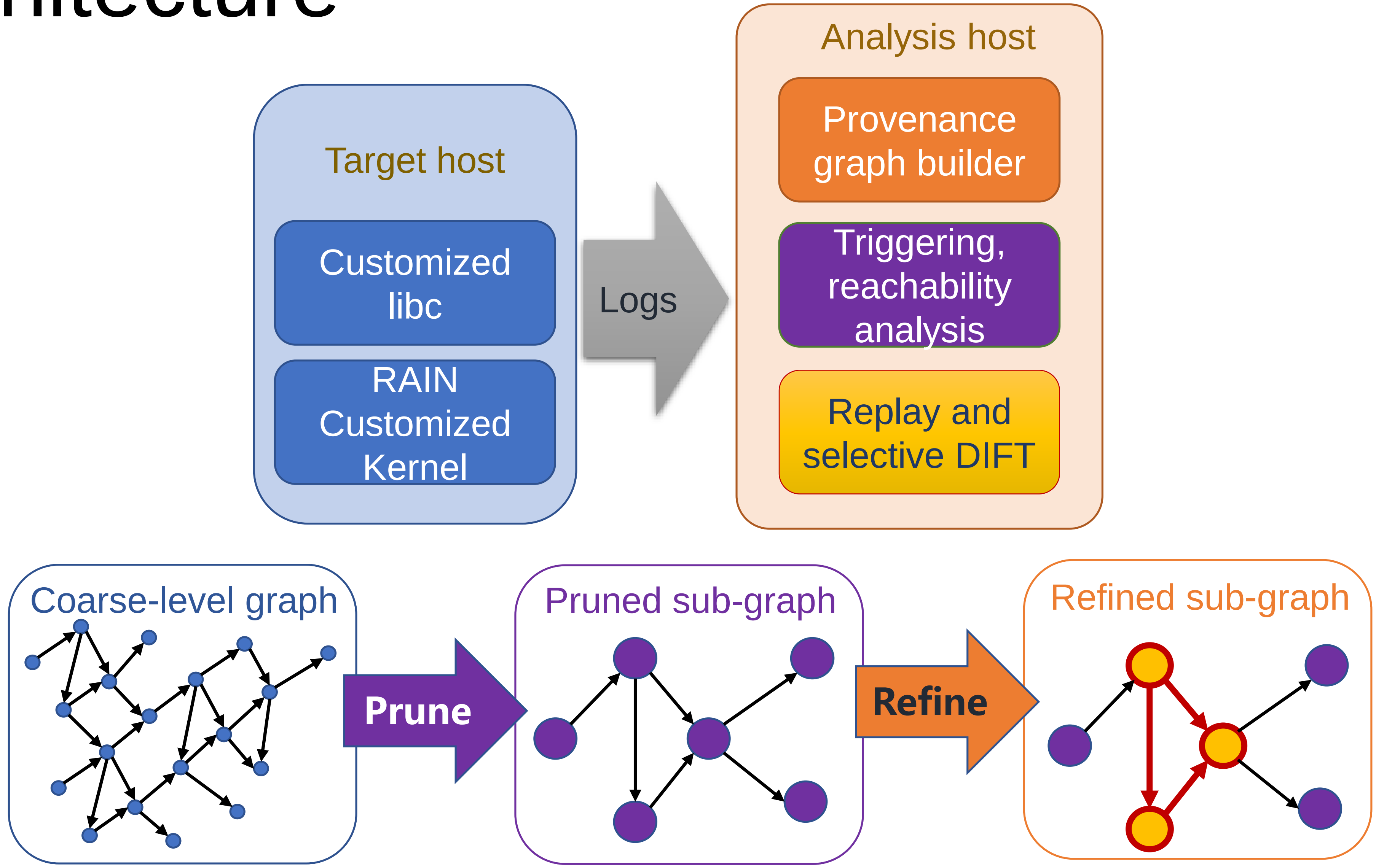
Triggering, reachability analysis

Coarse-level graph

Prune

Pruned sub-graph

Architecture



Target host

Customized libc

RAIN Customized Kernel

Logs

Analysis host

Provenance graph builder

Triggering, reachability analysis

Replay and selective DIFT

Coarse-level graph

Prune

Pruned sub-graph

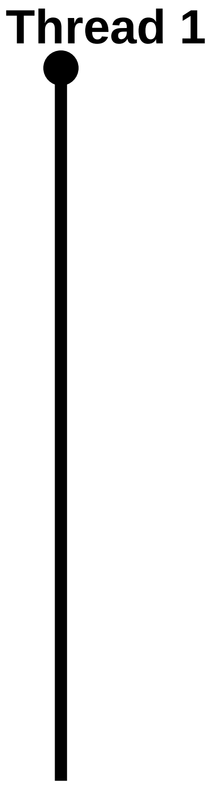
Refine

Refined sub-graph

OS-level record replay

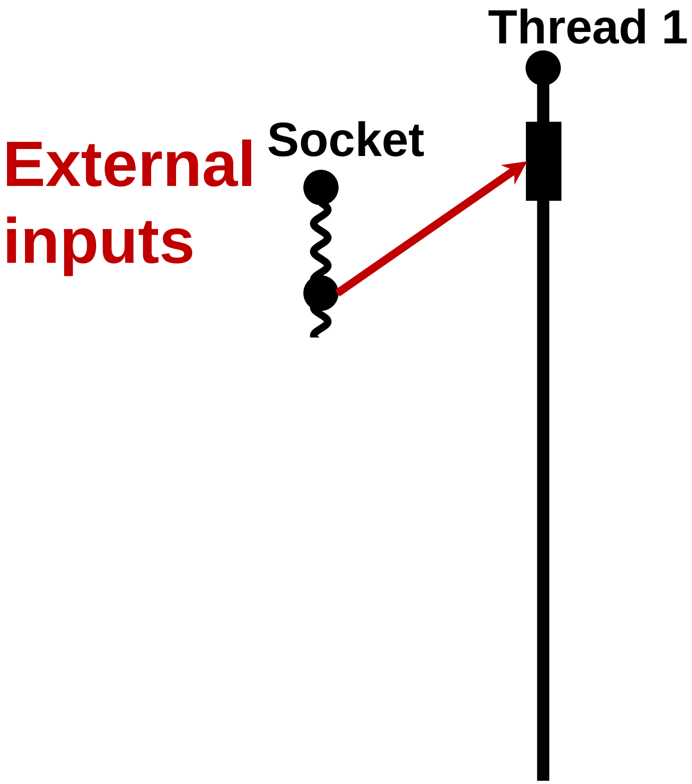
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



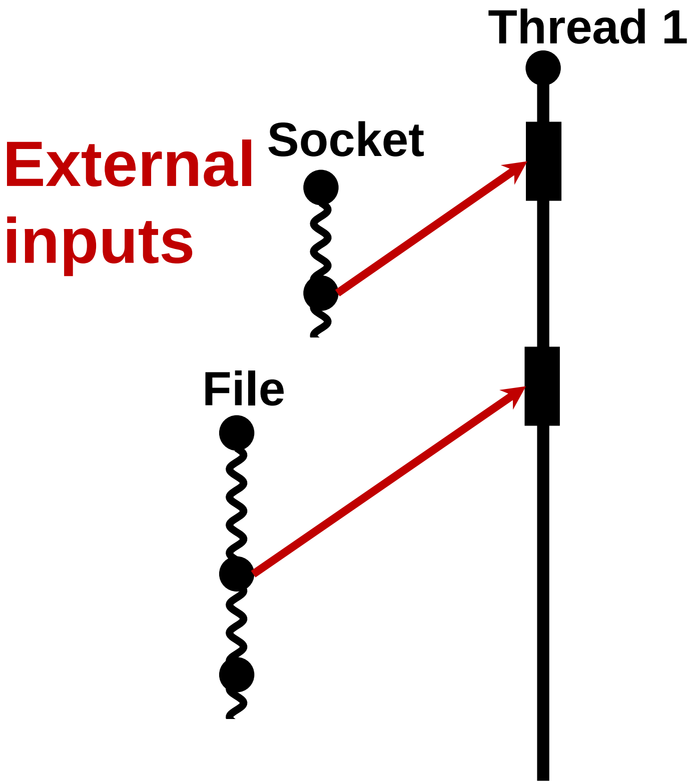
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



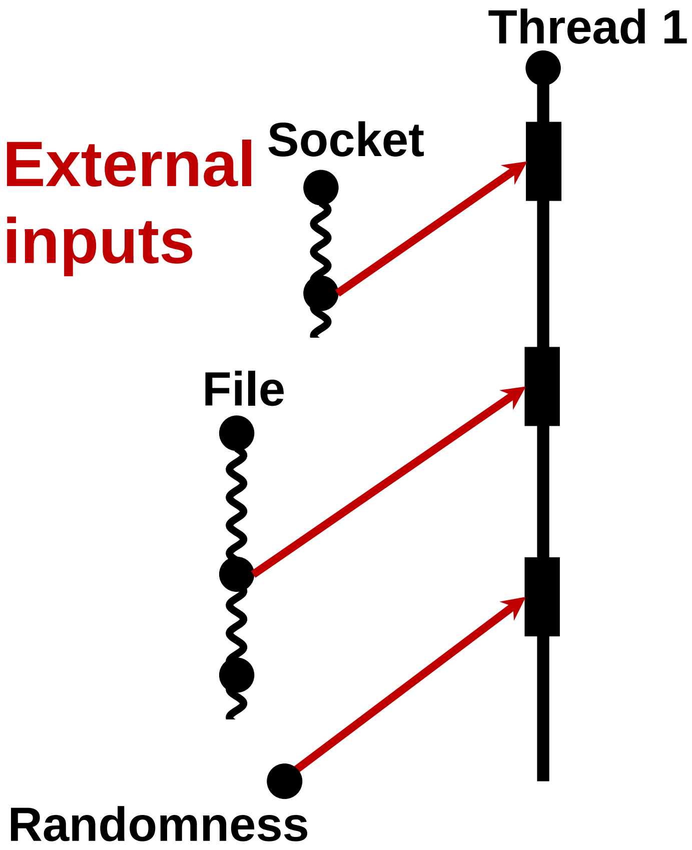
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



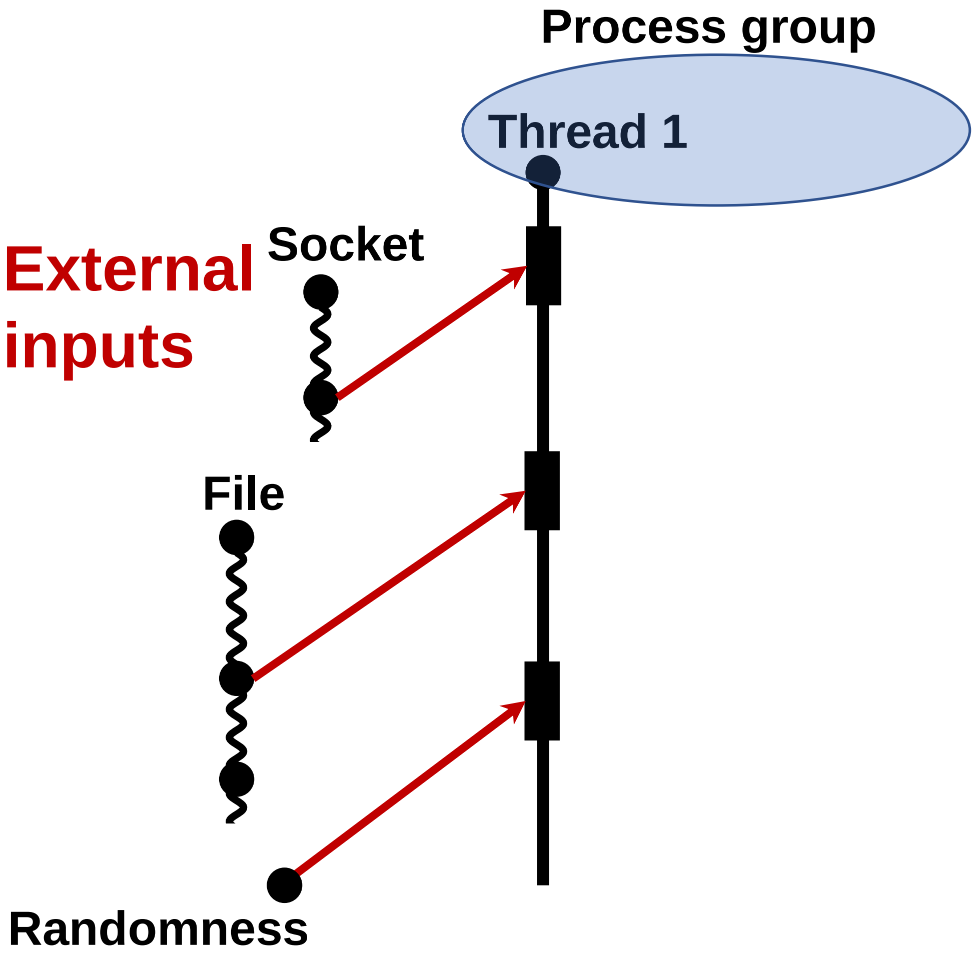
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



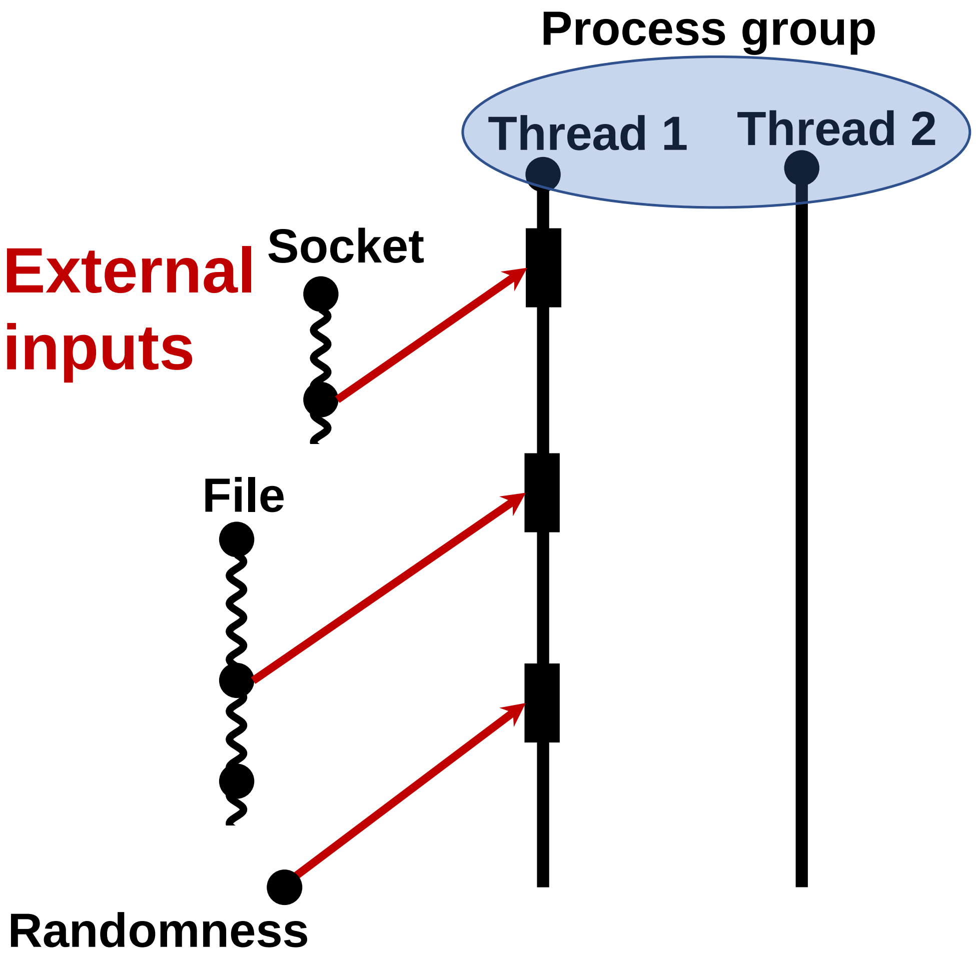
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



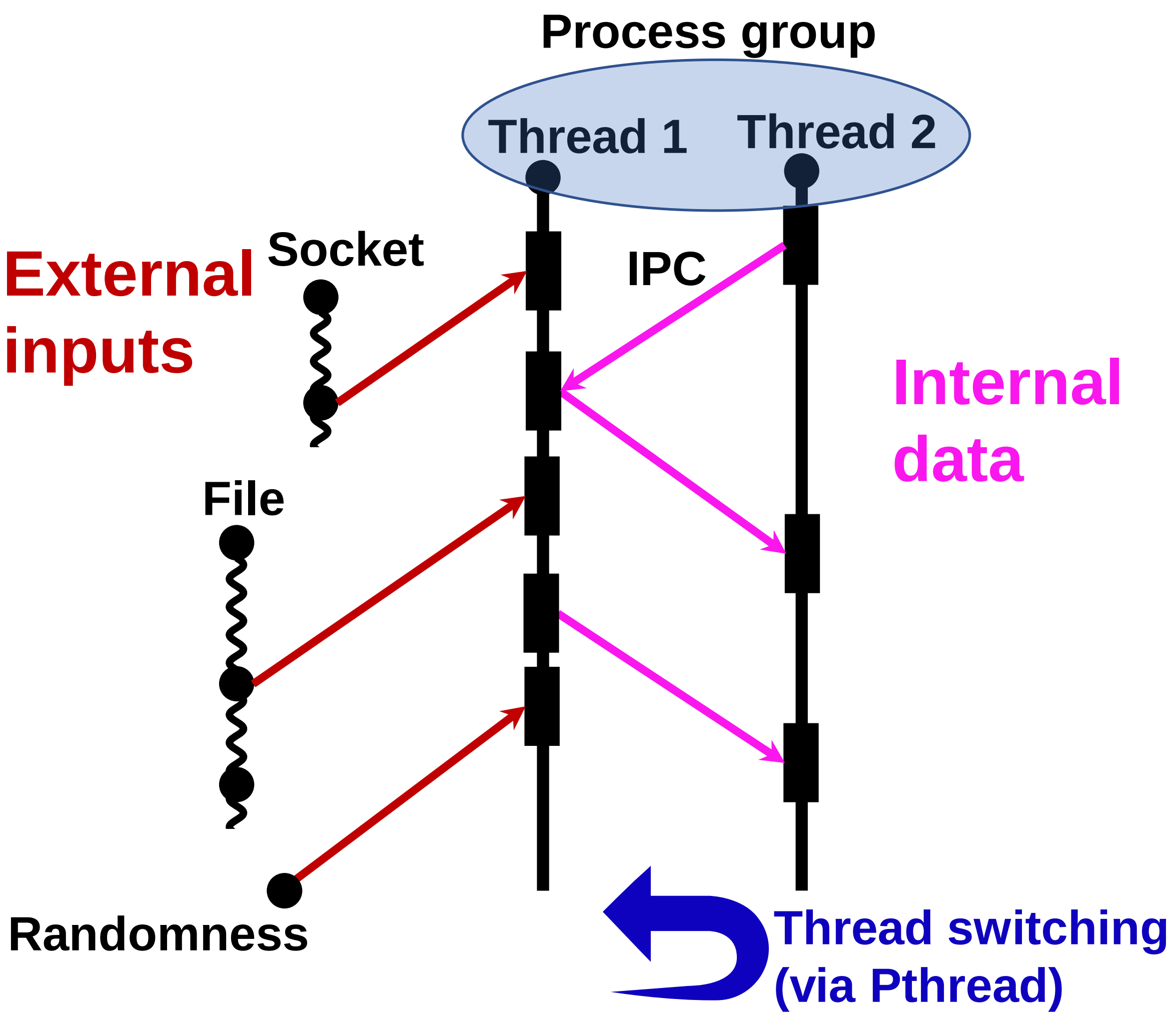
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



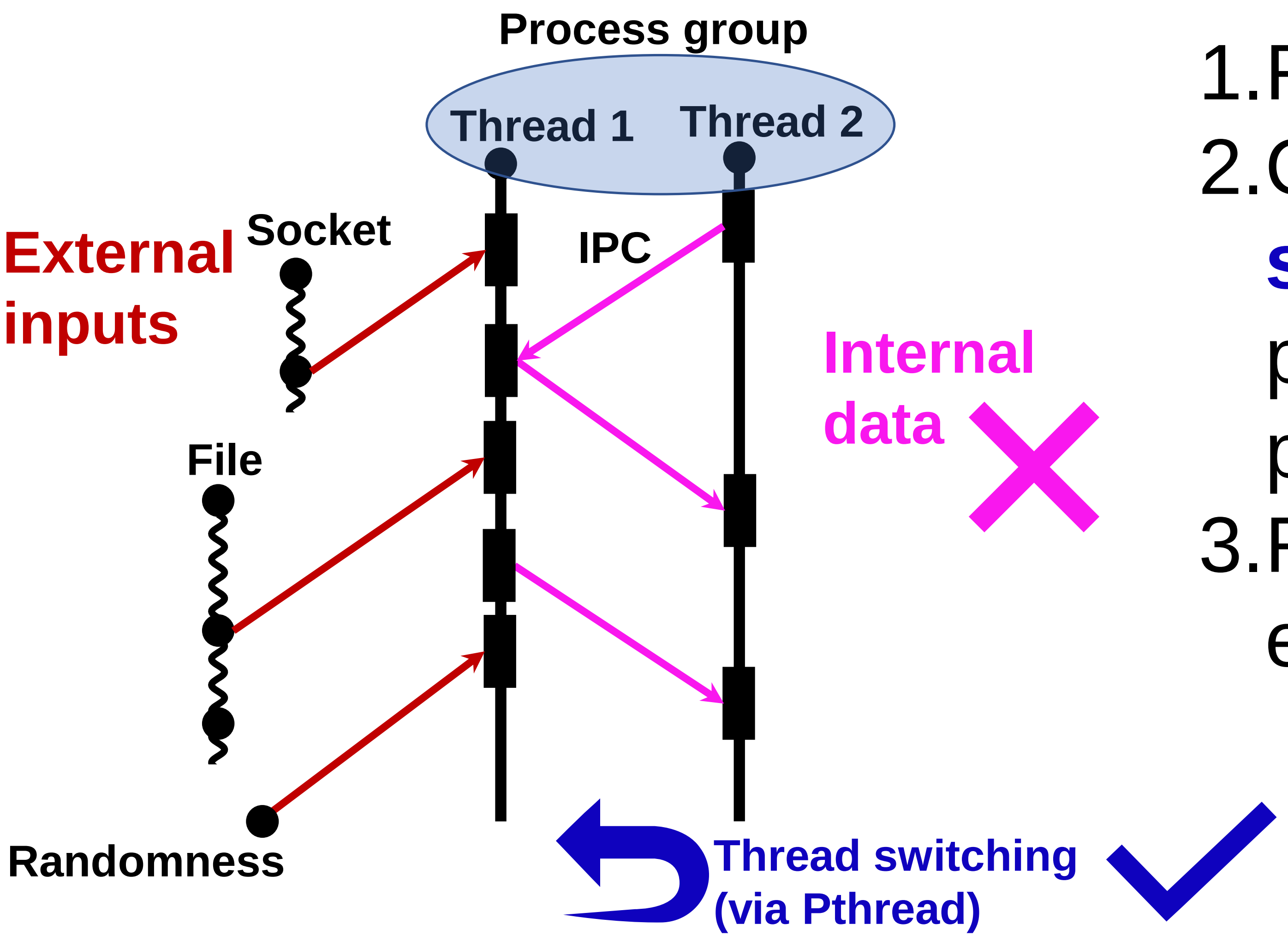
1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

OS-level record replay



1. Records **external inputs**
2. Captures the **thread switching** from the pthread interface, not the produced **internal data**
3. Records ***system-wide*** executions

Coarse-level logging and graph building

- Keeps logging system-call events
- Constructs a graph to represent:
  - the processes, files, and sockets as nodes
  - the events as causality edges

A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip

P1: /usr/bin/firefox

Coarse-level logging and graph building

- Keeps logging system-call events
- Constructs a graph to represent:
  - the processes, files, and sockets as nodes
  - the events as causality edges

A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip

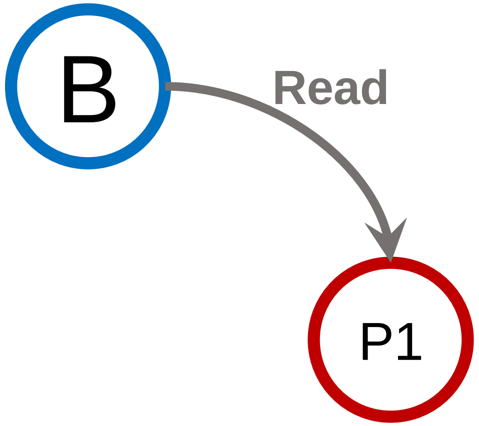
P1: /usr/bin/firefox

Coarse-level logging and graph building

- Keeps logging system-call events
- Constructs a graph to represent:
  - the processes, files, and sockets as nodes
  - the events as causality edges

A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip

P1: /usr/bin/firefox

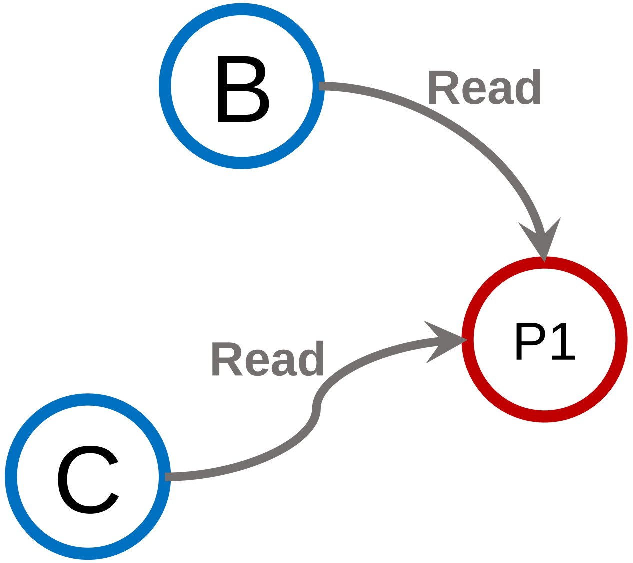


Coarse-level logging and graph building

- Keeps logging system-call events
- Constructs a graph to represent:
  - the processes, files, and sockets as nodes
  - the events as causality edges

A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip

P1: /usr/bin/firefox

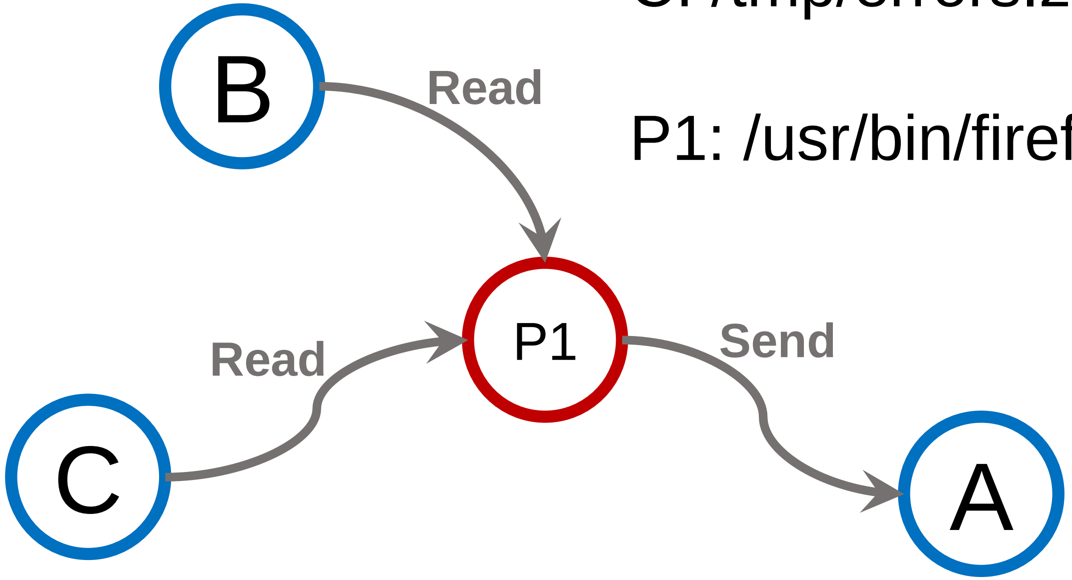


Coarse-level logging and graph building

- Keeps logging system-call events
- Constructs a graph to represent:
  - the processes, files, and sockets as nodes
  - the events as causality edges

A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip

P1: /usr/bin/firefox



- *Does every recorded execution need replay and DIFT?*

- *Does every recorded execution need replay and DIFT?* **No!**

Pruning

- *Does every recorded execution need replay and DIFT?* **No!**
- Prunes the data in the graph based on trigger analysis results
  - Upstream
  - Downstream
  - Point-to-point
  - Interference

Upstream

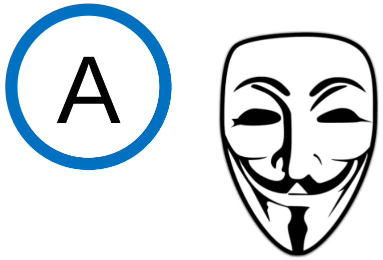
A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip
D: /docs/ctct1.csv
E: /docs/ctct2.pdf
F: /docs/loss.csv

P1: /usr/bin/firefox
P2: /usr/bin/TextEditor
P3: /bin/gzip

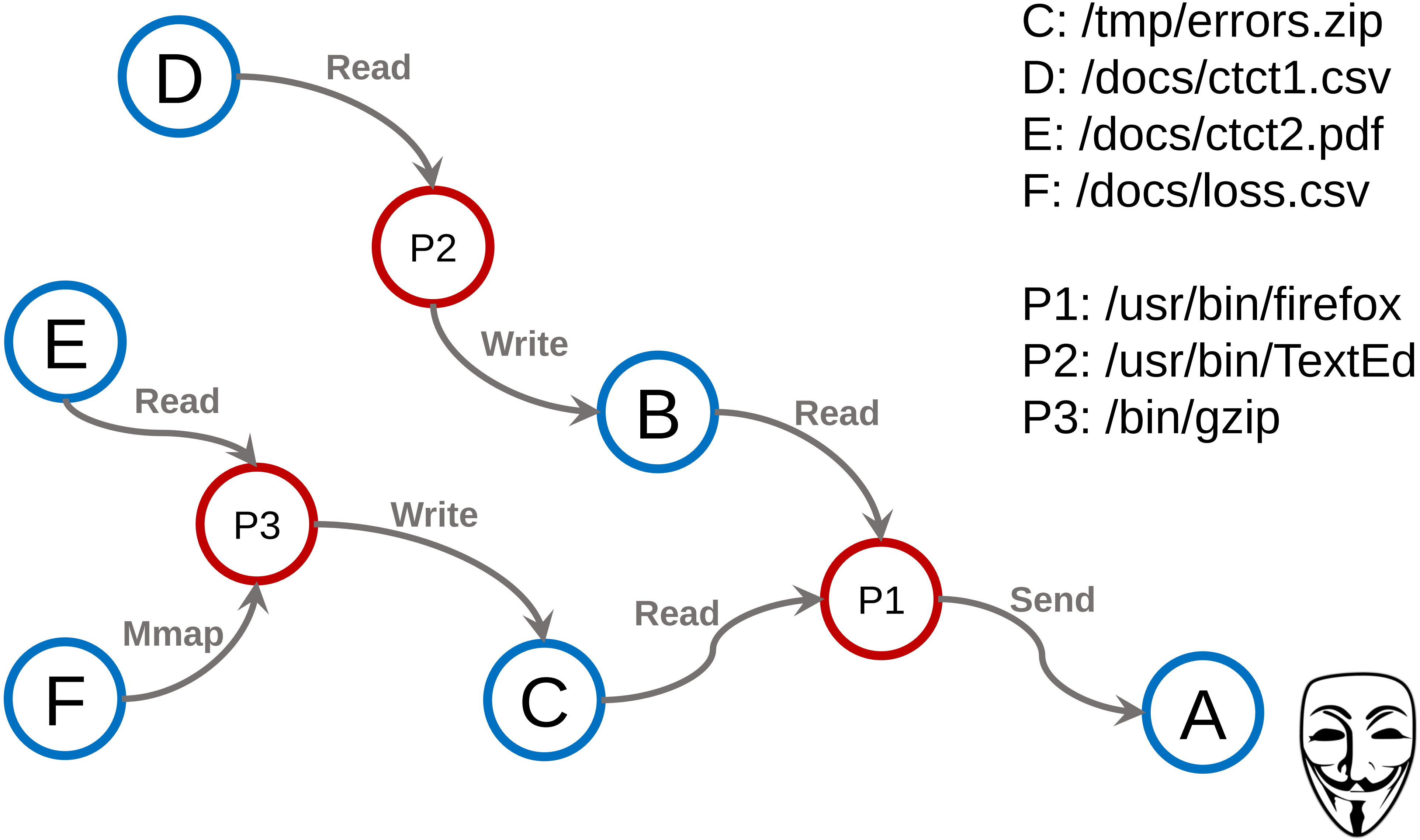
Upstream

A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip
D: /docs/ctct1.csv
E: /docs/ctct2.pdf
F: /docs/loss.csv

P1: /usr/bin/firefox
P2: /usr/bin/TextEditor
P3: /bin/gzip



Upstream



A: Attacker site
B: /docs/report.doc
C: /tmp/errors.zip
D: /docs/ctct1.csv
E: /docs/ctct2.pdf
F: /docs/loss.csv

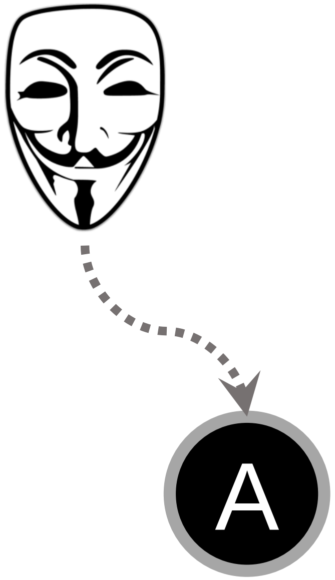
P1: /usr/bin/firefox
P2: /usr/bin/TextEditor
P3: /bin/gzip

Downstream

A: **Tampered file** /docs/ctct.csv
B: Seasonal report docs/s1.csv
C: Seasonal report docs/s2.csv
D: Budget report docs/bgt.csv
E: Half-year report docs/h2.pdf

P1: Spreadsheet editor
P2: Auto-budget program
P3: Auto-report program

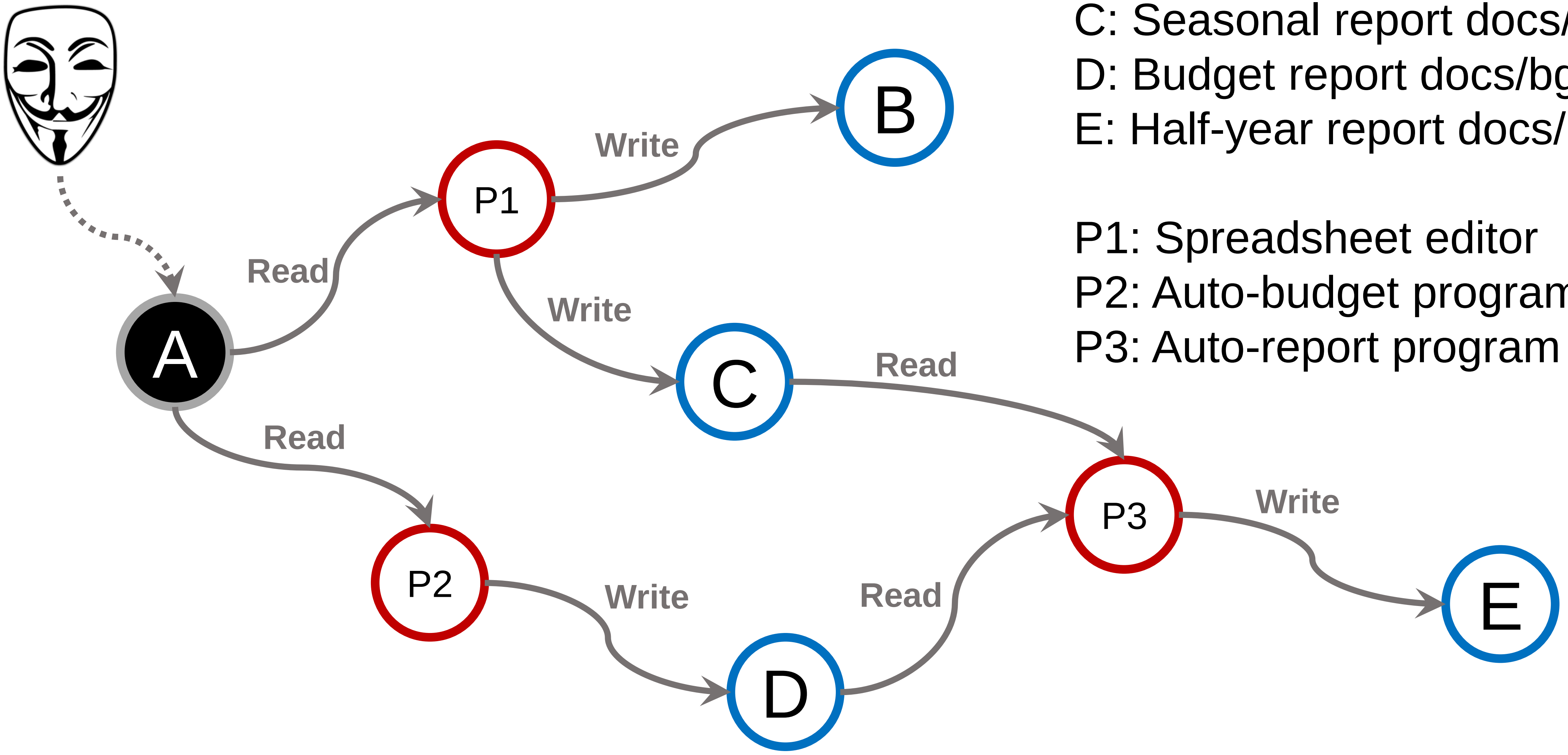
Downstream



A: **Tampered file** /docs/ctct.csv
B: Seasonal report docs/s1.csv
C: Seasonal report docs/s2.csv
D: Budget report docs/bgt.csv
E: Half-year report docs/h2.pdf

P1: Spreadsheet editor
P2: Auto-budget program
P3: Auto-report program

Downstream



A: **Tampered file** /docs/ctct.csv
B: Seasonal report docs/s1.csv
C: Seasonal report docs/s2.csv
D: Budget report docs/bgt.csv
E: Half-year report docs/h2.pdf

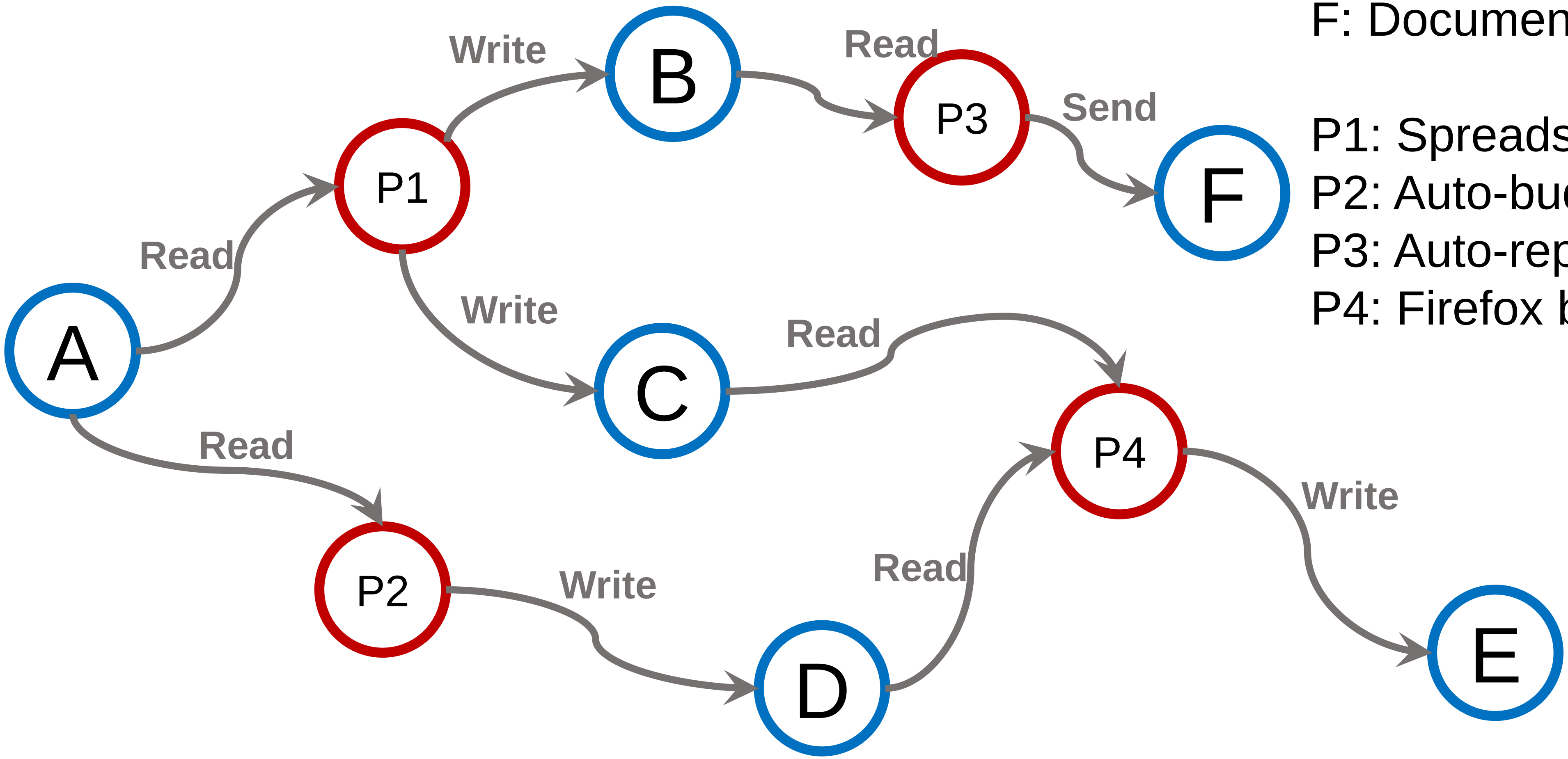
P1: Spreadsheet editor
P2: Auto-budget program
P3: Auto-report program

Point-to-point

A: **Tampered file** /docs/ctct.csv
B: Seasonal report docs/s1.csv
C: Seasonal report docs/s2.csv
D: Budget report docs/bgt.csv
E: Half-year report docs/h2.pdf
F: Document archive server

P1: Spreadsheet editor
P2: Auto-budget program
P3: Auto-report program
P4: Firefox browser

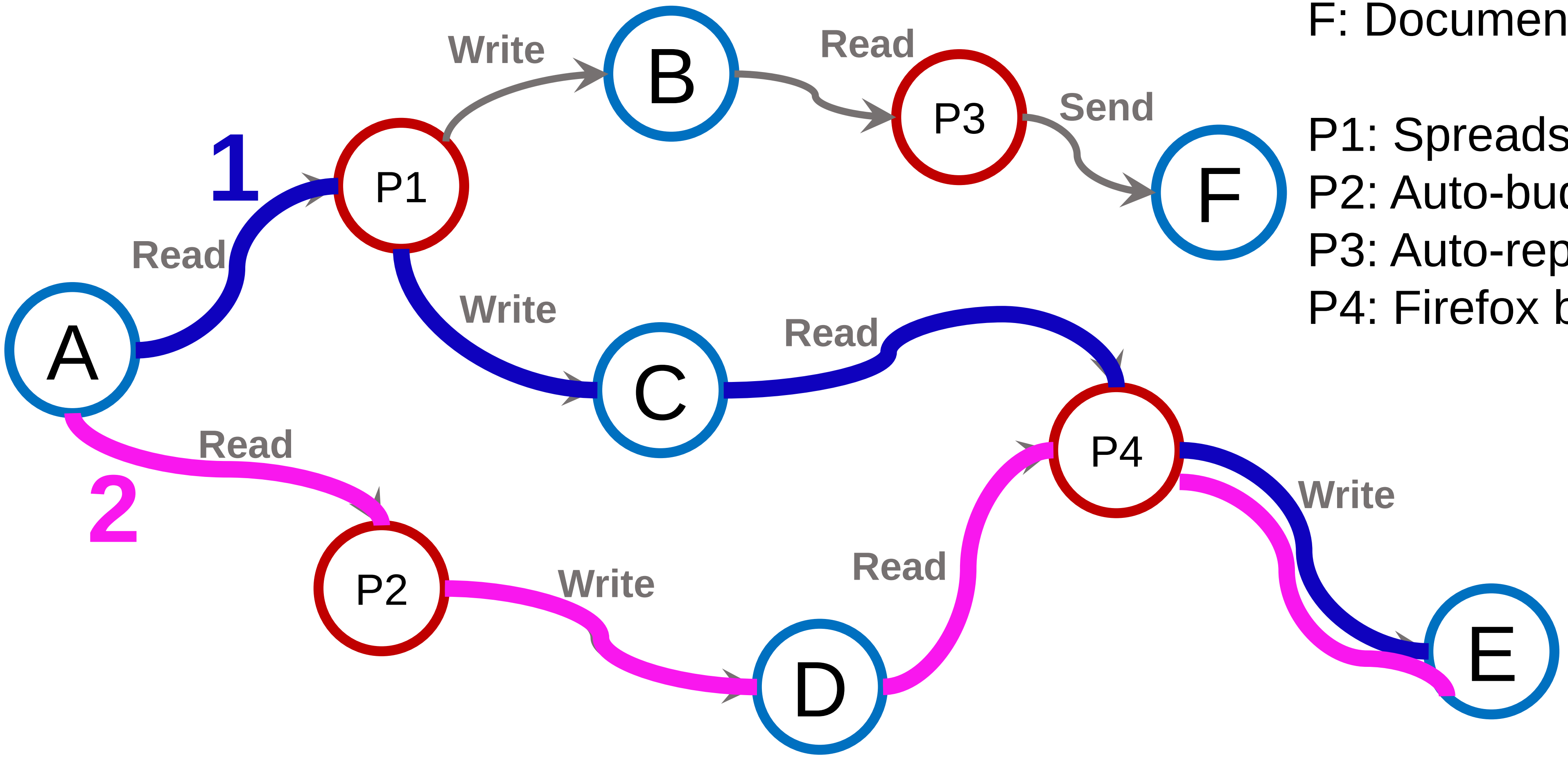
Point-to-point



A: **Tampered file** /docs/ctct.csv
B: Seasonal report docs/s1.csv
C: Seasonal report docs/s2.csv
D: Budget report docs/bgt.csv
E: Half-year report docs/h2.pdf
F: Document archive server

P1: Spreadsheet editor
P2: Auto-budget program
P3: Auto-report program
P4: Firefox browser

Point-to-point



A: **Tampered file** /docs/ctct.csv

B: Seasonal report docs/s1.csv

C: Seasonal report docs/s2.csv

D: Budget report docs/bgt.csv

E: Half-year report docs/h2.pdf

F: Document archive server

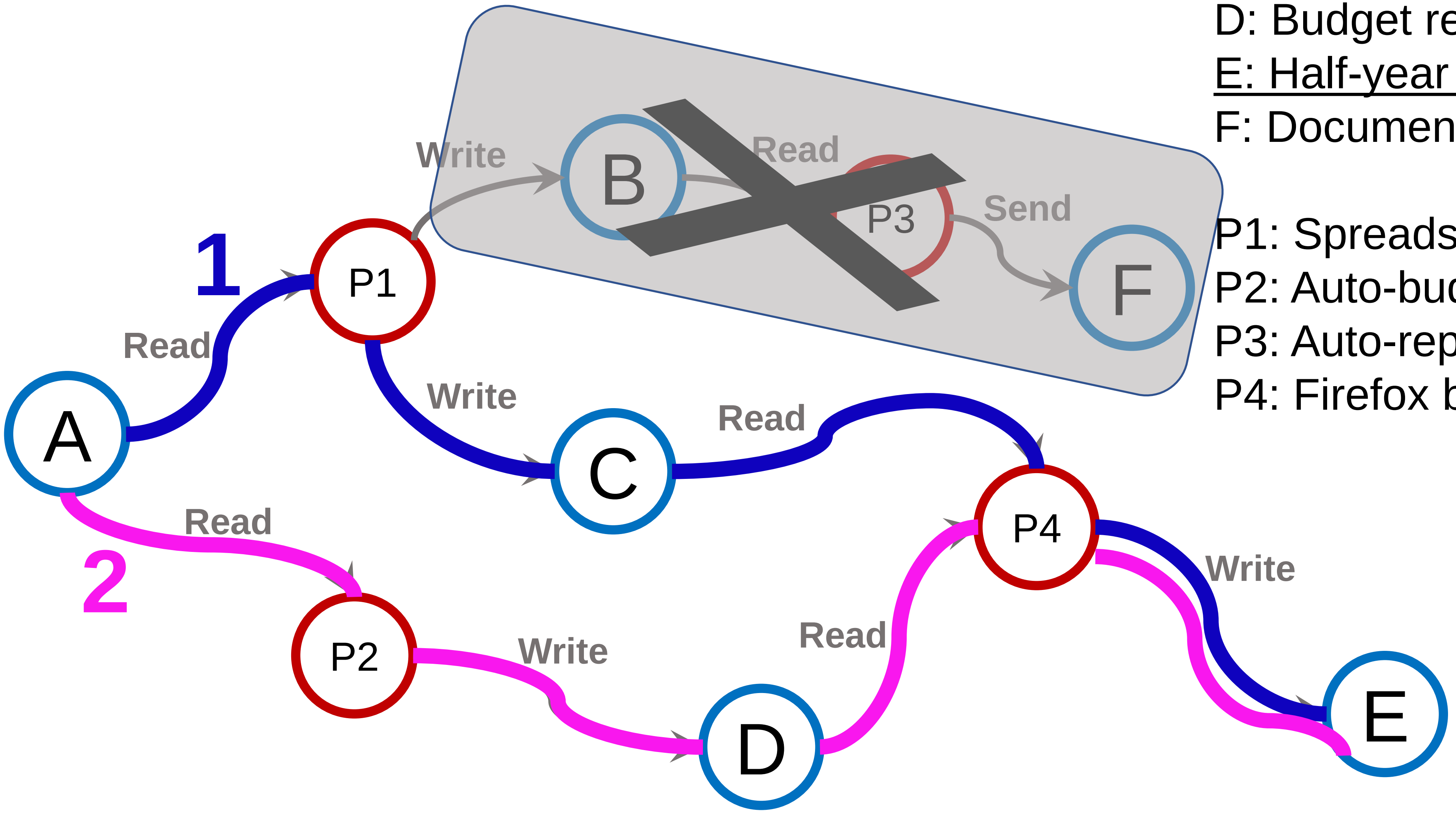
P1: Spreadsheet editor

P2: Auto-budget program

P3: Auto-report program

P4: Firefox browser

Point-to-point



A: **Tampered file** /docs/ctct.csv

B: Seasonal report docs/s1.csv

C: Seasonal report docs/s2.csv

D: Budget report docs/bgt.csv

E: Half-year report docs/h2.pdf

F: Document archive server

P1: Spreadsheet editor

P2: Auto-budget program

P3: Auto-report program

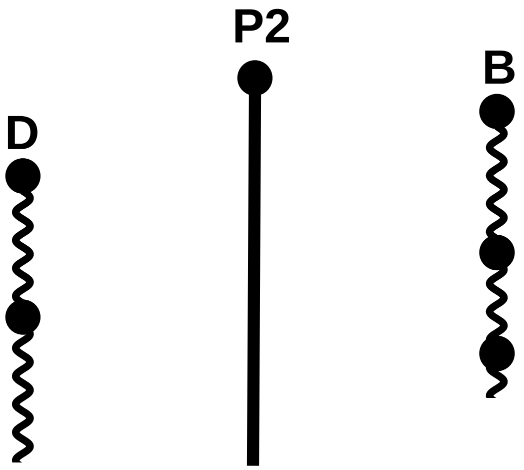
P4: Firefox browser

Interference

- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
    - We determine interference according to the time order of inbound and outbound IO events.

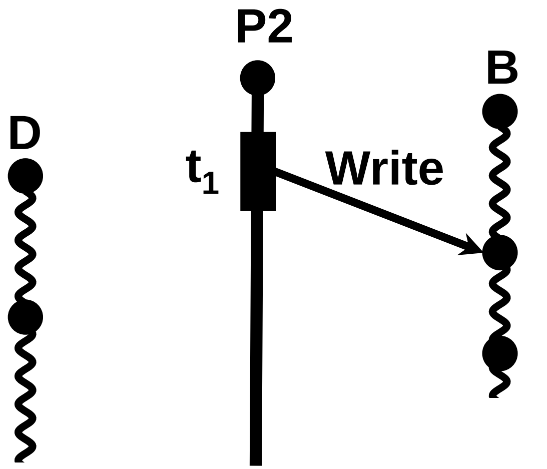
Interference

- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
    - We determine interference according to the time order of inbound and outbound IO events.



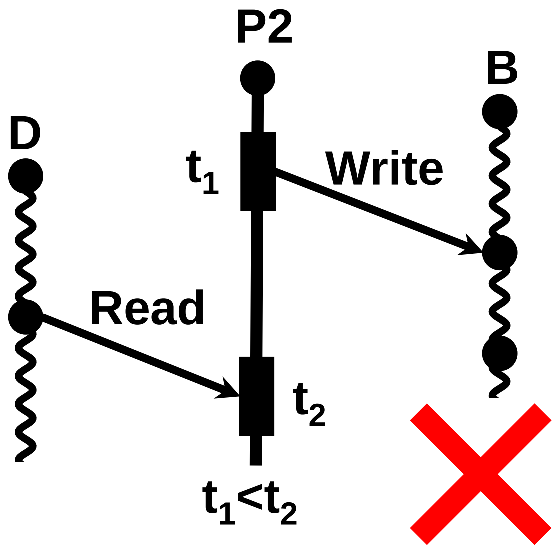
Interference

- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
    - We determine interference according to the time order of inbound and outbound IO events.



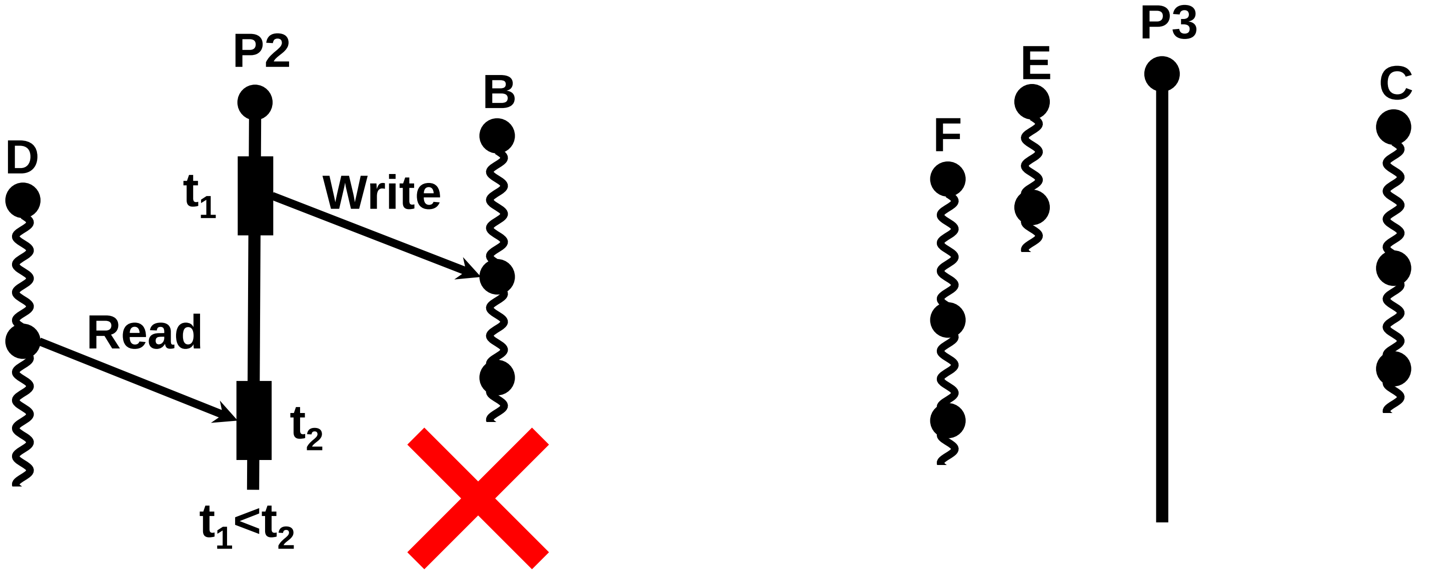
Interference

- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
  - We determine interference according to the time order of inbound and outbound IO events.



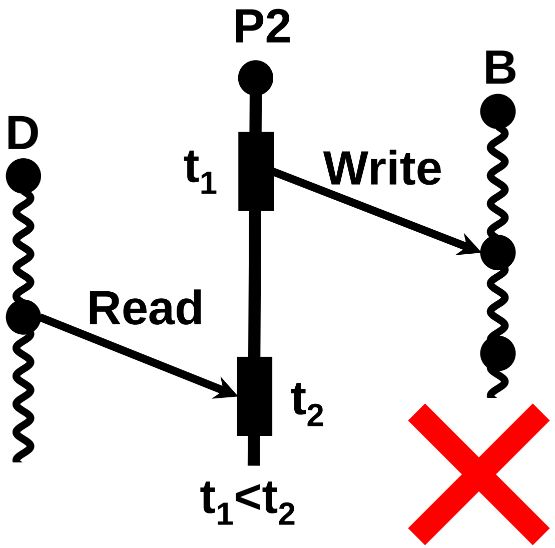
Interference

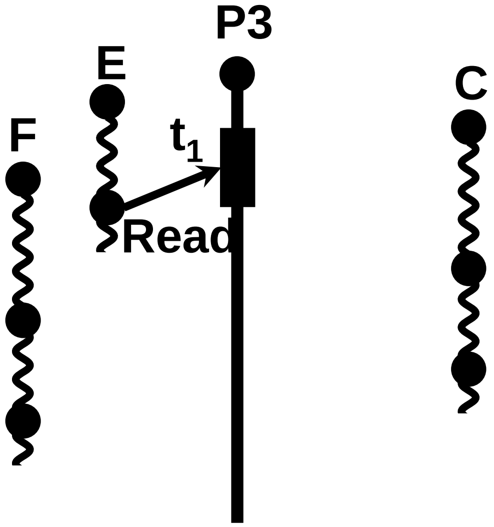
- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
    - We determine interference according to the time order of inbound and outbound IO events.



Interference

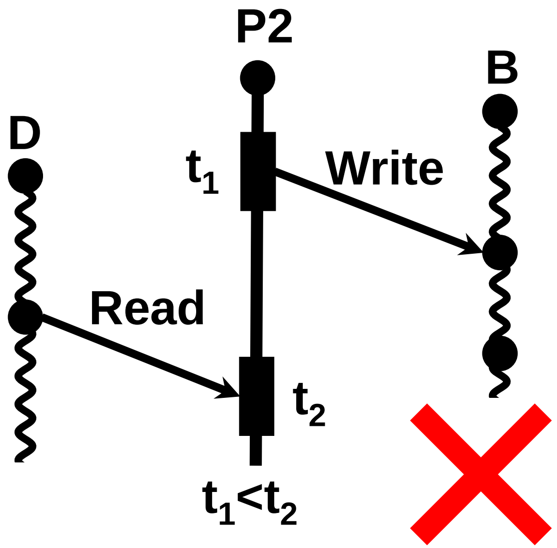
- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
  - We determine interference according to the time order of inbound and outbound IO events.

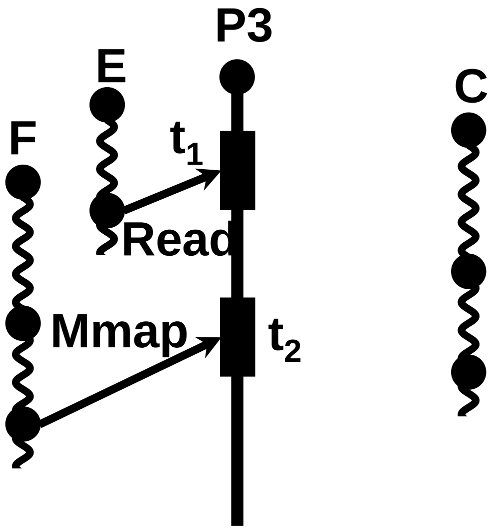




Interference

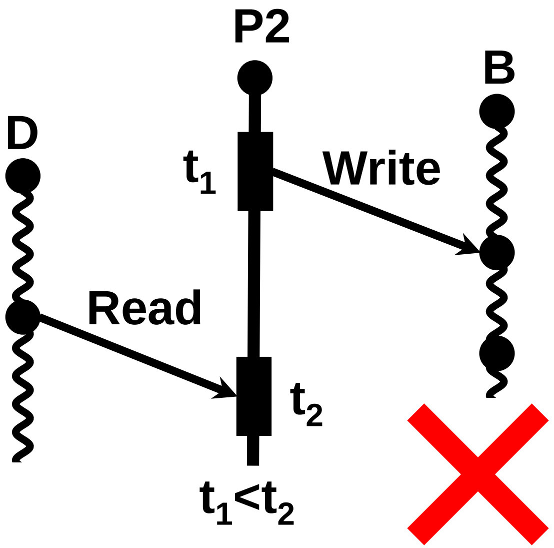
- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
  - We determine interference according to the time order of inbound and outbound IO events.

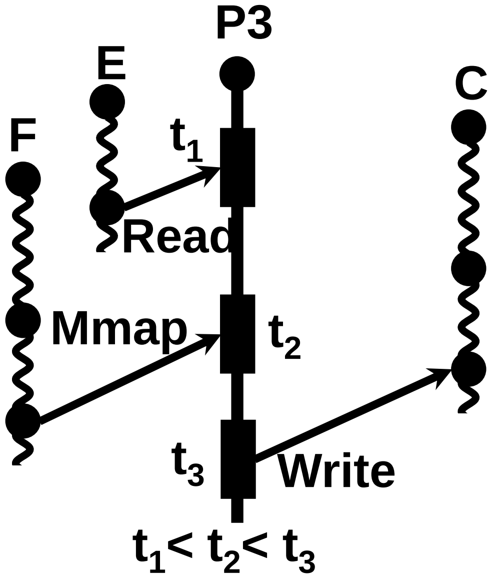




Interference

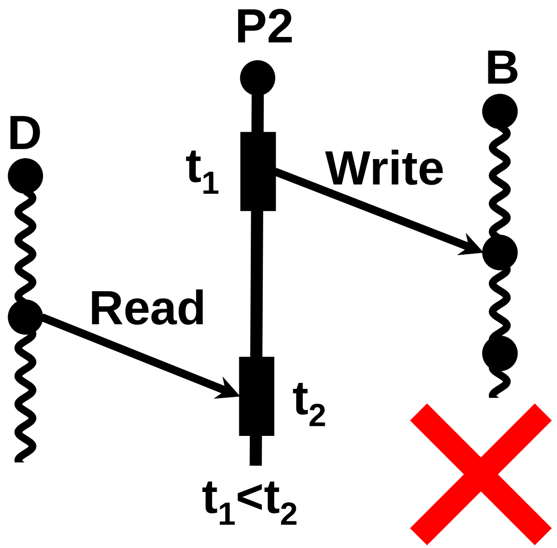
- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
  - We determine interference according to the time order of inbound and outbound IO events.

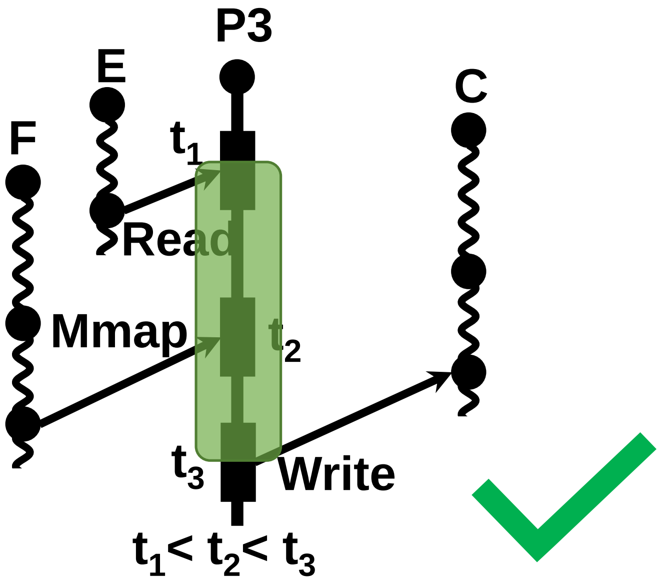




Interference

- Insight: only inbound and outbound files that interfere in a process will possibly produce causality.
  - We determine interference according to the time order of inbound and outbound IO events.

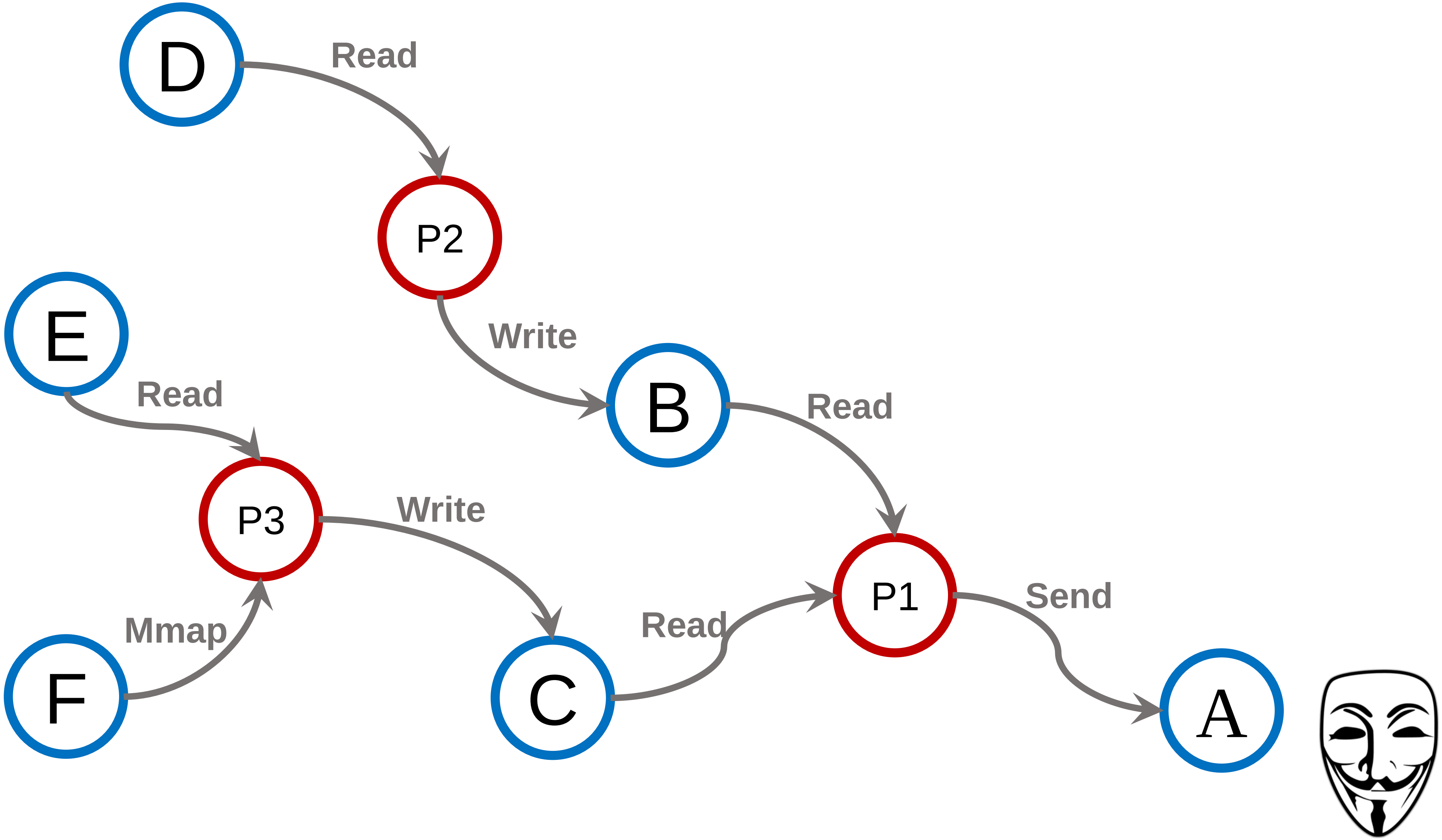




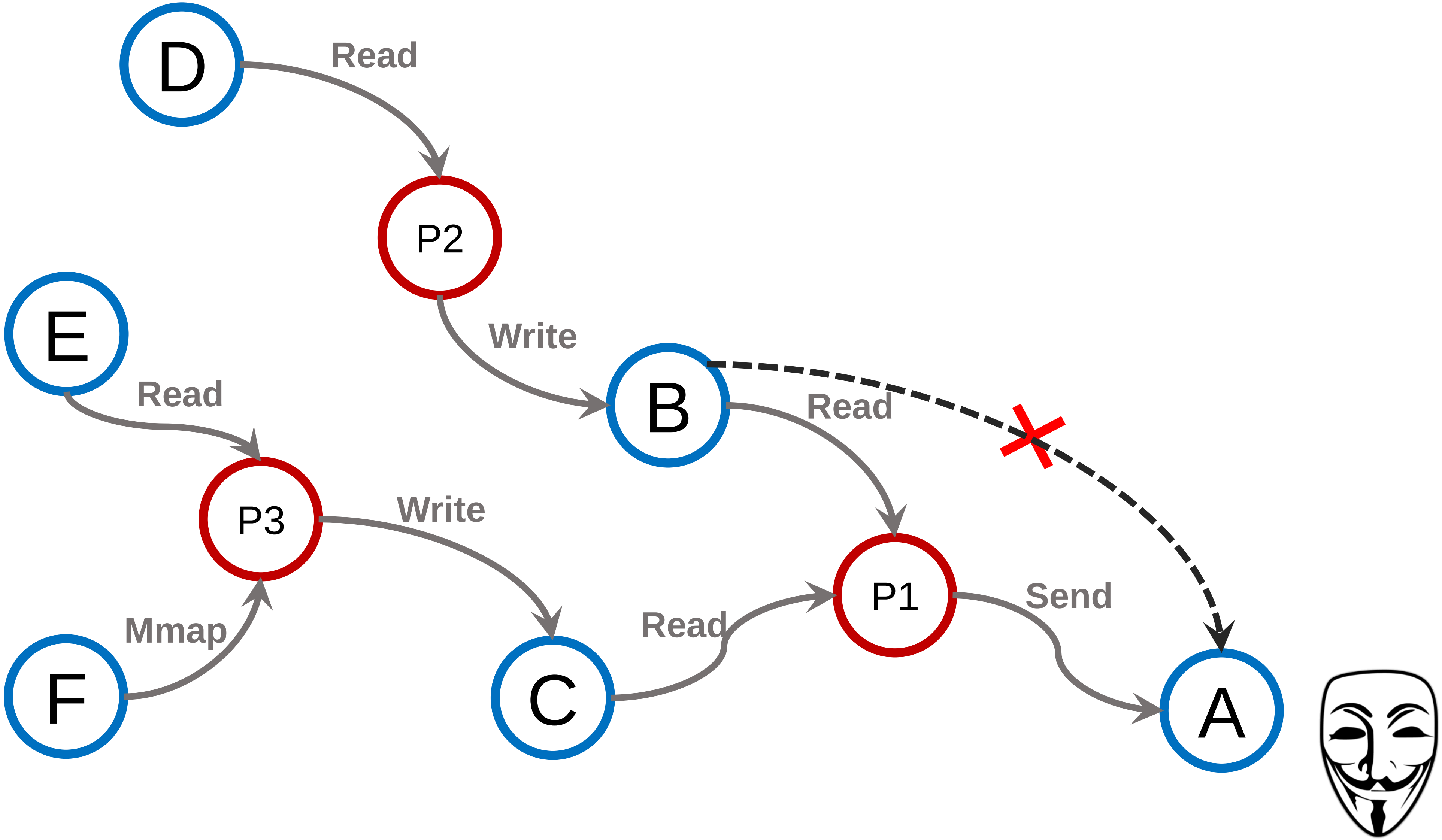
Refinement - selective DIFT

- Replays and conducts DIFT to the necessary part of the execution
  - Aggregation
  - Upstream
  - Downstream
  - Point-to-point

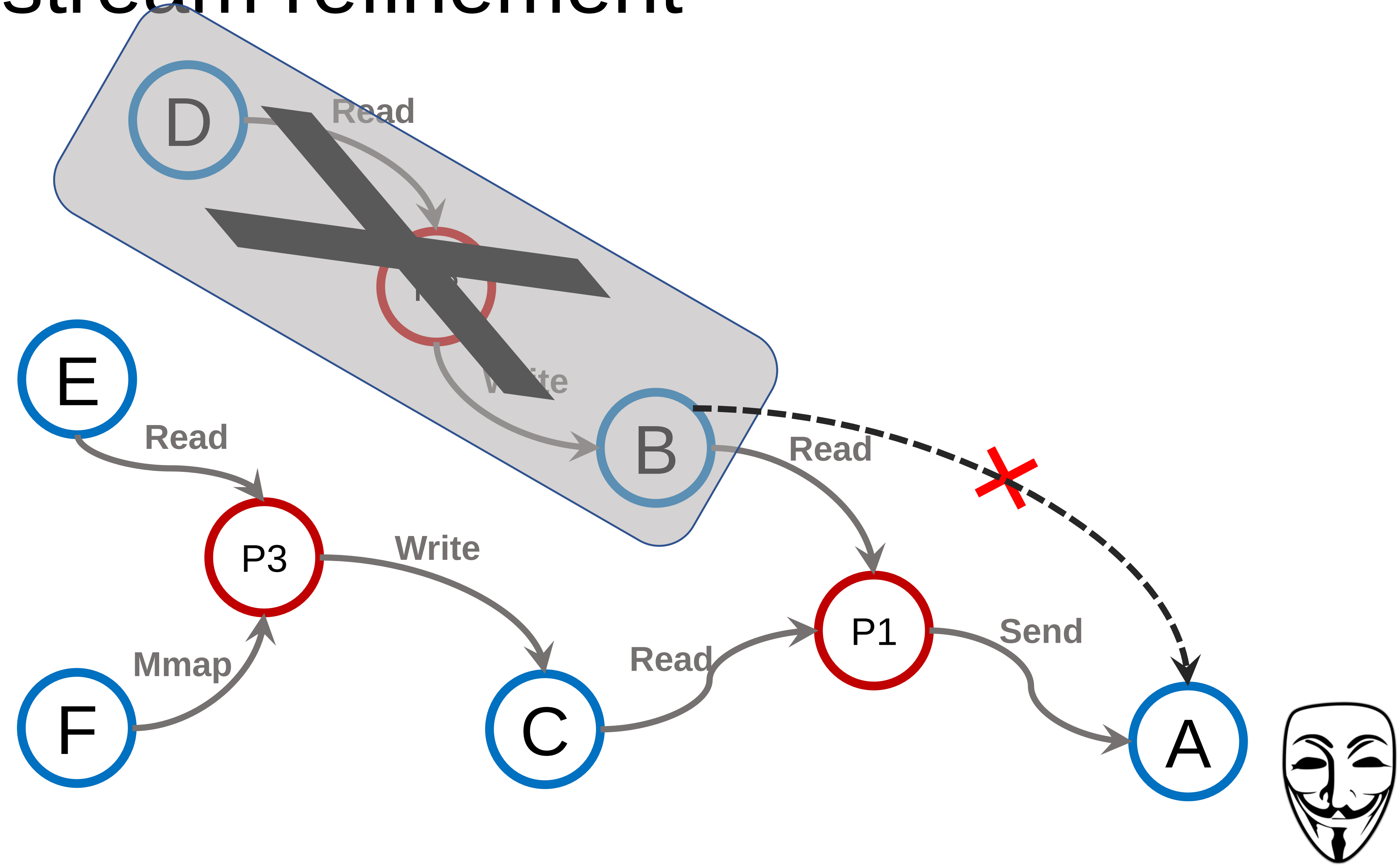
Upstream refinement



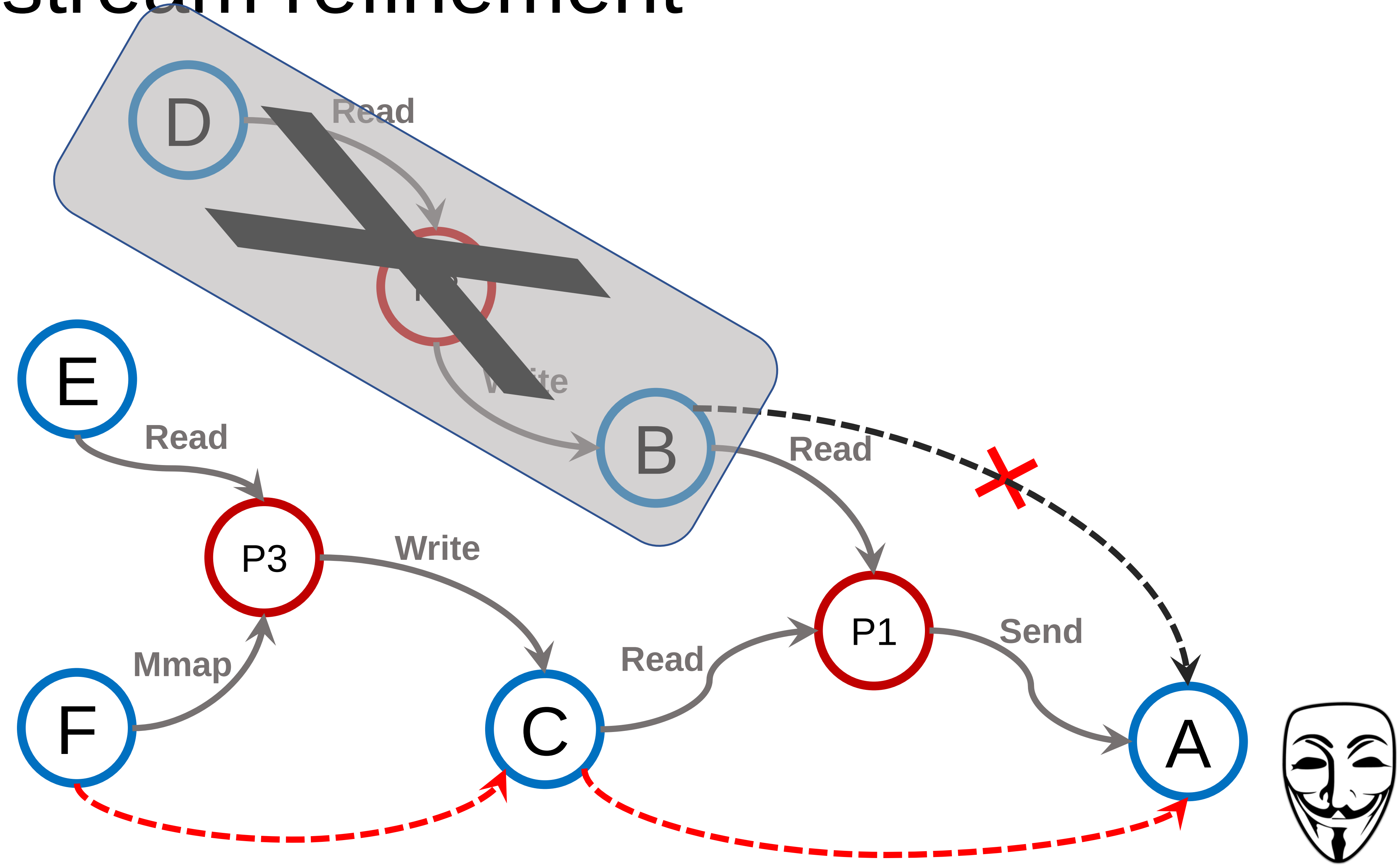
Upstream refinement



Upstream refinement



Upstream refinement



Implementation summary

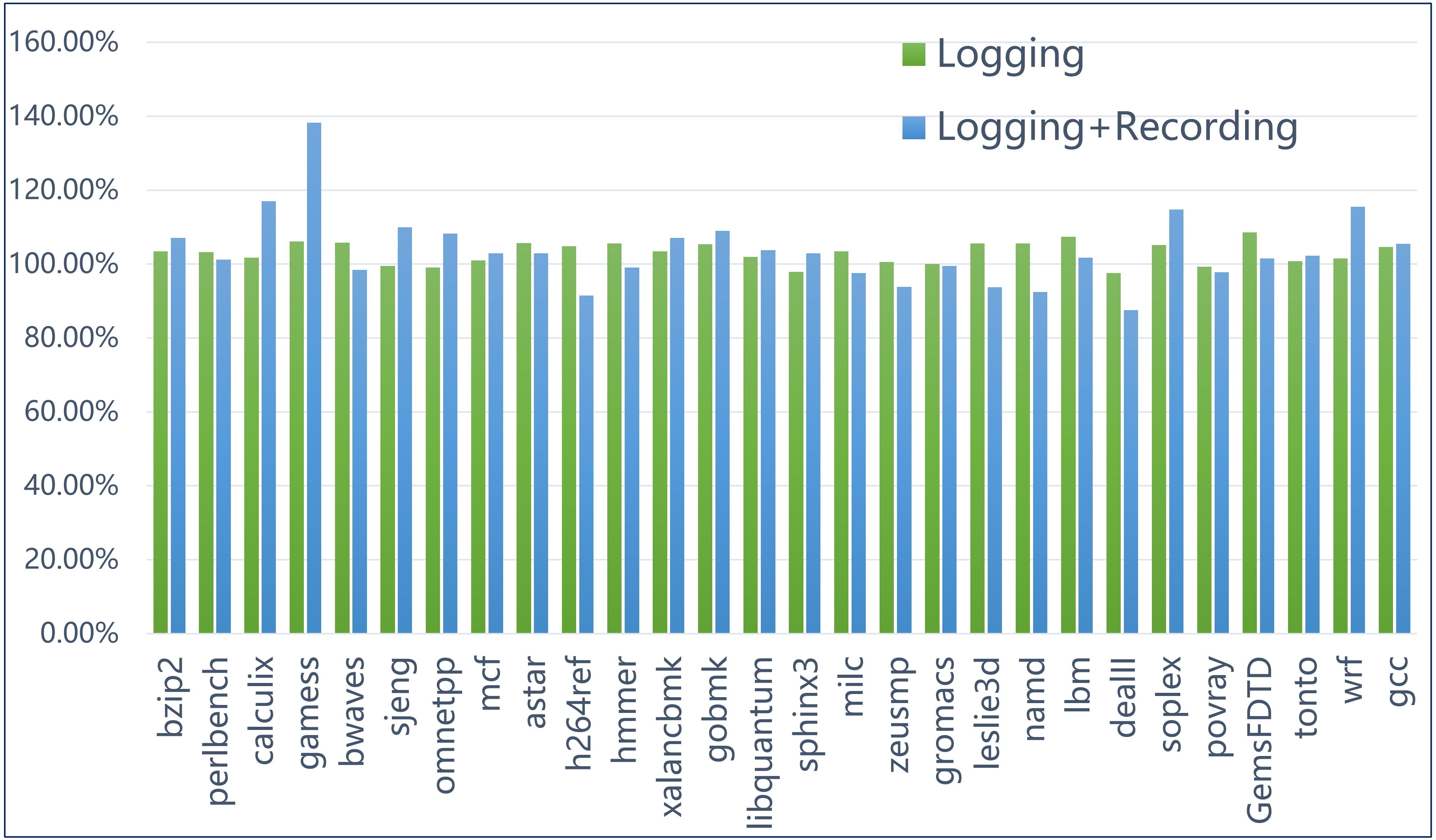
- RAIN is built on top of:
  - Arnold, the record replay framework
  - Dtracker (Libdft) and Dytan, the taint engines

Host	Module	LoC
Target host	Kernel module	2,200 C (Diff)
	Trace logistics	1,100 C
Analysis host	Provenance graph	6,800 C++
	Trigger/Pruning	1,100 Python
	Selective refinement	900 Python
	DIFT Pin tools	3,500 C/C++ (Diff)

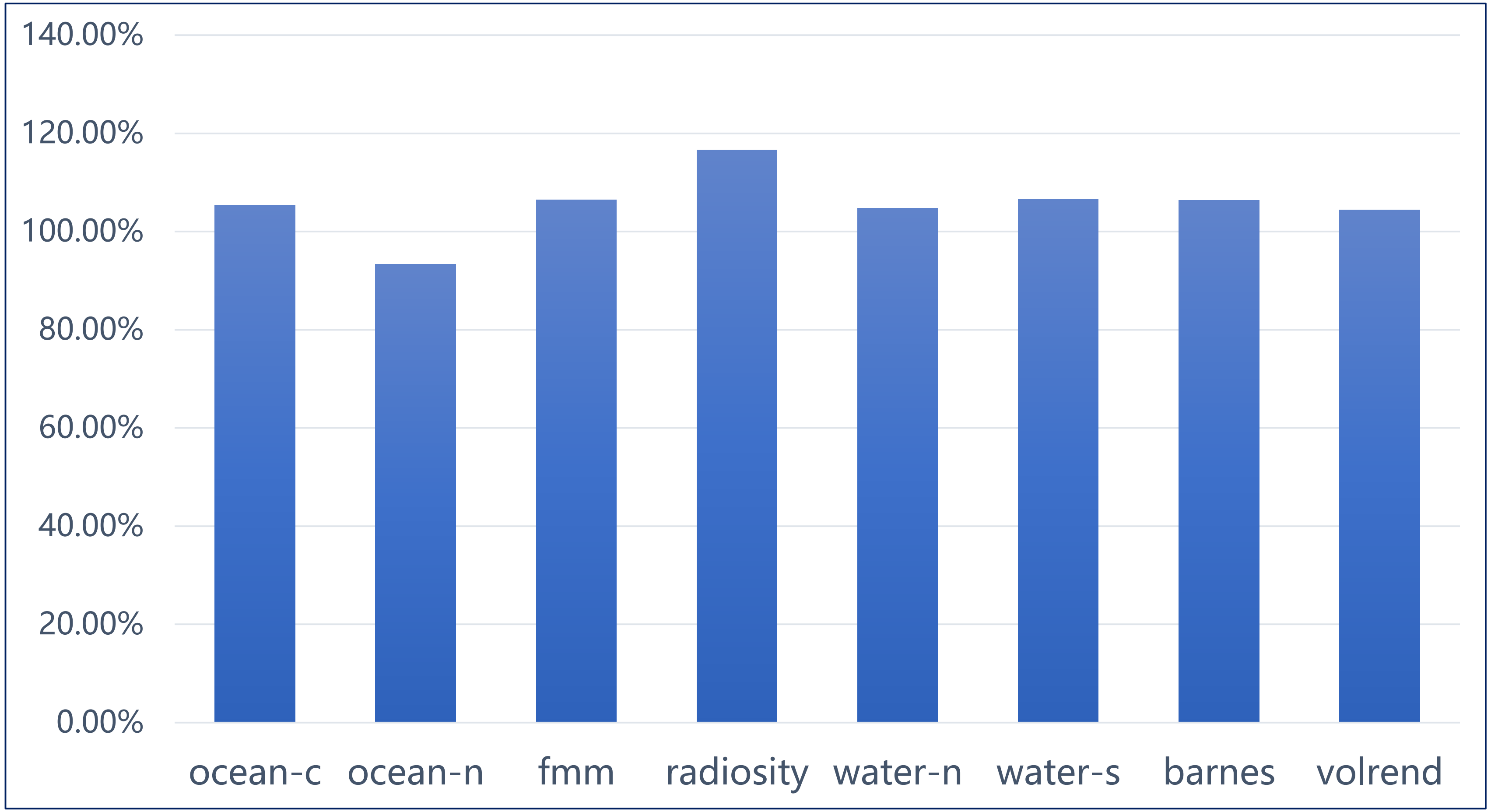
Evaluations

- Runtime performance
- Accuracy
- Analysis cost
- Storage footprint

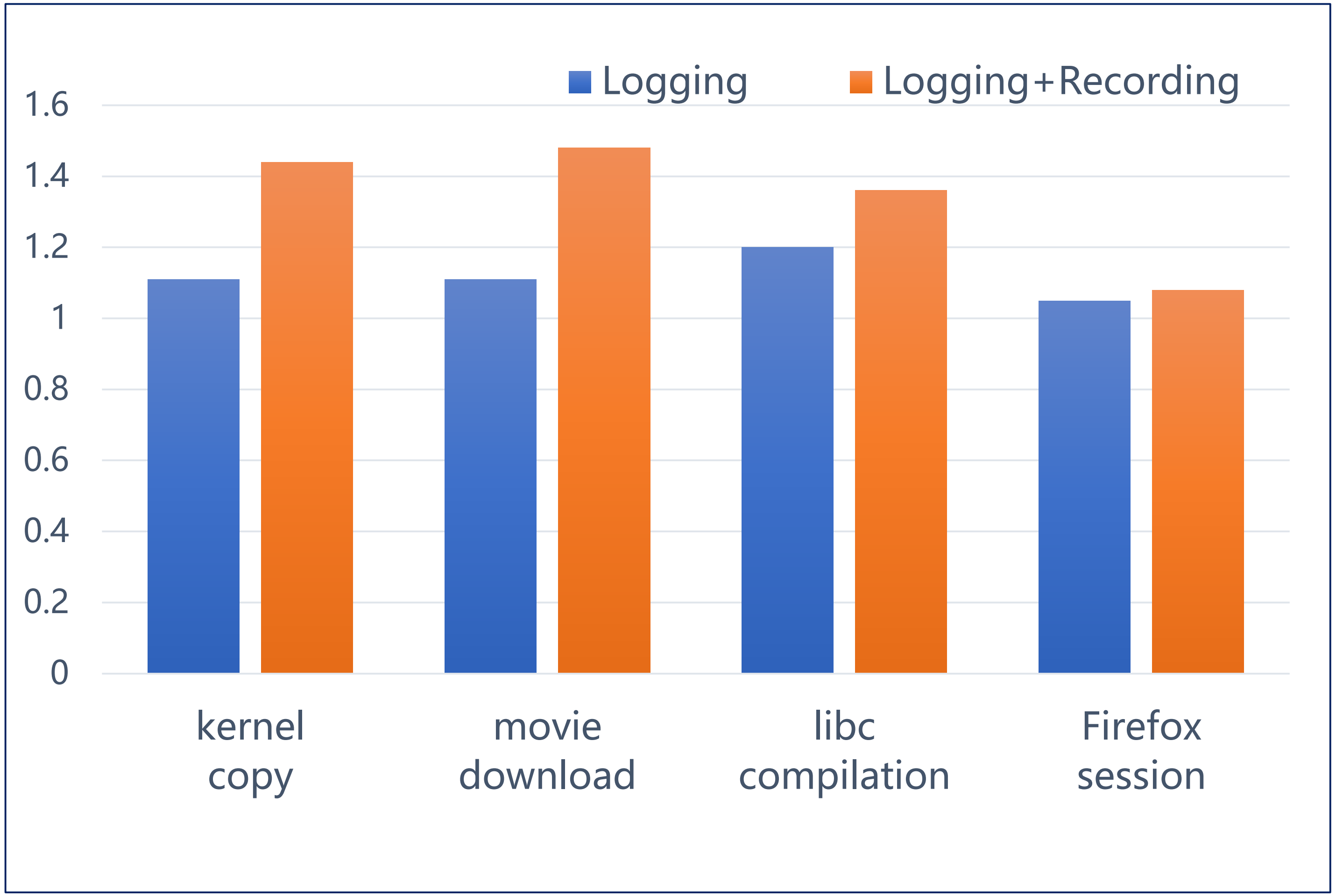
Runtime overhead: 3.22% SPEC CPU2006



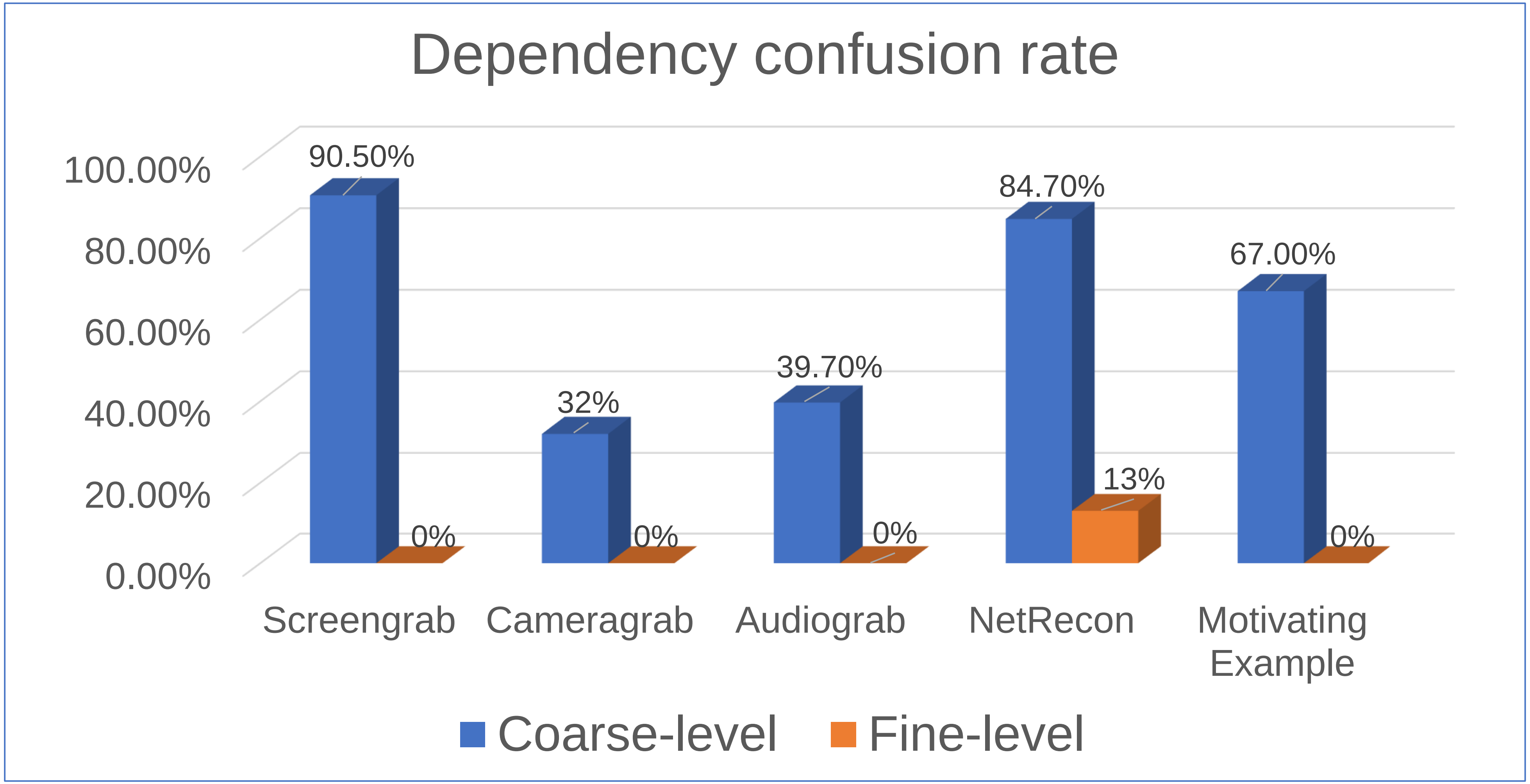
Multi-thread runtime overhead: 5.35% SPLASH-3



IO intensive application: less than 50%

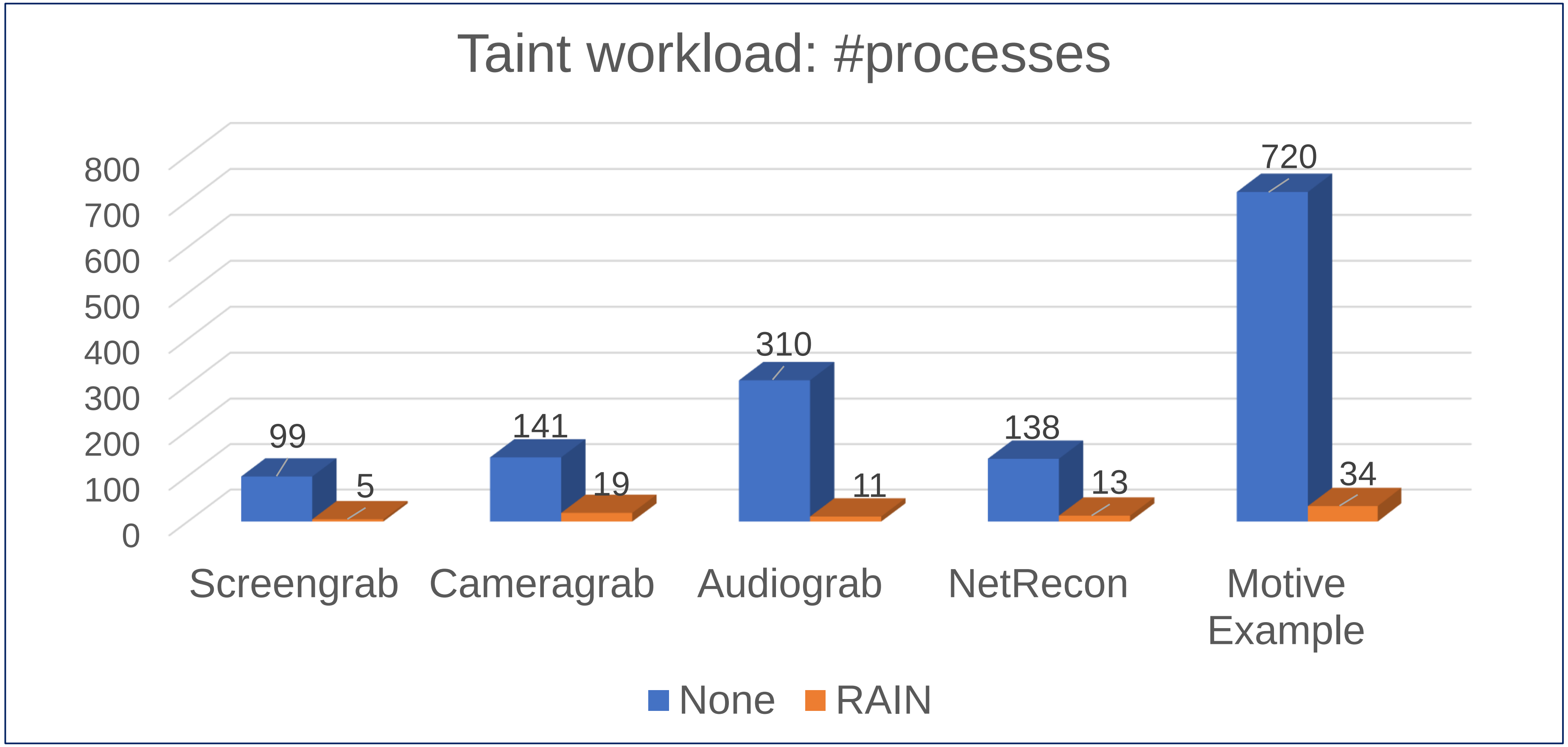


High analysis accuracy

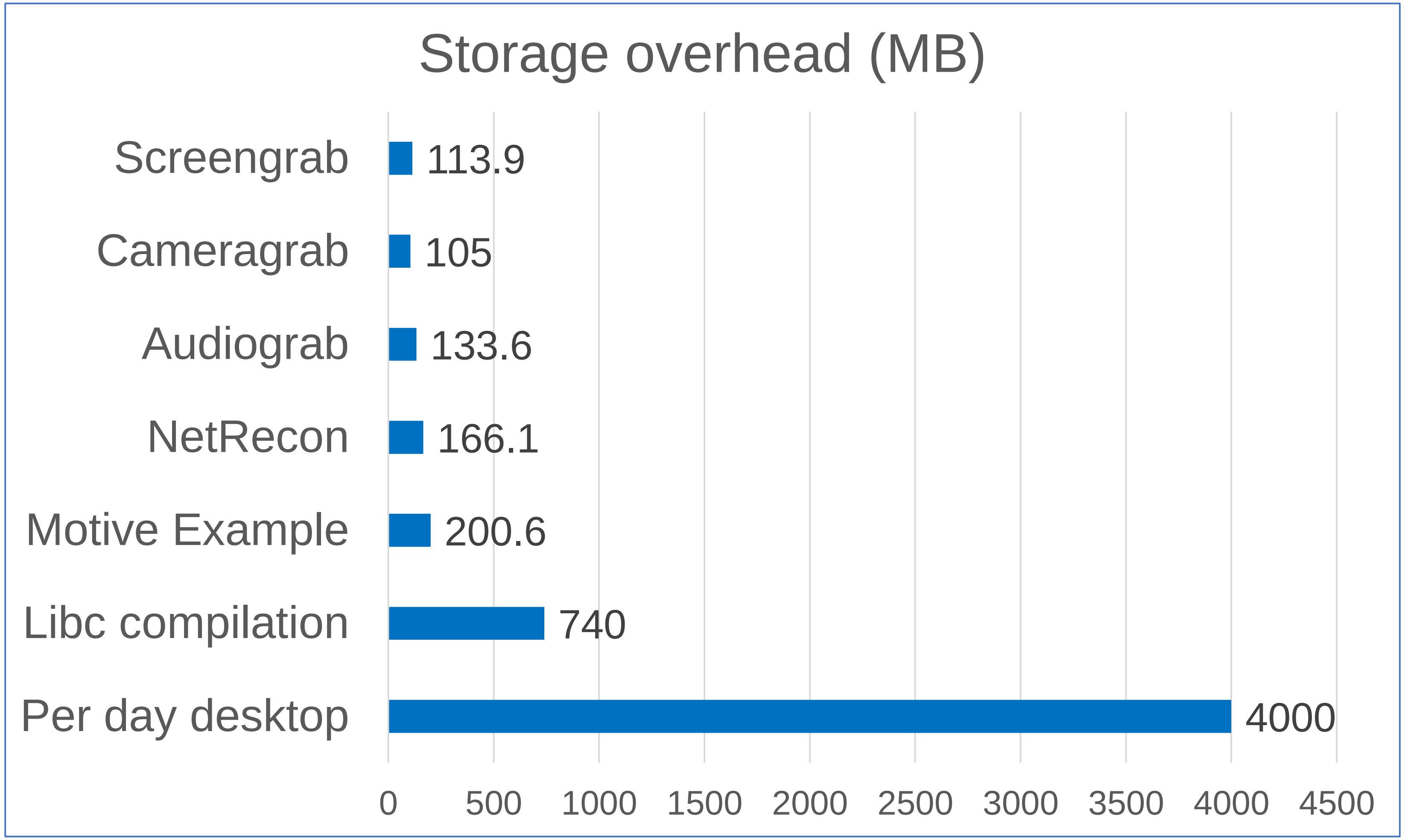


Scenarios from red team exercise of DARPA Transparent Computing program

Pruning effectiveness: ~94.2% reduction



Storage cost: ~4GB per day (1.5TB per year)



Discussion

- Limitations
  - RAIN trusts the OS that needs kernel integrity protection.
  - Over-tainting issue
- Direction
  - Hypervisor-based RAIN
  - Further reduce storage overhead

Conclusion

- RAIN adopts a multi-level provenance system to facilitate fine-grained analysis that enables accurate attack investigation.
- RAIN has low runtime overhead, as well as significantly improved analysis cost.