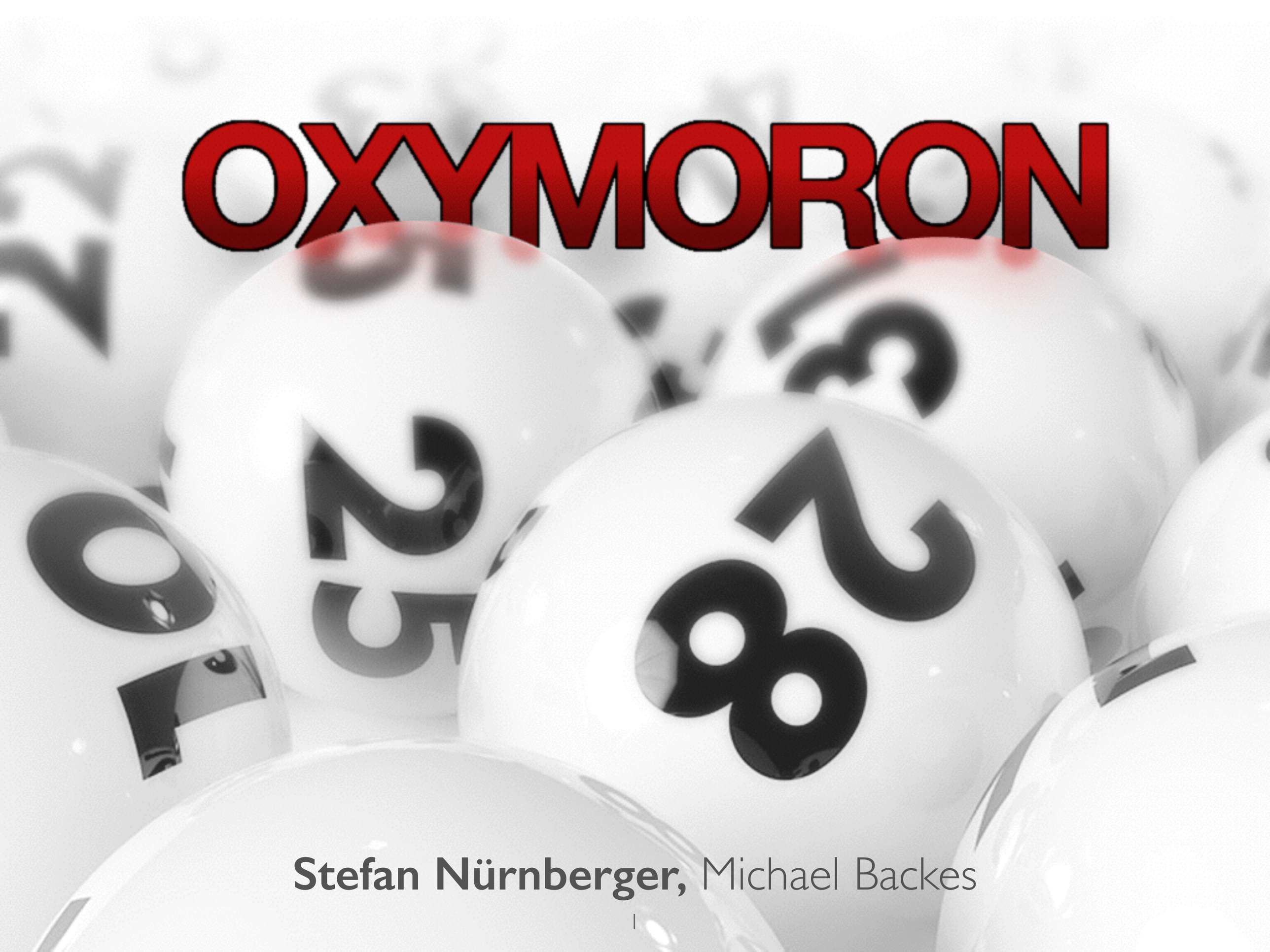
OXYMORON



Stefan Nürnberger, Michael Backes

STEFAN

PhD

SAARLAND UNIVERSITY



GERMANY


Göteborg
North Sea
Danmark
Denmark
United
Kingdom
Hamburg
Ireland
Éire
Birmingham
London
België
Belgique
Belgium
Deutschland
Germany
Paris
Vaduz
France
Milano
Torino
Bay of Biscay
Hrvatska
Croatia
Budapest
Eesti
Estonia
Latvija
Latvia
România
Romania
Србија

EARLY 2011



EARLY 2011

You are an IT Security Competence Centre!



SAARLAND UNIVERSITY

EARLY 2011

Center for **I**T **S**ecurity **P**rivacy and **A**ccountability

CISPA



Cyber Intelligence Sharing and Protection Act

MAKING FINE-GRAINED MEMORY RANDOMISATION PRACTICAL BY ALLOWING CODE SHARING

OXYMORON

Oxymoron /ˌɒk.sɪˈmɔː.rɒn/ (noun)
Greek. A figure of speech that combines contradictory terms.

OXYMORON

Code Reuse Attacks

Fine-Grained Randomization

Modern OS

Shared Libraries

ILR: Where'd My Gadgets Go?

Software Diversity

E unibus pluram

Smashing the Gadgets

Binary Stirring



Oxymoron

CODE-REUSE ATTACKS

Adversary



Program

I'm the most lovely and benign program you can imagine. I embody functionality like document analysis of different types. I'm not harm ful in any way.

CF

The same program

I am a lovely and benign program that just runs on your computer without the intent of doing anything harmful. Not even my words are malicious.

ASLR

Adversary

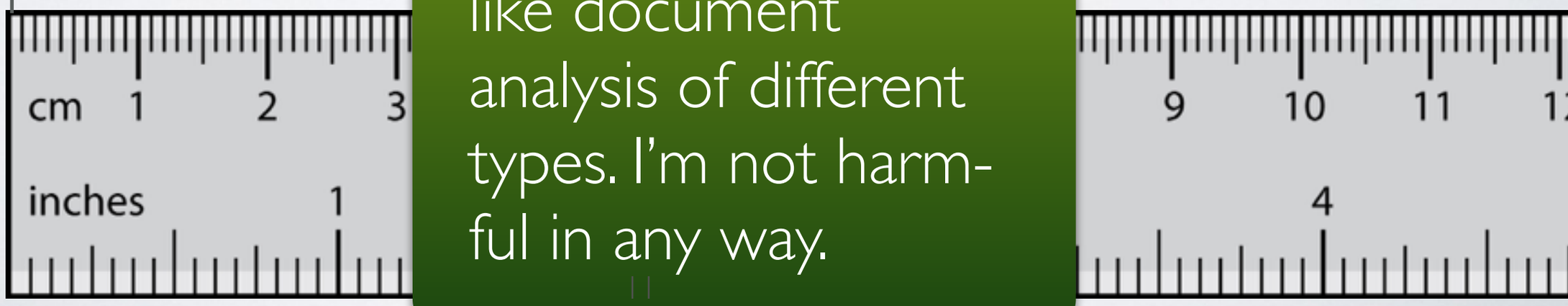


Program

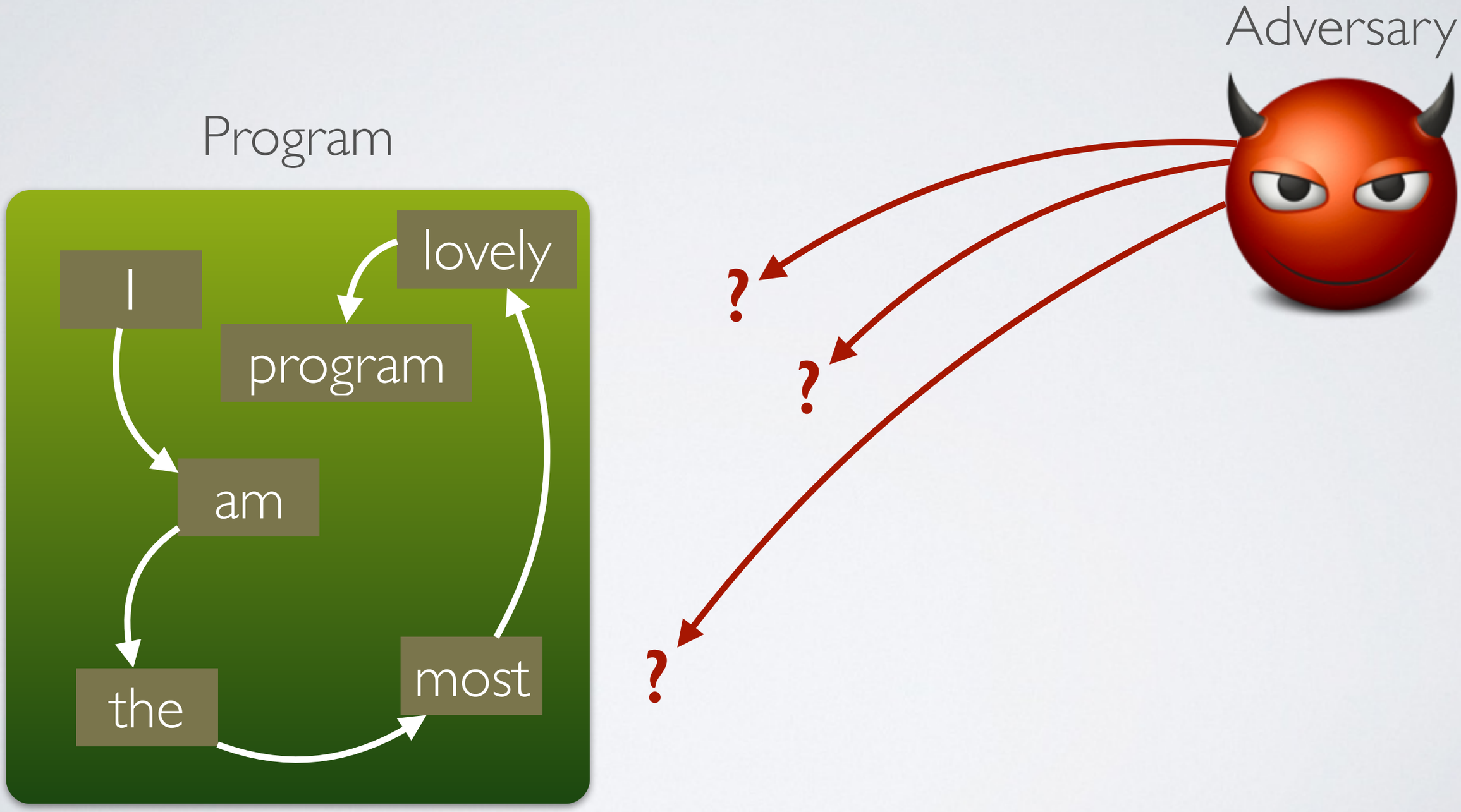


?

I'm the most lovely and benign program you can imagine. I embody functionality like document analysis of different types. I'm not harm-ful in any way.

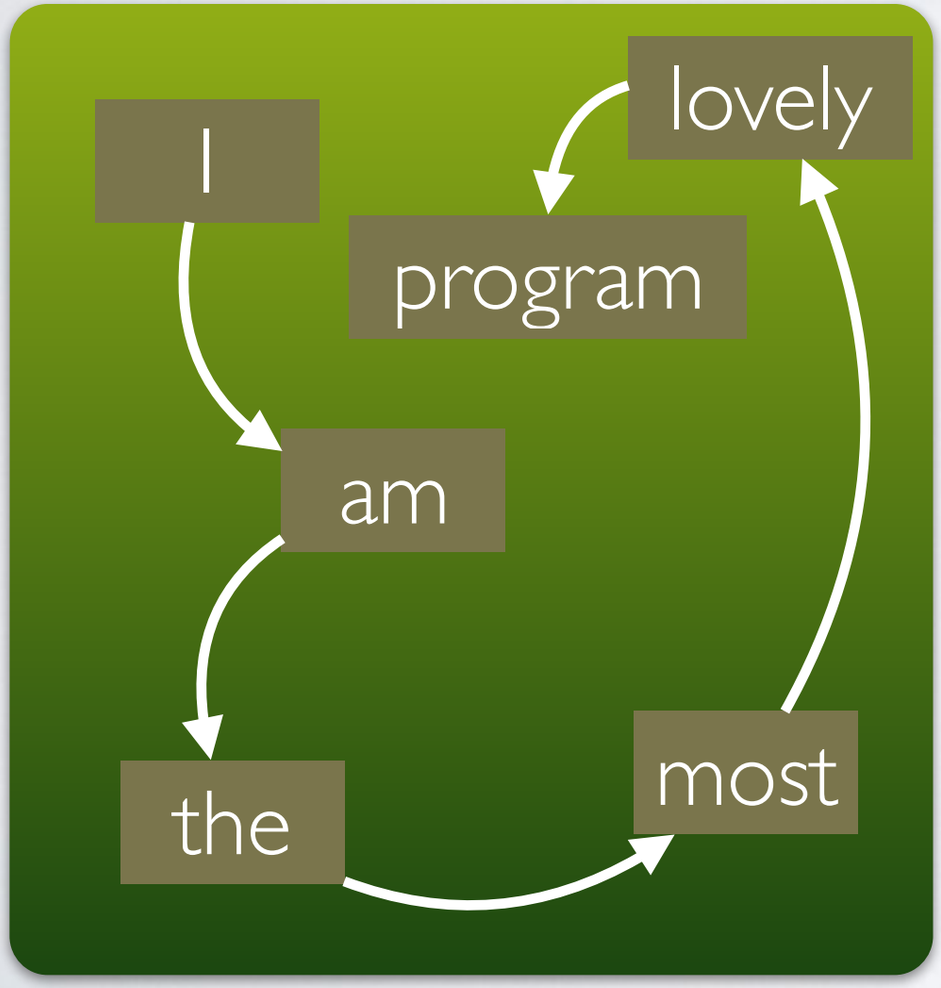


FINE-GRAINED RANDOMIZATION

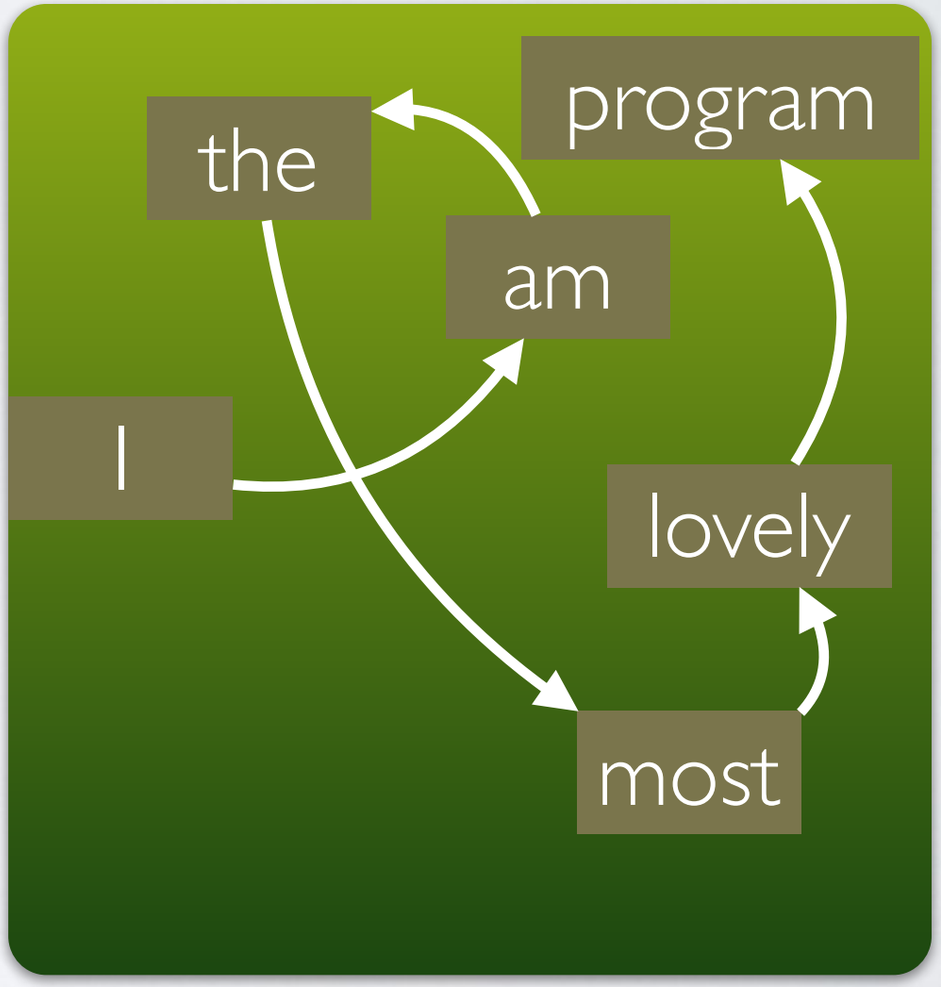


CROSS-PROCESS RANDOMIZATION

Process 1



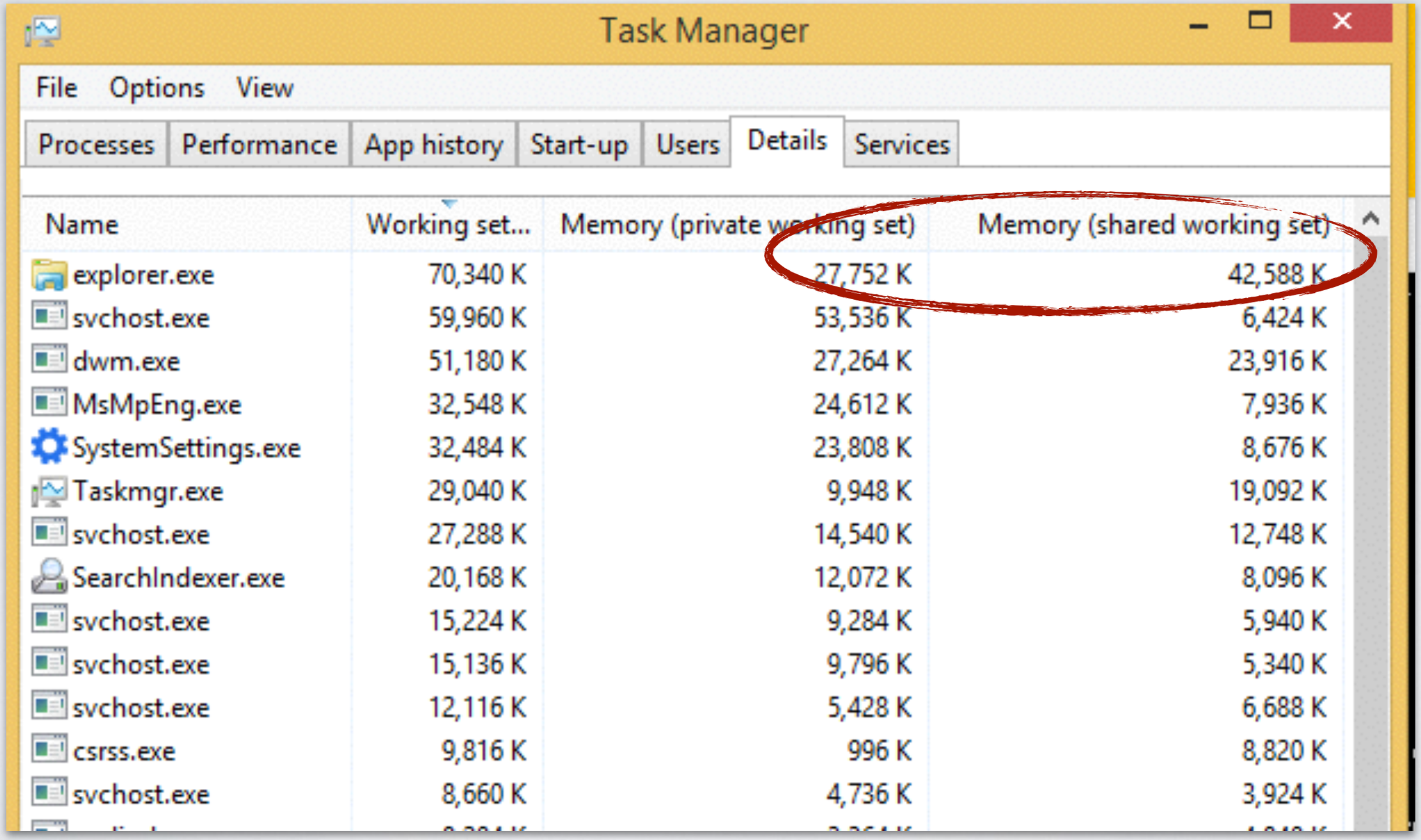
Process 2



SO WE'VE SOLVED THE PROBLEM.

RIGHT?

NOT QUITE


Task Manager
File Options View
Processes Performance App history Start-up Users Details Services
Name Working set... Memory (private working set) Memory (shared working set)
explorer.exe 70,340 K 27,752 K 42,588 K
svchost.exe 59,960 K 53,536 K 6,424 K
dwm.exe 51,180 K 27,264 K 23,916 K
MsMpEng.exe 32,548 K 24,612 K 7,936 K
SystemSettings.exe 32,484 K 23,808 K 8,676 K
Taskmgr.exe 29,040 K 9,948 K 19,092 K
svchost.exe 27,288 K 14,540 K 12,748 K
SearchIndexer.exe 20,168 K 12,072 K 8,096 K
svchost.exe 15,224 K 9,284 K 5,940 K
svchost.exe 15,136 K 9,796 K 5,340 K
svchost.exe 12,116 K 5,428 K 6,688 K
csrss.exe 9,816 K 996 K 8,820 K
svchost.exe 8,660 K 4,736 K 3,924 K

OBSERVATION

System Monitor

Processes | Resources | File Systems

Load averages for the last 1, 5, 15 minutes: 0,36, 0,39, 0,21 — Refresh — View

Process Name	Virtual Memory	Resident Memory	% CPU	ID
compiz	396,3 MiB	171,6 MiB	1	1831
Xorg	150,8 MiB	46,6 MiB	0	816
evolution-calendar-factory	116,9 MiB	29,6 MiB	0	1896

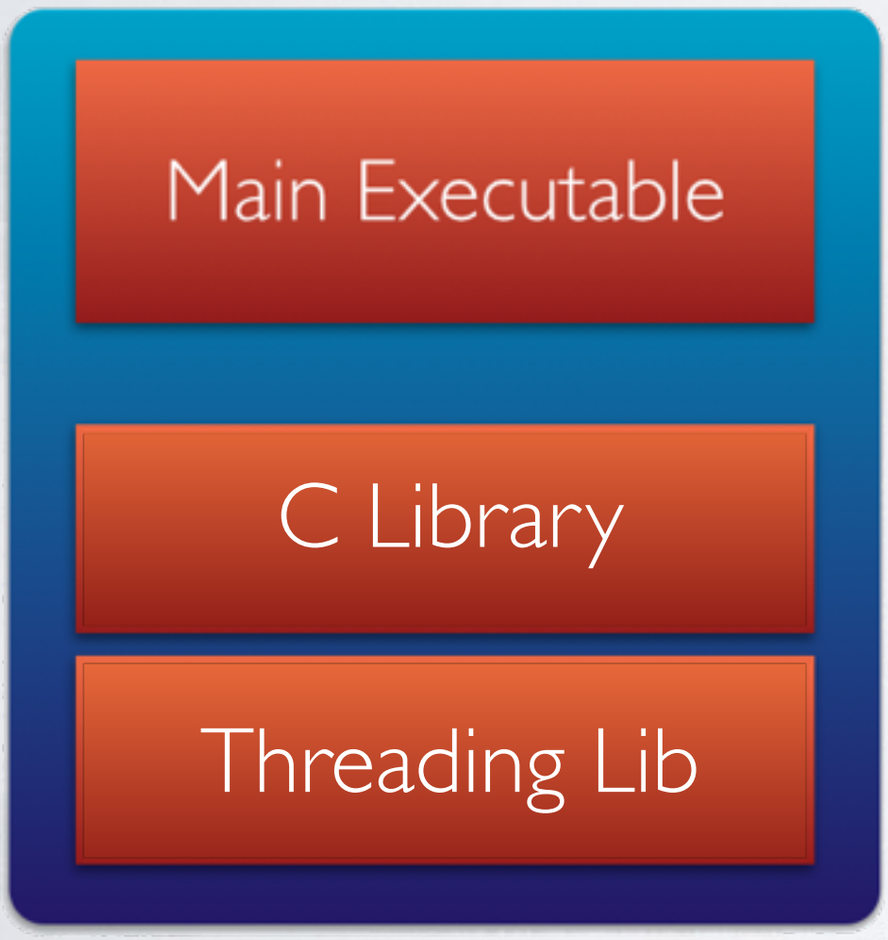
Activity Monitor (All Processes)

CPU | Memory | Energy | Disk | Network

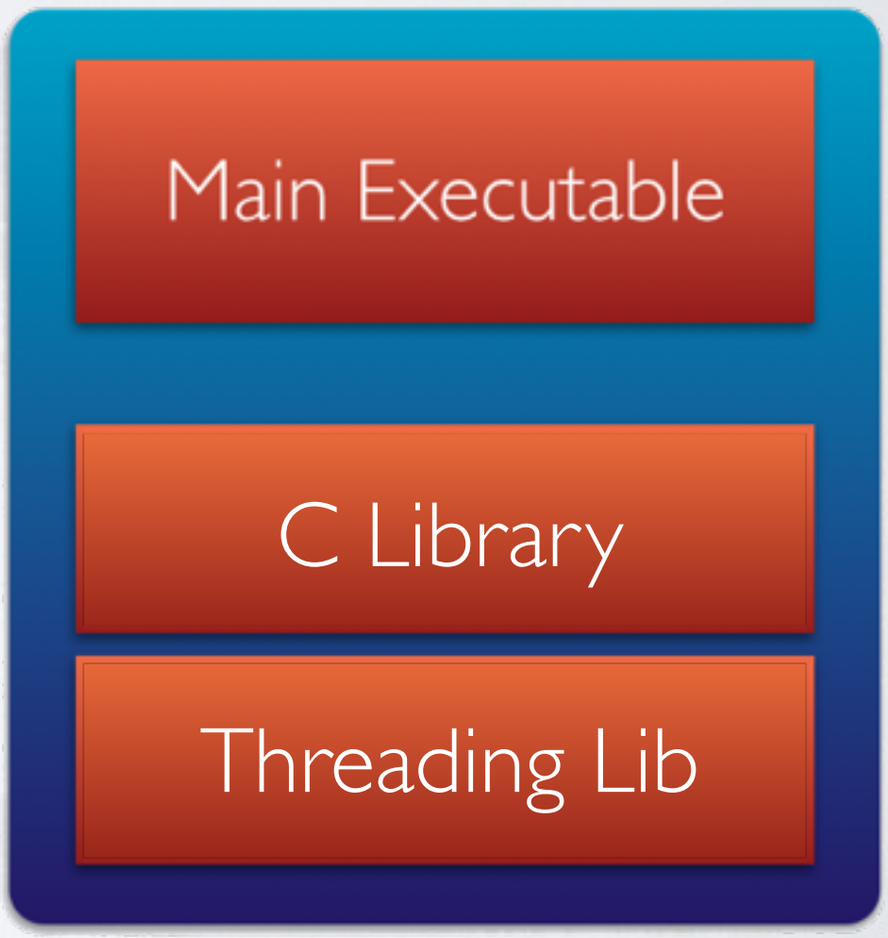
Memory	Threads	Ports	PID	User	Private Mem	Shared Mem
302.7 MB	6	443	479	stefan	255.3 MB	72.2 MB
126.0 MB	31	244	304	stefan	119.5 MB	35.9 MB
79.8 MB	36	577	303	stefan	74.7 MB	107.5 MB
35.4 MB	6	65	219	root	70.9 MB	12.7 MB
55.2 MB	10	81	999	stefan	54.5 MB	81.2 MB
57.9 MB	5	510	115	_window	52.2 MB	111.4 MB
56.4 MB	10	81	3421	stefan	51.8 MB	91.0 MB
44.3 MB	10	84	1167	stefan	44.1 MB	88.5 MB

SHARING CODE AND DATA

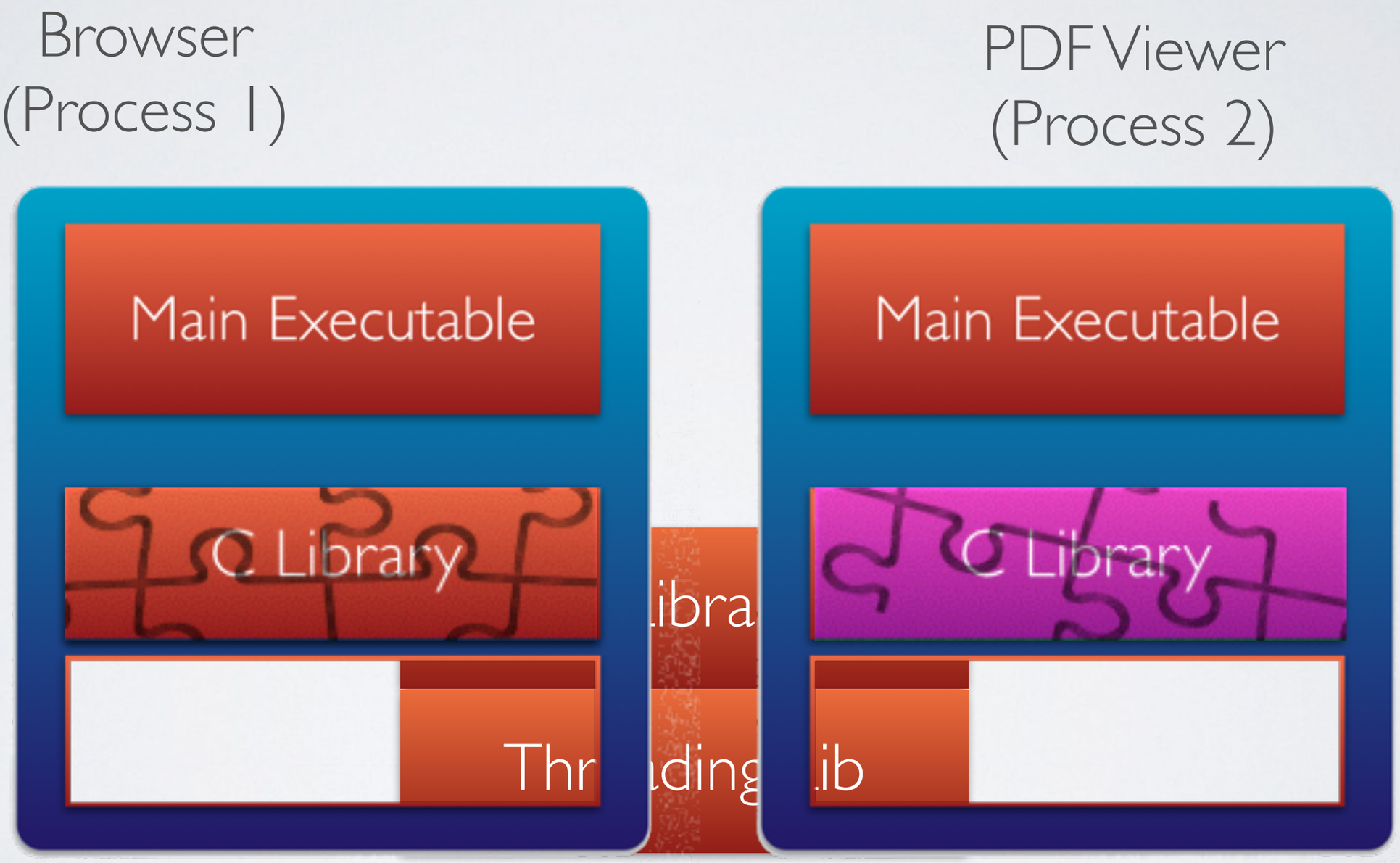
Browser
(Process 1)



PDF Viewer
(Process 2)

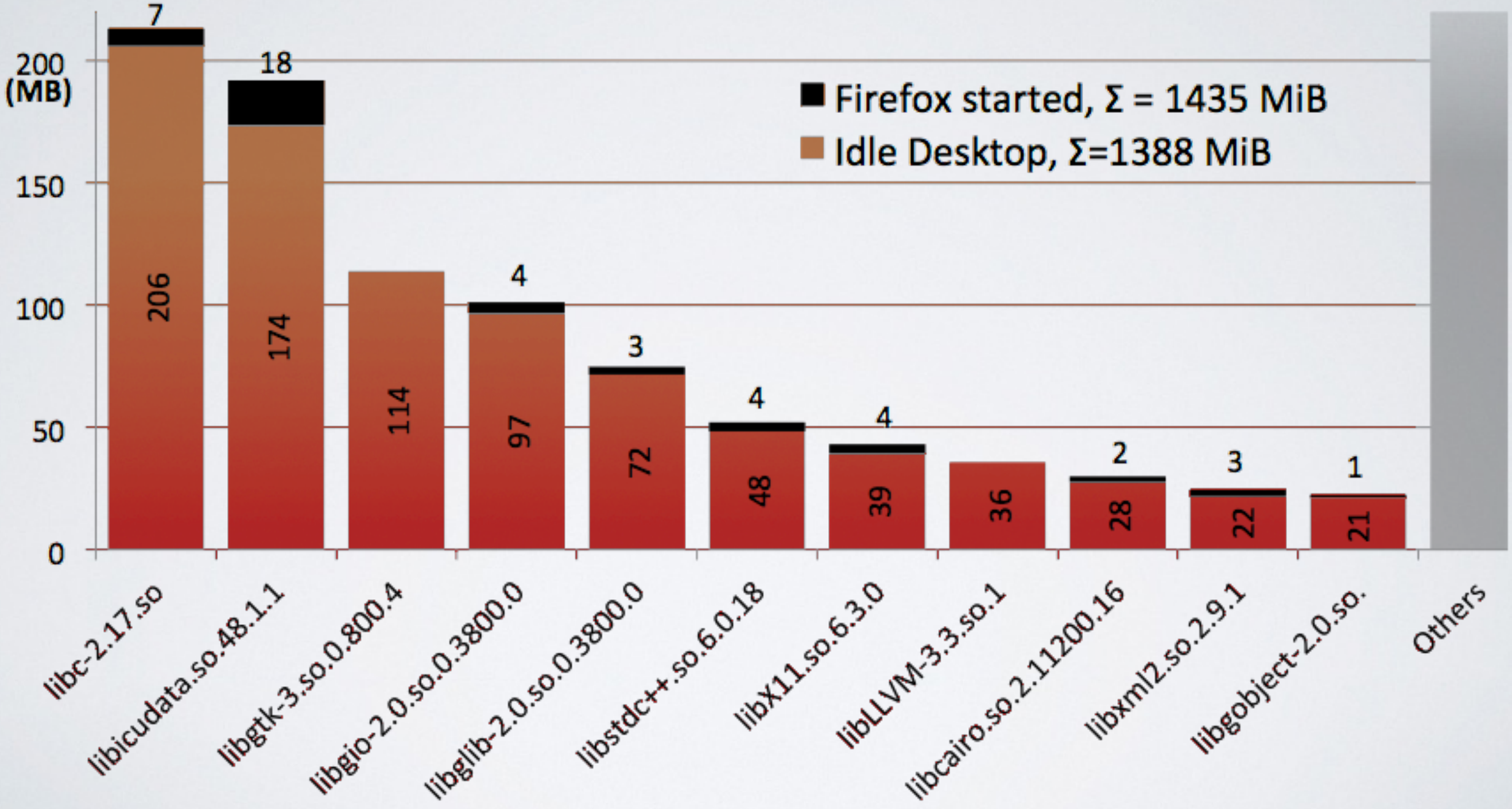


SHARING CODE AND DATA



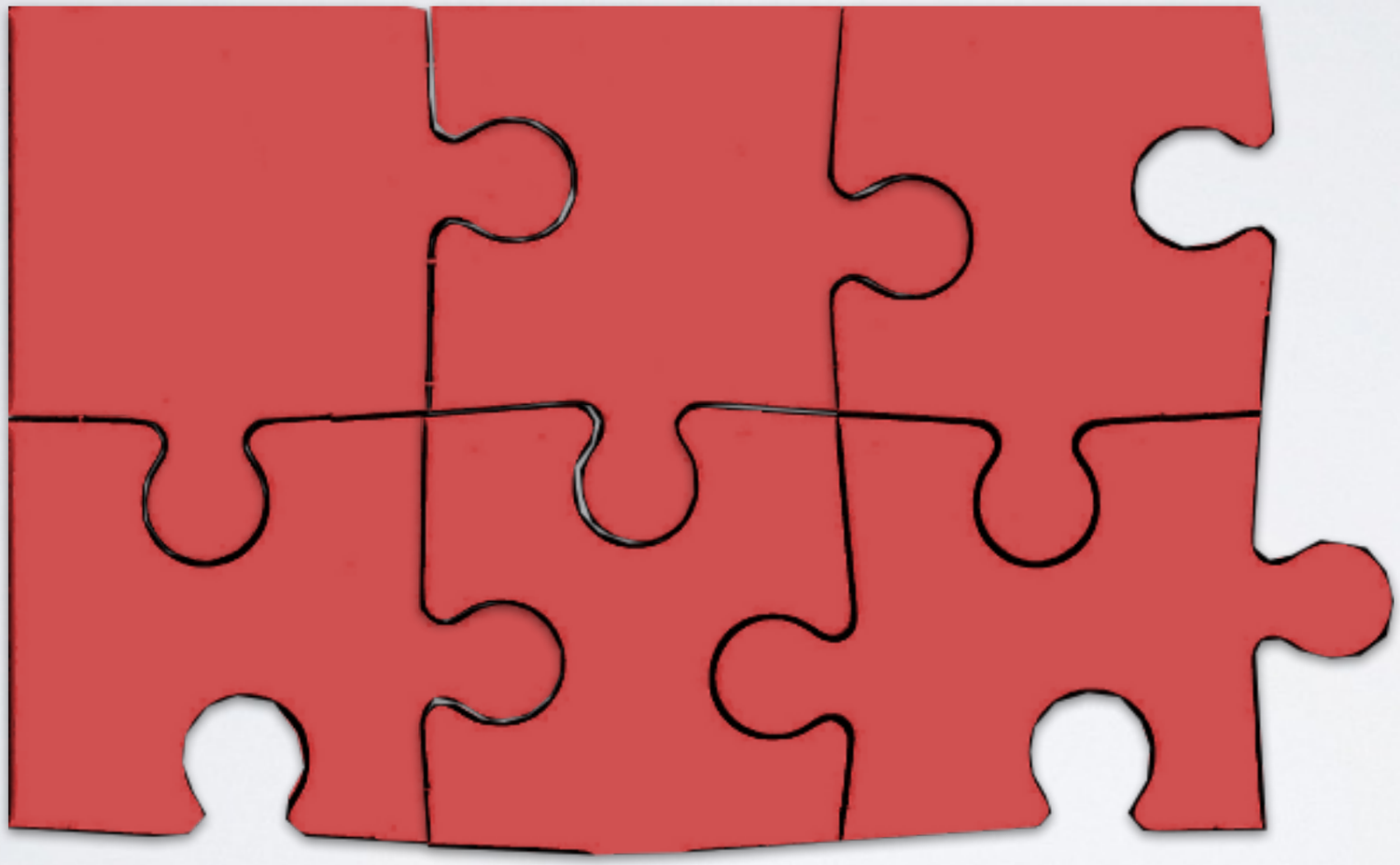
IMPACT

234 Processes



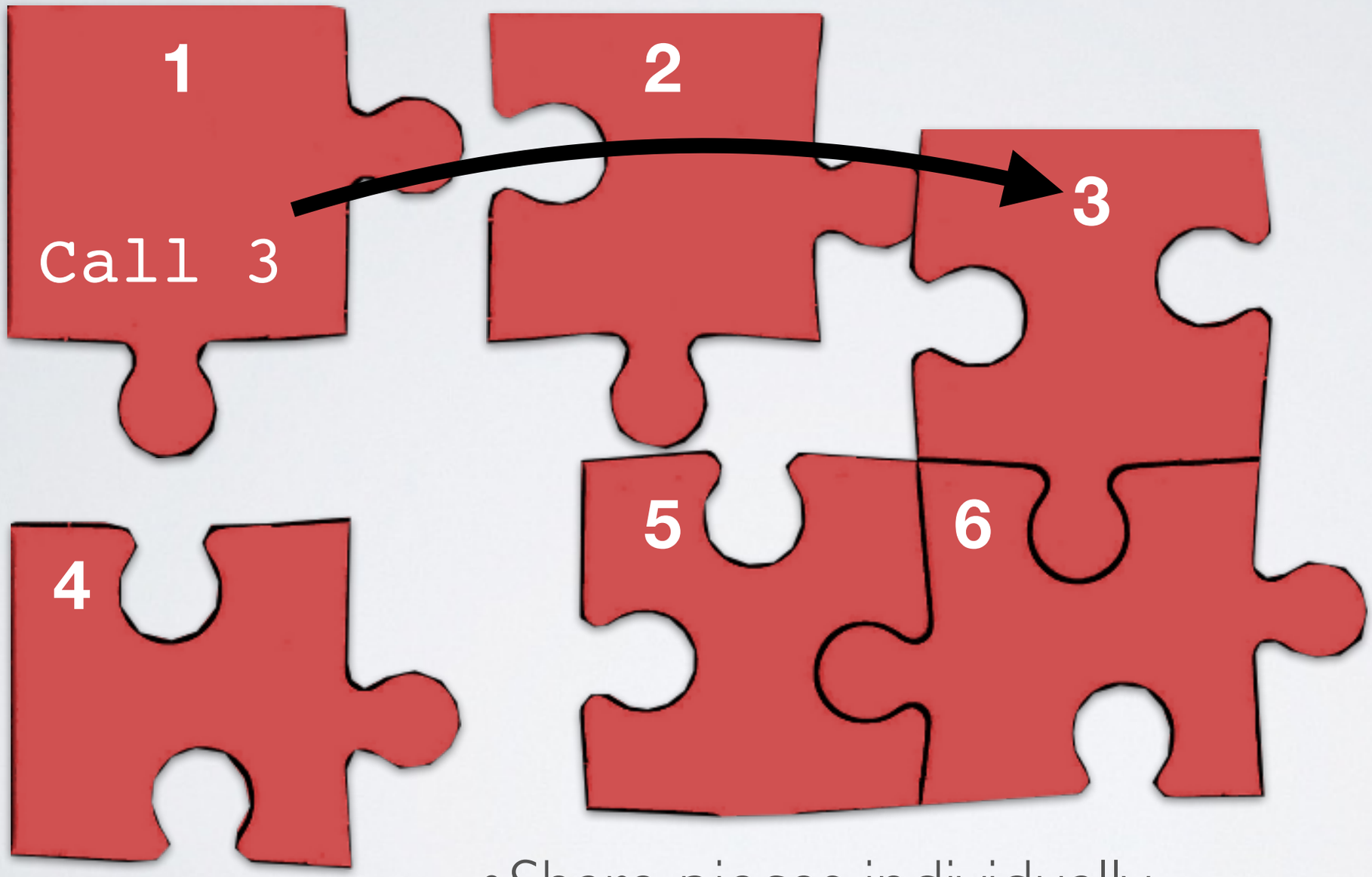
ON THE QUEST TO A SOLUTION

ONE POSSIBLE SOLUTION



- Split Code in pieces

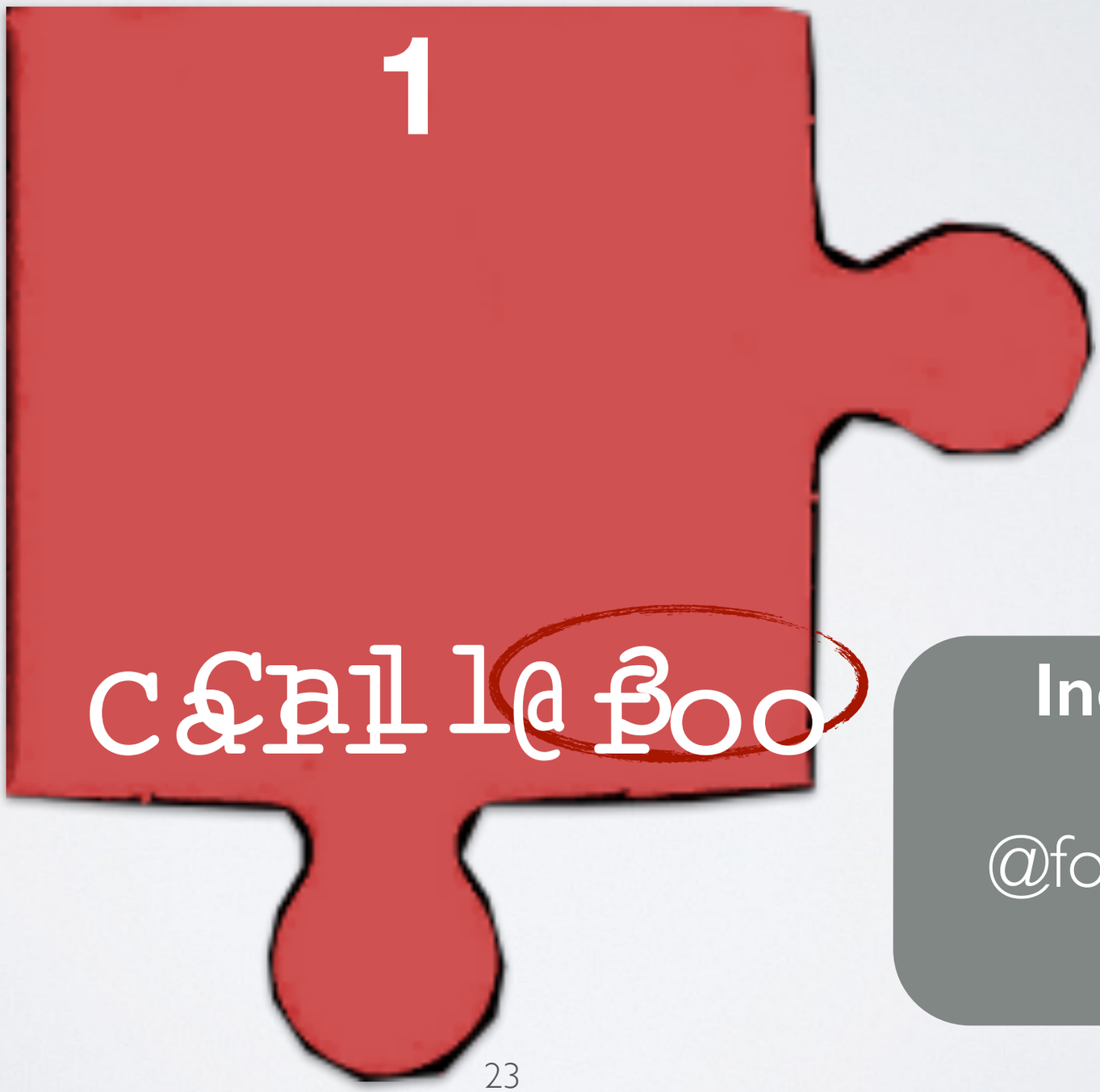
ONE POSSIBLE SOLUTION



Problem:
They still need to be connected!

- Share pieces individually
- Enough entropy

VALUES ENCODED IN CODE



Indirection

@foo: 0x82ab7

OXYMORON

- Enables individual pieces to be shared
- Each process can be randomised separately
- Hides all addresses from the attacker
- It's a modified x86 calling convention

PALACE

(**P**OSITION **A**ND **L**AYOUT **A**GNOSTIC **C**OD**E**)

PALACE

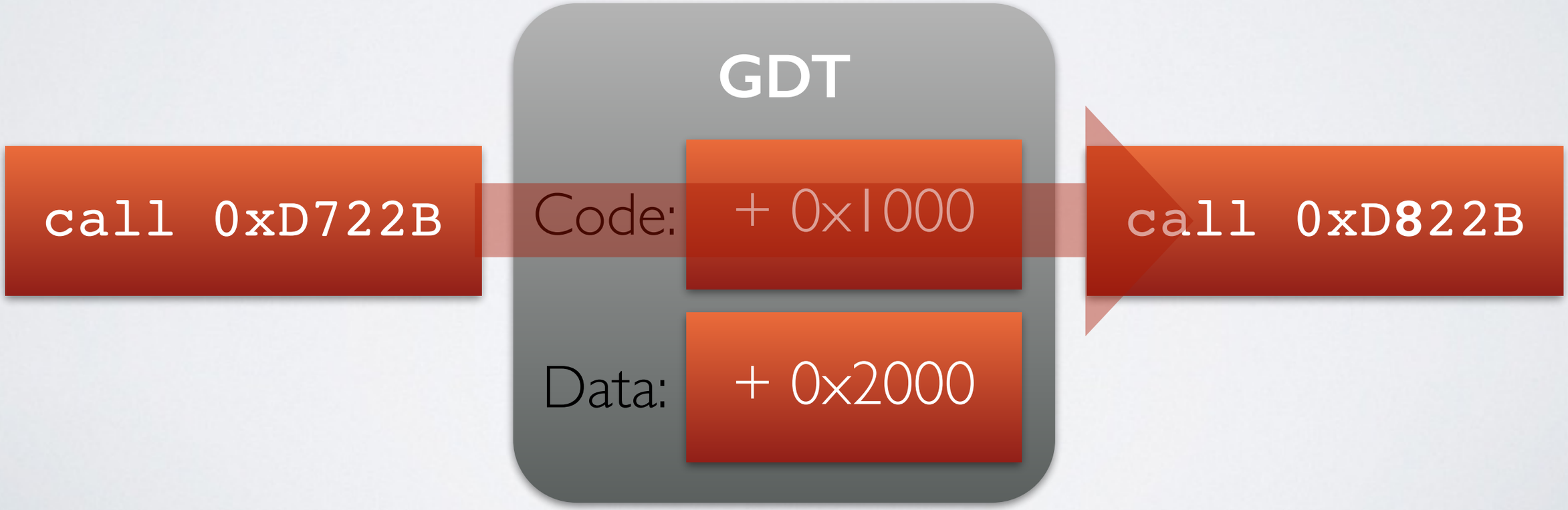
This is how x86 is structured



16 Bit Feature: Segmentation

SEGMENTATION

- To move programs in memory before MMUs existed
- Dedicated registers to add offsets to code/data/stack



X86 INDIRECT CALLS

```
call *0x1234
```

Address	Value
0x1234	**0x9876**
0x1238	…
0x123C	…
…	…

call 0x9876

0x9876

```
push %ebp
mov %esp, %ebp
```

RATTLE

Randomisation-**A**gnostic **T**ranslation Tab**LE**

GDT

Code: + 0x1230

```
call *0x4
```

```
call *0x1234
```

RATTLE

Address	Value
0x1234	**0x9876**
0x1238	…
0x123C	…
…	…

```
call 0x9876
```

0x9876

```
push %ebp
mov %esp, %ebp
```

RATTLE

Randomisation-**A**gnostic **T**ranslation Tab**LE**

GDT

fs: + 0x1000

```
call %fs:*0x4
```

RATTLE

Address	Value
0x1234	**0x9876**
0x1238	…
0x123C	…
…	…

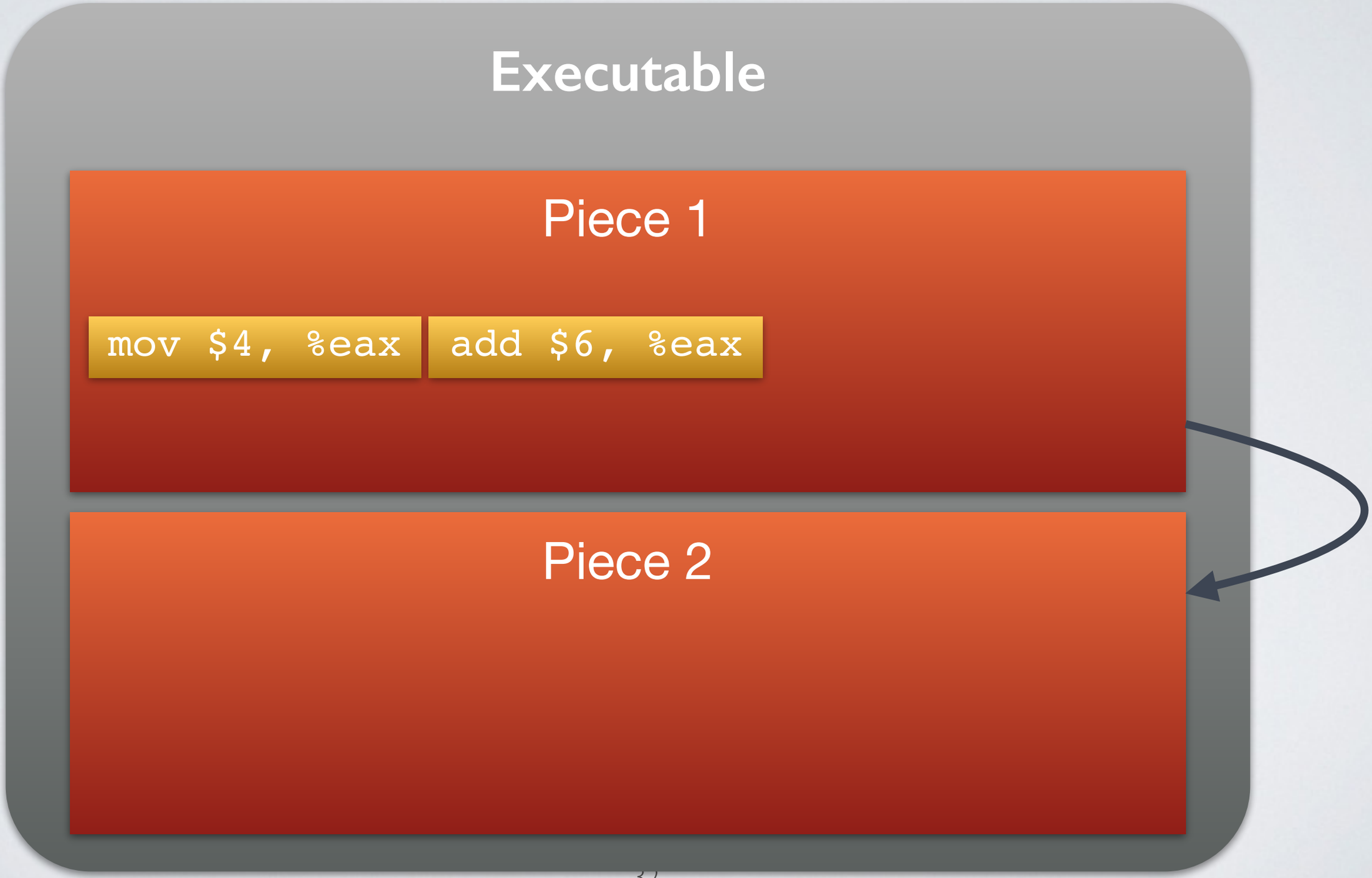
0x9876

```
push %ebp
mov %esp, %ebp
```

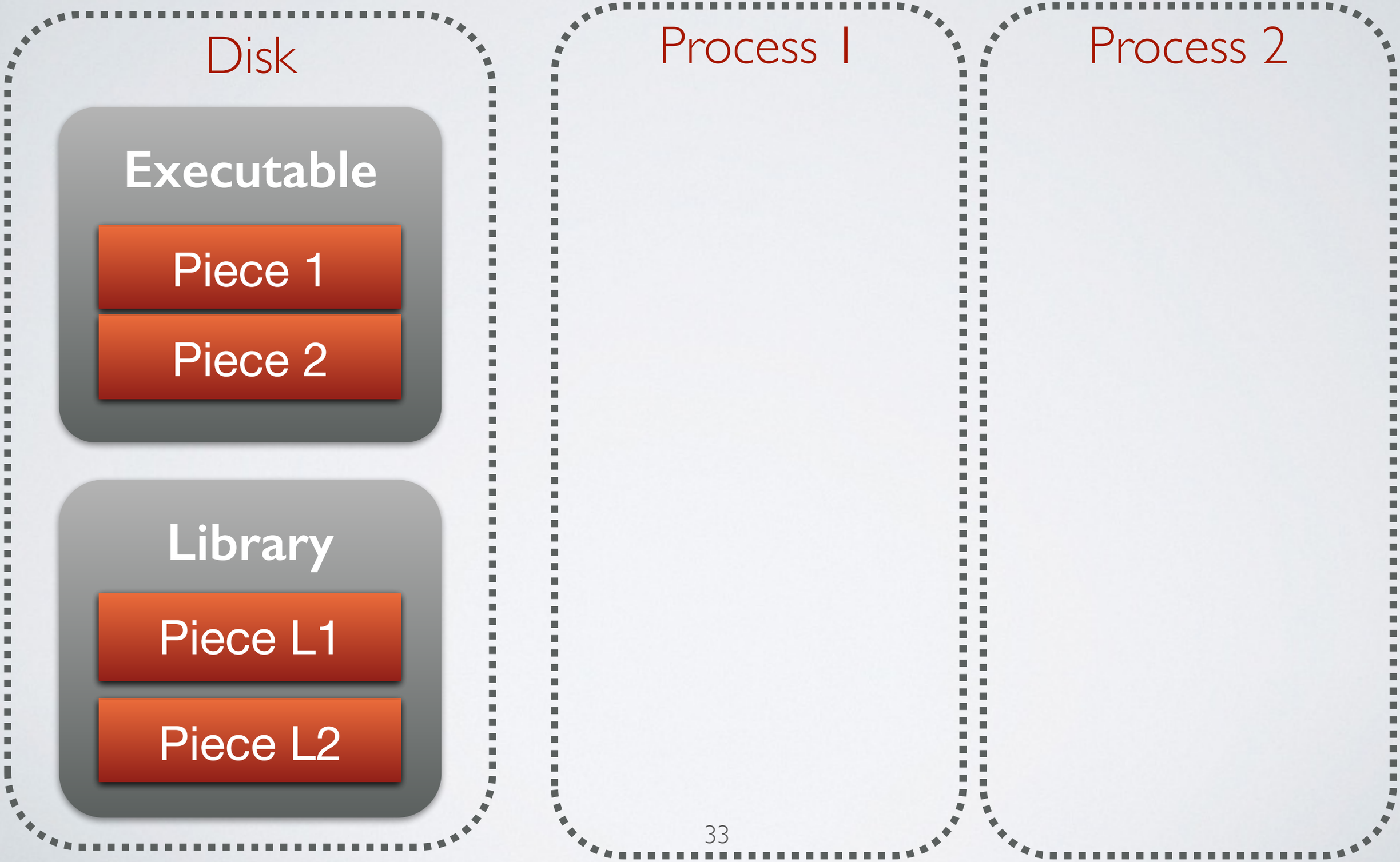
ADVANTAGES

- %fs does not reveal an address
- Encodings of **`call`** and **`mov`** are all identical across randomisations
- Code stored in a memory page can be shared among different processes
- Different processes have different permutation of memory pages
- Pure user-mode implementation (works on every Linux kernel)

OXYMORON



OXYMORON


Disk
Executable
Piece 1
Piece 2
Library
Piece L1
Piece L2
Process 1
Process 2

OXYMORON

```
mov     0x807f364,%edx
```

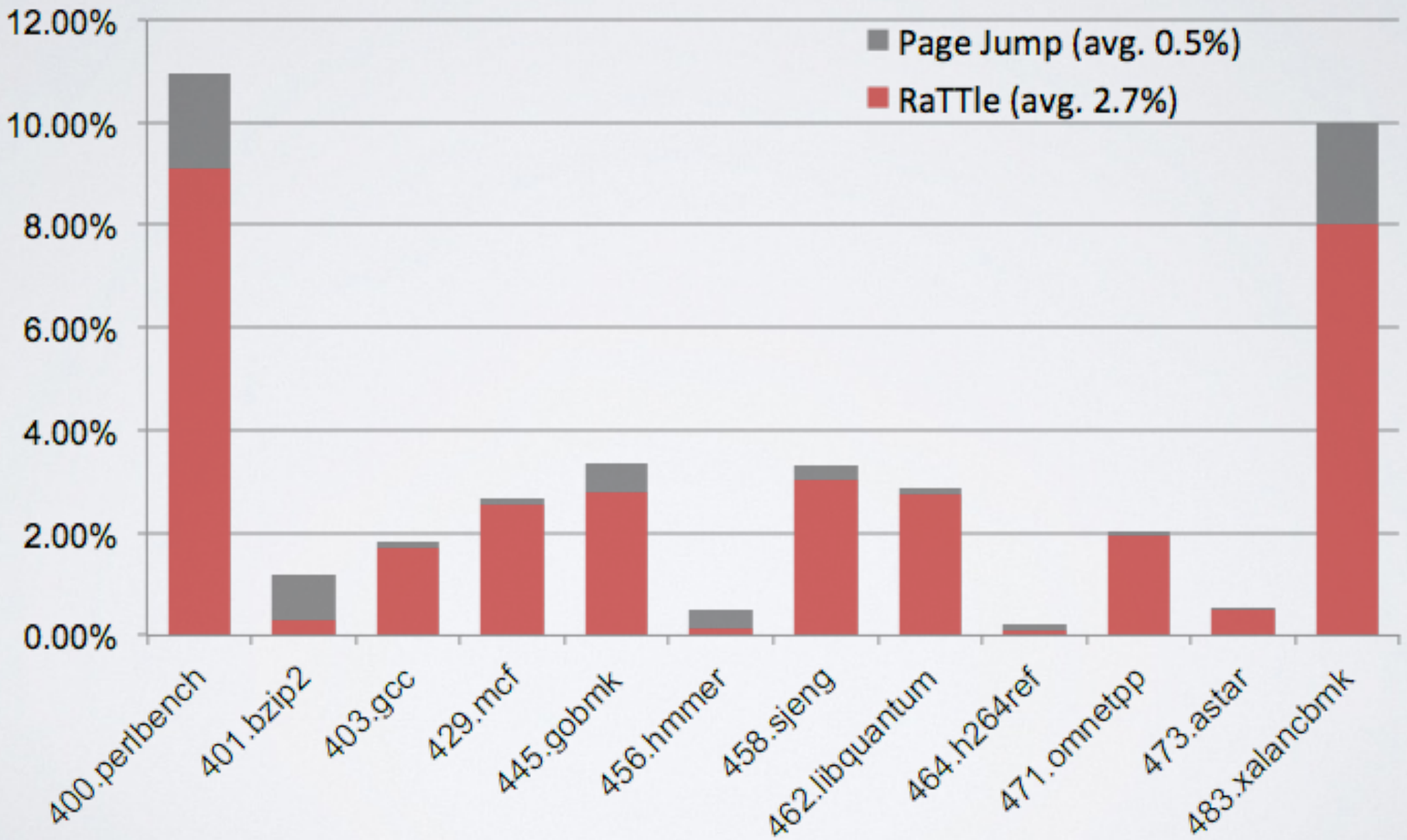
```
mov     %fs:0x8a,%edx
nop
jmp     %fs:0x7f
```

```
cmp     %ebx,%edx
jb      80480fa
mov     $0x8066db7,%eax
test    %eax,%eax
je      804812b
sub     $0xc,%esp
push    $0x8073fa8
call    8066db7
```

```
cmp     %ebx,%edx
jb      80480fa
mov     %fs:0x991,%eax
test    %eax,%eax
je      804812b
sub     $0xc,%esp
push    %fs:0x44
call    *%fs:0x21
```

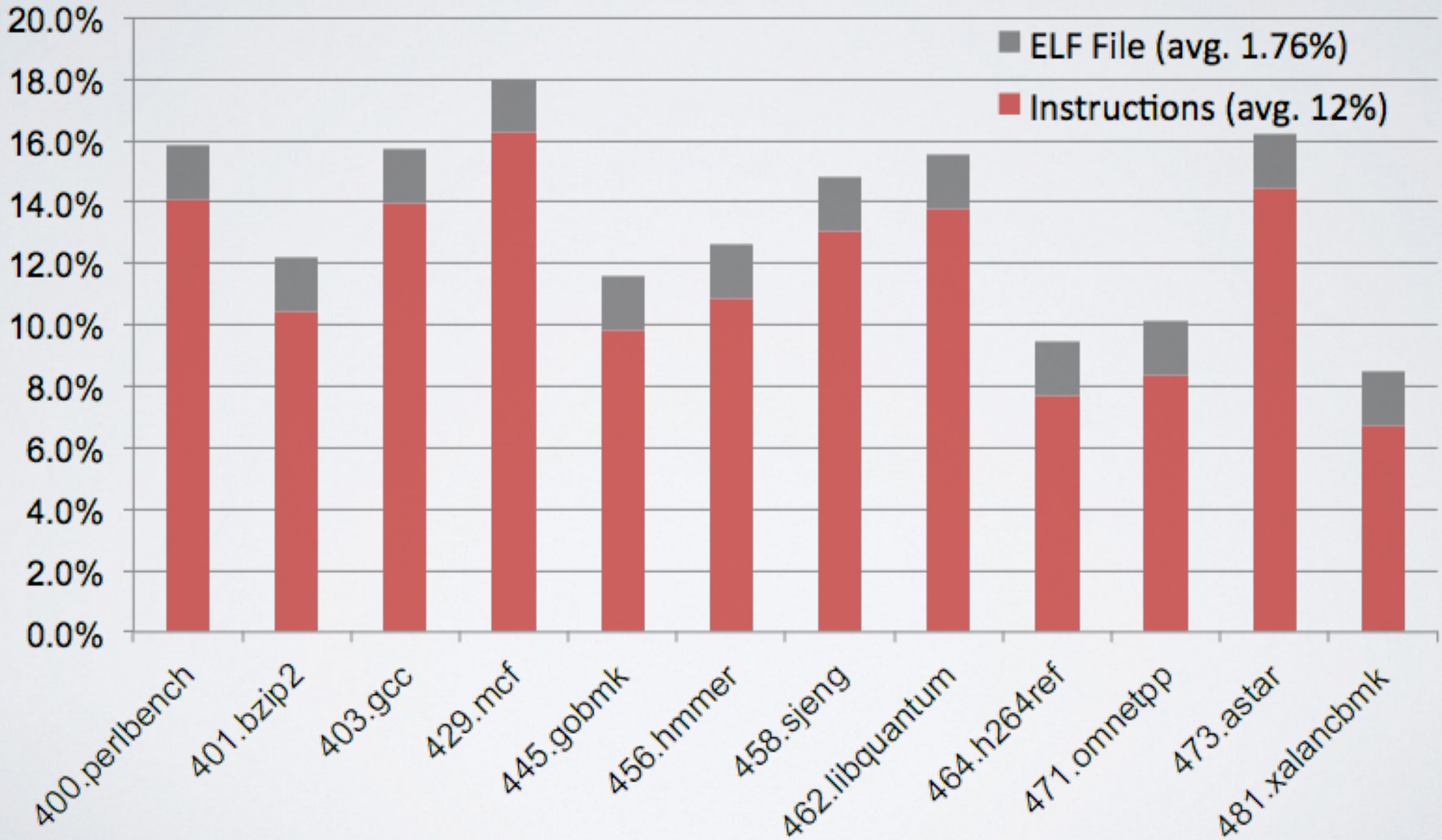
PERFORMANCE RESULTS

Runtime Performance Overhead



PERFORMANCE RESULTS

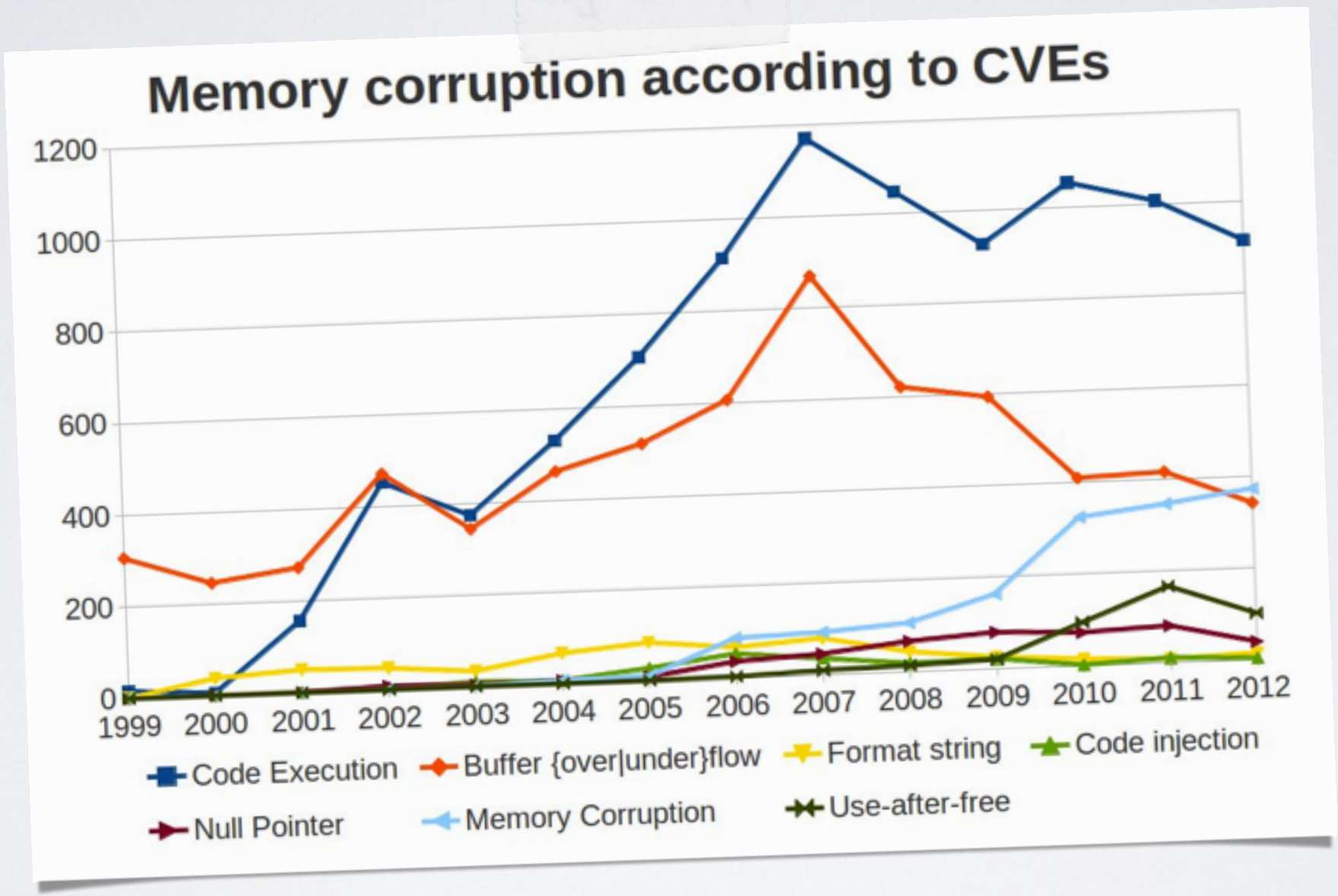
File Size Overhead



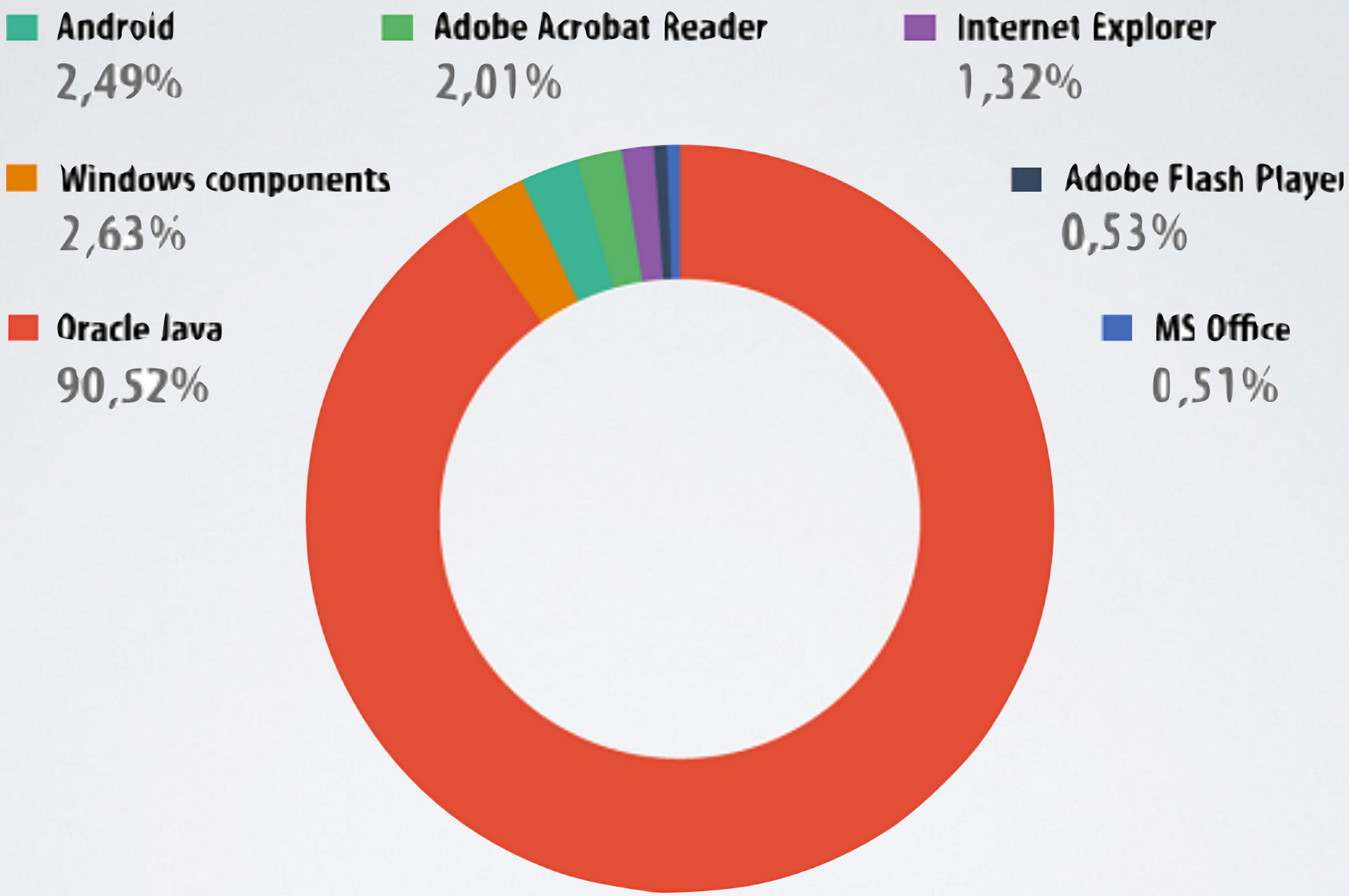
OXYMORON?



MEMORY CORRUPTION EXPLOITS (DEVELOPMENT)



ATTACK TARGETS



Distribution of exploits by type of attacked application

Source: Kaspersky Security Bulletin 58265