# In the news…

## Data Leaks

**facebook**

**Mar '19**

*passwords stored in readable format*

600M 👤

**Google+**

**Apr '19**

*shutdown after data leaks*

0.5M 👤

**AADHAAR**

**Mar '18**

*exposed user data*

1B 👤

## Data Breaches

**Marriott**

**Nov '18** 500M 👤

**EQUIFAX**

**Sep '17** 143M 👤

**IBM** **Cost of a Data Breach Study**

www.ibm.com/security/data-breach

**1.8B U$** **~500**

**companies** **2018**

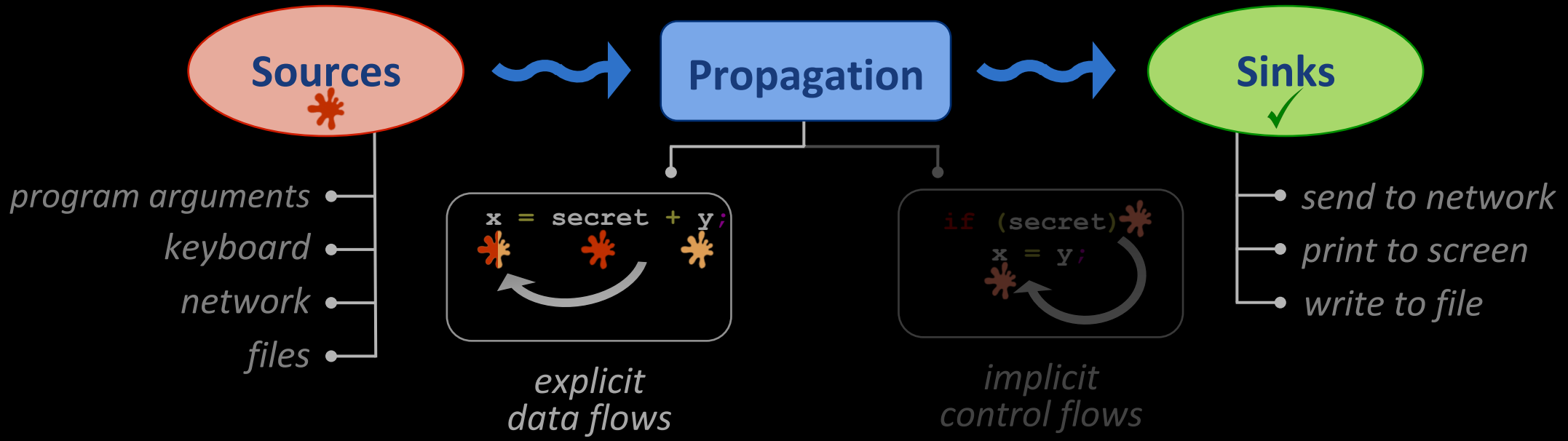
# Dynamic Taint Tacking

tracks information flow

# Dynamic Taint Tacking

can prevent information leak

- Associate taints with sensitive data
- Propagate taints to derived values
- ✓ Check tainted values don't reach untrusted channels



# Dynamic Taint Tacking

enables powerful analyses

*security*

overwrite attacks

command injection attacks

XSS attacks

*privacy*

information leakage


*software engineering*

testing and debugging

semantic analysis

# Problem

Dynamic Taint Tracking is **expensive!**

# Dynamic Taint Tacking is expensive!

```
main (…) {
  x = c + 3;
  t(x)=t(c);
  y = secret;
  t(y)=t(secret);
  if (p < 0) {
    z = c * y;
    t(z)=t(c)|t(y);
  }
  out = z;
  t(out)=t(z);
  assert(!t(out));
  printf(out);
}
```

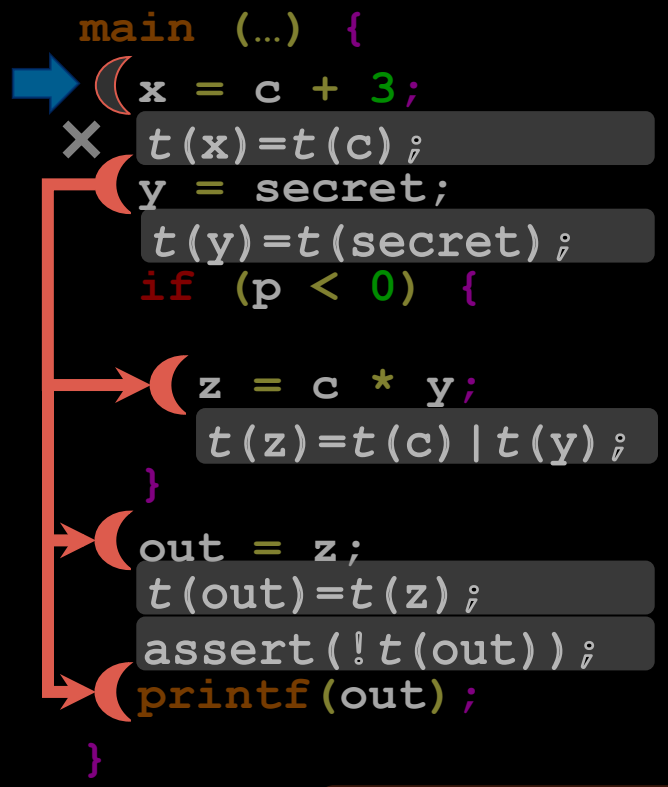
*track*

*check*

| | |
|---|---|
| ✱ | secret |
| ✱ | c |
| ✱ | p |
| ✱ | x |
| ✱ | y |
| ✱ | z |
| ✱ | out |
| ⋮ | ⋮ |

slowdown [Newsome et al. ‘05]

# Static Analysis can help ?




*Static analyses—*
*dataflow taint analysis*
*+ pointer analysis*

✓ *sound*

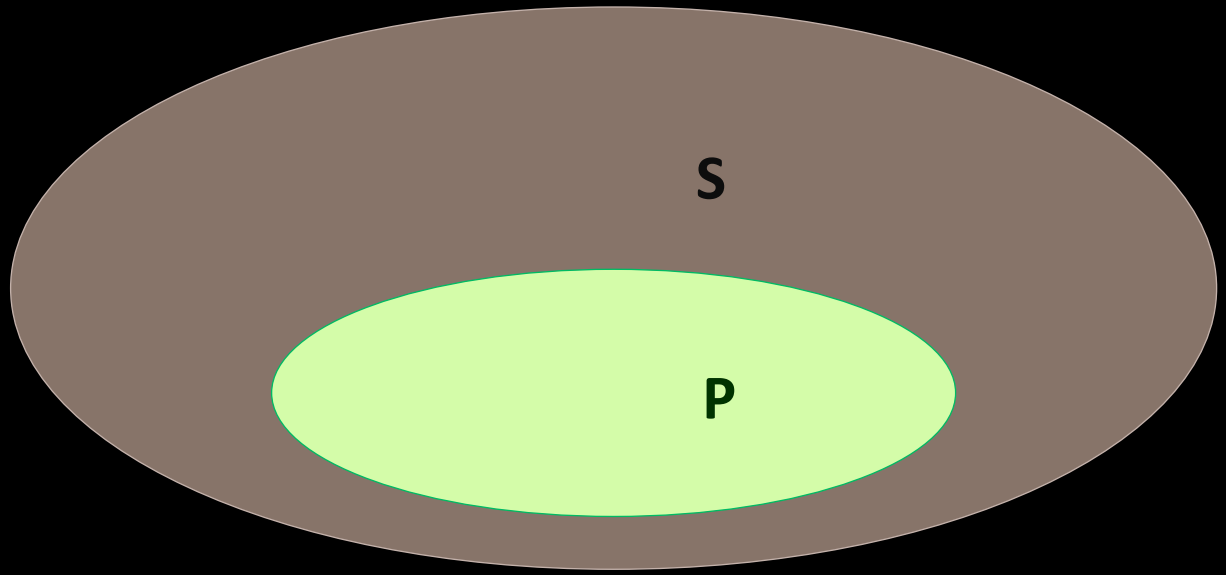
✗ *imprecise*

✗ *not scalable*

5× → 2.7×

∴ not effective enough…

# Static Analysis Limitation




**P** : **Possible** program states

**S** : **Sound Static** analysis' state space

✓ *sound*

✗ *imprecise*

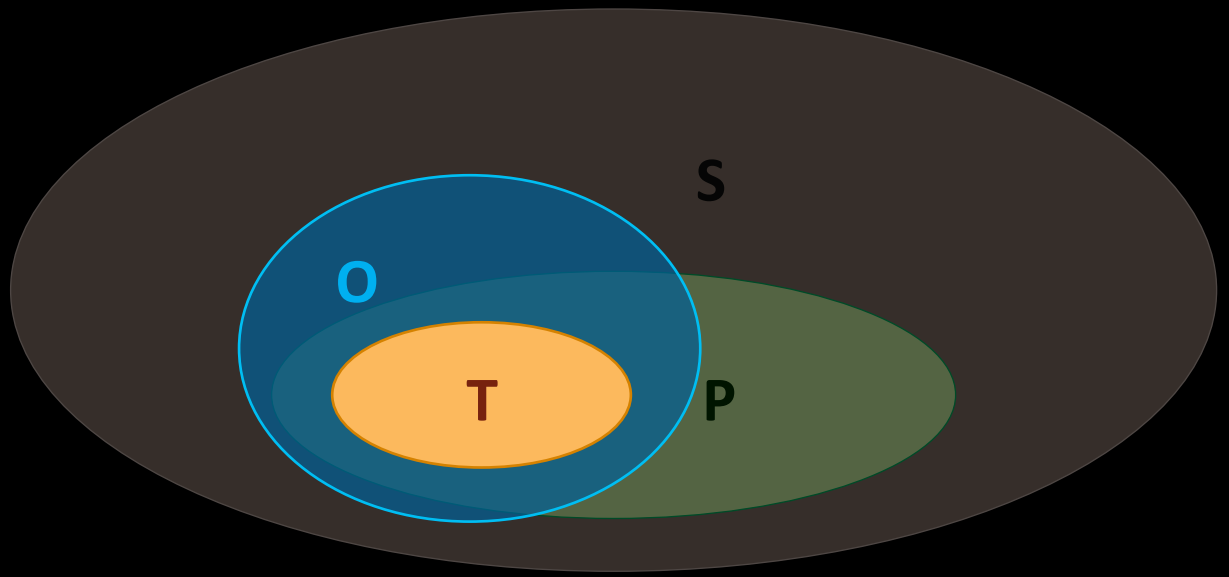
✗ *not scalable*

# Solution

## Optimistic Hybrid Analysis

Taint Tracking | is slow ! | Optimistic Hybrid Analysis | with *Safe Elisions* | improves !

# Predicated Static Analysis




**P** : **Possible** program states

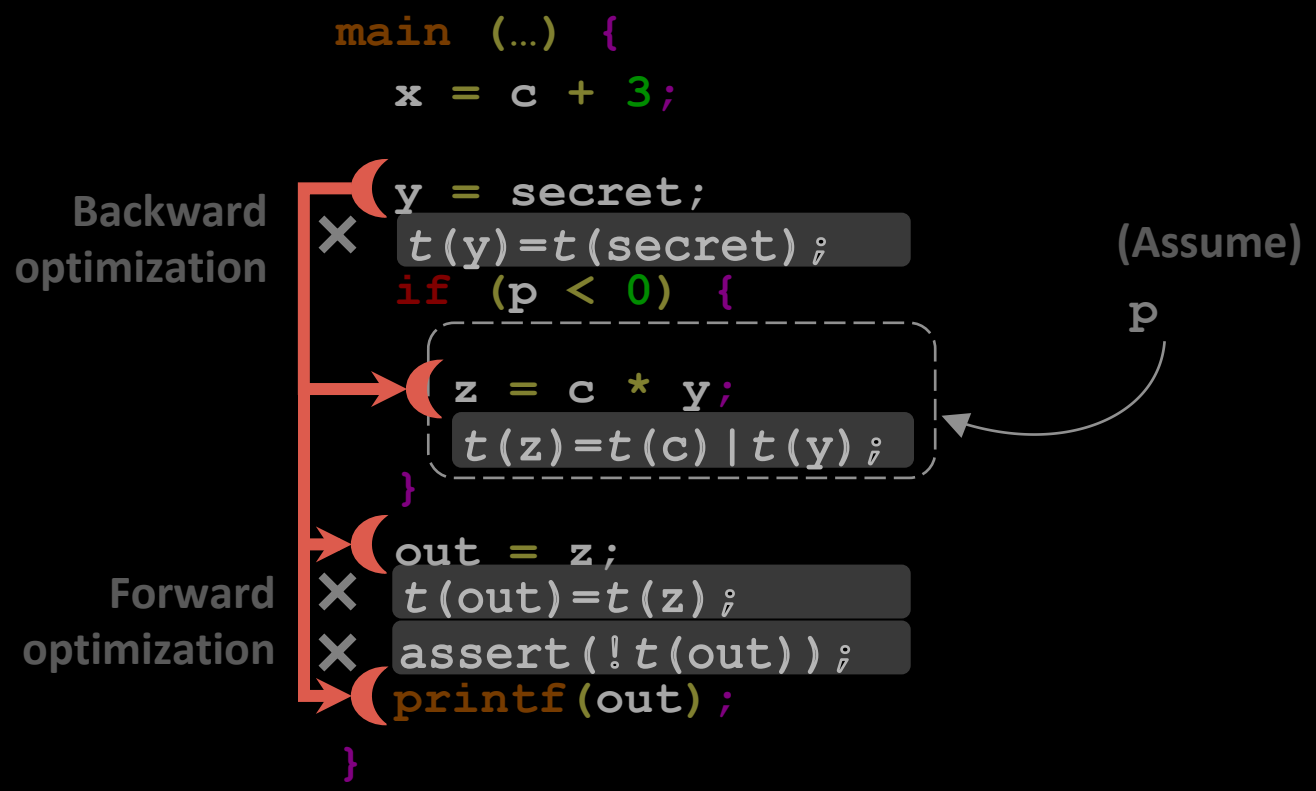
**S** : **Sound Static** analysis' state space

**T** : **Tested** program states

**O** : **Predicated Static** analysis' state space

✗ *unsound* ✓ *precise* ✓ *scalable*

# Predicated Static Analysis




*Optimistic analyses—*
*dataflow taint analysis*
*+ pointer analysis*

***+ invariant assumption***

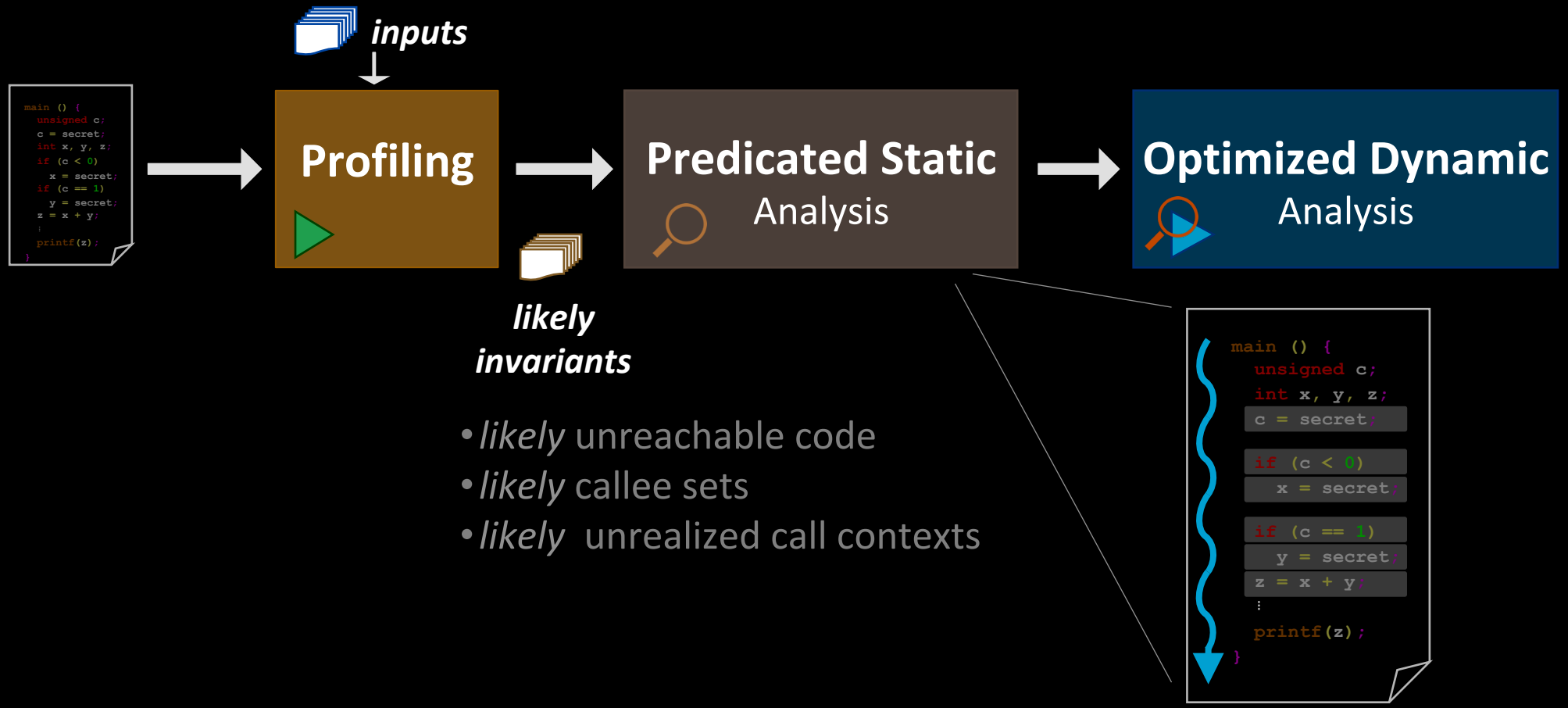
✓ ***precise***

✓ ***optimized for common case***

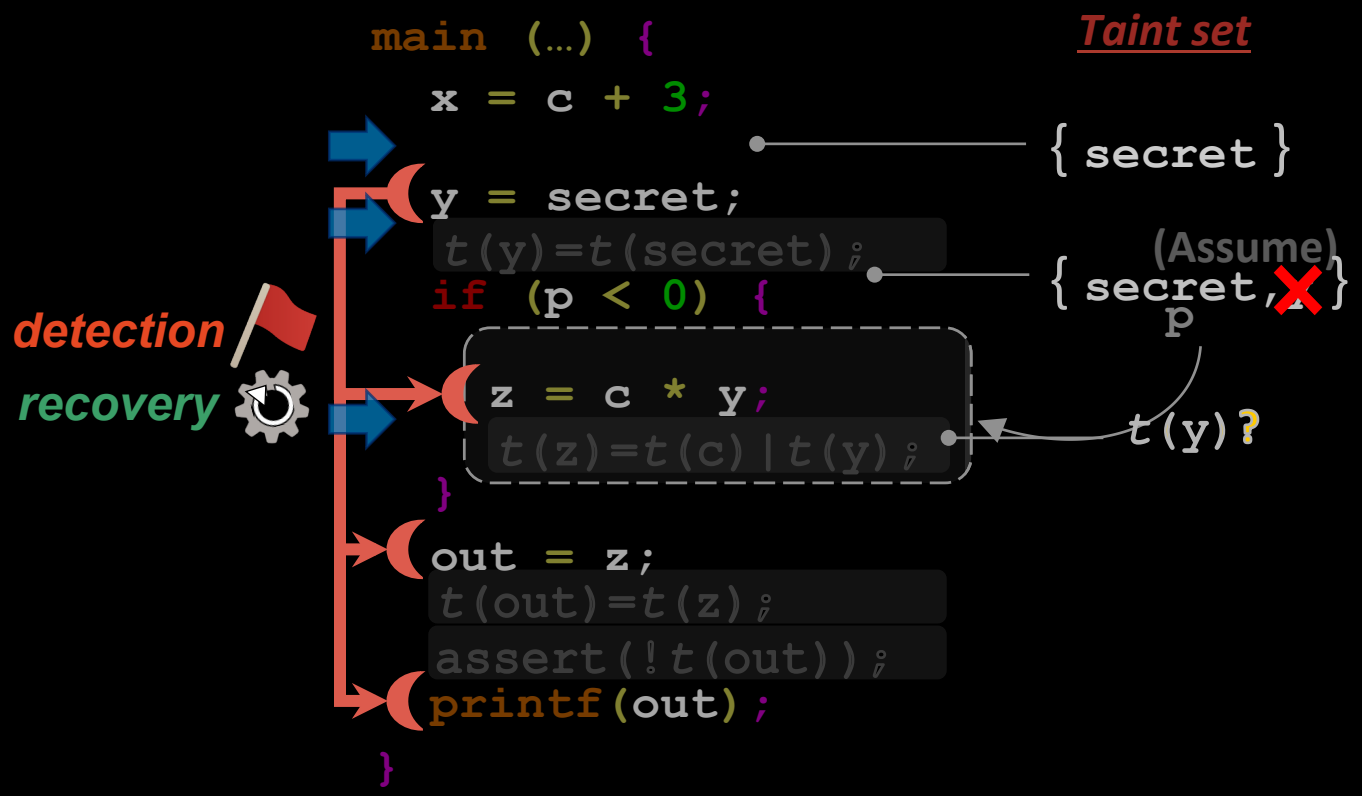
✓ ***scalable***

# Optimistic Hybrid Analysis

[Devecsery et al. ‘18]



# Optimistic Assumptions ⇒ missed state ?




*likely* Unreachable Code

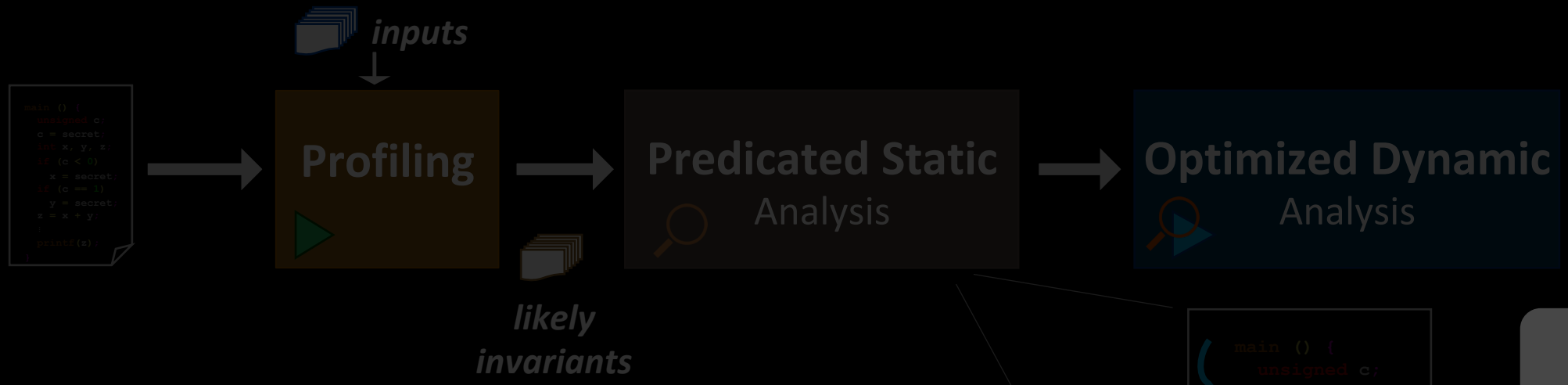
*likely* Callee Sets

*likely* Unrealized Call Contexts

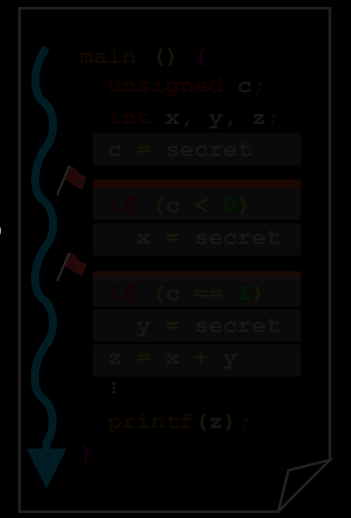
✓ *sound*

***invariant violation detection*** + ***analysis recovery***

# Recovery in OHA is a serious issue




Conservative approach: **Rollback** to the **beginning**
and **re-execute** with **unoptimized** analysis



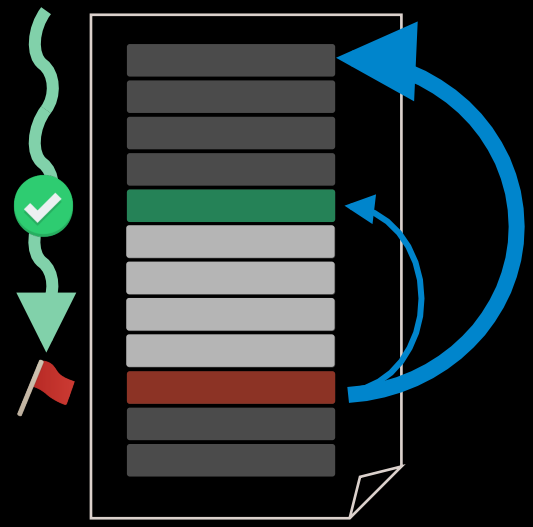
+

**Recovery Mechanism**

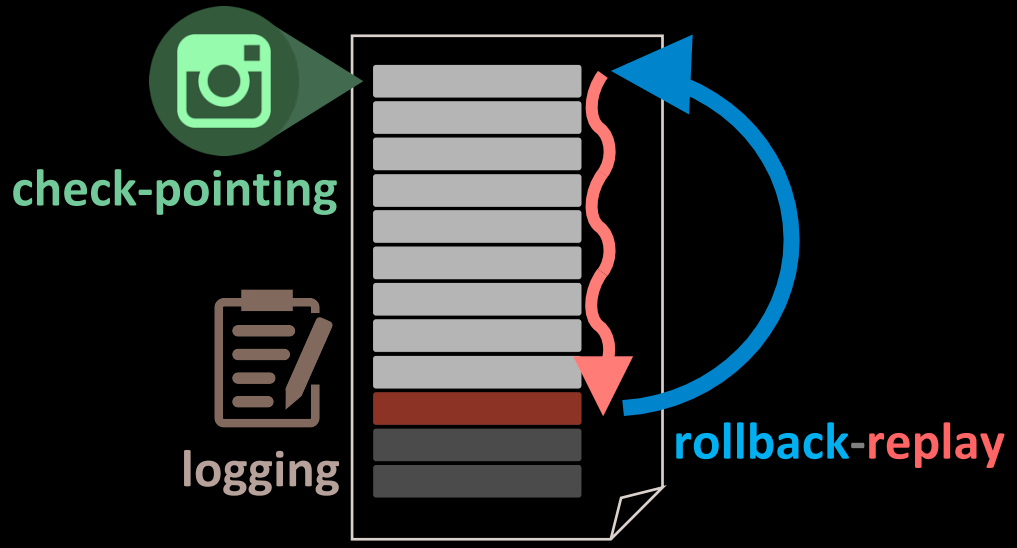
✓ **Sufficient for offline analysis**

✗ **Prohibitive for live executions**

# Rollback Recovery is Problematic !



Unbounded Rollbacks




Overheads !

# Recap

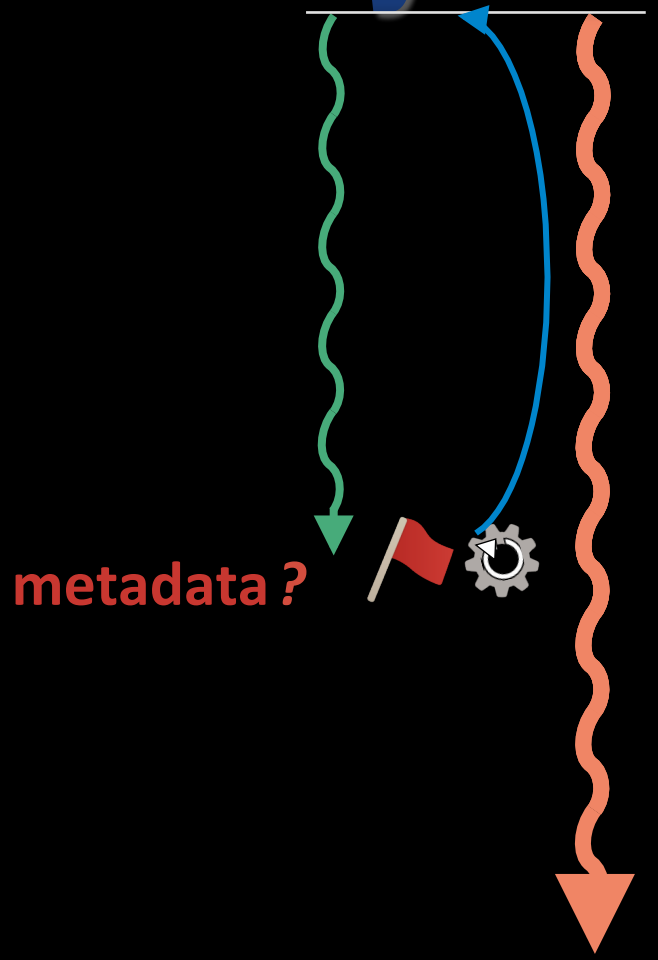
- **Full Dynamic Analysis** is prohibitively expensive.
- **Conservative Hybrid Analysis** is imprecise and inefficient.
- **Optimistic Hybrid Analysis** can improve.
  - But **Rollback Recovery** is challenging.

# Rollback-free

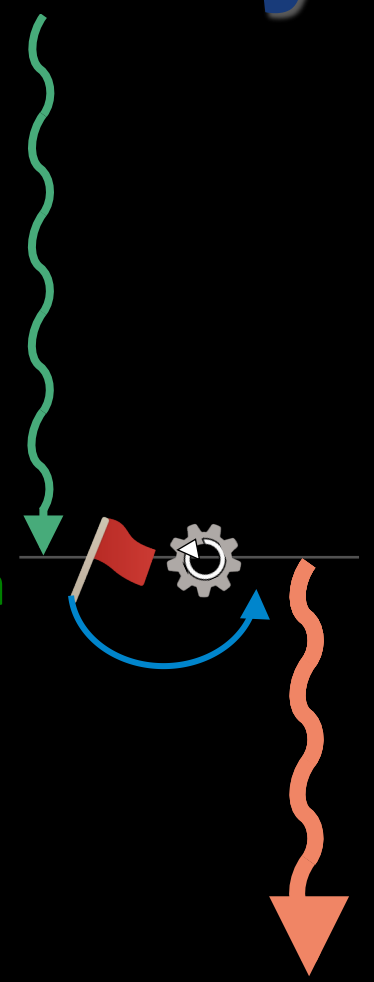
## Optimistic Hybrid Analysis

Taint Tracking → is slow ! → Optimistic Hybrid Analysis → with *Safe Elisions* → improves !

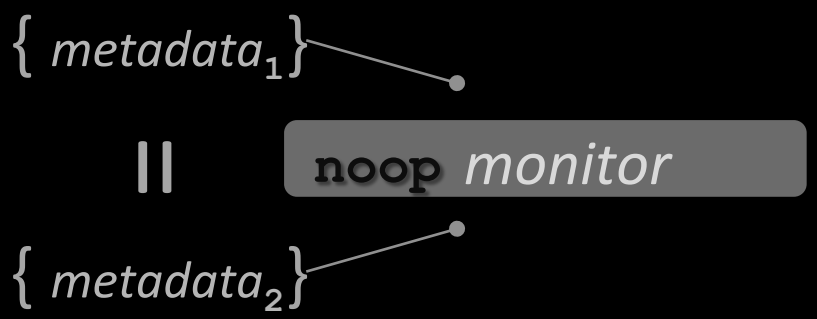
# Rollback Recovery

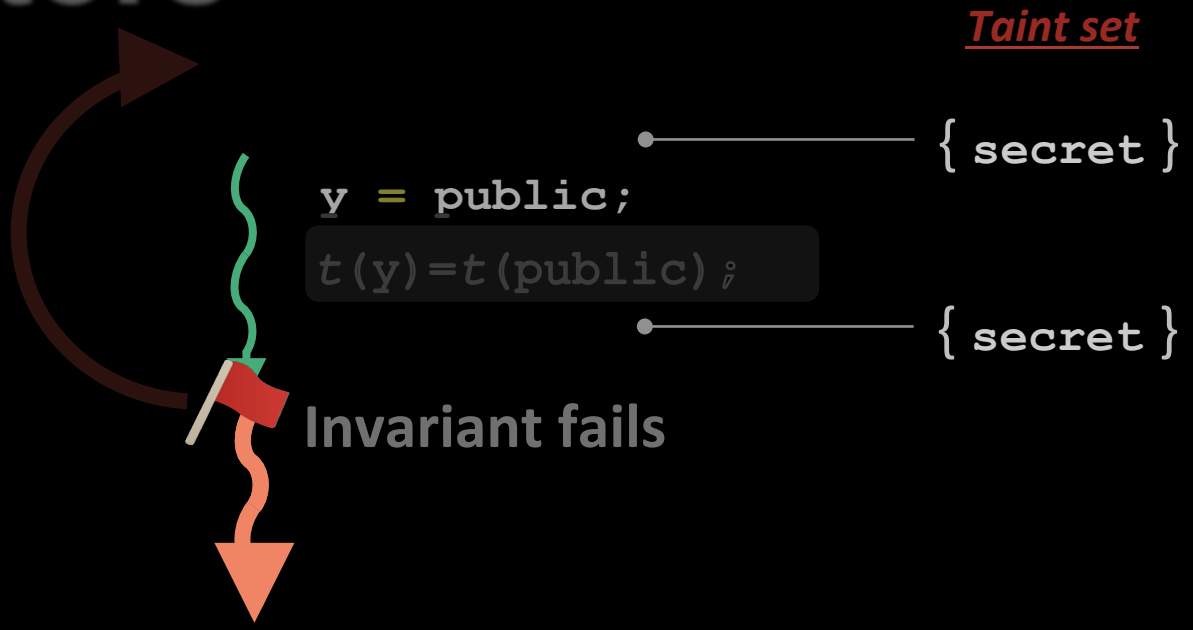



# Forwar Recover d y



# *Safe Elisions* of noop monitors

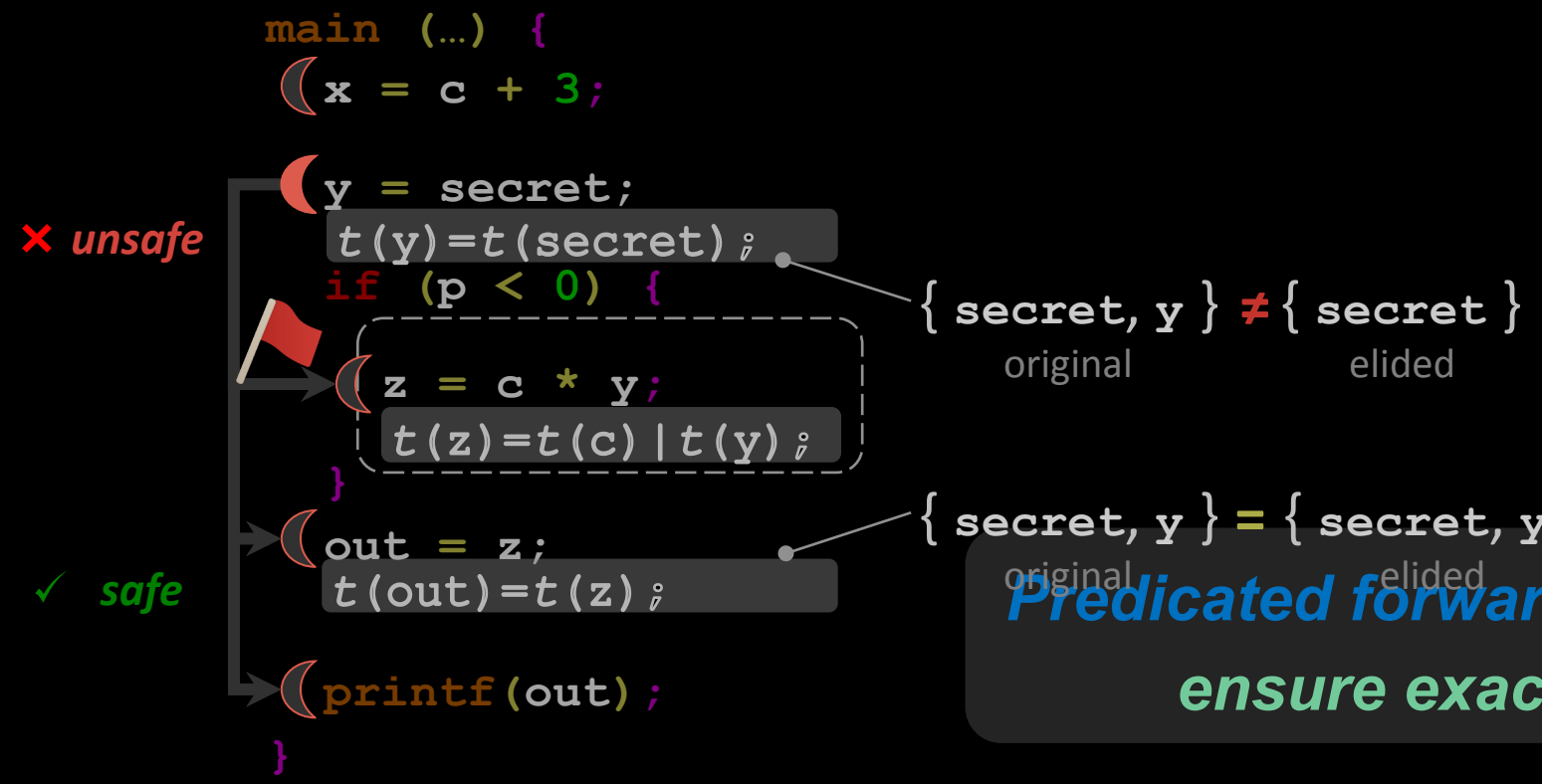



***ensures metadata equivalence !***

***exact semantics***

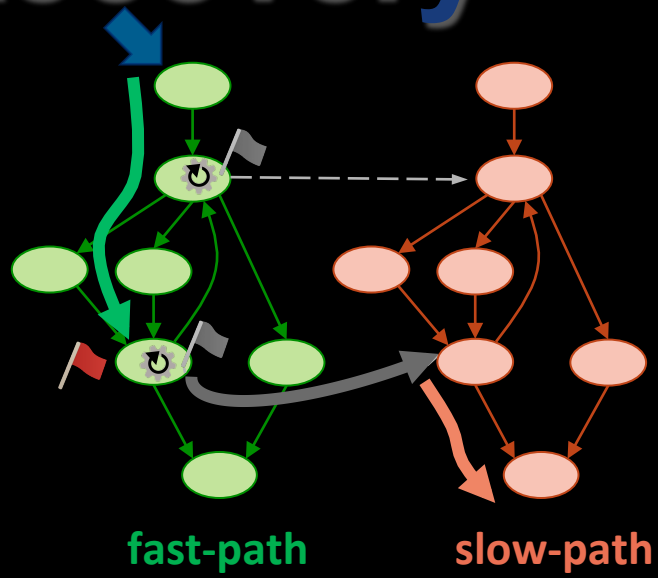
# *Safe Elisions* of noop monitors

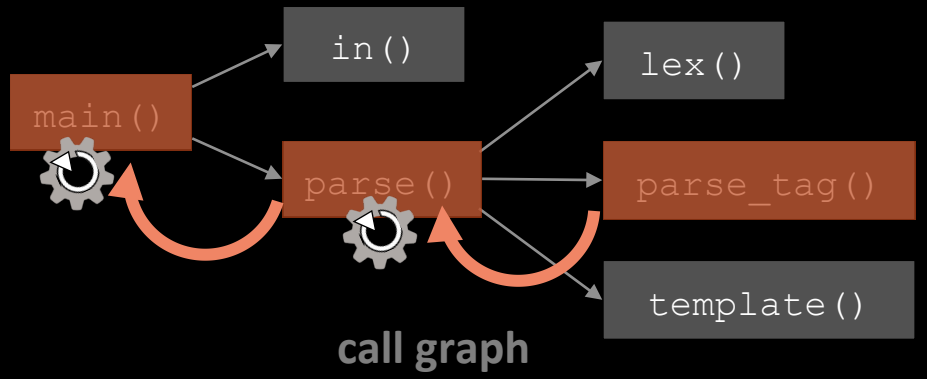



***Predicated forward optimizations are safe***

***ensure exact metadata state !***

# Forward Recovery :




call graph

## Switching to conservative analysis

- Separate control flow domains
  **fast-path** and **slow-path**
- Switch on invariant failure
- Switch on call return from slow-path

# Evaluation

# IODINE Implementation

- **LLVM 3.9** compiler infrastructure
- **C** programs

## Conservative Hybrid

*Conservative Static* :

- Andersen's pointer analysis (context insensitive)
- data-flow taint analysis

*Dynamic* :

- taint tracking instrumentation- LLVM Data Flow Sanitizer

## Rollback-free Optimistic Hybrid

*Profiling* : 3 likely invariant types

*Predicated Static* :

- Andersen's pointer analysis (context sensitive)
- taint analysis: predicated forward + conservative backward

*Optimized Dynamic* :

- optimized taint tracking
- invariant checking + forward recovery

# IODINE accelerates DIFT applications

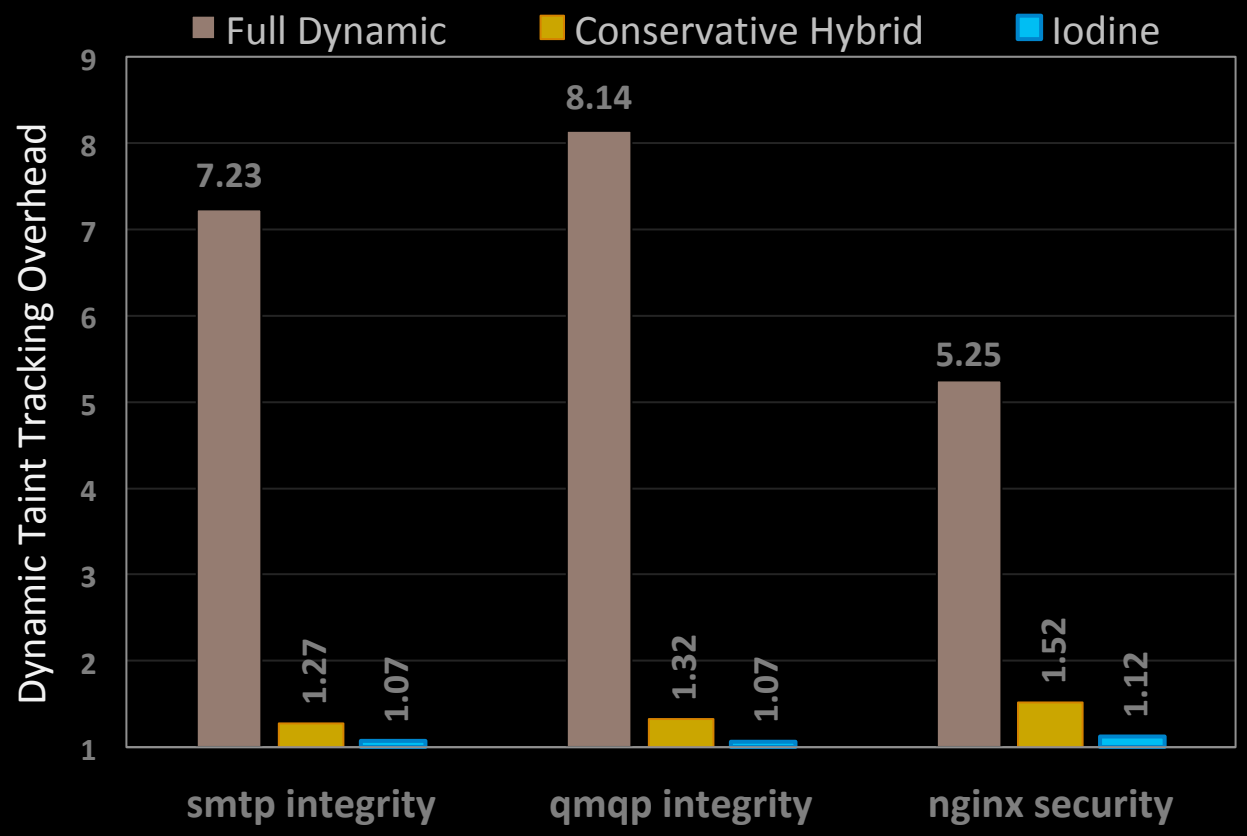
*Information flow security* policies —


Email integrity and privacy

NGINX
Web server

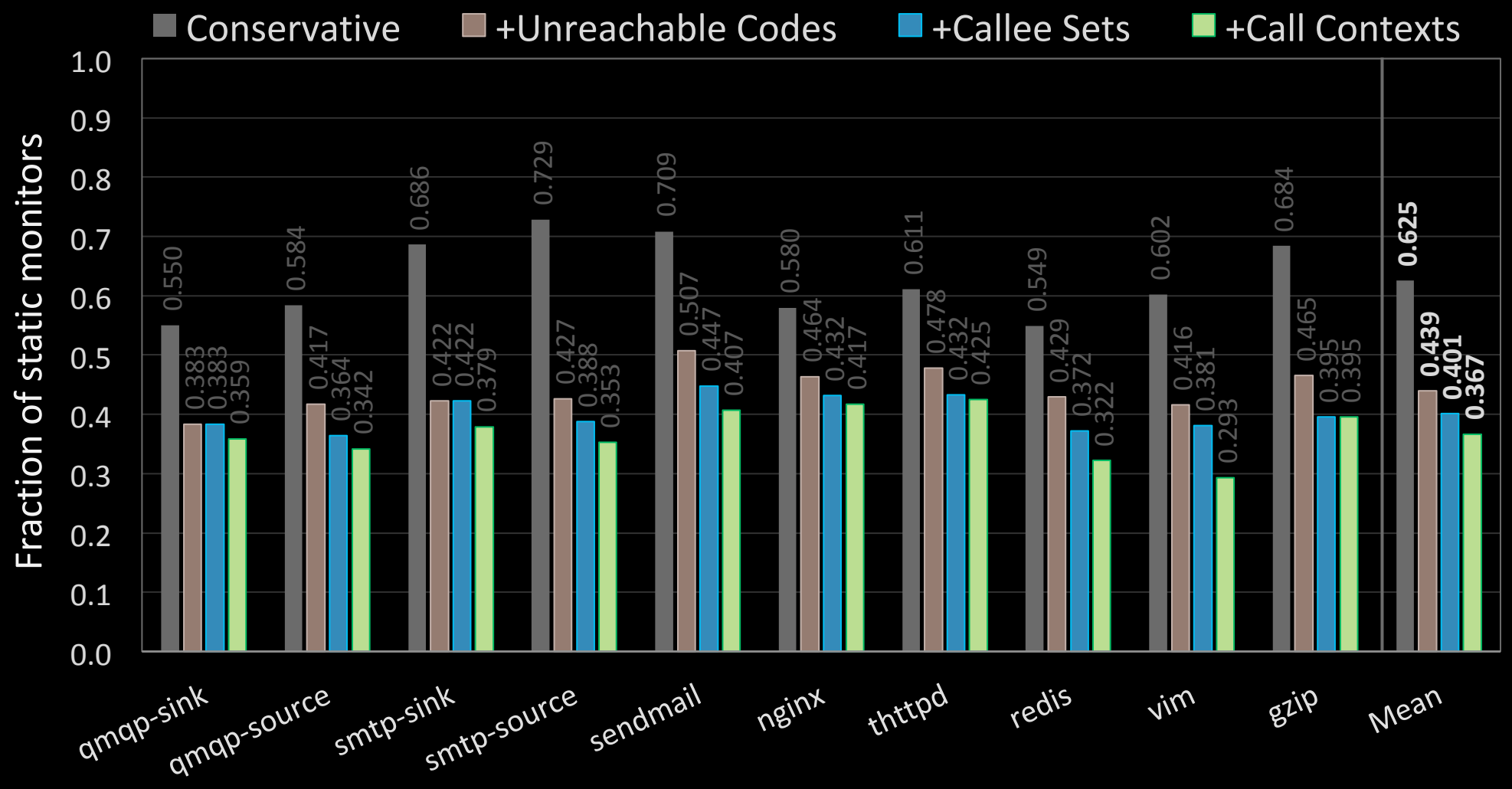
Overwrite attack detection

**4.4× faster than conservative**



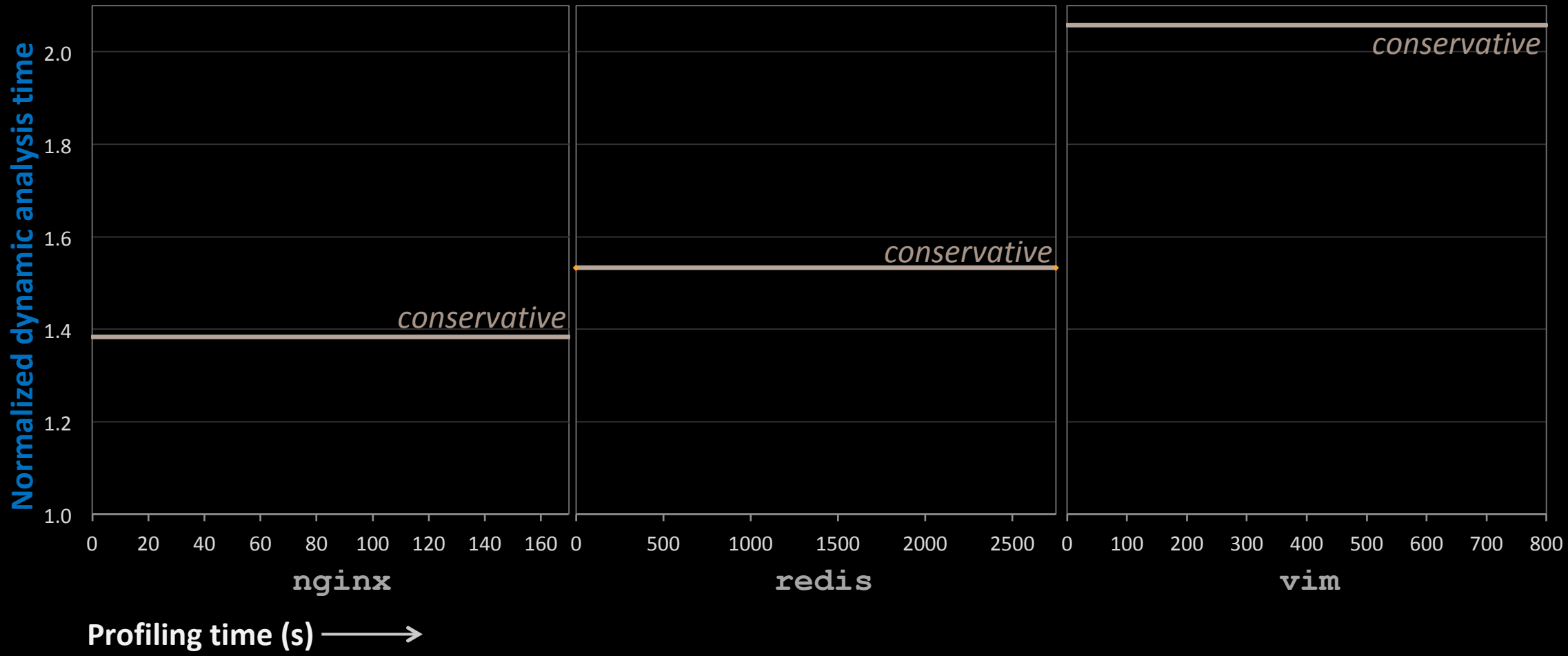

Taint Tracking | is slow ! | Optimistic Hybrid Analysis | with *Safe Elisions* | improves !

# Static Analysis Precision Improved by

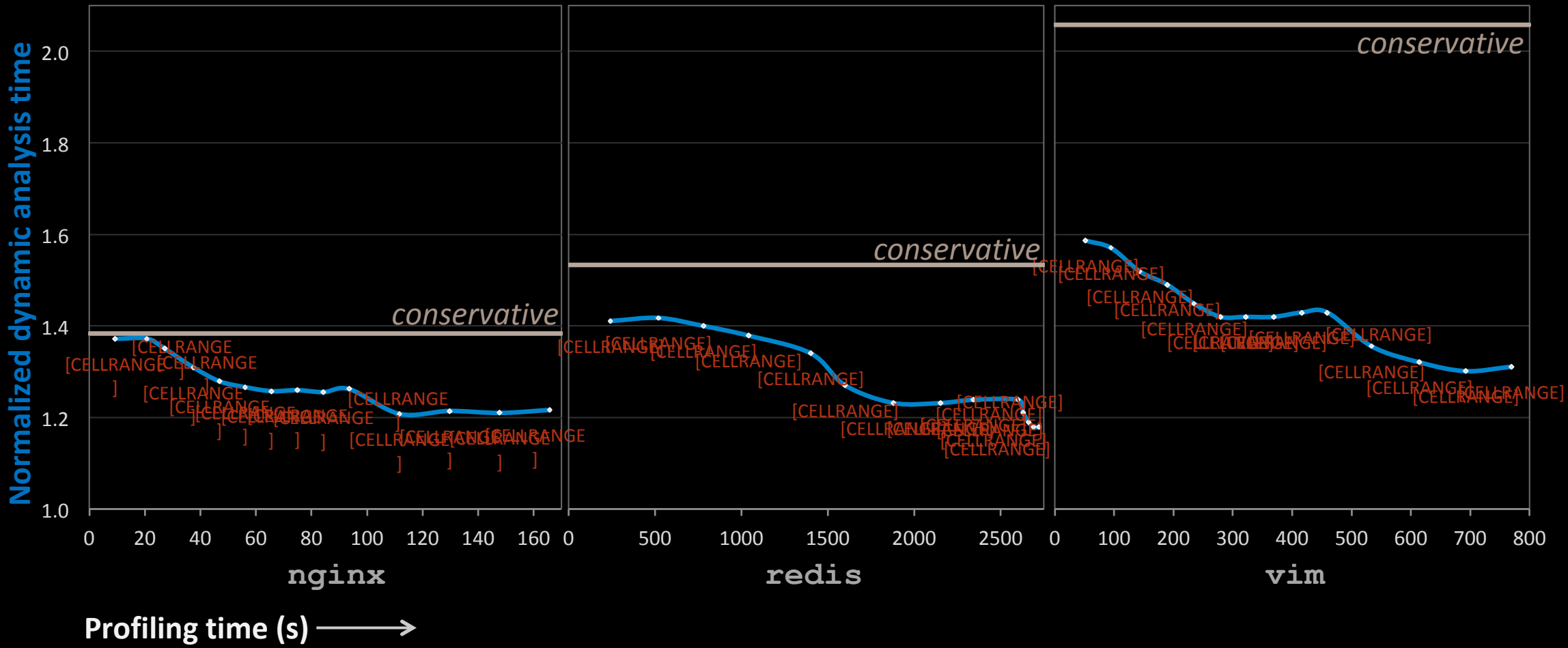



Taint Tracking | is slow ! | Optimistic Hybrid Analysis | with *Safe Elisions* | improves !

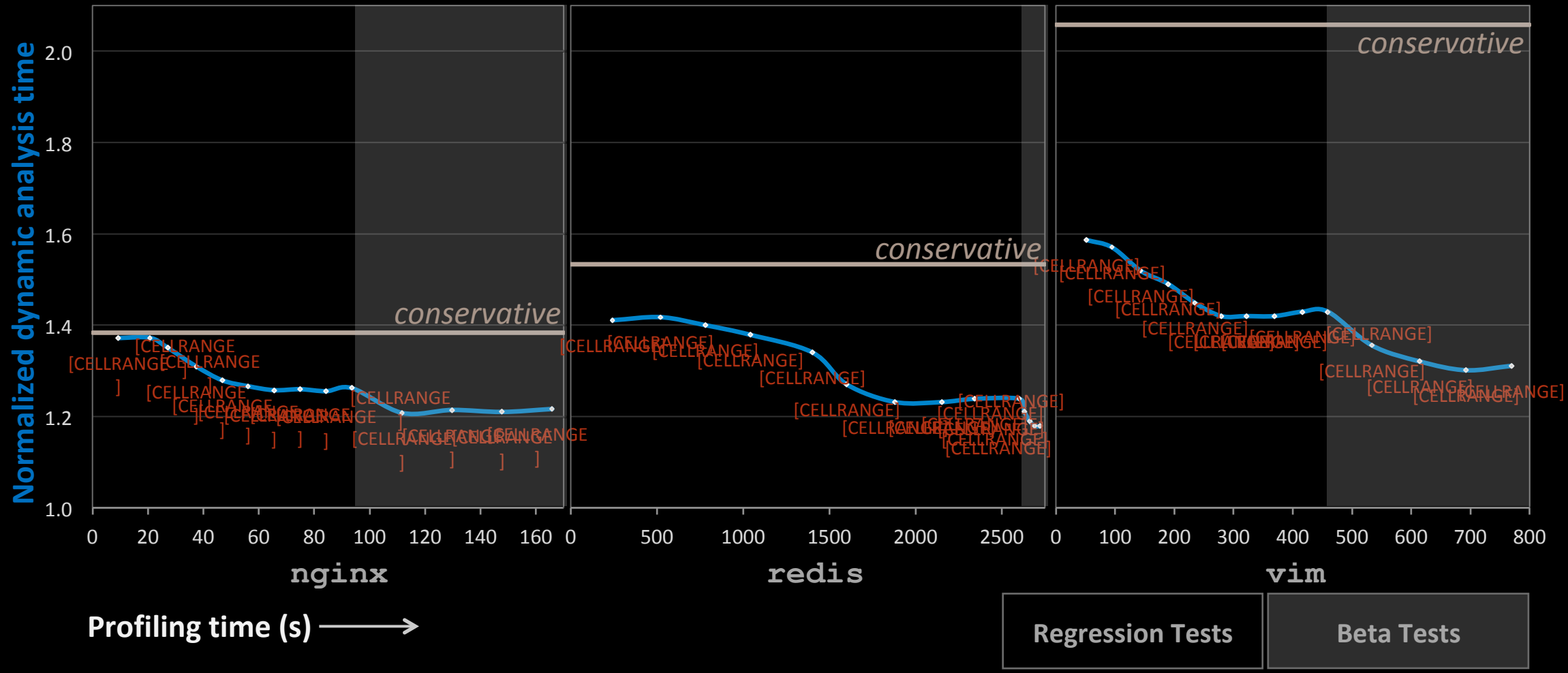
# Profiling Effort : regression test suites are adequate !



# Profiling Effort : regression test suites are adequate !



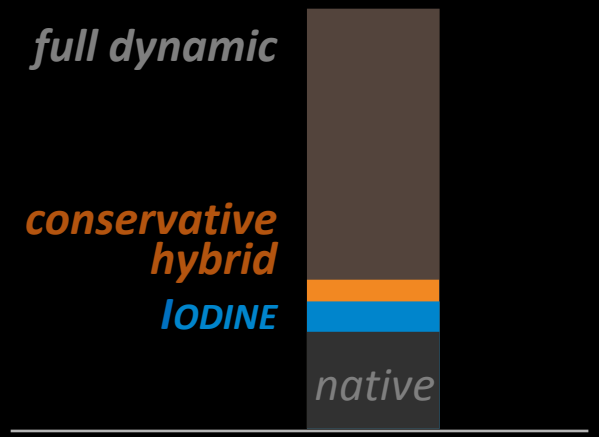
# Profiling Effort : regression test suites are adequate !



# Takeaways

Taint Tracking is slow ! Optimistic Hybrid Analysis with *Safe Elisions* improves !

# IODINE Summary




## Practical Dynamic Taint Tracking

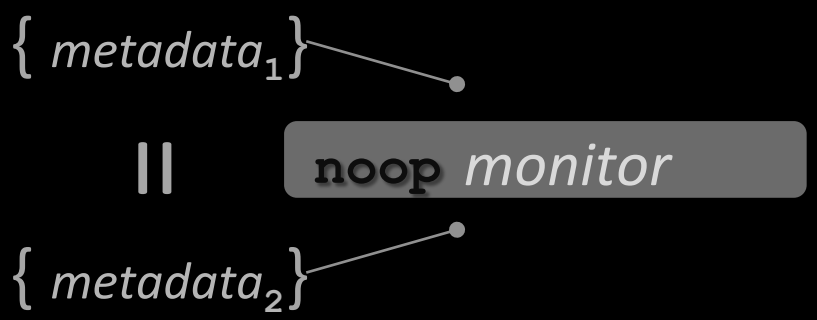
lower overhead than conservative hybrid analysis

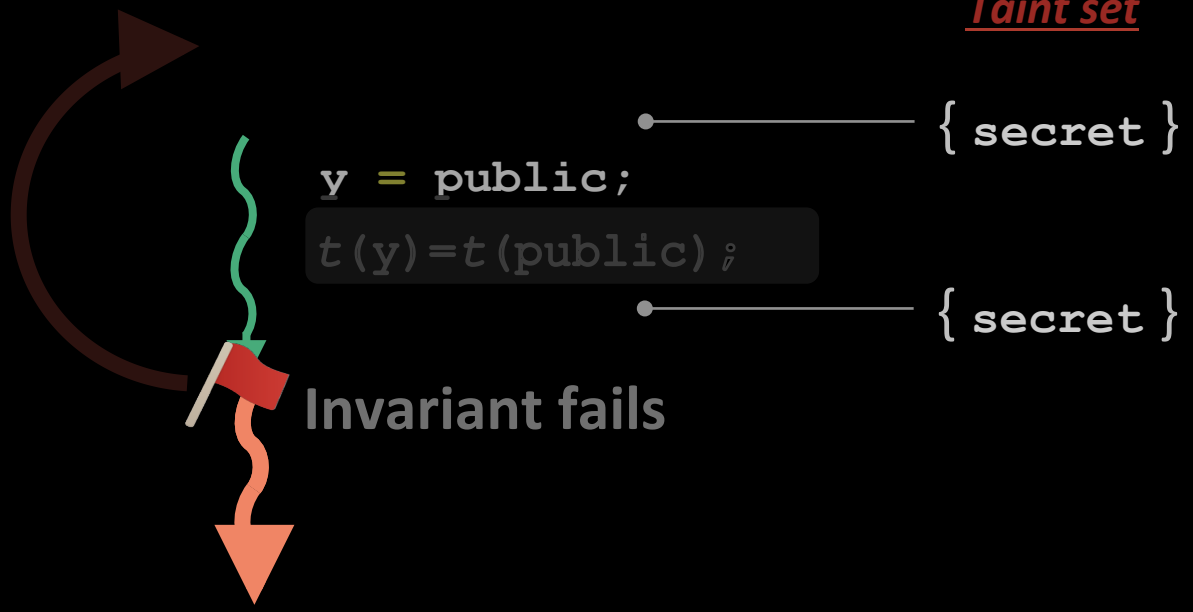
[ShadowReplica ‘13, TaintPipe ‘15, StraightTaint ‘16]

## Improves Optimistic Hybrid Analysis

- **Rollback-free** using only ***safe elisions***
- **Profiling** using **test suites** is adequate

# Safety Guarantee

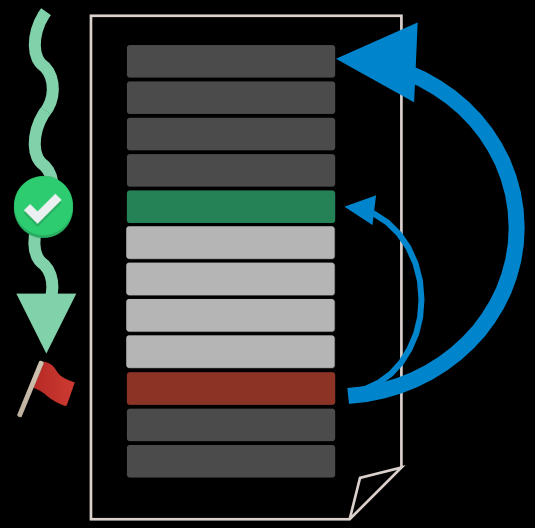


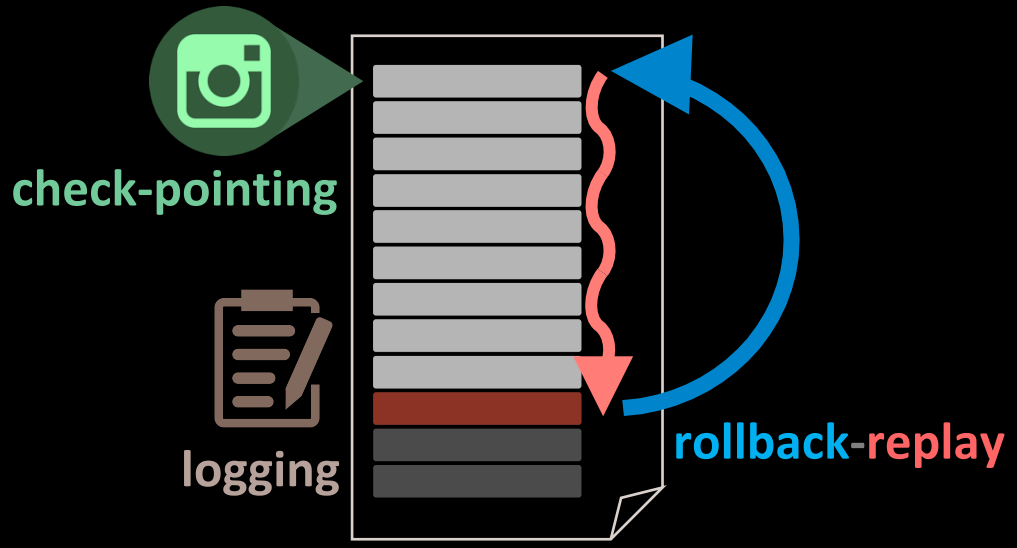

*ensures metadata equivalence !*

*exact semantics*

# Rollbacks!

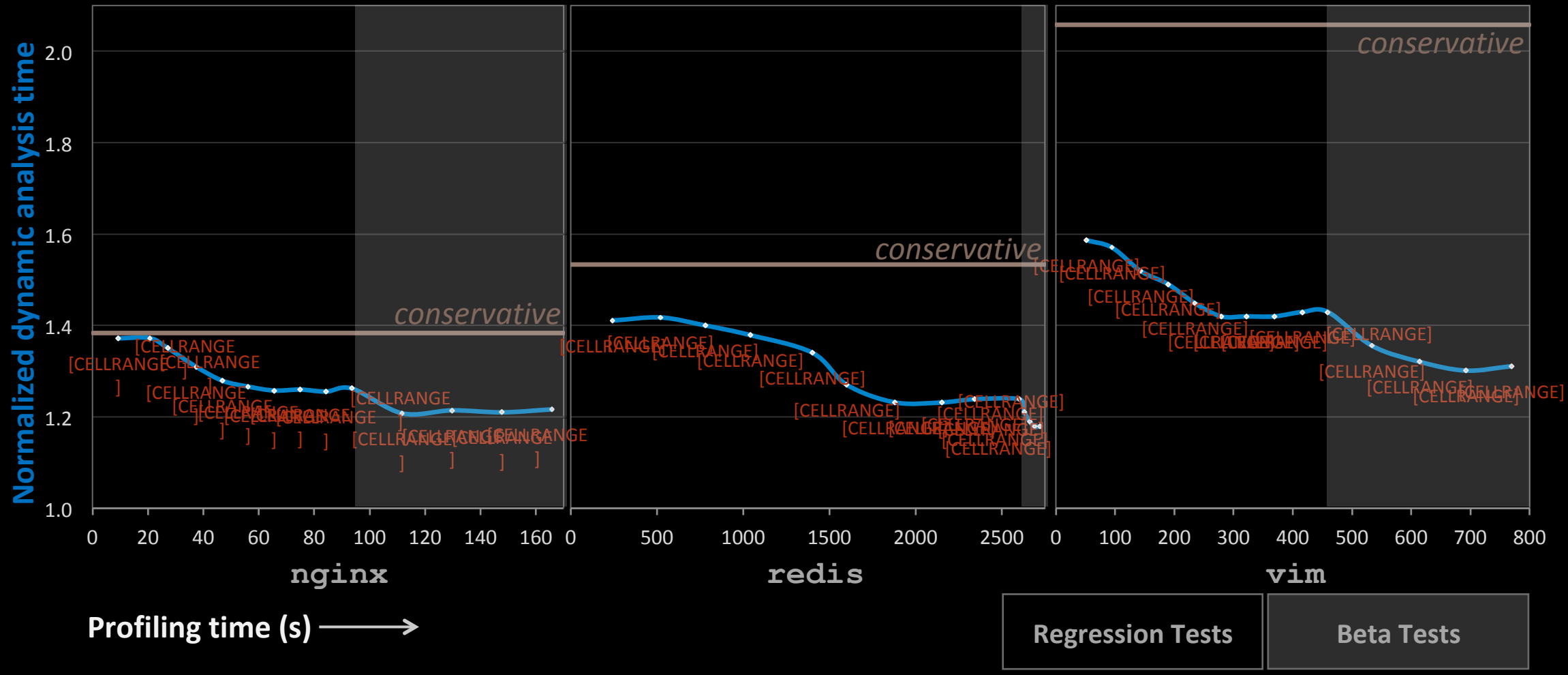


Unbounded Rollbacks




Overheads !

# Sensitivity to Profiling



# Attacks on Availability

***New Attack Vector*** : violate likely invariants

***Bounded Slowdown*** : best available conservative analysis

***Adapting Invariants*** : re-analyze excluding failed invariant

***Early Detection*** : forces attacker to induce unusual behavior