# UPGRADVISOR: Early Adopting Dependency Updates Using Hybrid Program Analysis and Hardware Tracing

**Yaniv David**[1], Xudong Sun[2], Raphael J Sofaer[1], Aditya Senthilnathan[3],
Junfeng Yang[1], Zhiqiang Zuo[2], Guoqing Harry Xu[4], Jason Nieh[1] and Ronghui Gu[1]

[1]Columbia University [2]Nanjing University [3]IIT Delhi [4]UCLA

# Modern Software Development is fast-paced

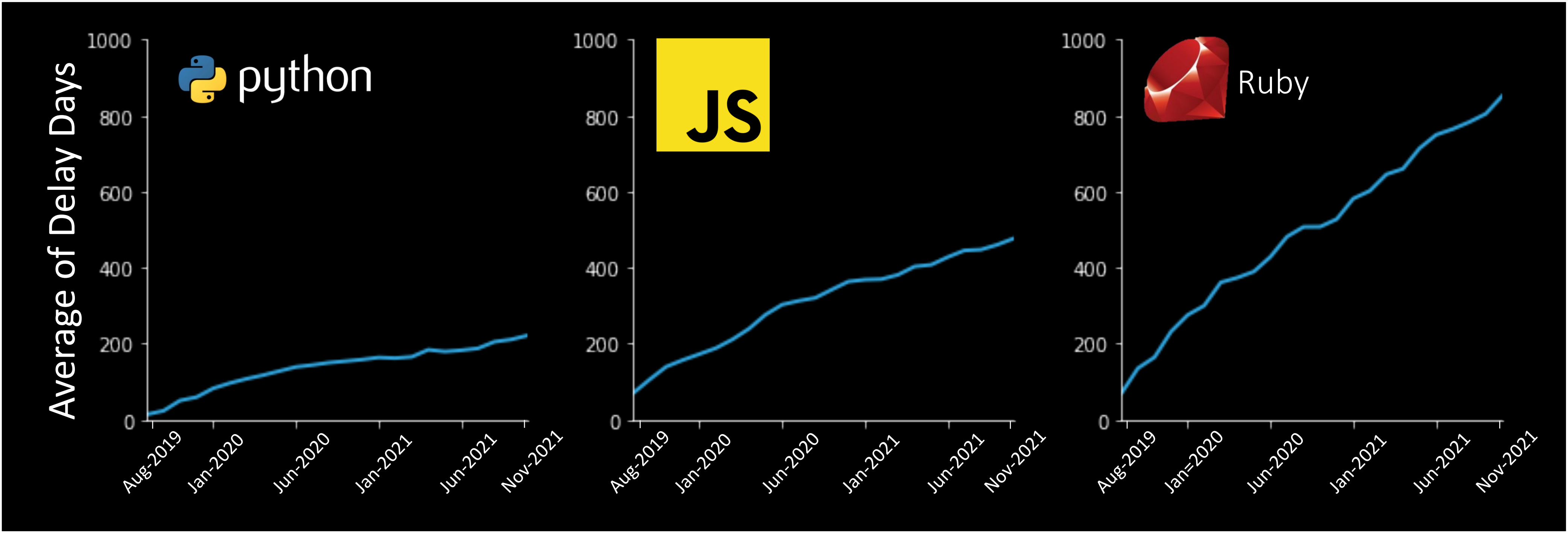
- Facebook(Meta) updates their front-end **three times a day**, and release new iOS and Android apps **every week**

- A key enabler for new features is **pre-exiting libraries**

- Average of 12 direct dependencies 100+ transitive dependencies
  (Our survey of top-stared Python, JS, Ruby GitHub projects)

# Dependency Update Adoption Is Slow

- Dependencies' developers are releasing updates frequently, too

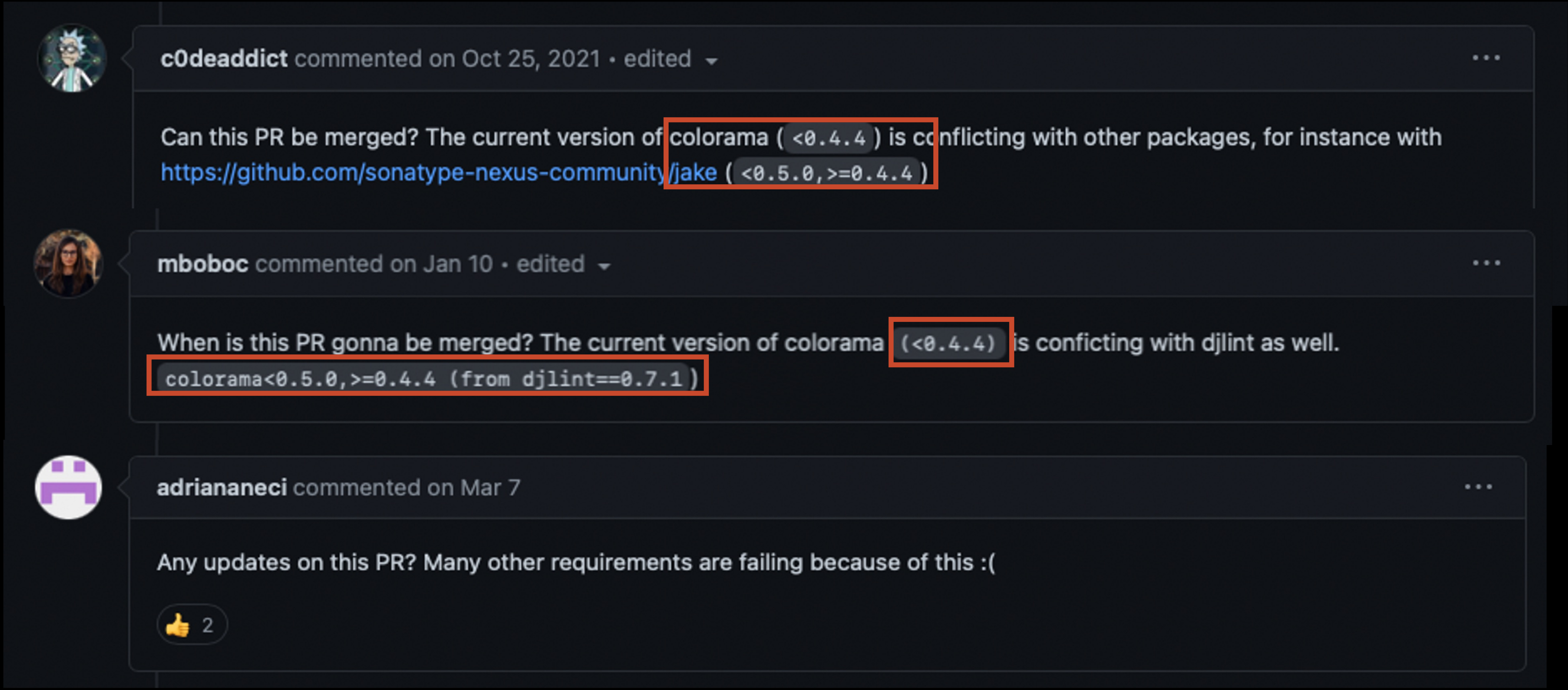


- Currently averaging **400 days** in update adaption delay

# Dependency Delays Lead to Bad Consequences

- Fixed bugs in dependencies continue to affect applications
- Closed security holes put dependent applications at risk
- Conflicts arise in transitive dependency graphs
  - Some can be resolved by using the oldest supported version
  - Other fall into a “dependency hell”

# Dependency Delays Lead to Bad Consequences



**c0deaddict** commented on Oct 25, 2021 • edited

Can this PR be merged? The current version of colorama ( `<0.4.4` ) is conflicting with other packages, for instance with https://github.com/sonatype-nexus-community/jake ( `<0.5.0,>=0.4.4` )

**mboboc** commented on Jan 10 • edited

When is this PR gonna be merged? The current version of colorama `(<0.4.4)` is conficting with djlint as well. `colorama<0.5.0,>=0.4.4 (from djlint==0.7.1` )

**adriananeci** commented on Mar 7

Any updates on this PR? Many other requirements are failing because of this :(

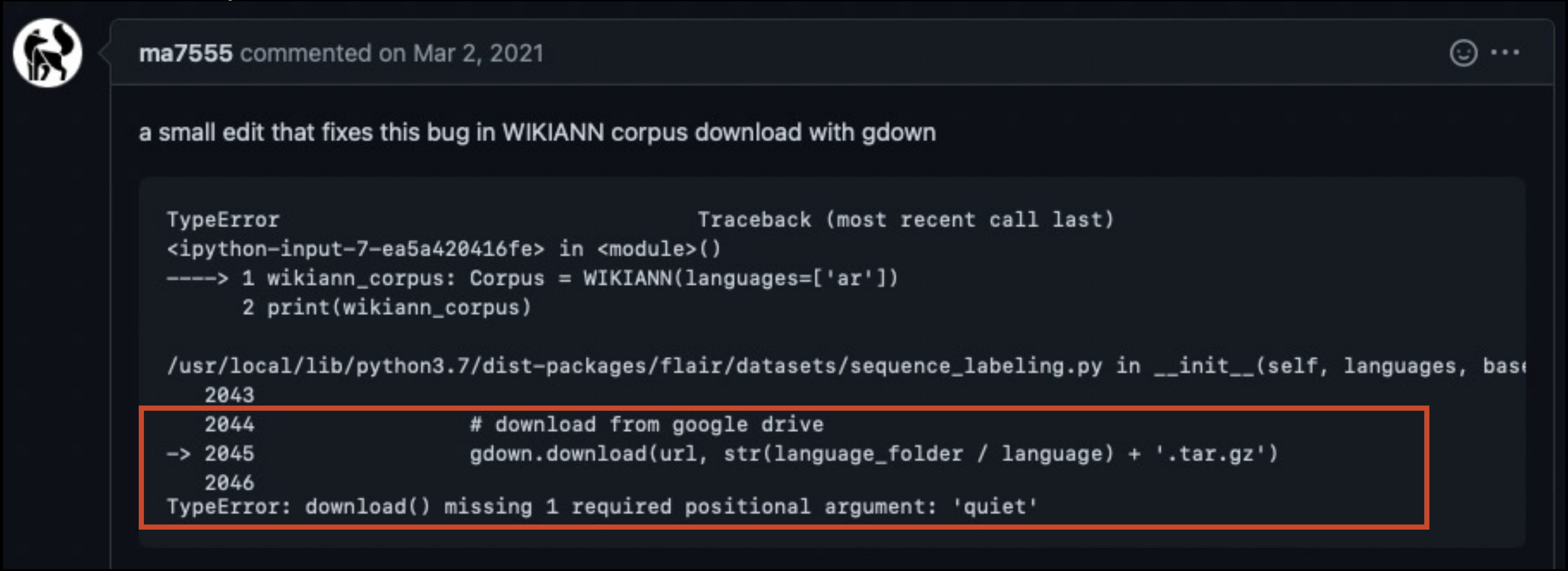
👍 2

# Naïve Solutions Fall Short

- As dependency APIs change, blindly updating might fail
  - Noisy run-time crash

# Naïve Solutions Fall Short

- As dependency APIs change, blindly updating might fail
    - Noisy run-time crash



ma7555 commented on Mar 2, 2021

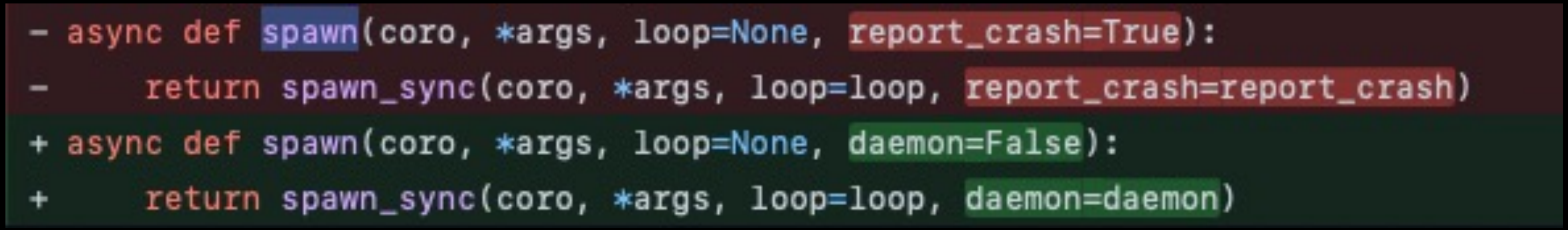
a small edit that fixes this bug in WIKIANN corpus download with gdown

```
TypeError                                 Traceback (most recent call last)
<ipython-input-7-ea5a420416fe> in <module>()
----> 1 wikiann_corpus: Corpus = WIKIANN(languages=['ar'])
      2 print(wikiann_corpus)

/usr/local/lib/python3.7/dist-packages/flair/datasets/sequence_labeling.py in __init__(self, languages, base
   2043
   2044                 # download from google drive
-> 2045                 gdown.download(url, str(language_folder / language) + '.tar.gz')
   2046
TypeError: download() missing 1 required positional argument: 'quiet'
```

# Naïve Solutions Fall Short

- As dependency APIs change, blindly updating might fail
  - Noisy run-time crash
  - Or worse, fail silently



```
- async def spawn(coro, *args, loop=None, report_crash=True):
-     return spawn_sync(coro, *args, loop=loop, report_crash=report_crash)
+ async def spawn(coro, *args, loop=None, daemon=False):
+     return spawn_sync(coro, *args, loop=loop, daemon=daemon)
```

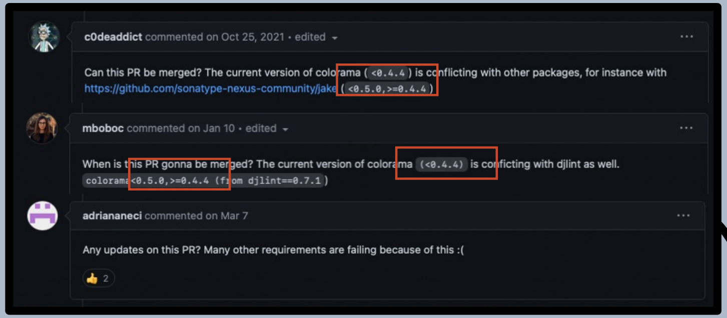
# Naïve Solutions Fall Short

- As dependency APIs change, blindly updating might fail
  - Noisy run-time crash
  - Or worse, fail silently
- Integration tests fail or don’t even try covering dependency interfaces[1]

[1] Joseph Hejderup & Georgios Gousios, “Can we trust tests to automate dependency updates? A case study of Java Projects”, Journal of Systems and Software

# Naïve Solutions Fall Short

- As dependency APIs change
  - Noisy run-time crash
  - Or worse, fail silently



c0deaddict commented on Oct 25, 2021 • edited

Can this PR be merged? The current version of colorama ( `<0.4.4` ) is conflicting with other packages, for instance with https://github.com/sonatype-nexus-community/jake ( `<0.5.0,>=0.4.4` )

mboboc commented on Jan 10 • edited

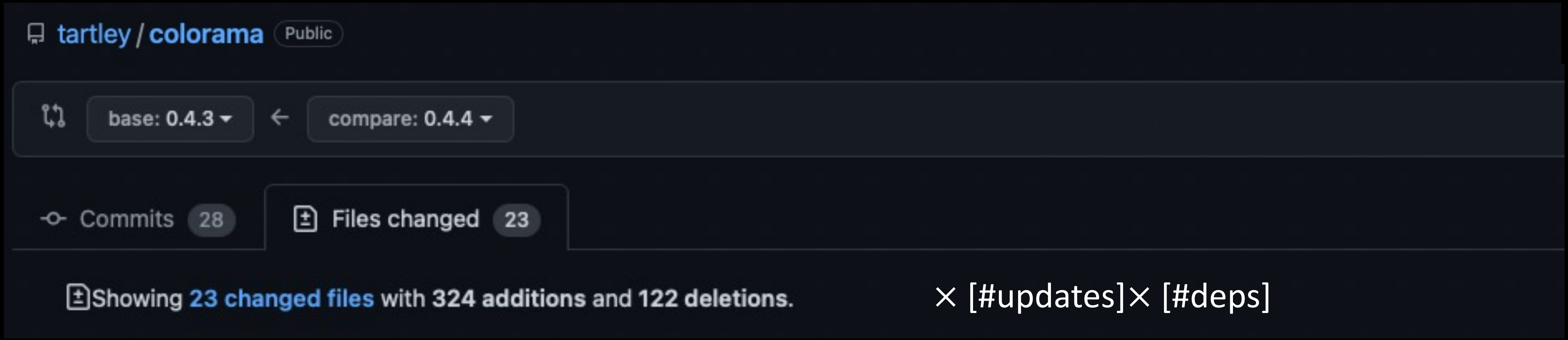
When is this PR gonna be merged? The current version of colorama `(<0.4.4)` is conficting with djlint as well. `colorama<0.5.0,>=0.4.4 (from djlint==0.7.1` )

adriananeci commented on Mar 7

Any updates on this PR? Many other requirements are failing because of this :(

👍 2

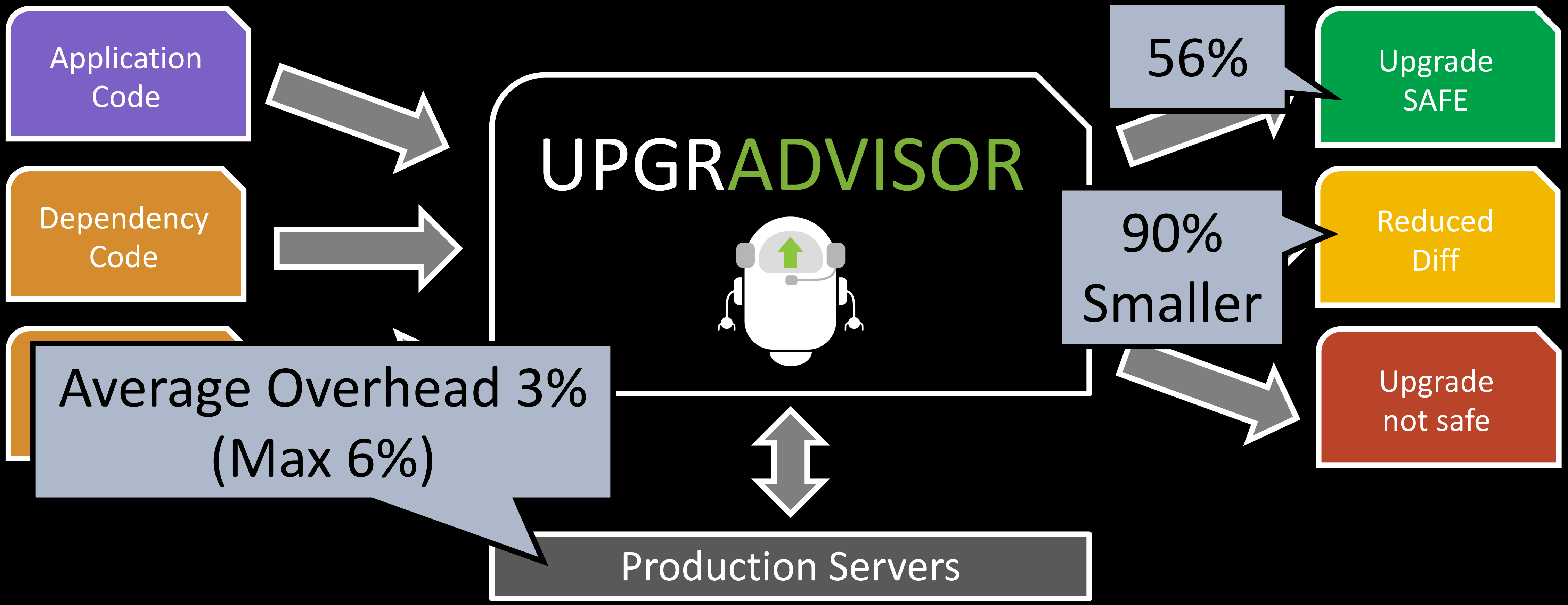
- Integration tests fail or don’t even try covering dependency interfaces[1]
- Manual inspection is not feasible



Showing 23 changed files with 324 additions and 122 deletions.

× [#updates]× [#deps]

# UPGRADVISOR: Upgrade-Advisor



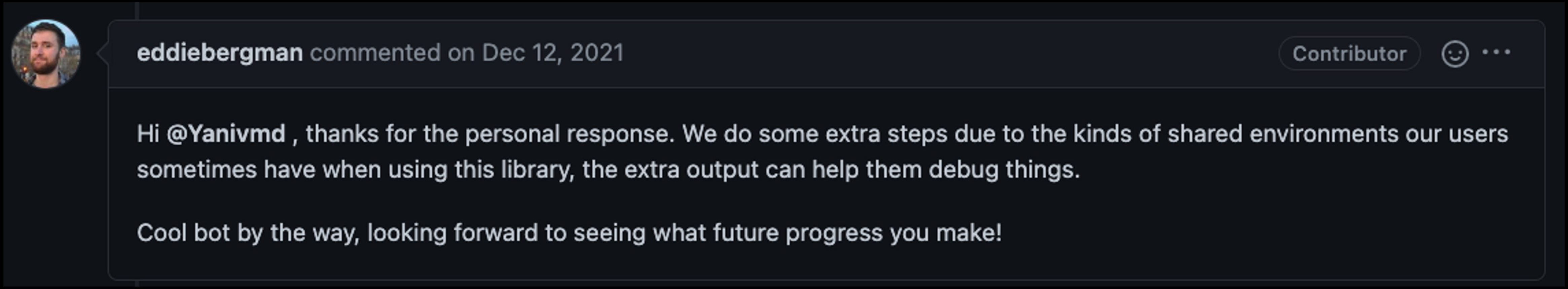
Average Overhead 3% (Max 6%)

56%

90% Smaller

Built UPGRADVISOR-Python3 and evaluated on 172 dependency updates

# UPGRADVISOR: Upgrade-Advisor

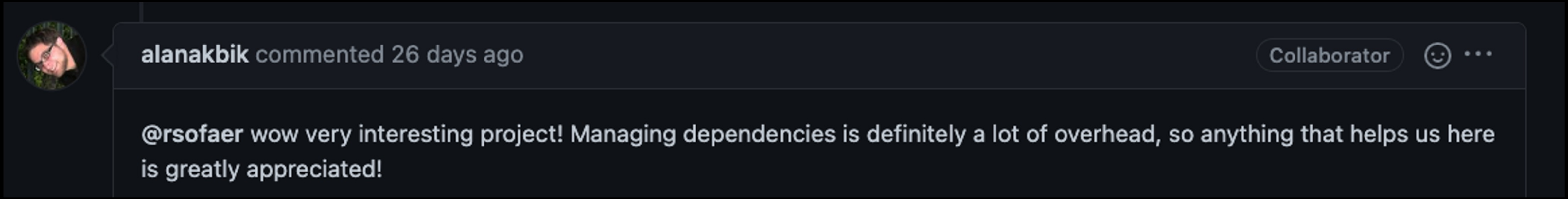


**eddiebergman** commented on Dec 12, 2021

Hi **@Yanivmd** , thanks for the personal response. We do some extra steps due to the kinds of shared environments our users sometimes have when using this library, the extra output can help them debug things.

Cool bot by the way, looking forward to seeing what future progress you make!

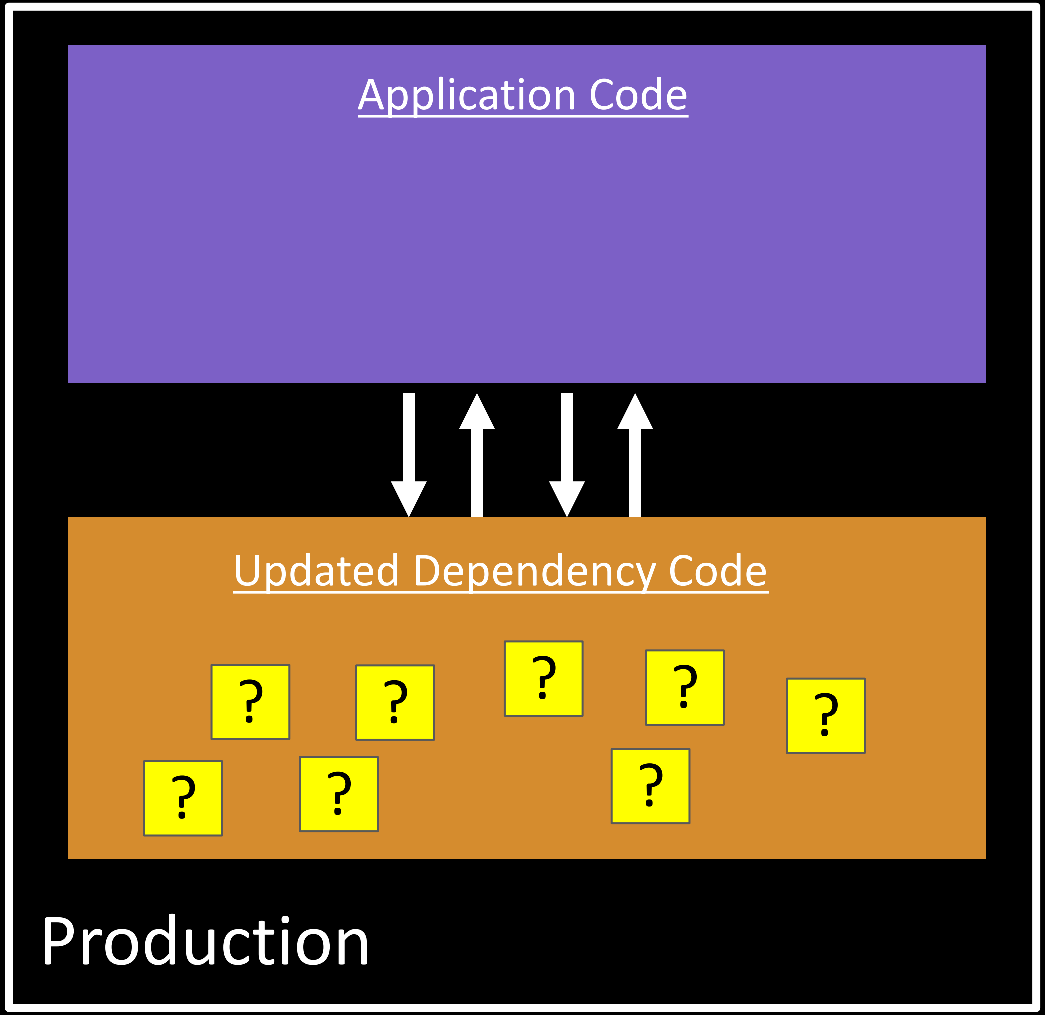
Manually submitted a sample of 9 PR
7 already merged



**alanakbik** commented 26 days ago

**@rsofaer** wow very interesting project! Managing dependencies is definitely a lot of overhead, so anything that helps us here is greatly appreciated!

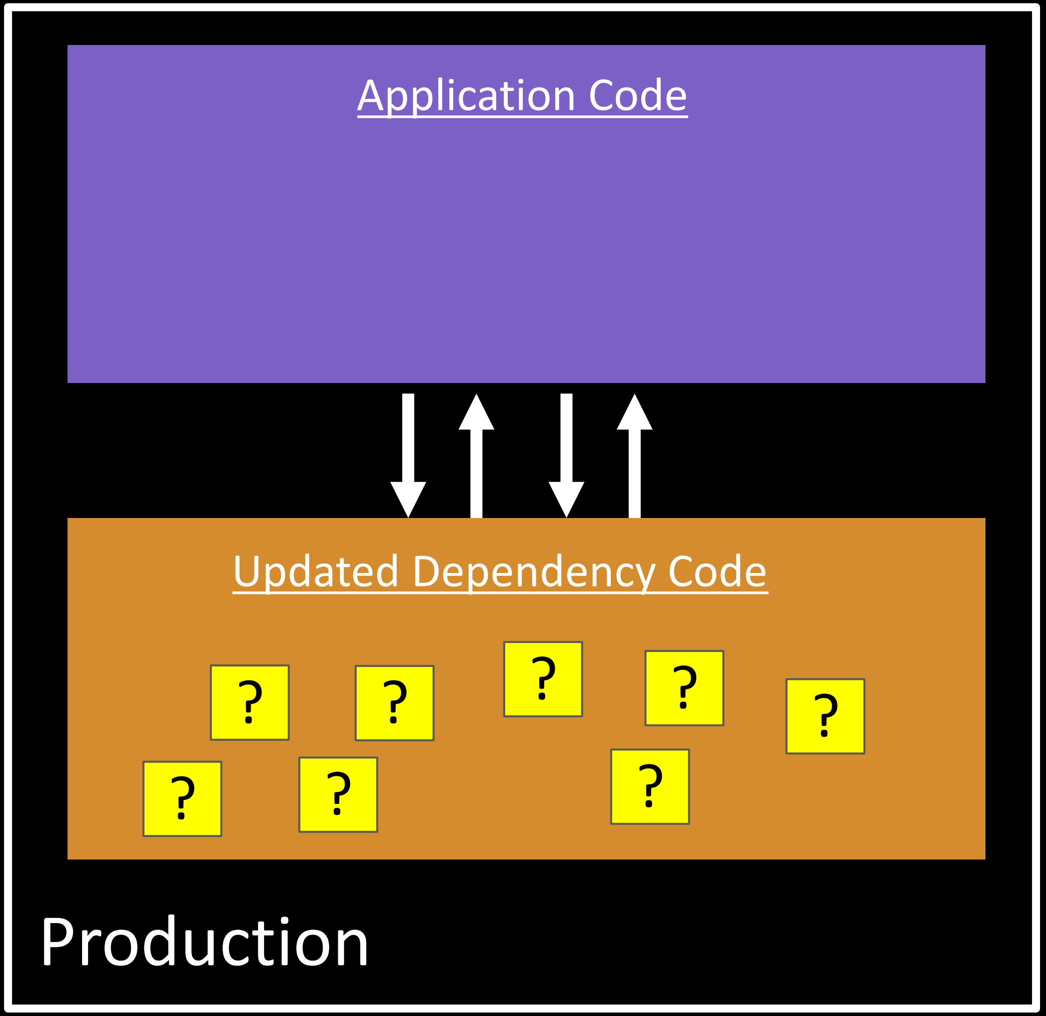
# Insight: What You Can’t Reach Won’t Hurt You



# Insight: What You Can’t Reach Won’t Hurt You

Requirements for production run:

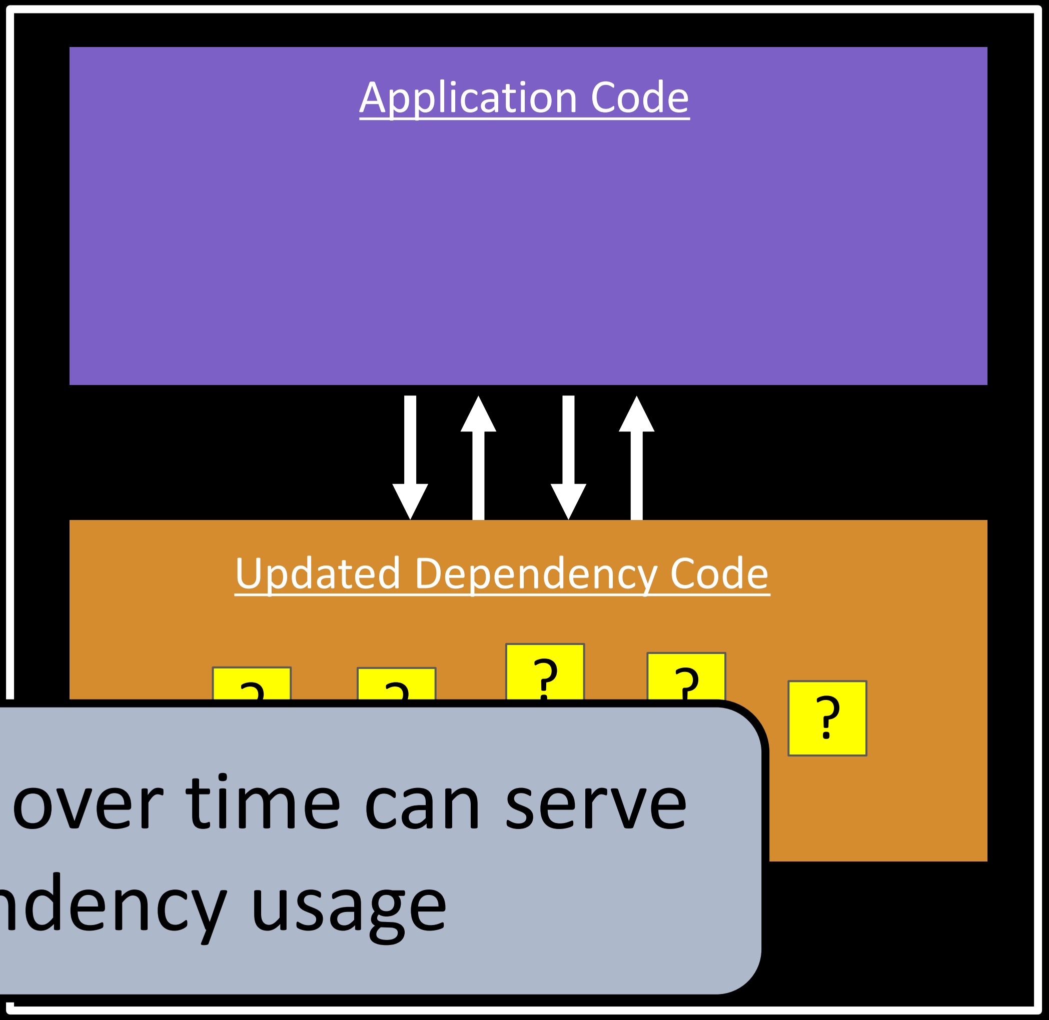
- Study the update **without** applying it
- No interruption
- Incur low overhead



# Insight: What You Can't Reach Won't Hurt You

Requirements for production run:

- Study the update **without** applying it
- No interruption
- Incur low overhead

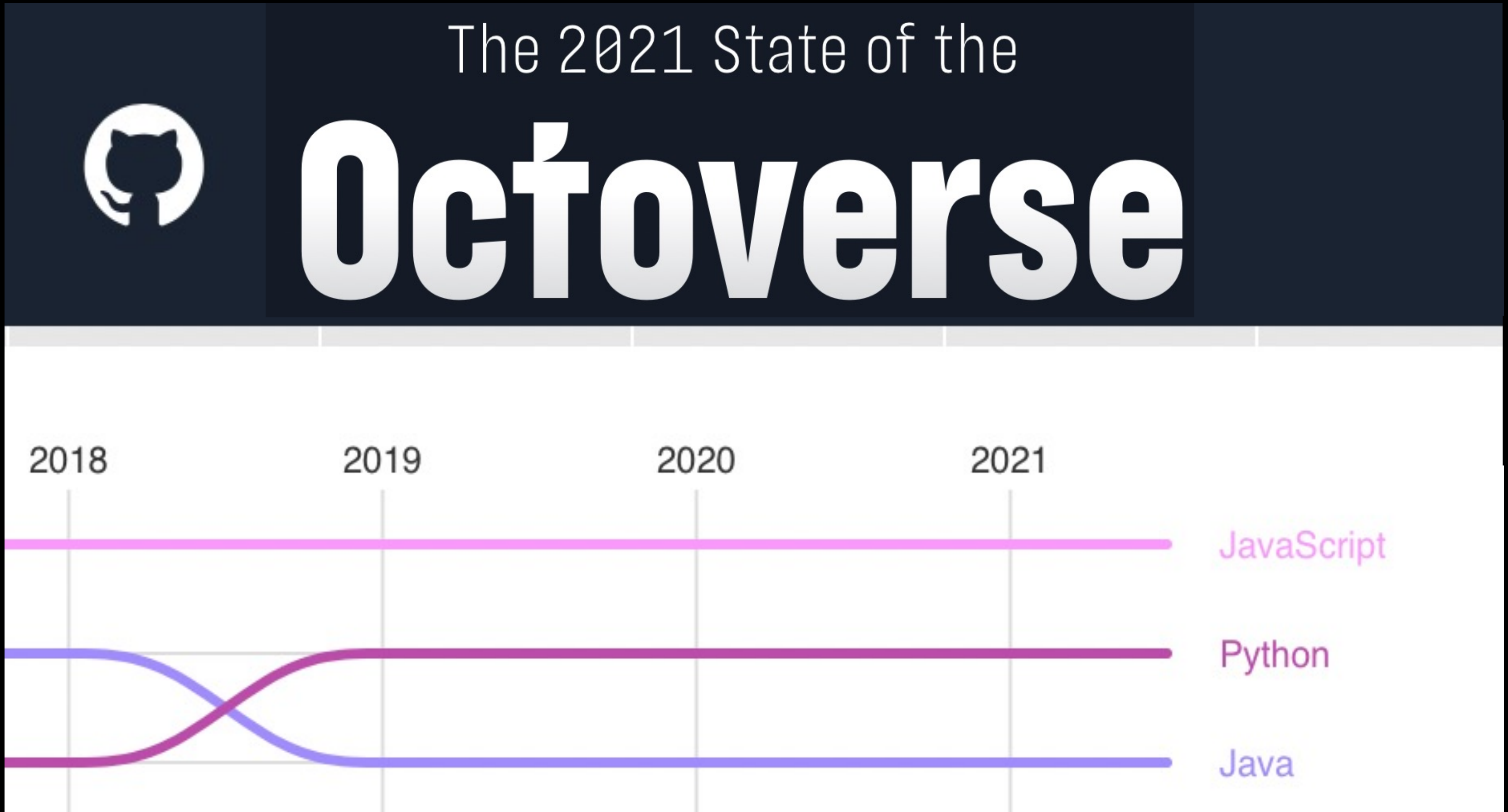


Tracing production environment over time can serve as **ground truth** for dependency usage

# Key Ideas Driving UPGRADVISOR's Design

- Safely discard non-reachable changes via hybrid program analysis
  - Static analysis to discard **never-reachable** changes
  - Dynamic analysis to test **maybe-reachable** changes
- Achieve low-overhead by employing hardware-based tracer

# Key Ideas Driving UPGRADVISOR’s Design



| | |
|---|---|
| npm | 2.42 M |
| python™ Package Index | 455 K |
| RubyGems | 179 K |

# Key Ideas Driving UPGRADVISOR’s Design

- Safely discard non-reachable changes via hybrid program analysis
  - Static analysis to discard **never-reachable** changes
  - Dynamic analysis to test **maybe-reachable** changes

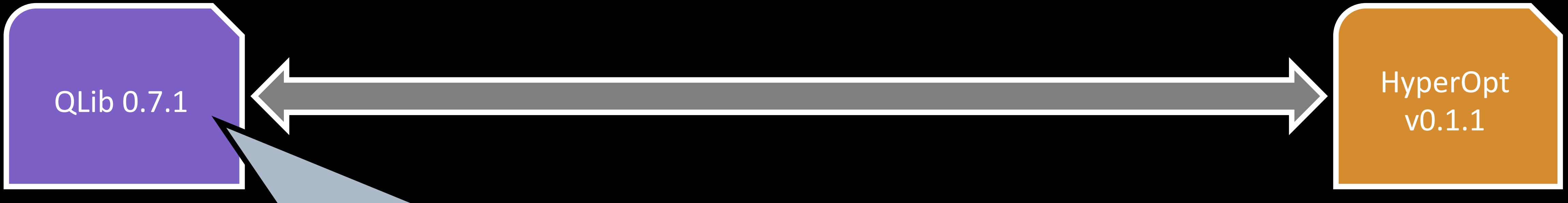
No Types

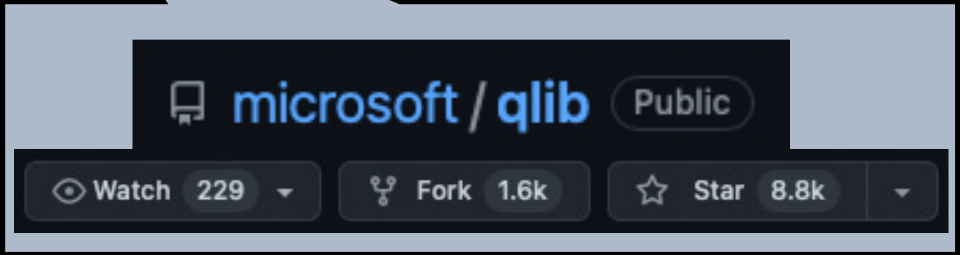
- Achieve low-overhead by employing hardware-based tracer

Interpreted

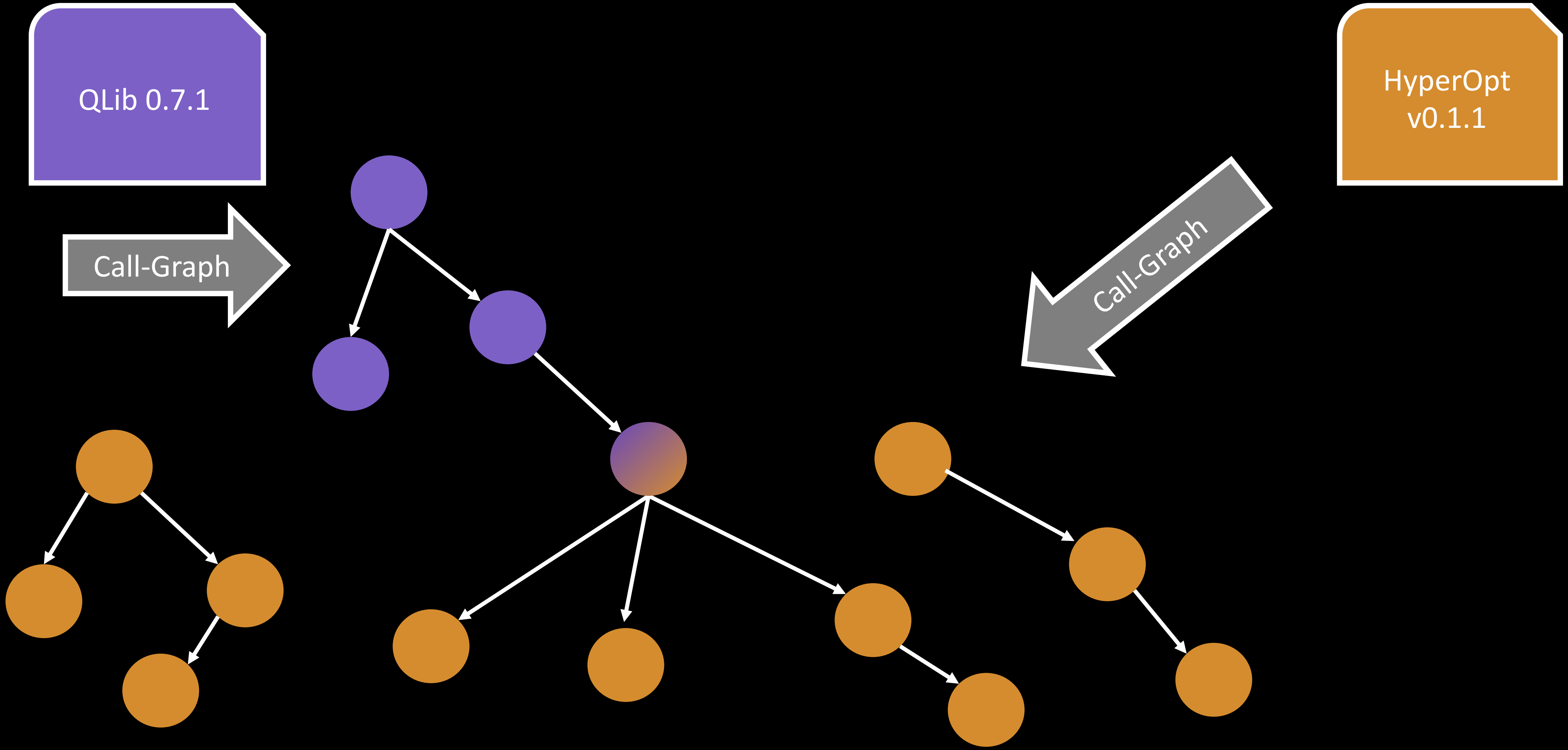
- Design for dynamic languages to maximize usability

# Analyzing Our Motivating Example

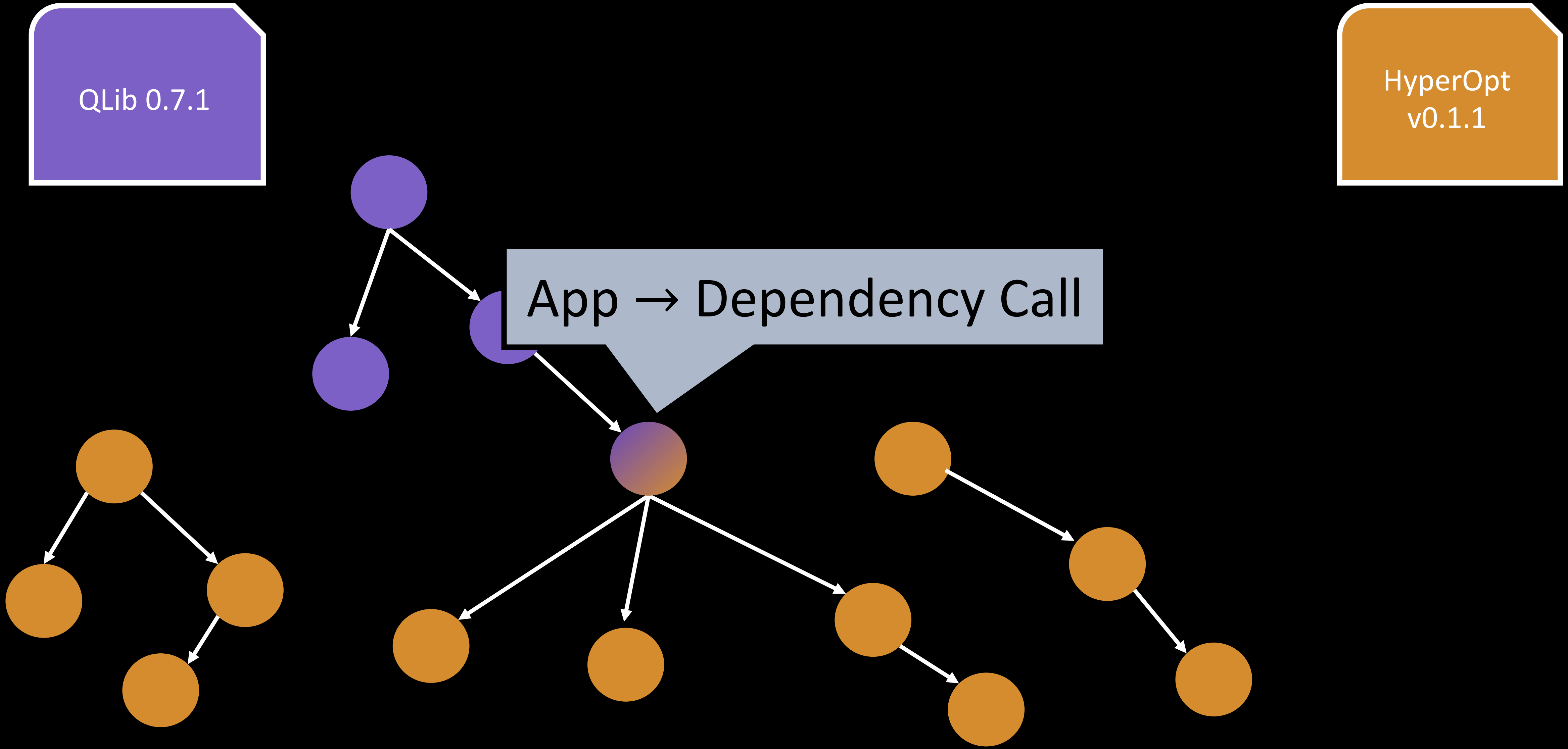




# Analyzing Our Motivating Example

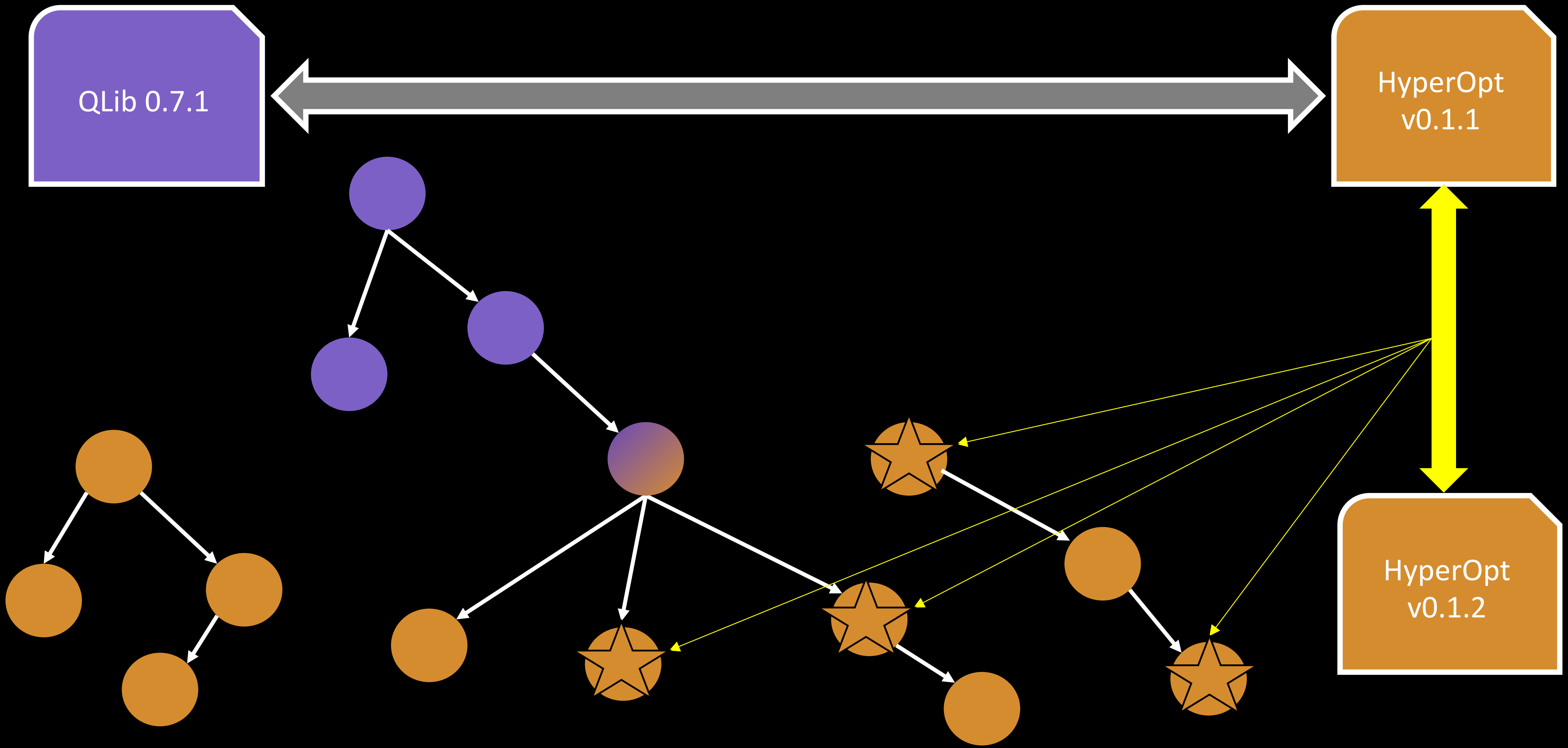


# Analyzing Our Motivating Example

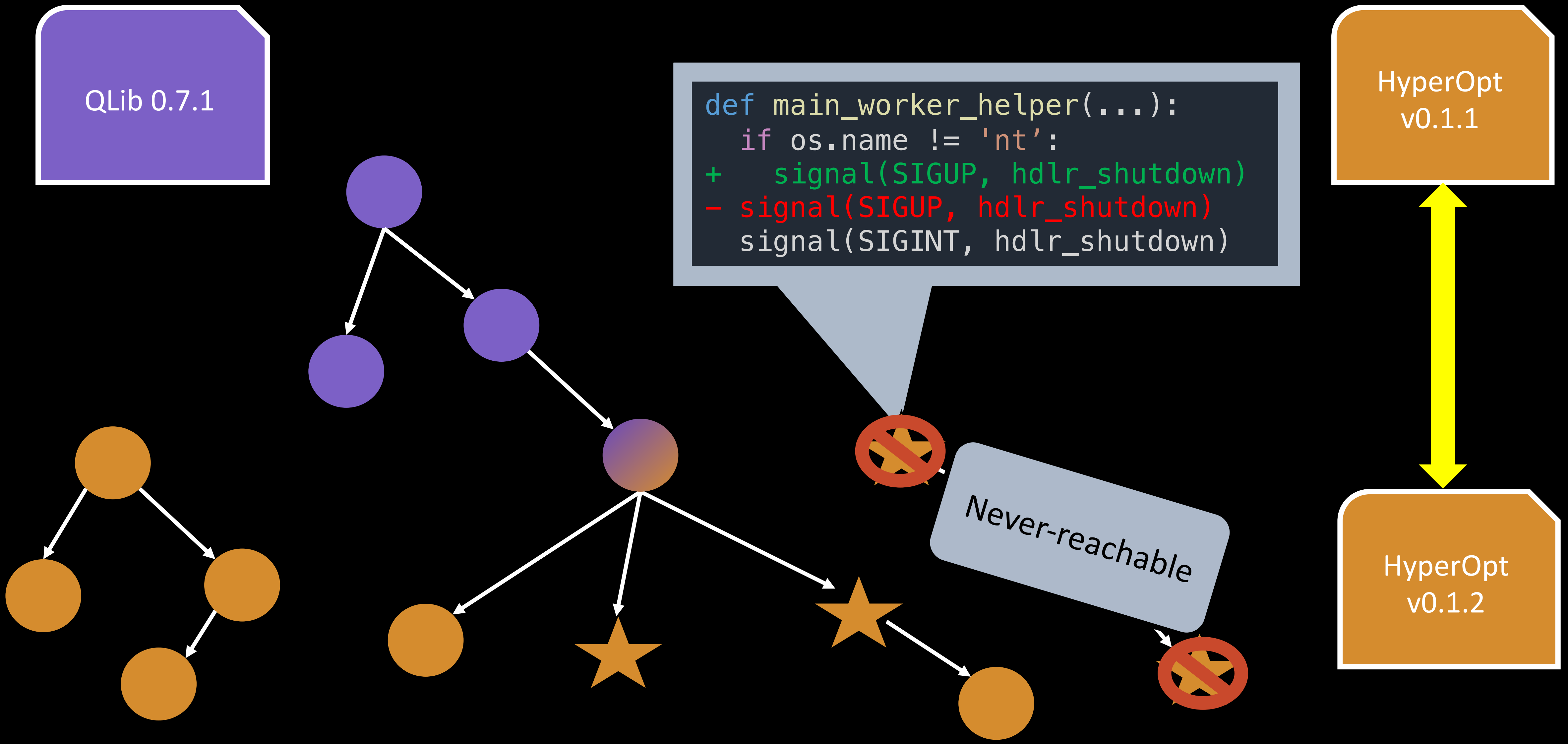


App → Dependency Call

# Analyzing Our Motivating Example



# Classify Changes In Motivating Example



QLib 0.7.1

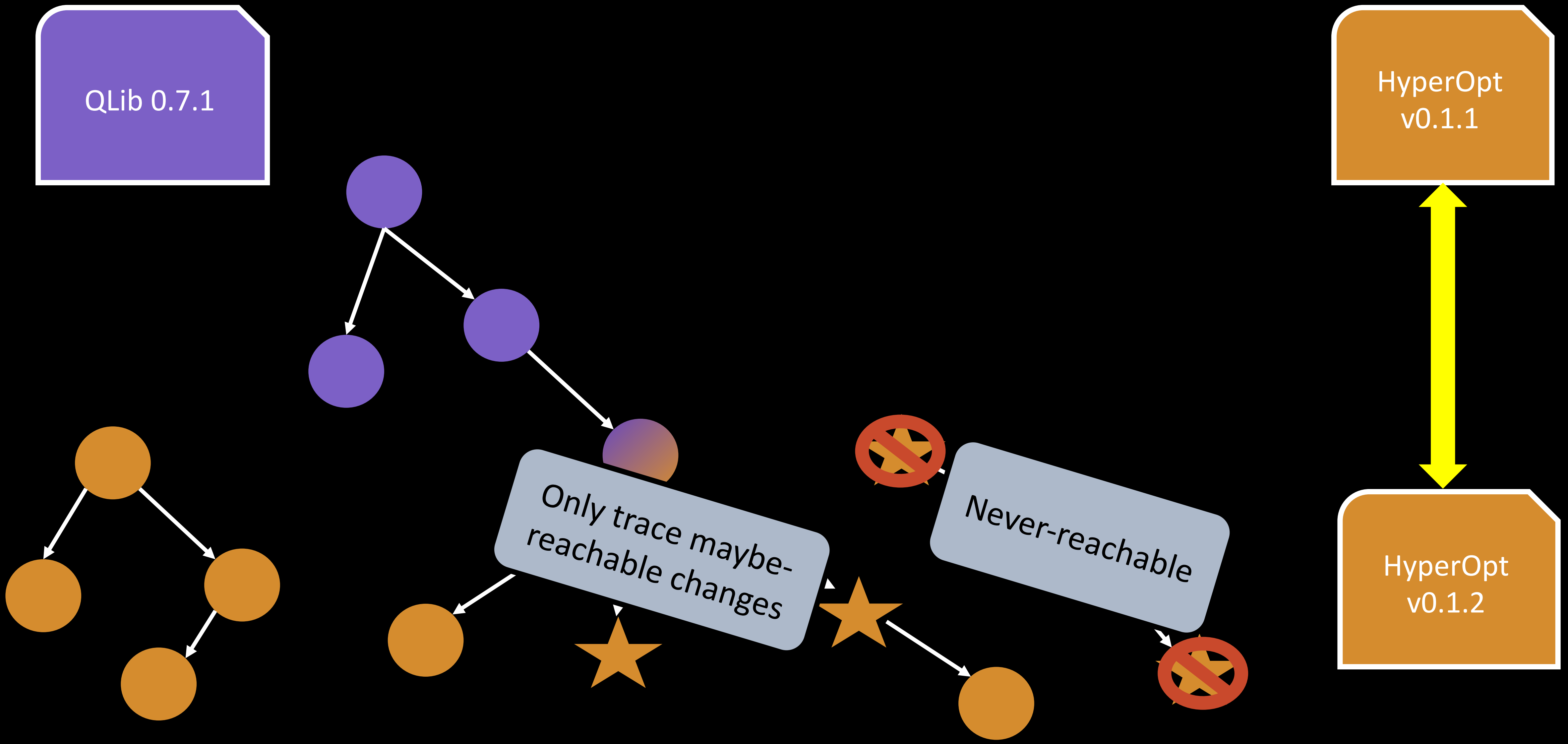
```
def main_worker_helper(...):
  if os.name != 'nt’:
+   signal(SIGUP, hdlr_shutdown)
- signal(SIGUP, hdlr_shutdown)
  signal(SIGINT, hdlr_shutdown)
```

Never-reachable

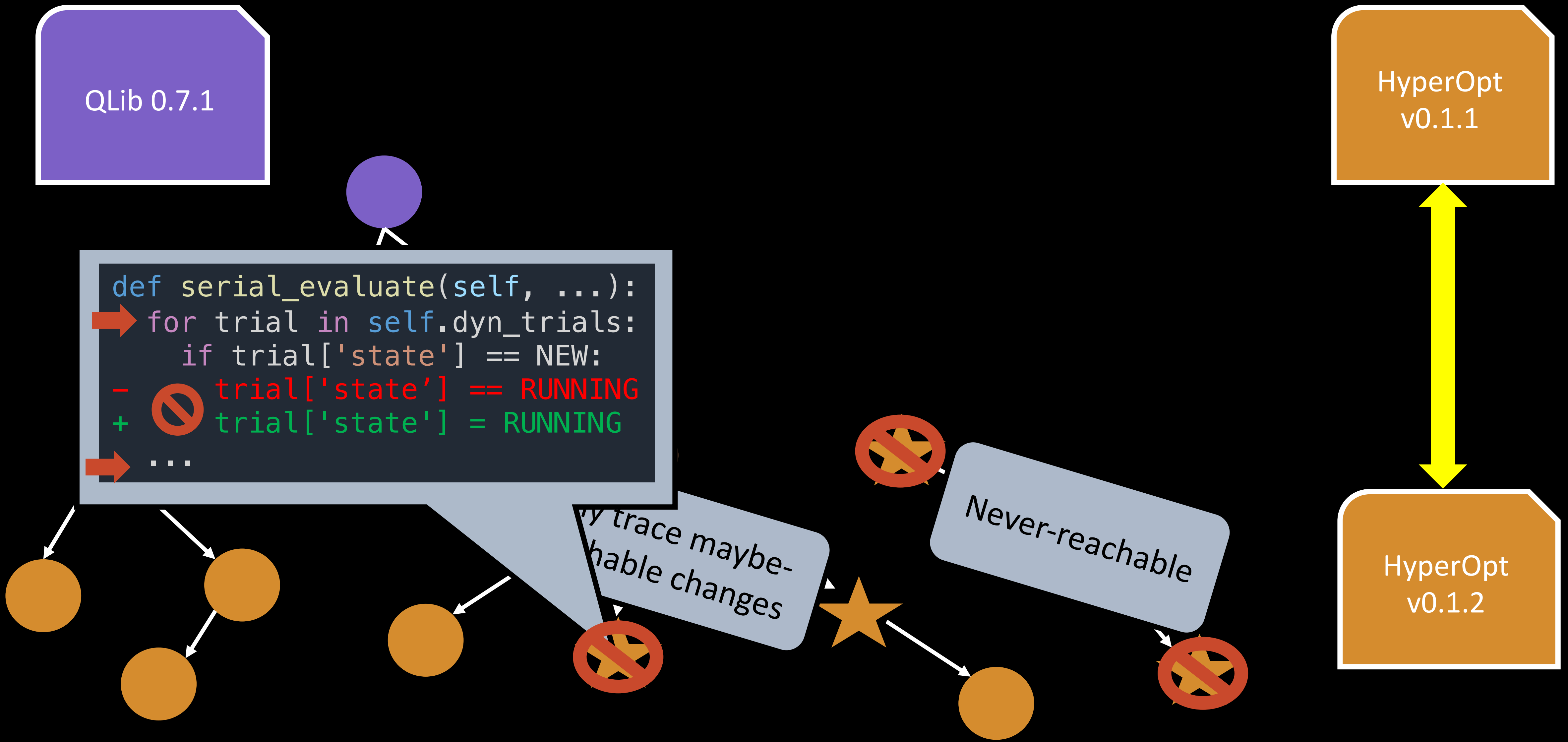
HyperOpt v0.1.1

HyperOpt v0.1.2

# Classify Changes In Motivating Example

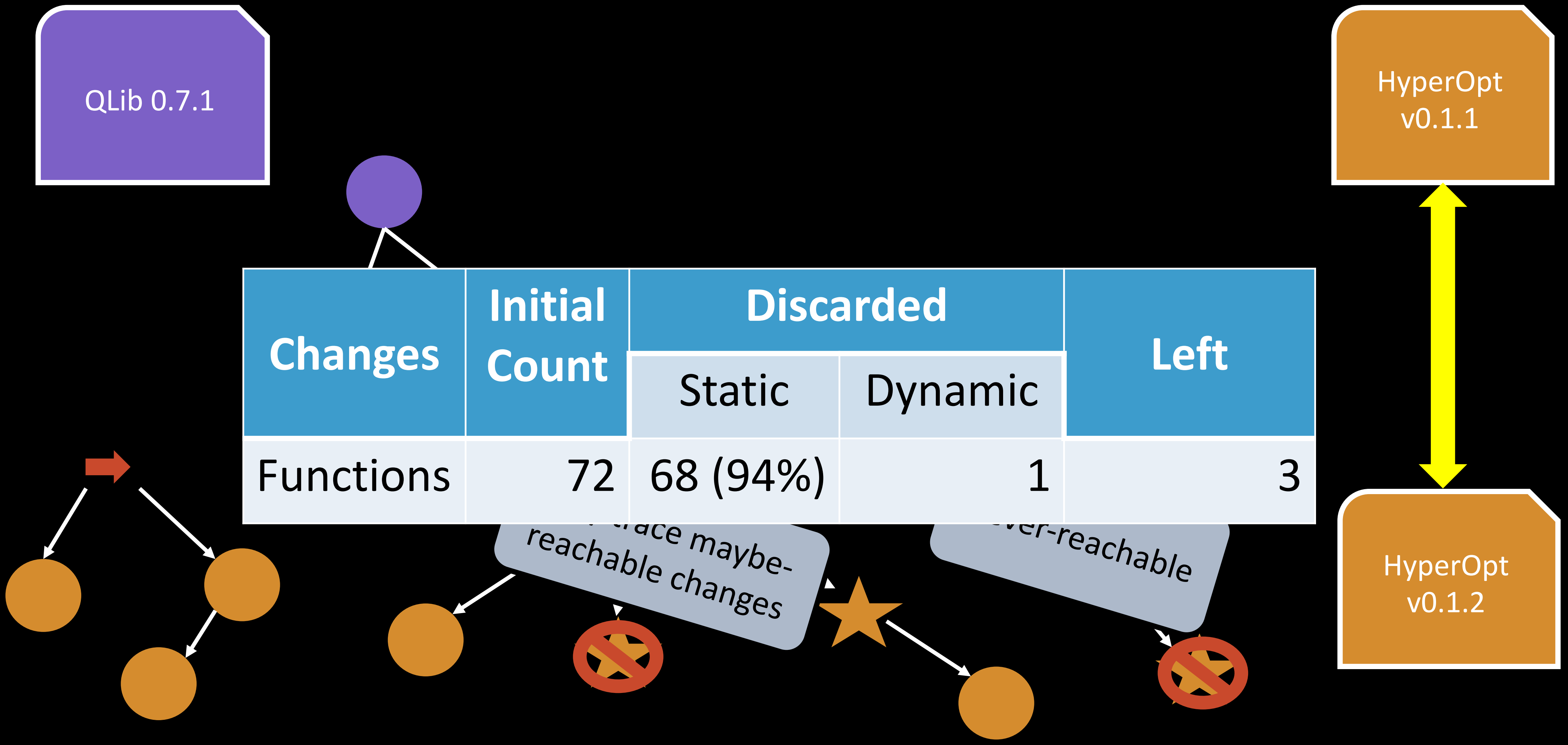


# Classify Changes In Motivating Example



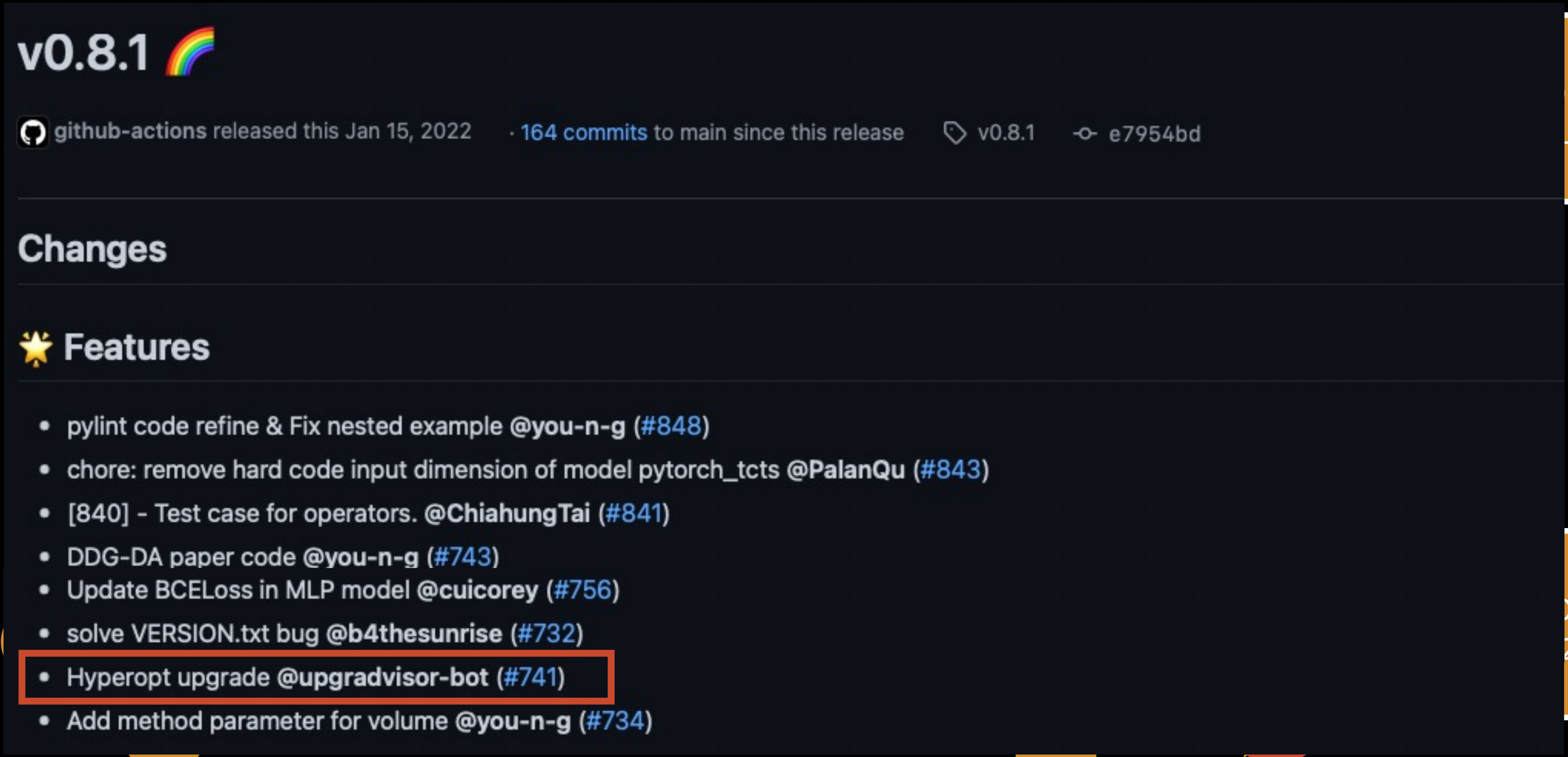
```
def serial_evaluate(self, ...):
    for trial in self.dyn_trials:
        if trial['state'] == NEW:
-           trial['state’] == RUNNING
+           trial['state'] = RUNNING
    ...
```

# Classify Changes In Motivating Example



| Changes | Initial Count | Discarded | | Left |
|---|---|---|---|---|
| | | Static | Dynamic | |
| Functions | 72 | 68 (94%) | 1 | 3 |

# Classify Changes In Motivating Example



## v0.8.1 🌈

github-actions released this Jan 15, 2022 · 164 commits to main since this release  v0.8.1  e7954bd

### Changes

### 🌟 Features

- pylint code refine & Fix nested example **@you-n-g** (#848)
- chore: remove hard code input dimension of model pytorch_tcts **@PalanQu** (#843)
- [840] - Test case for operators. **@ChiahungTai** (#841)
- DDG-DA paper code **@you-n-g** (#743)
- Update BCELoss in MLP model **@cuicorey** (#756)
- solve VERSION.txt bug **@b4thesunrise** (#732)
- Hyperopt upgrade **@upgradvisor-bot** (#741)
- Add method parameter for volume **@you-n-g** (#734)

# Key Challenges for Designing UPGRADVISOR

- Hybrid program analysis to safely discard non-reachable changes
  - Safely discard non-reachable changes via hybrid program analysis
    - Create sound call-graphs
    - Reachable-but-non-affecting changes
  - Dynamic analysis to test maybe-reachable changes
- Achieve low-overhead using a hardware-based tracer
    - Low-overhead & selective tracing

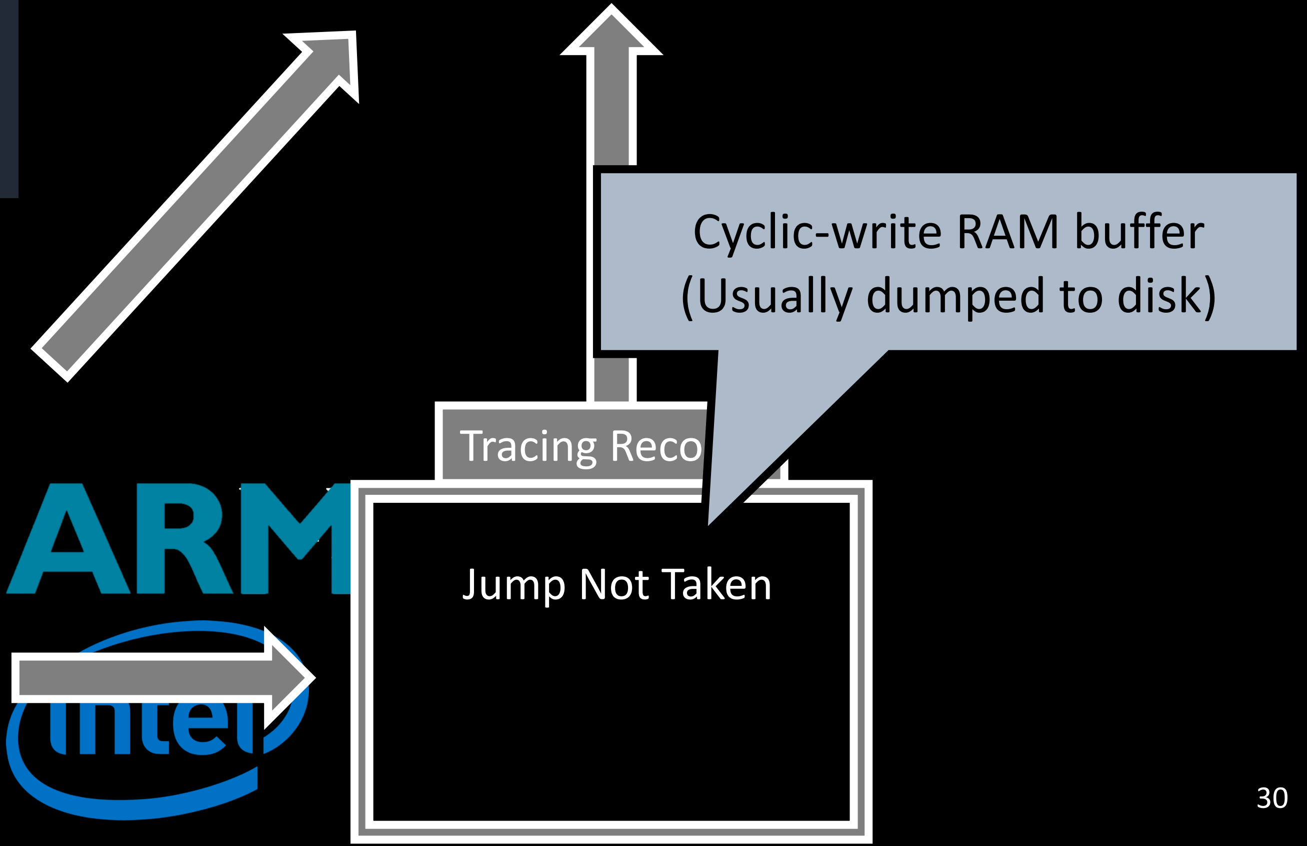
# Key Challenges for Designing UPGRADVISOR

- Hybrid program analysis to safely discard non-reachable changes
  - Safely discard non-reachable changes via hybrid program analysis
    - Create sound call-graphs
    - Reachable-but-non-affecting changes
  - Dynamic analysis to test maybe-reachable changes
- Achieve low-overhead using a hardware-based tracer
    - Low-overhead & selective tracing

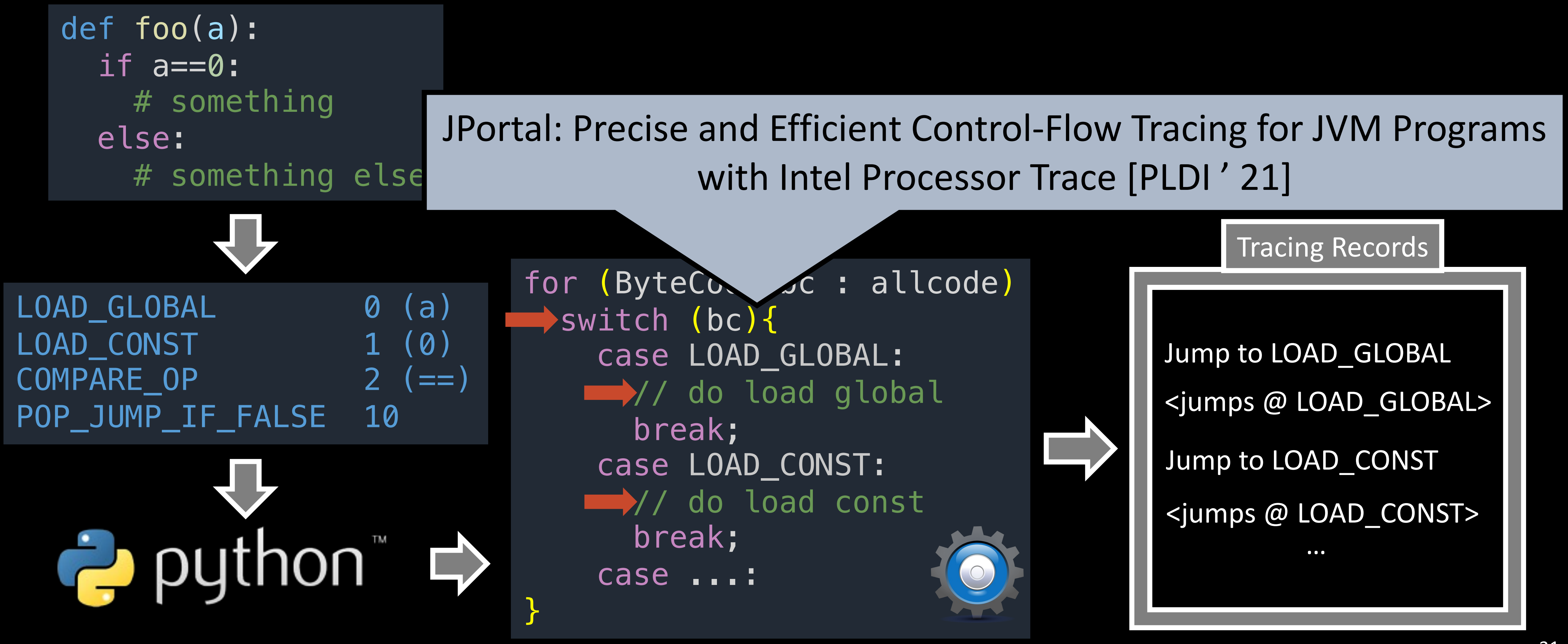
# Hardware Tracing for Native Code

```
void foo(int a){
  if (a==0){
    // something
  } else {
    // something else
  }
}
```

```
    cmp        rdi, 0
    jne        .EL
    nop
    jmp RET
.EL
    nop
.RET
    ret
```



# Hardware Tracing for Interpreter Code



```
def foo(a):
  if a==0:
    # something
  else:
    # something else
```

```
LOAD_GLOBAL          0 (a)
LOAD_CONST           1 (0)
COMPARE_OP           2 (==)
POP_JUMP_IF_FALSE   10
```

JPortal: Precise and Efficient Control-Flow Tracing for JVM Programs with Intel Processor Trace [PLDI ’ 21]

```
for (ByteCo    c : allcode)
  switch (bc){
    case LOAD_GLOBAL:
      // do load global
      break;
    case LOAD_CONST:
      // do load const
      break;
    case ...:
}
```

Tracing Records

Jump to LOAD_GLOBAL

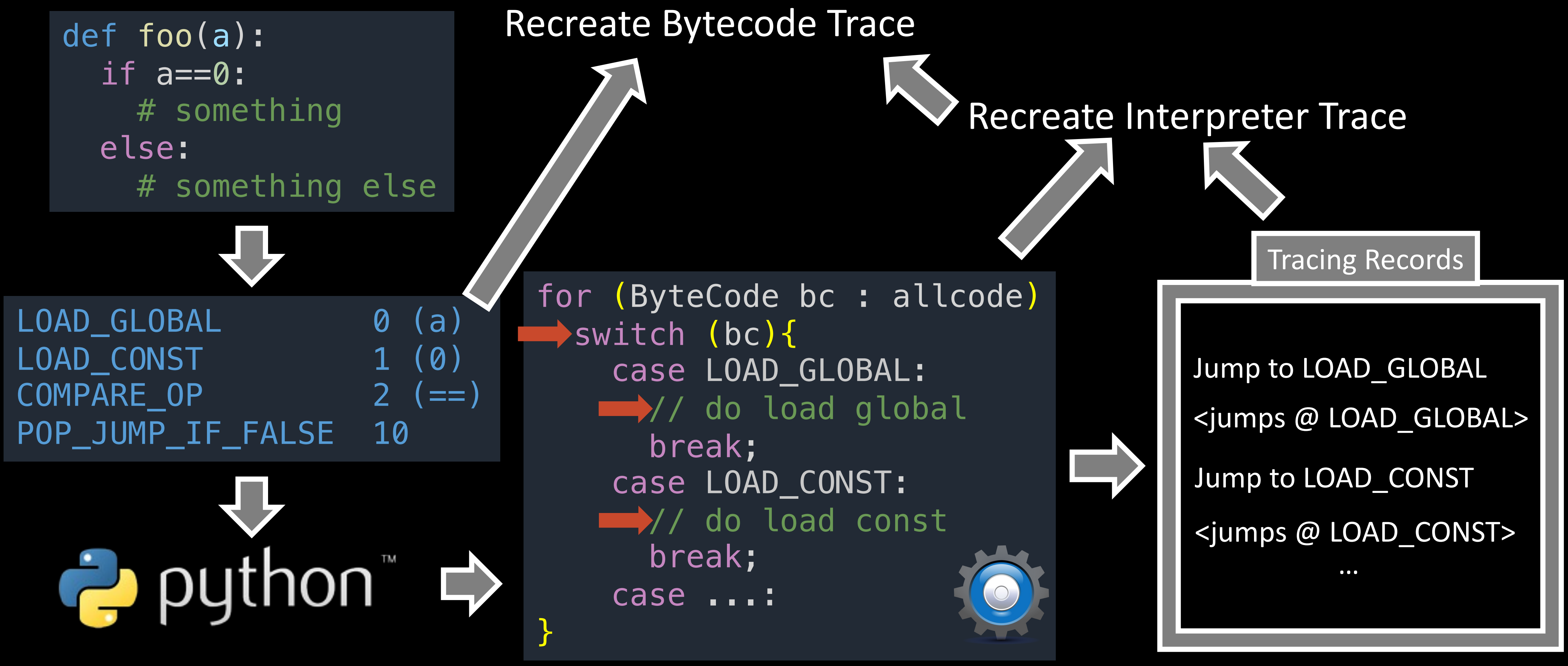
<jumps @ LOAD_GLOBAL>

Jump to LOAD_CONST

<jumps @ LOAD_CONST>

…

# Hardware Tracing for Interpreter Code



```
def foo(a):
  if a==0:
    # something
  else:
    # something else
```

```
LOAD_GLOBAL         0 (a)
LOAD_CONST          1 (0)
COMPARE_OP          2 (==)
POP_JUMP_IF_FALSE   10
```

python™

```
for (ByteCode bc : allcode)
switch (bc){
  case LOAD_GLOBAL:
    // do load global
    break;
  case LOAD_CONST:
    // do load const
    break;
  case ...:
}
```

Recreate Bytecode Trace

Recreate Interpreter Trace

Tracing Records

Jump to LOAD_GLOBAL

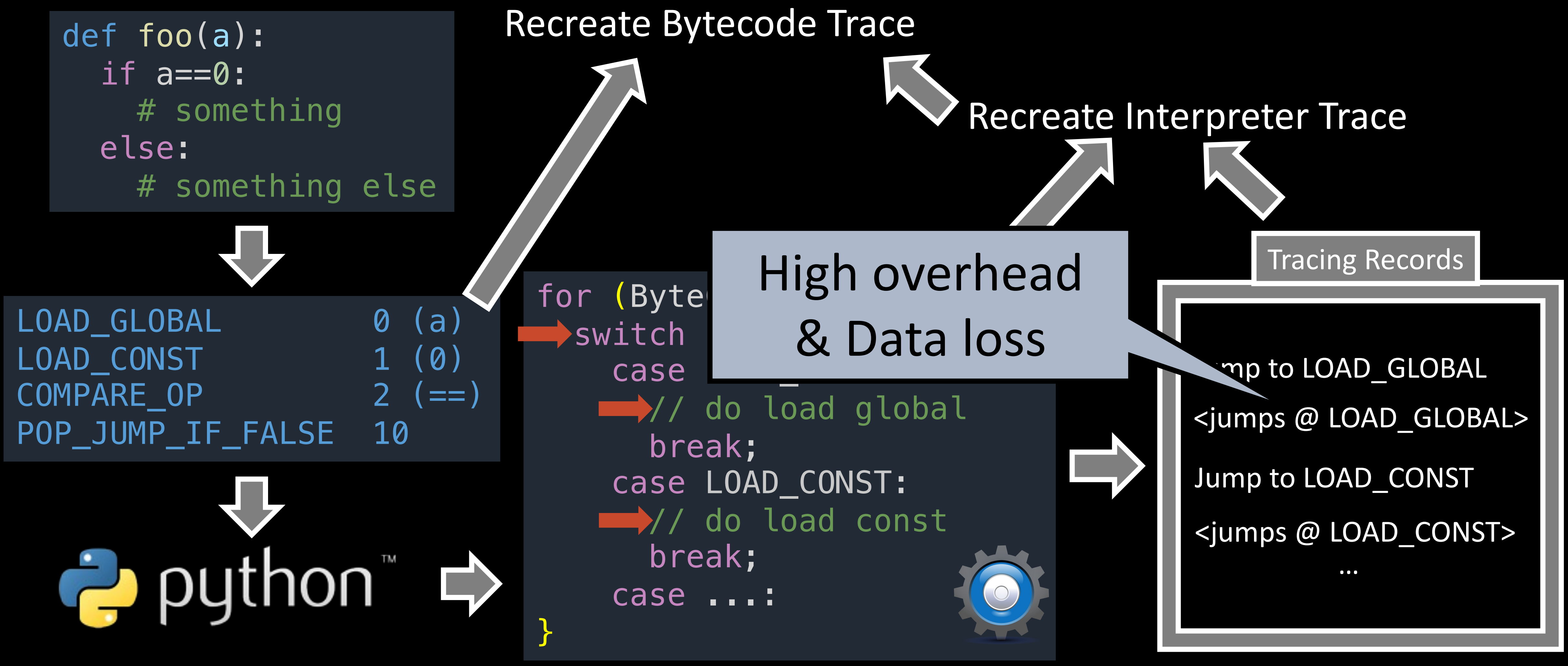
<jumps @ LOAD_GLOBAL>

Jump to LOAD_CONST

<jumps @ LOAD_CONST>

…

# Hardware Tracing for Interpreter Code



```
def foo(a):
  if a==0:
    # something
  else:
    # something else
```

```
LOAD_GLOBAL         0 (a)
LOAD_CONST          1 (0)
COMPARE_OP          2 (==)
POP_JUMP_IF_FALSE   10
```

```
for (Byte
  switch
    case
      // do load global
      break;
    case LOAD_CONST:
      // do load const
      break;
    case ...:
}
```

High overhead & Data loss

Tracing Records

mp to LOAD_GLOBAL
<jumps @ LOAD_GLOBAL>
Jump to LOAD_CONST
<jumps @ LOAD_CONST>
…

# Hardware Tracing for Interpreted Code

Only trace maybe-reachable changes

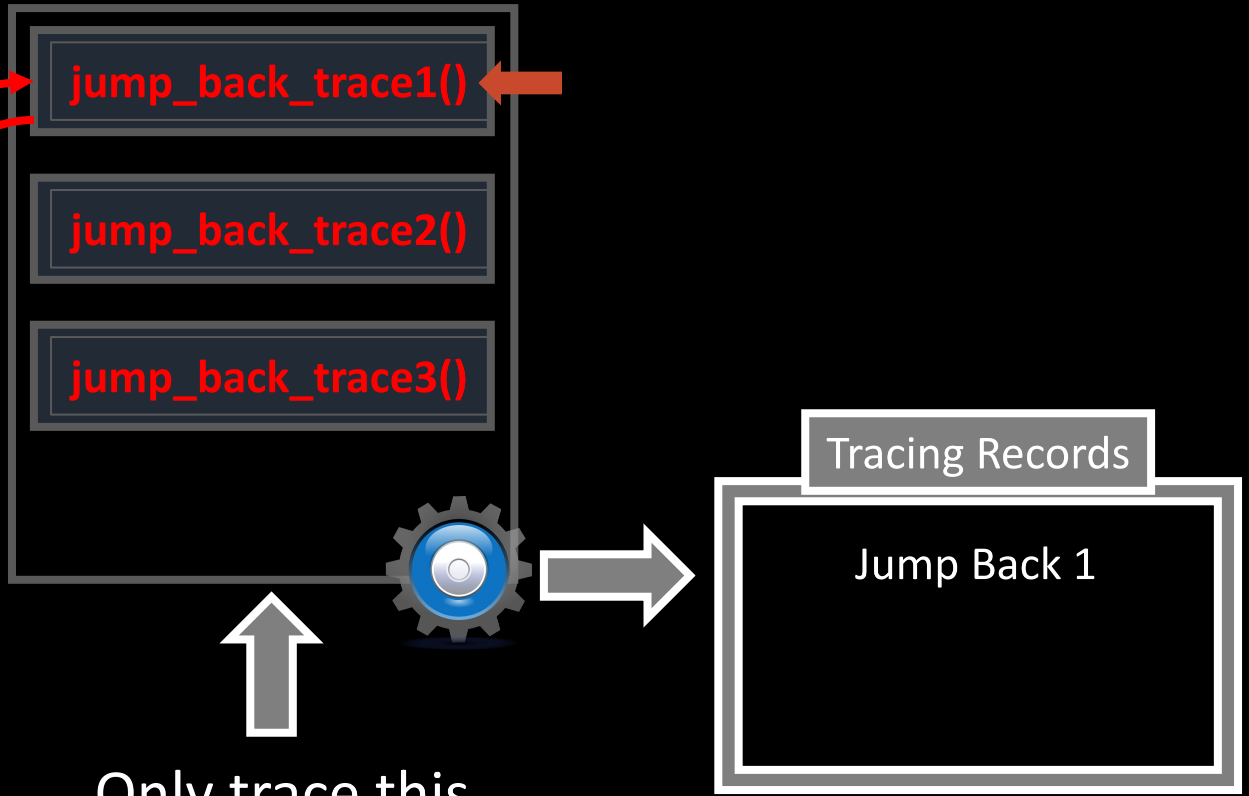
Only Selected Functions

code segment

# Hardware Tracing for Interpreted Code

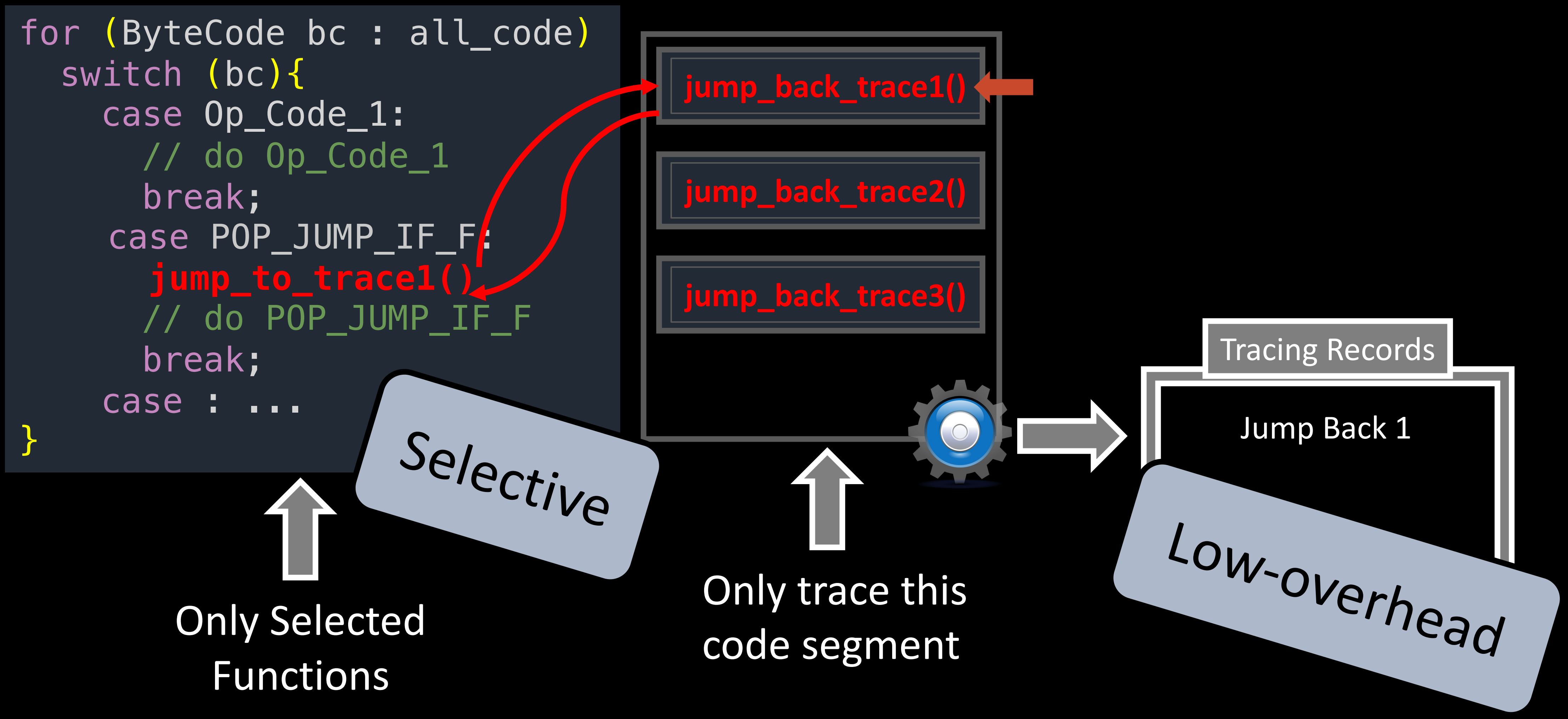
```
for (ByteCode bc : all_code)
  switch (bc){
    case Op_Code_1:
      // do Op_Code_1
      break;
     case POP_JUMP_IF_F:
      jump_to_trace1()
      // do POP_JUMP_IF_F
      break;
    case : ...
}
```

Only Selected Functions



Only trace this code segment

# Hardware Tracing for Interpreted Code

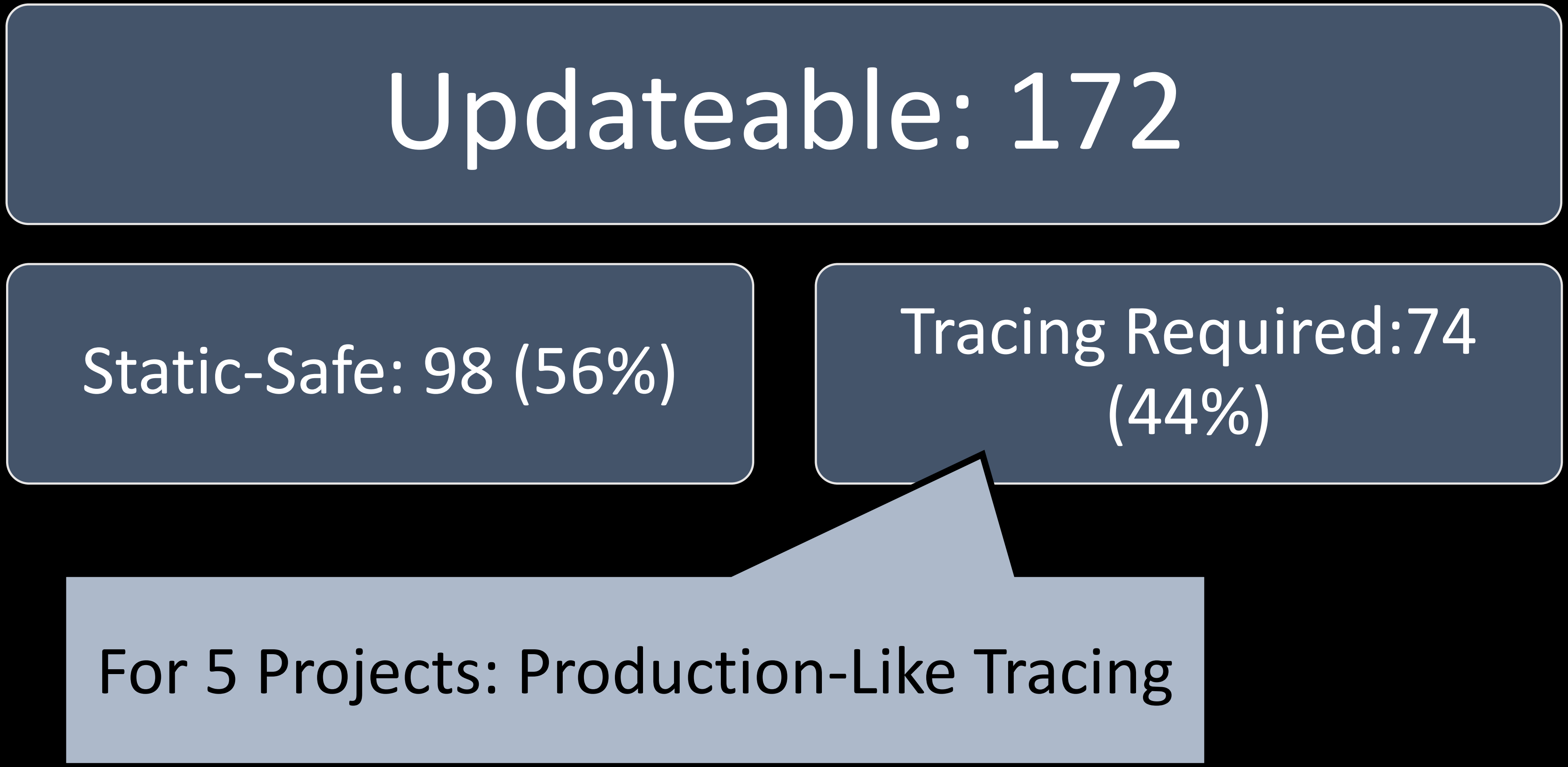


```
for (ByteCode bc : all_code)
  switch (bc){
    case Op_Code_1:
      // do Op_Code_1
      break;
     case POP_JUMP_IF_F:
      jump_to_trace1()
      // do POP_JUMP_IF_F
      break;
    case : ...
}
```

Only Selected Functions

Only trace this code segment

# Evaluation – Facilitating Dependency Updates



Updateable: 172

Static-Safe: 98 (56%)

Tracing Required:74 (44%)

For 5 Projects: Production-Like Tracing

# The Dynamic Tracing Contribution

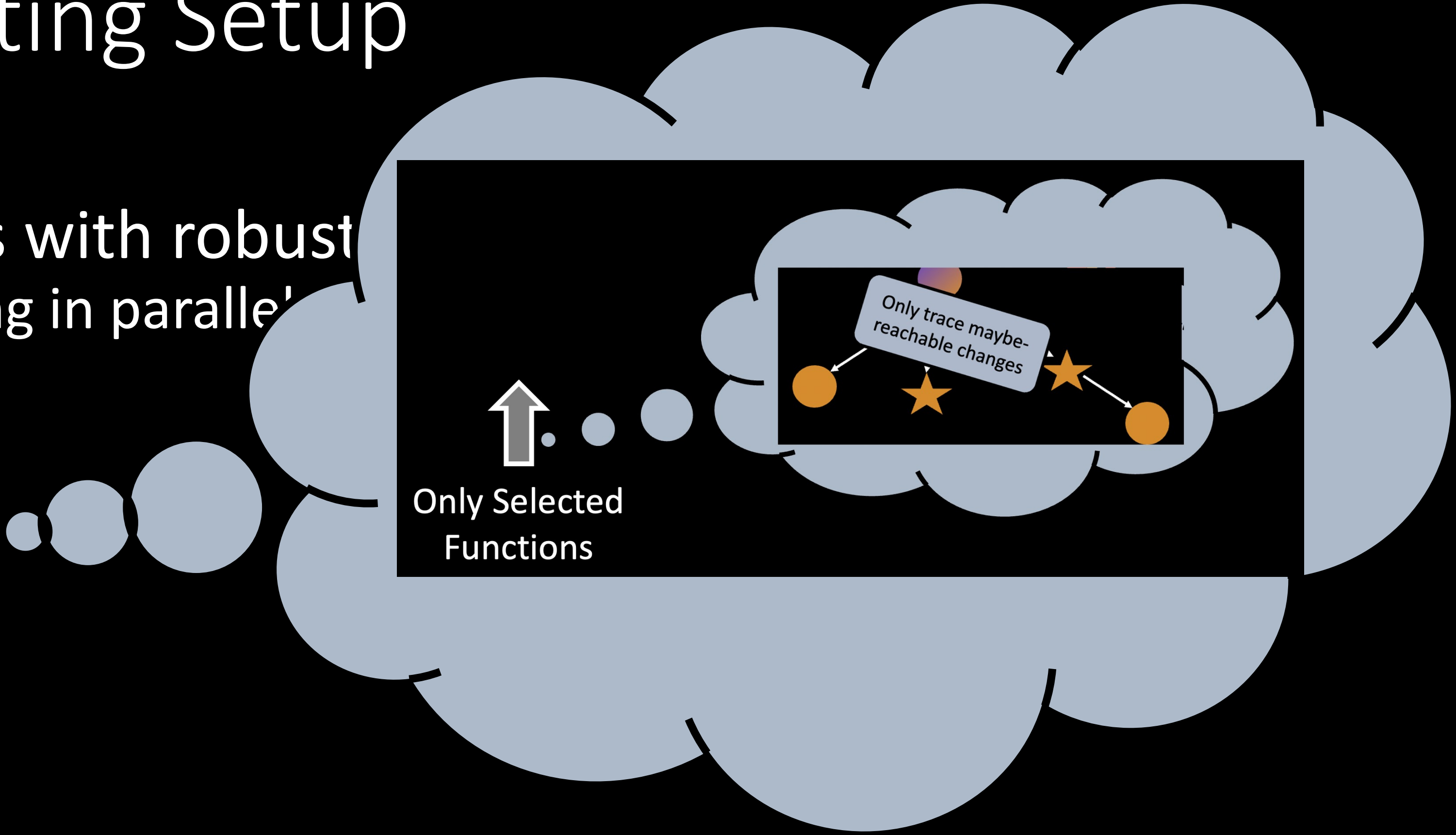
| Project (Dependency) | Diff (LOC) | % Discarded | | % Left |
|---|---|---|---|---|
| | | Static | Dynamic | |
| AutoML(Distributed) | 820 | 95 | 5 | 0 |
| Electrum (qdarkstyle) | 641 | 88 | 8 | 4 |
| Flair (gdown) | 1500 | 71 | 29 | 0 |
| Qlib (Hyperopt) | 828 | 90 | 9 | 1 |
| Scylla (requests) | 449 | 90 | 8 | 2 |

# Tracer Overhead Testing Setup

- Selected Python projects with robust test-suites from our data-set
  - For Django's also running in parallel: using 1, 8, and 16 cores

# Tracer Overhead Testing Setup

- Selected Python projects with robust
  - For Django's also running in paralle'
- UPGRADVISOR-Targeted



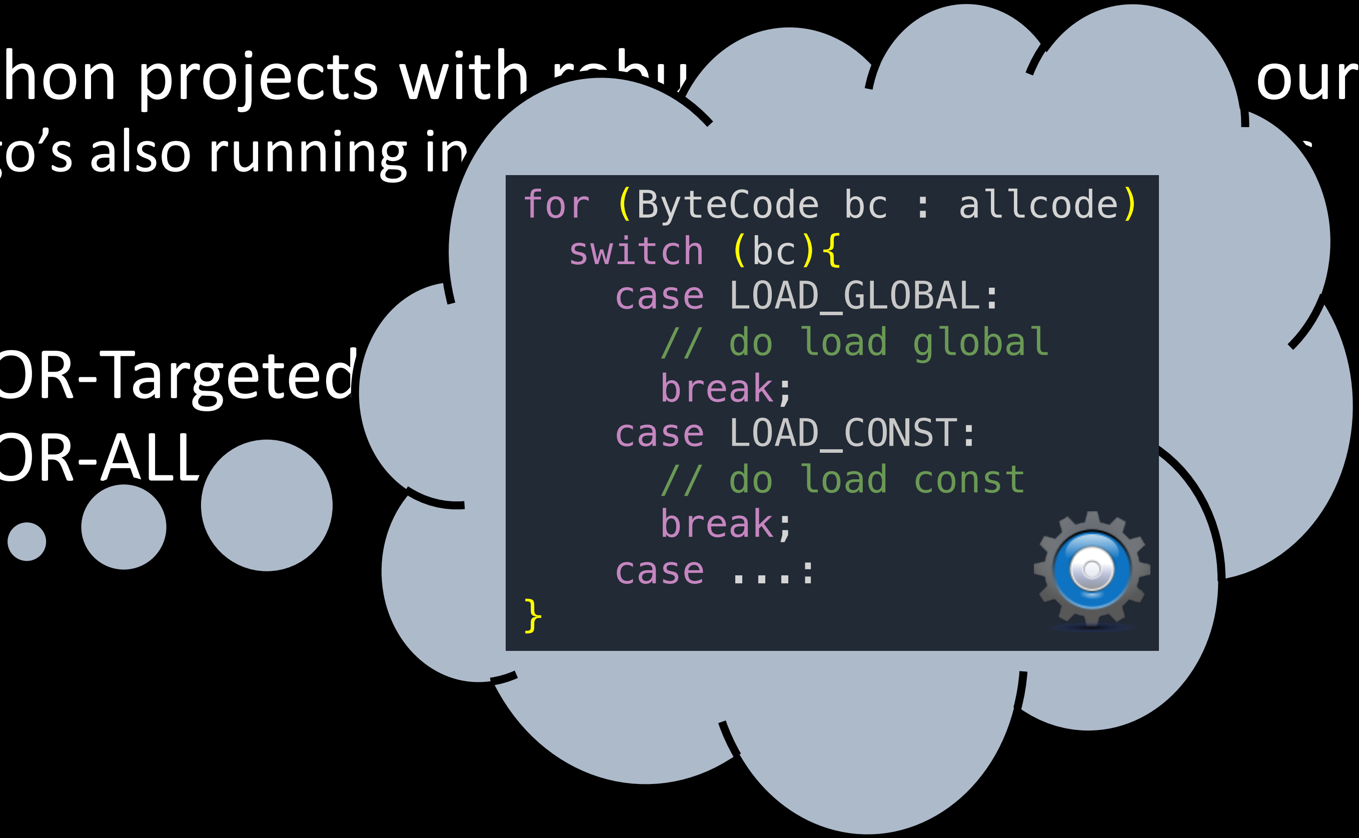
Only Selected Functions

Only trace maybe-reachable changes

# Tracer Overhead Testing Setup

- Selected Python projects with robu our data-set
  - For Django’s also running in

- UPGRADVISOR-Targeted
- UPGRADVISOR-ALL
- Jportal4Py



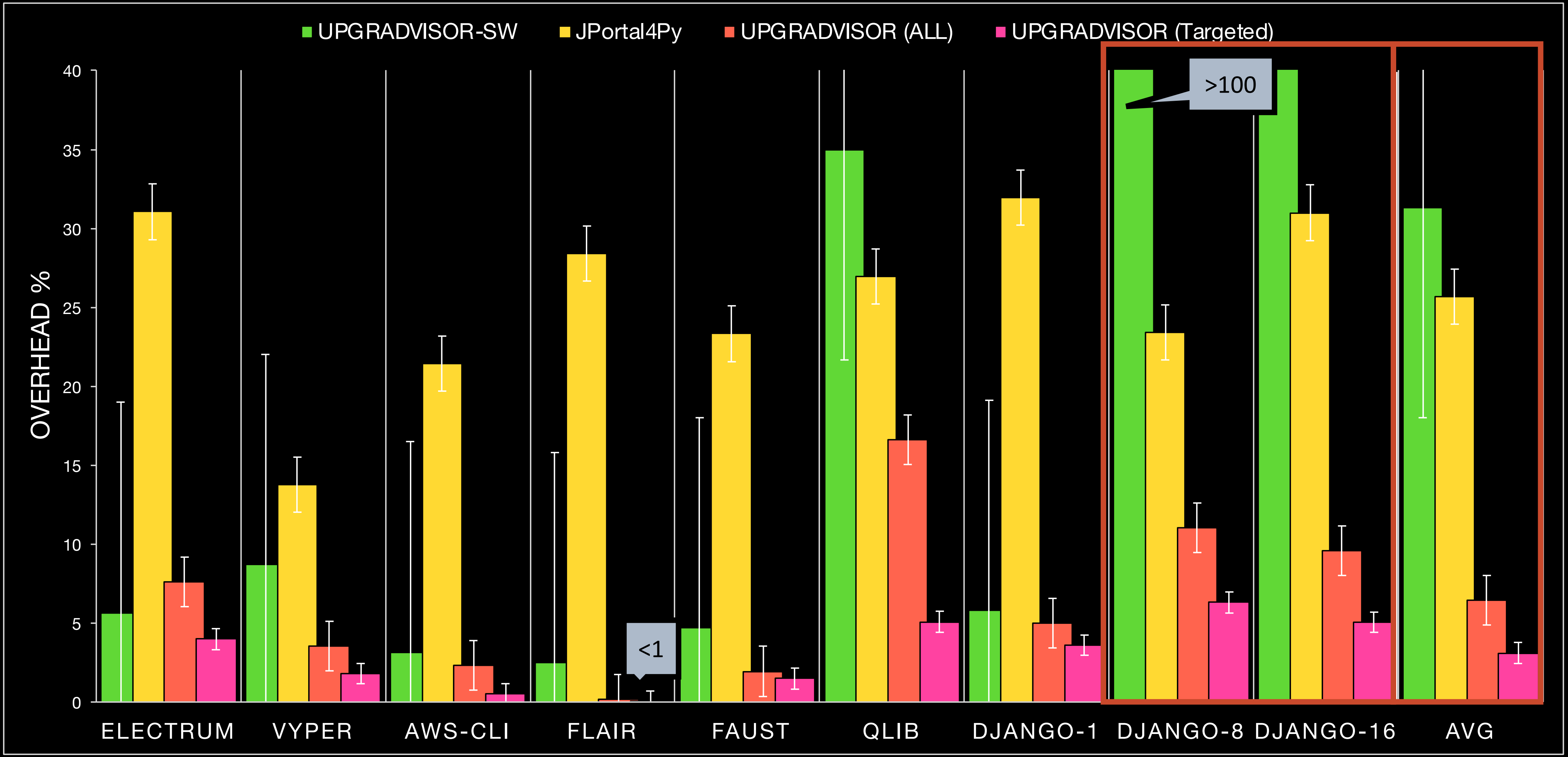
```
for (ByteCode bc : allcode)
  switch (bc){
    case LOAD_GLOBAL:
      // do load global
      break;
    case LOAD_CONST:
      // do load const
      break;
    case ...:
}
```

# Tracer Overhead Testing Setup

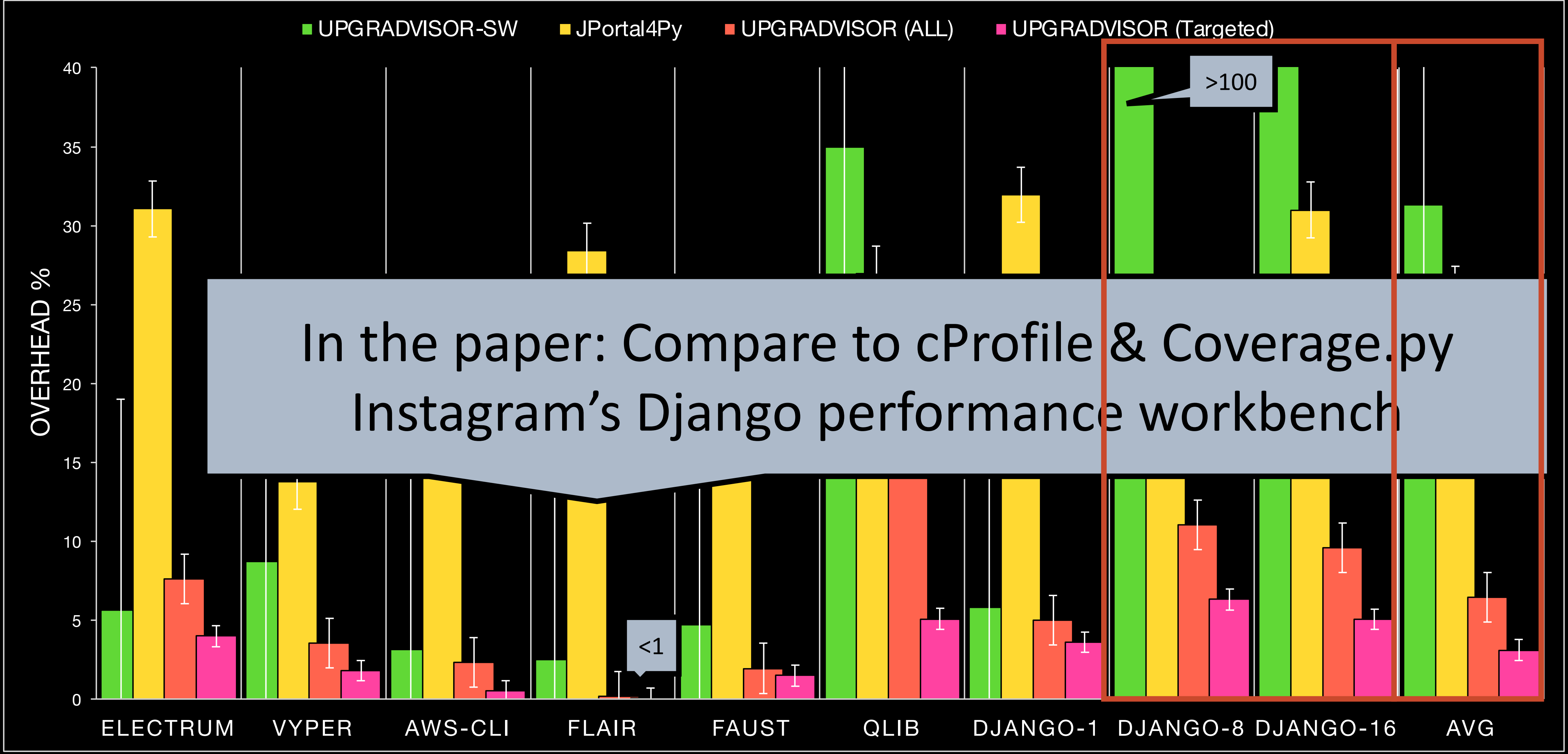
- Selected Python projects with robust test-suites from our data-set
  - For Django's also running in parallel: using 1, 8, and 16 cores

- UPGRADVISOR-Targeted
- UPGRADVISOR-ALL
- Jportal4Py
- UPGRADVISOR-SW

# UPGRADVISOR’s Tracer Incurs Low-Overhead



# UPGRADVISOR’s Tracer Incurs Low-Overhead



In the paper: Compare to cProfile & Coverage.py
Instagram’s Django performance workbench

# Conclusion

- We presented UPGRADVISOR: a system for reducing developer effort and error risk in adopting dependency updates

- Want to know more? See our website!

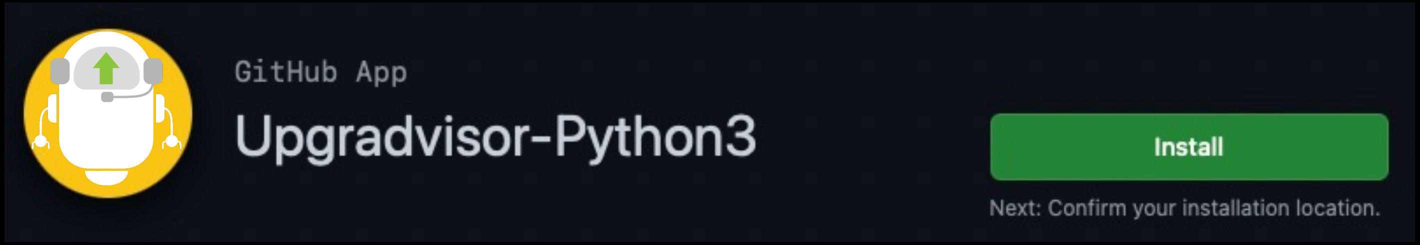
  [https://upgradvisor.github.io](https://upgradvisor.github.io)

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

- Want to use UPGRADVISOR-Python3? Install our free GitHub App



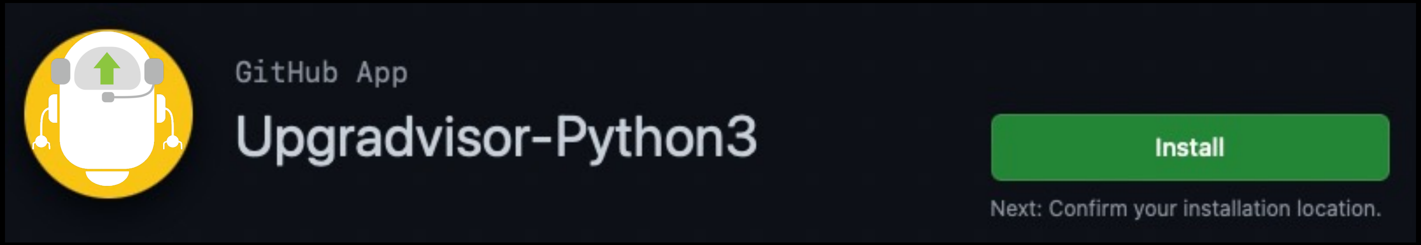
GitHub App

Upgradvisor-Python3

Next: Confirm your installation location.

# Conclusion

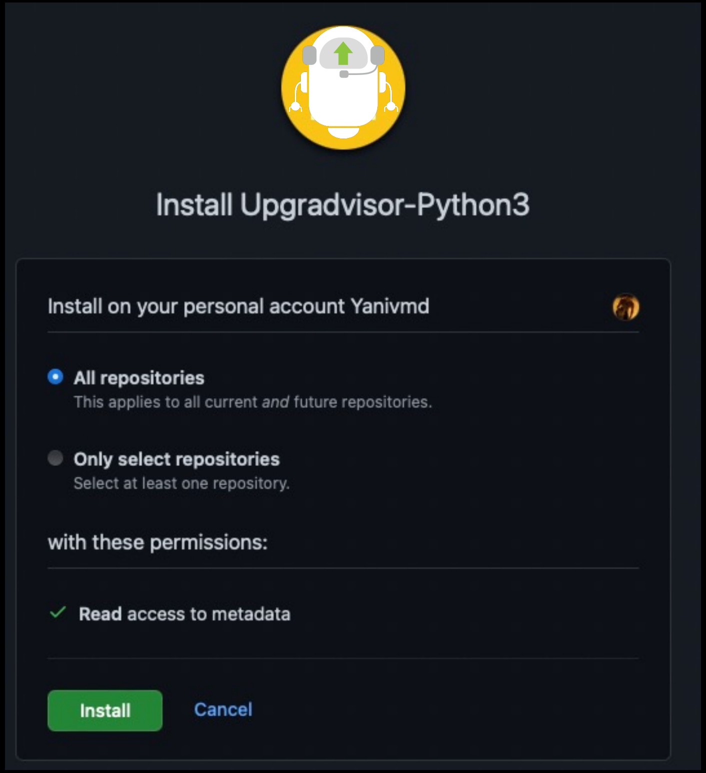
- Want to use UPGRADVISOR? Install our free GitHub App



GitHub App

Upgradvisor-Python3

Next: Confirm your installation location.



Install Upgradvisor-Python3

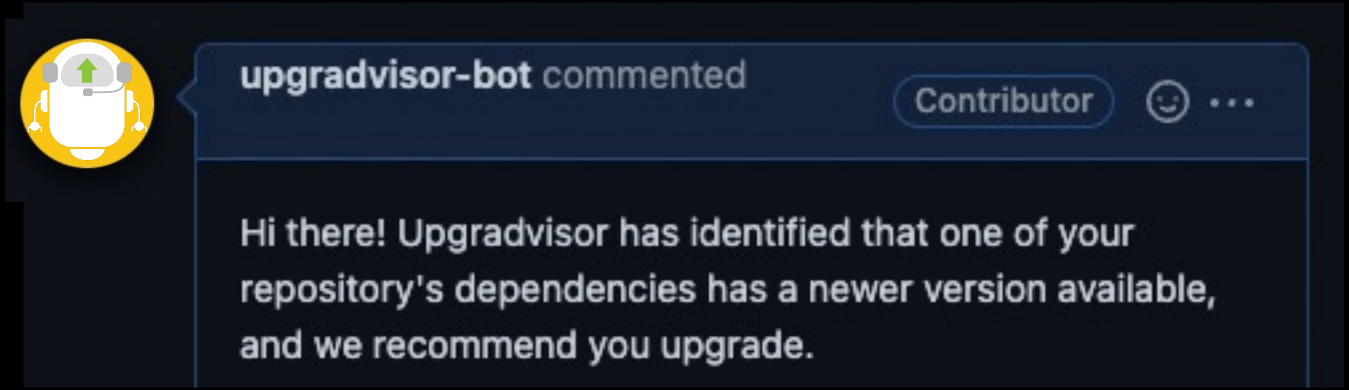
Install on your personal account Yanivmd

● **All repositories**
This applies to all current *and* future repositories.

○ **Only select repositories**
Select at least one repository.

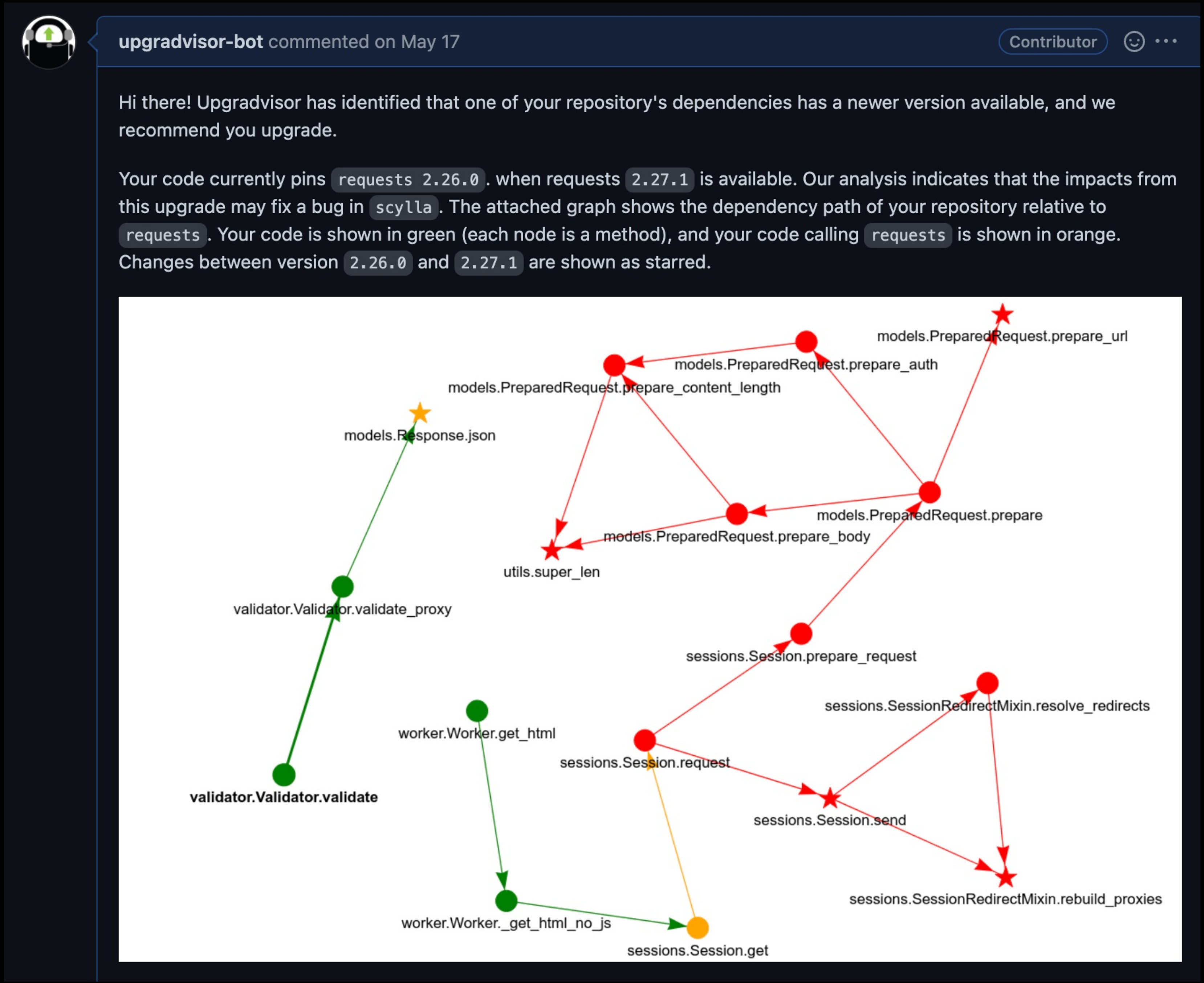
with these permissions:

✓ Read access to metadata



**upgradvisor-bot** commented

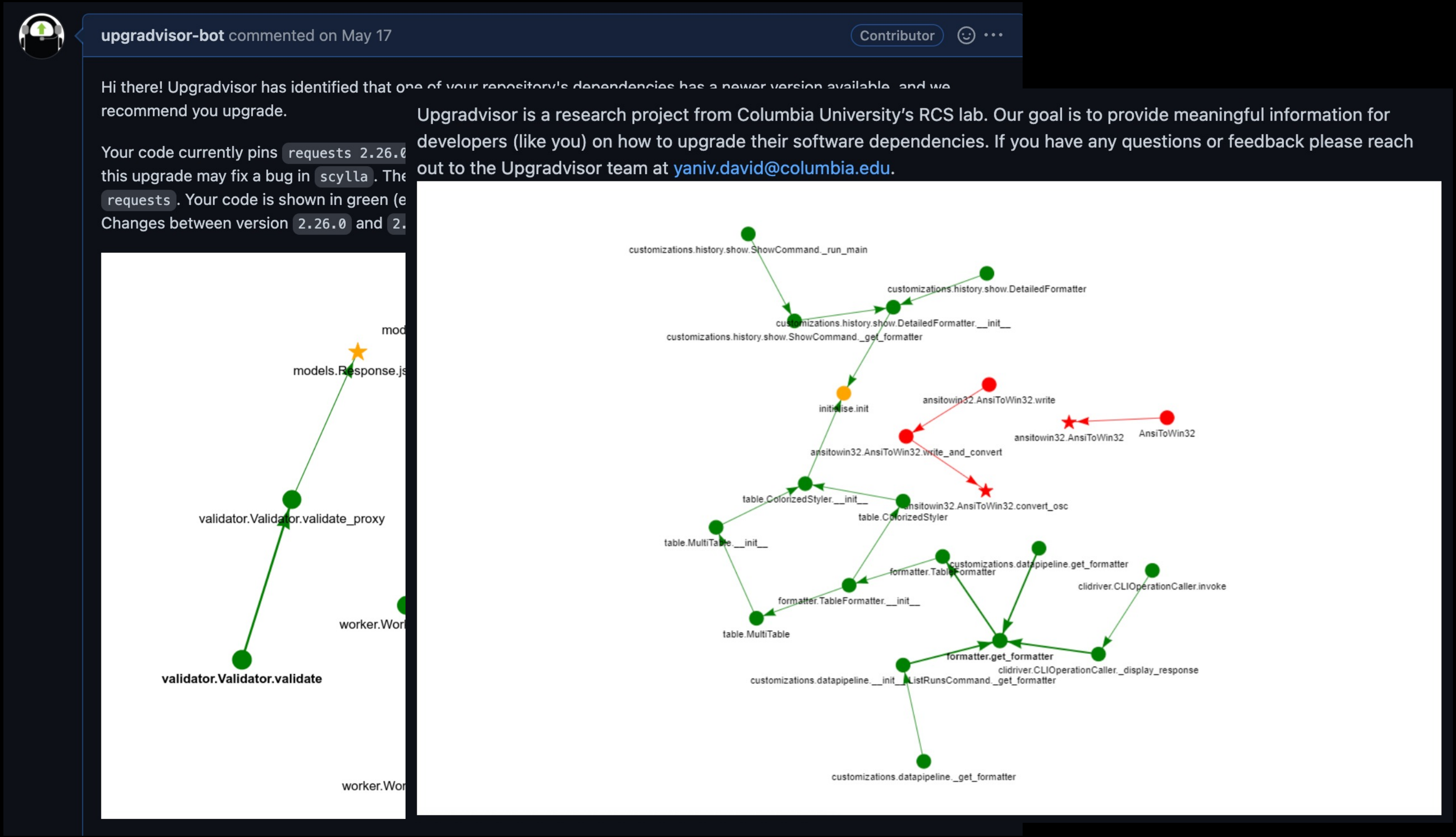
Hi there! Upgradvisor has identified that one of your repository's dependencies has a newer version available, and we recommend you upgrade.



Hi there! Upgradvisor has identified that one of your repository's dependencies has a newer version available, and we recommend you upgrade.

Your code currently pins `requests 2.26.0`. when requests `2.27.1` is available. Our analysis indicates that the impacts from this upgrade may fix a bug in `scylla`. The attached graph shows the dependency path of your repository relative to `requests`. Your code is shown in green (each node is a method), and your code calling `requests` is shown in orange. Changes between version `2.26.0` and `2.27.1` are shown as starred.

models.PreparedRequest.prepare_url
models.PreparedRequest.prepare_auth
models.PreparedRequest.prepare_content_length
models.Response.json
models.PreparedRequest.prepare
models.PreparedRequest.prepare_body
utils.super_len
validator.Validator.validate_proxy
sessions.Session.prepare_request
sessions.SessionRedirectMixin.resolve_redirects
worker.Worker.get_html
sessions.Session.request
validator.Validator.validate
sessions.Session.send
sessions.SessionRedirectMixin.rebuild_proxies
worker.Worker._get_html_no_js
sessions.Session.get



upgradvisor-bot commented on May 17

Hi there! Upgradvisor has identified that one of your repository's dependencies has a newer version available, and we recommend you upgrade.

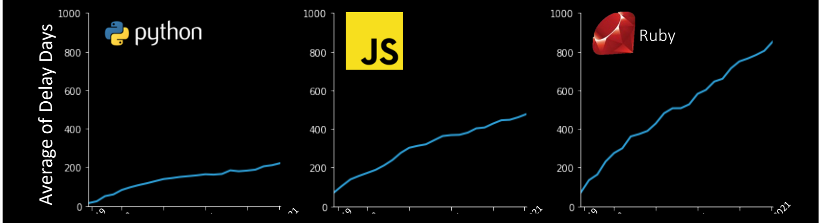
Your code currently pins `requests 2.26.0`
this upgrade may fix a bug in `scylla`. The
`requests`. Your code is shown in green (e
Changes between version `2.26.0` and `2.`

mod
models.Response.js
validator.Validator.validate_proxy
worker.Wor
validator.Validator.validate
worker.Wor

Upgradvisor is a research project from Columbia University's RCS lab. Our goal is to provide meaningful information for developers (like you) on how to upgrade their software dependencies. If you have any questions or feedback please reach out to the Upgradvisor team at yaniv.david@columbia.edu.

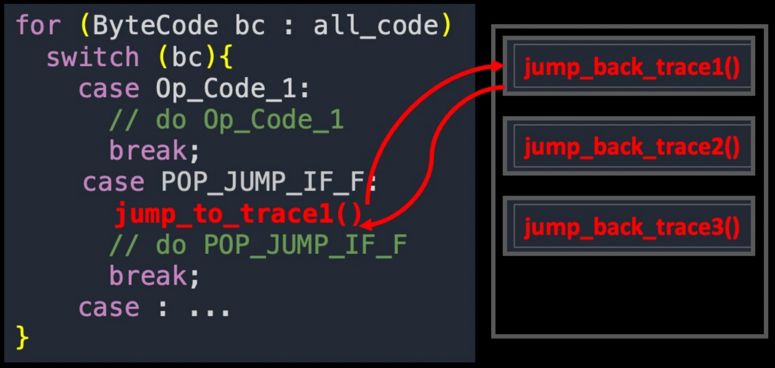
customizations.history.show.ShowCommand._run_main
customizations.history.show.DetailedFormatter
customizations.history.show.DetailedFormatter.__init__
customizations.history.show.ShowCommand._get_formatter
initialise.init
ansitowin32.AnsiToWin32.write
ansitowin32.AnsiToWin32
AnsiToWin32
ansitowin32.AnsiToWin32.write_and_convert
table.ColorizedStyler.__init__
ansitowin32.AnsiToWin32.convert_osc
table.ColorizedStyler
table.MultiTable.__init__
customizations.datapipeline.get_formatter
formatter.TableFormatter
clidriver.CLIOperationCaller.invoke
formatter.TableFormatter.__init__
table.MultiTable
formatter.get_formatter
clidriver.CLIOperationCaller._display_response
customizations.datapipeline.__init__.ListRunsCommand._get_formatter
customizations.datapipeline._get_formatter

## Dependency Update Adoption Is Slow



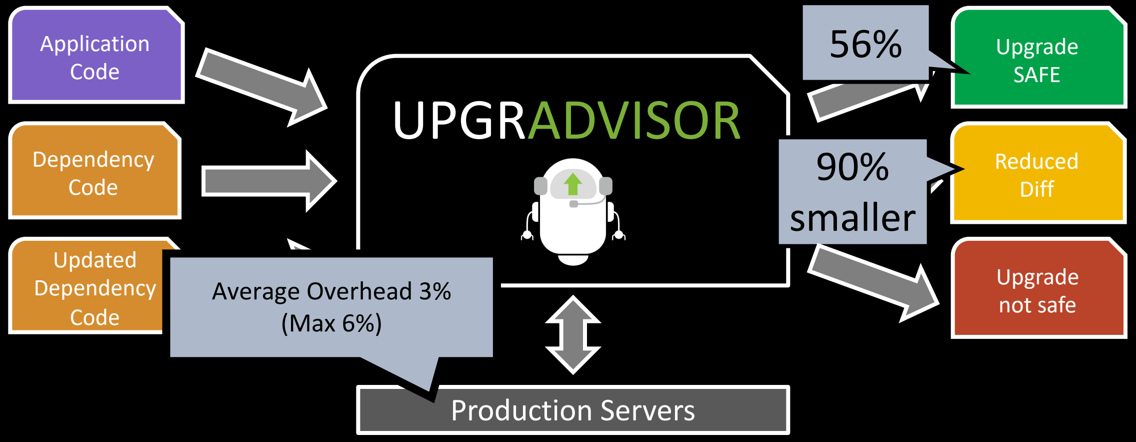
# Thank You!
# Questions?

## HW-based Tracing is Production-Ready



```
for (ByteCode bc : all_code)
  switch (bc){
    case Op_Code_1:
      // do Op_Code_1
      break;
    case POP_JUMP_IF_F:
      jump_to_trace1()
      // do POP_JUMP_IF_F
      break;
    case : ...
}
```

## UPGRADVISOR: a system for reducing developer effort and error risk in adopting dependency updates



Average Overhead 3% (Max 6%)