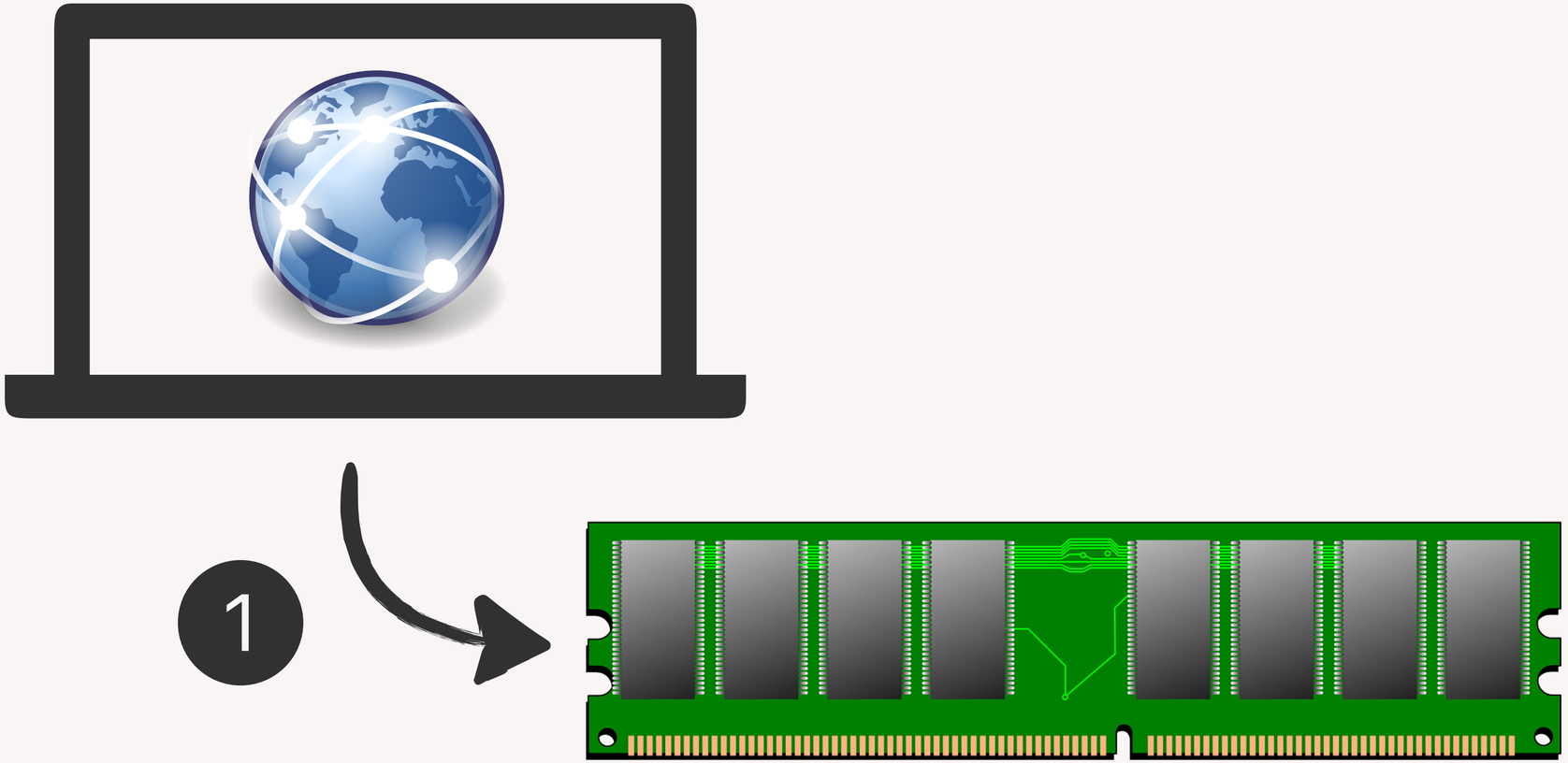
# *SMASH*

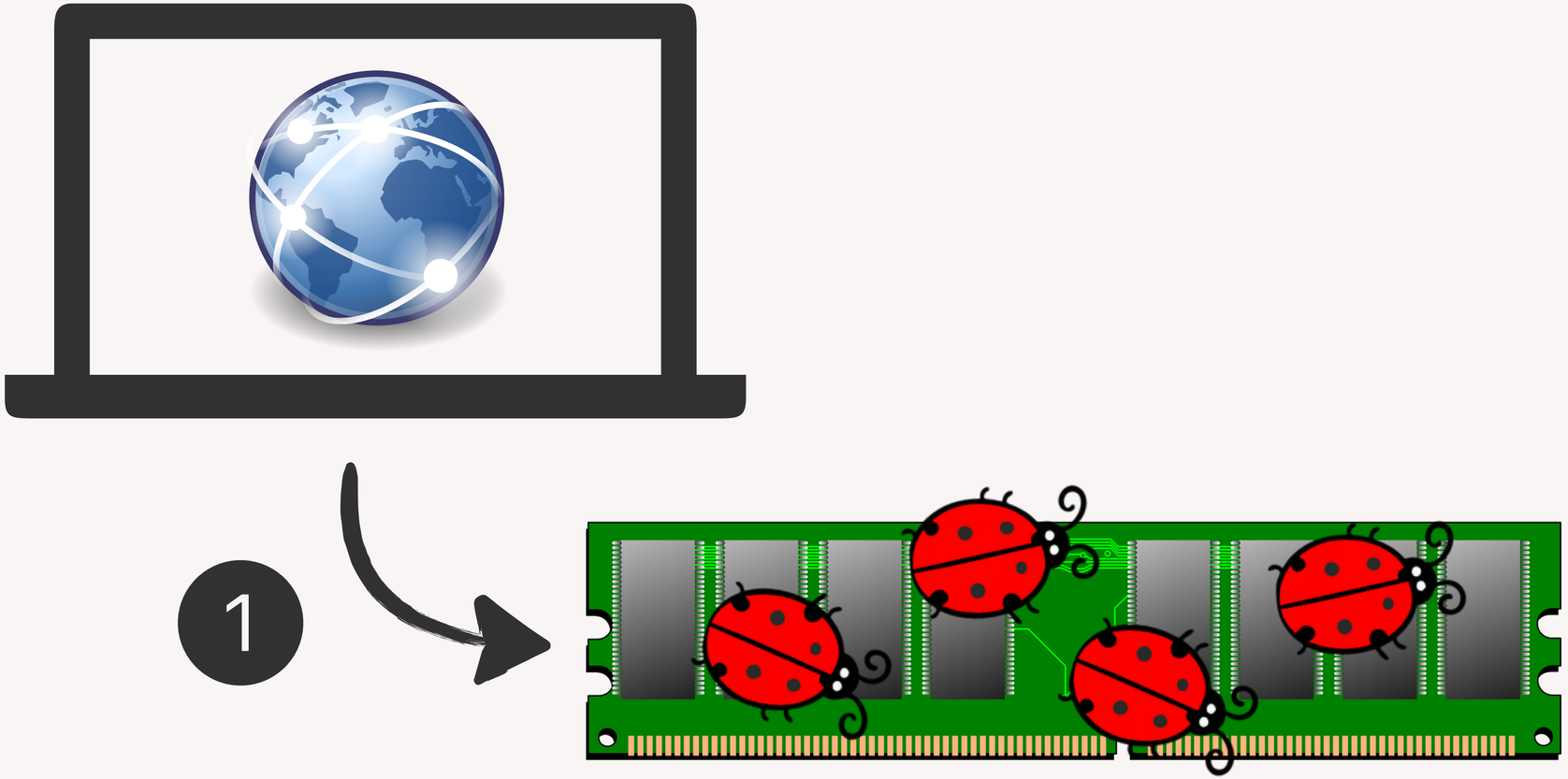
## Synchronized Many-sided Rowhammer Attacks from JavaScript

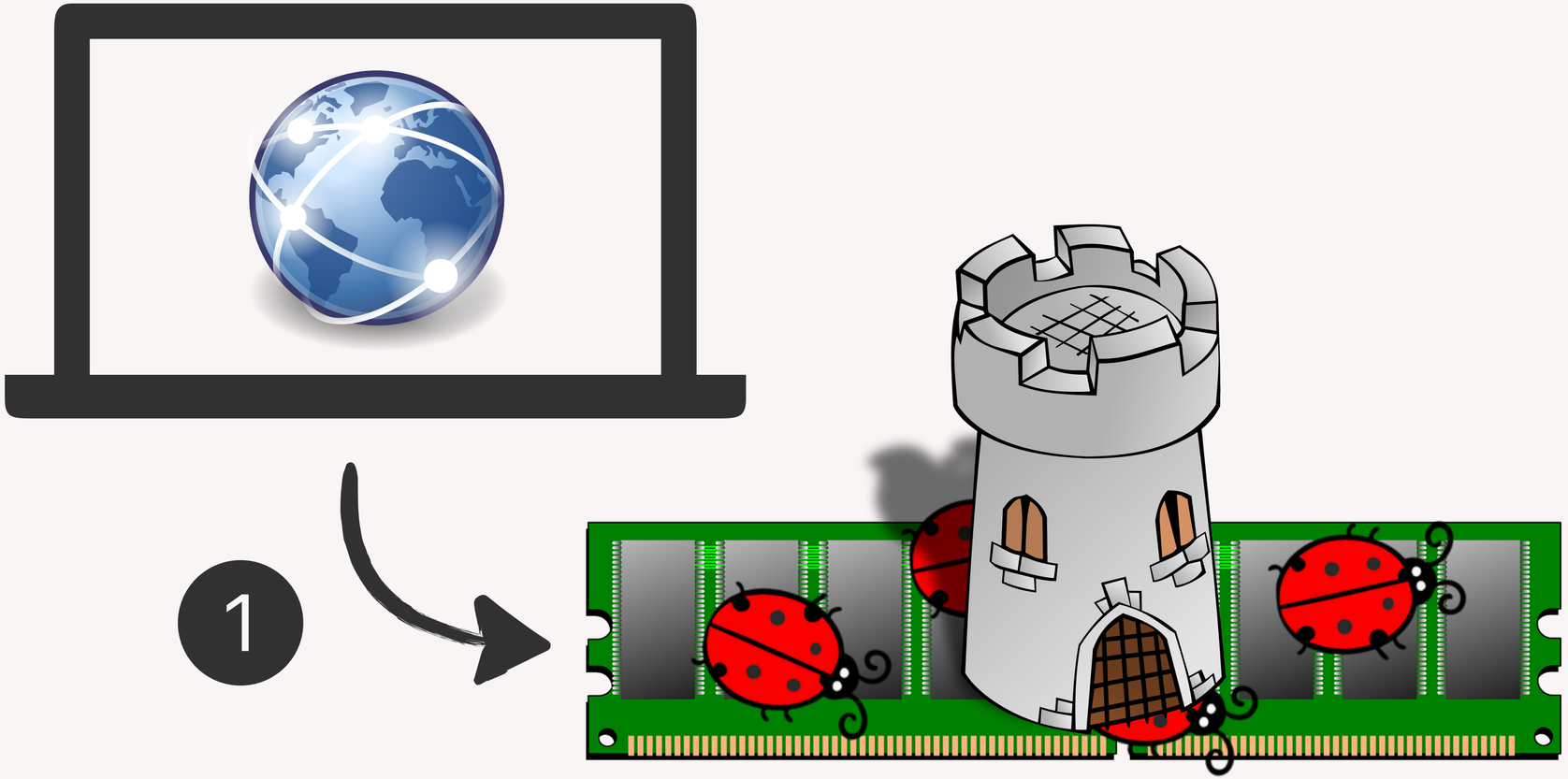
**Demo on YouTube and source code on GitHub**

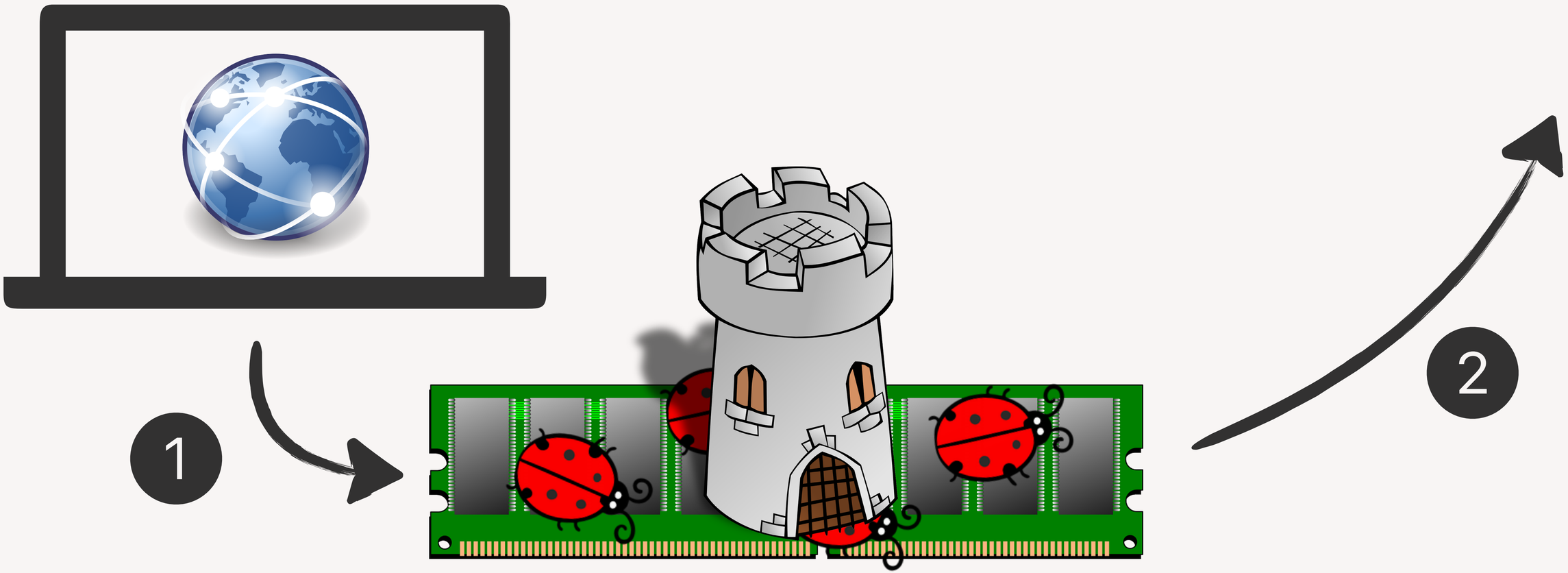
**Finn de Ridder · Pietro Frigo · Emanuele Vannacci · Herbert Bos · Cristiano Giuffrida · Kaveh Razavi**
**ETH Zurich and VU Amsterdam**

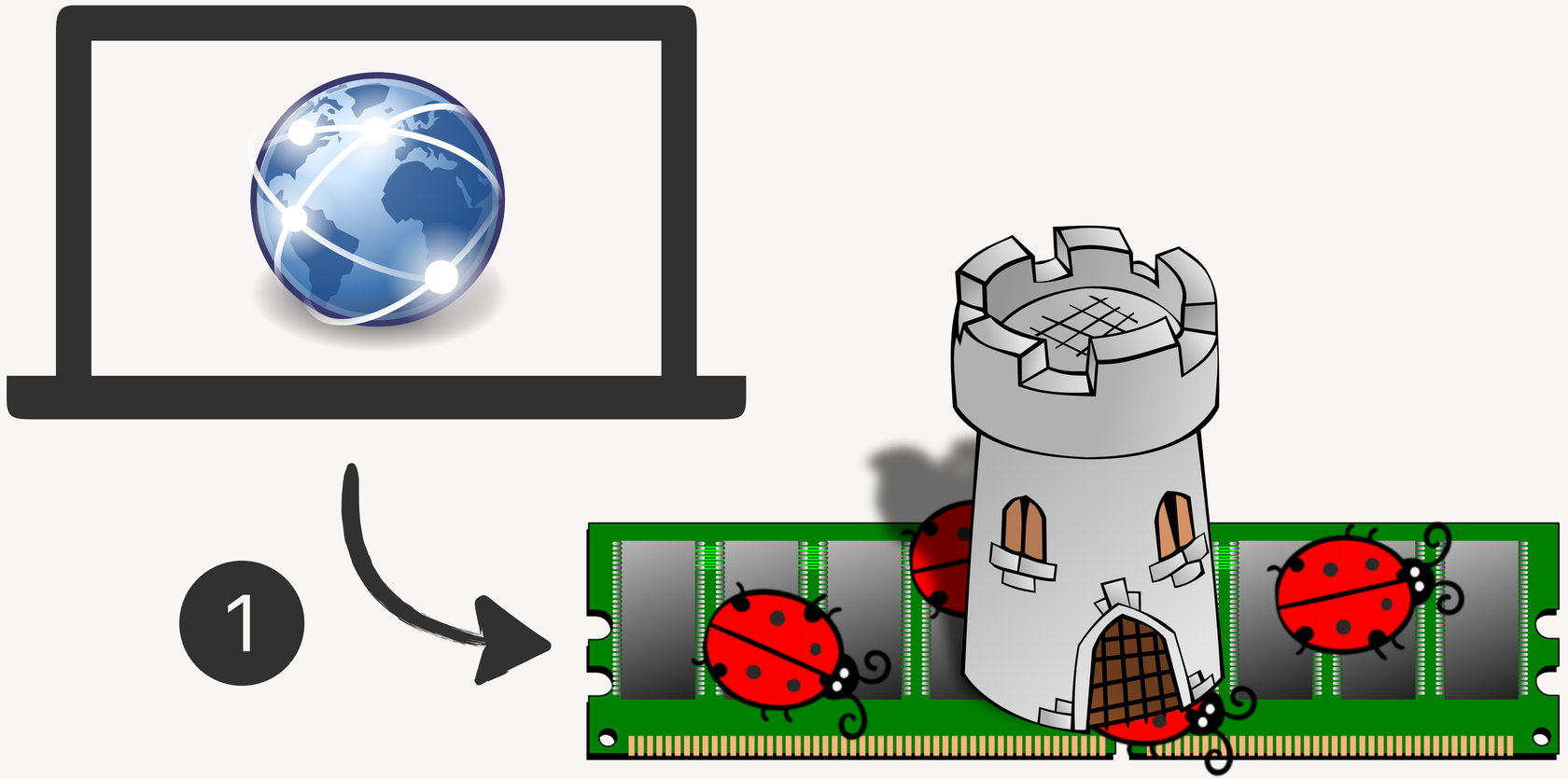

1


1


1


1

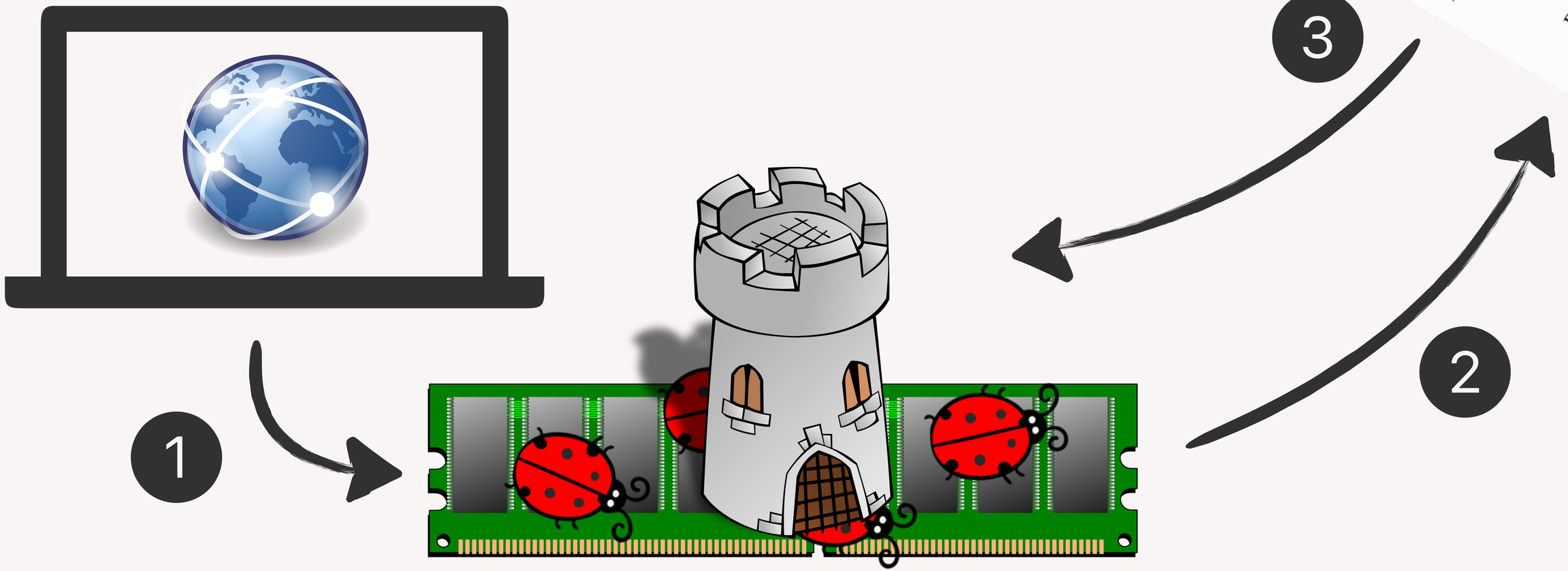

1
2




2020 IEEE Symposium

TRRespass: Exploiting
Target Ro

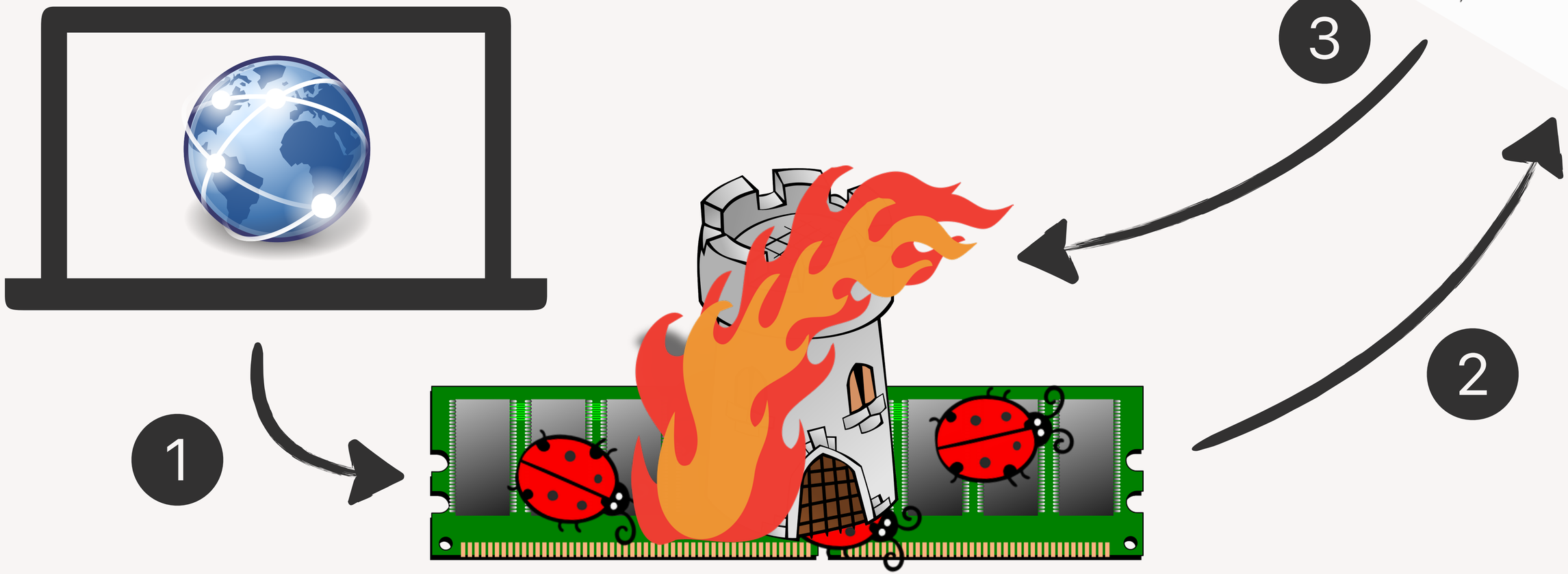
Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit




2020 IEEE Symposium

TRRespass: Exploiting
Target Ro

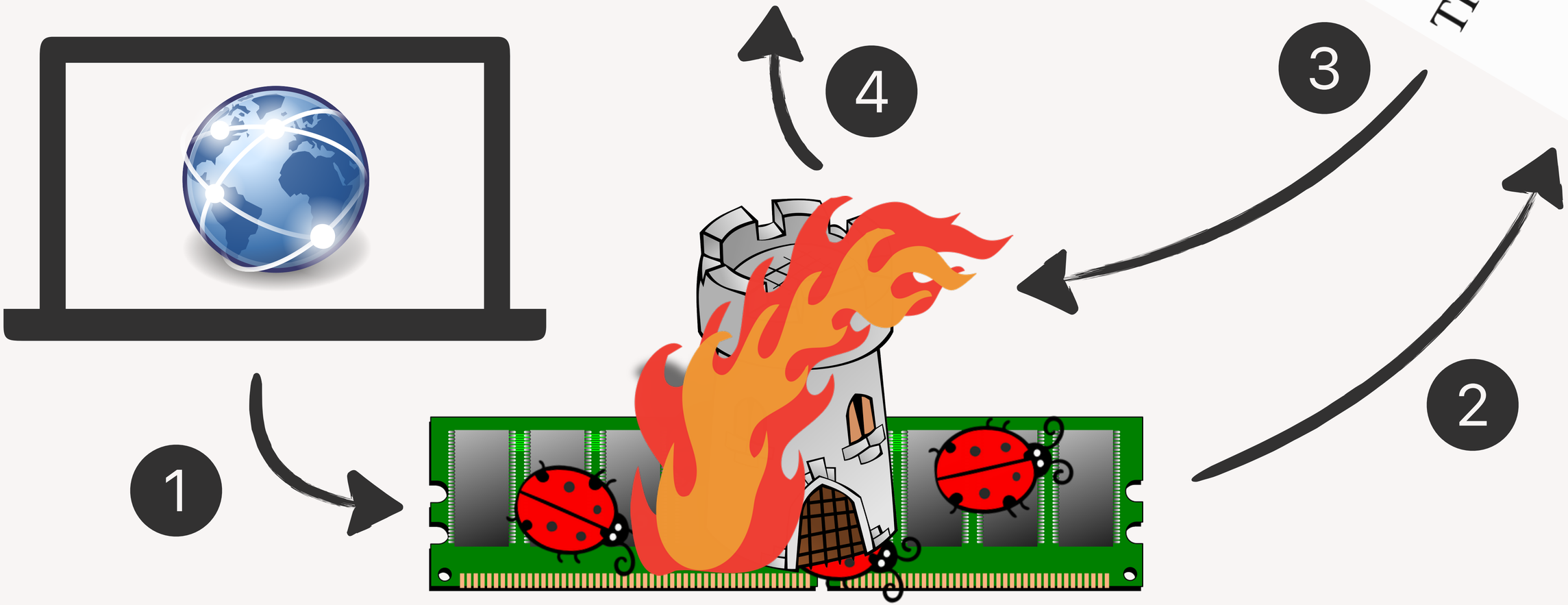
Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit




2020 IEEE Symposi

TRRespass: Exploiting
Target Ro

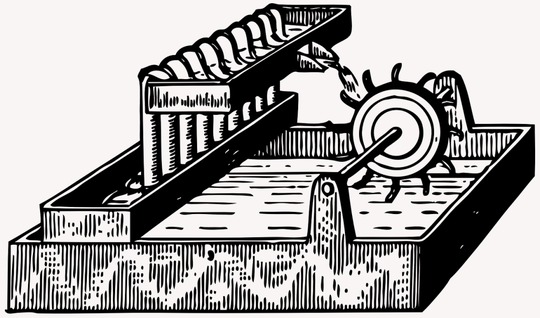
Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit




2020 IEEE Symposi

TRRespass: Exploiting
Target Ro

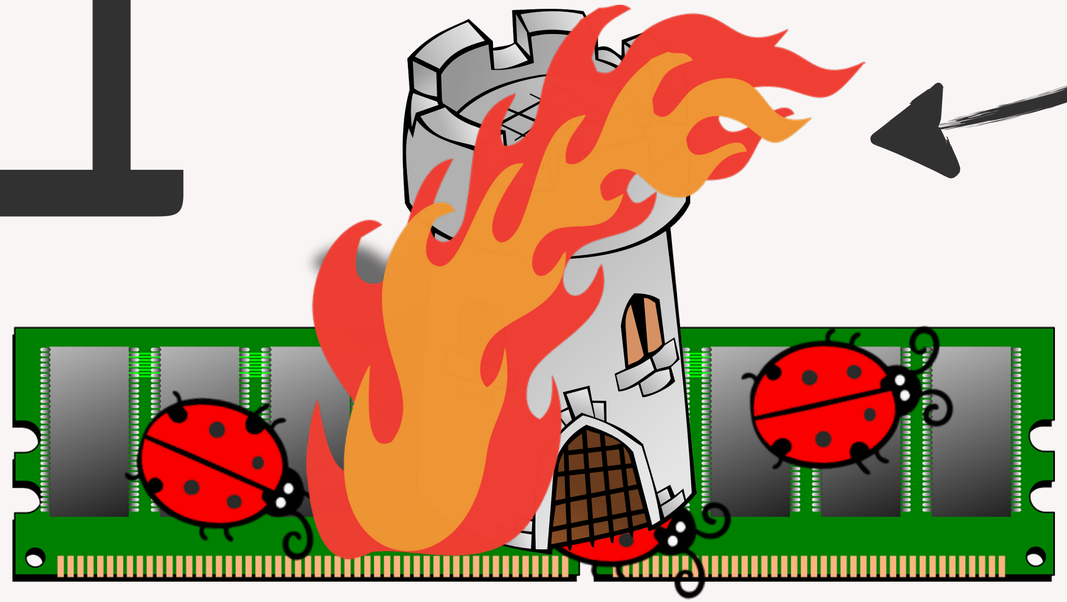
Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit



2020 IEEE Sympos

TRRespass: Exploiting
Target Ro

Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit

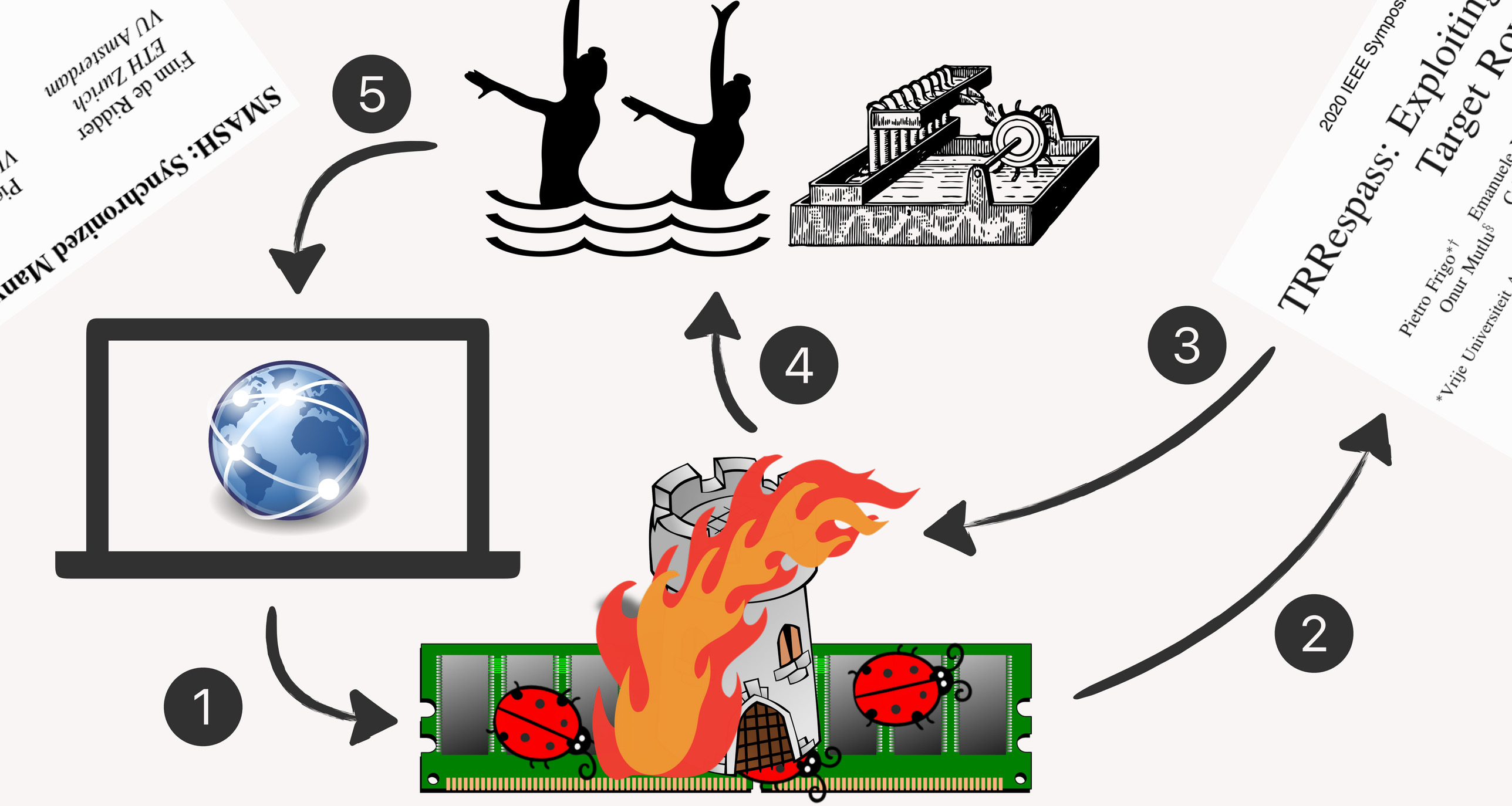


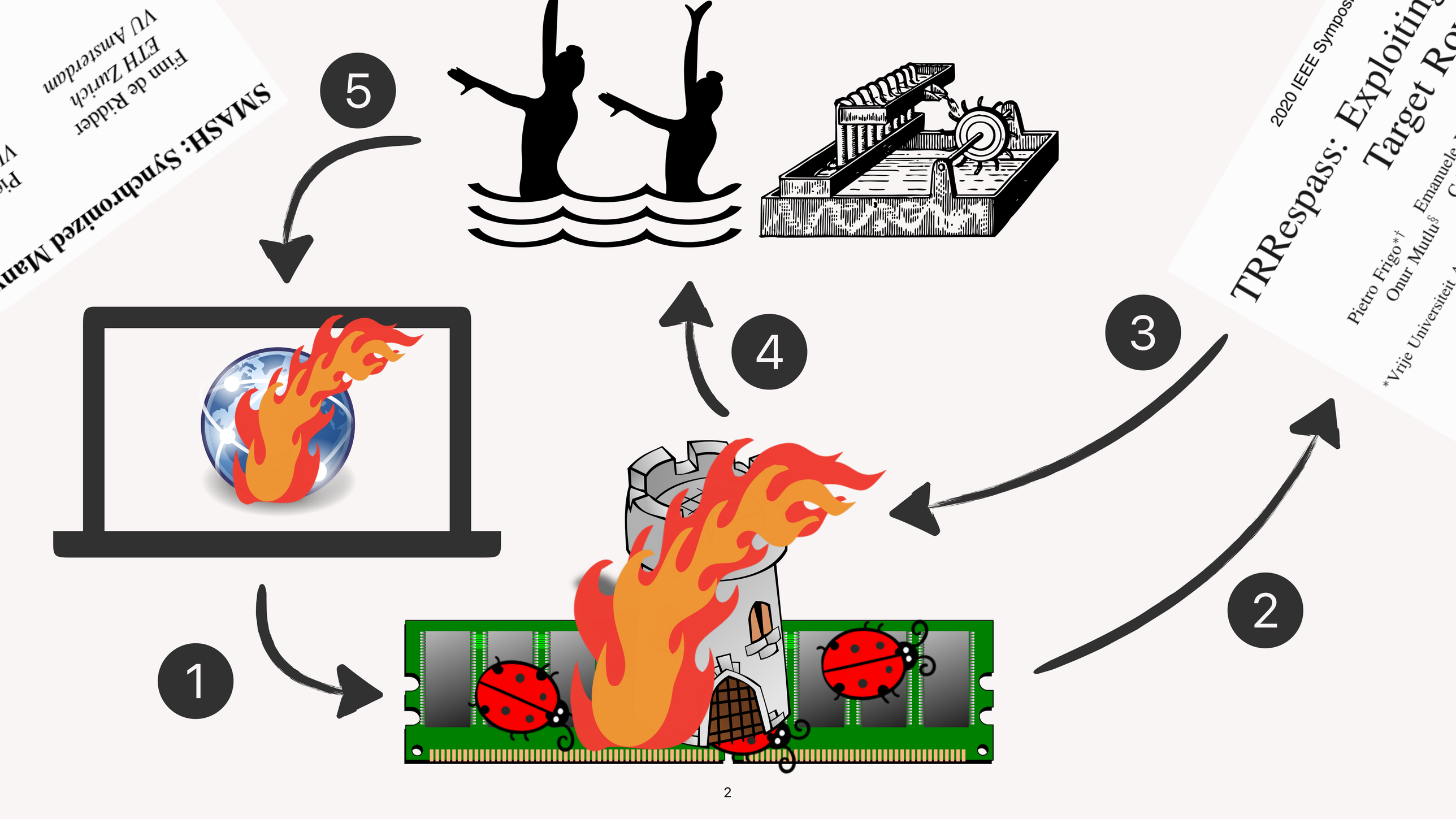
1

2

3

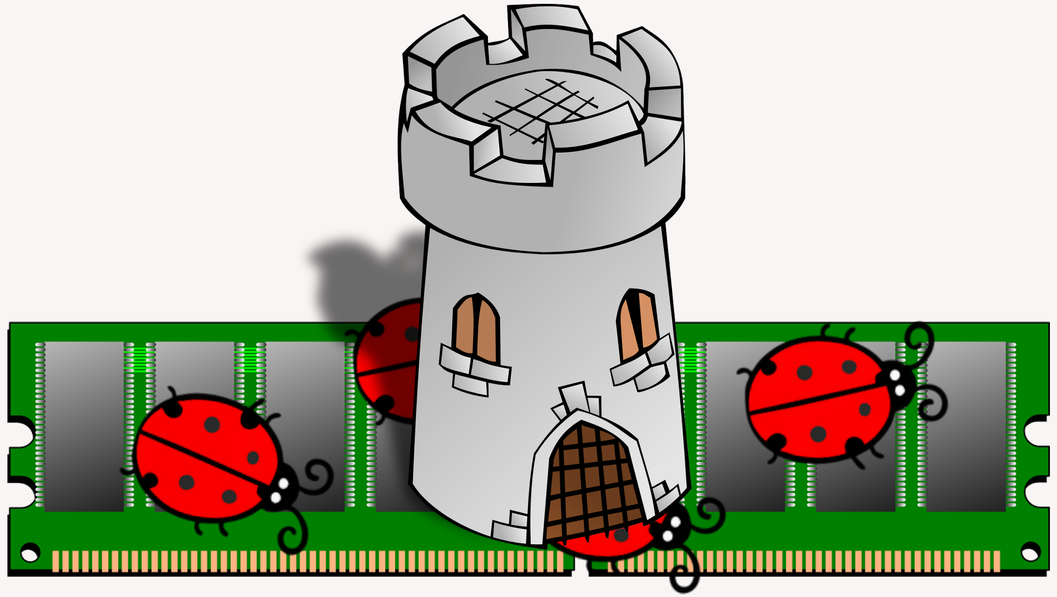
4


SMASH: Synchronized Man
Finn de Ridder
ETH Zurich
VU Amsterdam
2020 IEEE Symposi
TRRespass: Exploiting
Target Ro
Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit
5
4
3
2
1


5
4
3
2
1

# Rowhammer Bugs and TRR

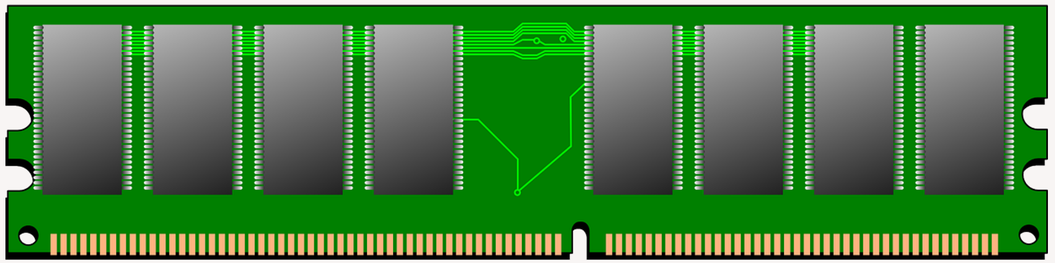
## Part I



# Rowhammer

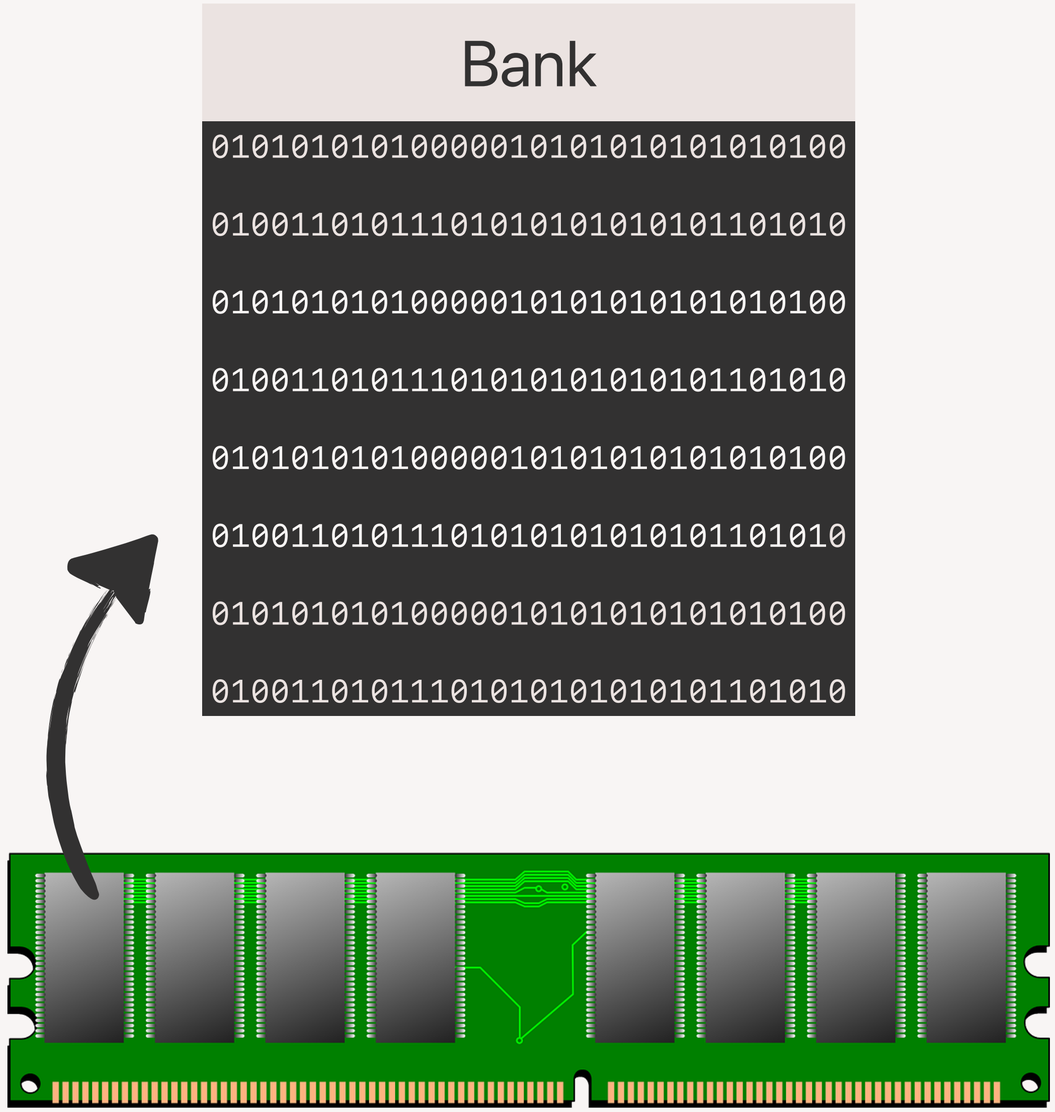
## Bugs

| Bank |
| --- |
| 01010101010000010101010101010100 |
| 01001101011101010101010101101010 |
| 01010101010000010101010101010100 |
| 01001101011101010101010101101010 |
| 01010101010000010101010101010100 |
| 01001101011101010101010101101010 |
| 01010101010000010101010101010100 |
| 01001101011101010101010101101010 |



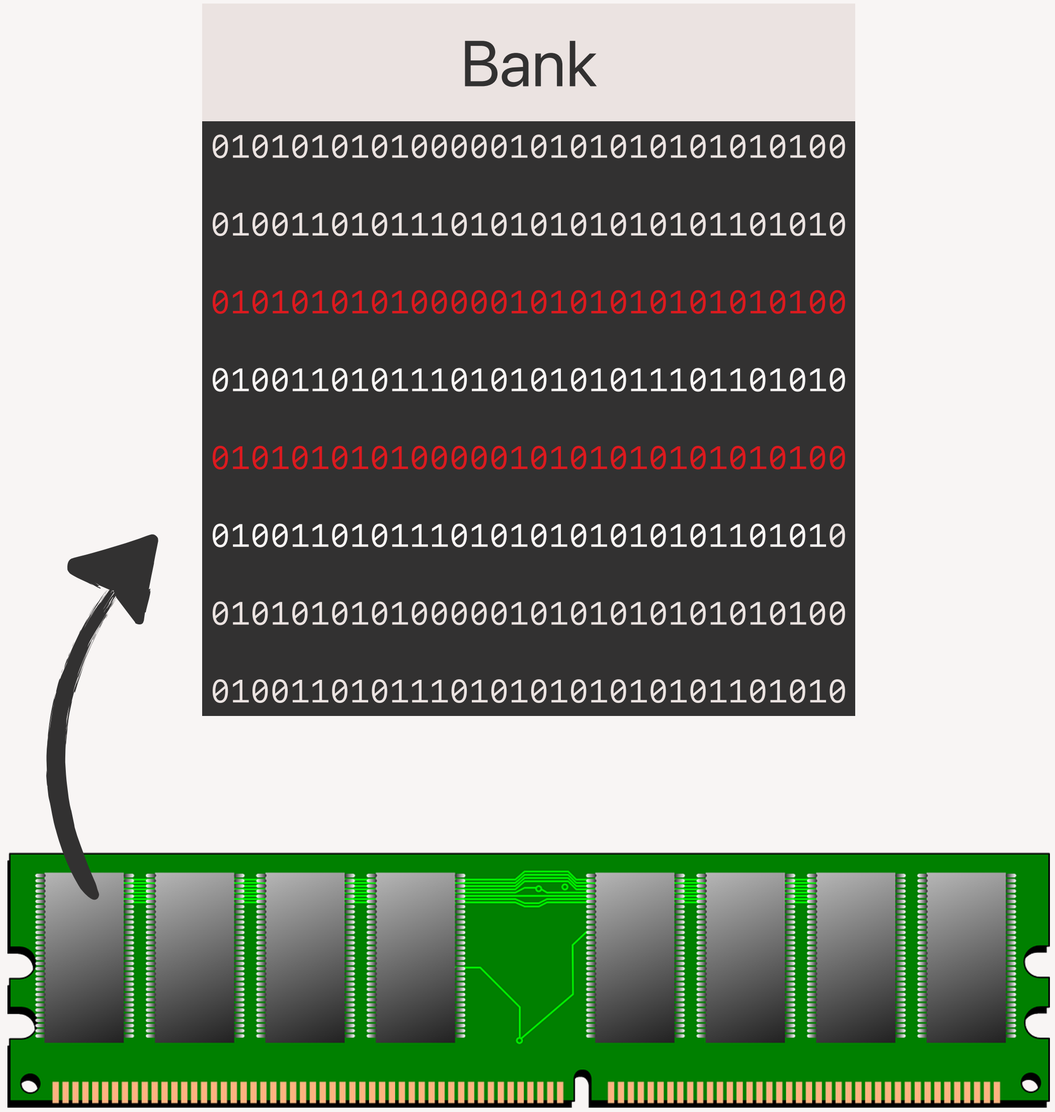
# Rowhammer

## Bugs



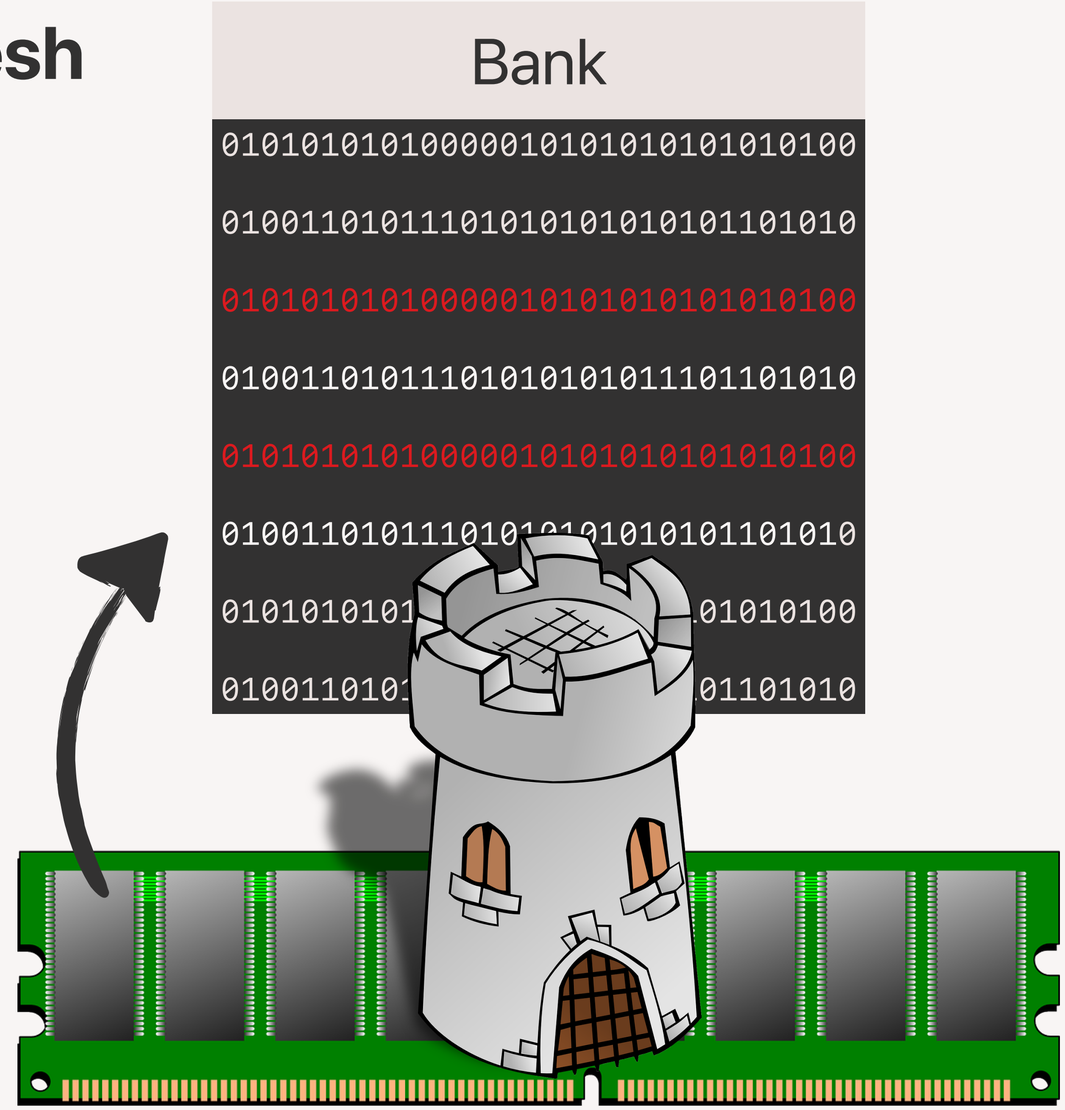
# Rowhammer

## Bugs



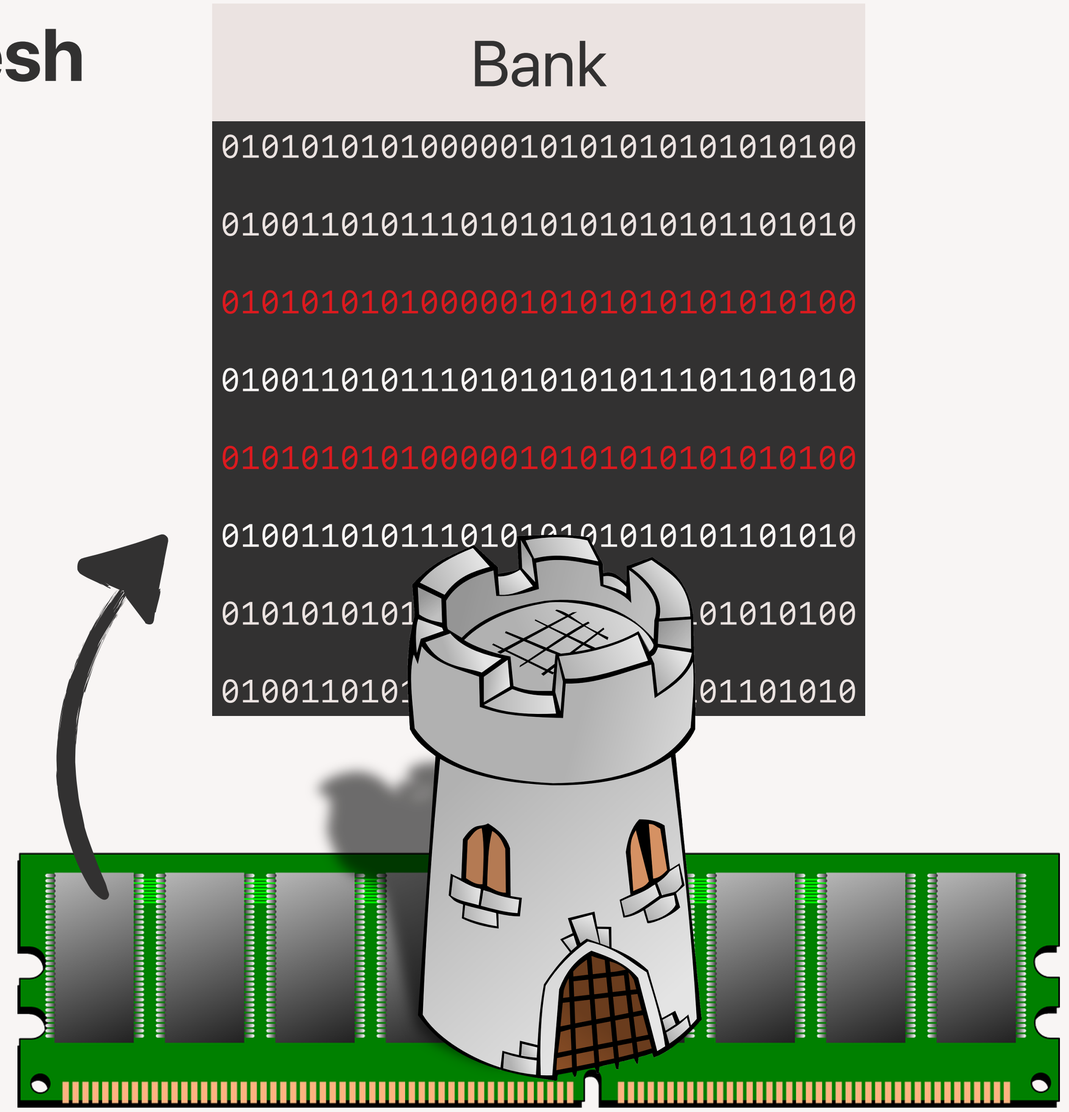
# TRR

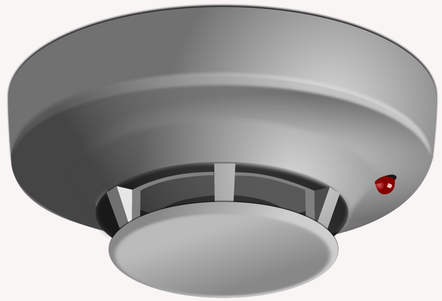
## Target Row Refresh



# TRR

## Target Row Refresh



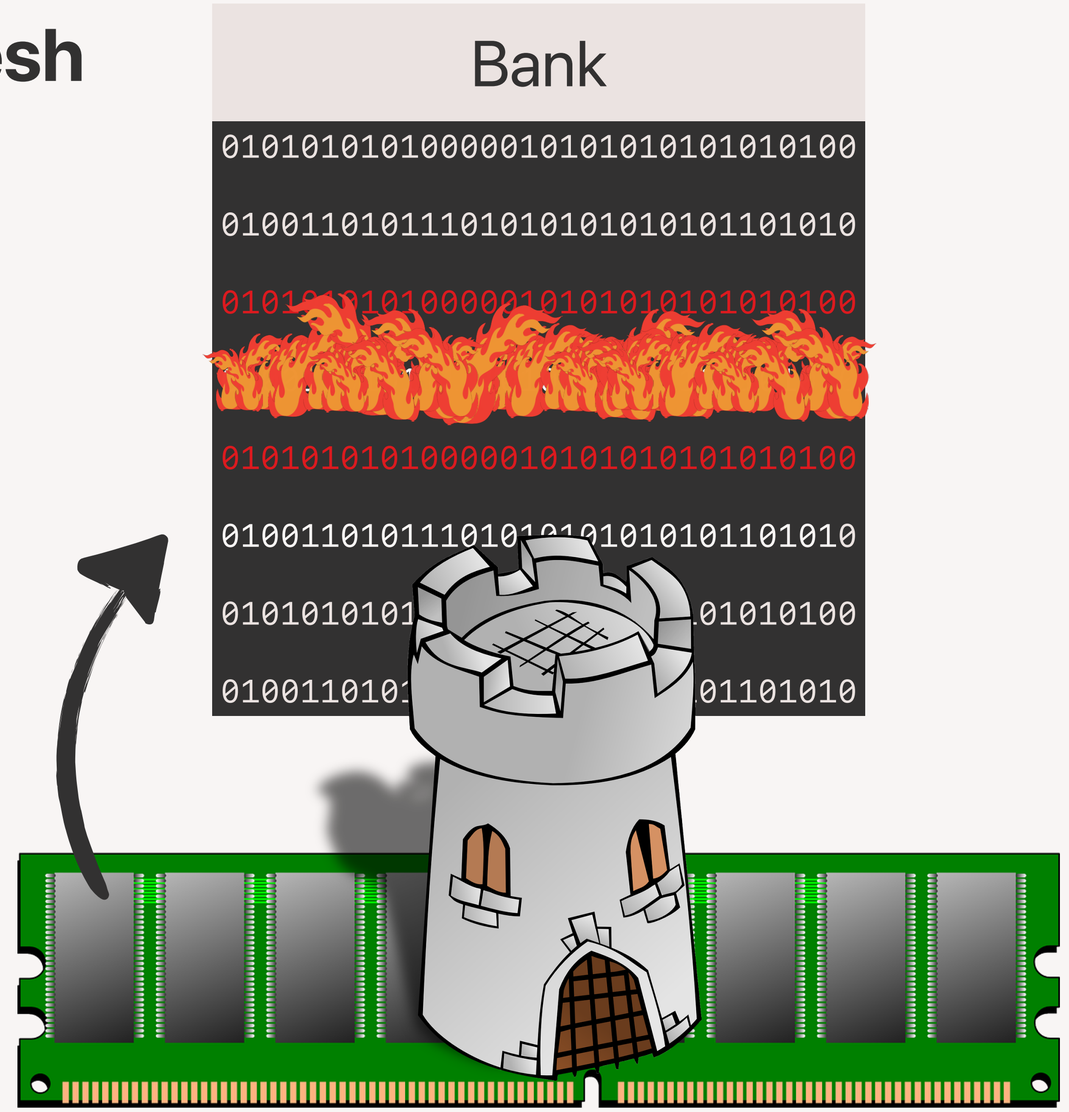



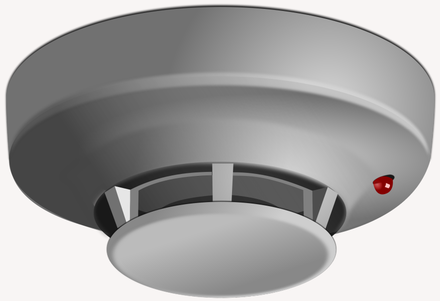
Sampler


Inhibitor

# TRR

## Target Row Refresh



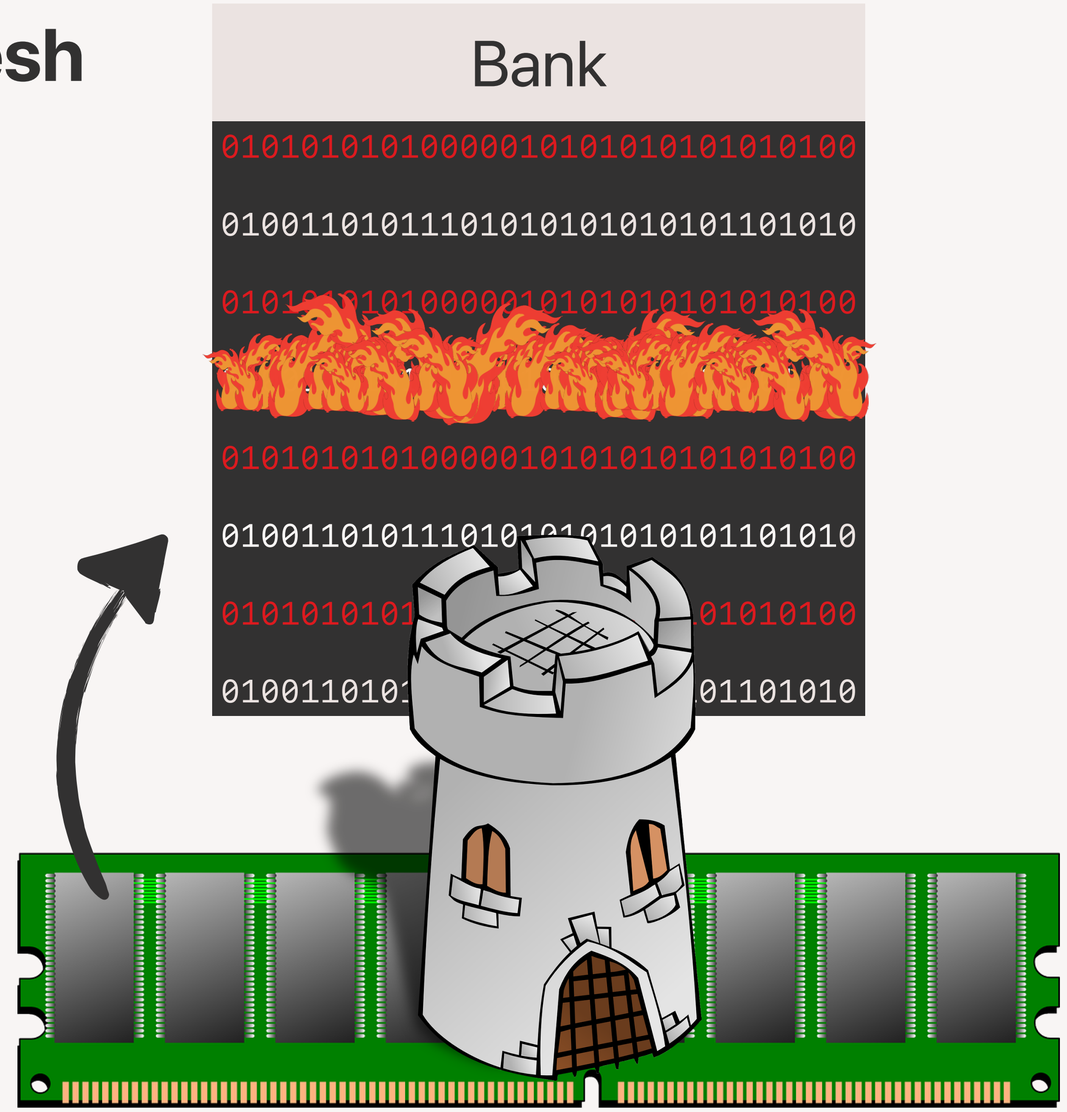



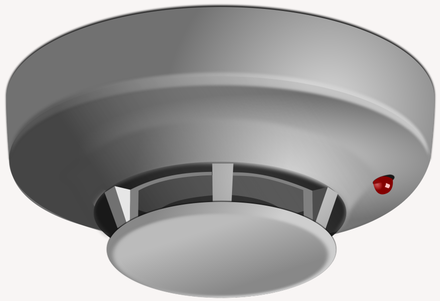
Sampler


Inhibitor

# TRR

## Target Row Refresh



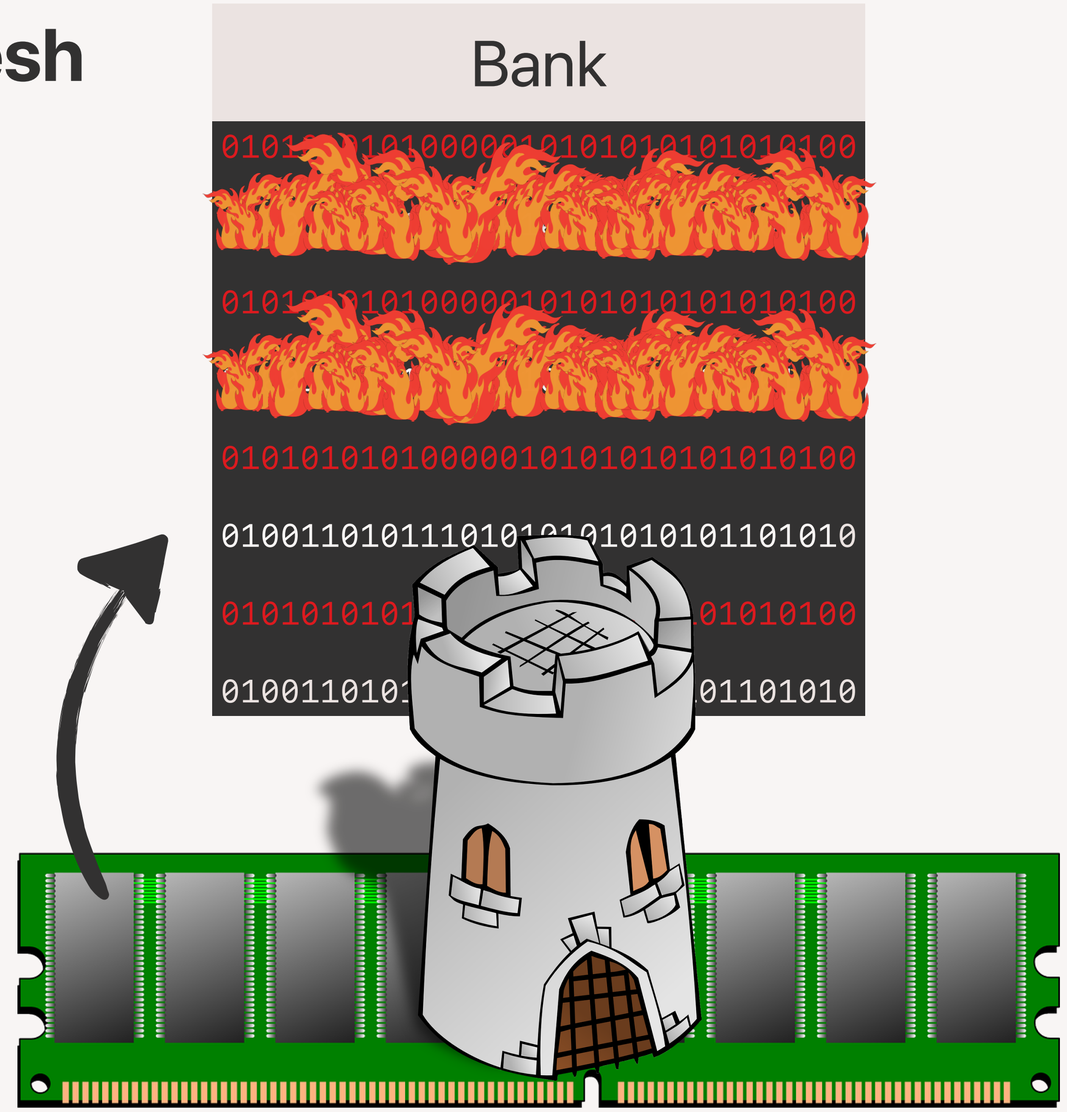



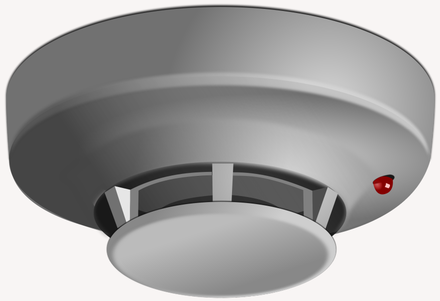
Sampler


Inhibitor

# TRR

## Target Row Refresh



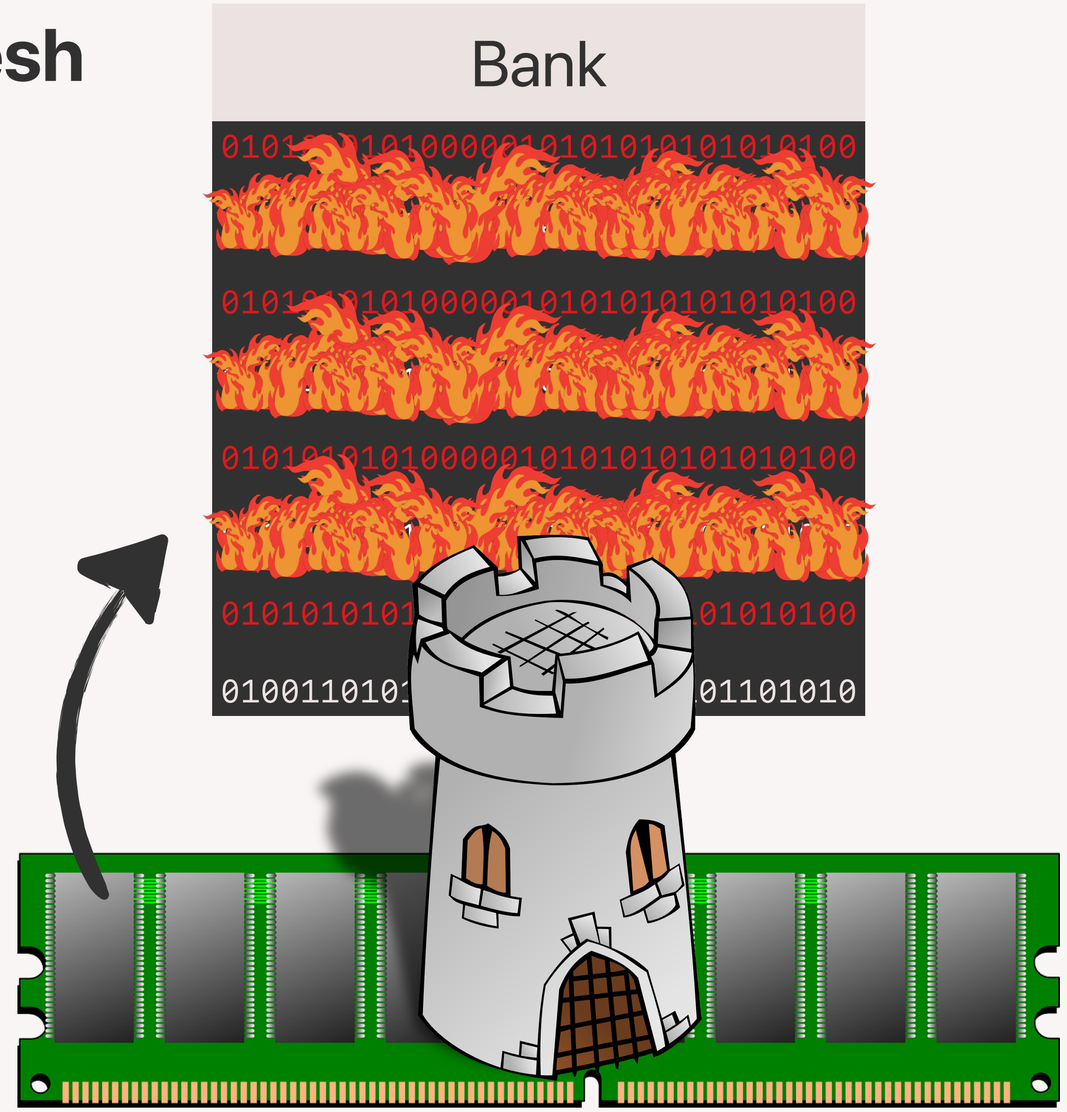



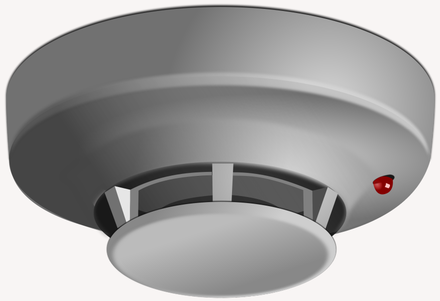
Sampler


Inhibitor

# TRR

## Target Row Refresh



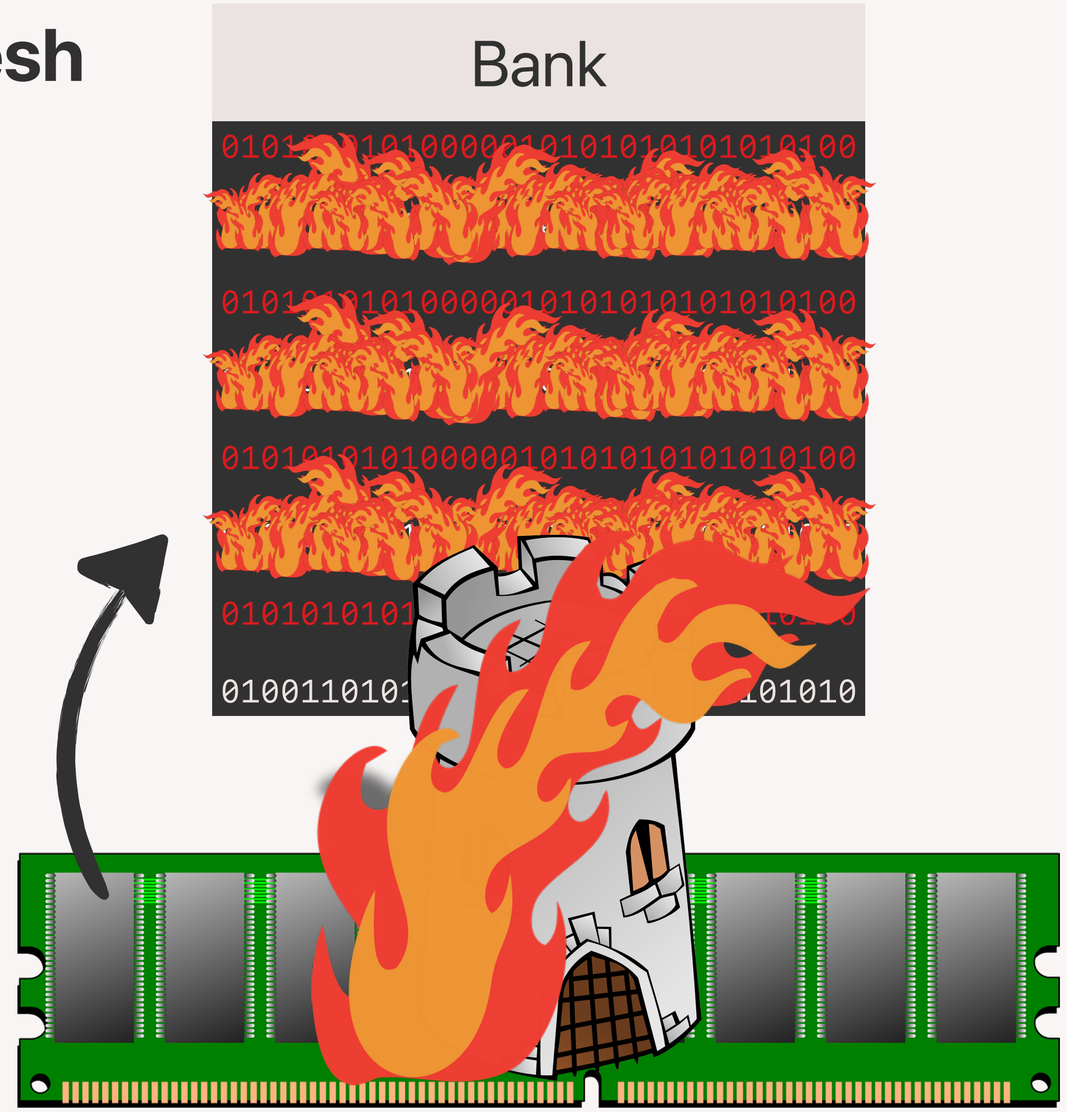



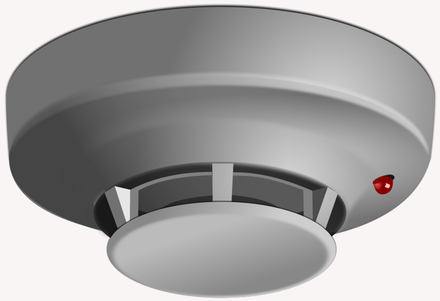
Sampler


Inhibitor

# TRR

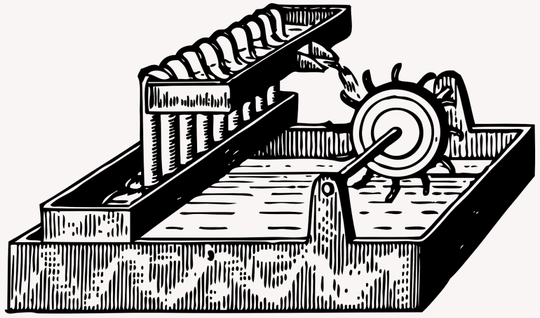
## Target Row Refresh

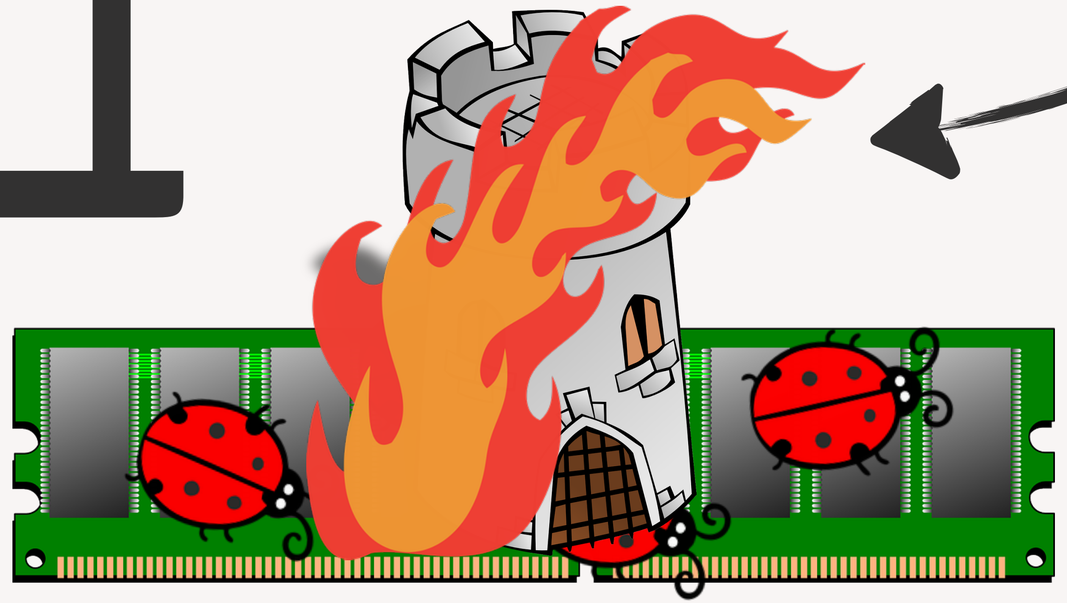


Sampler


Inhibitor





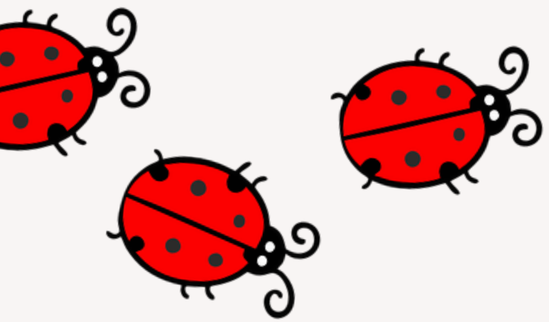
2020 IEEE Symposi

TRRespass: Exploiting
Target Ro

Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit

# Rowhammer From JavaScript

## Part II



# Rowhammer From JavaScript

## Complicated

Virtual address space

Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```

# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical address space

Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```

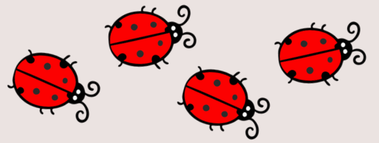
# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical address space

"DRAM" address space



Bank

```
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
```

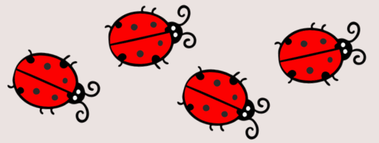
# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical address space

“DRAM” address space



Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```

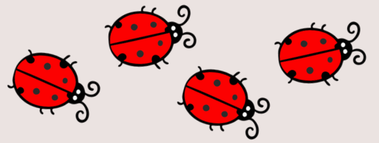
# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical address space

“DRAM” address space



Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```

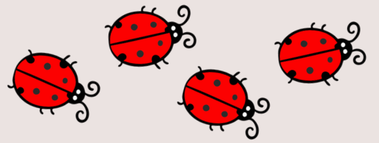
# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical address space

"DRAM" address space



Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```

Caches

# Many-sided to Many Double-sided Rowhammer

## Removing constraints

| Bank |
| --- |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |

# Many-sided to Many Double-sided Rowhammer

## Removing constraints

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
```

**4-sided**
Requires a large chunk

# Many-sided to Many Double-sided Rowhammer

## Removing constraints

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
```

**4-sided**
Requires a large chunk

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
```

# Many-sided to Many Double-sided Rowhammer

## Removing constraints

Bank


**4-sided**
Requires a large chunk

**2 × 2-sided**
Requires several
“smaller” chunks

Bank

# Many-sided to Many Double-sided Rowhammer

## Removing constraints

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
```

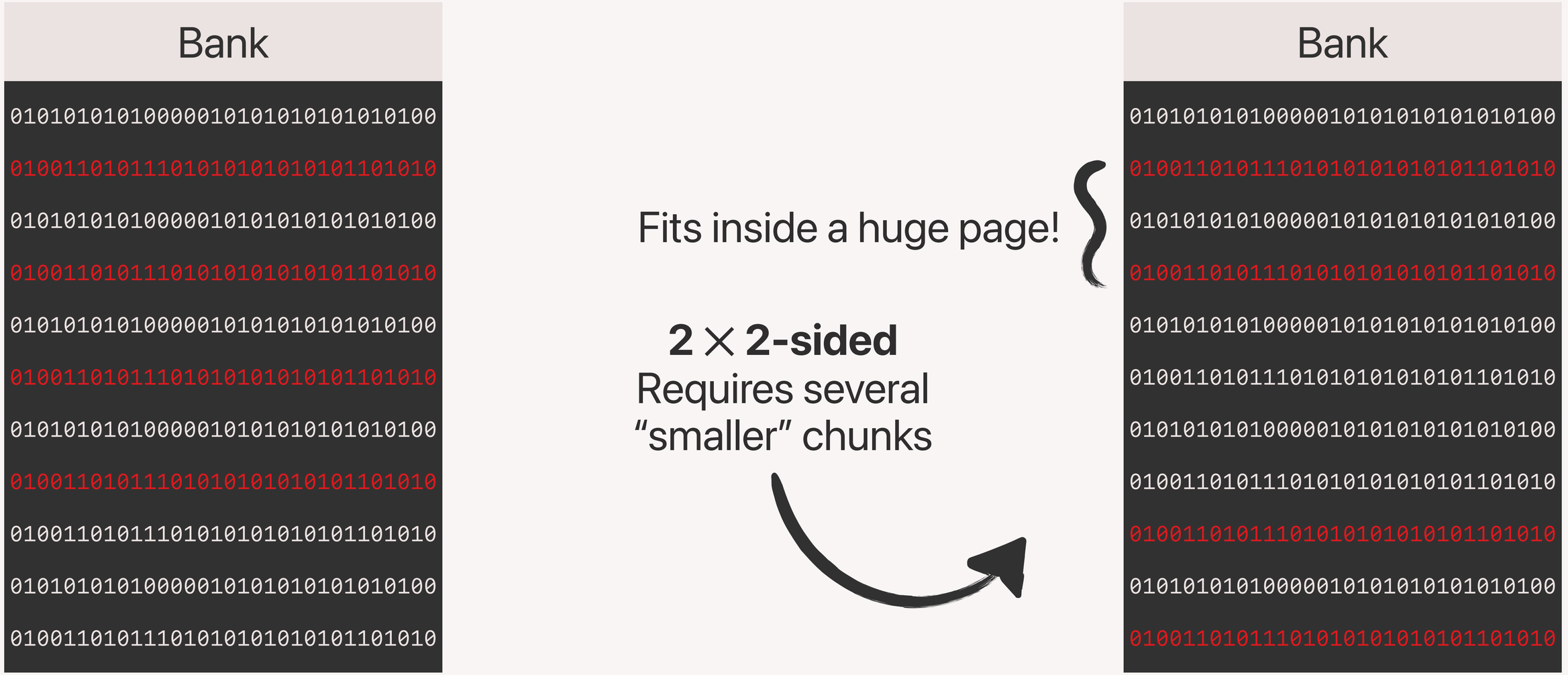
**2 × 2-sided**
Requires several
“smaller” chunks

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
```

# Many-sided to Many Double-sided Rowhammer

## Removing constraints



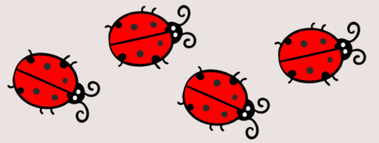
# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical address space

"DRAM" address space



Bank

```
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
```

Caches

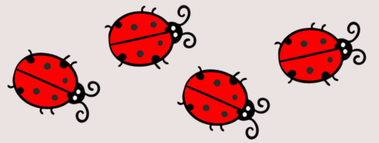
# Rowhammer From JavaScript

## Complicated

Virtual address space

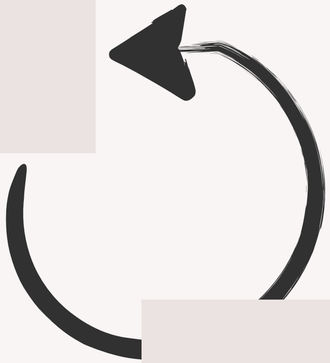
Physical address space

“DRAM” address space



Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```



Caches

# Rowhammer From JavaScript

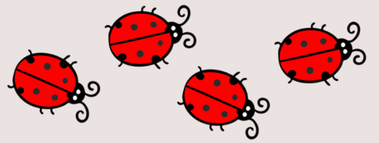
## Complicated

Virtual address space

Physical space

Huge pages

"DRAM" address space



Bank

```
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
```

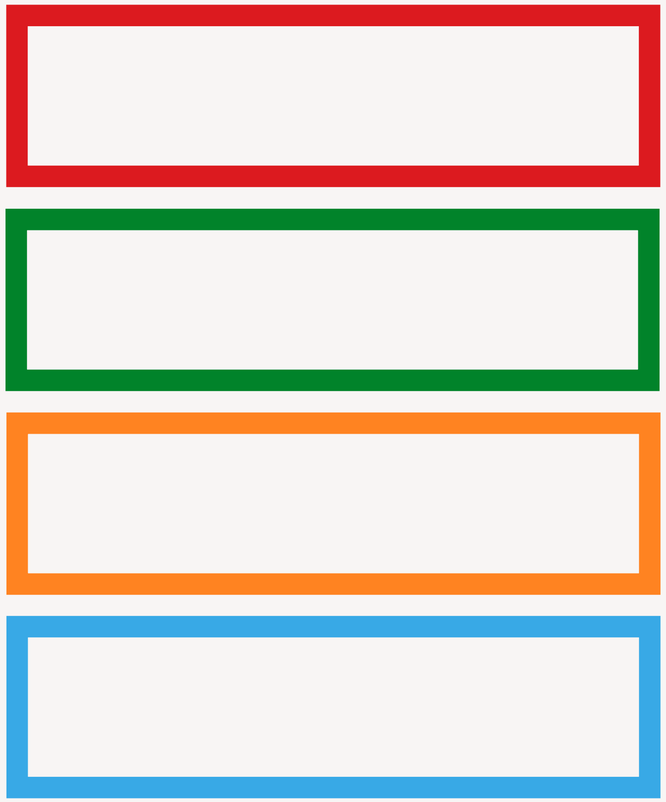
Caches

# Cache Eviction

## Eviction sets

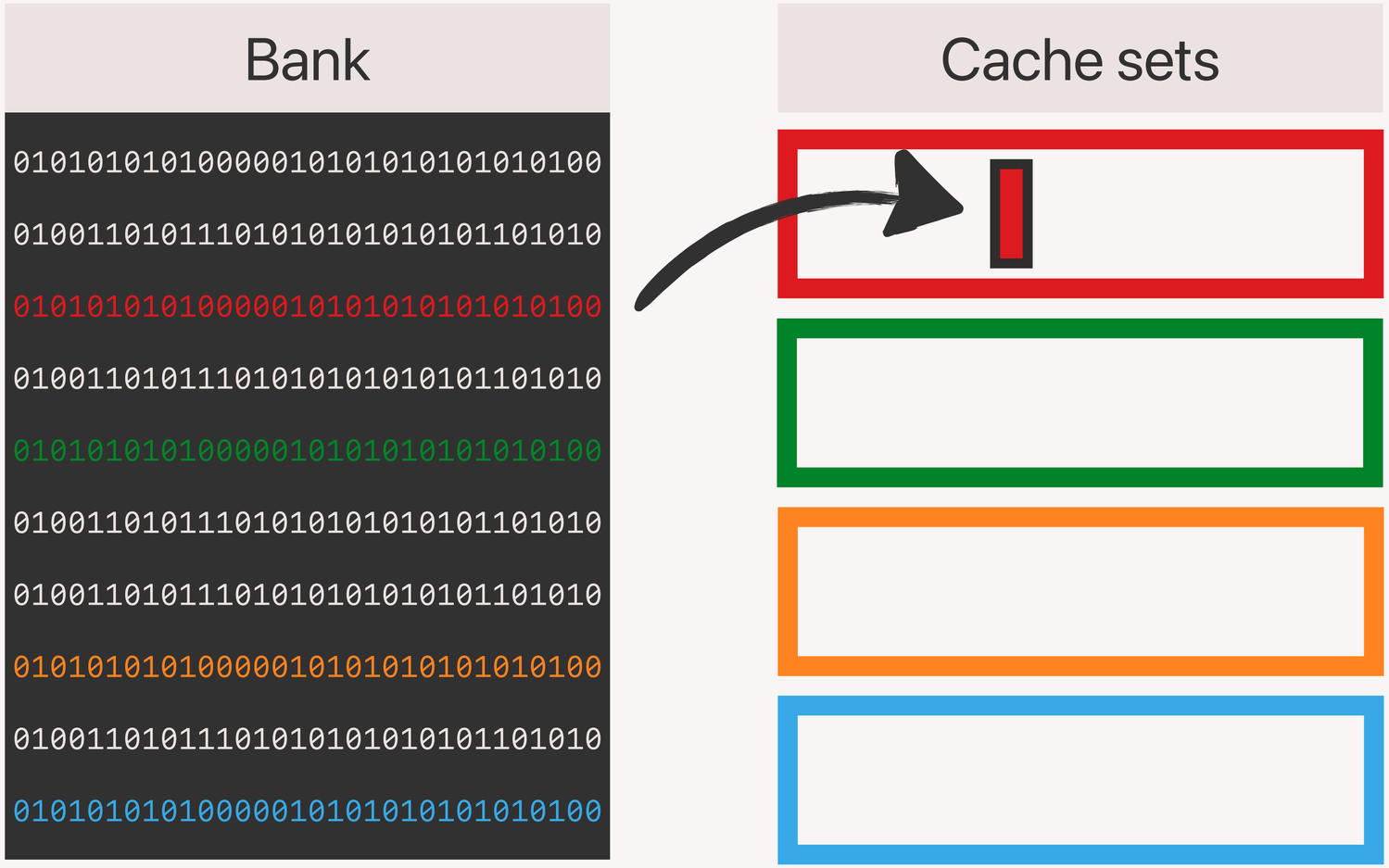
Bank

```
0101010101000001010101010101010100
0100110101110101010101010101101010
0101010101000001010101010101010100
0100110101110101010101010101101010
0101010101000001010101010101010100
0100110101110101010101010101101010
0100110101110101010101010101101010
0101010101000001010101010101010100
0100110101110101010101010101101010
0101010101000001010101010101010100
```



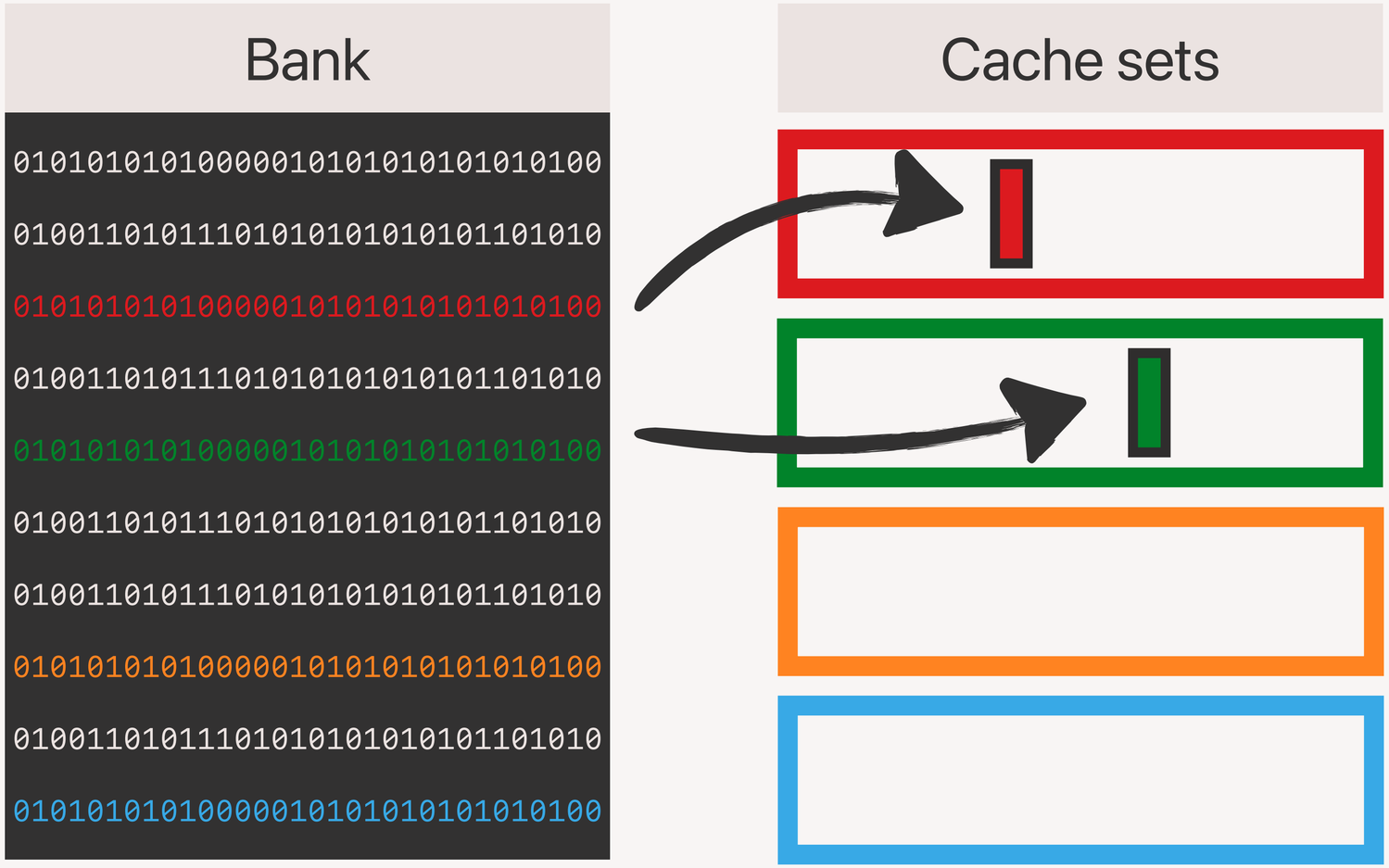
# Cache Eviction

## Eviction sets



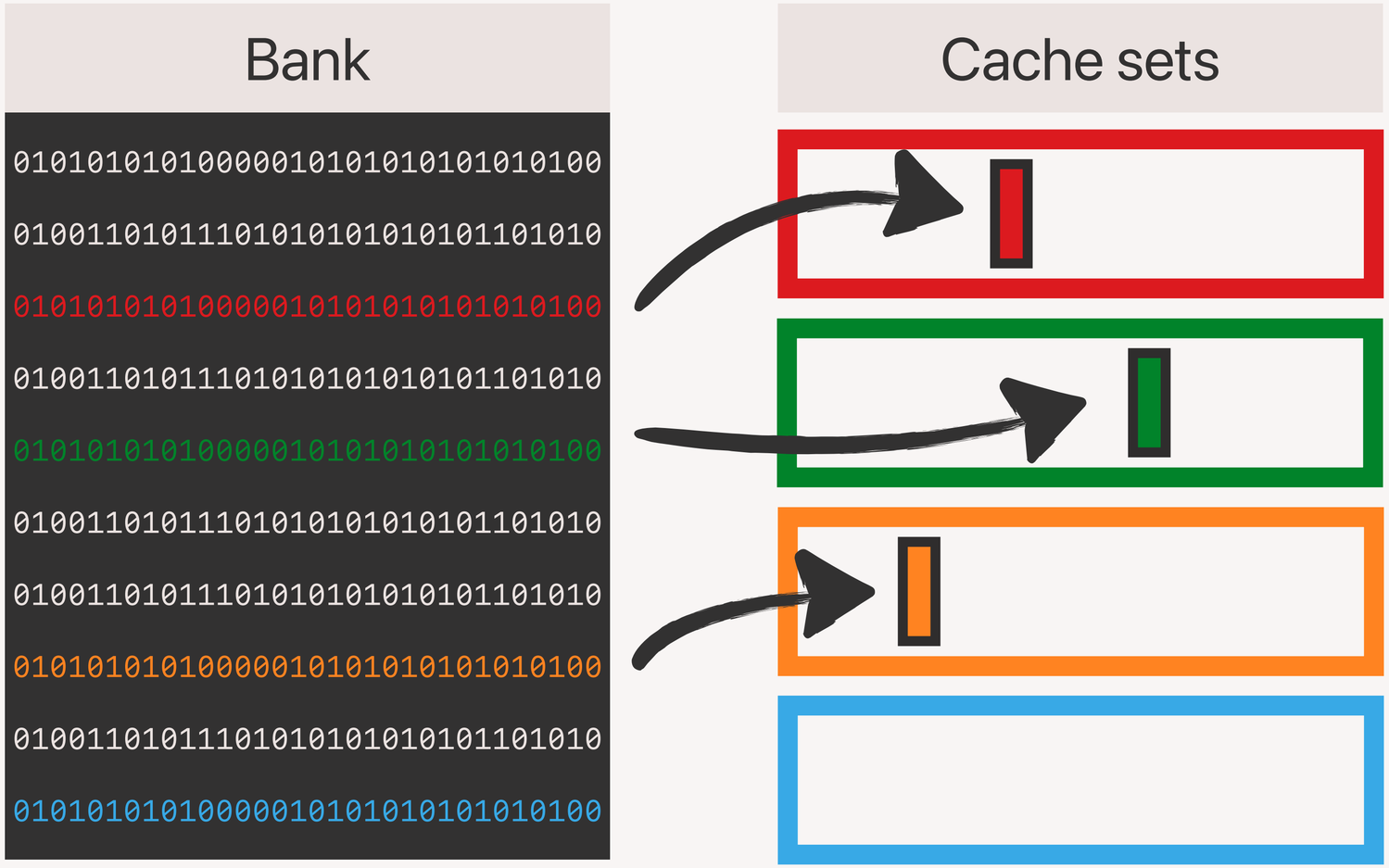
# Cache Eviction

## Eviction sets



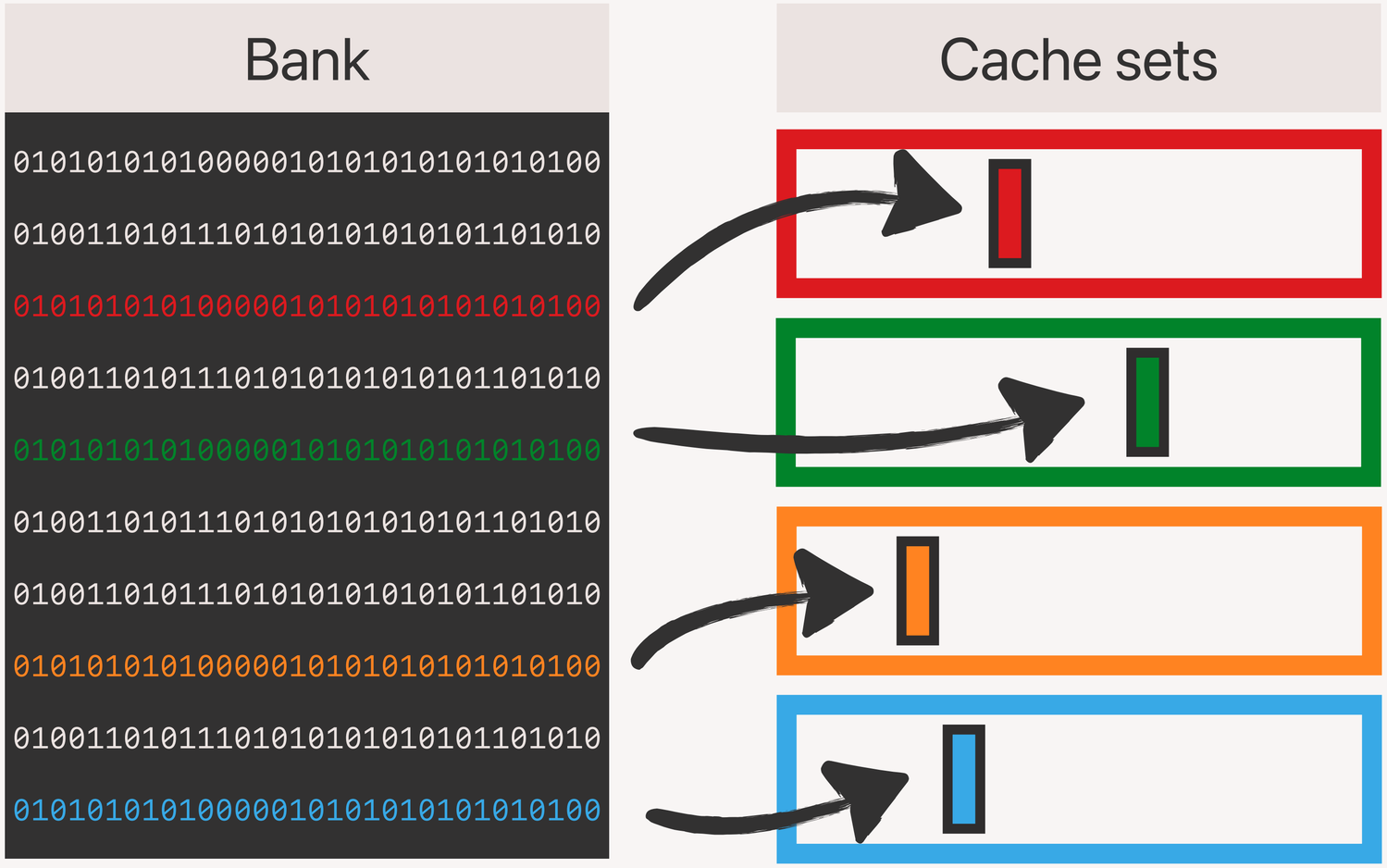
# Cache Eviction

## Eviction sets



# Cache Eviction

## Eviction sets

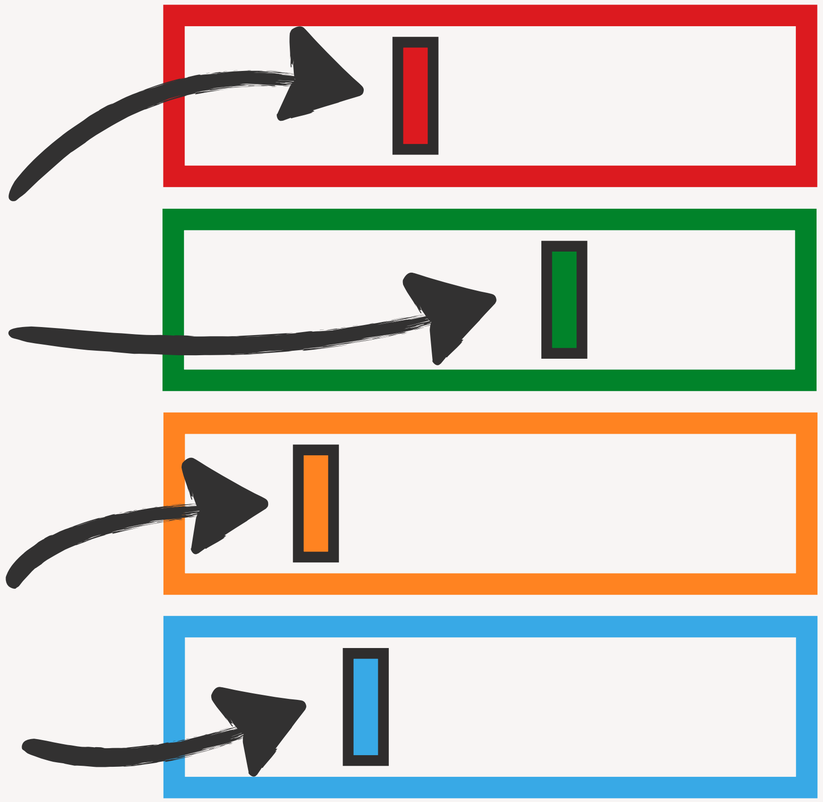


# Cache Eviction

## Eviction sets

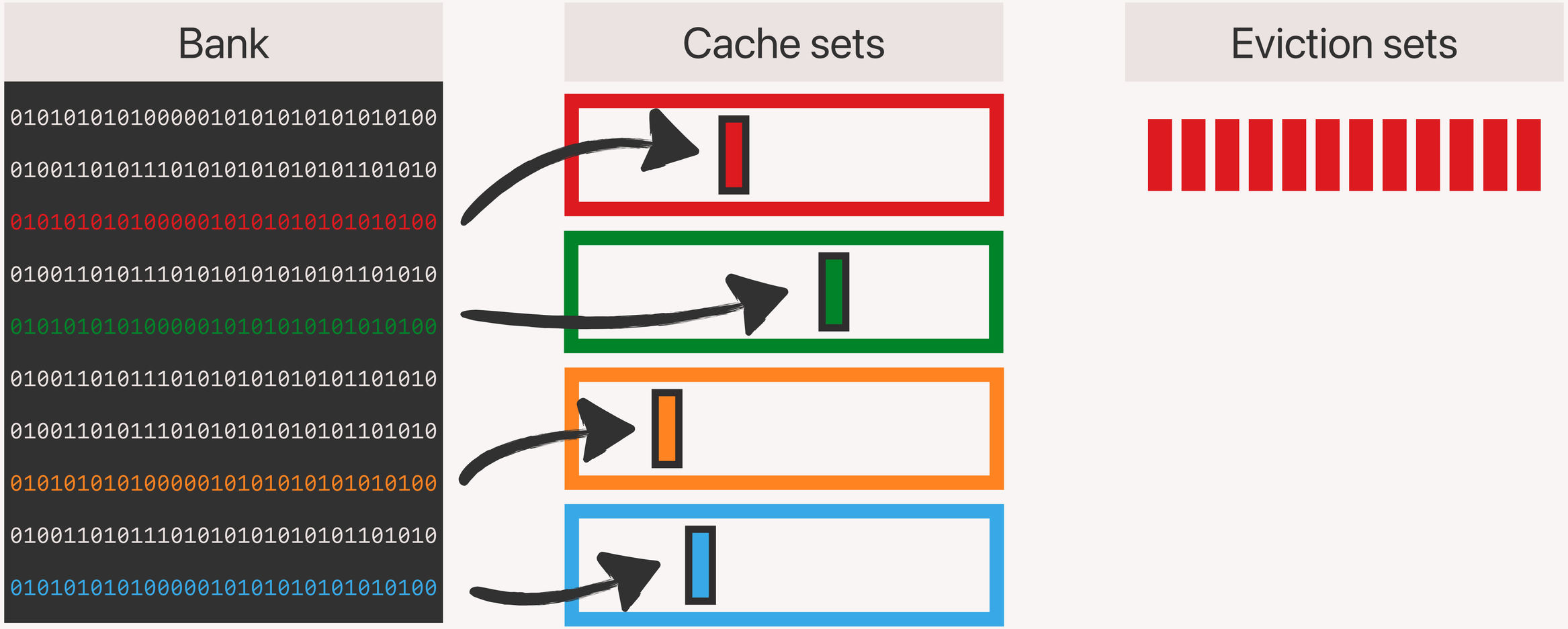
| Bank | Cache sets | Eviction sets |
| --- | --- | --- |

```
0101010101000001010101010101010100
0100110101110101010101010101101010
0101010101000001010101010101010100
0100110101110101010101010101101010
0101010101000001010101010101010100
0100110101110101010101010101101010
0100110101110101010101010101101010
0101010101000001010101010101010100
0100110101110101010101010101101010
0101010101000001010101010101010100
```



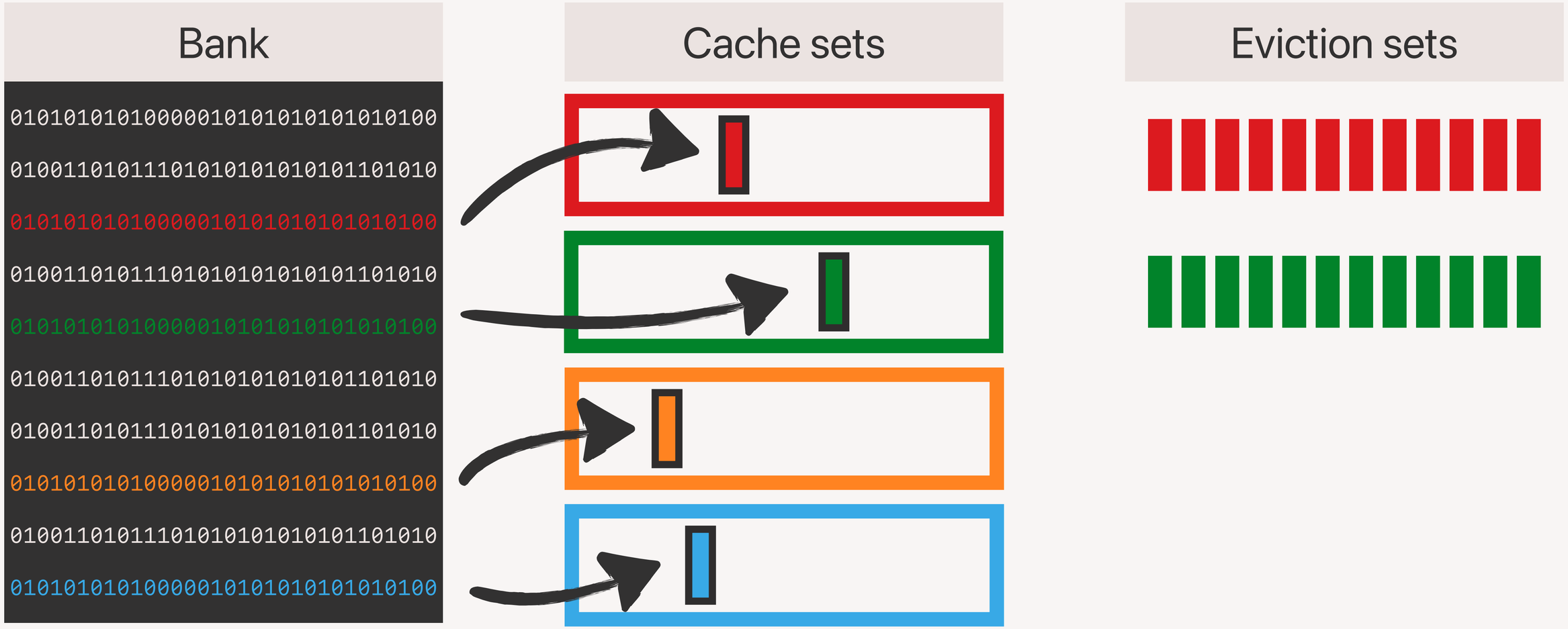
# Cache Eviction

## Eviction sets



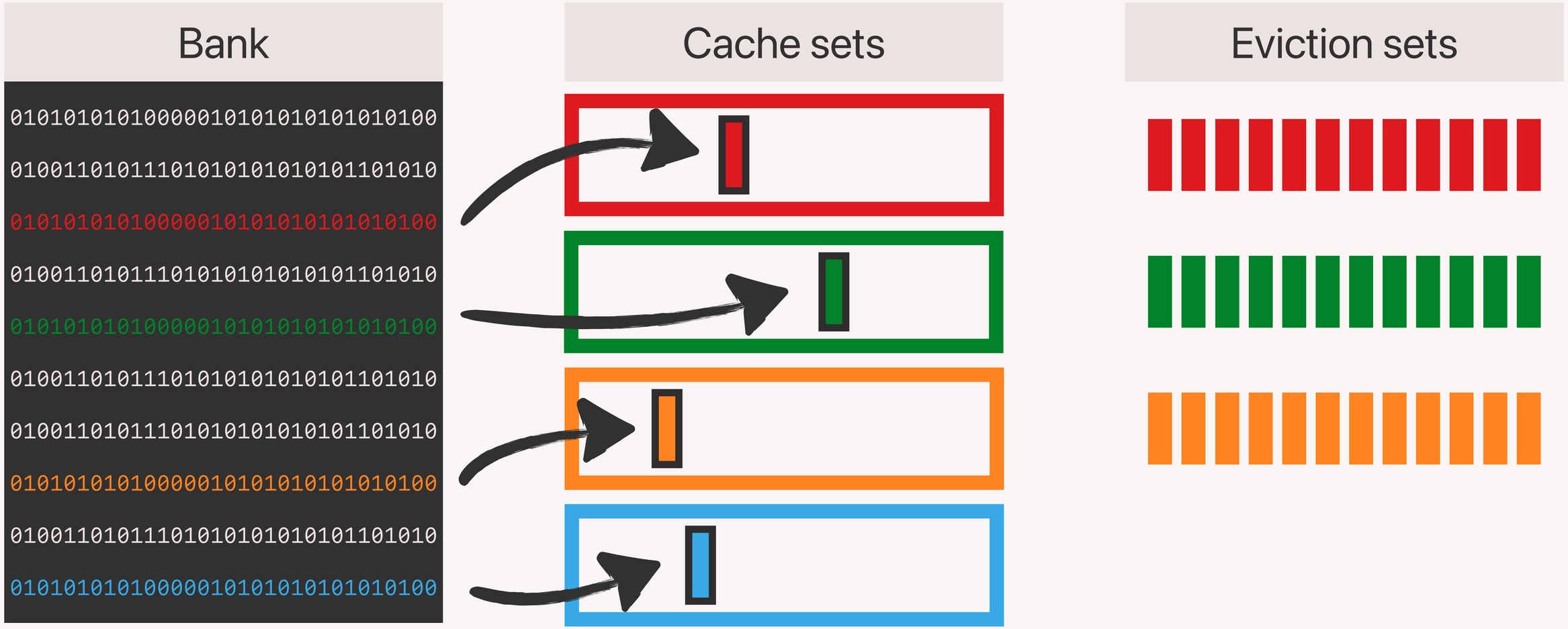
# Cache Eviction

## Eviction sets



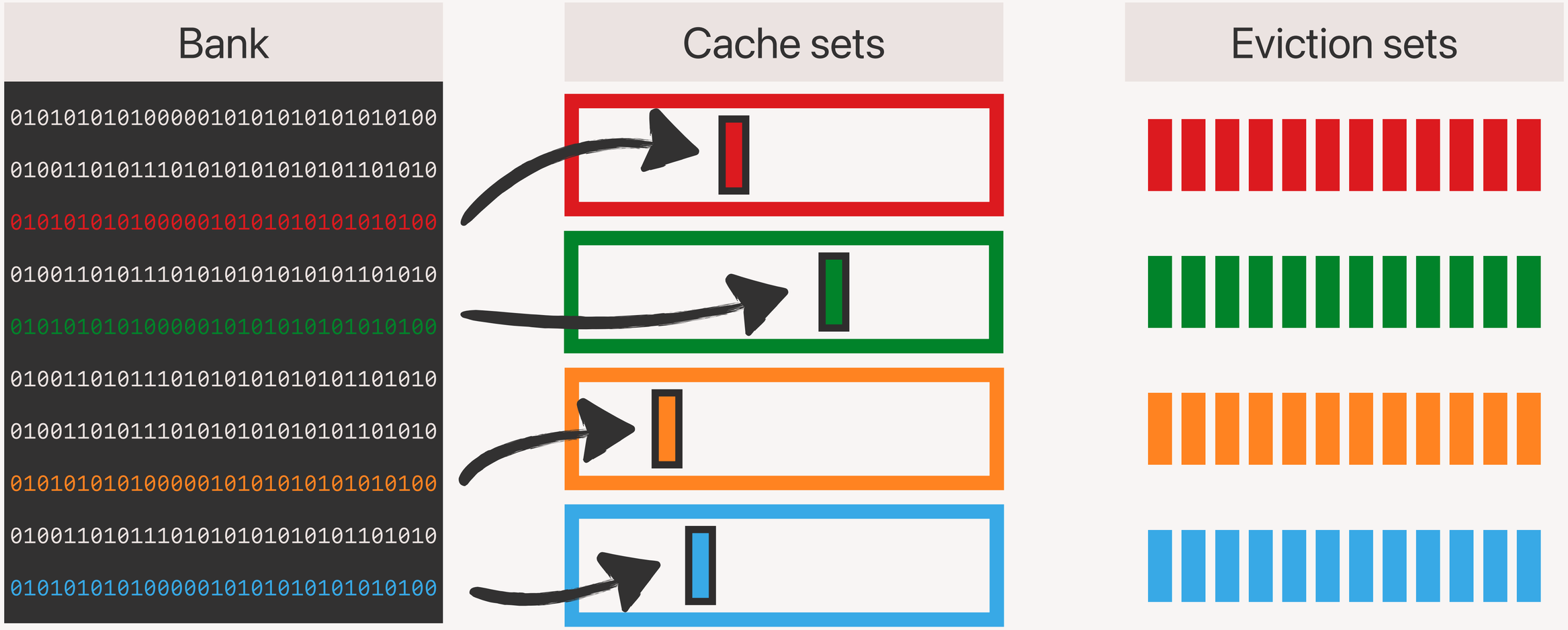
# Cache Eviction

## Eviction sets



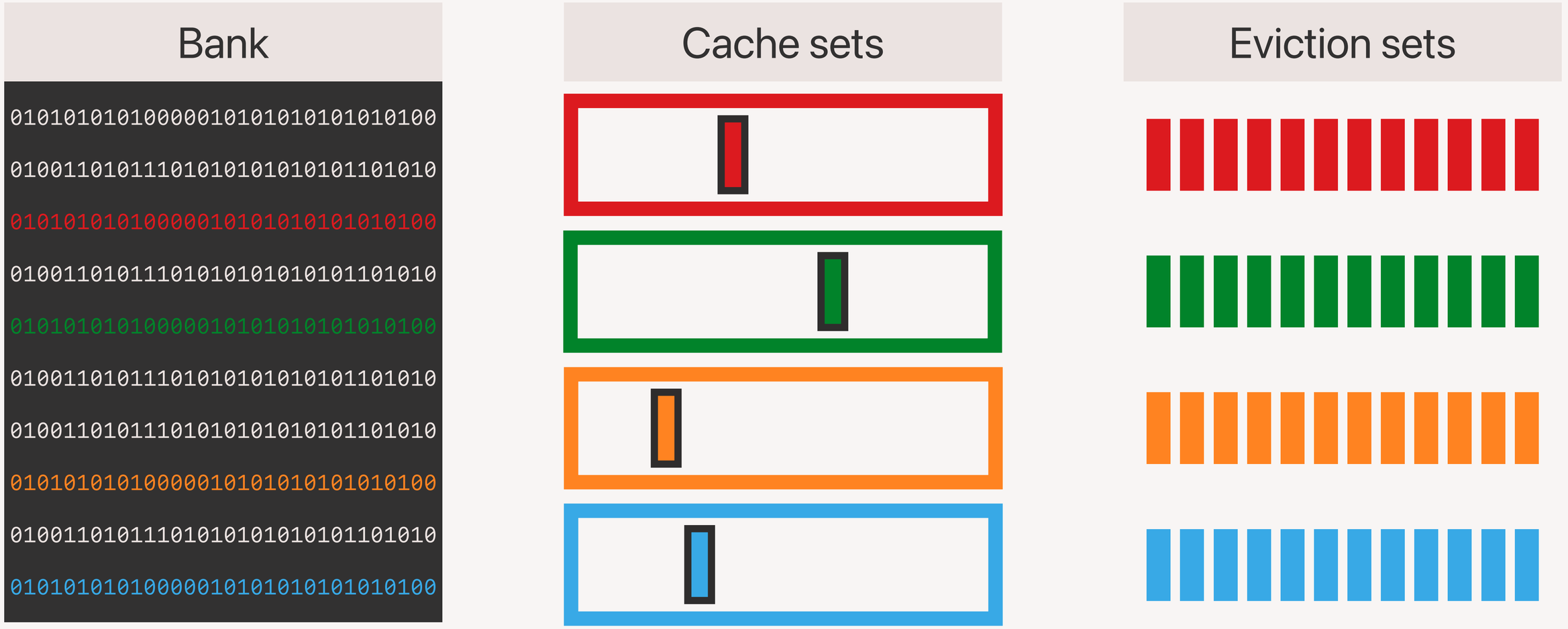
# Cache Eviction

## Eviction sets



# Cache Eviction

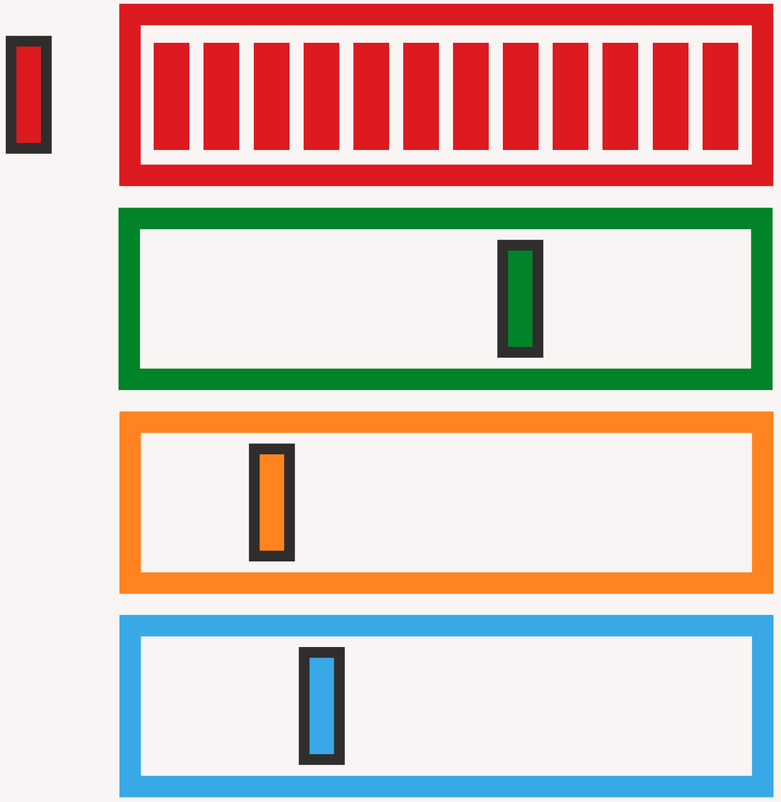
## Eviction sets

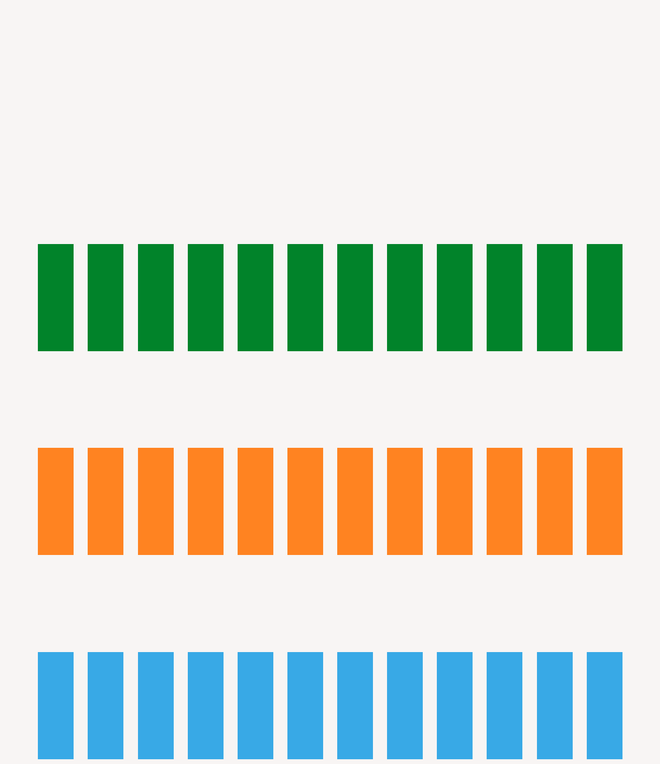


# Cache Eviction

## Eviction sets

Bank

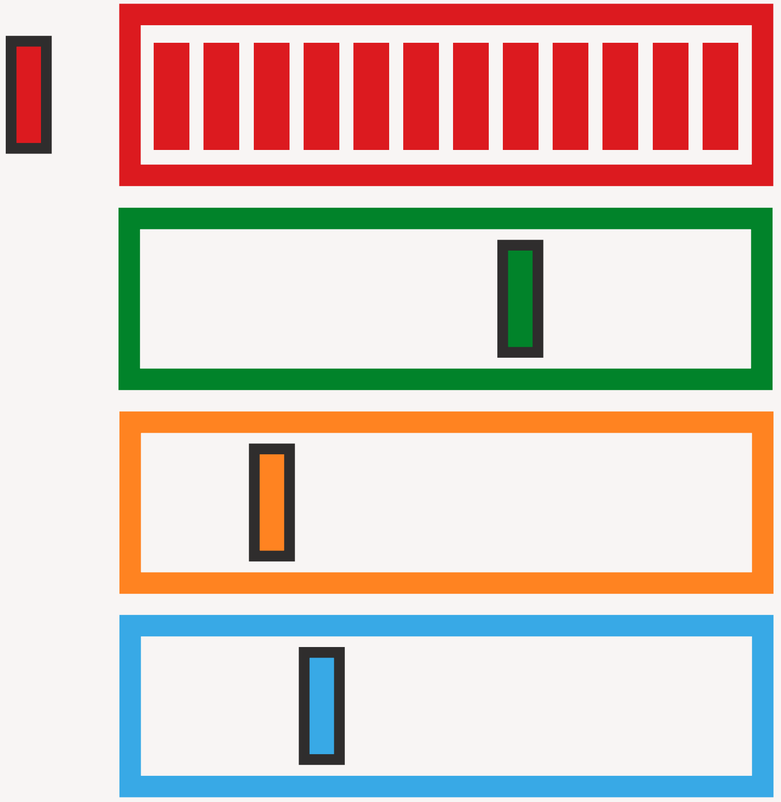



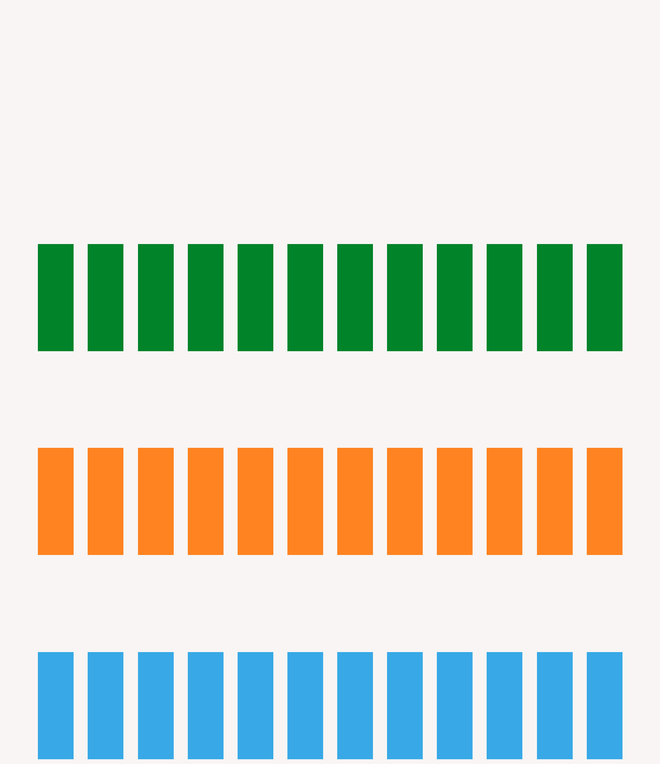



# Cache Eviction

## Eviction sets

Bank

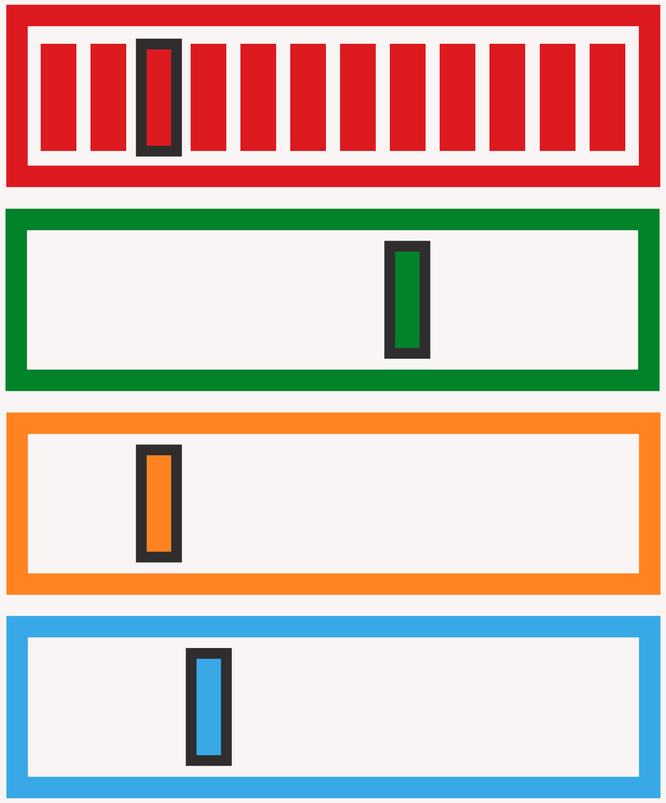
# Cache Eviction

## Eviction sets

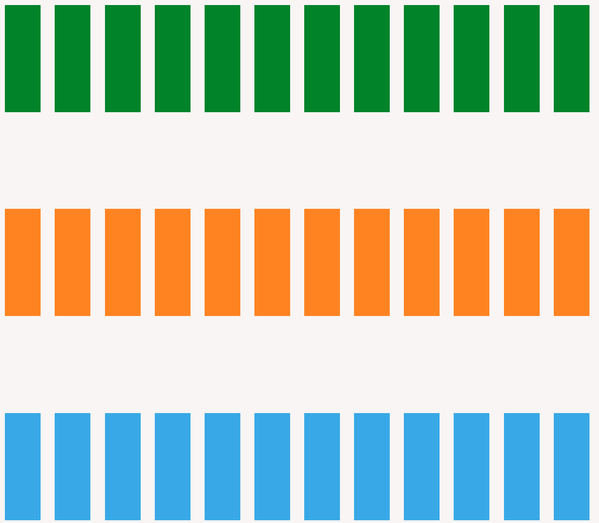
Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
```

Cache sets

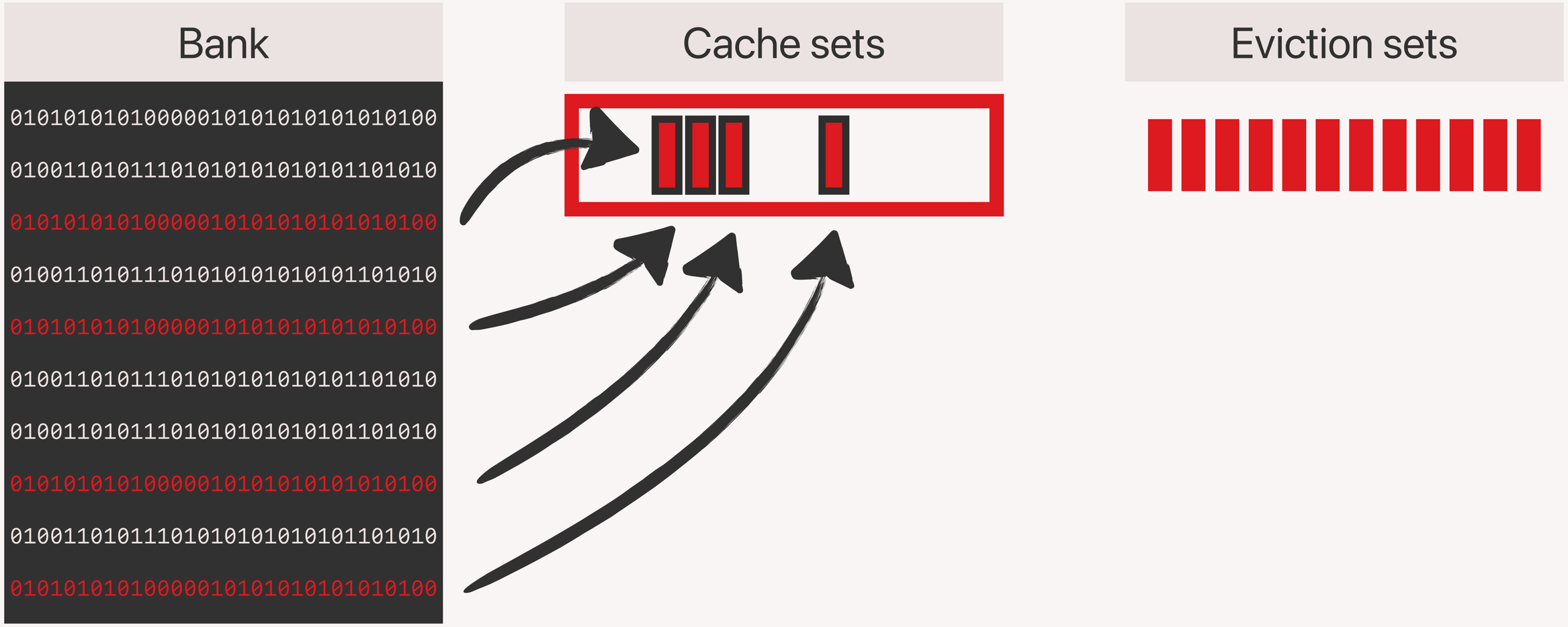


Eviction sets



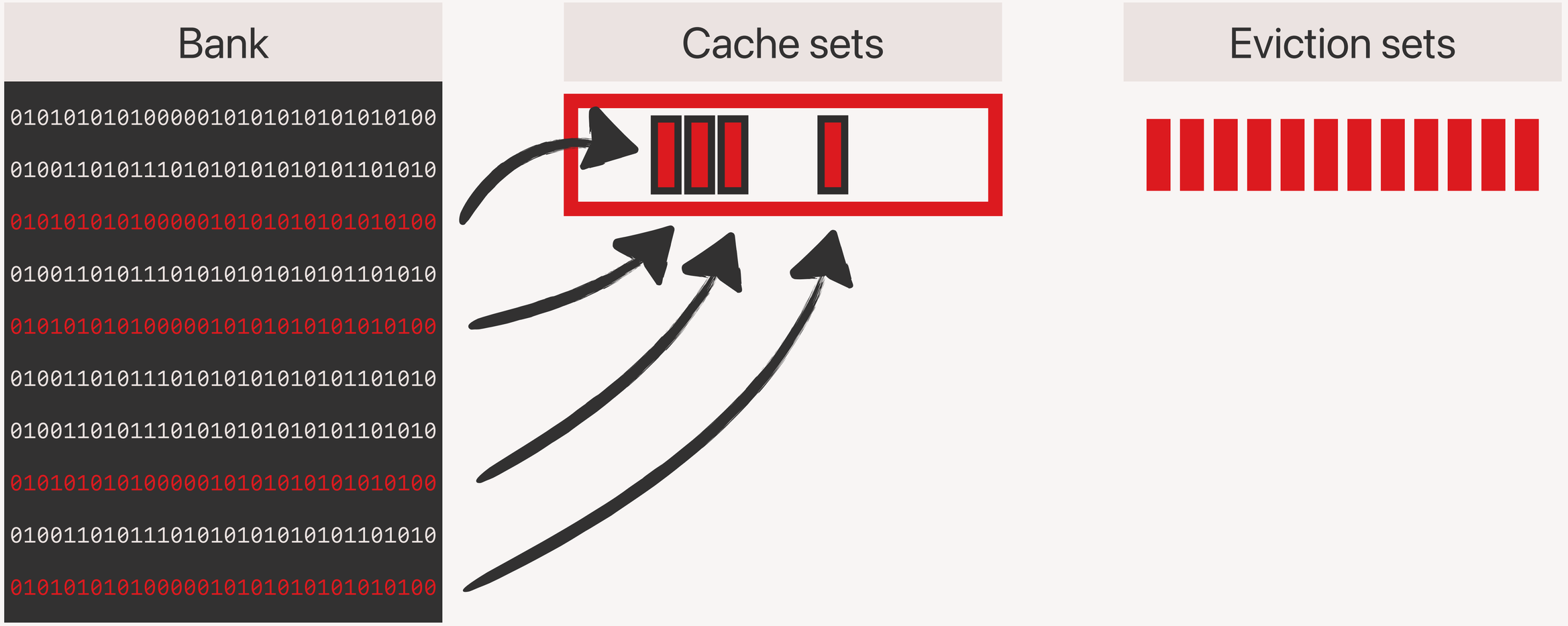
# Cache Eviction

## Same set?



# Cache Eviction

## Same set? Not possible!

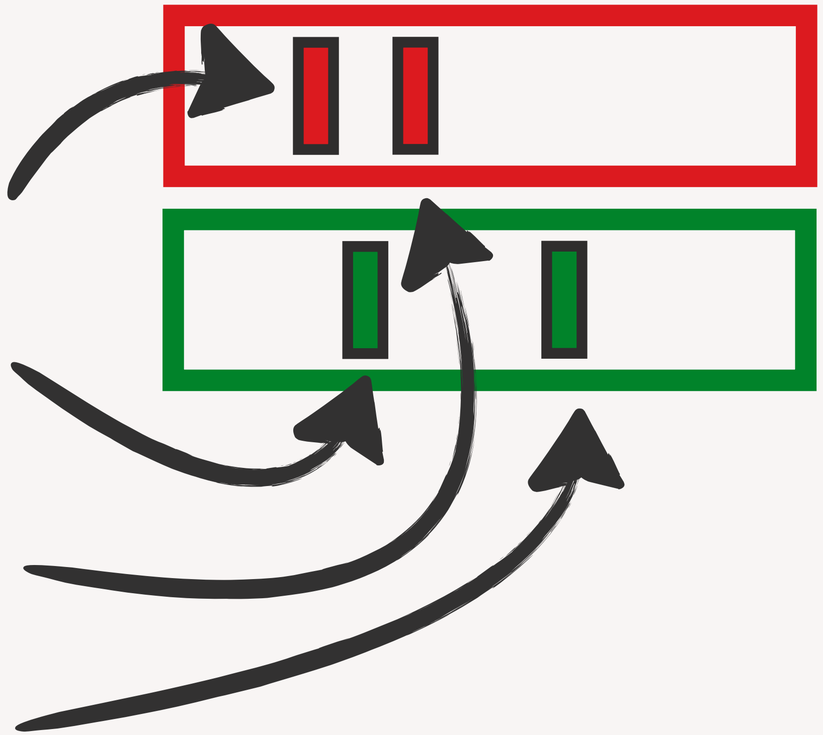


# Cache Eviction

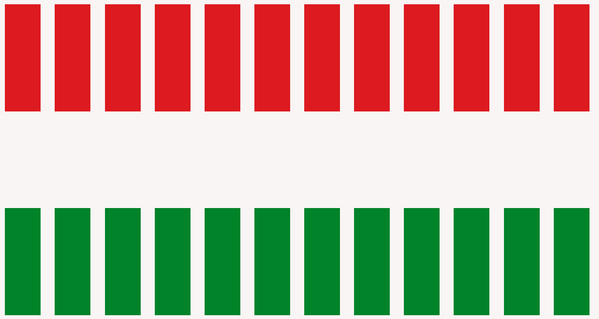
## Two sets

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
```



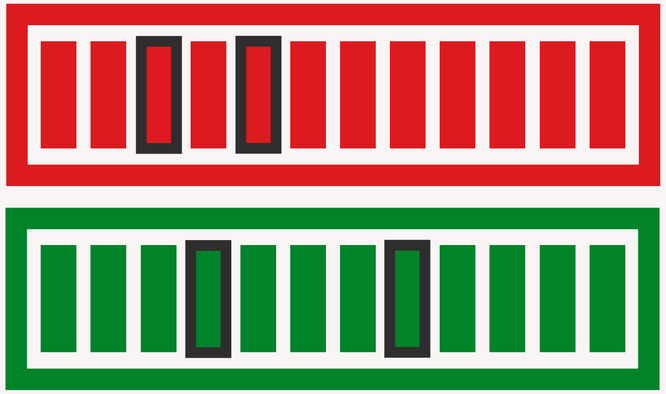

Eviction sets



# Cache Eviction

## Two sets + cache hits

| Bank |
| --- |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |
| 0100110101110101010101010101101010 |
| 0101010101000001010101010101010100 |

| Cache sets |
| --- |
|  |

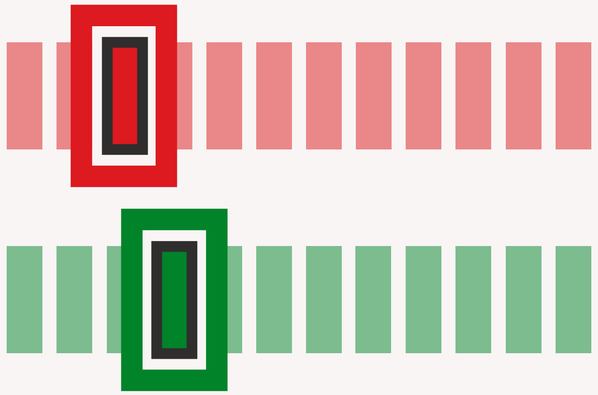
| Eviction sets |
| --- |

# Cache Eviction

## Two sets + cache hits

| Bank | Cache sets | Eviction sets |
| --- | --- | --- |

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
```

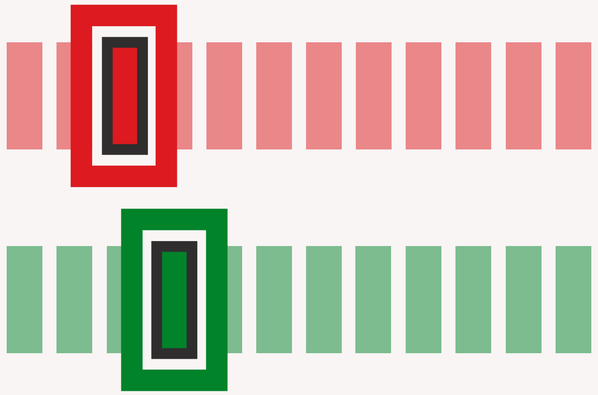


# Cache Eviction

## Two sets + cache hits + self-eviction

| Bank | Cache sets | Eviction sets |
| --- | --- | --- |

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
```



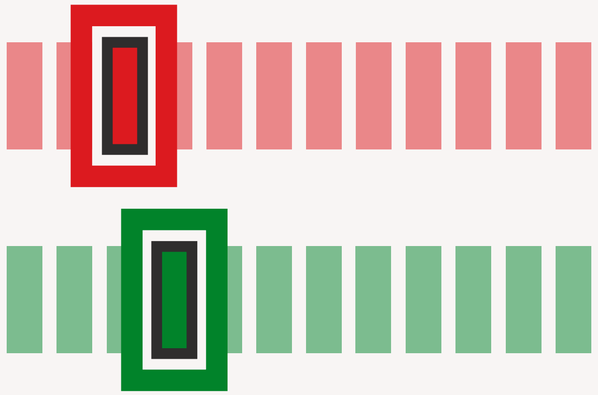
# Cache Eviction

## Two sets + cache hits + self-eviction

Bank

```
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01001101011101010101010101101010
01010101010000010101010101010100
01001101011101010101010101101010
01010101010000010101010101010100
```

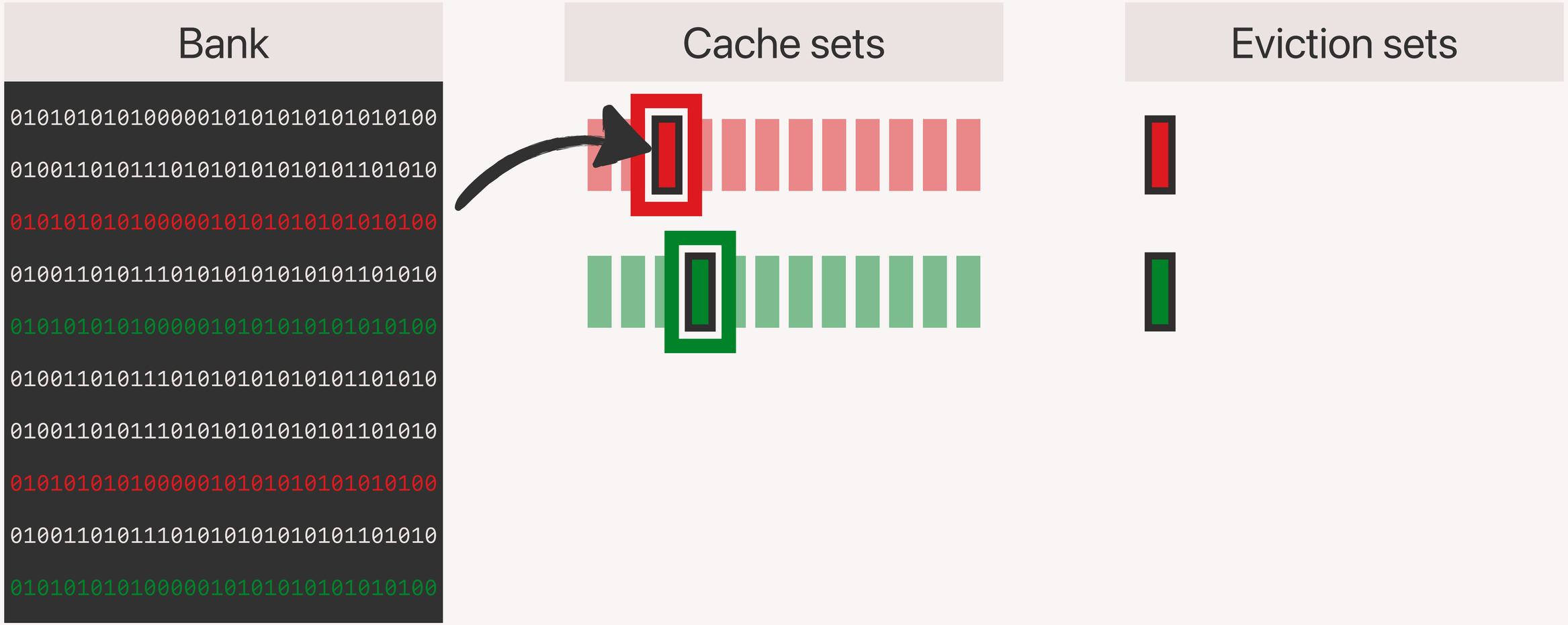
Cache sets



Eviction sets

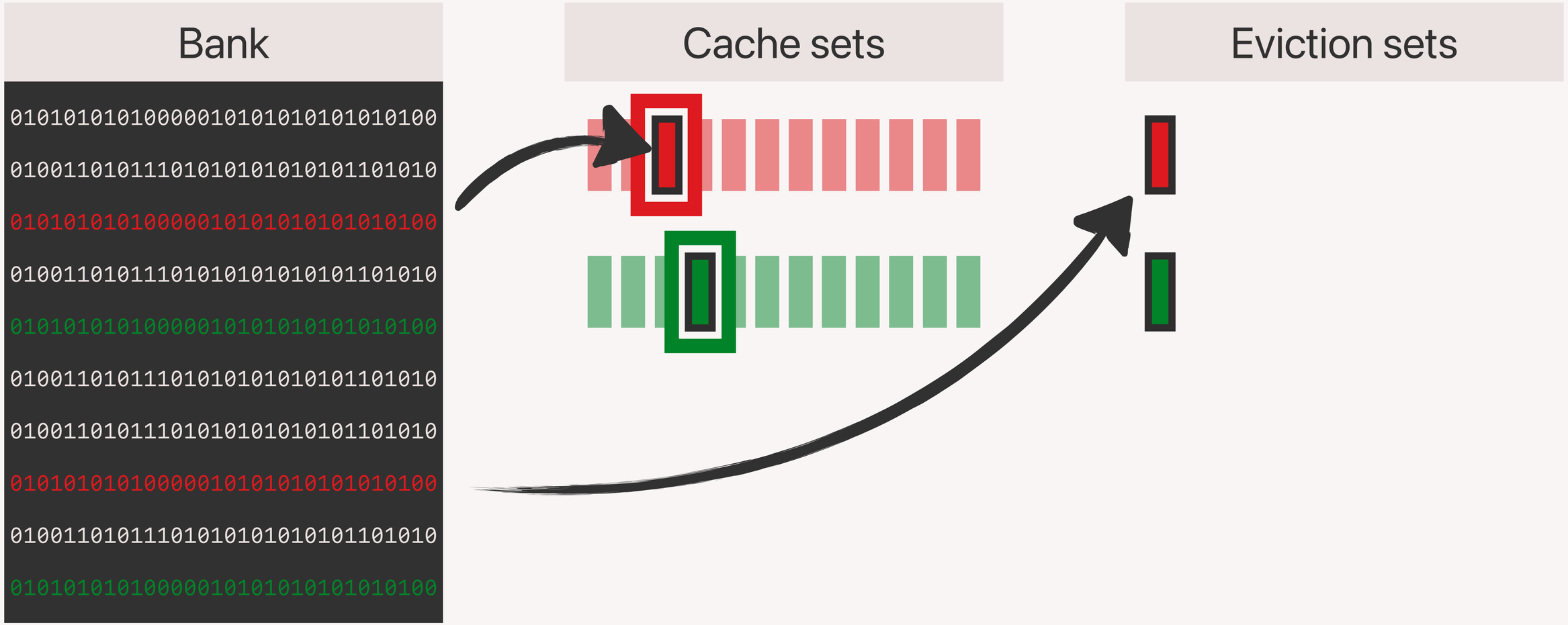
# Cache Eviction

## Two sets + cache hits + self-eviction



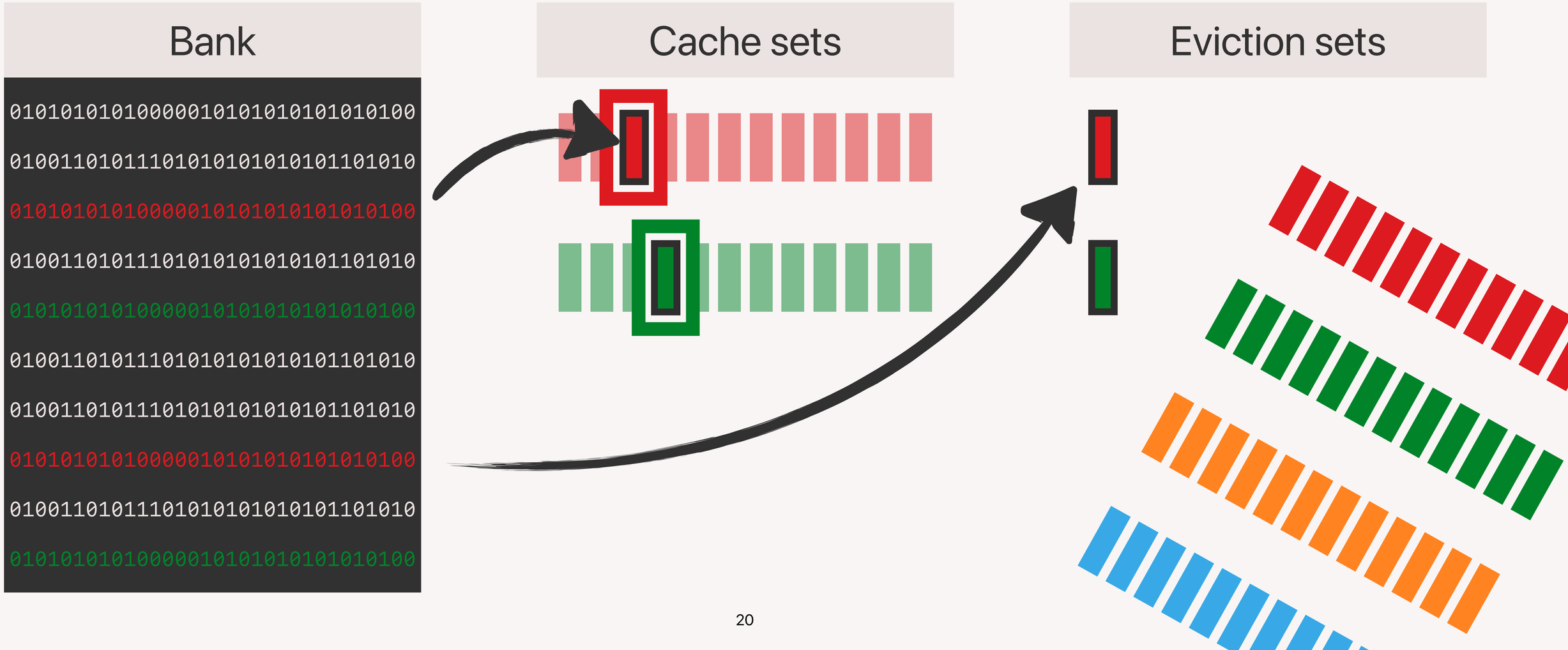
# Cache Eviction

## Two sets + cache hits + self-eviction



# Cache Eviction

## Two sets + cache hits + self-eviction



# Rowhammer From JavaScript

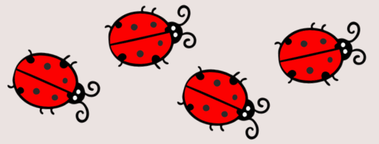
## Complicated

Virtual address space

Physical space

Huge pages

"DRAM" address space



Bank

```
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
0101010101000001010
0100110101110101010
```

Caches

# Rowhammer From JavaScript

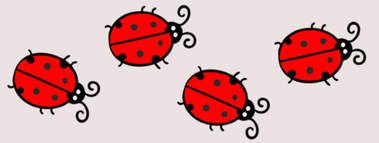
## Complicated

Virtual address space

Physical space

Huge pages

"DRAM" address space



Bank

```
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
```

Caches

# Rowhammer From JavaScript

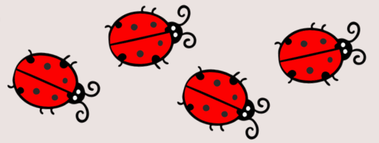
## Complicated

Virtual address space

Physical space

Huge pages

"DRAM" address space



Bank

```
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
01010101010000010101
01001101011101010101
```

Self-evict

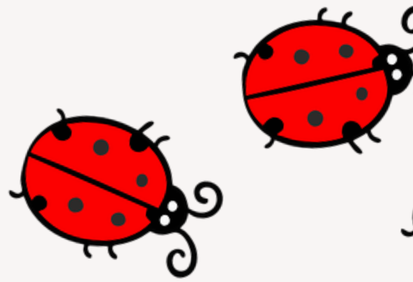
Caches

# Done!

# Done!?

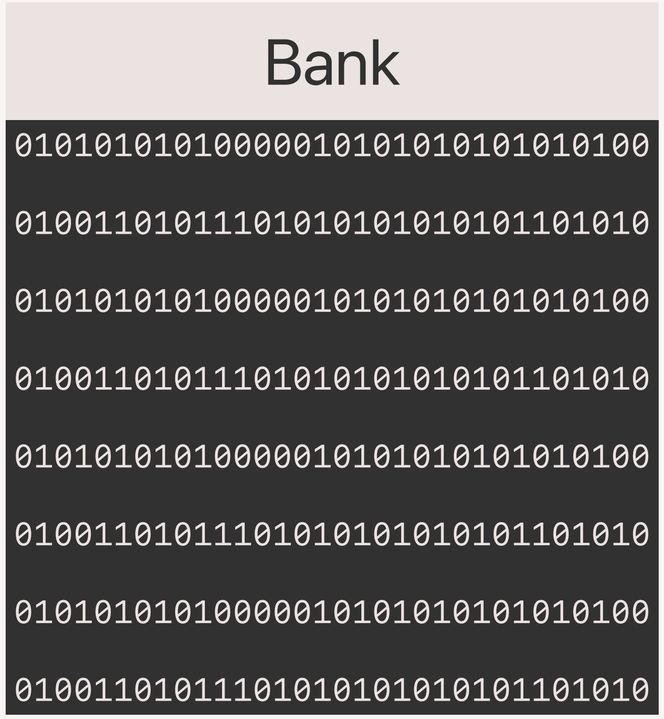
# TRR's Little Secret

## Part III



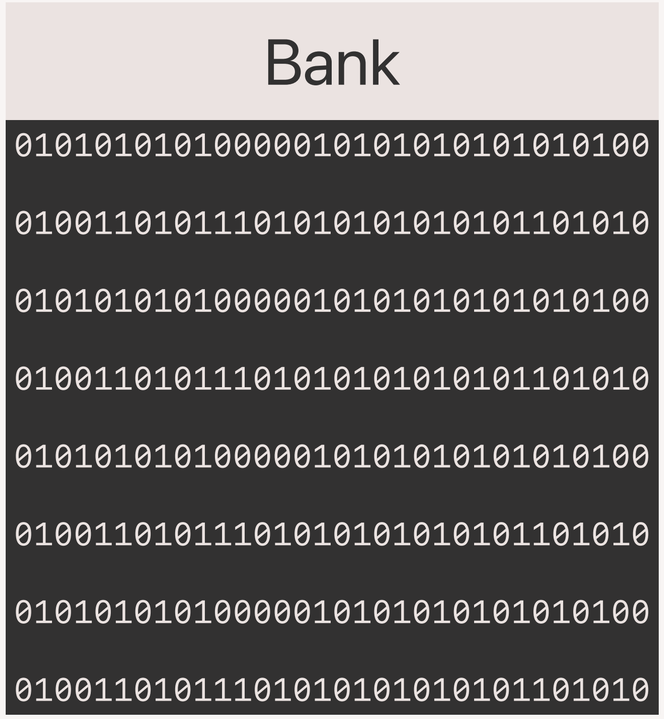
# Refresh Commands

## Prevent data loss



# Refresh Commands

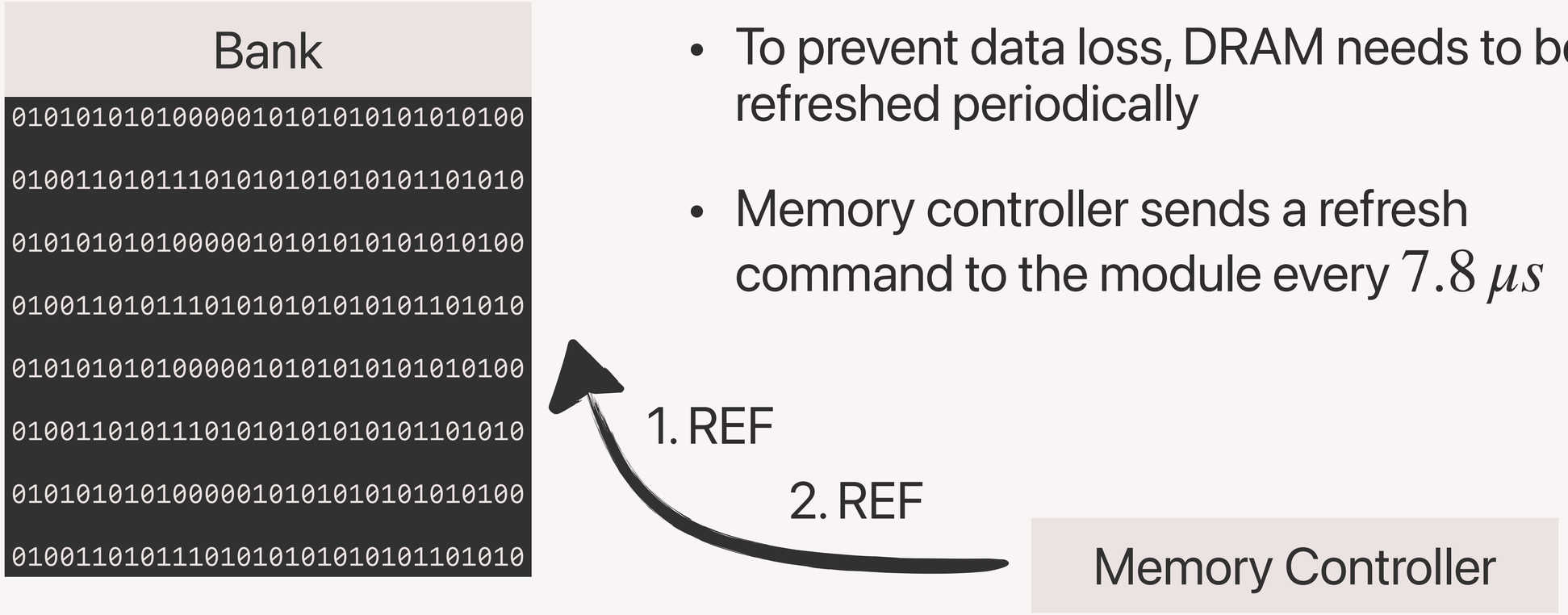
## Prevent data loss




- To prevent data loss, DRAM needs to be refreshed periodically
- Memory controller sends a refresh command to the module every $7.8\ \mu s$

# Refresh Commands

## Prevent data loss



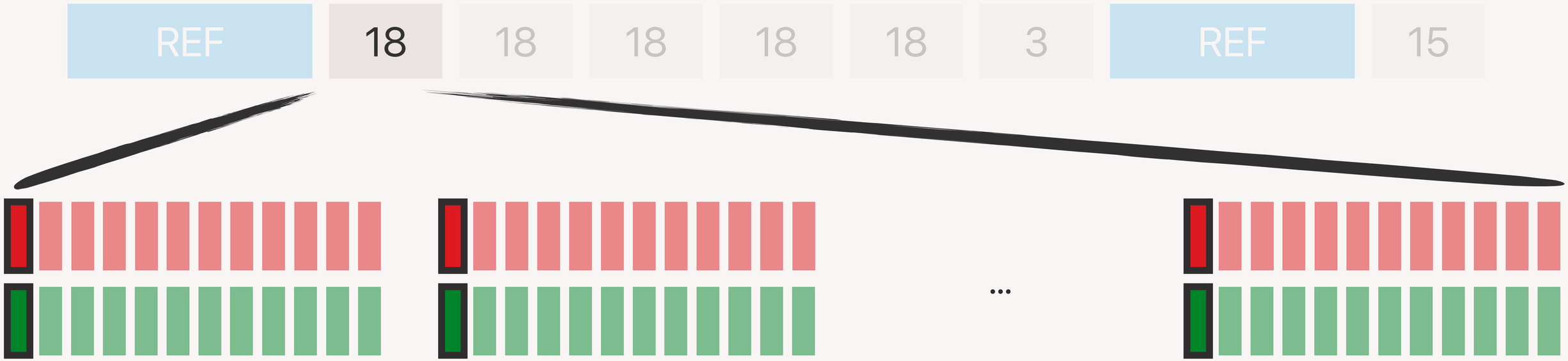

- To prevent data loss, DRAM needs to be refreshed periodically
- Memory controller sends a refresh command to the module every $7.8\ \mu s$

# Request Layout

## Example

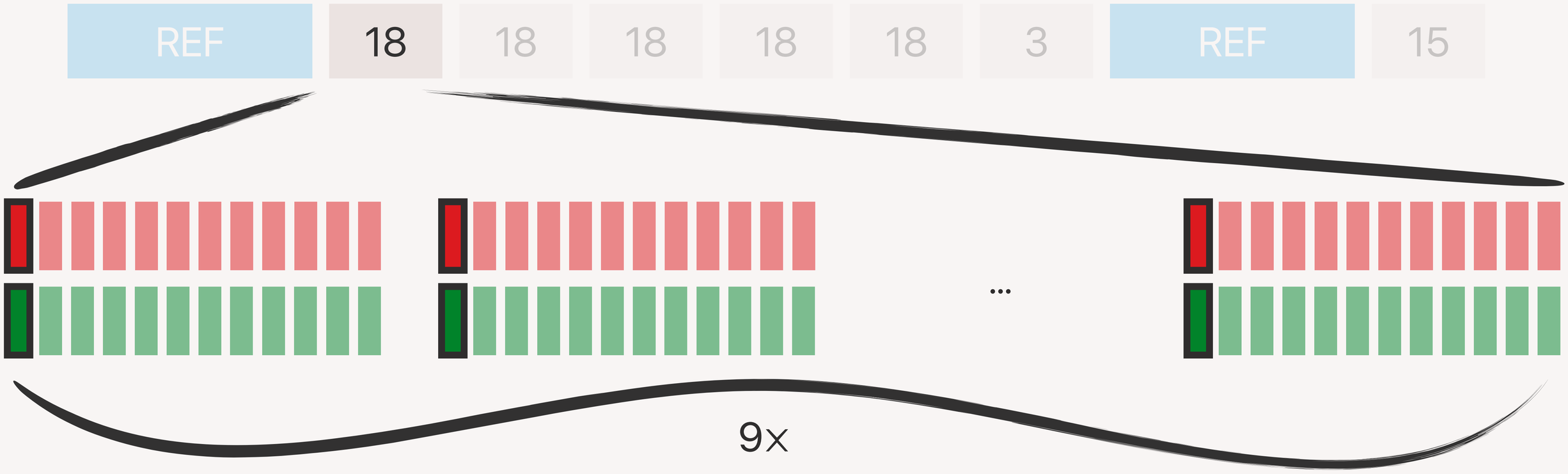
# Request Layout

## Example



# Request Layout

## Example

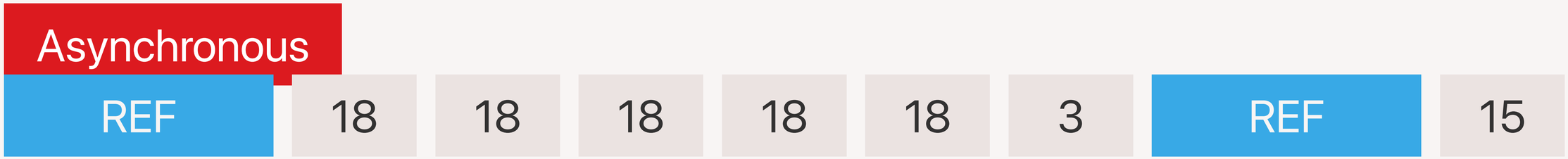


# Request Layout

## Example

# Request Layout

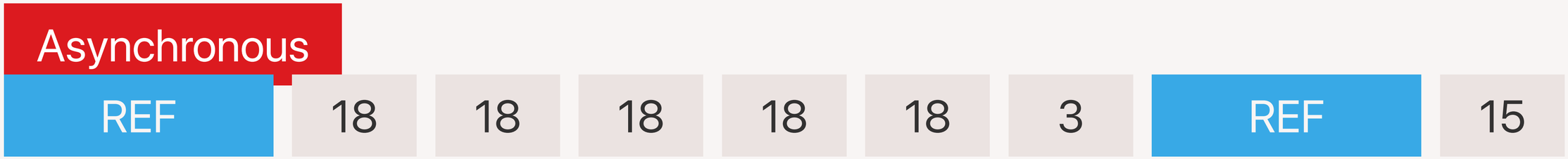
## Example



# Request Layout

## Example

Will sample all aggressors over time



# Request Layout

## Example

Will sample all aggressors over time

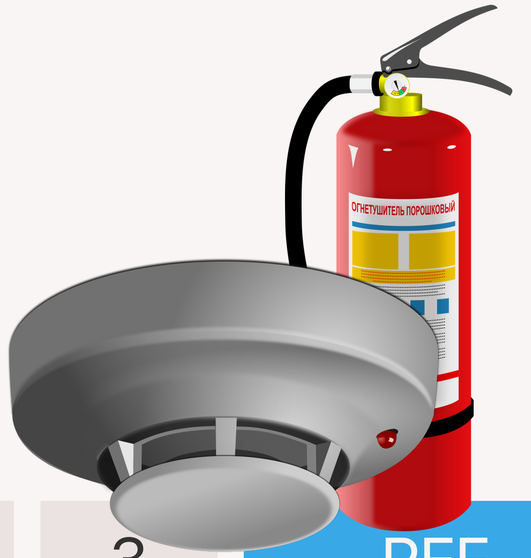

Asynchronous

| REF | 18 | 18 | 18 | 18 | 18 | 3 | REF | 15 |
|---|---|---|---|---|---|---|---|---|

# Request Layout

## Example

Will sample all aggressors over time




Asynchronous

| REF | 18 | 18 | 18 | 18 | 18 | 3 | REF | 15 |
|---|---|---|---|---|---|---|---|---|

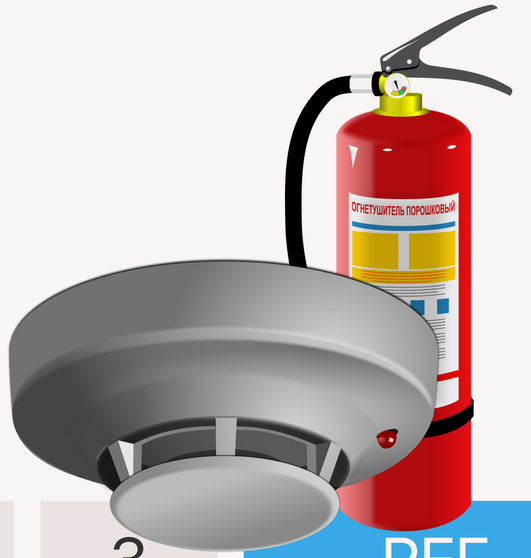
| REF | NOP | 18 | NOP | 18 | NOP | 18 | REF | NOP |
|---|---|---|---|---|---|---|---|---|

$7.8\,\mu s$

# Request Layout

## Example

Will sample all aggressors over time




Asynchronous

| REF | 18 | 18 | 18 | 18 | 18 | 3 | REF | 15 |
|---|---|---|---|---|---|---|---|---|

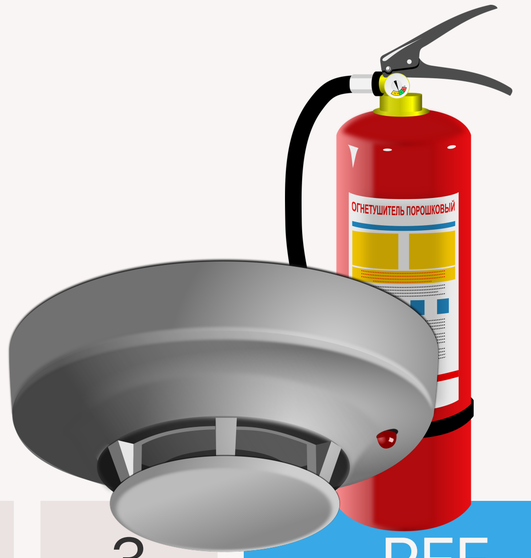
Synchronous

| REF | NOP | 18 | NOP | 18 | NOP | 18 | REF | NOP |
|---|---|---|---|---|---|---|---|---|

$7.8\,\mu s$

# Request Layout

## Example




Will sample all aggressors over time

Asynchronous

| REF | 18 | 18 | 18 | 18 | 18 | 3 | REF | 15 |
|---|---|---|---|---|---|---|---|---|

Will always sample the same aggressors, some will never be sampled

Synchronous

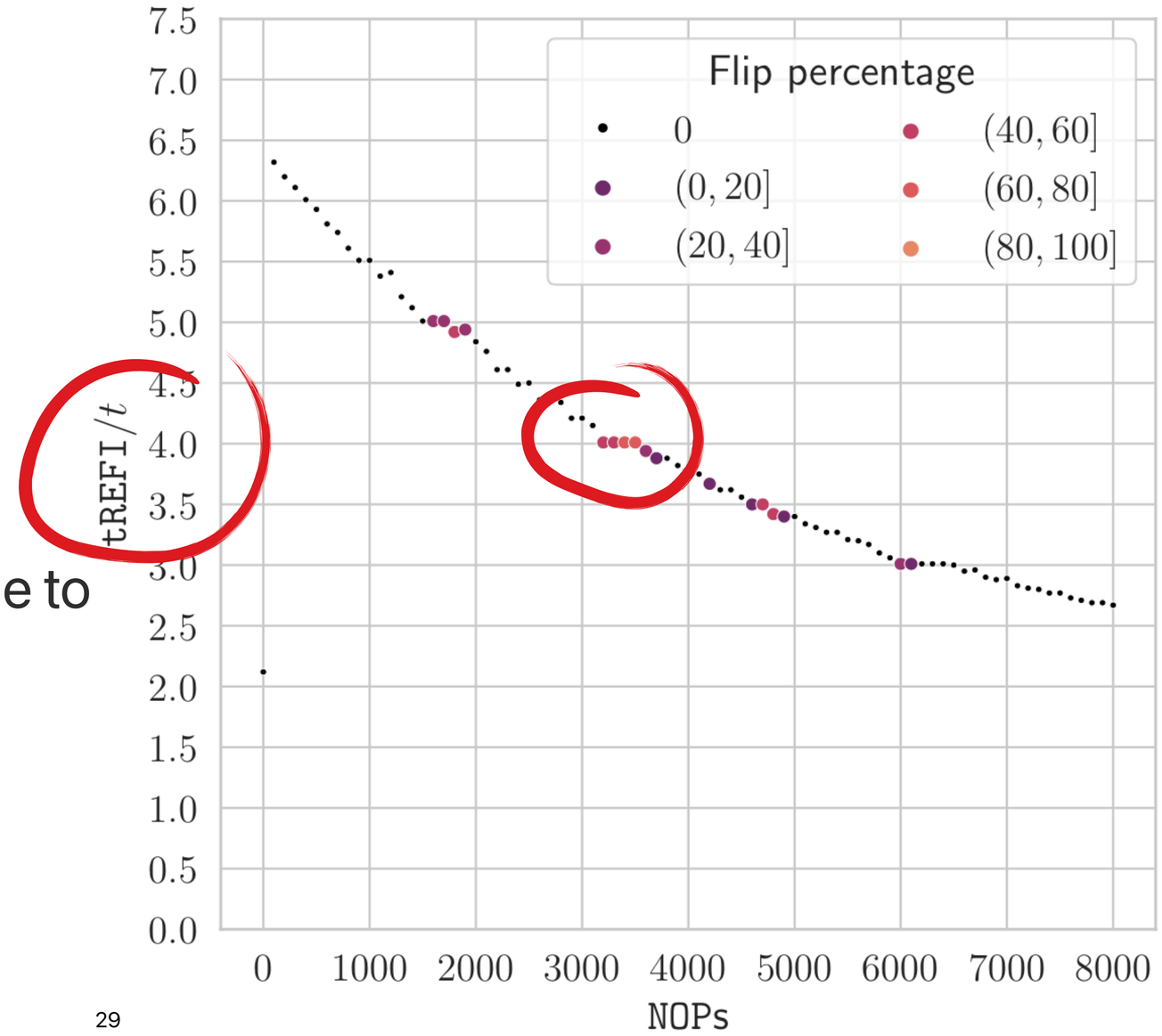
| REF | NOP | 18 | NOP | 18 | NOP | 18 | REF | NOP |
|---|---|---|---|---|---|---|---|---|

$7.8\,\mu s$

# Bit Flips When Synchronized

## By inserting nothing (NOPs)

1. We observe synchronization with refresh commands
2. Only when synchronized we are able to trigger bit flips



# Rowhammer From JavaScript

## Complicated

Virtual address space

Physical space

"DRAM" address space

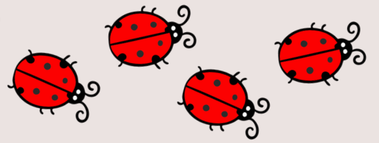
Huge pages

Hammer

REF

Self-evict

Caches



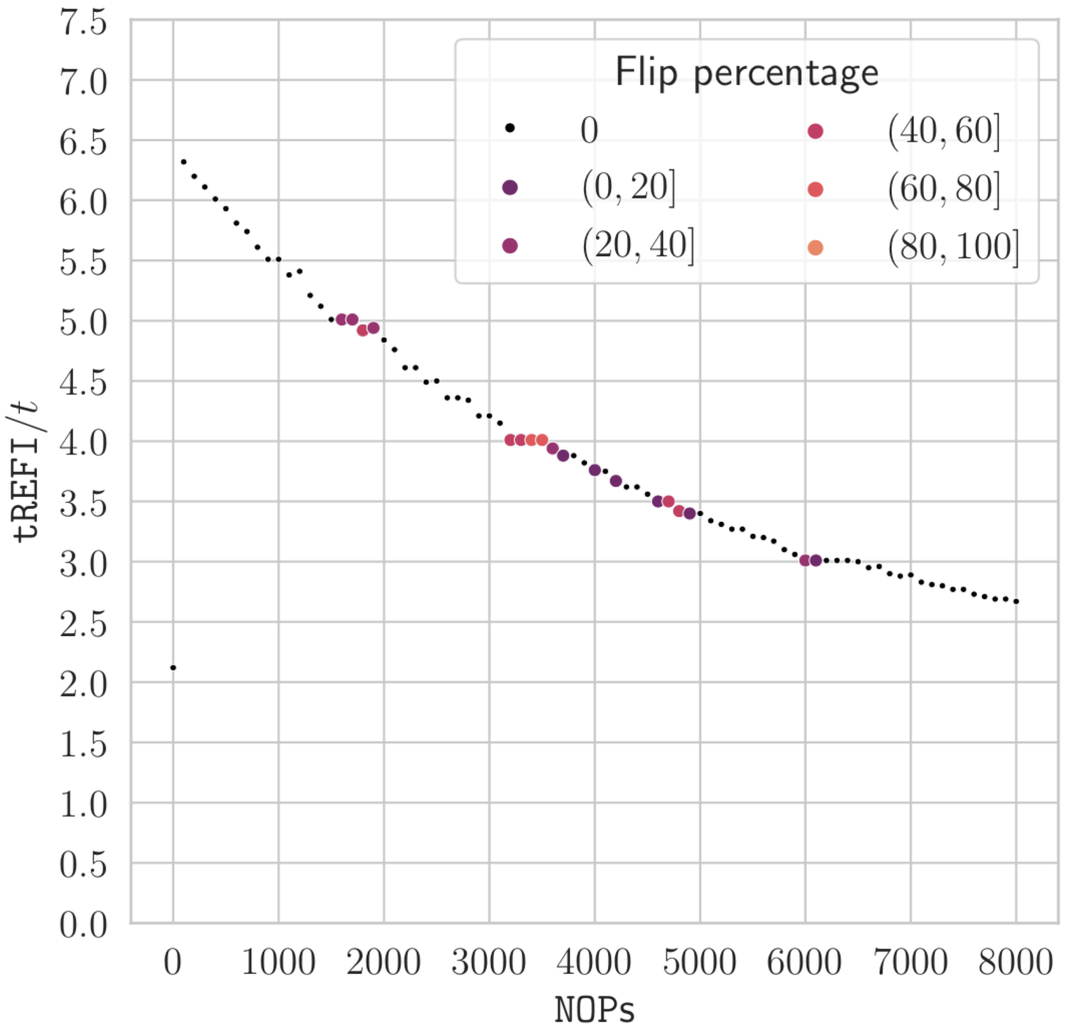
Bank

# Synchronizing in JavaScript May Be Difficult

## Can we do better?

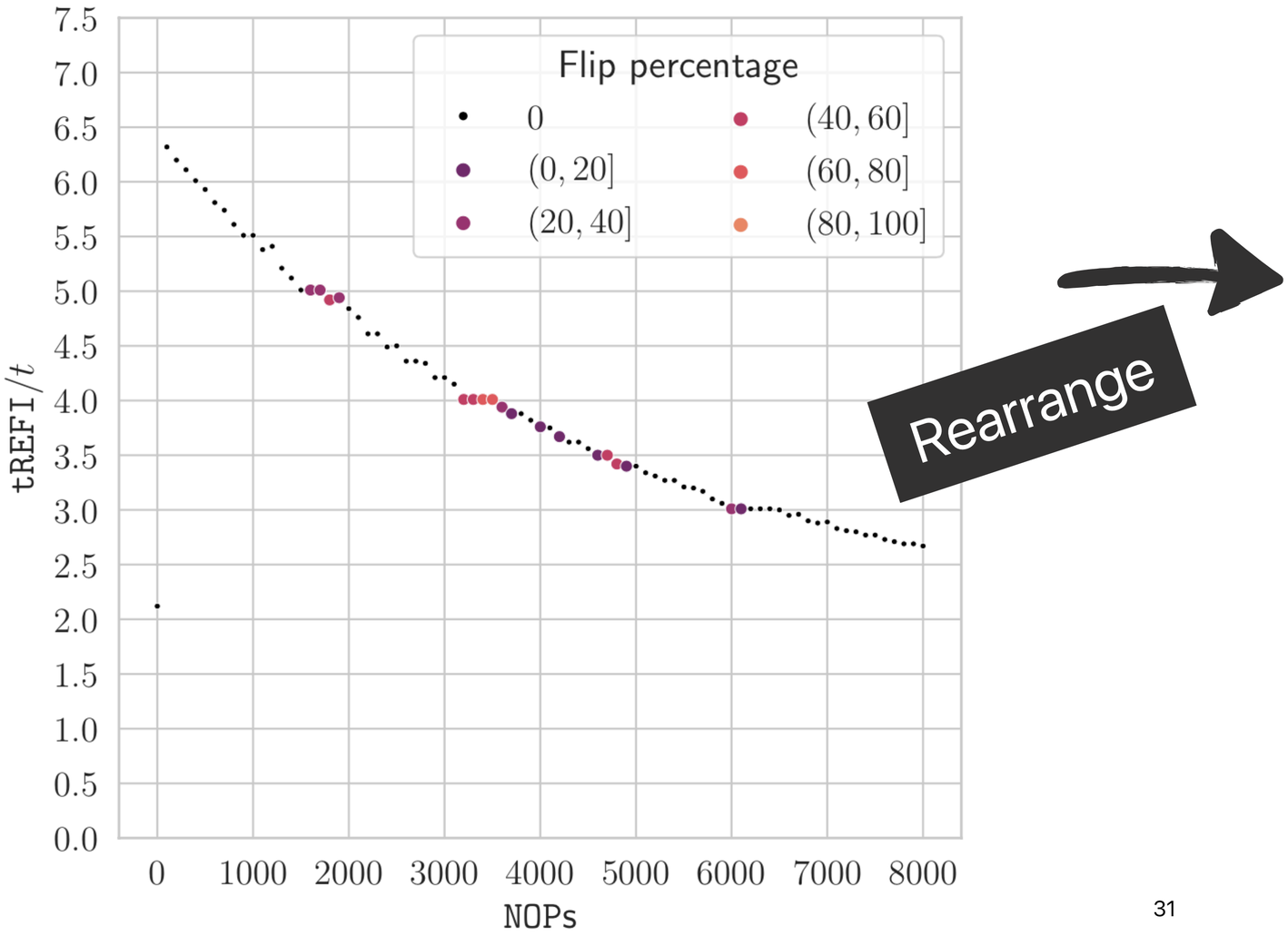
# Synchronizing in JavaScript May Be Difficult

## Can we do better?



# Synchronizing in JavaScript May Be Difficult

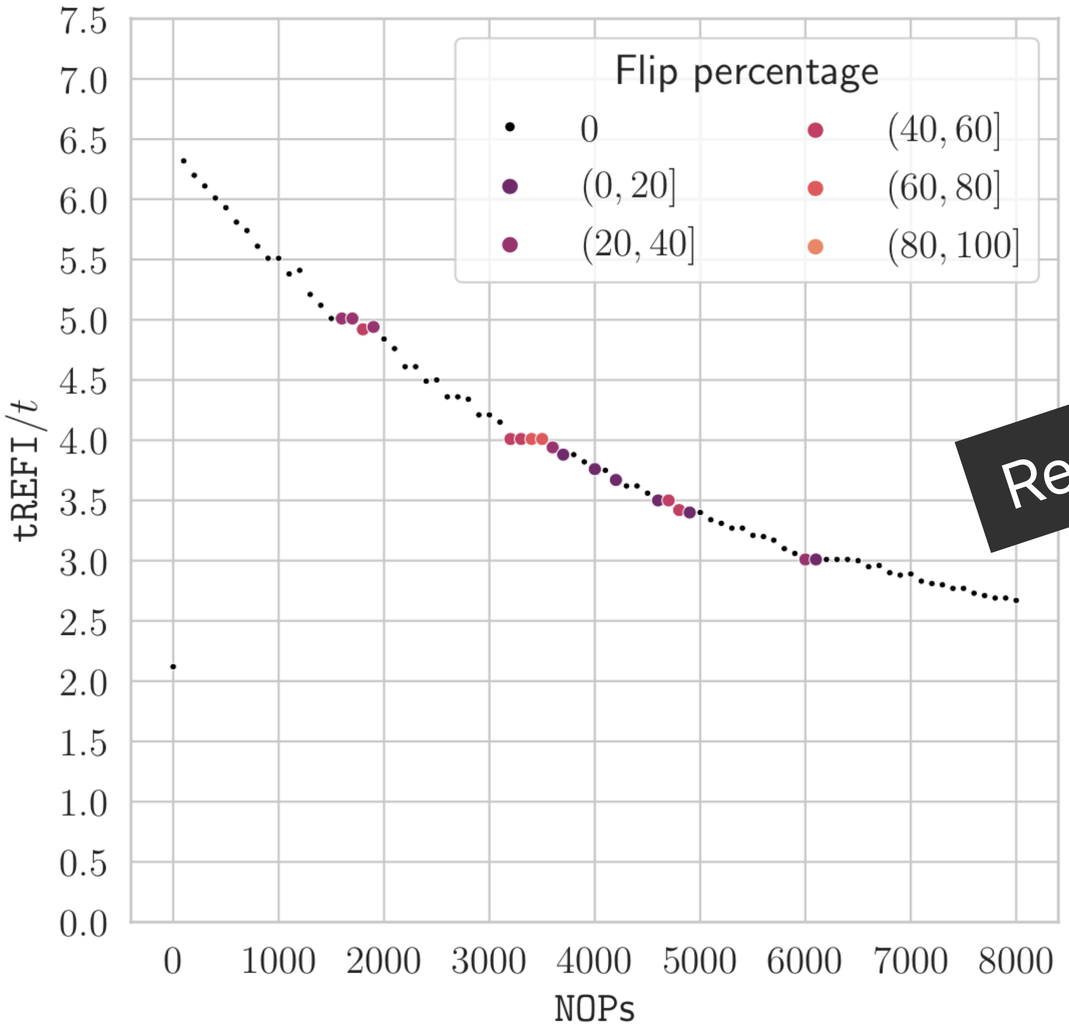
## Can we do better?



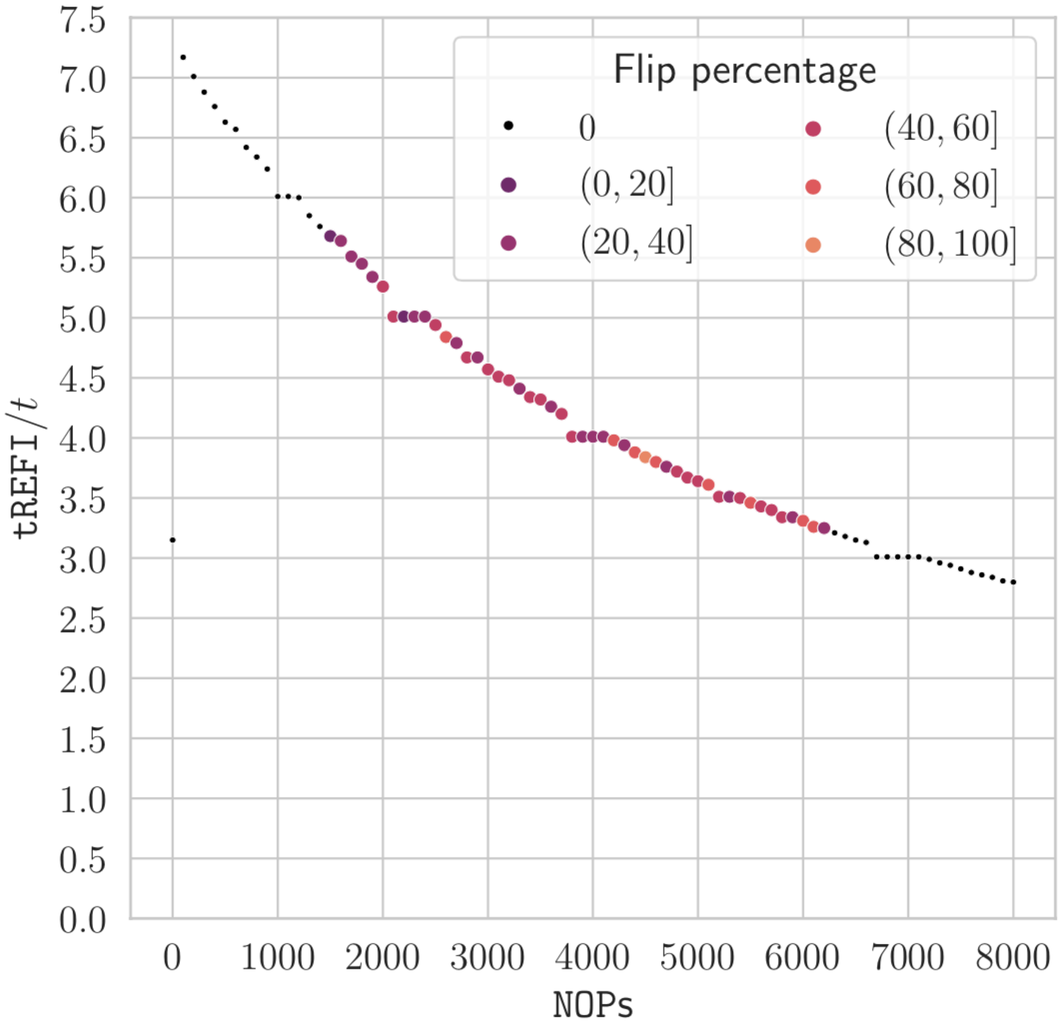

Rearrange

# Synchronizing in JavaScript May Be Difficult

## Can we do better?

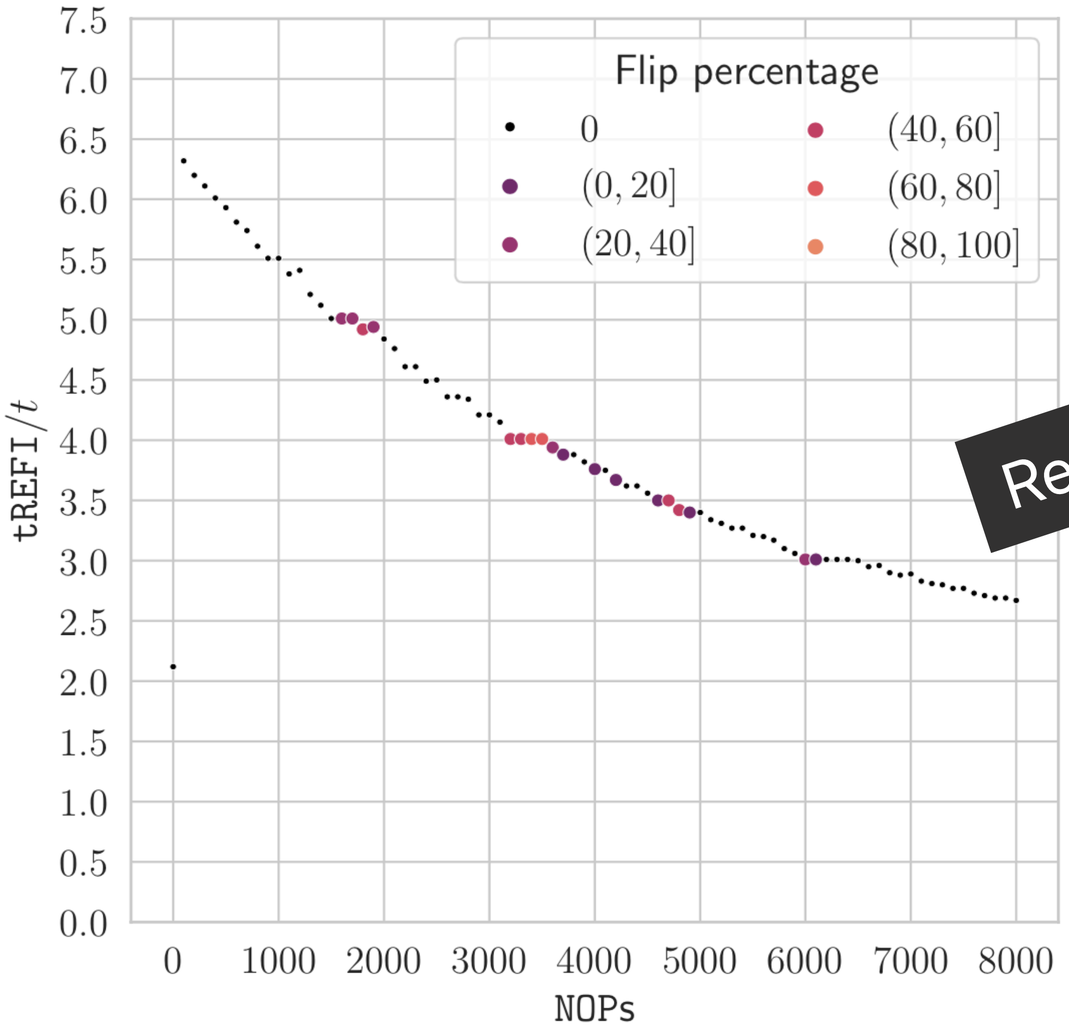



Rearrange

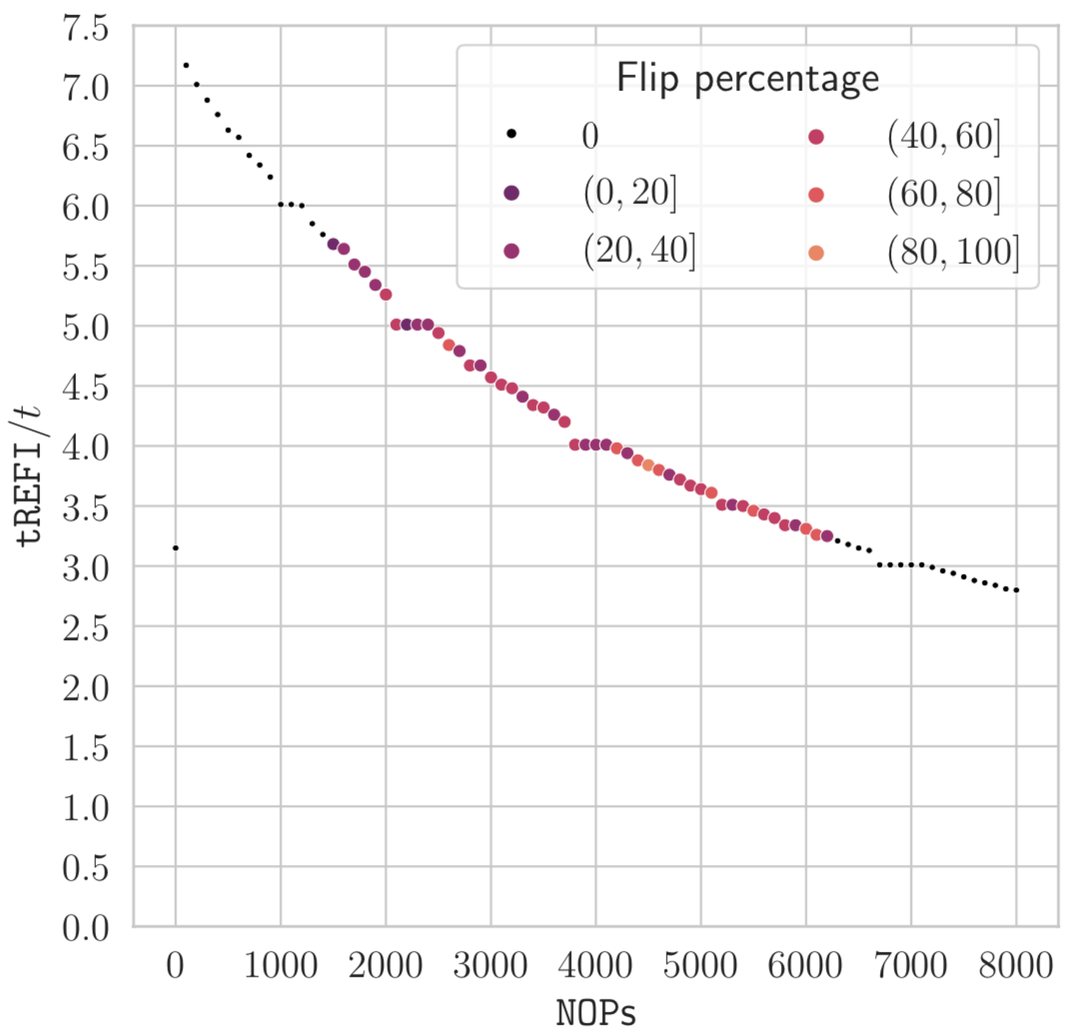


# Synchronizing in JavaScript May Be Difficult

## Can we do better? Yes!




Rearrange



# Results

## Pattern characteristics

| System | Best TRRespass | Aggressors | Total length incl. hits |
|---|---|---|---|
| 0 | 19-sided | 18 | 96 |
| 1 | 10-sided | 10 | 160 |
| 2 | 3-sided | 4 | 64 |

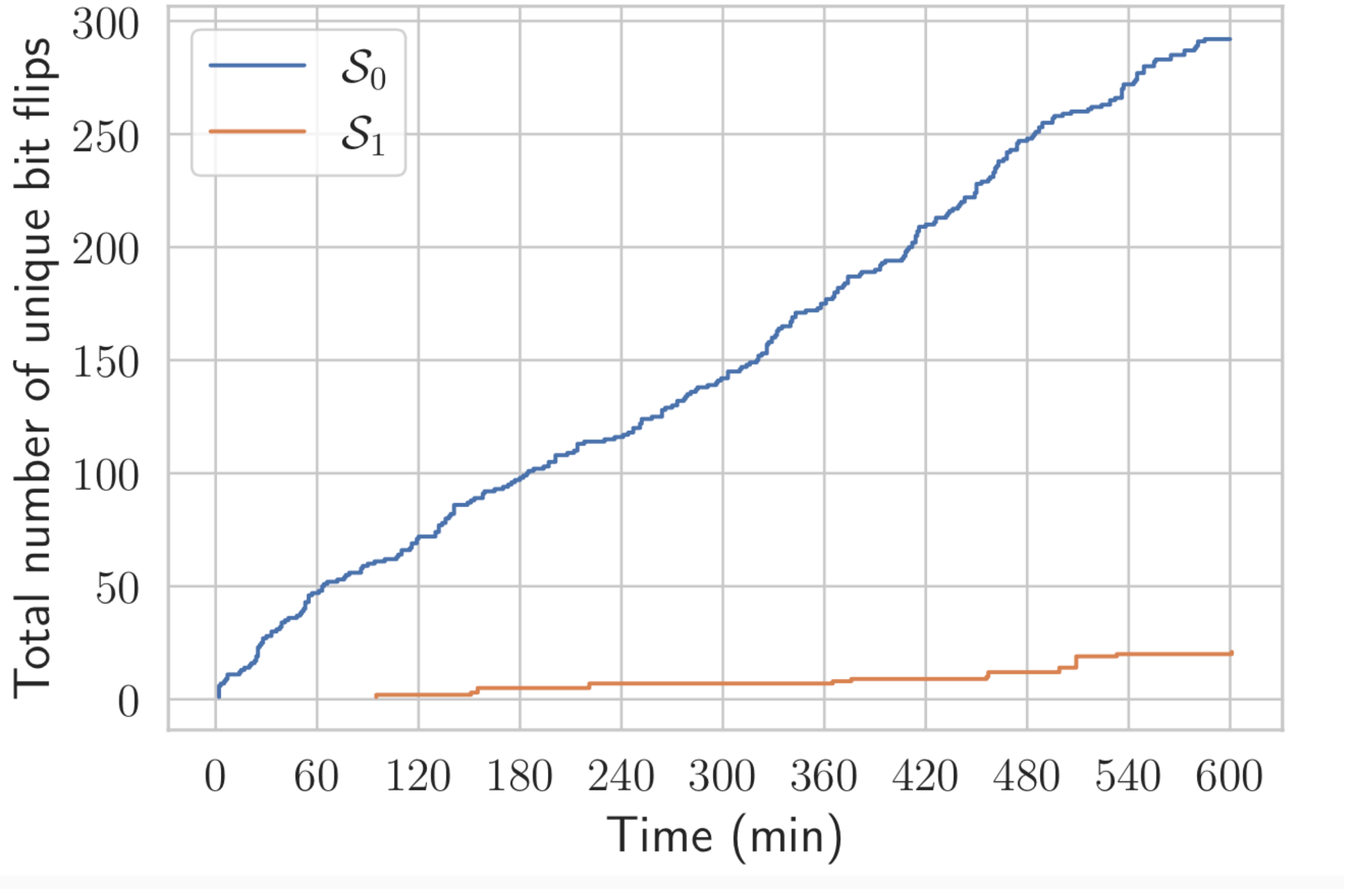
# Results

## Bit flips

| System | Native flips | NOPs | JavaScript flips | XORs |
|---|---|---|---|---|
| 0 | Yes | 1500-6200 | Yes | 300-900 |
| 1 | Yes | 100-1900, 3500-3700 | Yes | 0-400, 700 |
| 2 | Yes | 100 | — | — |

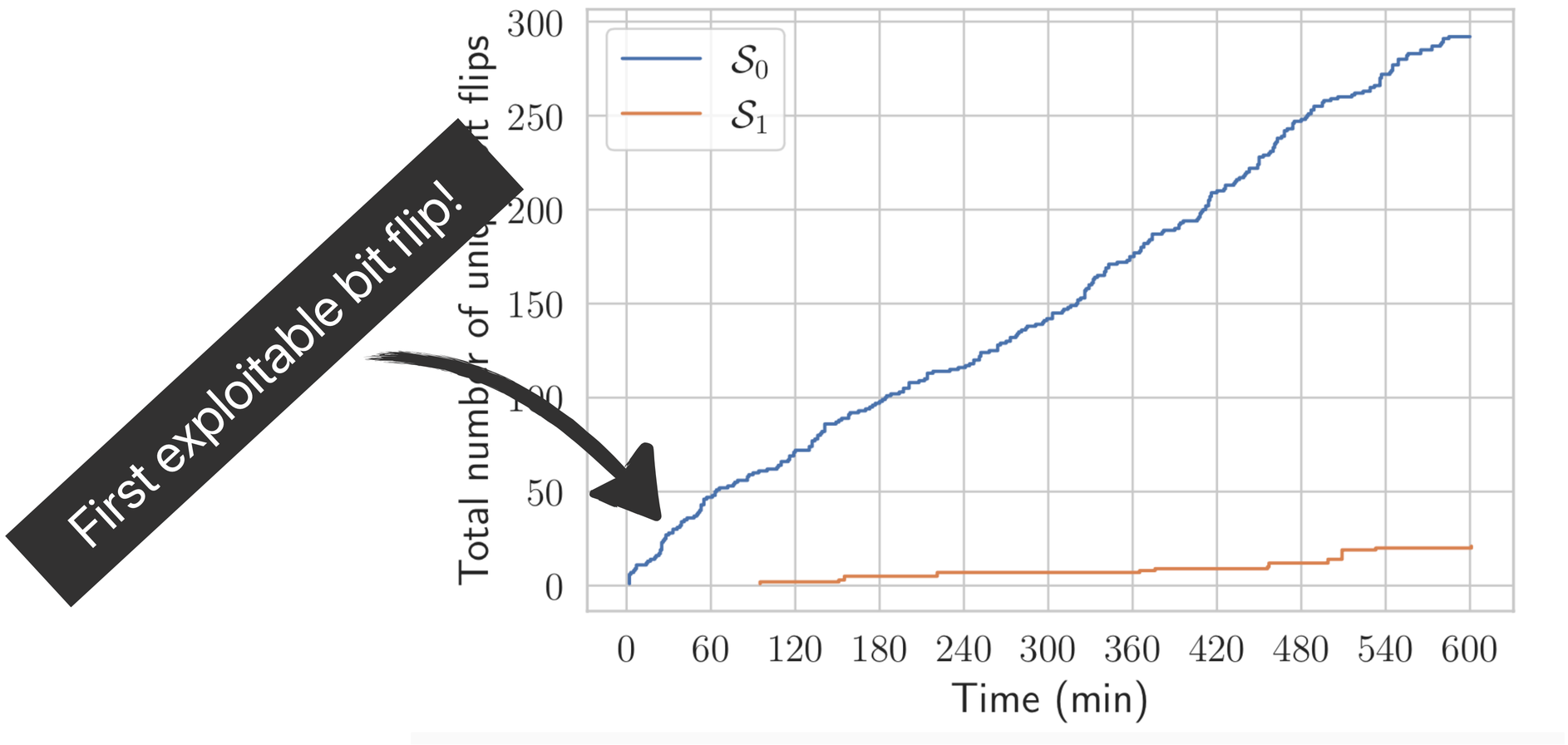
# Results

## Bit flips over time



# Results

## Bit flips over time



# Conclusion

## Part IV

# Many-sided Rowhammer in JavaScript Is...

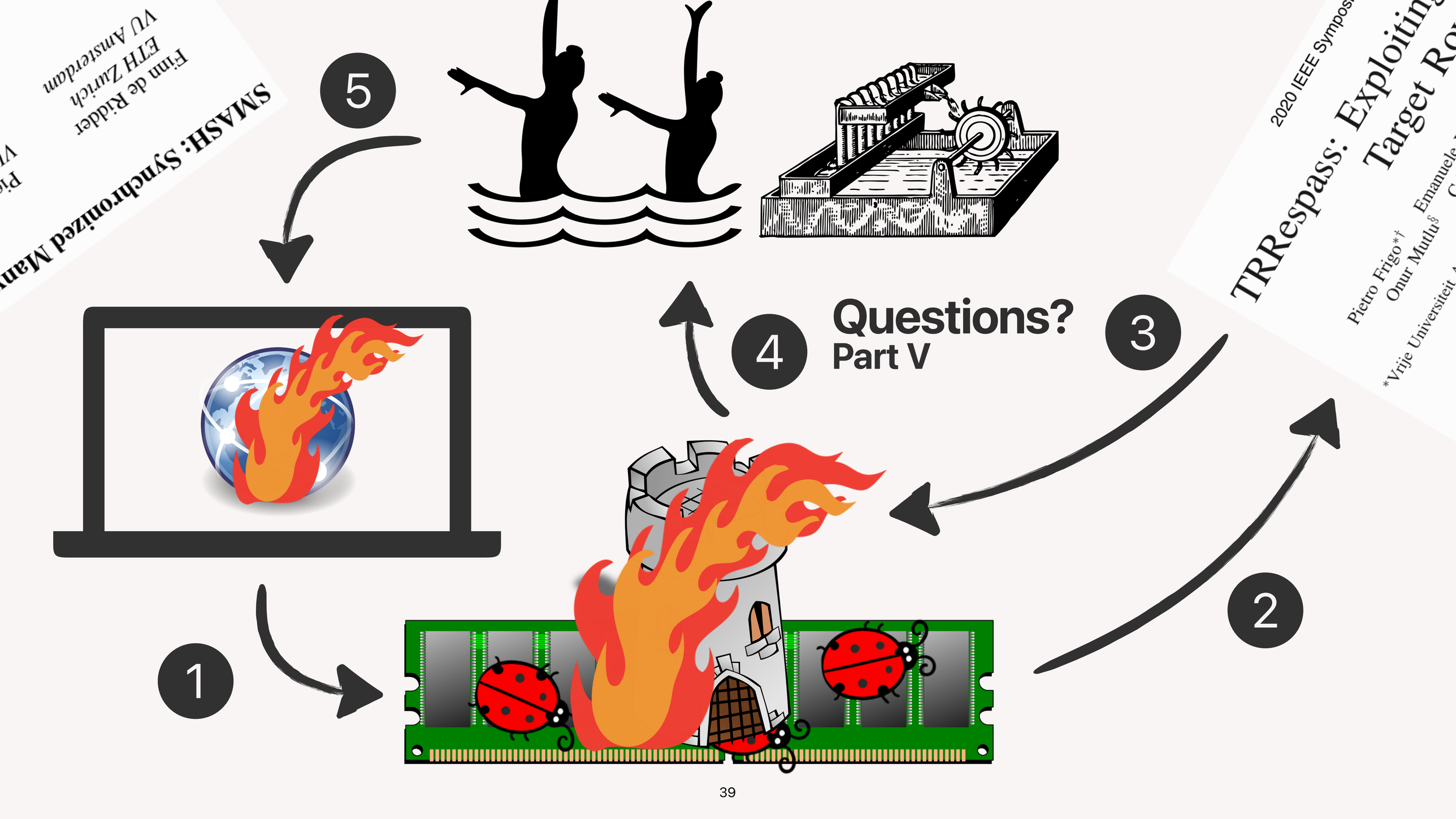
## JS we can

- Possible if you create a synchronized self-evicting pattern
- Despite all hardware and software mitigations
- Re-enables past JavaScript-based Rowhammer attacks in 2021

# Many-sided Rowhammer in JavaScript Is...

## JS we can

- Possible if you create a synchronized self-evicting pattern
- Despite all hardware and software mitigations
- Re-enables past JavaScript-based Rowhammer attacks in 2021 (unfortunately)


SMASH: Synchronized Man
Finn de Ridder
ETH Zurich
VU Amsterdam
2020 IEEE Symposi
TRRespass: Exploiting
Target Ro
Pietro Frigo*†
Onur Mutlu§
Emanuele
*Vrije Universiteit
5
4
Questions?
Part V
3
2
1

Source code available at https://github.com/vusec/smash

Demo at https://www.youtube.com/watch?v=k2D4D-kF-ic

# Questions?

## Part V