Trellis: Privilege Separation for Multi-User Applications Made Easy

Andrea Mambretti

Kaan Onarlioglu, Collin Mulliner, William Robertson, Engin Kirda
Northeastern University

Federico Maggi, Stefano Zanero
Politecnico di Milano

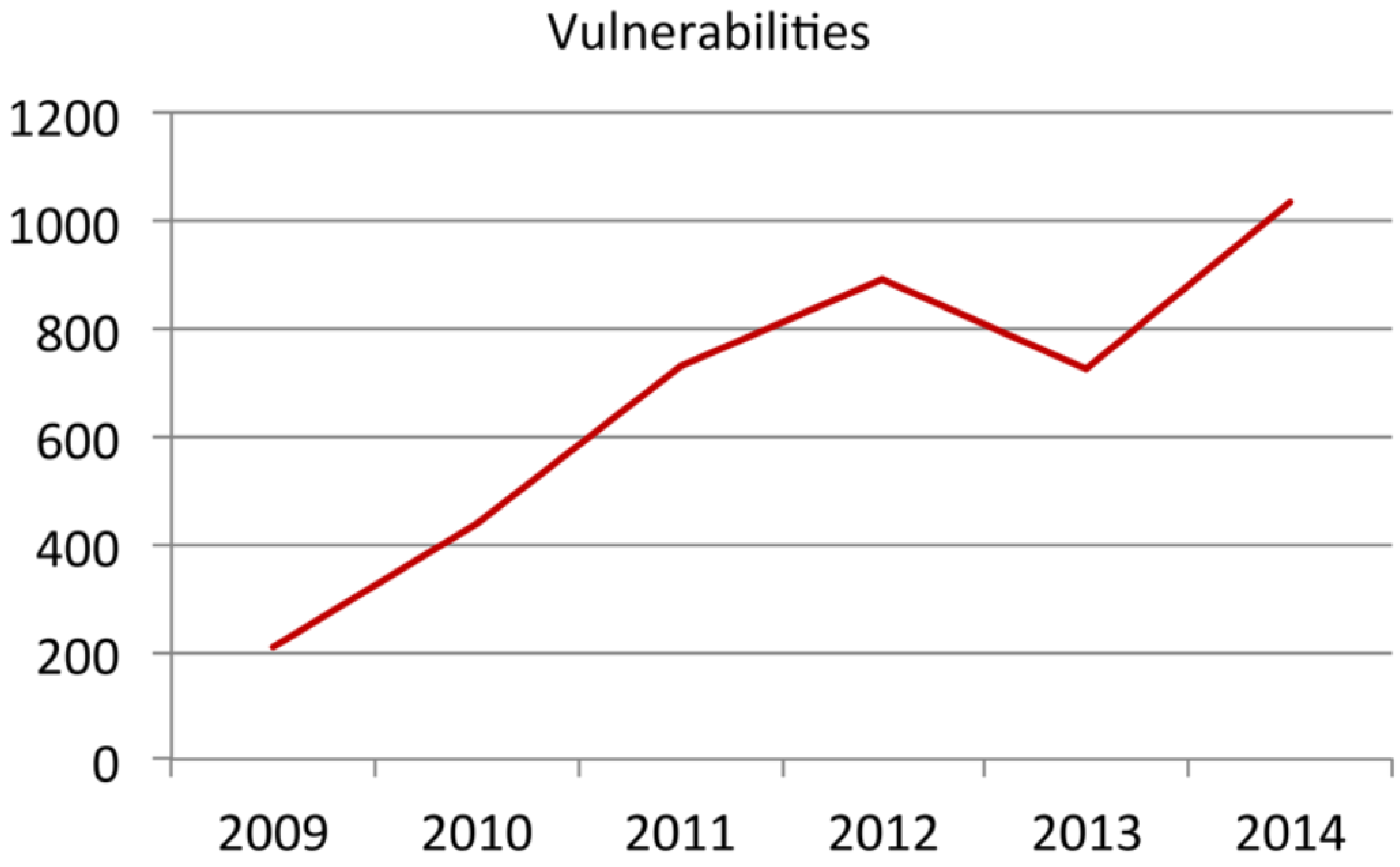
Trellis: Privilege Separation for Multi-User Applications Made Easy

Andrea Mambretti

Kaan Onarlioglu, Collin Mulliner, William Robertson, Engin Kirda
Northeastern University

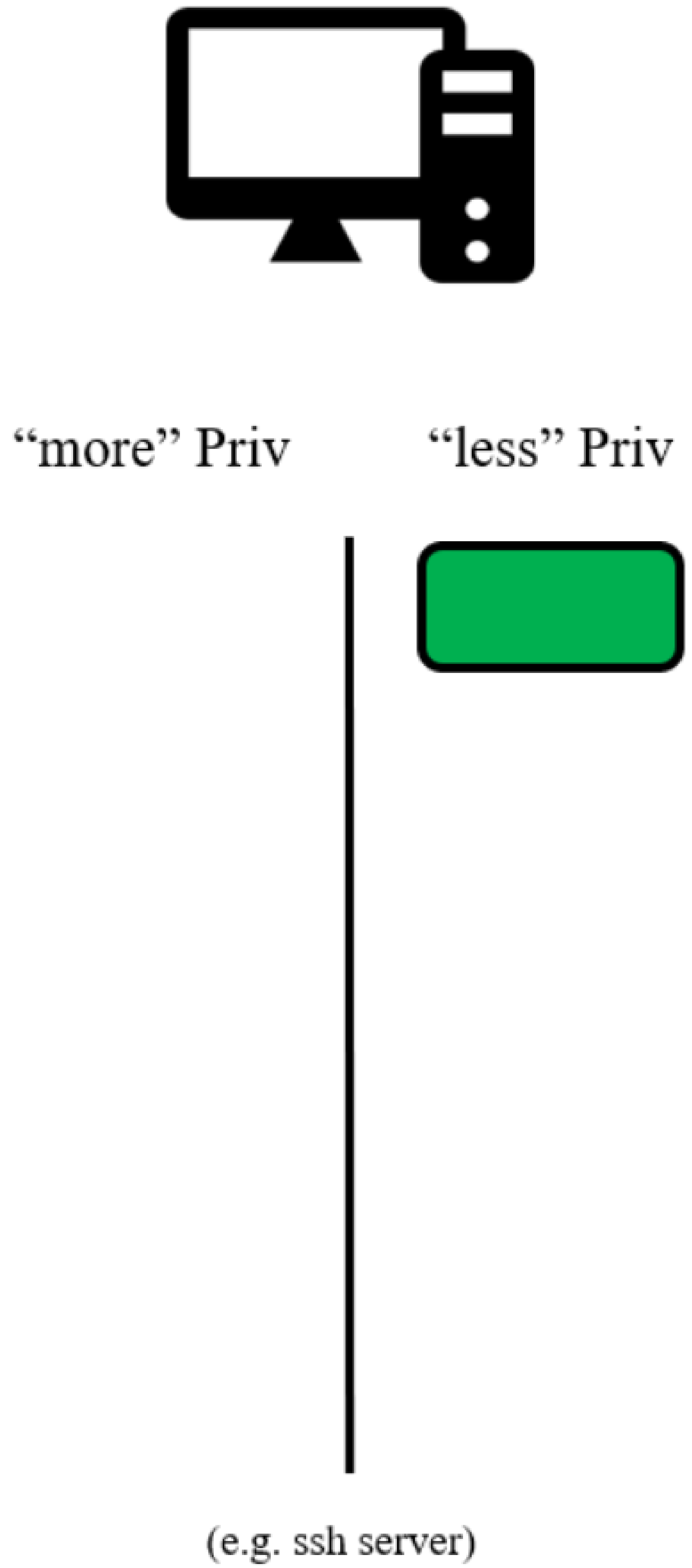
Federico Maggi, Stefano Zanero
Politecnico di Milano

Unsolved Problem?

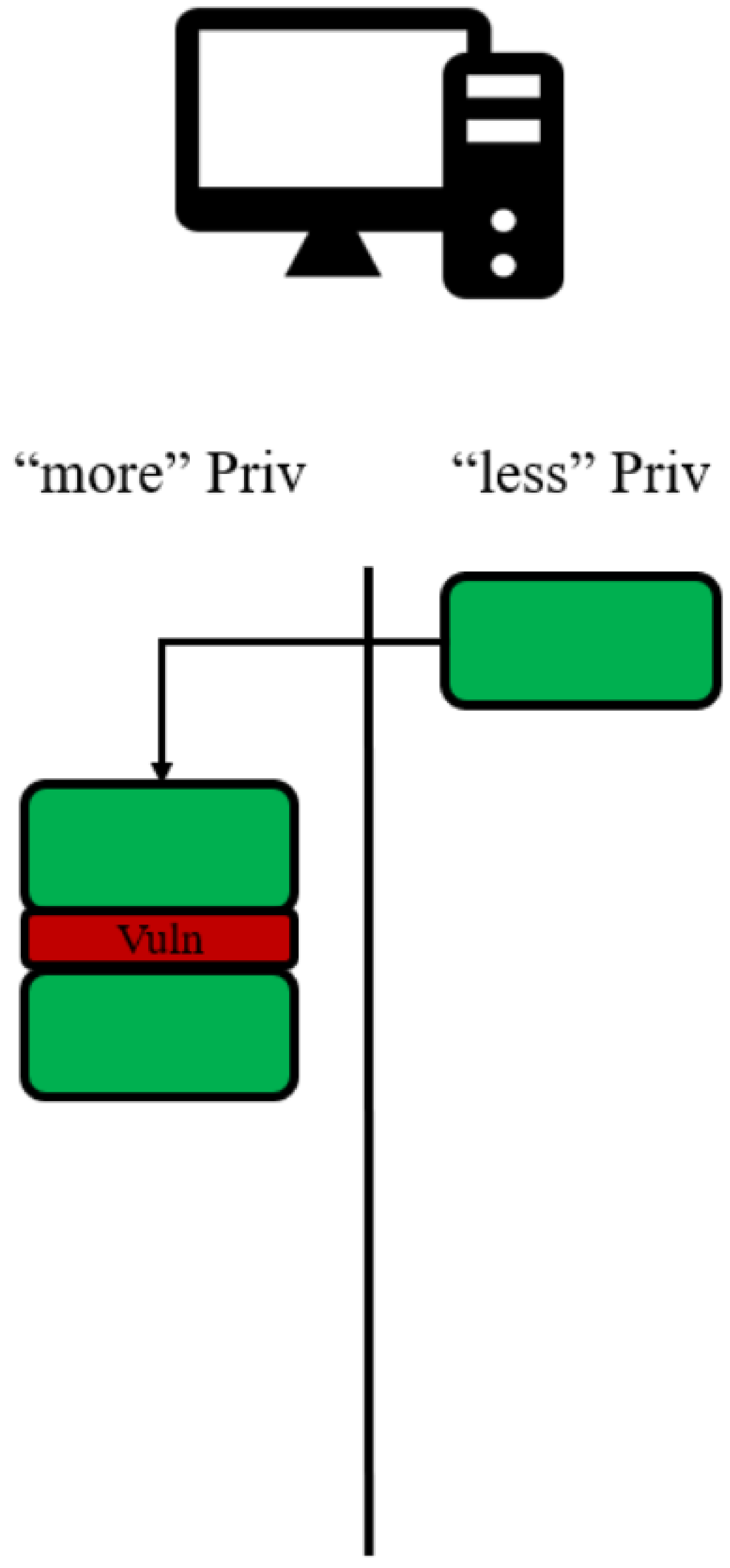


Secunia review 2015 – Vulnerabilities in the 5 Most Popular Browser

Privilege Escalation

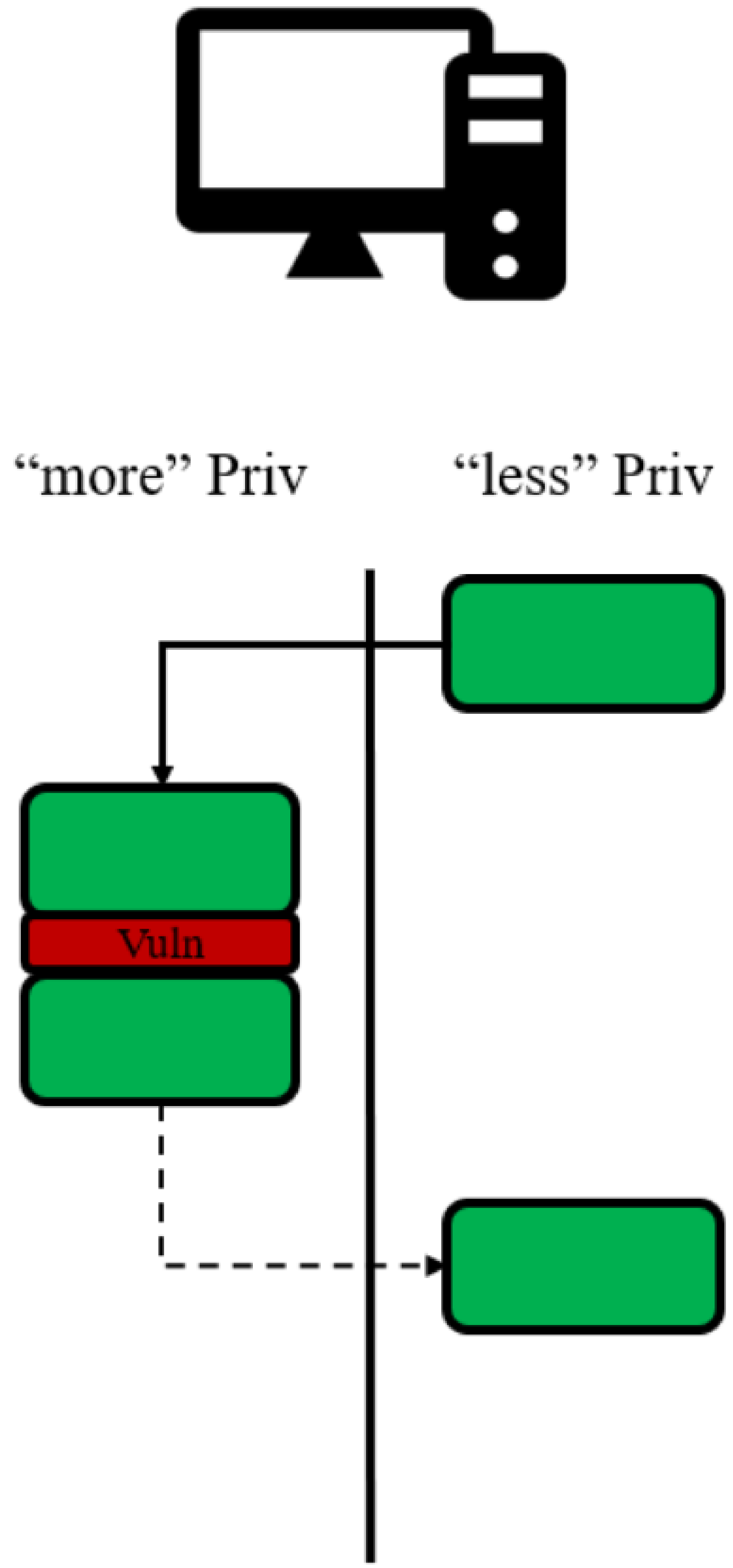


Privilege Escalation



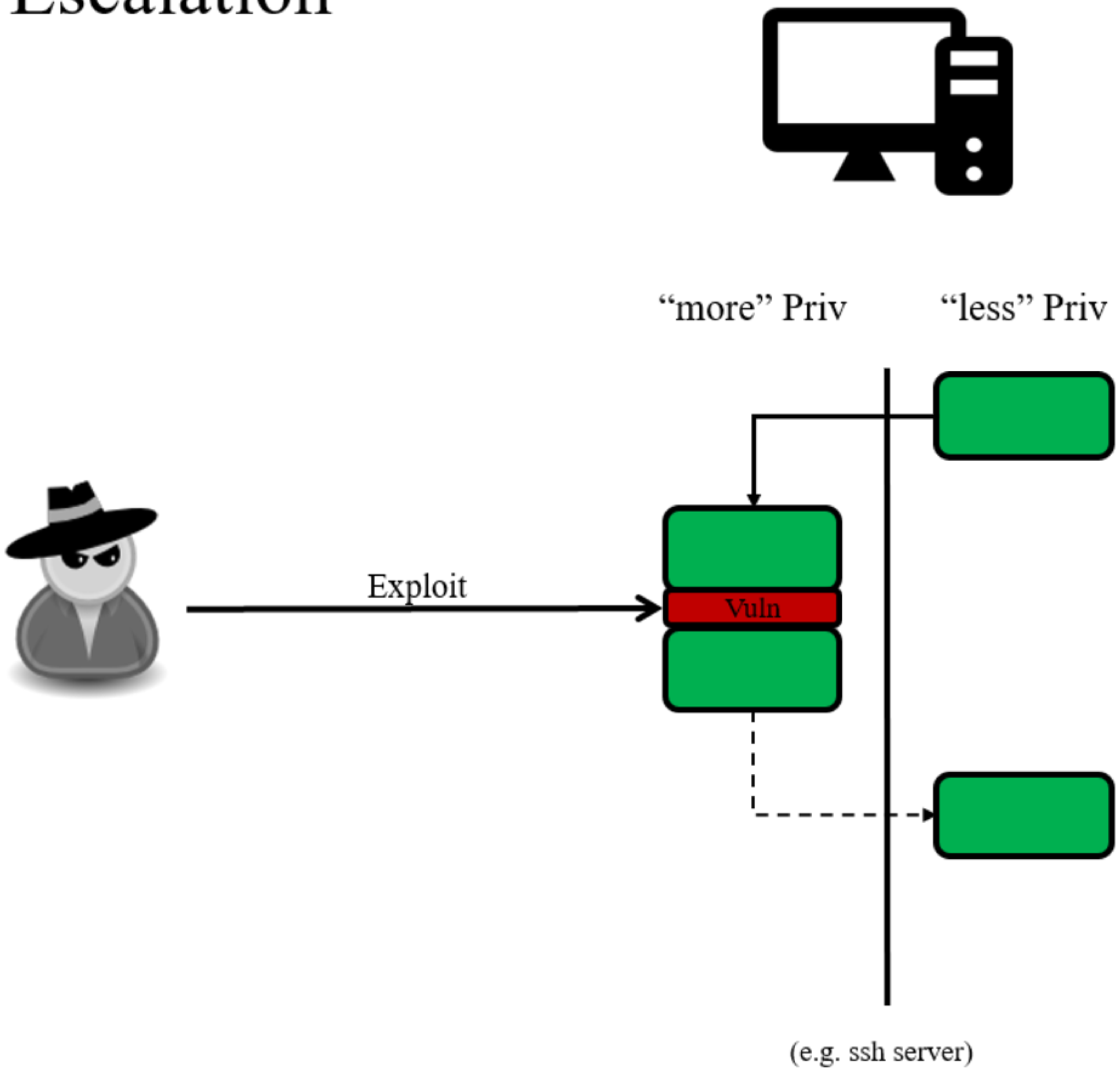
(e.g. ssh server)

Privilege Escalation

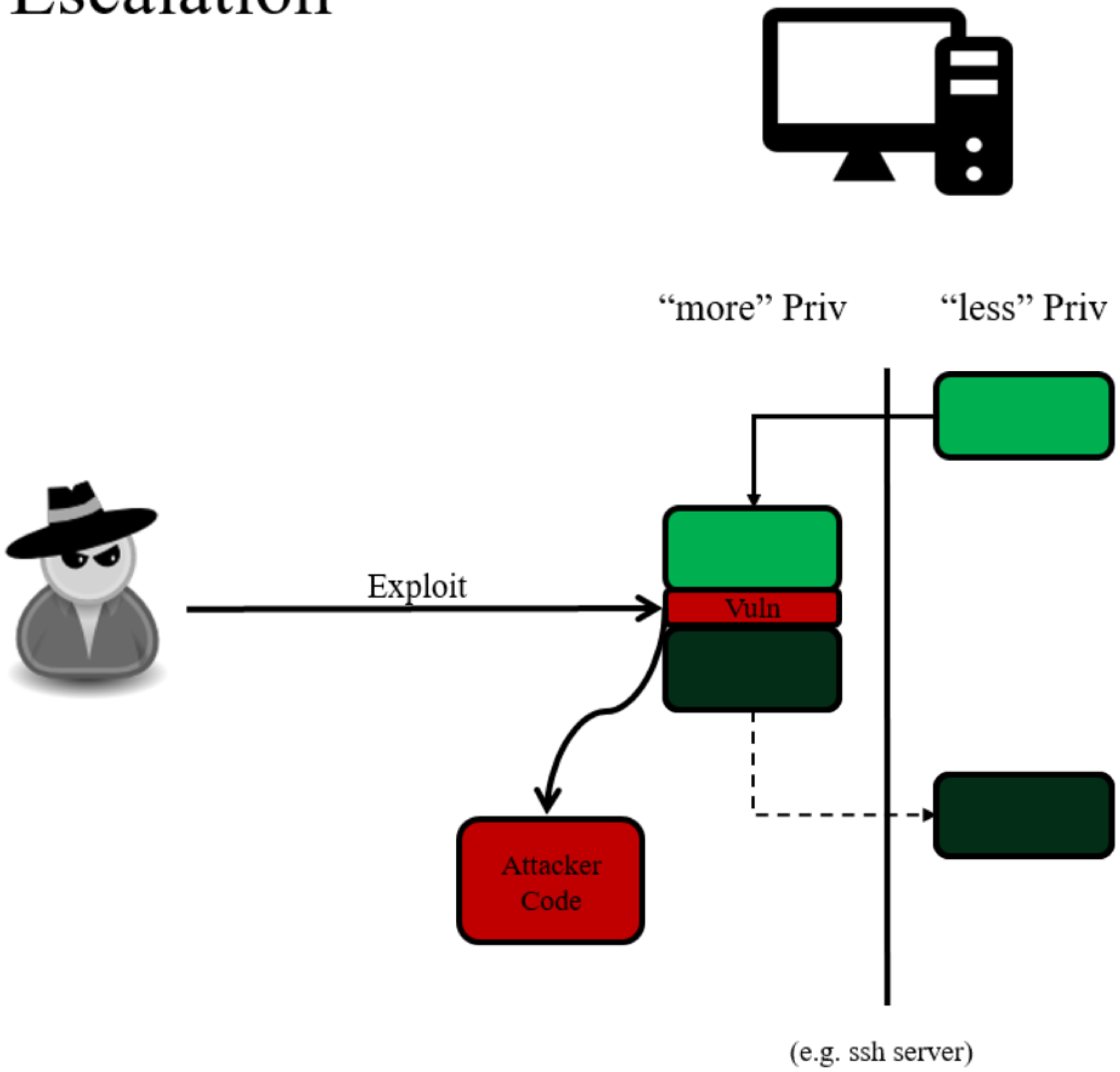


(e.g. ssh server)

Privilege Escalation



Privilege Escalation



Current Mitigations

GEM [Mulliner(2014)]

GEM [Mulliner(2014)]

Multi-User Applications?

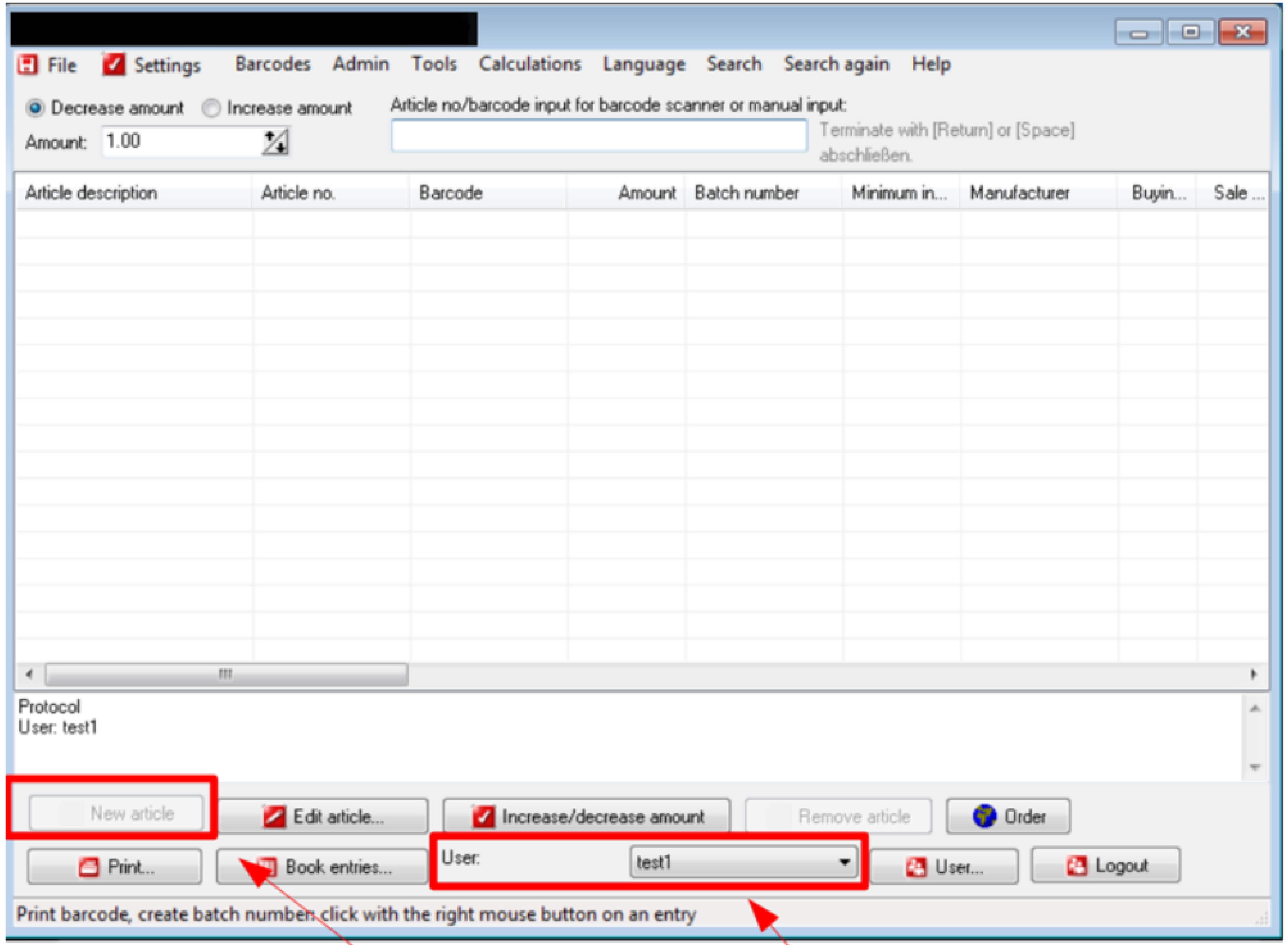
GEM [Mulliner(2014)]

Multi-User Applications?

GEM [Mulliner(2014)]

Multi-User Applications?

GEM [Mulliner(2014)]



[redacted]

● Decrease amount ○ Increase amount

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:

Terminate with [Return] or [Space] abschließen.

Article description	Article no.	Barcode	Amount	Batch number	Minimum in...	Manufacturer	Buyin...	Sale ...

Protocol
User: test1

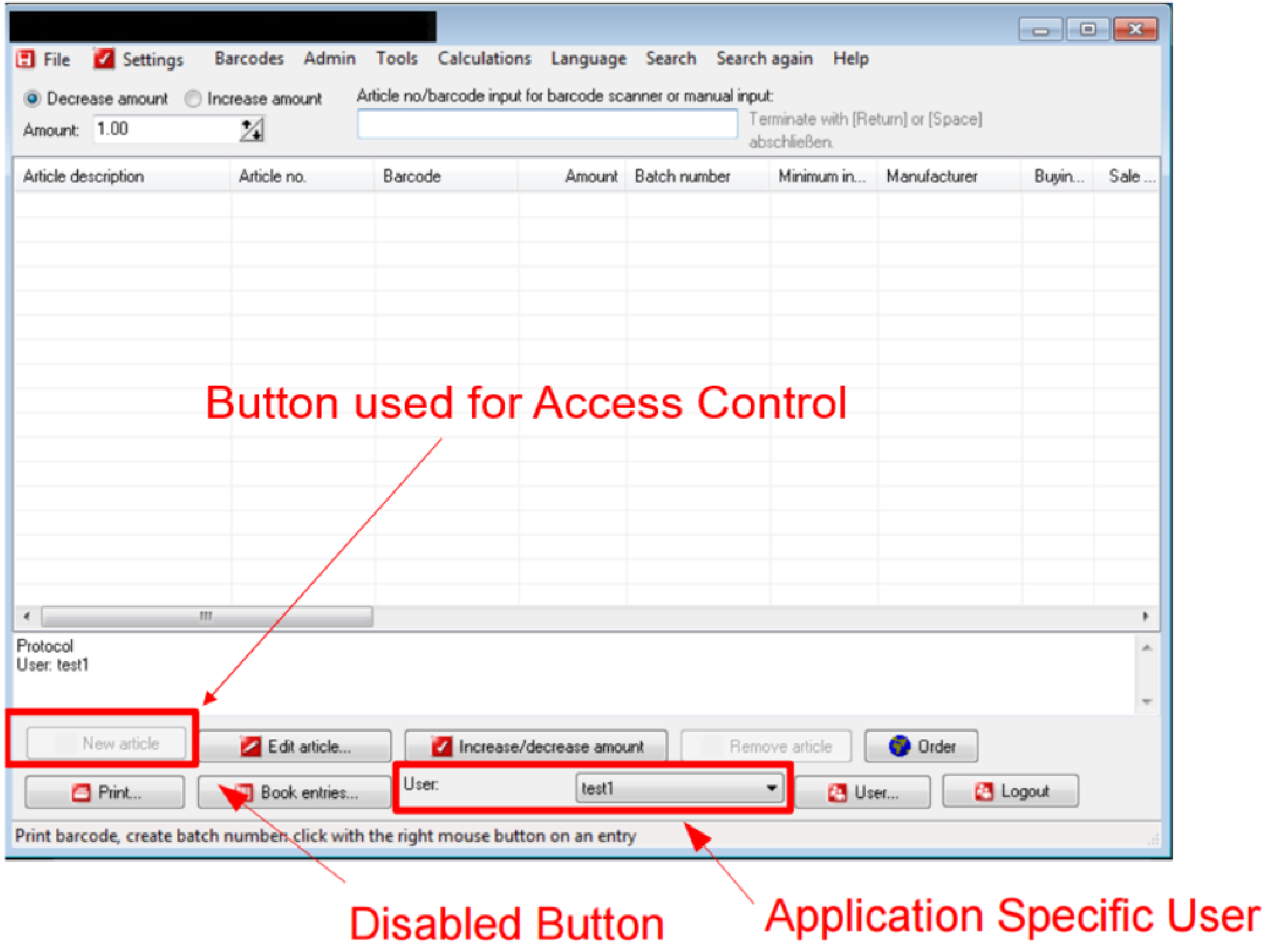
User: test1

Print barcode, create batch number: click with the right mouse button on an entry

Disabled Button

Application Specific User

GEM [Mulliner(2014)]


[redacted]

● Decrease amount ○ Increase amount

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:

Terminate with [Return] or [Space] abschließen.

Protocol
User: test1

User: test1

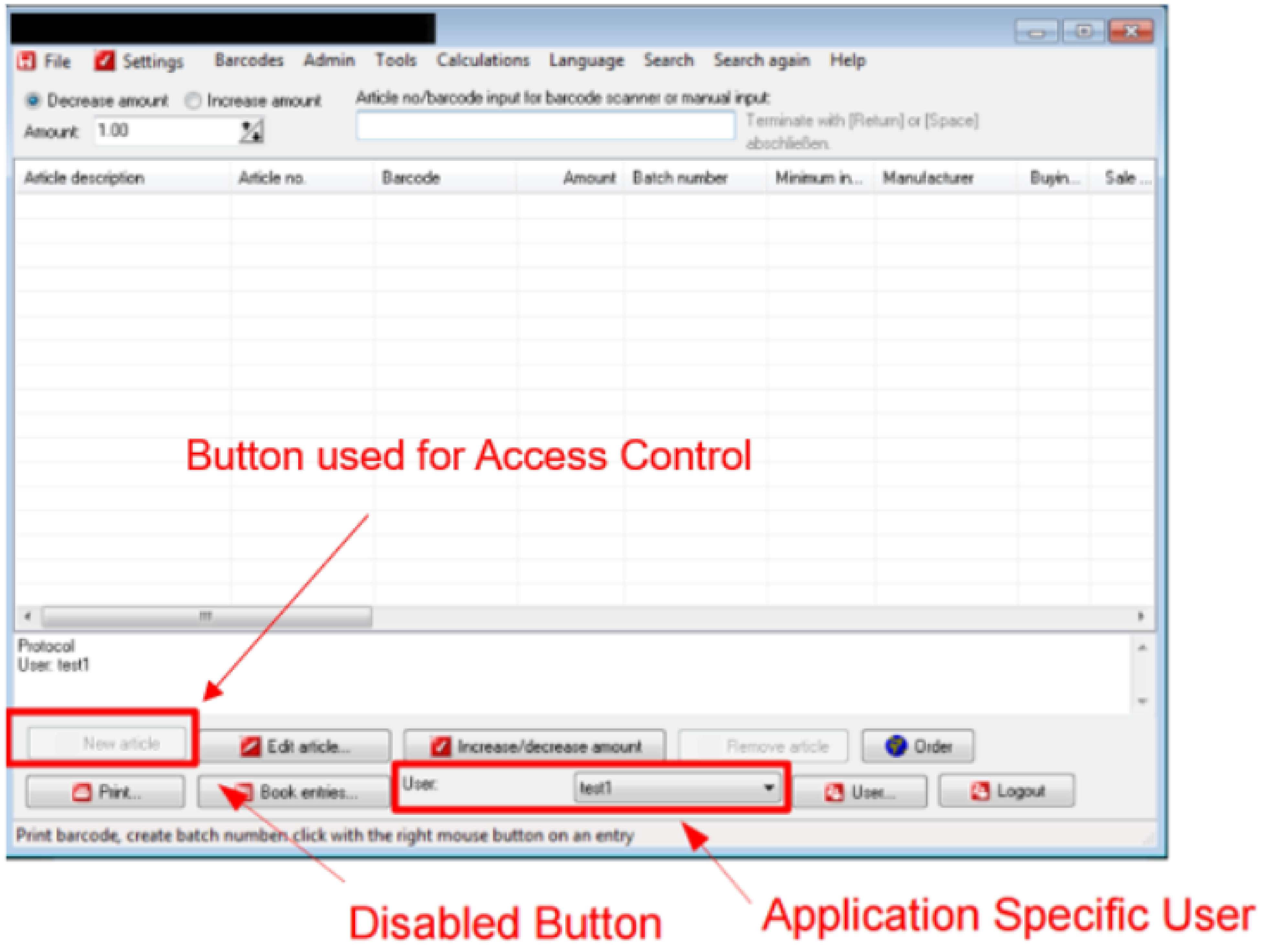
Print barcode, create batch number: click with the right mouse button on an entry

Button used for Access Control

Disabled Button

Application Specific User

GEM [Mulliner(2014)]



[redacted]

● Decrease amount ○ Increase amount

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:

Terminate with [Return] or [Space] abschließen.

Article description	Article no.	Barcode	Amount	Batch number	Minimum in...	Manufacturer	Buyin...	Sale ...

Protocol
User: test1

User: test1

Print barcode, create batch number, click with the right mouse button on an entry

Button used for Access Control

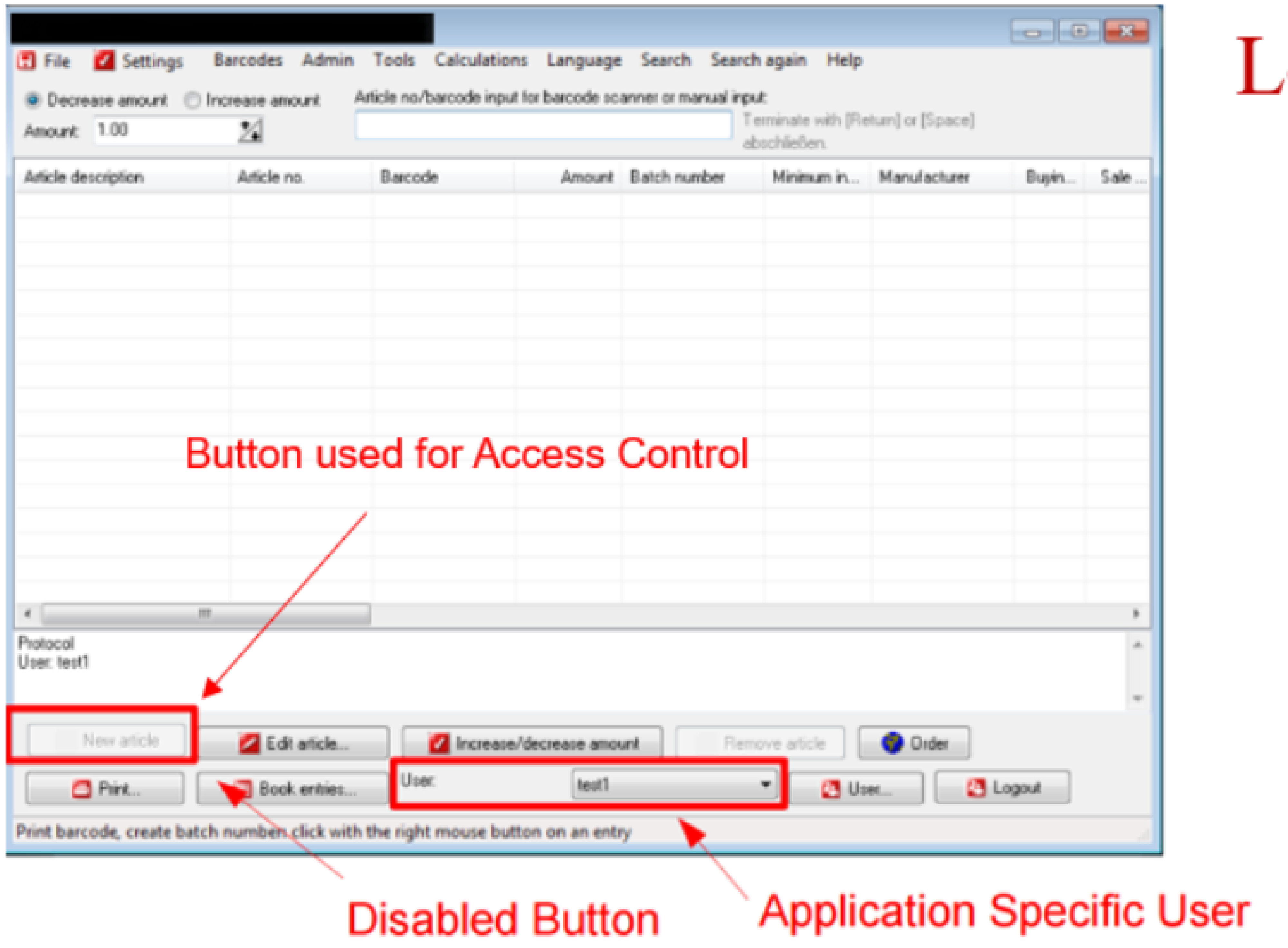
Disabled Button

Application Specific User

Behind the scenes

```
void new_article() {
    /* Not for test1 user */
    article = gettext();
    record(article);
}
```

Threat Model



[redacted]

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:
Terminate with [Return] or [Space] abschließen.
Article description
Article no.
Barcode
Amount
Batch number
Minimum in...
Manufacturer
Buyin...
Sale ...

Protocol
User: test1

User: test1

Print barcode, create batch number: click with the right mouse button on an entry

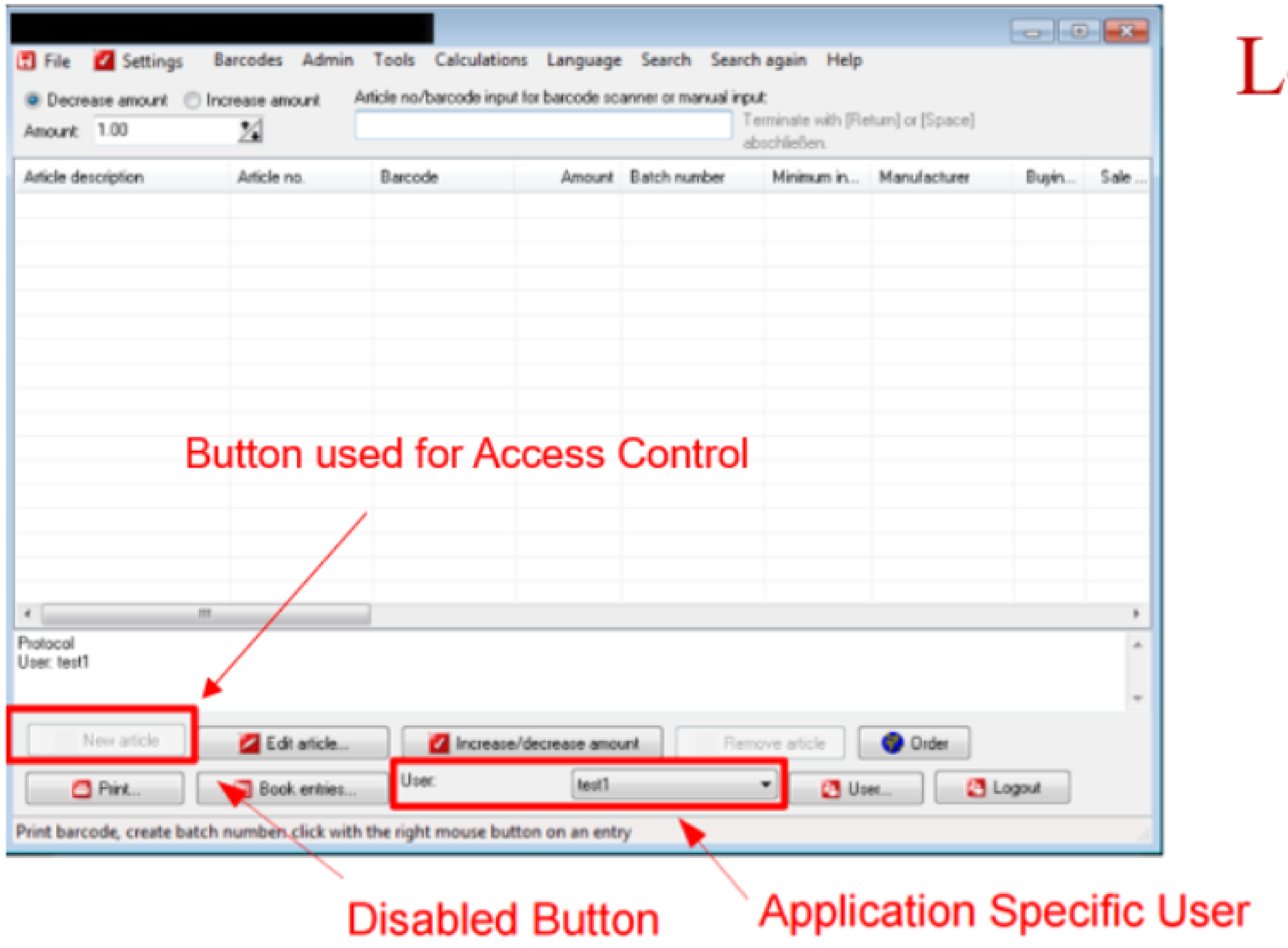
Button used for Access Control

Disabled Button

Application Specific User

Logic vulnerability

Threat Model



[redacted]

● Decrease amount ○ Increase amount

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:

Terminate with [Return] or [Space] abschließen.

Article description	Article no.	Barcode	Amount	Batch number	Minimum in...	Manufacturer	Buyin...	Sale ...

Protocol
User: test1

User: test1

Print barcode, create batch number, click with the right mouse button on an entry

Button used for Access Control

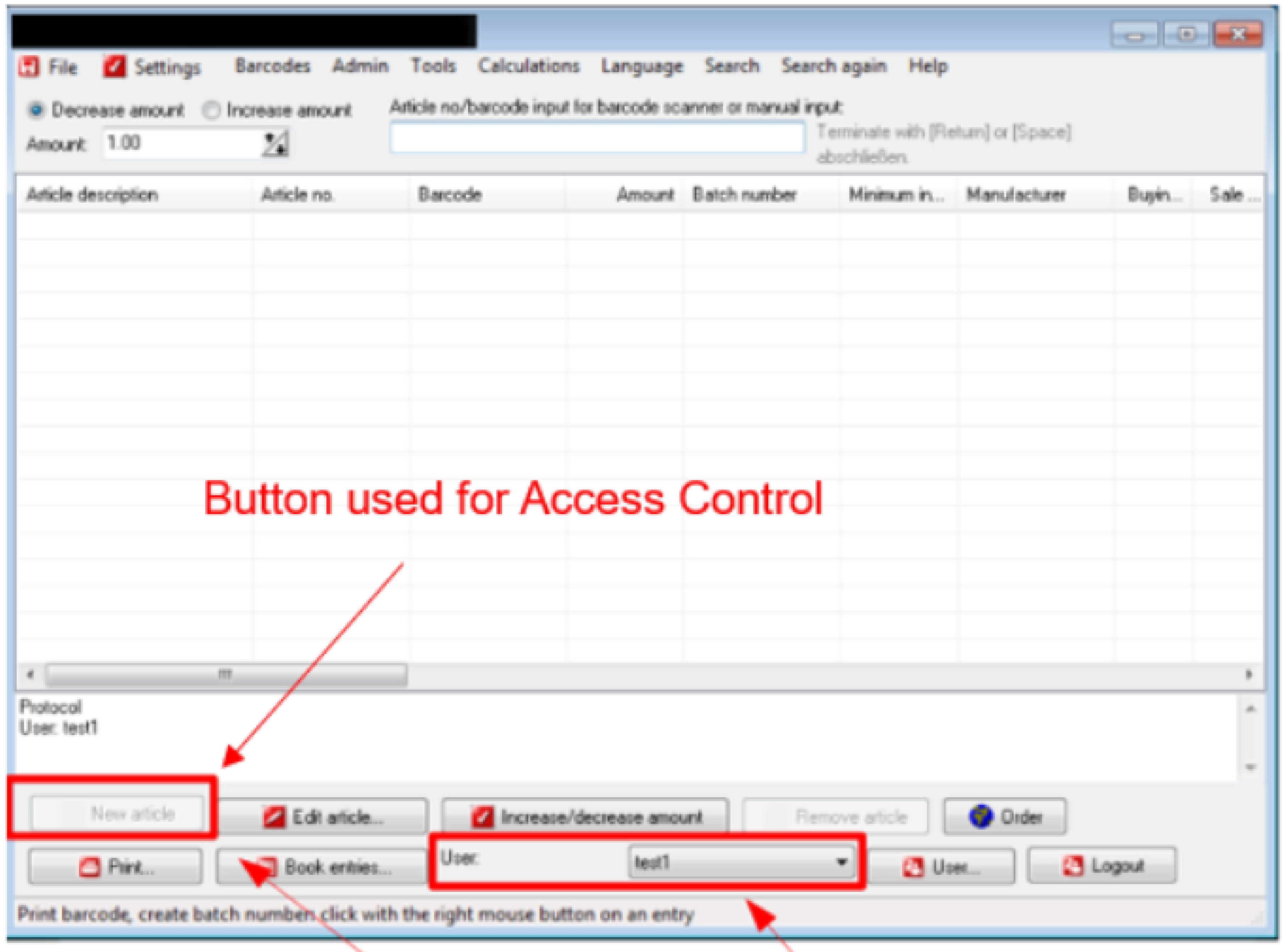
Disabled Button

Application Specific User

Logic vulnerability

- Enforcing the access to **new_article()** using the GUI (e.g., disabling the button)

Threat Model



[redacted]

● Decrease amount ○ Increase amount

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:

Terminate with [Return] or [Space] abschließen.

Article description	Article no.	Barcode	Amount	Batch number	Minimum in...	Manufacturer	Buyin...	Sale ...

Protocol
User: test1

User: test1

Print barcode, create batch number: click with the right mouse button on an entry

Button used for Access Control

Disabled Button

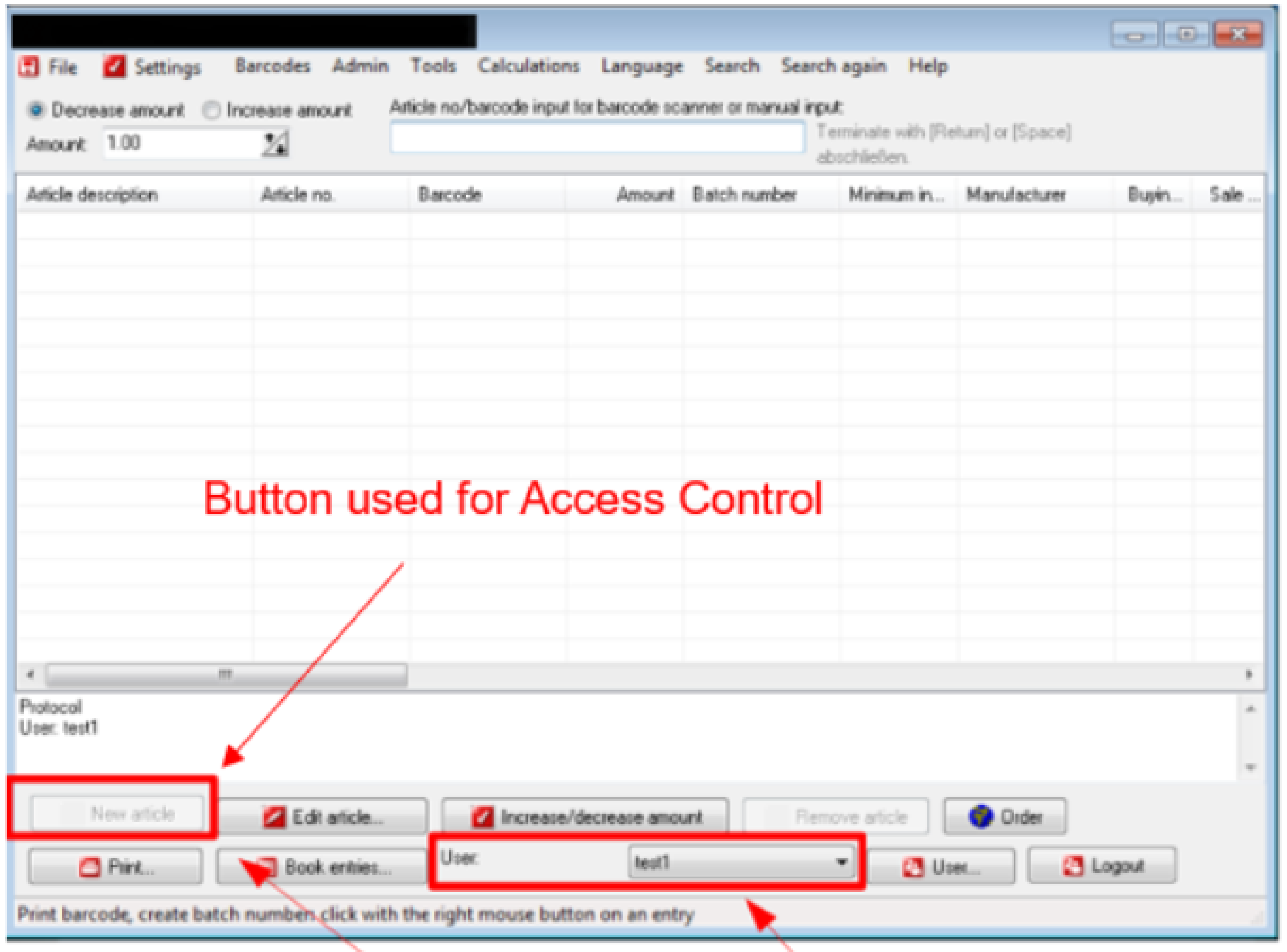
Application Specific User

Logic vulnerability

- Enforcing the access to **new_article()** using the GUI (e.g., disabling the button)

Memory corruption vulnerability

Threat Model



[redacted]

● Decrease amount ○ Increase amount

Amount: 1.00

Article no/barcode input for barcode scanner or manual input:

Terminate with [Return] or [Space] abschließen.

Article description	Article no.	Barcode	Amount	Batch number	Minimum in...	Manufacturer	Buyin...	Sale ...

Protocol
User: test1

User: test1

Print barcode, create batch number, click with the right mouse button on an entry

Button used for Access Control

Disabled Button

Application Specific User

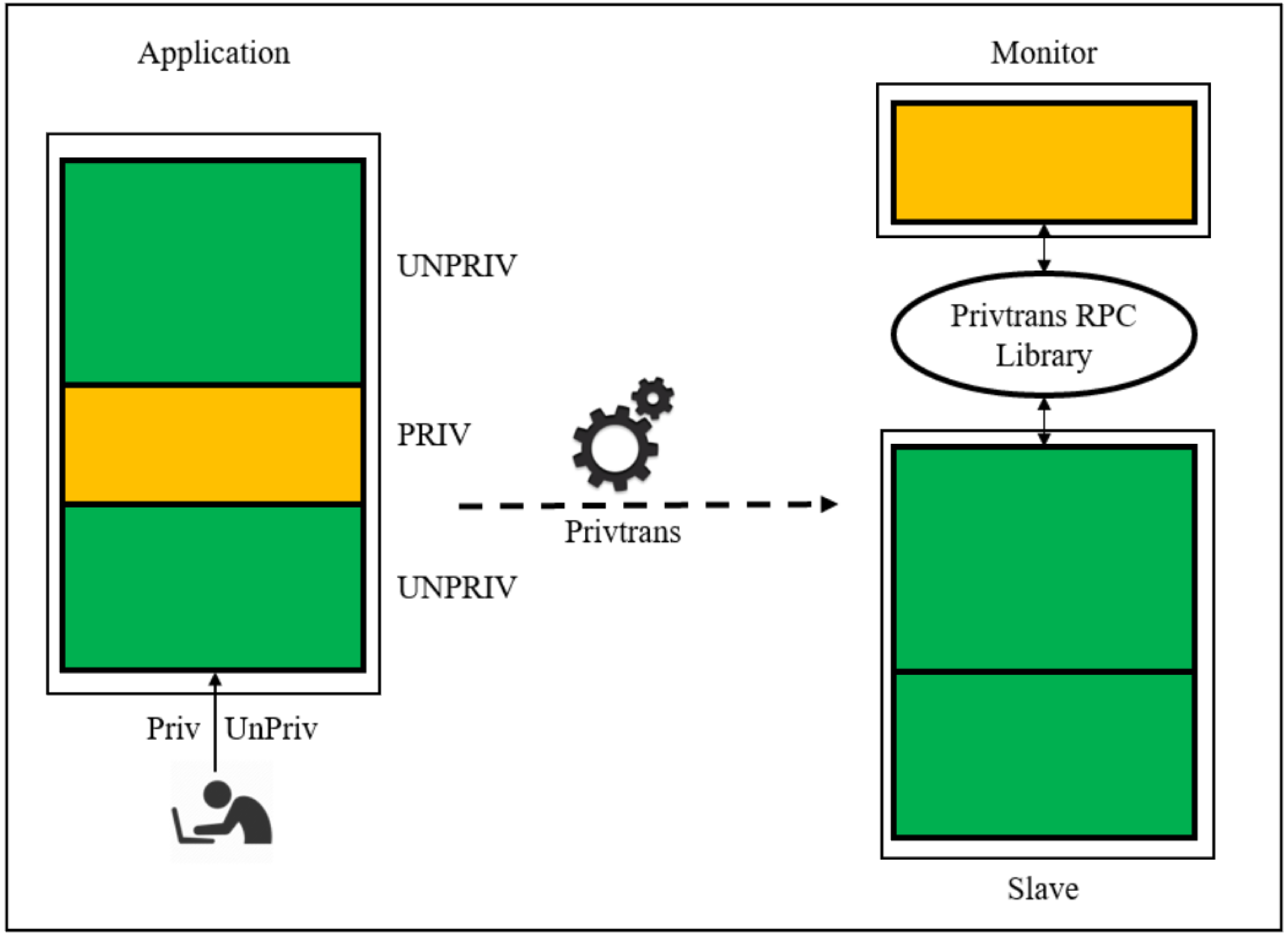
Logic vulnerability

- Enforcing the access to **new_article()** using the GUI (e.g., disabling the button)

Memory corruption vulnerability

- Exploit “test1” available functions and hijack the control flow to **new_article()**

Previous Work - Privtrans



D. Brumley, and D. Song. *USENIX Security ’04*

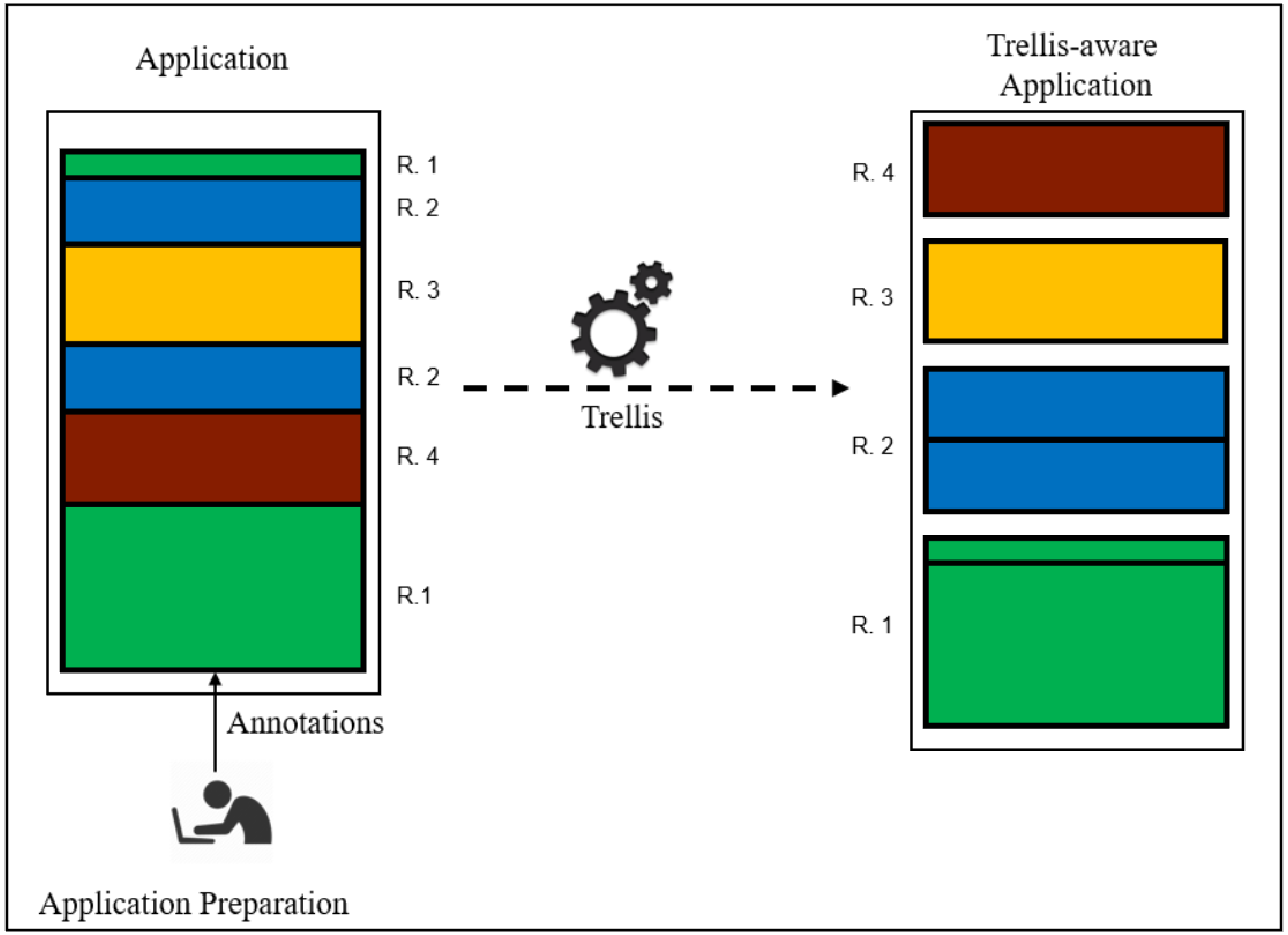
Previous Work - Limitations

Does not support multiple roles

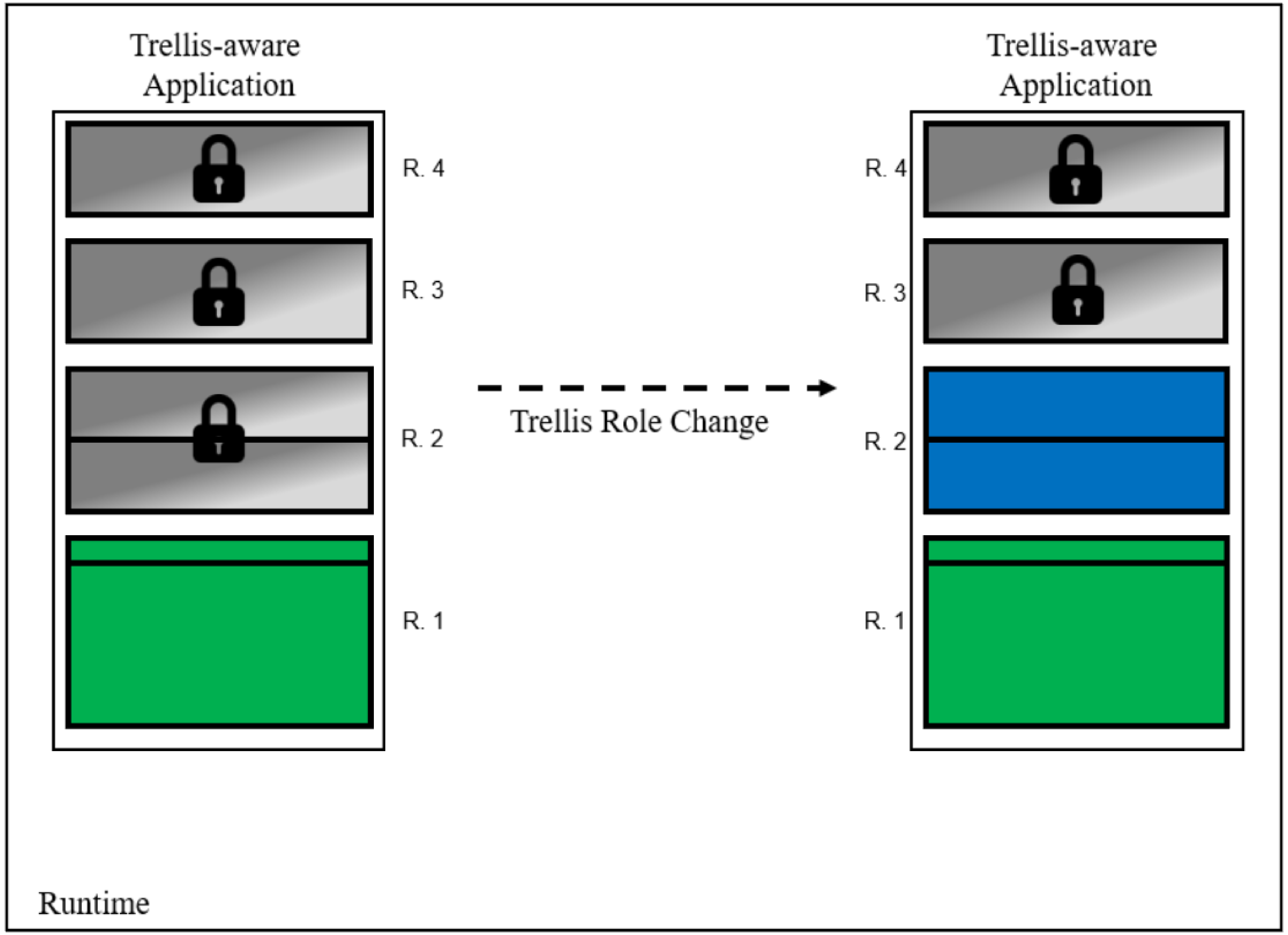
Physical application partitioning

IPC overhead

Trellis Approach



Trellis Approach



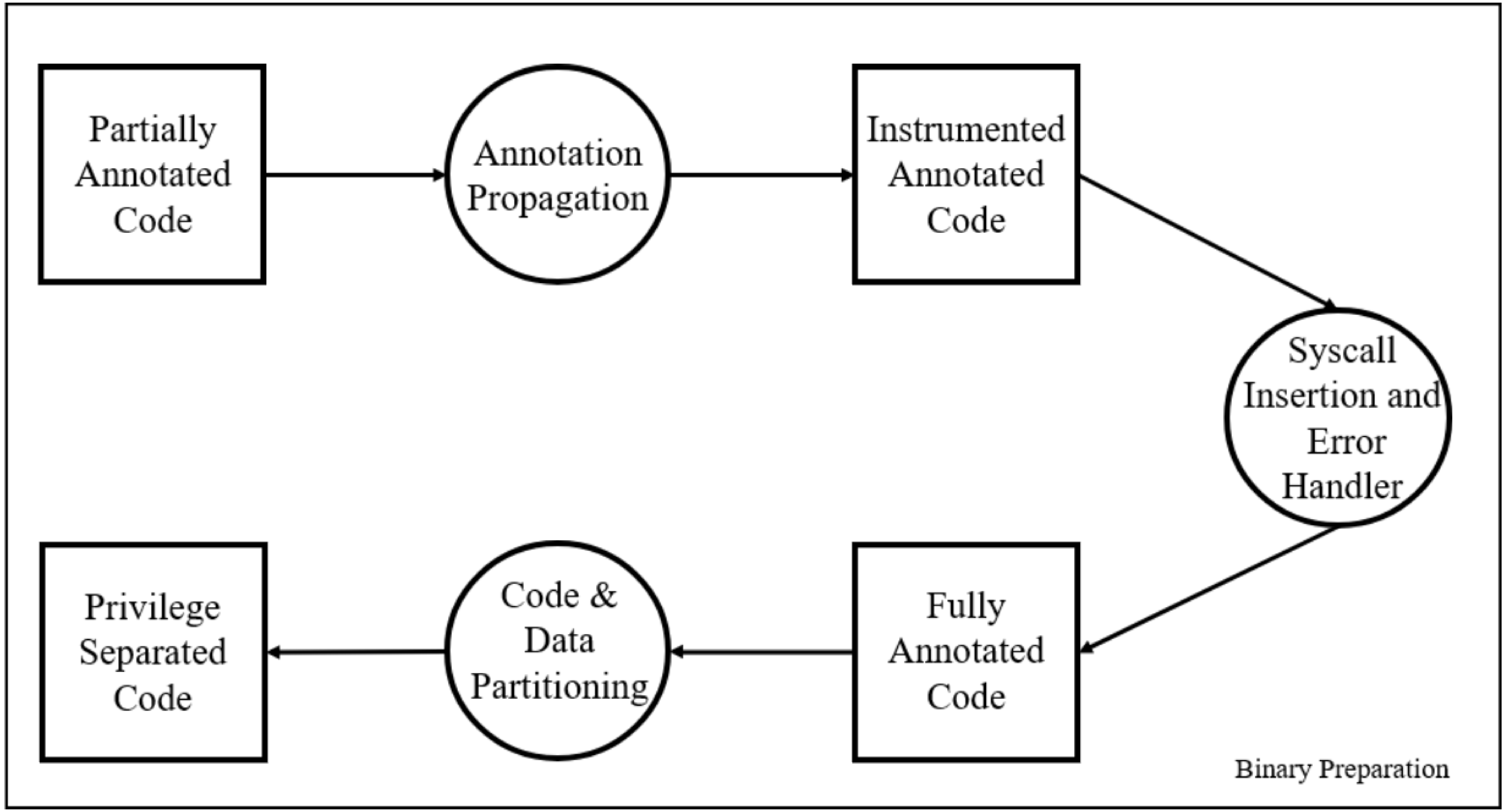
Trellis Contributions

OS-based application development framework for multi-user application to enforce Hierarchical-Base Access Control policies (HBAC)

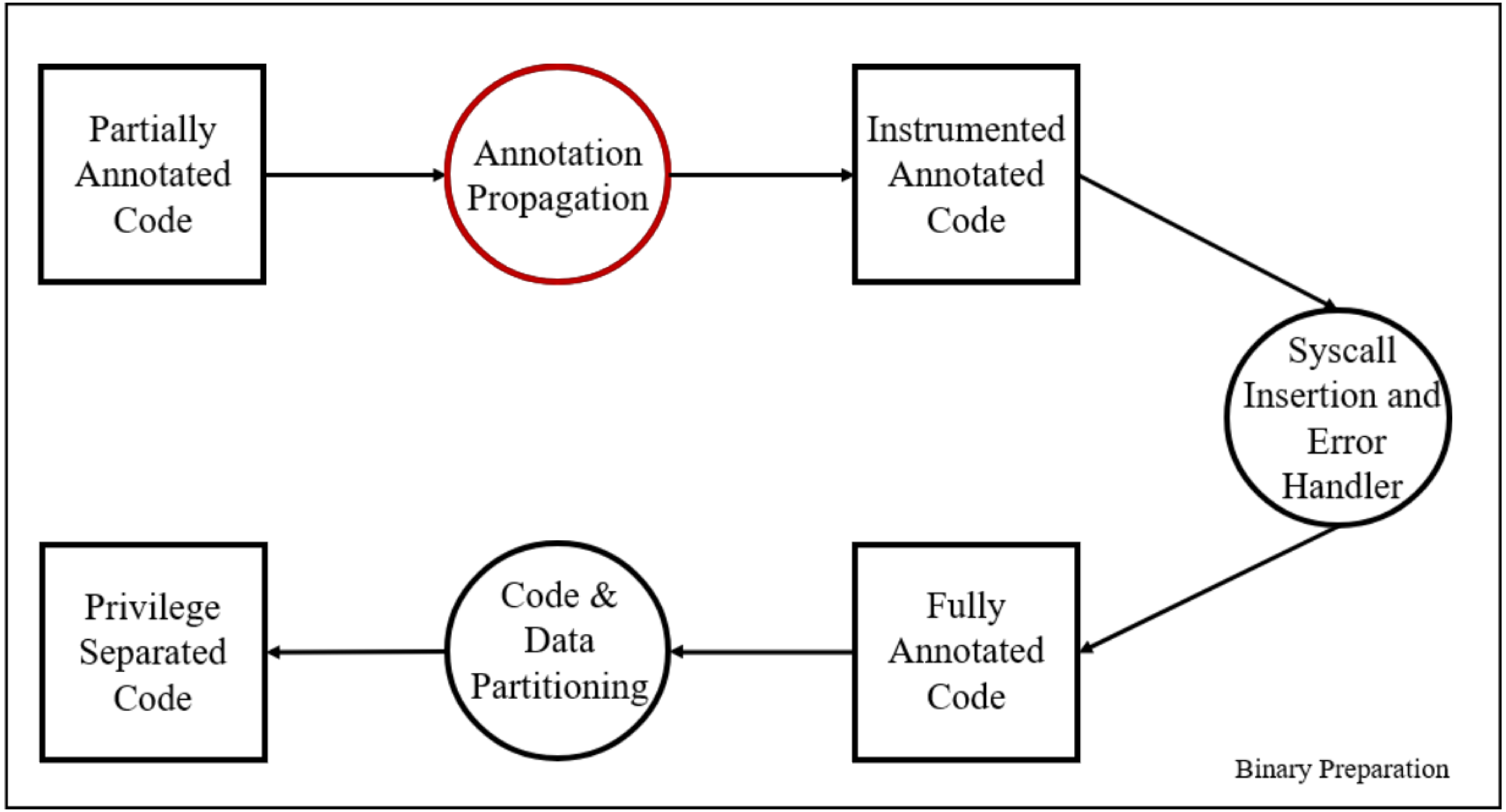
Prototype implementation based on LLVM/Clang, GNU C Library, and the Linux Kernel

Real-world application evaluation

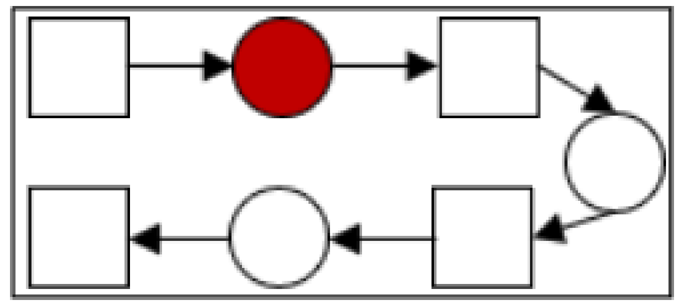
Implementation



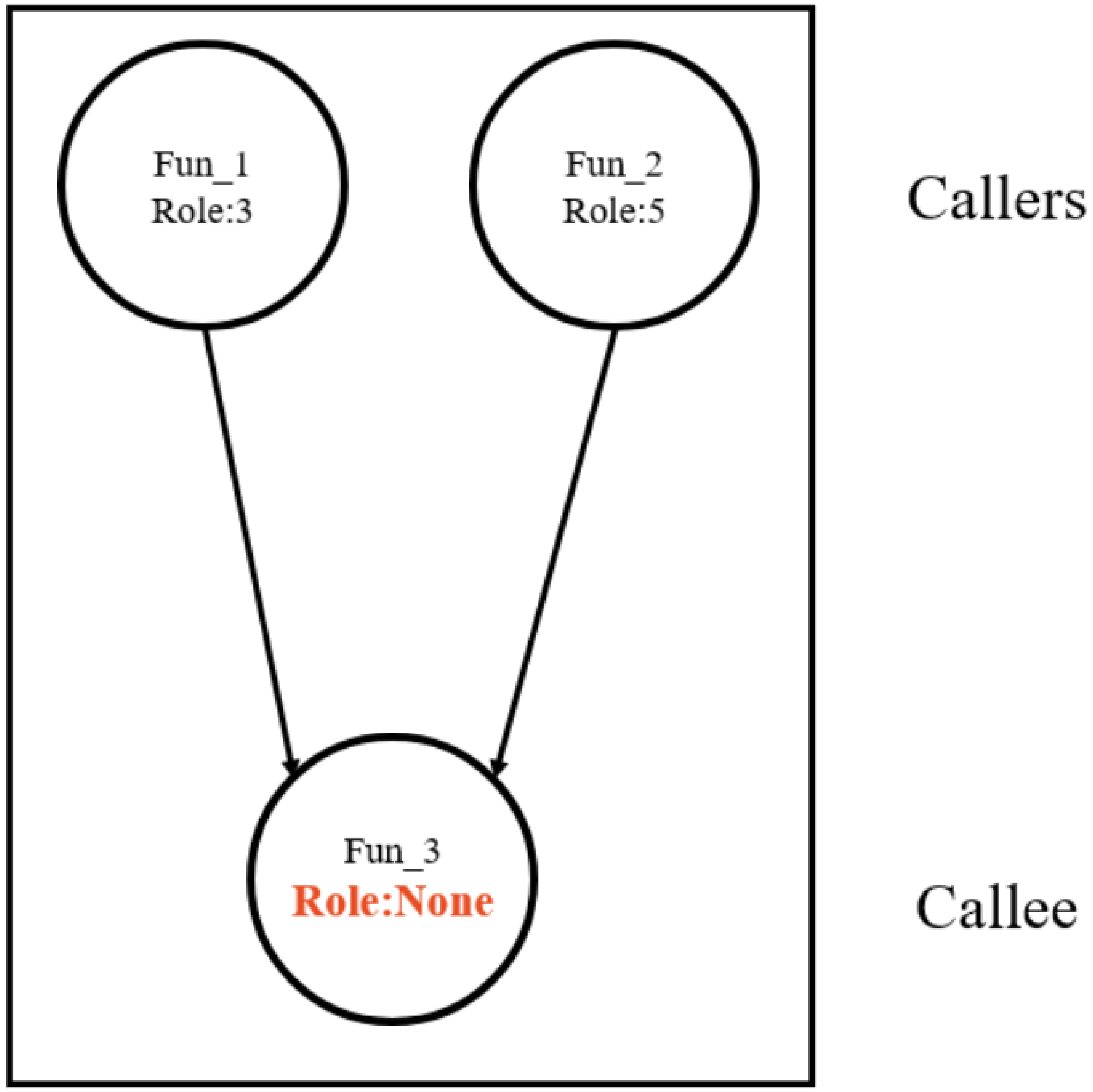
Implementation



Implementation

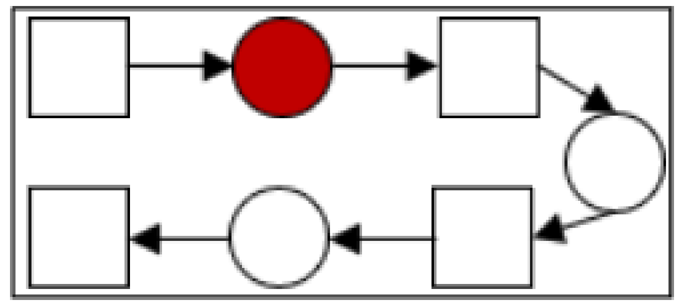


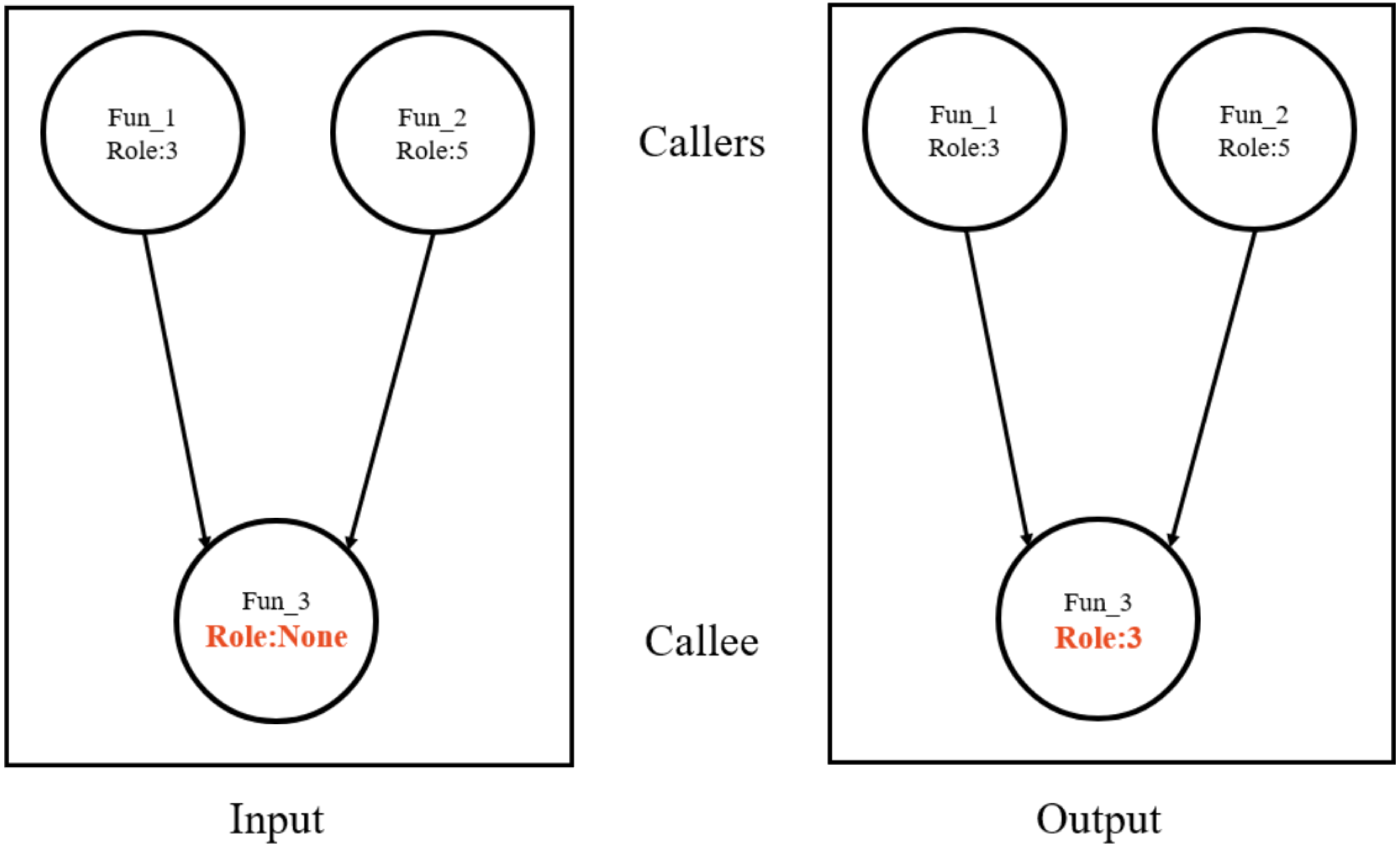
Implementation



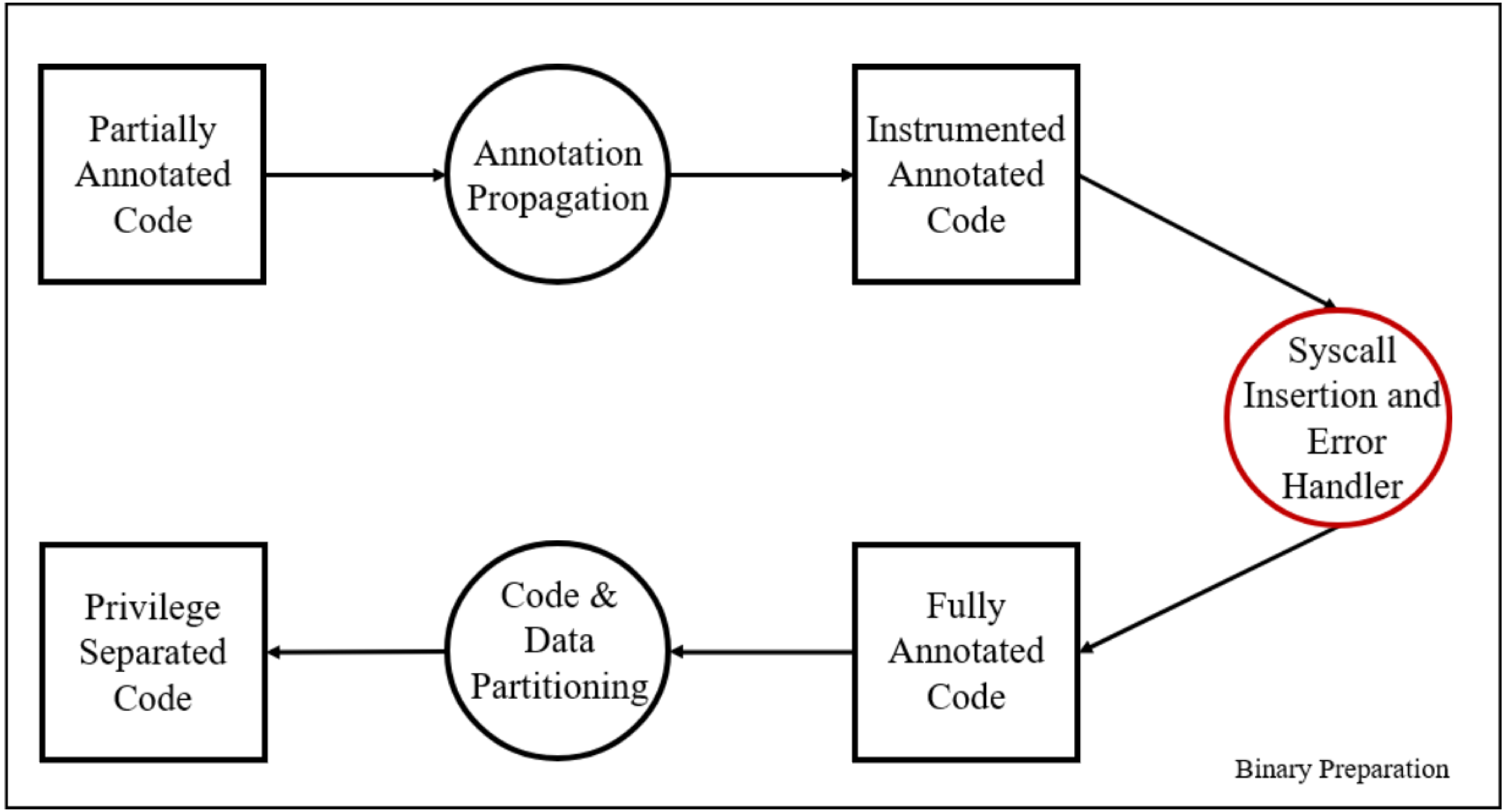
Input

Implementation

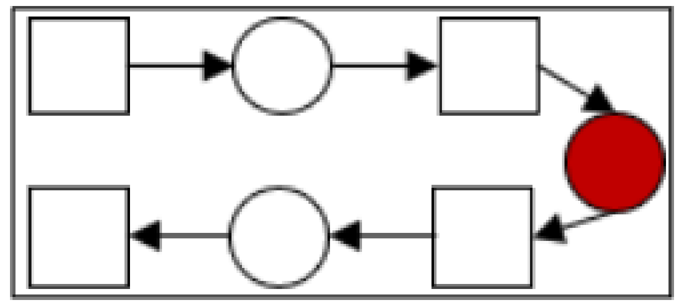


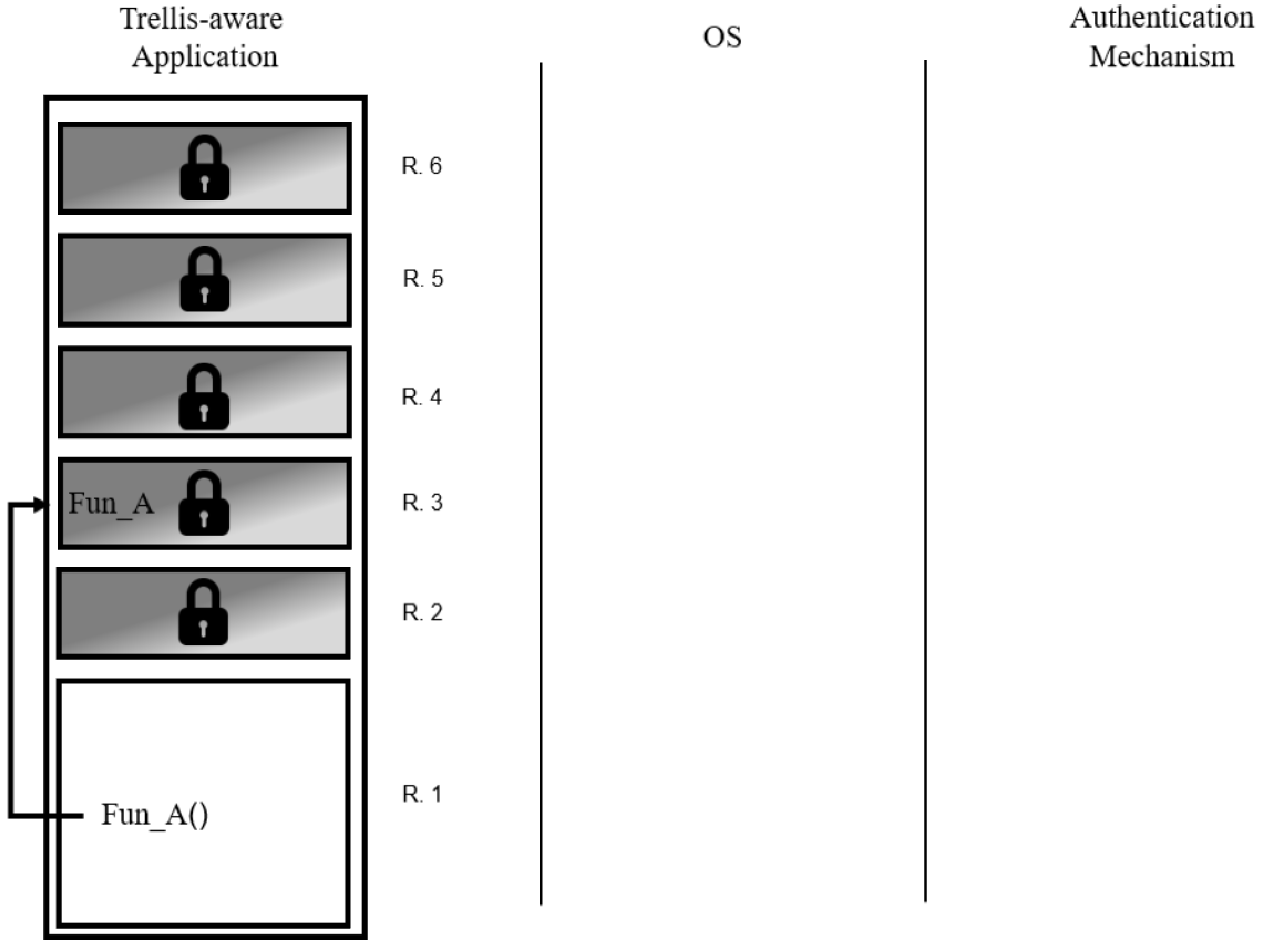


Implementation

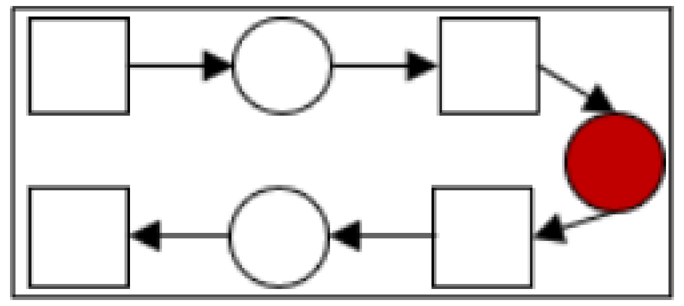


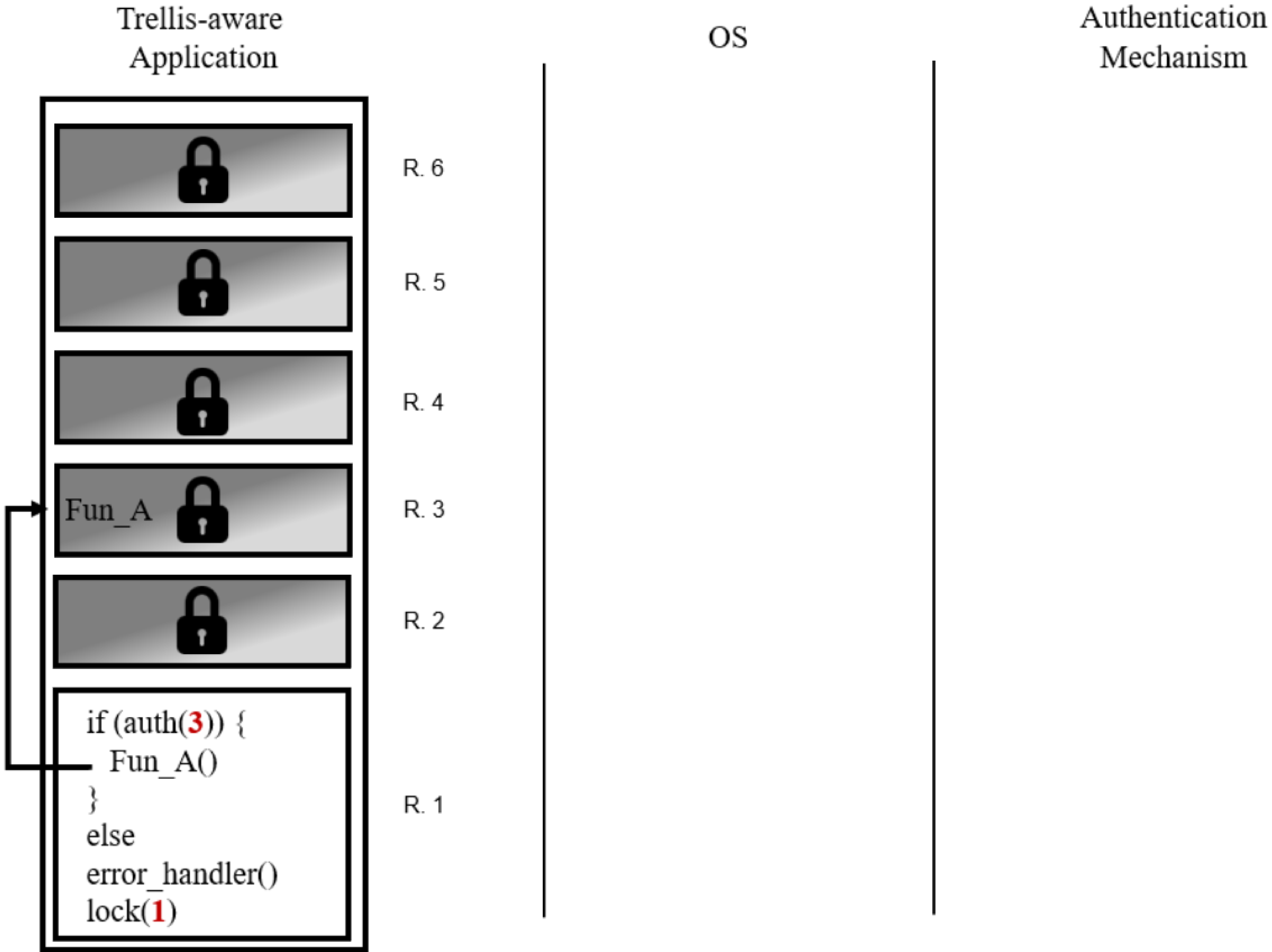
Implementation





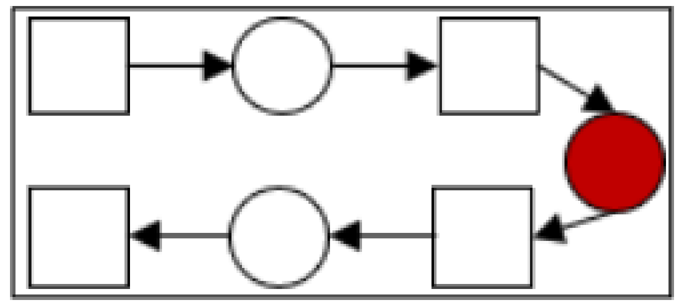
Implementation

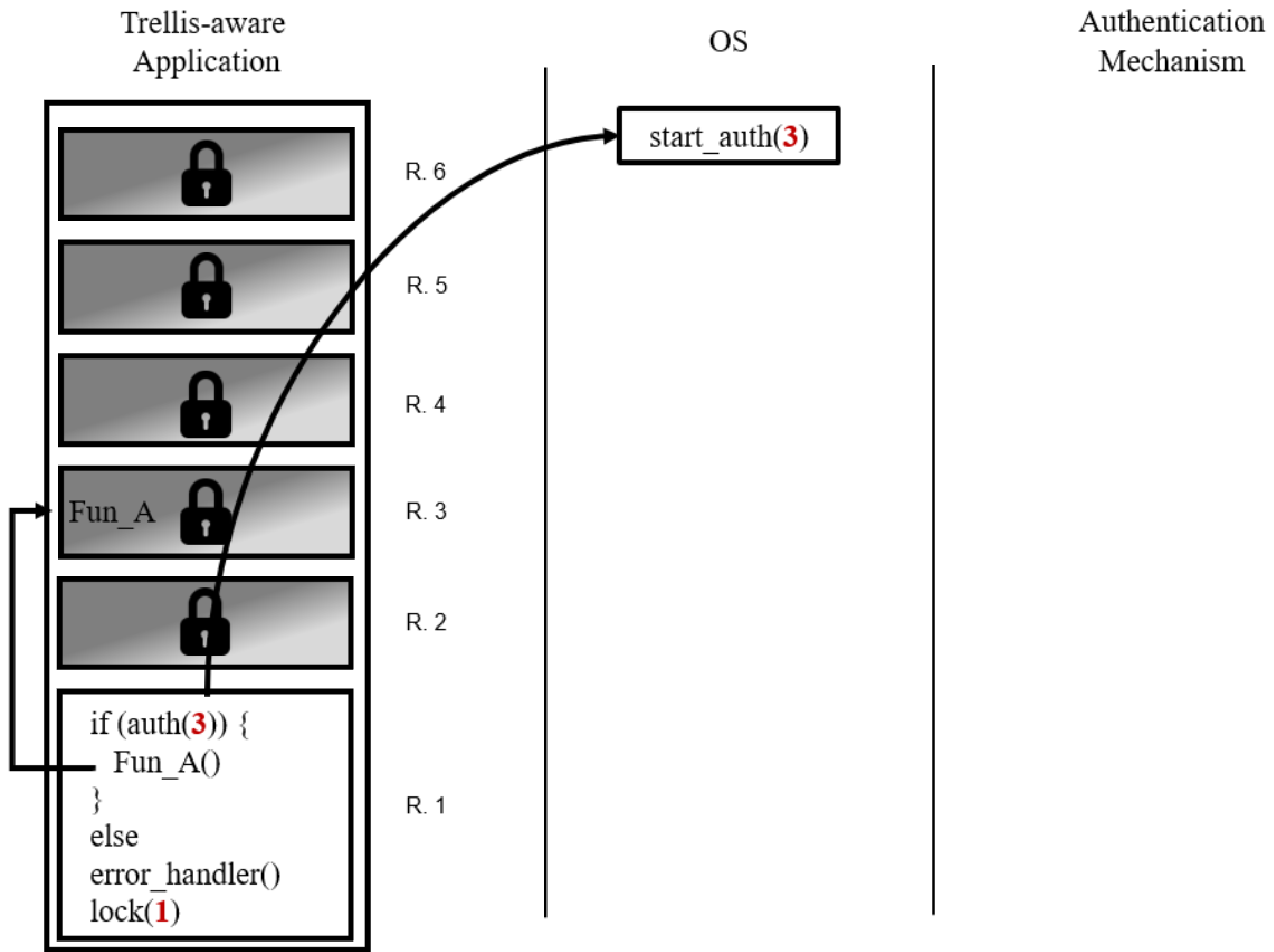




```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

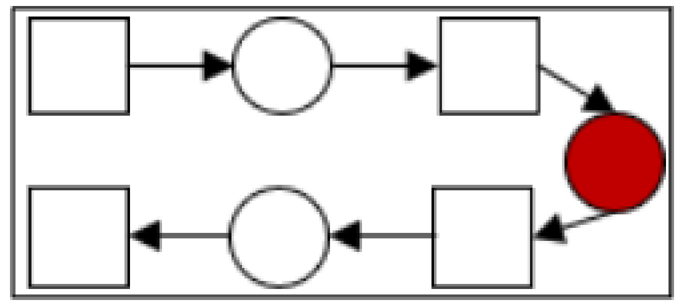
Implementation

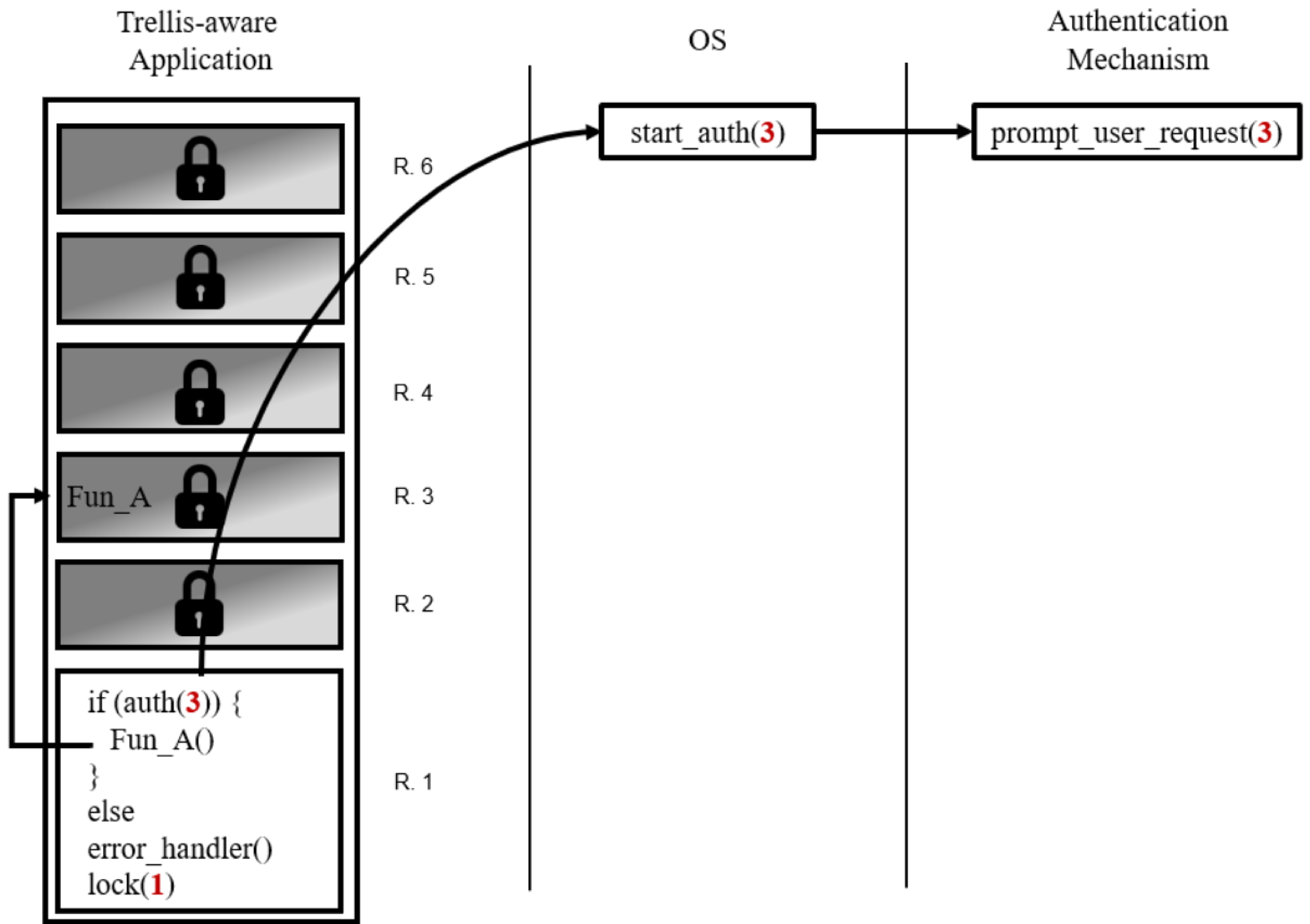




```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

Implementation





Trellis-aware Application

OS

Authentication Mechanism

start_auth(3)

prompt_user_request(3)

R. 6

R. 5

R. 4

R. 3

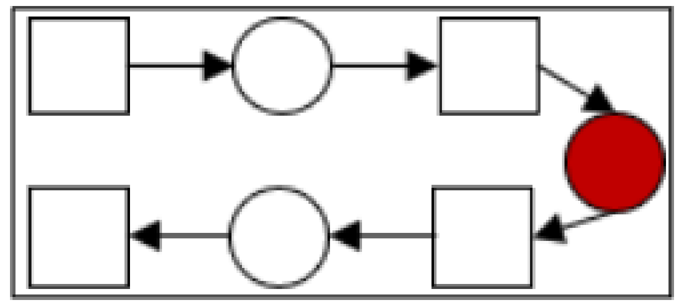
R. 2

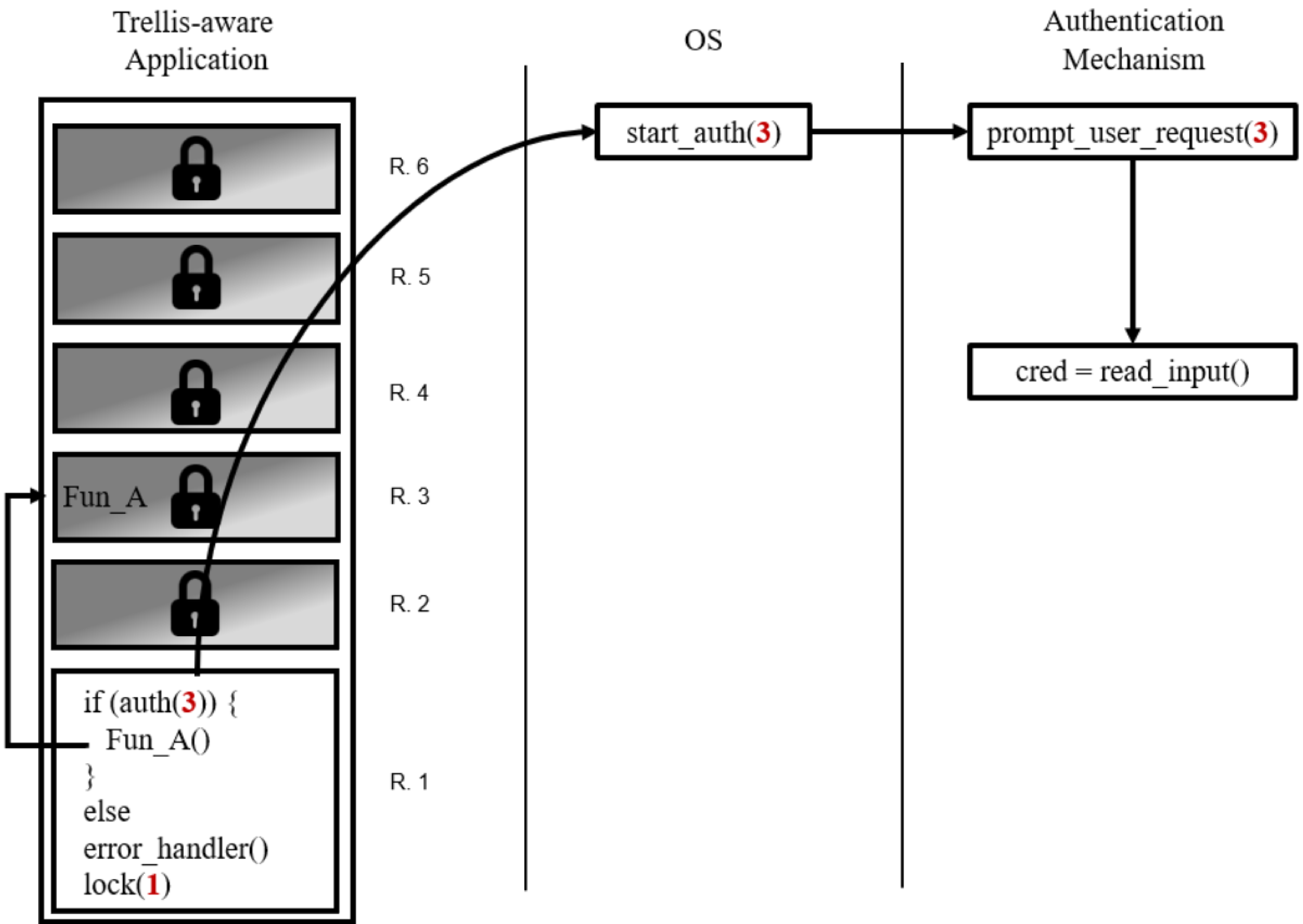
R. 1

Fun_A

```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

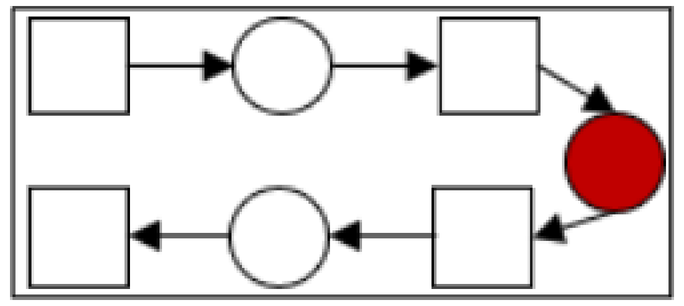
Implementation

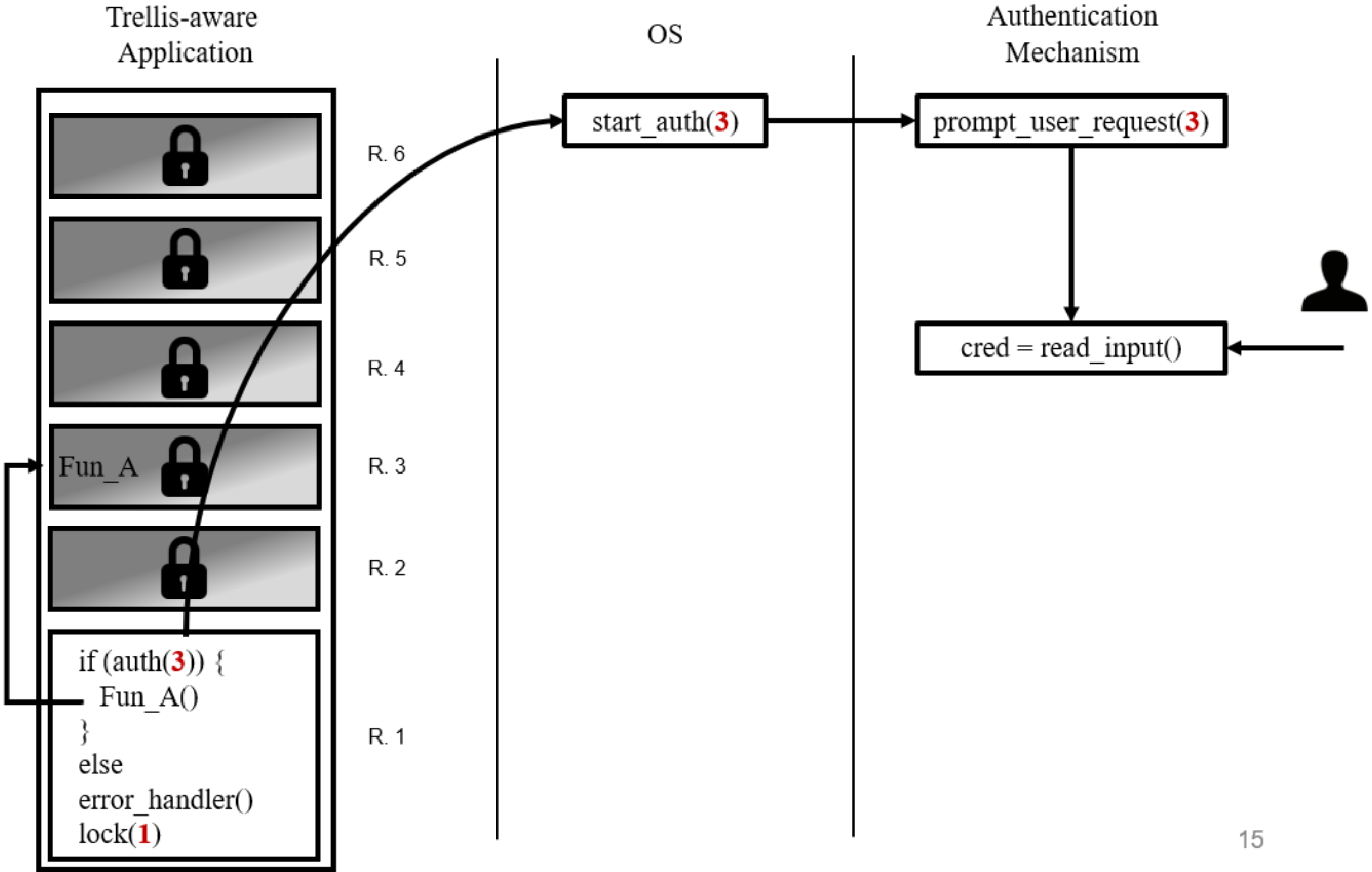




```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

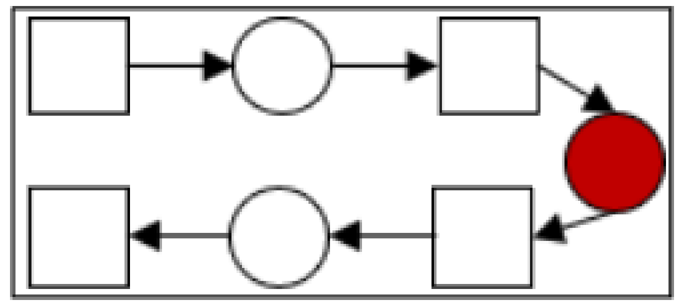
Implementation

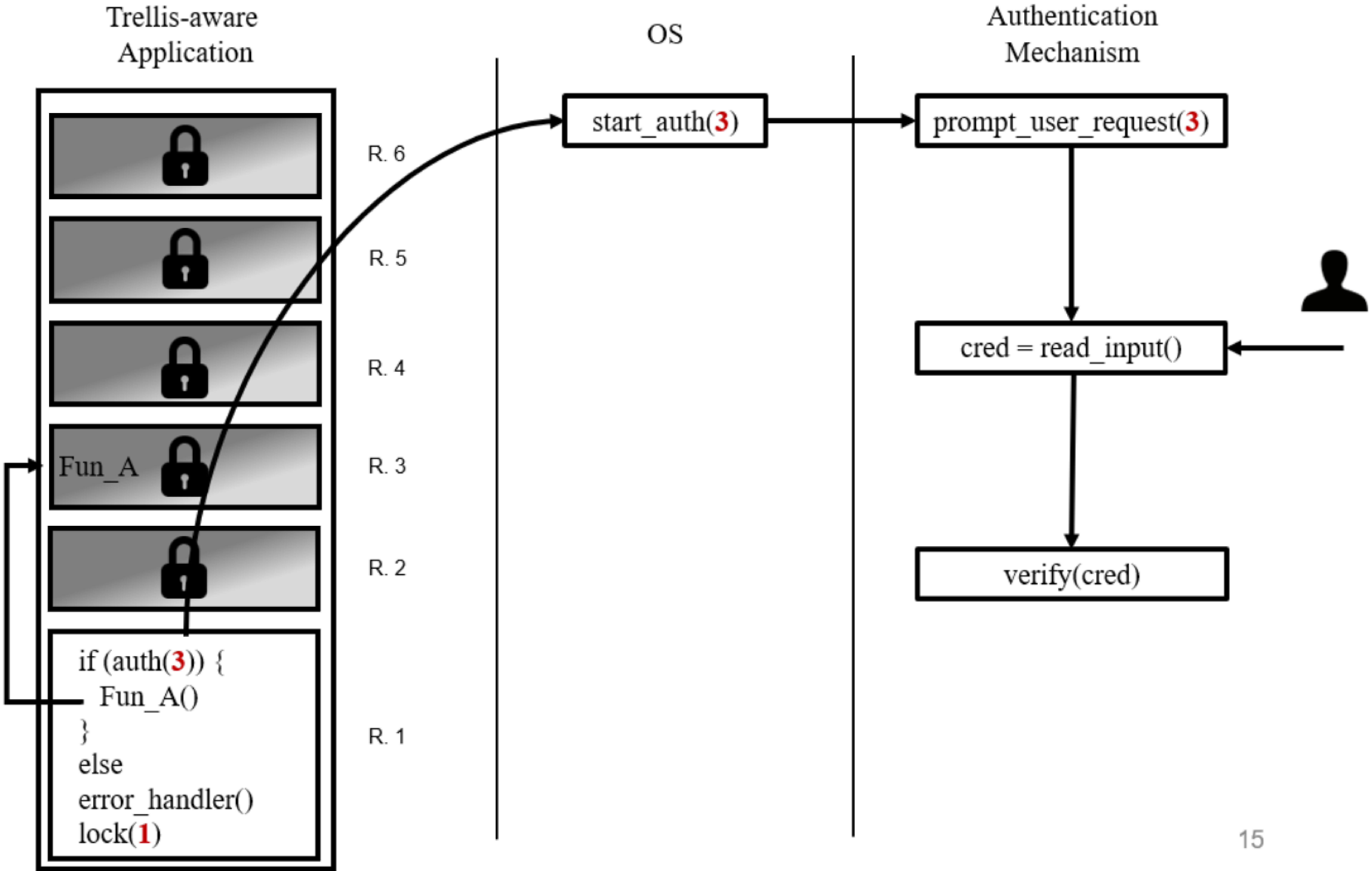




```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

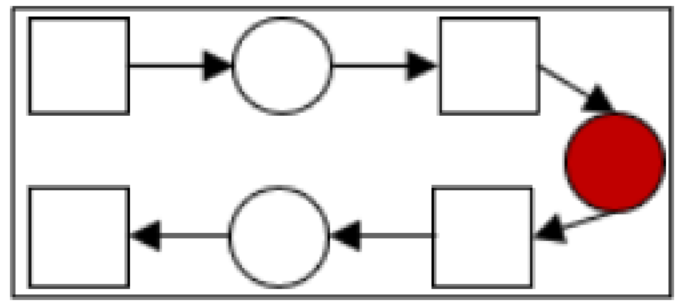
Implementation

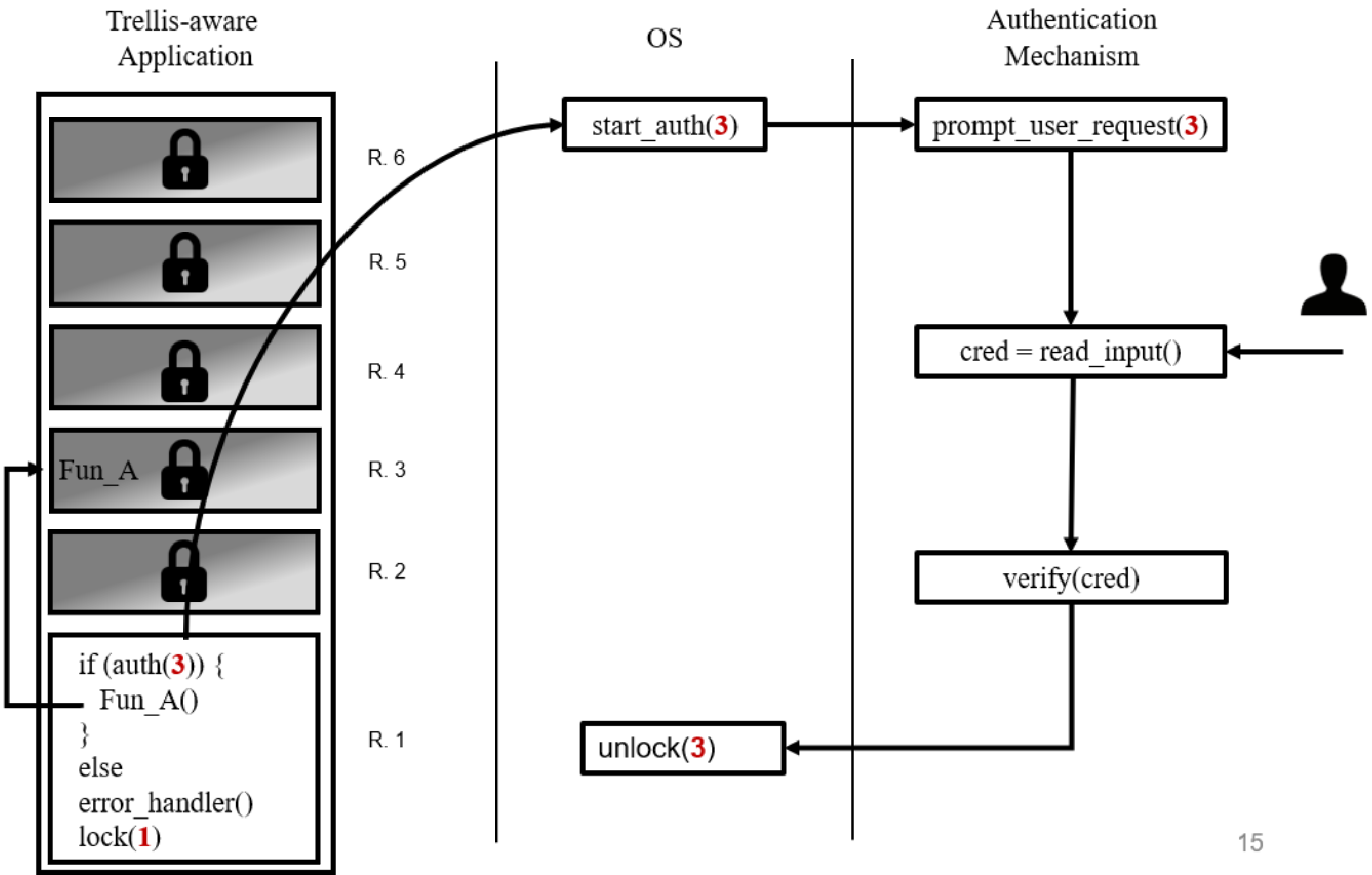




```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

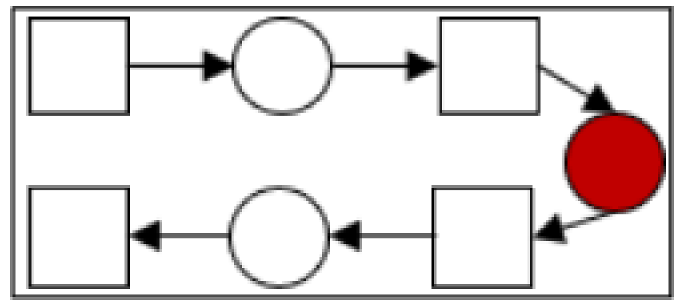
Implementation

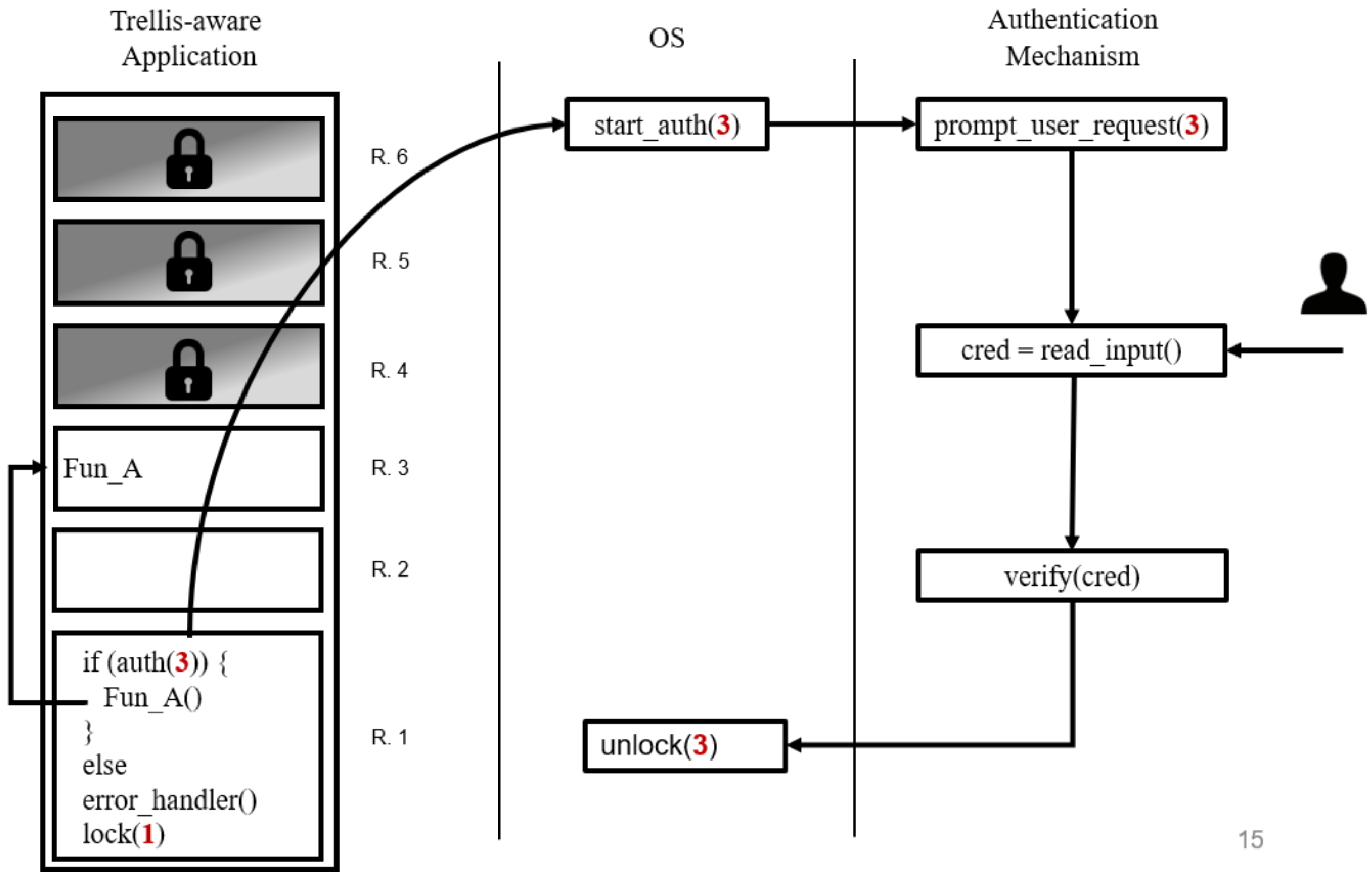




```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

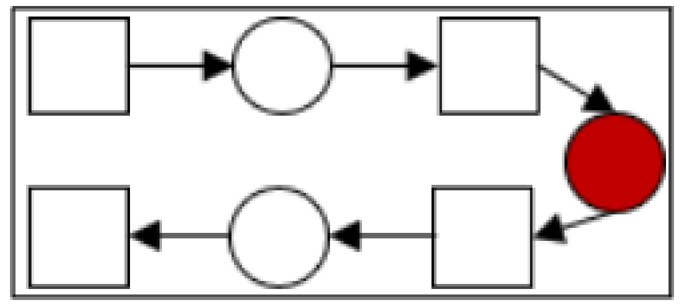
Implementation

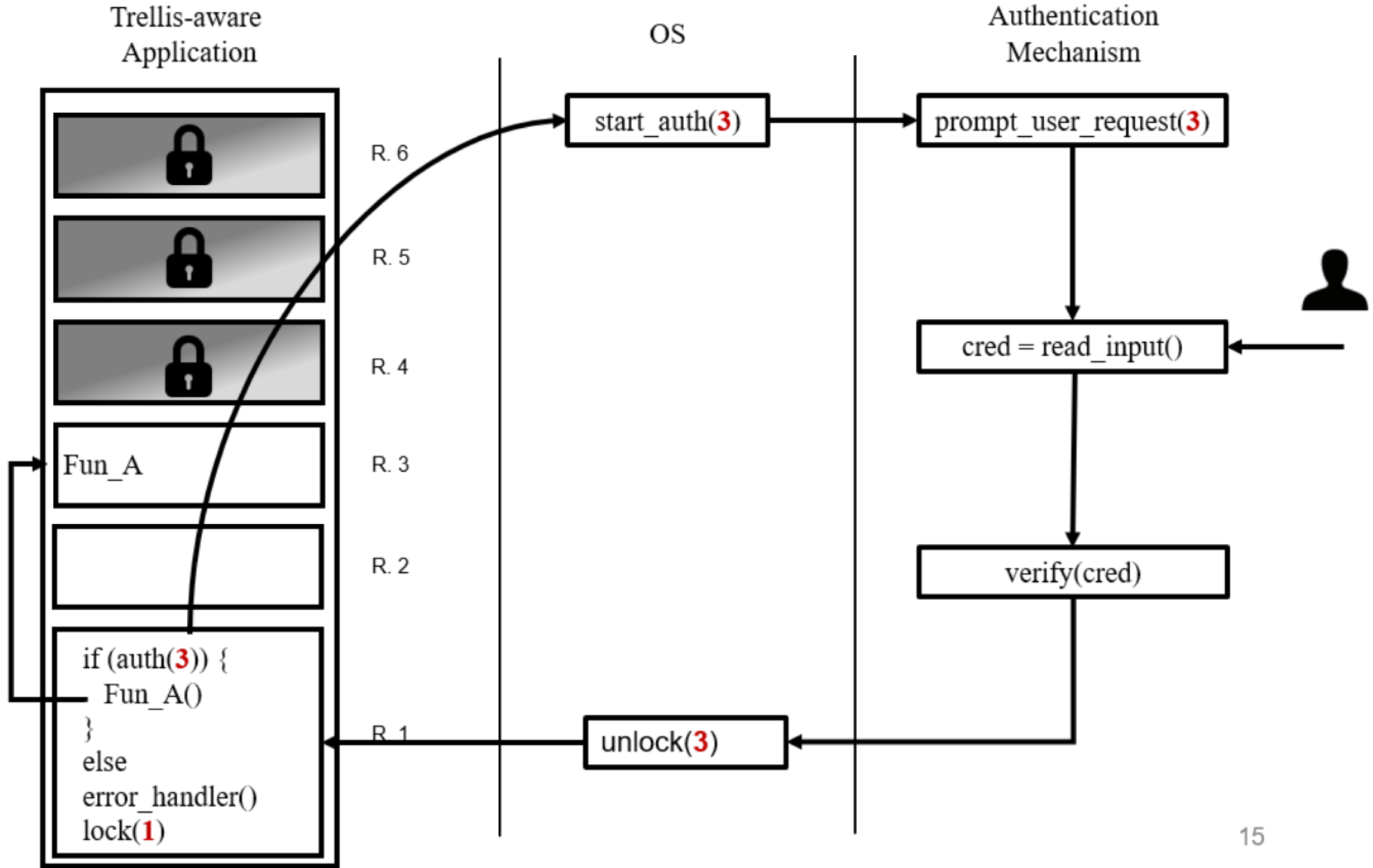




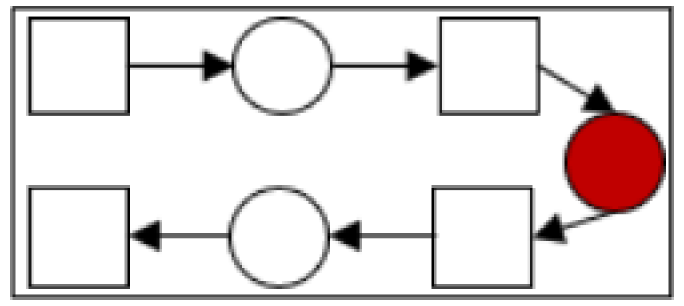
```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

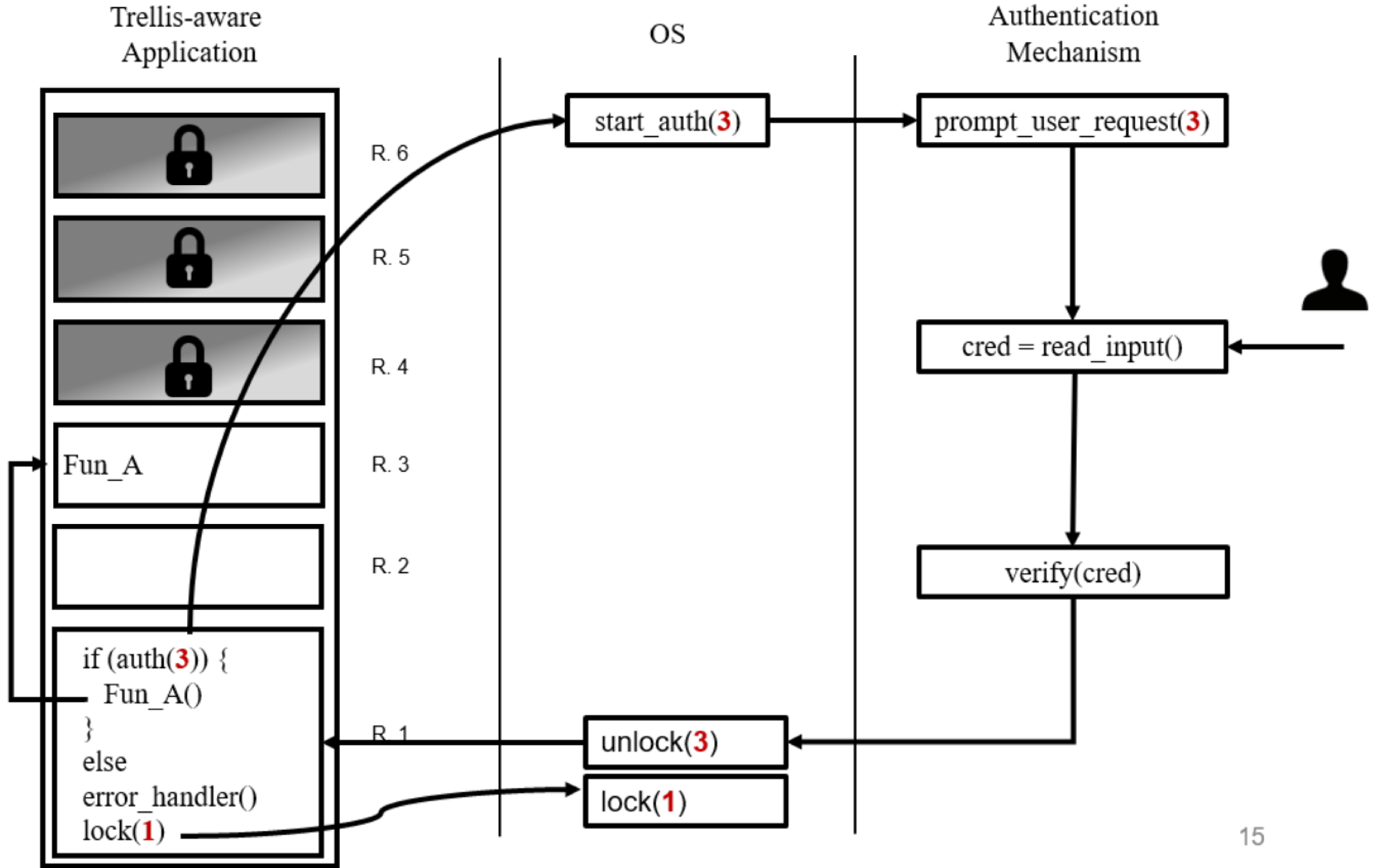
Implementation



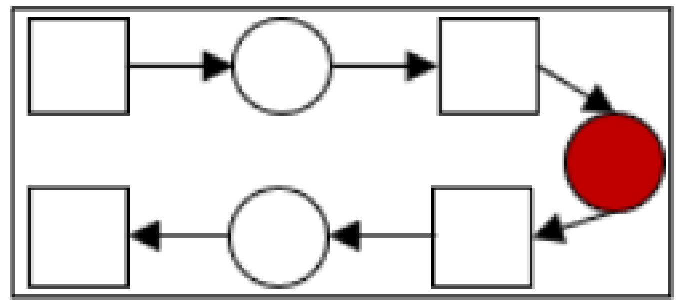


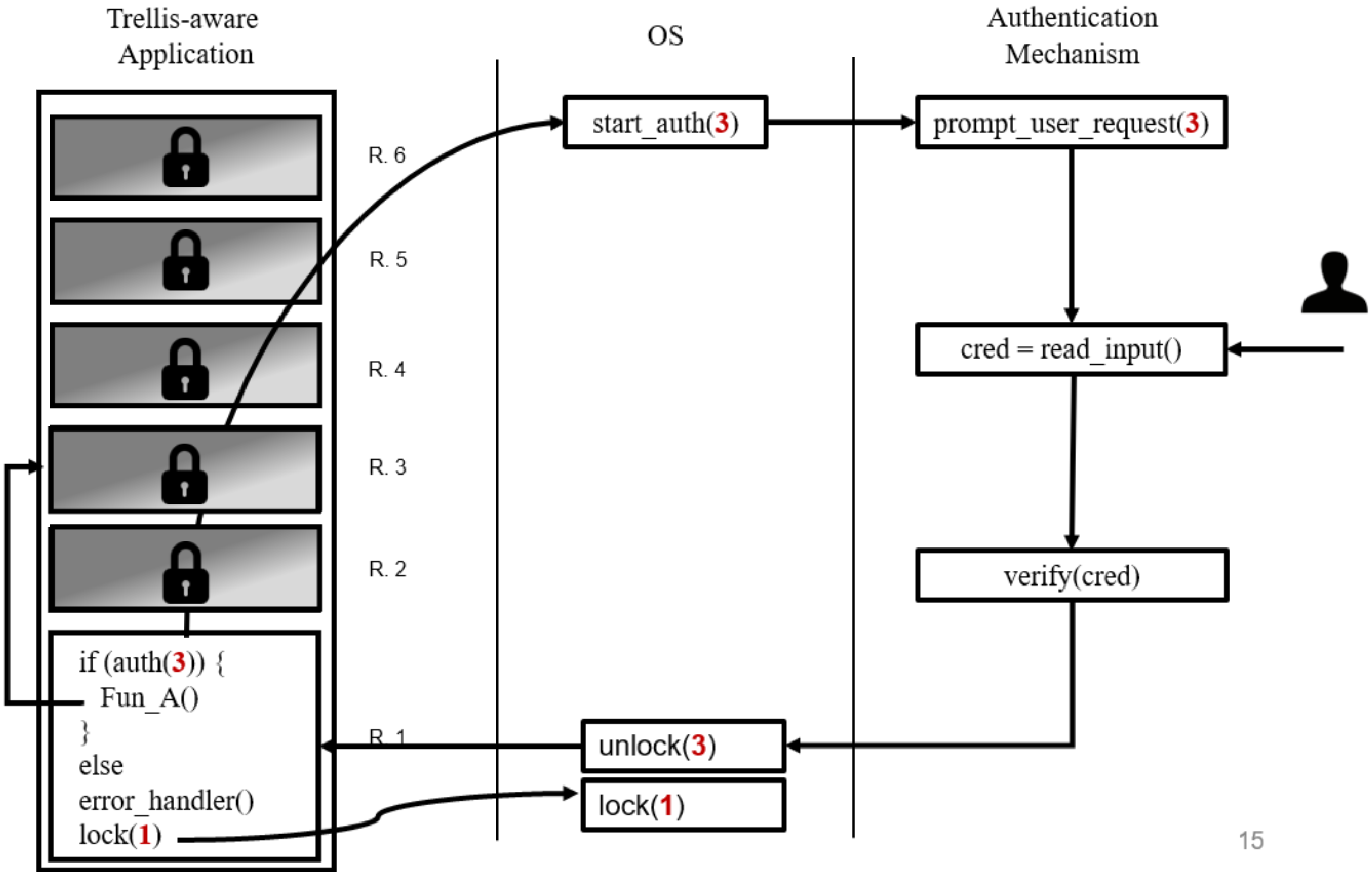
Implementation





Implementation





Trellis-aware Application

OS

Authentication Mechanism

R. 6

R. 5

R. 4

R. 3

R. 2

R. 1

```
if (auth(3)) {
  Fun_A()
}
else
error_handler()
lock(1)
```

start_auth(3)

prompt_user_request(3)

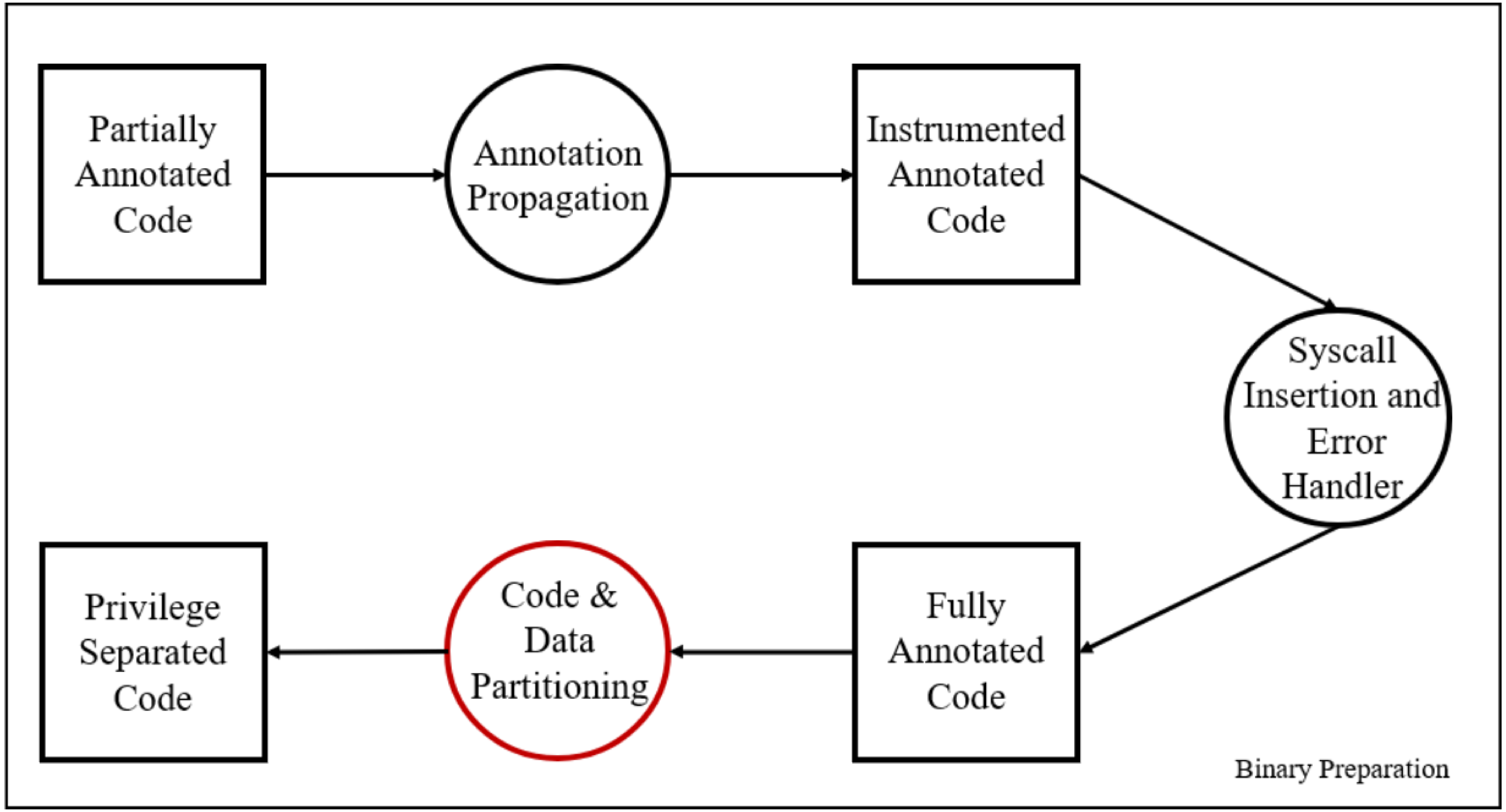
cred = read_input()

verify(cred)

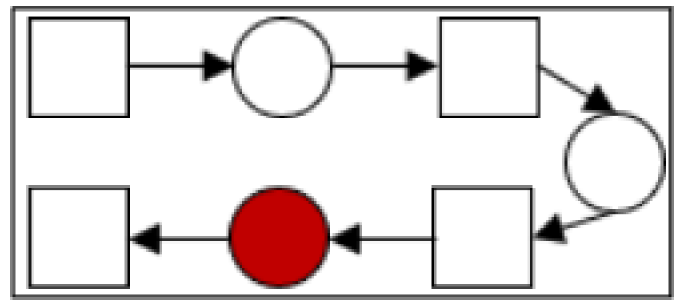
unlock(3)

lock(1)

Implementation



Implementation



Trellis-aware
App on disk

fun_1 : role 1

fun_2 : role 1

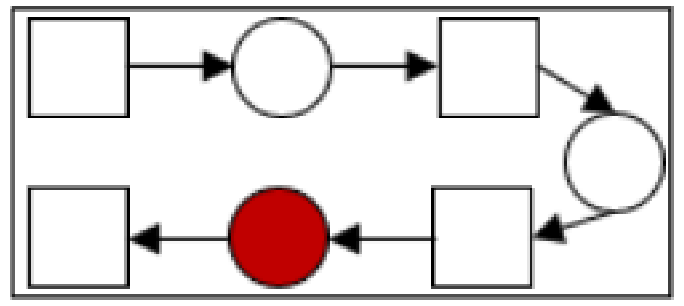
gv_1 : role 1

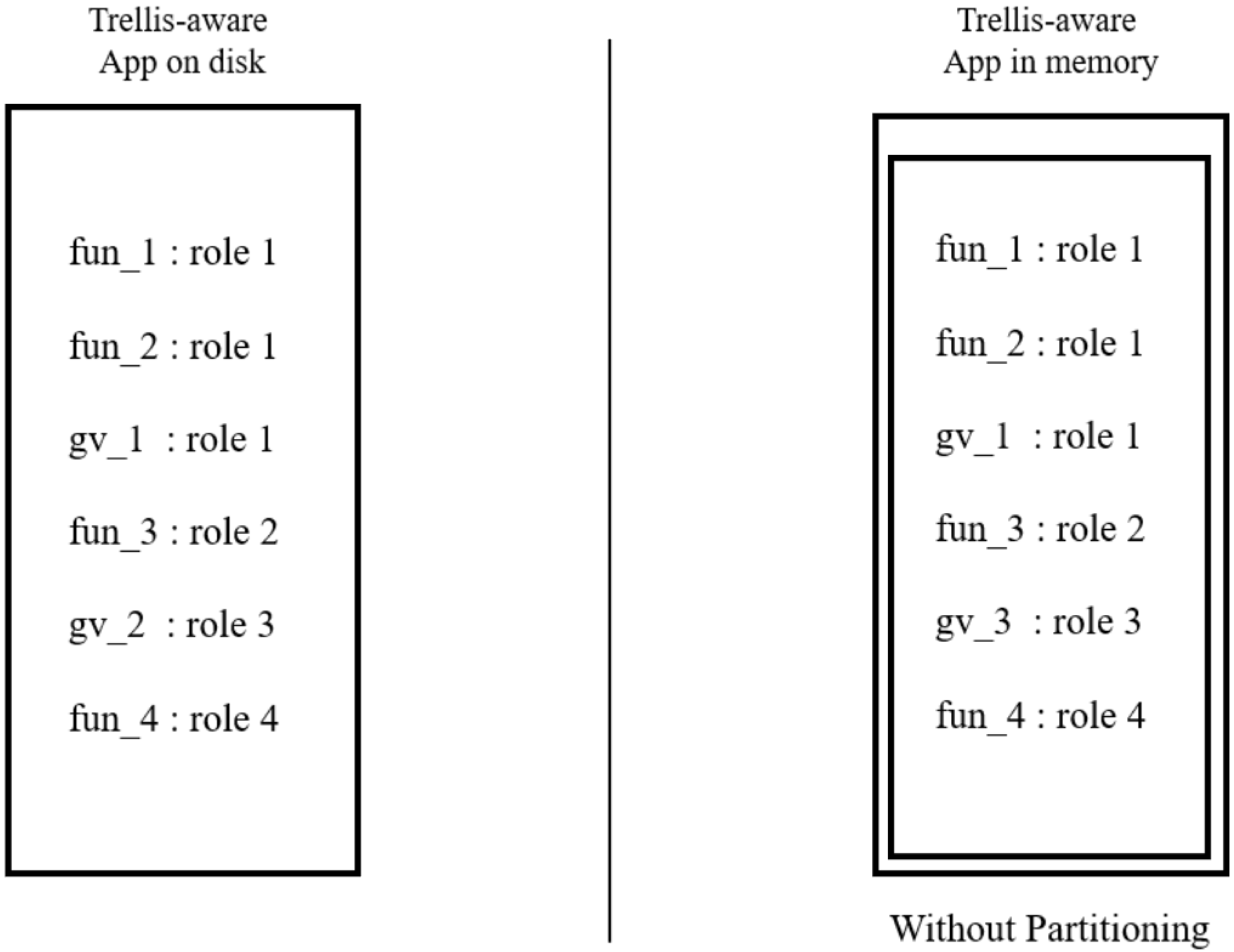
fun_3 : role 2

gv_2 : role 3

fun_4 : role 4

Implementation





Trellis-aware
App on disk

fun_1 : role 1

fun_2 : role 1

gv_1 : role 1

fun_3 : role 2

gv_2 : role 3

fun_4 : role 4

Trellis-aware
App in memory

fun_1 : role 1

fun_2 : role 1

gv_1 : role 1

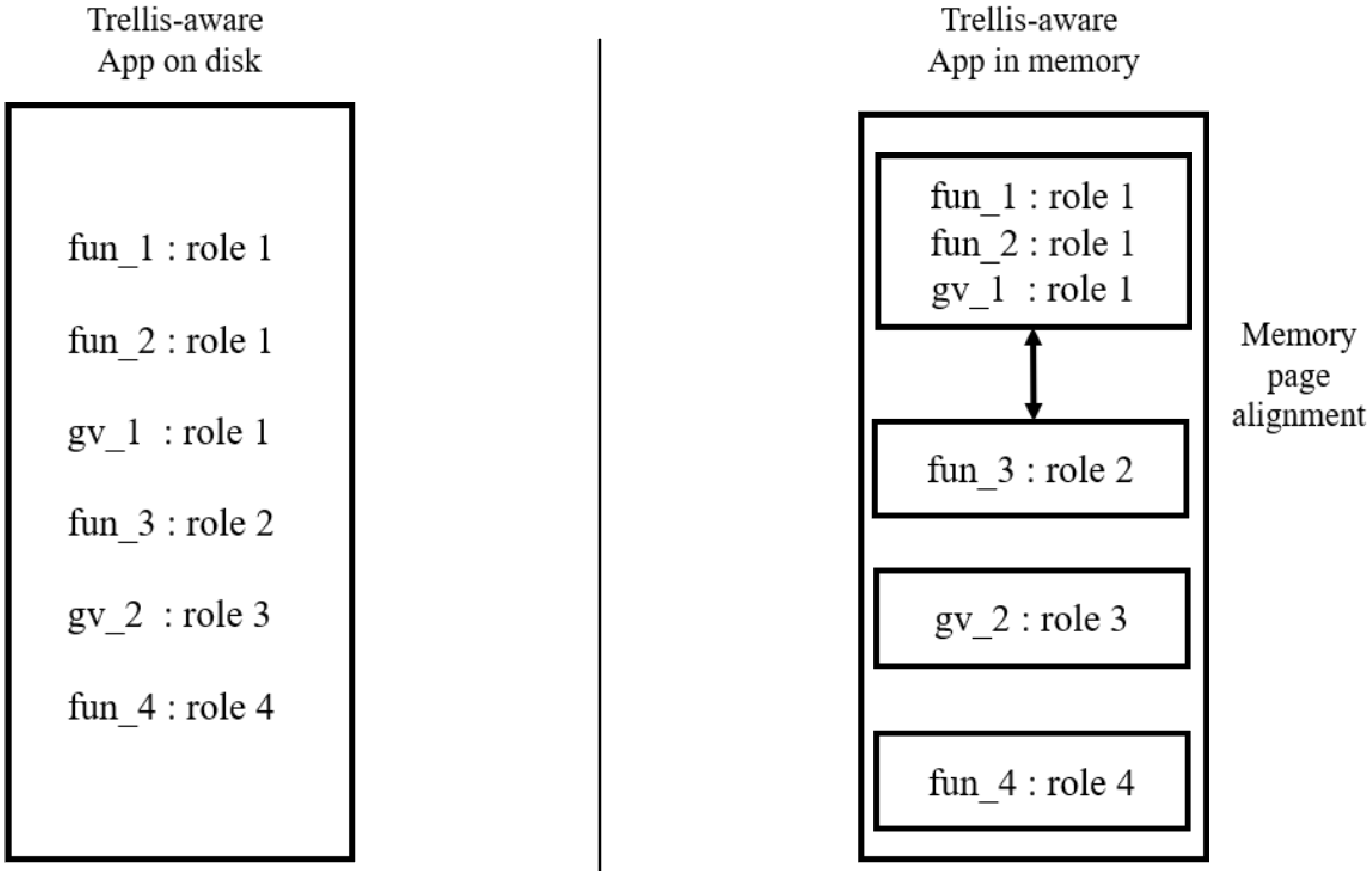
fun_3 : role 2

gv_3 : role 3

fun_4 : role 4

Without Partitioning

Implementation



Trellis-aware
App on disk

fun_1 : role 1

fun_2 : role 1

gv_1 : role 1

fun_3 : role 2

gv_2 : role 3

fun_4 : role 4

Trellis-aware
App in memory

fun_1 : role 1
fun_2 : role 1
gv_1 : role 1

fun_3 : role 2

gv_2 : role 3

fun_4 : role 4

Memory
page
alignment

With Partitioning

Implementation - Runtime

Remove access privileges for memory segments containing non-accessible roles

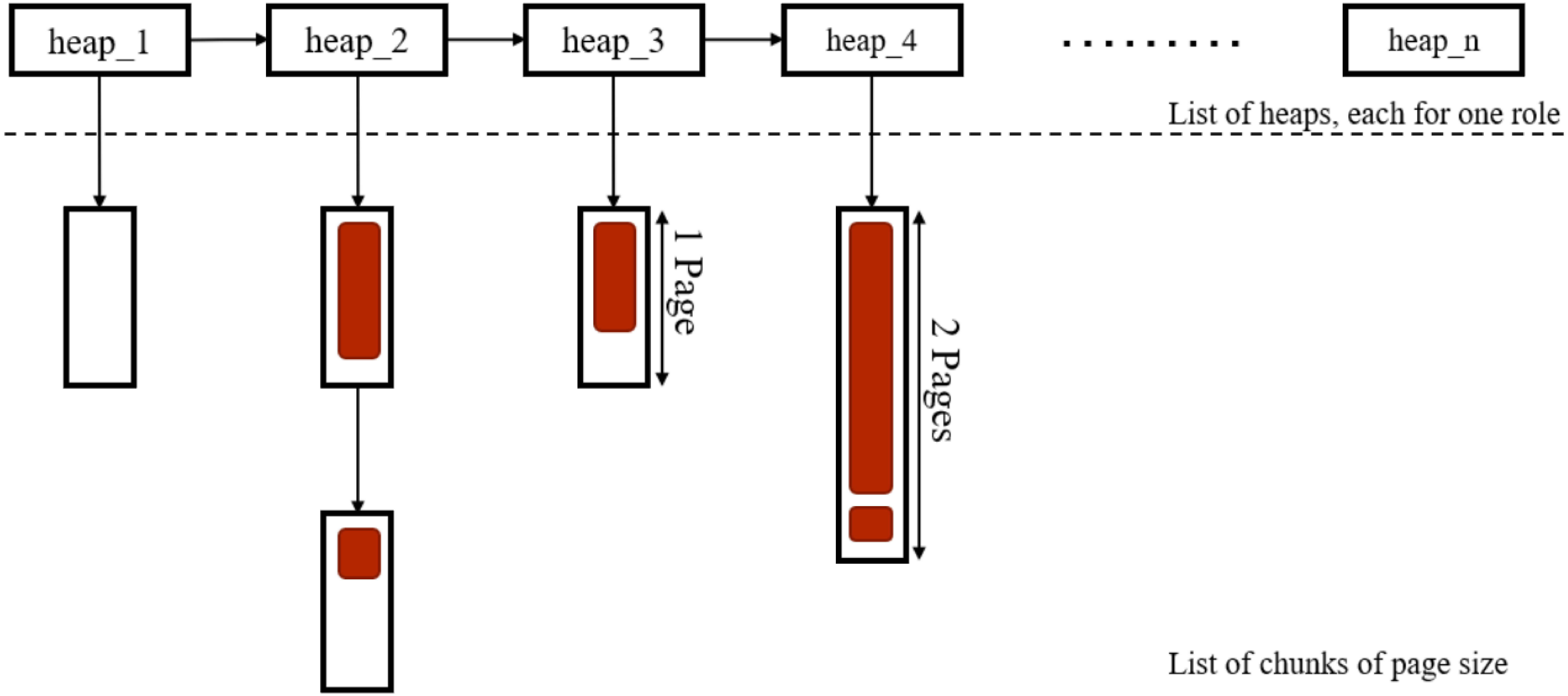
PAM-based Authentication during application role switching

Custom program loader to initialize the metadata in kernel-space

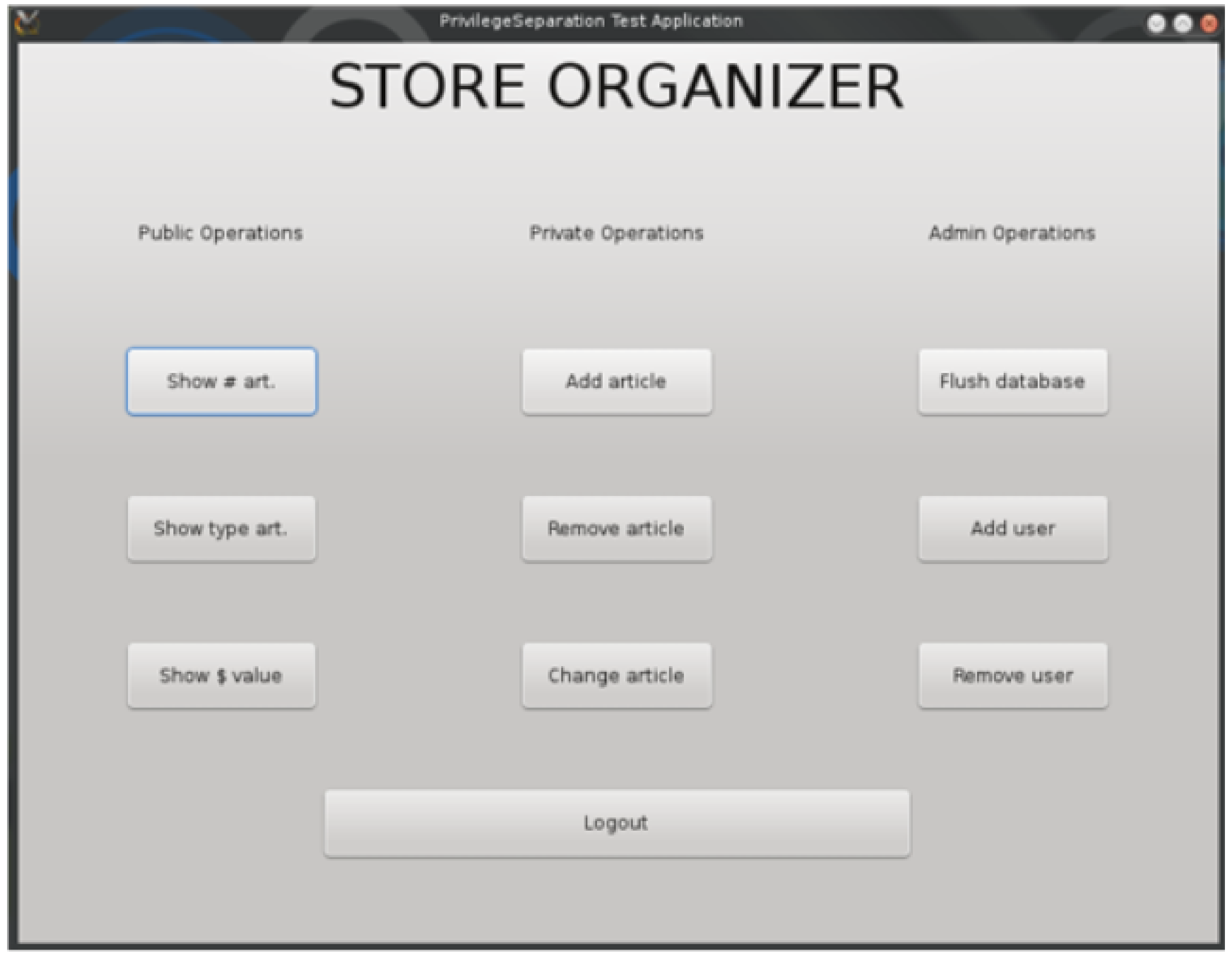
Multi-heap page-based Allocator for protecting dynamic allocated data

Implementation

trellis_alloc(size,role)



Evaluation


PrivilegeSeparation Test Application
STORE ORGANIZER
Public Operations
Private Operations
Admin Operations
Show # art.
Add article
Flush database
Show type art.
Remove article
Add user
Show $ value
Change article
Remove user
Logout

HomeBank
Free, easy, personal accounting for everyone.

Evaluation

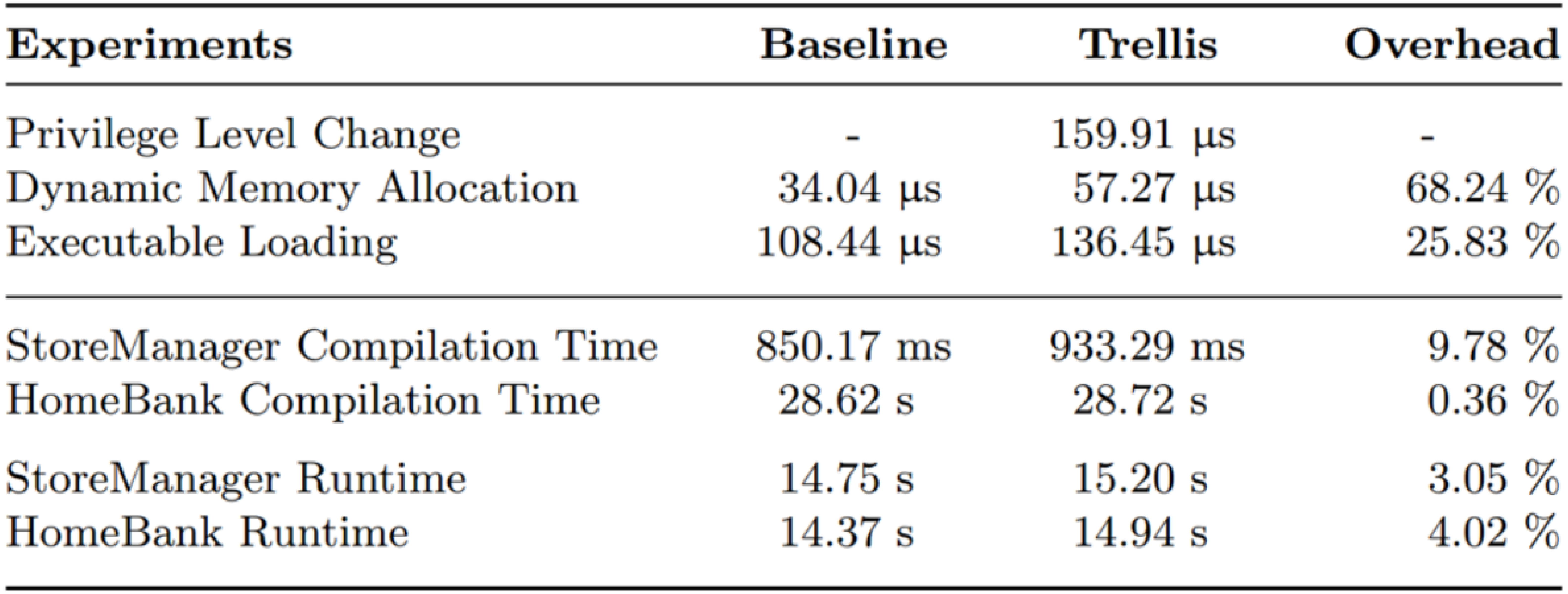
Micro-Benchmarks

End-to-end Performance

Developer Effort

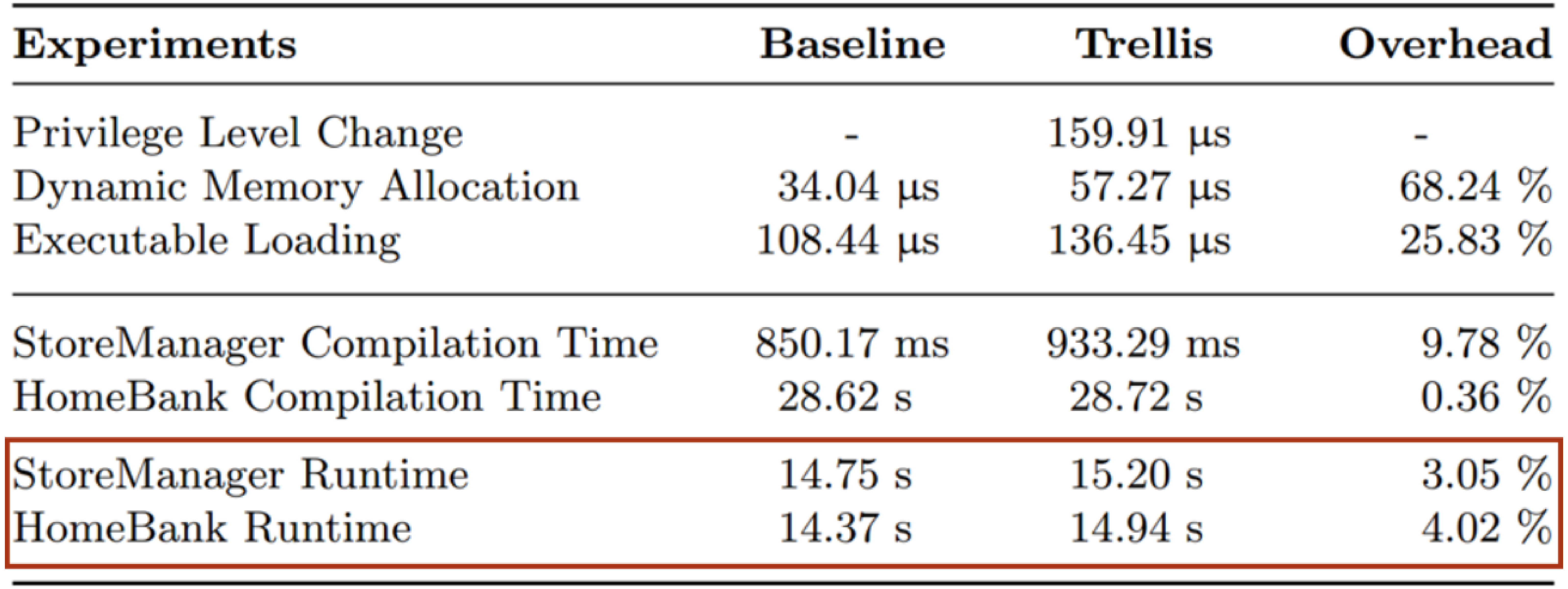
Security Experiments

Evaluation



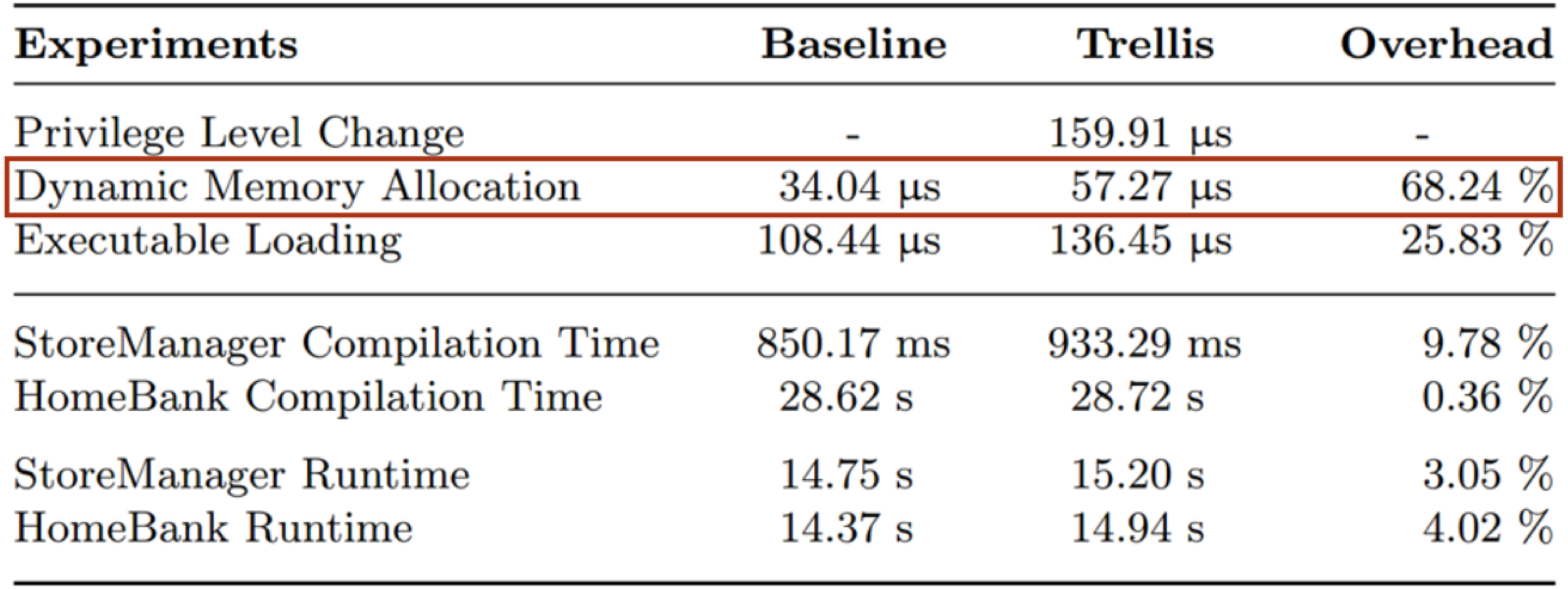
Experiments	**Baseline**	**Trellis**	**Overhead**
Privilege Level Change	-	159.91 µs	-
Dynamic Memory Allocation	34.04 µs	57.27 µs	68.24 %
Executable Loading	108.44 µs	136.45 µs	25.83 %
StoreManager Compilation Time	850.17 ms	933.29 ms	9.78 %
HomeBank Compilation Time	28.62 s	28.72 s	0.36 %
StoreManager Runtime	14.75 s	15.20 s	3.05 %
HomeBank Runtime	14.37 s	14.94 s	4.02 %

Evaluation



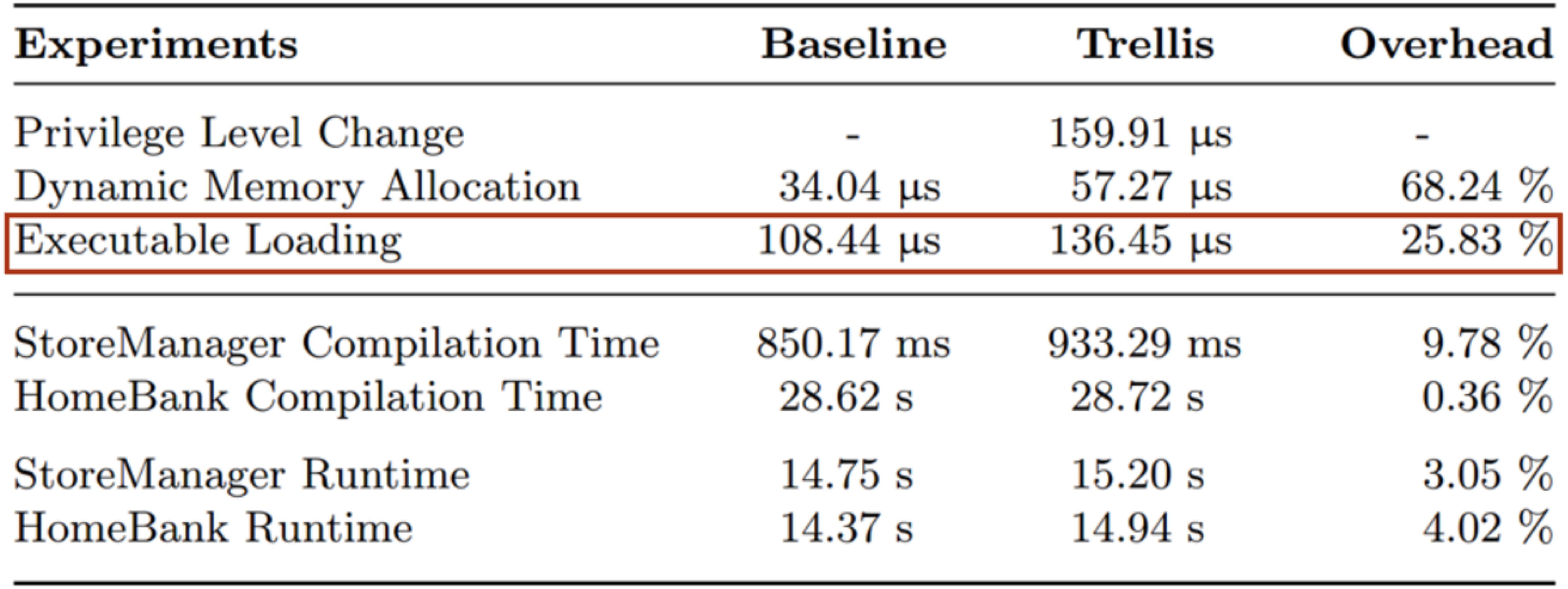
Experiments	Baseline	Trellis	Overhead
Privilege Level Change	-	159.91 µs	-
Dynamic Memory Allocation	34.04 µs	57.27 µs	68.24 %
Executable Loading	108.44 µs	136.45 µs	25.83 %
StoreManager Compilation Time	850.17 ms	933.29 ms	9.78 %
HomeBank Compilation Time	28.62 s	28.72 s	0.36 %
StoreManager Runtime	14.75 s	15.20 s	3.05 %
HomeBank Runtime	14.37 s	14.94 s	4.02 %

Evaluation



Experiments	Baseline	Trellis	Overhead
Privilege Level Change	-	159.91 µs	-
Dynamic Memory Allocation	34.04 µs	57.27 µs	68.24 %
Executable Loading	108.44 µs	136.45 µs	25.83 %
StoreManager Compilation Time	850.17 ms	933.29 ms	9.78 %
HomeBank Compilation Time	28.62 s	28.72 s	0.36 %
StoreManager Runtime	14.75 s	15.20 s	3.05 %
HomeBank Runtime	14.37 s	14.94 s	4.02 %

Evaluation



Experiments	Baseline	Trellis	Overhead
Privilege Level Change	-	159.91 µs	-
Dynamic Memory Allocation	34.04 µs	57.27 µs	68.24 %
Executable Loading	108.44 µs	136.45 µs	25.83 %
StoreManager Compilation Time	850.17 ms	933.29 ms	9.78 %
HomeBank Compilation Time	28.62 s	28.72 s	0.36 %
StoreManager Runtime	14.75 s	15.20 s	3.05 %
HomeBank Runtime	14.37 s	14.94 s	4.02 %

Discussion and Future works

Binary privilege separation

Just-in-time compiler support

Indirect function call analysis integration

Data declassification

Conclusions

OS-based application development framework for multi-user application to enforce Hierarchical-Base Access Control policies (HBAC)

Prototype implementation based on LLVM/Clang, GNU C Library, and the Linux Kernel

Real-world application evaluation

Trellis is now open-source and available at https://github.com/m4mbr3/trellis.git

Questions?