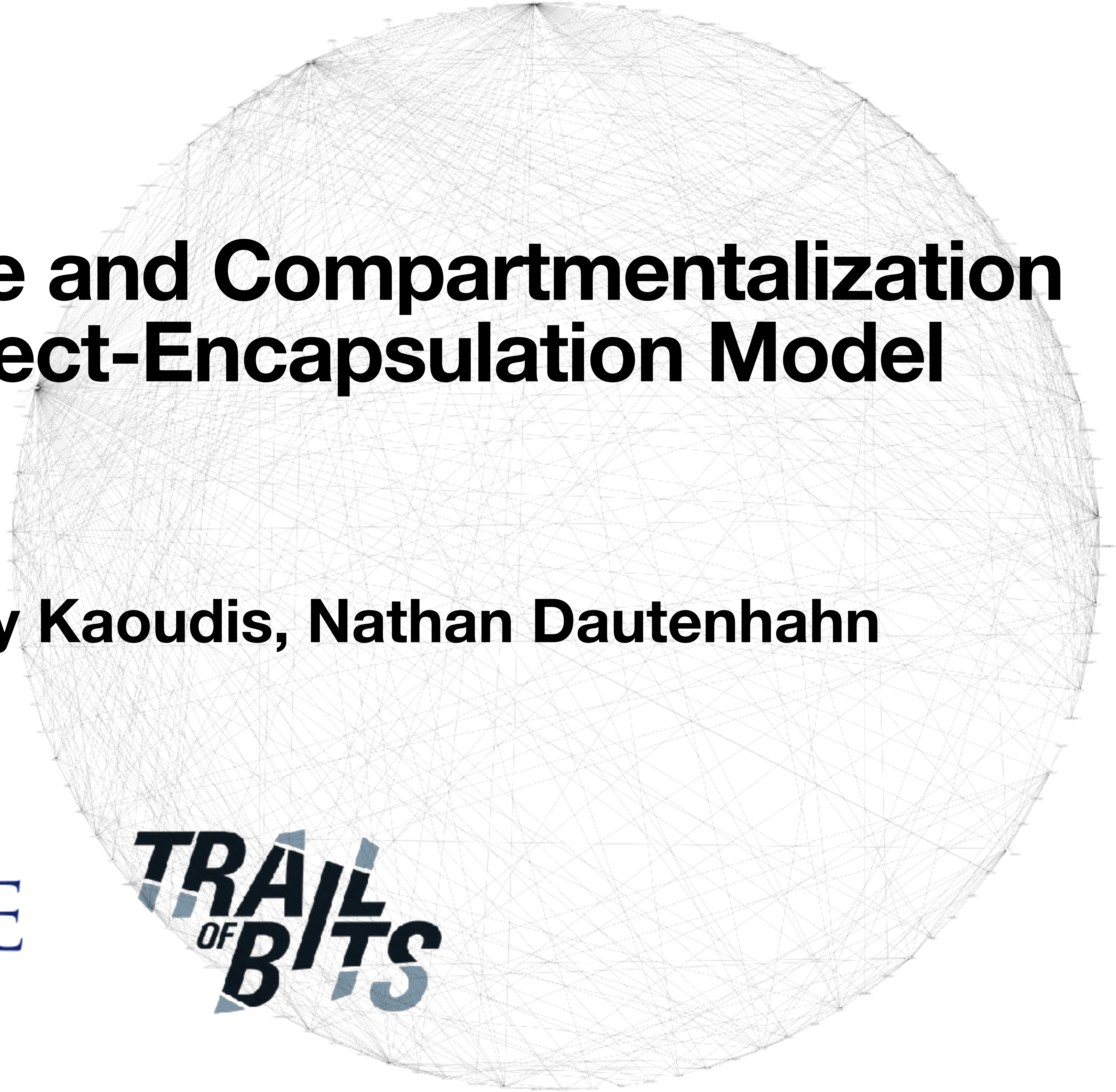
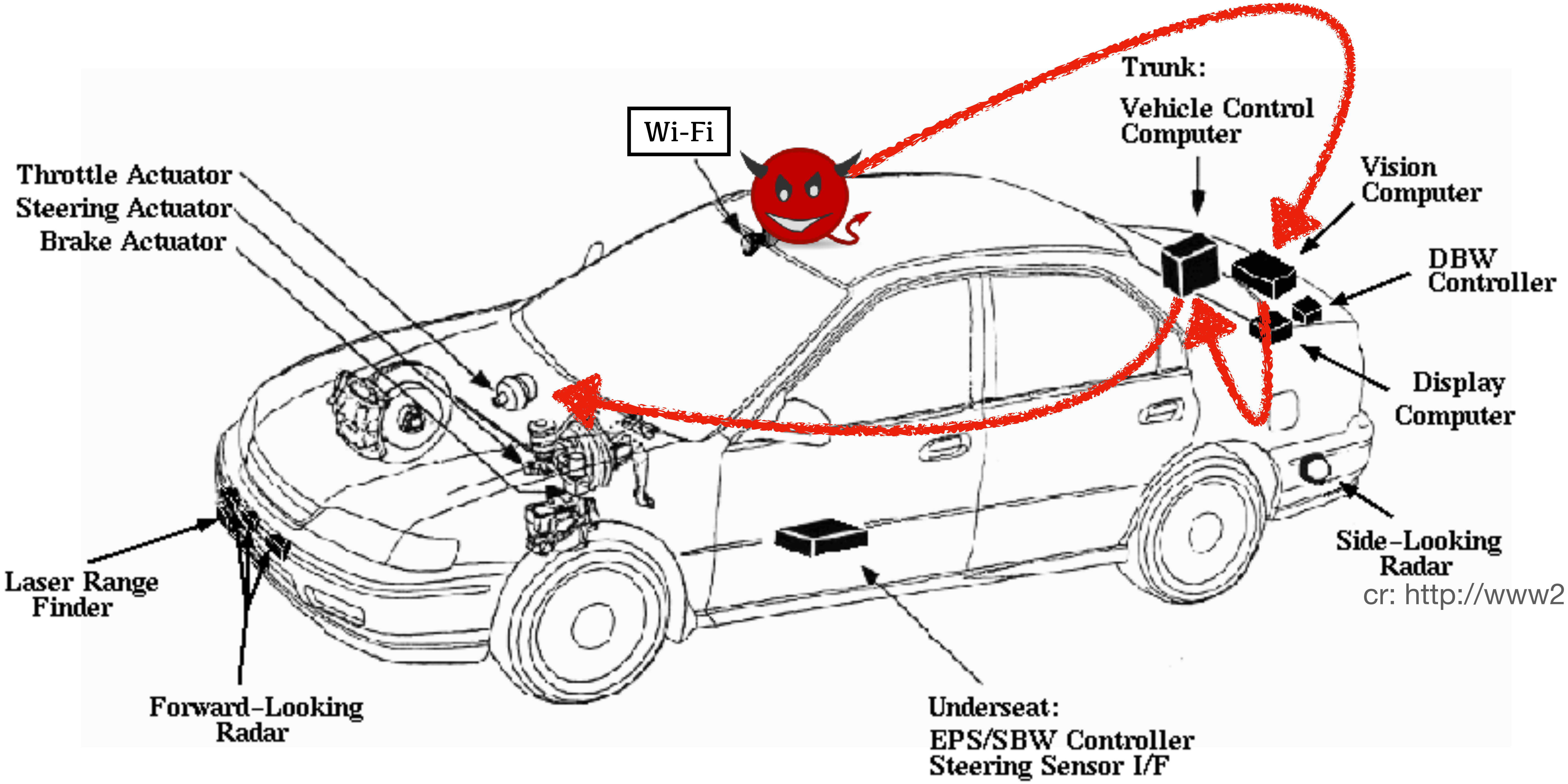
Whole-Program Privilege and Compartmentalization Analysis with the Object-Encapsulation Model

Yudi Yang, Weijie Huang, Kelly Kaoudis, Nathan Dautenhahn

LangSec 2023



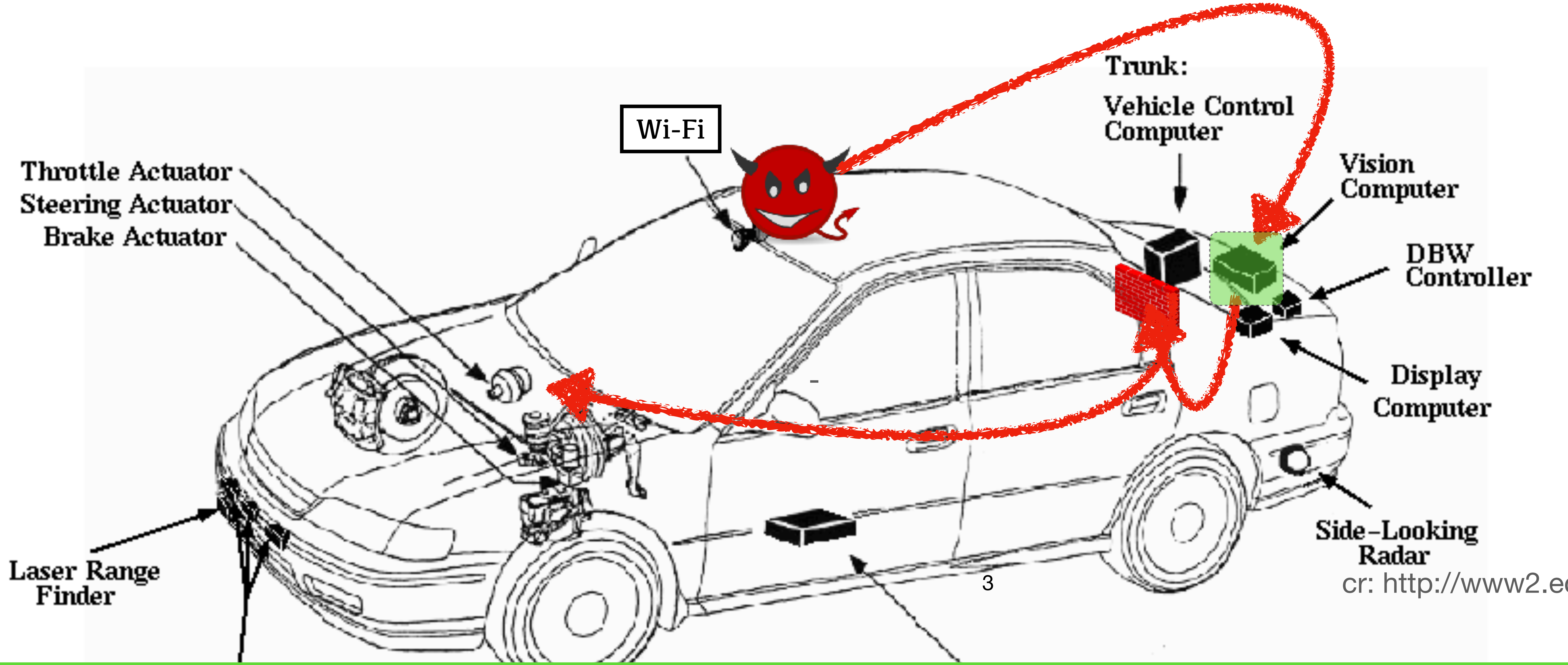
Computing systems contain over-privileged components



cr: http://www2.ece.ohio-state.edu/citr/Demo97/

cr: http://www2.ece.ohio-state.edu/citr/Demo97/osu-av.html

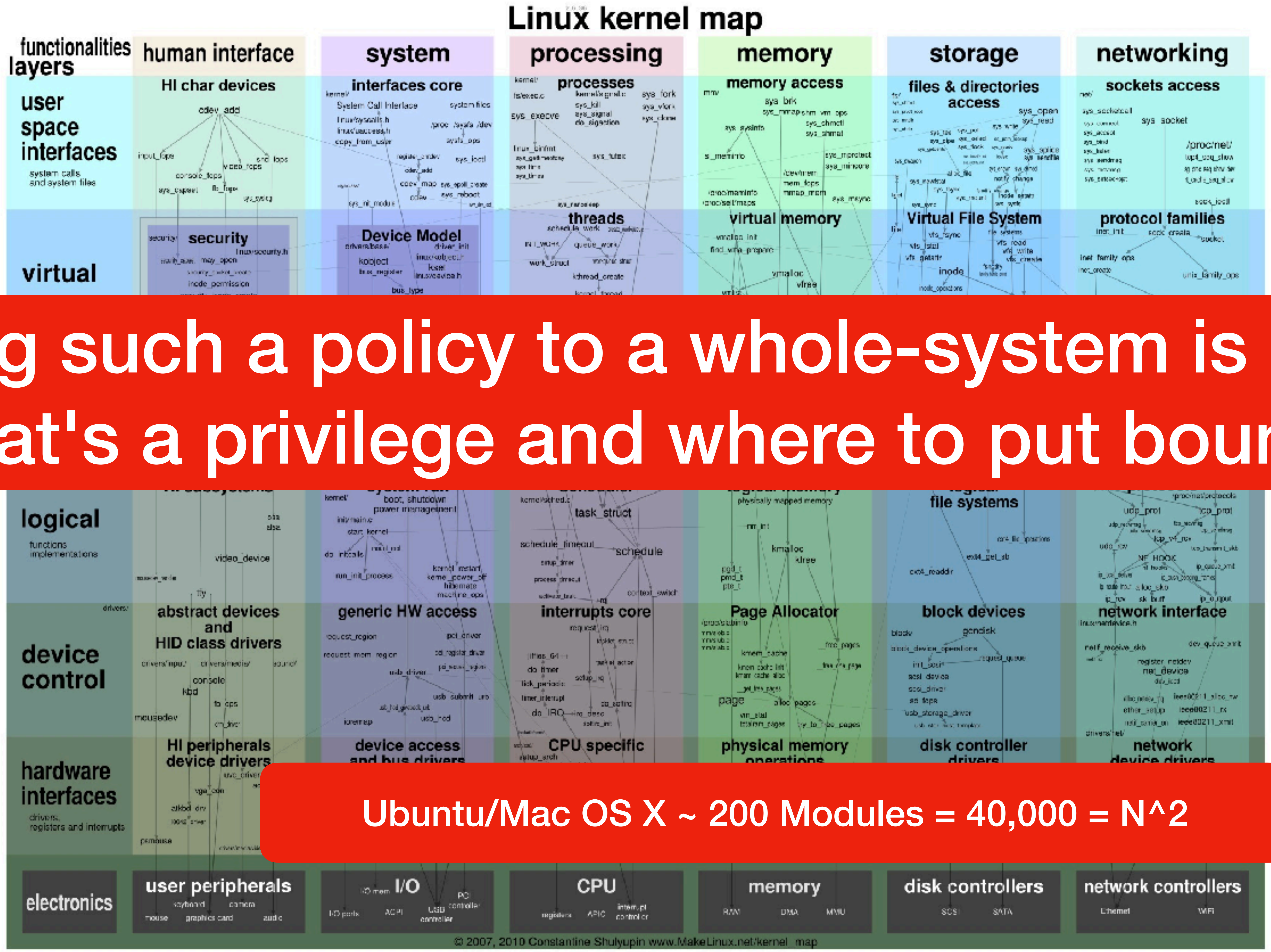
Ideally, restrict access to what it needs to do its job through compartmentalization



cr: http://www2.ece.ohio-state.edu/citr/Demo97

Stop and contain attacks through boundaries

But how do we retrofit into a system that was not built for compartmentalization?



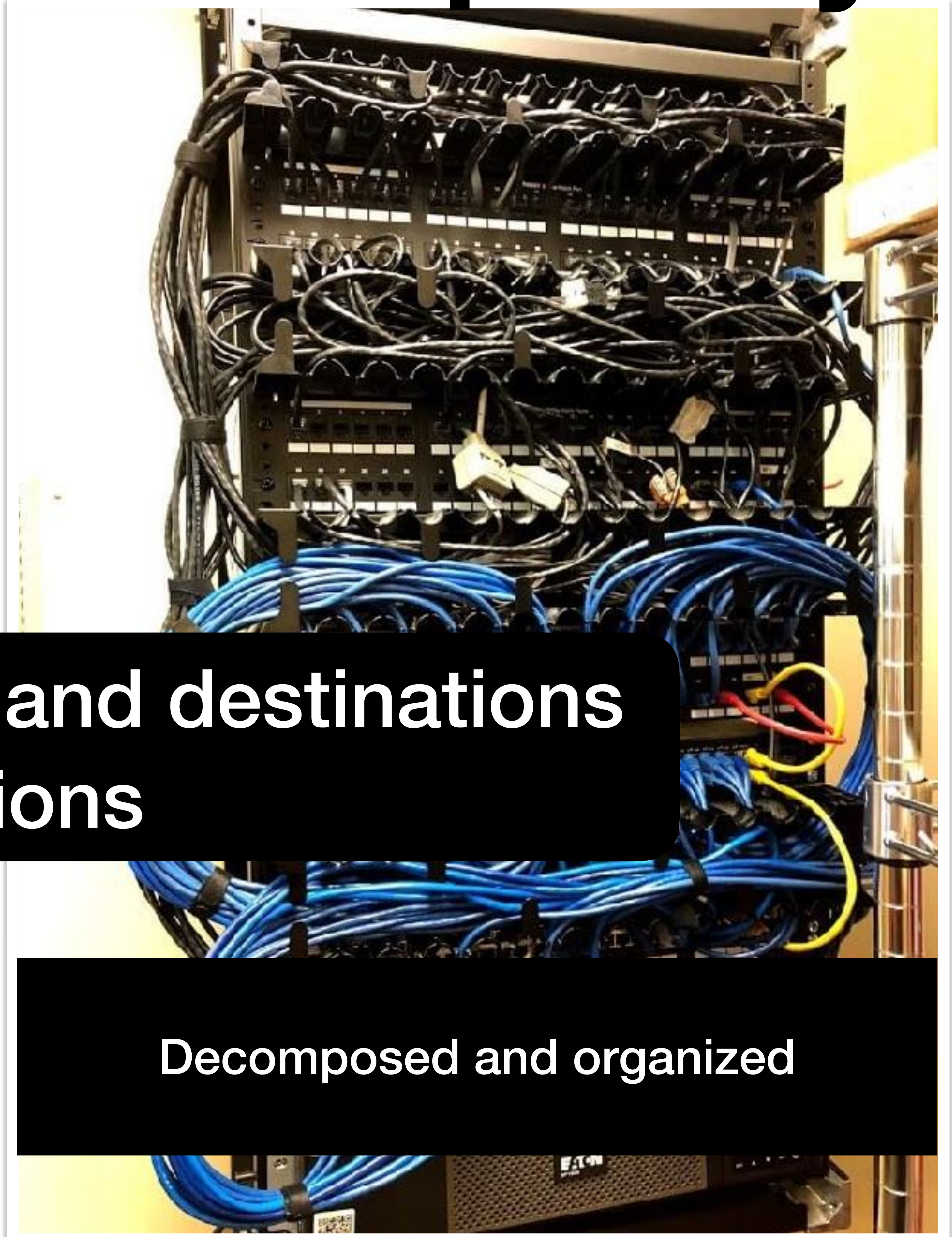
Mapping such a policy to a whole-system is not scalable: what's a privilege and where to put boundaries?

Ubuntu/Mac OS X ~ 200 Modules = 40,000 = N^2

Objective: automatically derive and systematically evaluate policy

Unknown and complex

We don't know labels on the sources and destinations of messages and operations



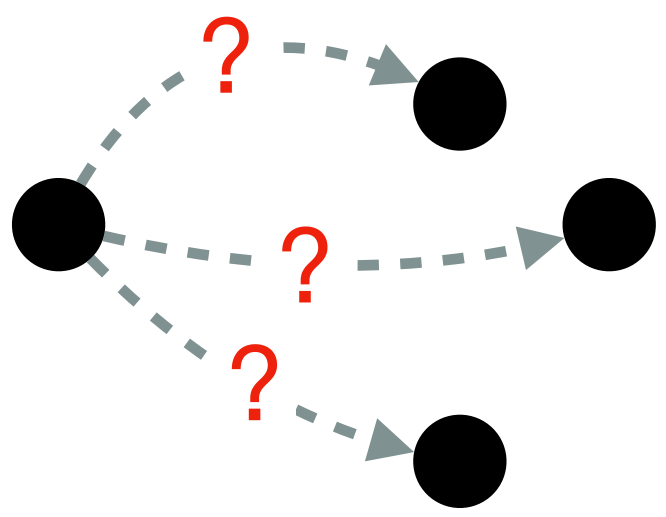
Decomposed and organized

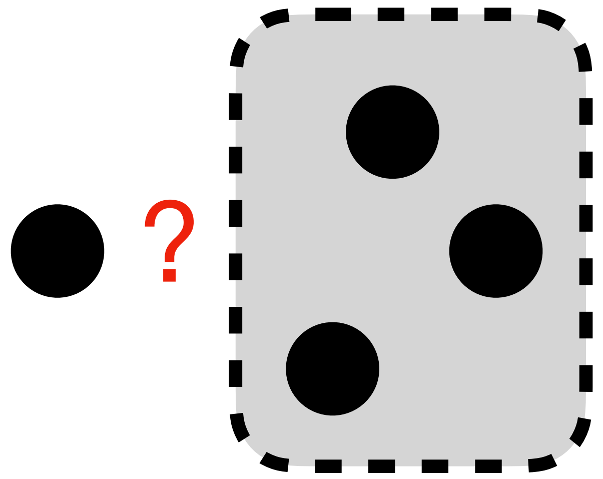
Hypothesis: emergent properties of modularity provide baseline policy

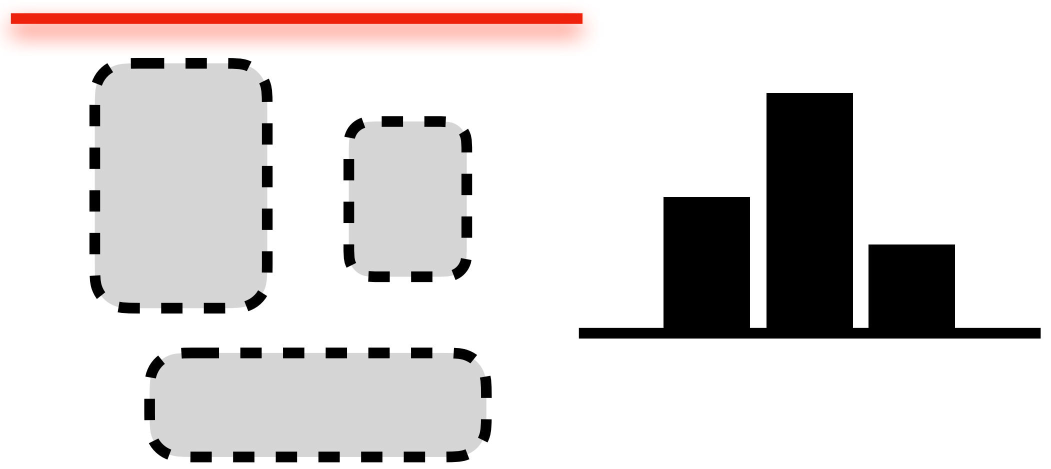
Key questions and approach overview

- What is a privilege?
- How to select boundaries?
- How to systematically evaluate?

- Program Capability Graph
- Object-Encapsulation Model
- Separability Analysis





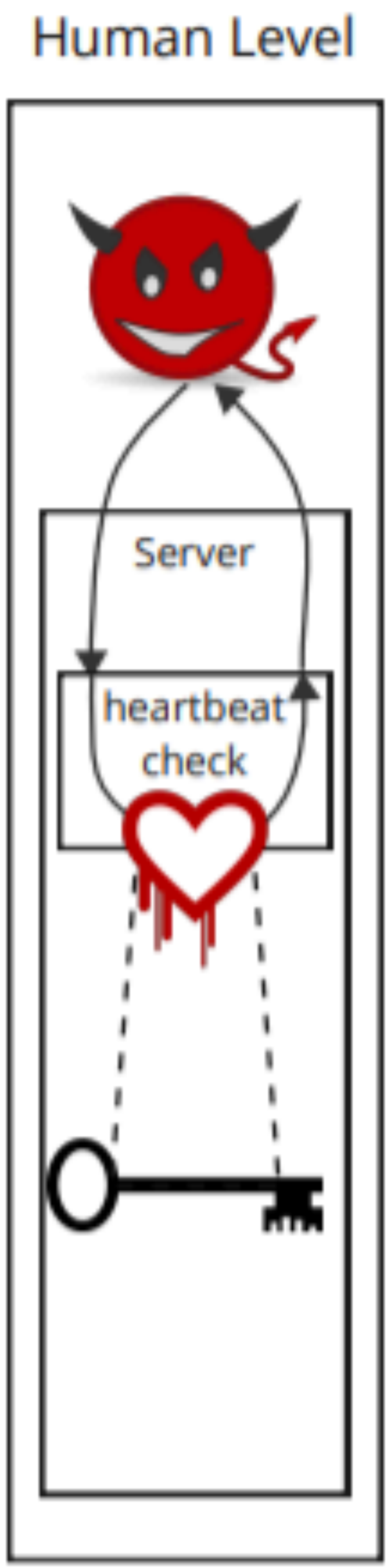


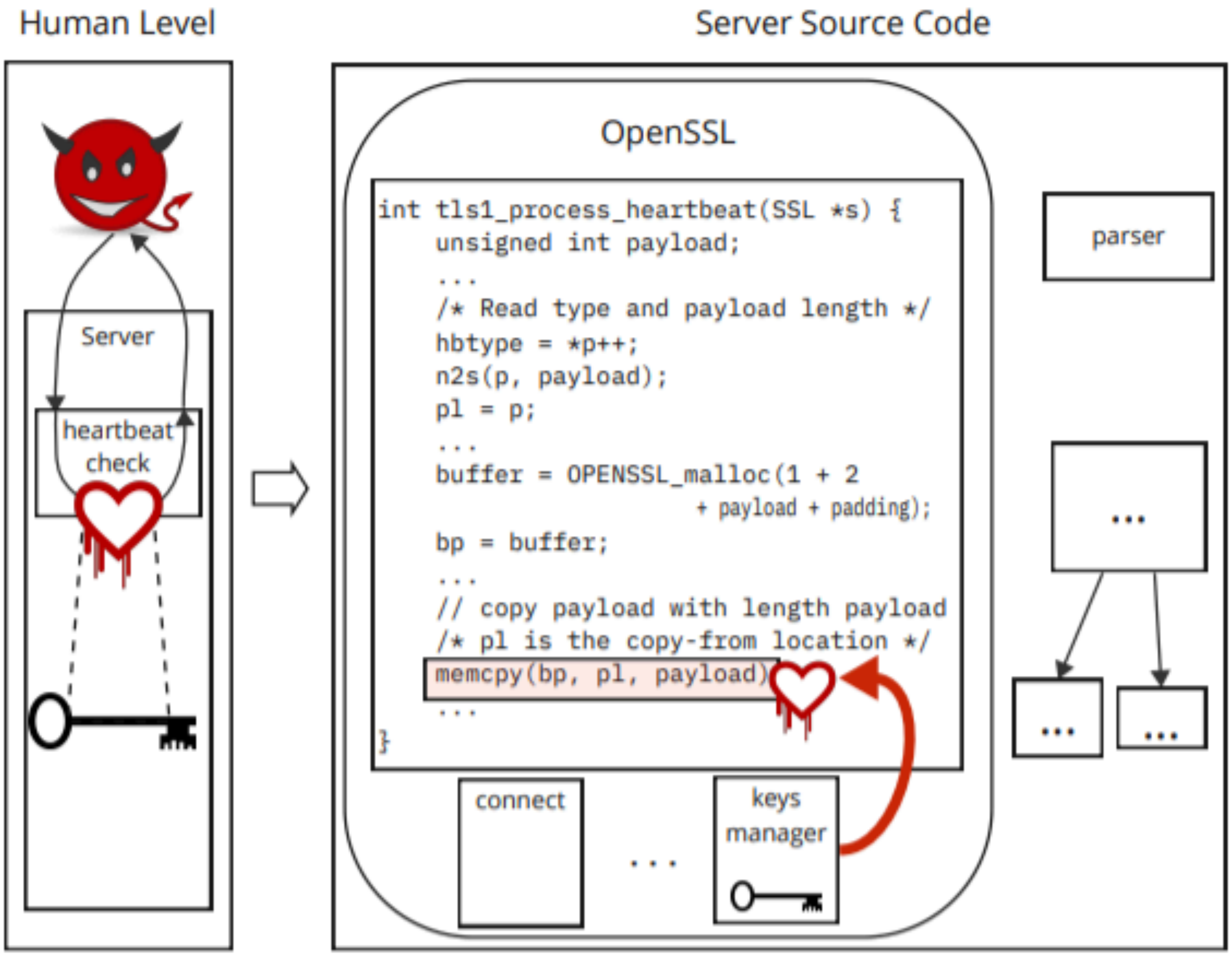
What is a privilege?

Insight #1: code as privilege

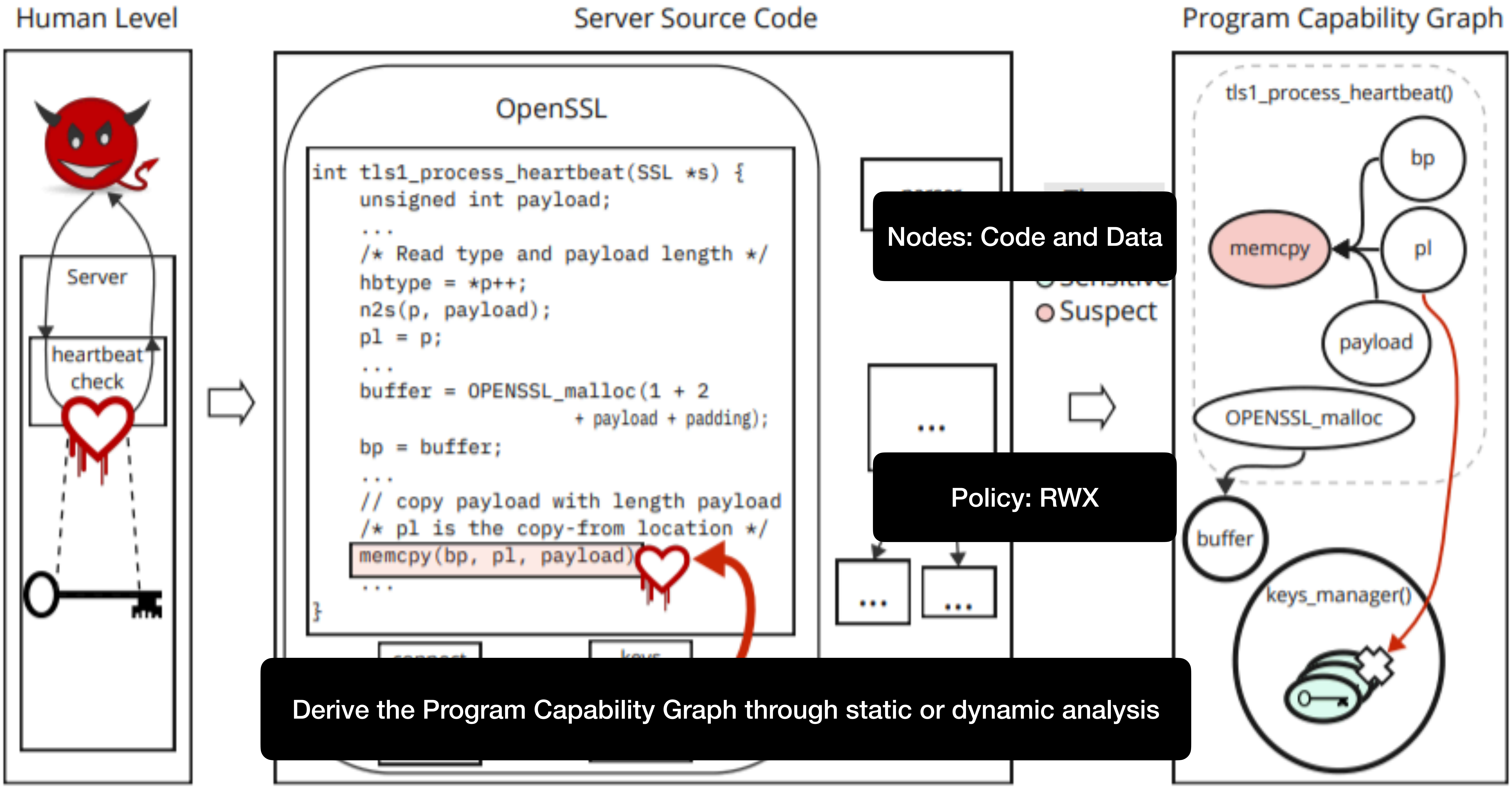
- Interaction between functions and modules: subjects
- Message sending through read / write / execute: operations
- Communication is a privilege: capabilities
- Derive through static or dynamic analysis

foo()
execute
read
write
ap()


Human Level
Server
heartbeat check



```
int tls1_process_heartbeat(SSL *s) {
    unsigned int payload;
    ...
    /* Read type and payload length */
    hbtype = *p++;
    n2s(p, payload);
    pl = p;
    ...
    buffer = OPENSSL_malloc(1 + 2
                        + payload + padding);
    bp = buffer;
    ...
    // copy payload with length payload
    /* pl is the copy-from location */
    memcpy(bp, pl, payload);
    ...
}
```



```
int tls1_process_heartbeat(SSL *s) {
    unsigned int payload;
    ...
    /* Read type and payload length */
    hbtype = *p++;
    n2s(p, payload);
    pl = p;
    ...
    buffer = OPENSSL_malloc(1 + 2
                          + payload + padding);
    bp = buffer;
    ...
    // copy payload with length payload
    /* pl is the copy-from location */
    memcpy(bp, pl, payload)
    ...
}
```

Nodes: Code and Data

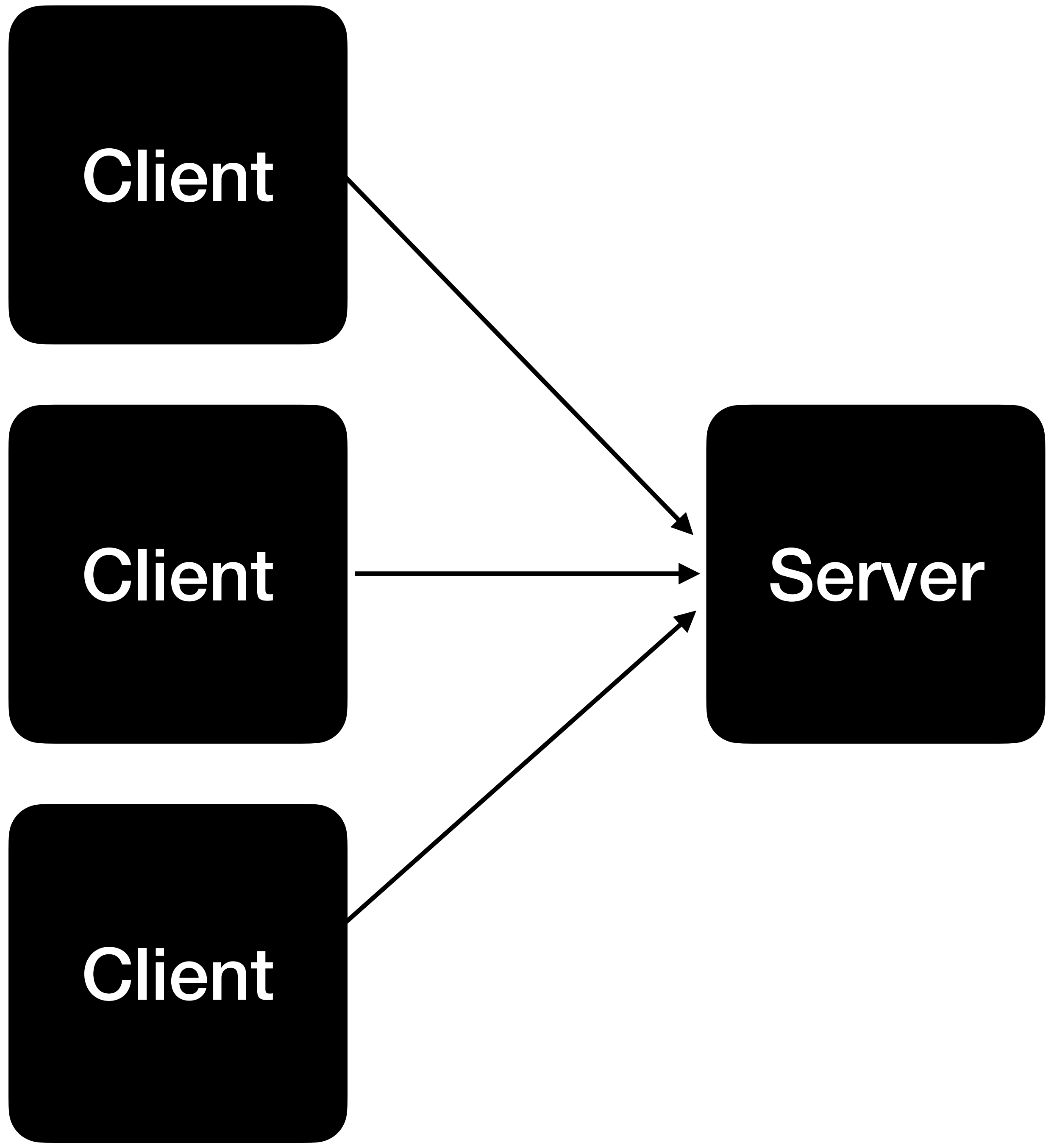
Policy: RWX

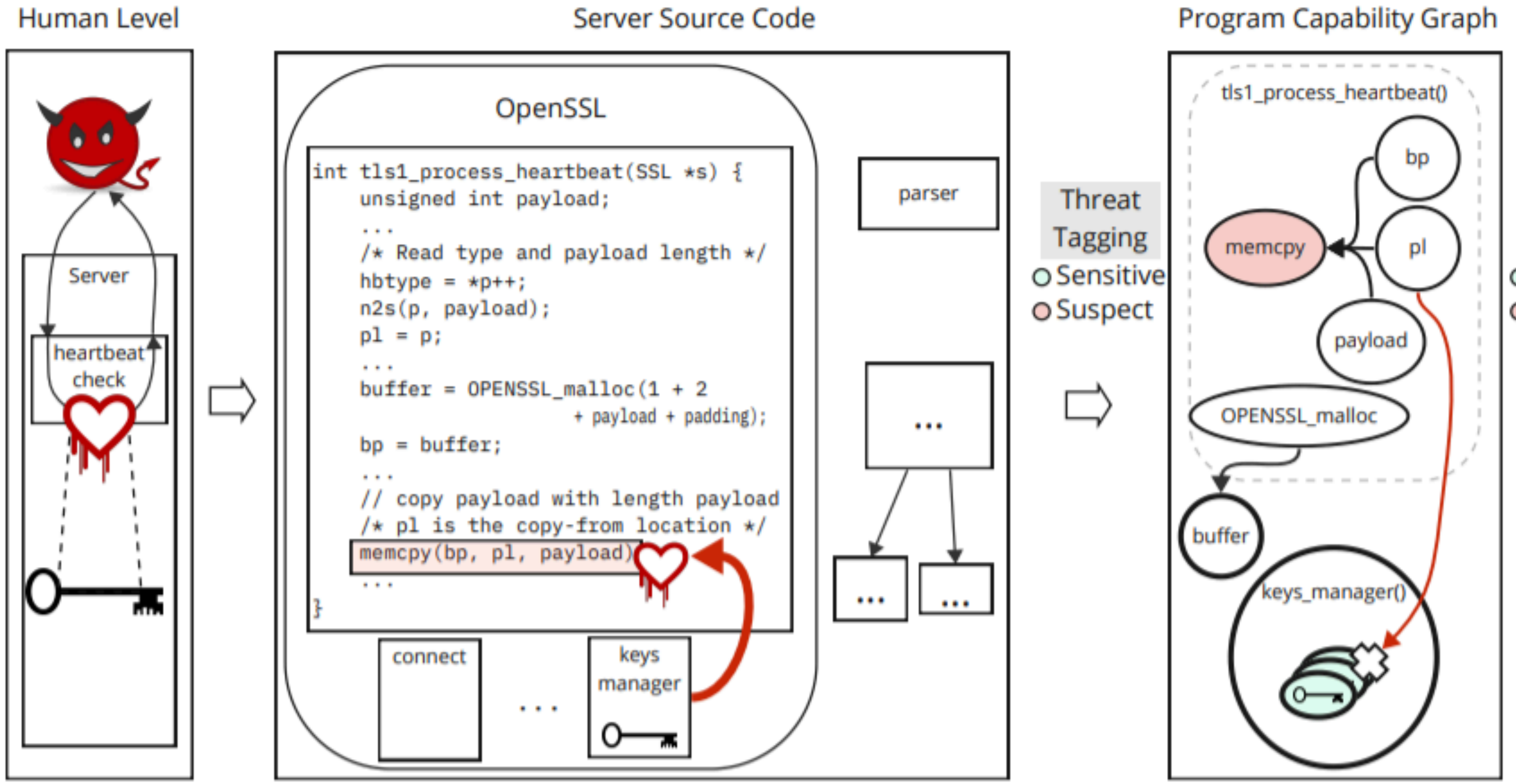
Derive the Program Capability Graph through static or dynamic analysis

What is a meaningful compartmentalization?

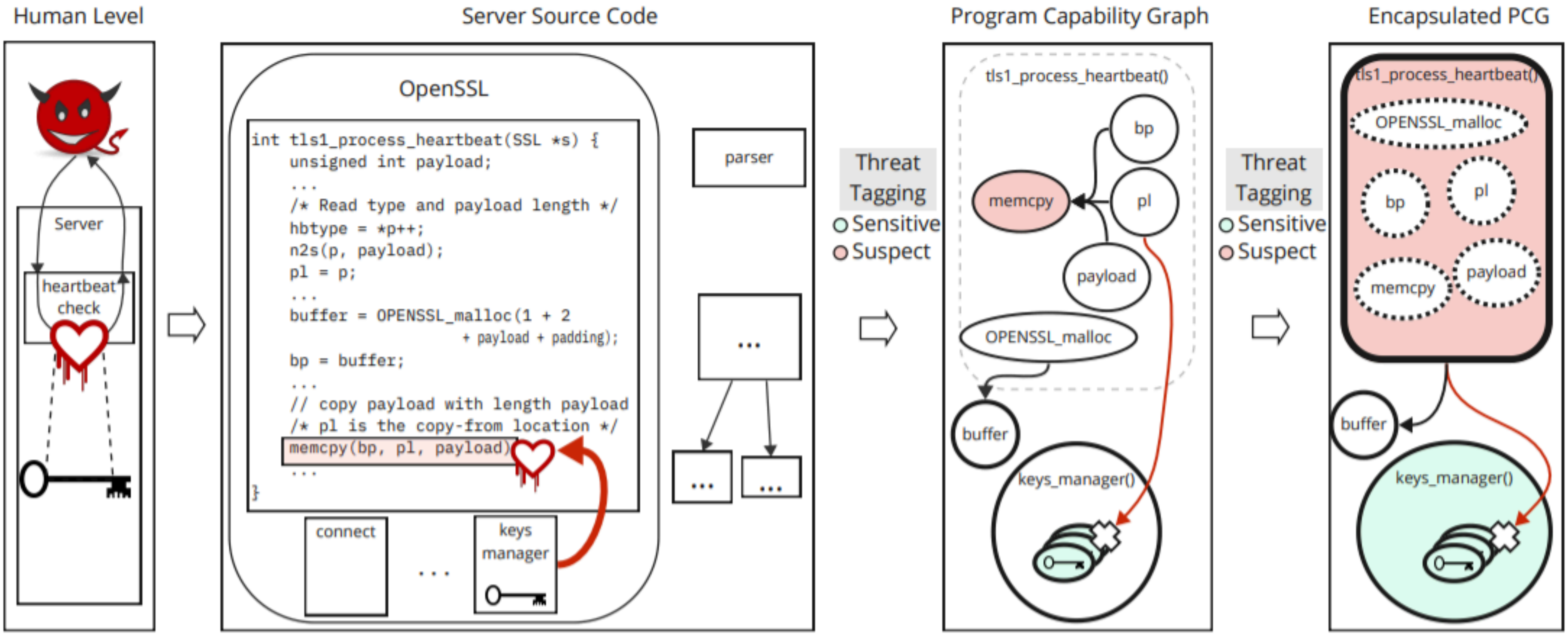
Insight #2: Emergent modularity as compartments

Developers use encapsulation to manage complexity





```
int tls1_process_heartbeat(SSL *s) {
    unsigned int payload;
    ...
    /* Read type and payload length */
    hbtype = *p++;
    n2s(p, payload);
    pl = p;
    ...
    buffer = OPENSSL_malloc(1 + 2
                       + payload + padding);
    bp = buffer;
    ...
    // copy payload with length payload
    /* pl is the copy-from location */
    memcpy(bp, pl, payload)
    ...
}
```

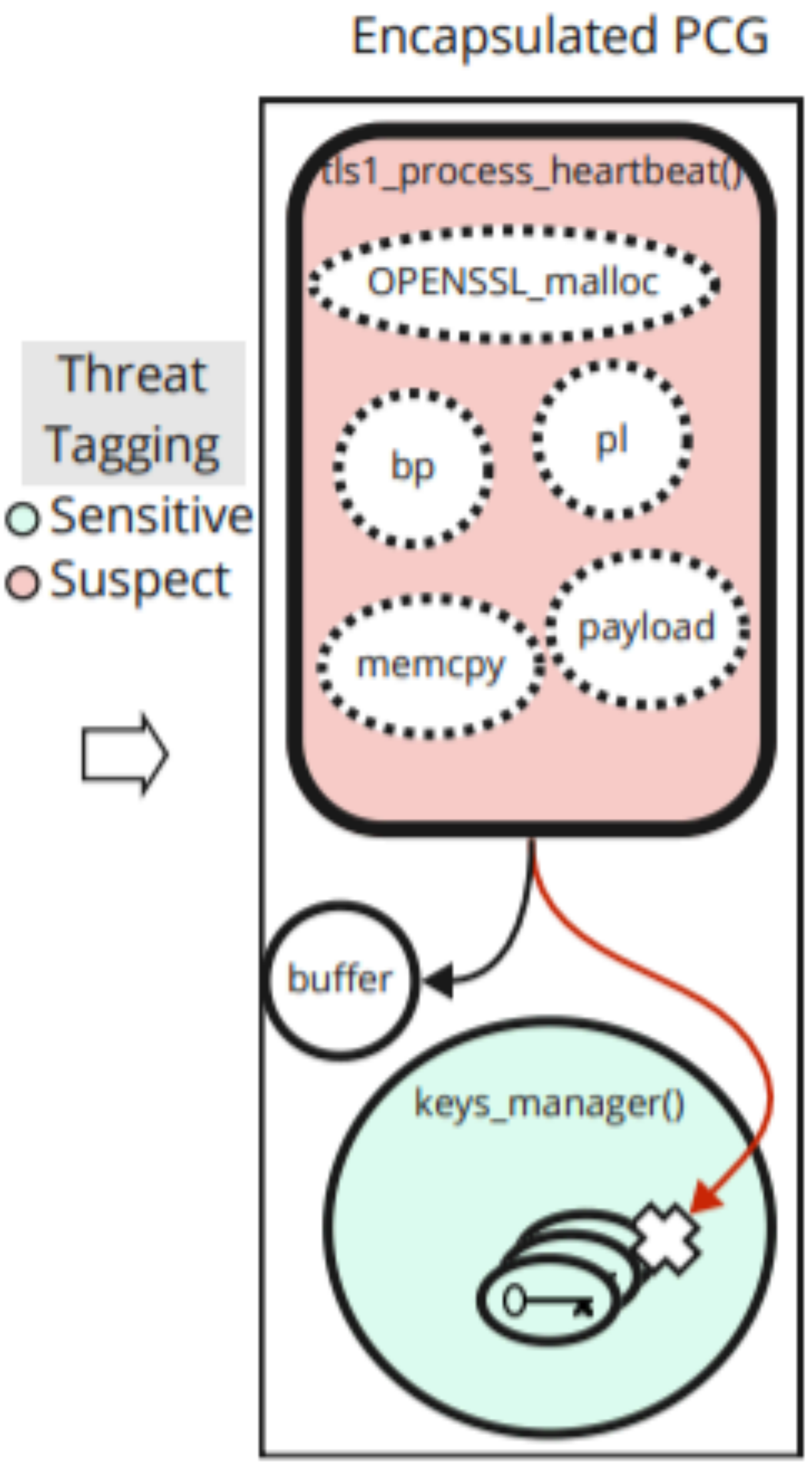


```
int tls1_process_heartbeat(SSL *s) {
    unsigned int payload;
    ...
    /* Read type and payload length */
    hbtype = *p++;
    n2s(p, payload);
    pl = p;
    ...
    buffer = OPENSSL_malloc(1 + 2
                     + payload + padding);
    bp = buffer;
    ...
    // copy payload with length payload
    /* pl is the copy-from location */
    memcpy(bp, pl, payload)
    ...
}
```

Group operations into encapsulated components with private and public data and interfaces

What is a compartment?

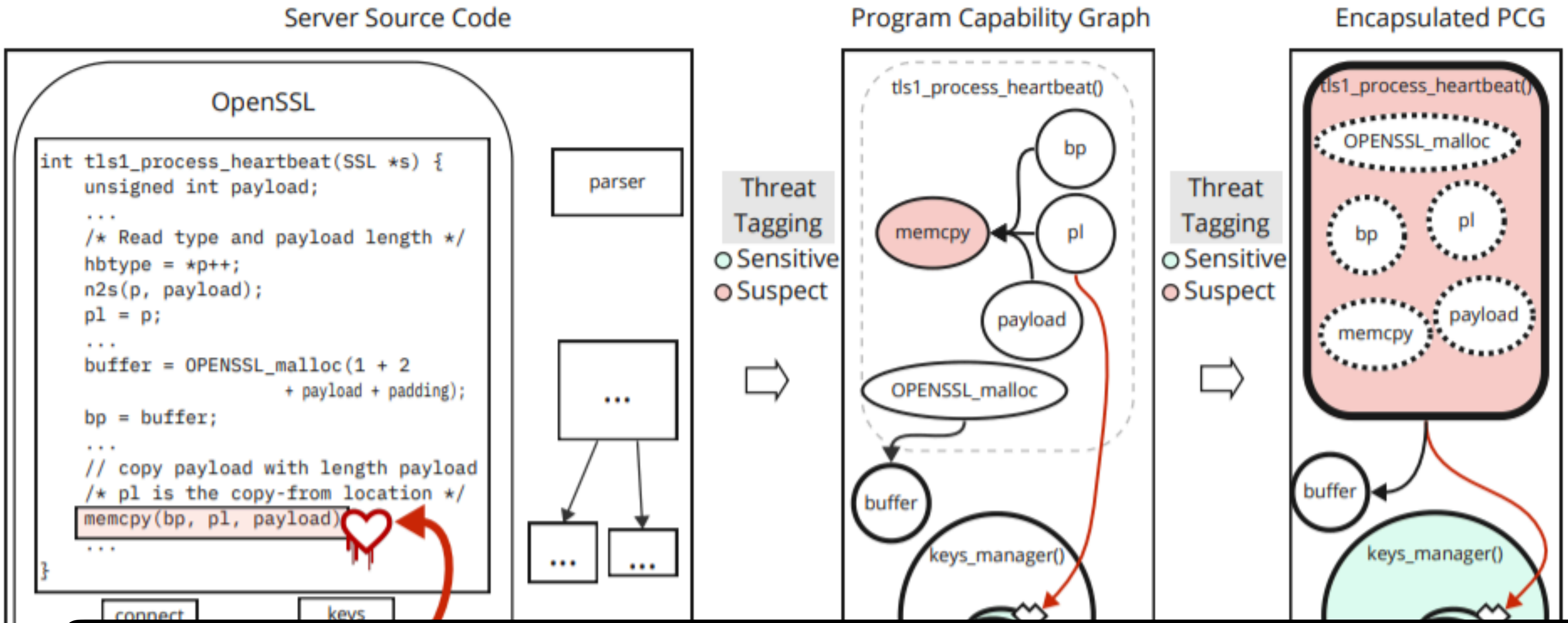
Answer: Object-Encapsulation Model



Partition based on belongs-to: tag each object and code as belonging to a specific compartment

Where do we start?

Solution: Lexical Scopes as Objects



```
int tls1_process_heartbeat(SSL *s) {
    unsigned int payload;
    ...
    /* Read type and payload length */
    hbtype = *p++;
    n2s(p, payload);
    pl = p;
    ...
    buffer = OPENSSL_malloc(1 + 2
                        + payload + padding);
    bp = buffer;
    ...
    // copy payload with length payload
    /* pl is the copy-from location */
    memcpy(bp, pl, payload)
    ...
}
```

Any label will do, but can use file scope as base compartment labels

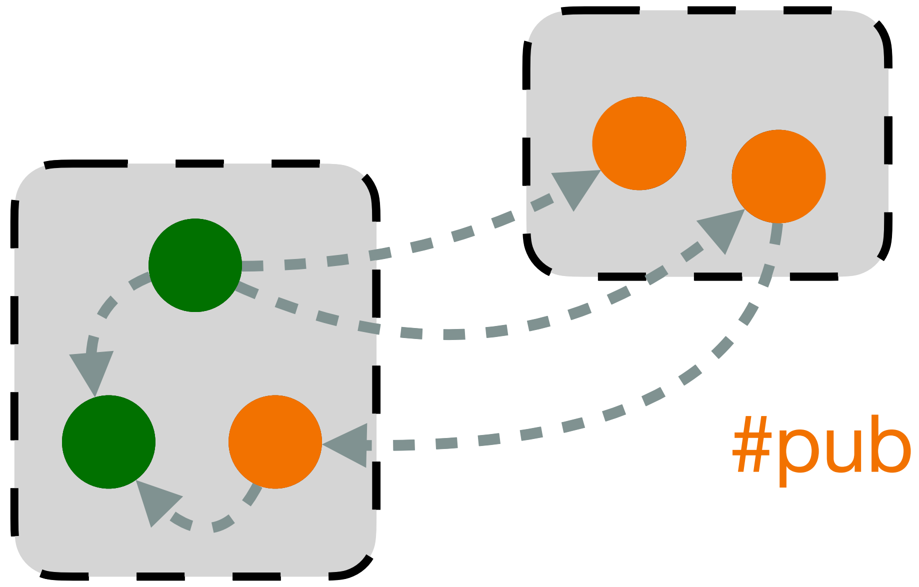
How to evaluate it?

Solution: Separability Analysis

- Privilege reduced: encapsulation analysis
- Security improved: threat modeling and measurement
- Compatibility reduced: 100%
- Transformation effort: Ongoing work
- Enforcement cost: Ongoing work

Encapsulation Analysis

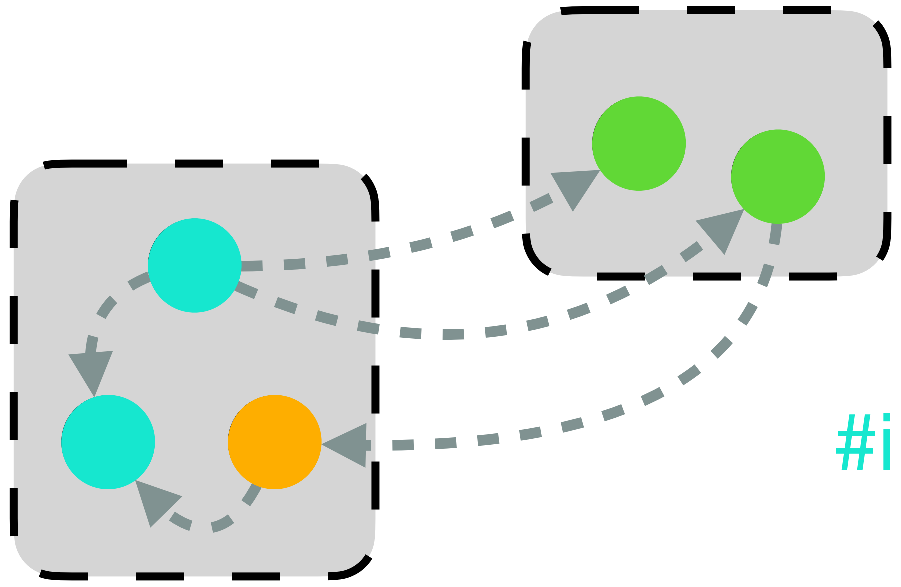
- The External Access Ratio metric



#public / #private

/

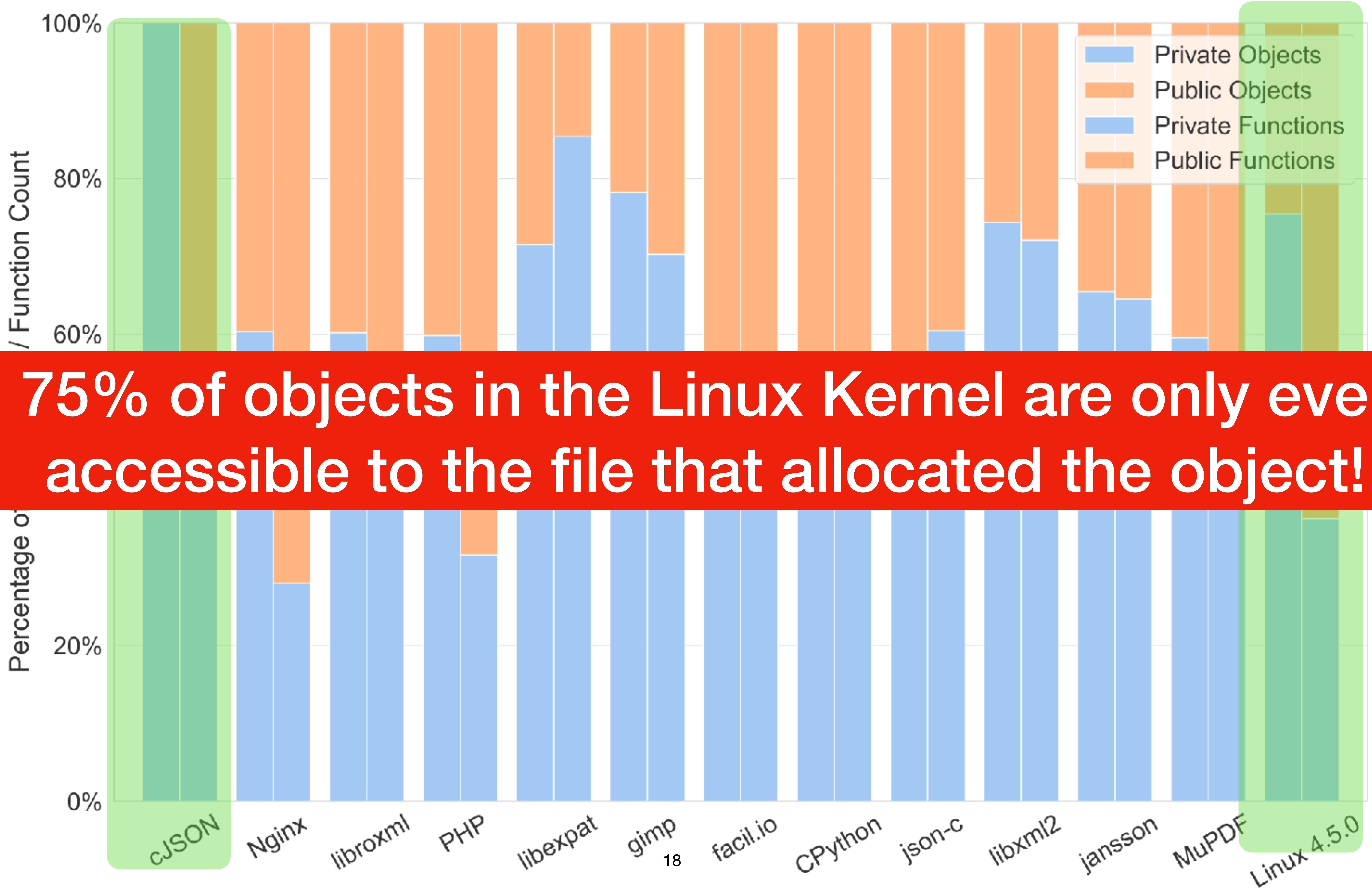
- The PS-From Ratio metric



#inaccessible / #external / #self

/ /

External Access Ratio Metric



75% of objects in the Linux Kernel are only ever accessible to the file that allocated the object!

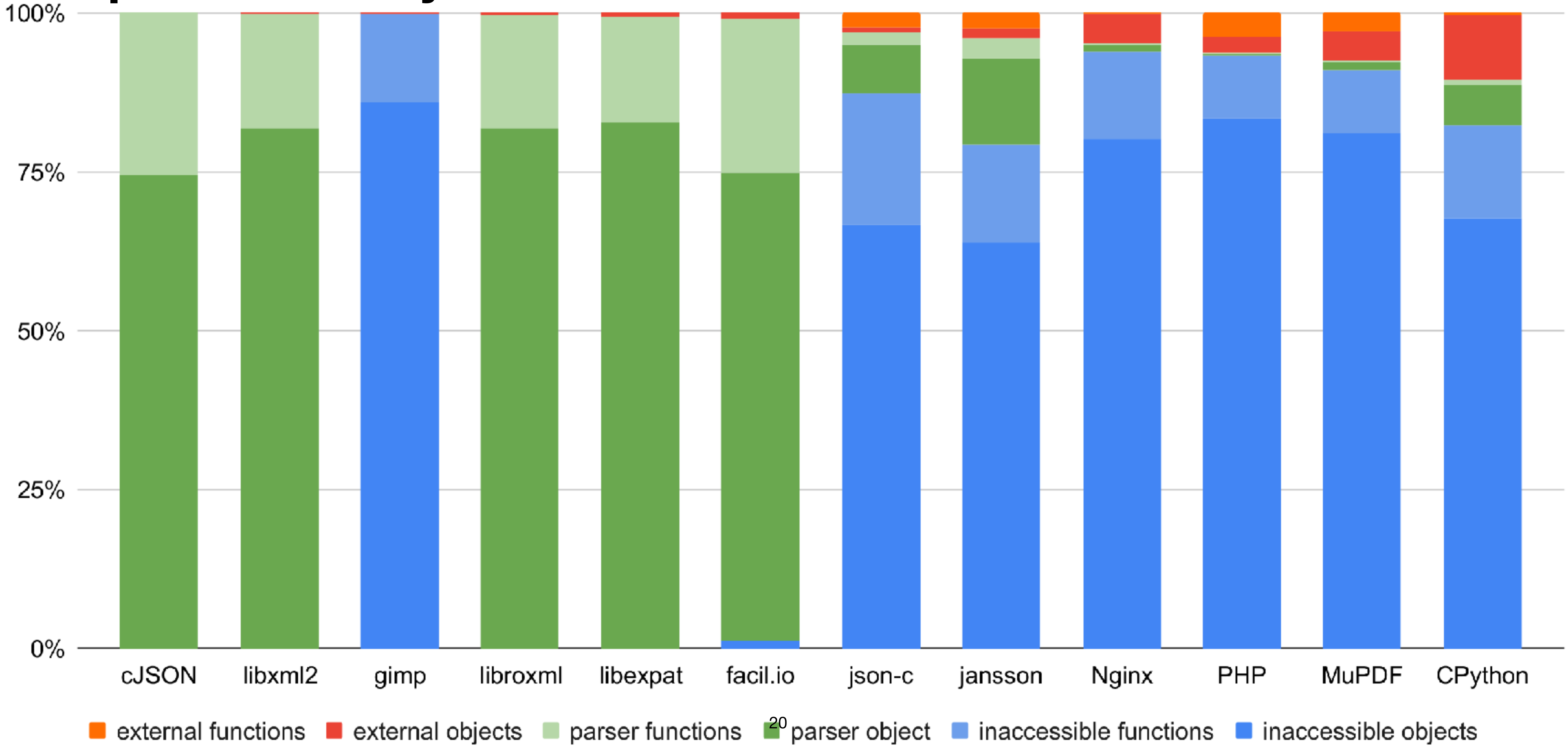
File-based Compartmentalization Accessibility

System	# of Files	CLOC (in C)	Encobjs.*	Objects*	Scalars*	% Reducible
libxml2	43	215,796	22	176	31,292	48.12%
jansson	14	7,529	4	15	607	70.88%
libroxml	12	7,205	3	10	1,199	74.24%
libexpat	8	25,452	2	13	1,259	76.79%
facil.io	30	22,602	6	39	851	78.28%
json-c	14	8,501	2	11	127	82.97%
Nginx	129	138,467	61	484	96,464	52.60%
CPython	262	530,504	106	6,327	394,867	59.53%
MuPDF	542	860,824	175	1,045	62,128	67.62%
PHP	479	1,162,182	144	2,167	236,744	69.82%
gimp	1,103	894,939	8	21	138	99.29%

* Accessiblity per encapsulated-object in Average

PS-From Ratio Metric

Comparative Analysis of Parsers



Comparative study of JSON libraries using PSFR

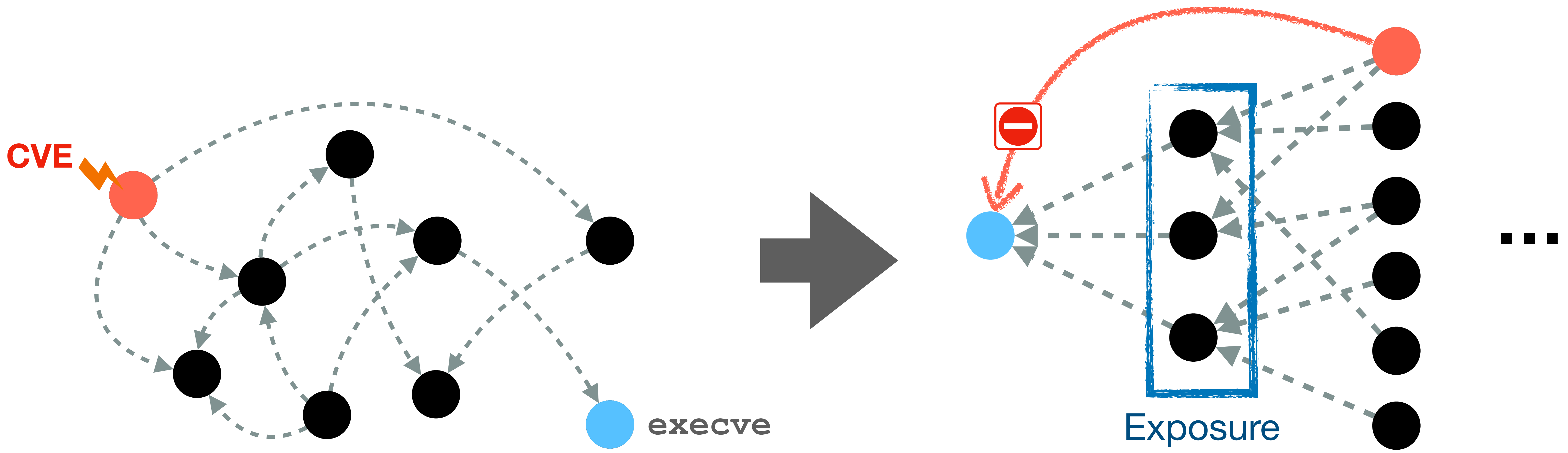
Name	Parser Data	Parser Code	External Data	External Code	Inaccessible Data	Inaccessible Code
json-c	67 (7.70%)	16 (1.84%)	8 (0.92%)	19 (2.18%)	580 (66.67%)	180 (20.69%)
facil.io	3284 (73.63%)	1078 (24.17%)	45 (1.01%)	0 (0.00%)	53 (1.19%)	0 (0.00%)
jansson	138 (13.5%)	32 (3.14%)	14 (1.37%)	26 (2.55%)	650 (63.79%)	159 (15.60%)
cJSON	325 (74.37%)	112 (25.63%)	0 (0.00%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

External Access Ratio Metric (Cont.)

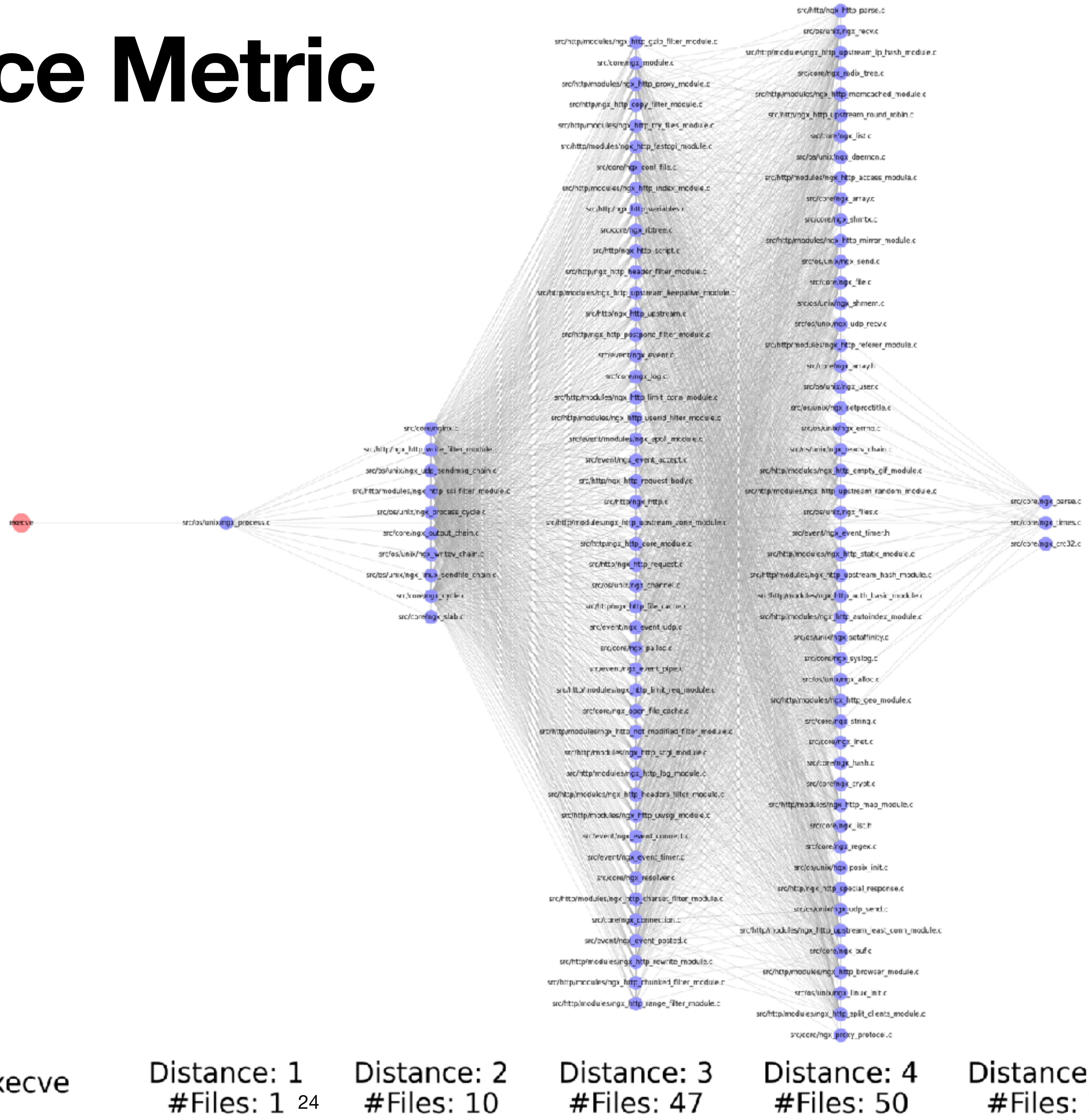
🟦 Private Objs.
🟥 Public Objs.
Percent of Total Object Count
100%
75%
50%
25%
0%
read
write
close
newstat
newfstat
newlstat
poll
lseek
mmap
mprotect
munmap
brk
rt_sigaction
rt_sigprocmask
rt_sigreturn
ioctl
pread64
access
pipe
select
mremap
dup
dup2
alarm
getpid
socket
connect
sendto
recvfrom
sendmsg
recvmsg
bind
getsockname
setsockopt
getsockopt
clone
execve
wait4
kill
newuname
fcntl
flock
fsync
fdatasync
ftruncate
getcwd
chdir
rename
mkdir
rmdir
creat
link
unlink
symlink
readlink
chmod
fchmod
chown
fchown
lchown
umask
sysinfo
getuid
getgid
setuid
setgid
geteuid
getegid
getppid
getpgrp
setsid
getgroups
setgroups
setresuid
getresuid
setresgid
getresgid
getpgid
getsid
capget
capset
sigaltstack
statfs
fstatfs
prctl
arch_prctl
gettid
getxattr
lgetxattr
futex
getdents64
set_tid_address
exit_group
utimes
openat
mkdirat
fchownat
newfstatat
unlinkat
symlinkat
readlinkat
faccessat
pselect6
ppoll
set_robust_list
sync_file_range
utimensat
dup3
pipe2
prlimit64
sendmmsg
renameat2
getrandom

Threat model labeling and analysis

- Threat Labeling: Sensitive/Suspicious
- Access Distance Metric
- Exposure Reduction Metric

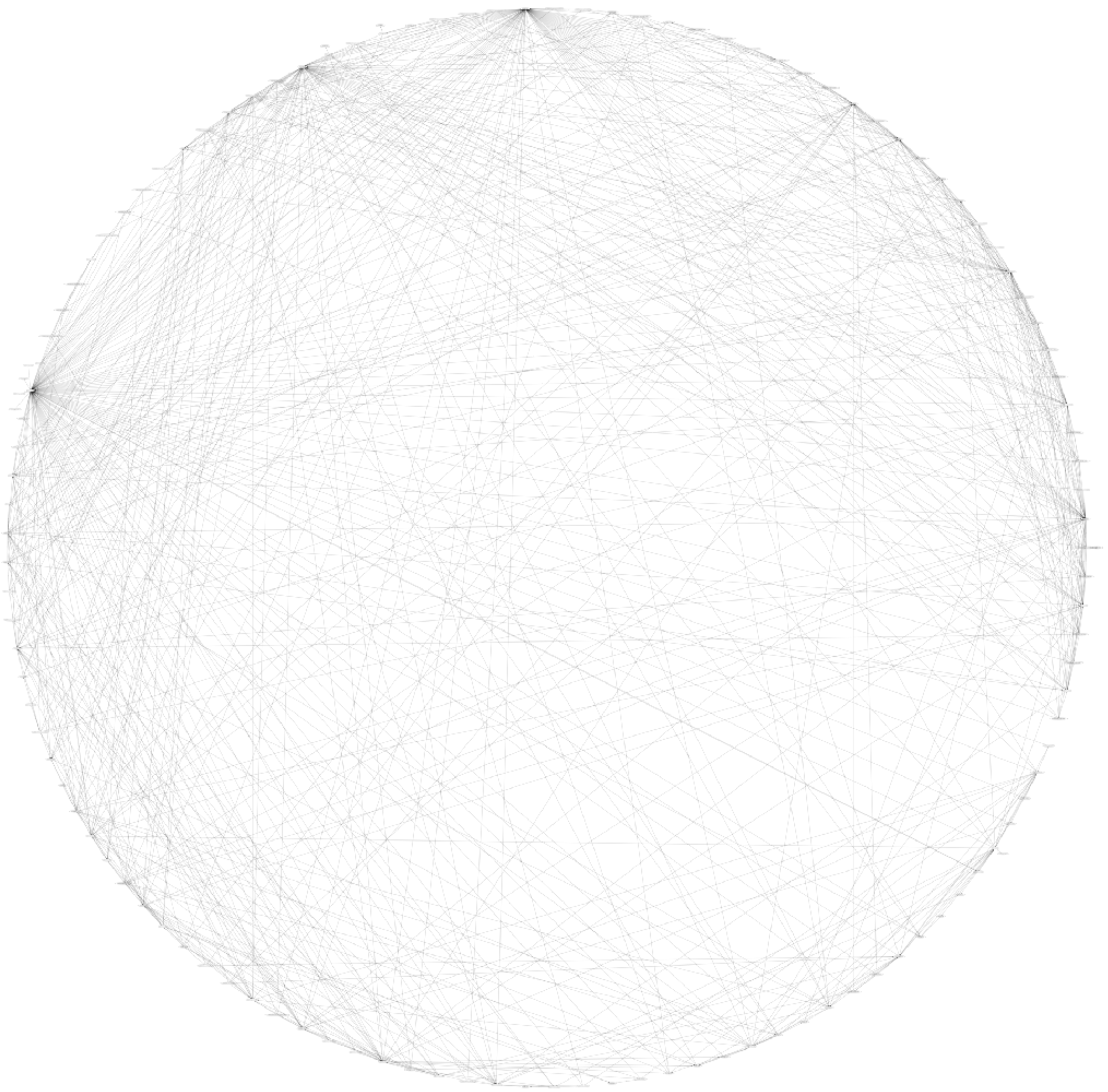


Access Distance Metric

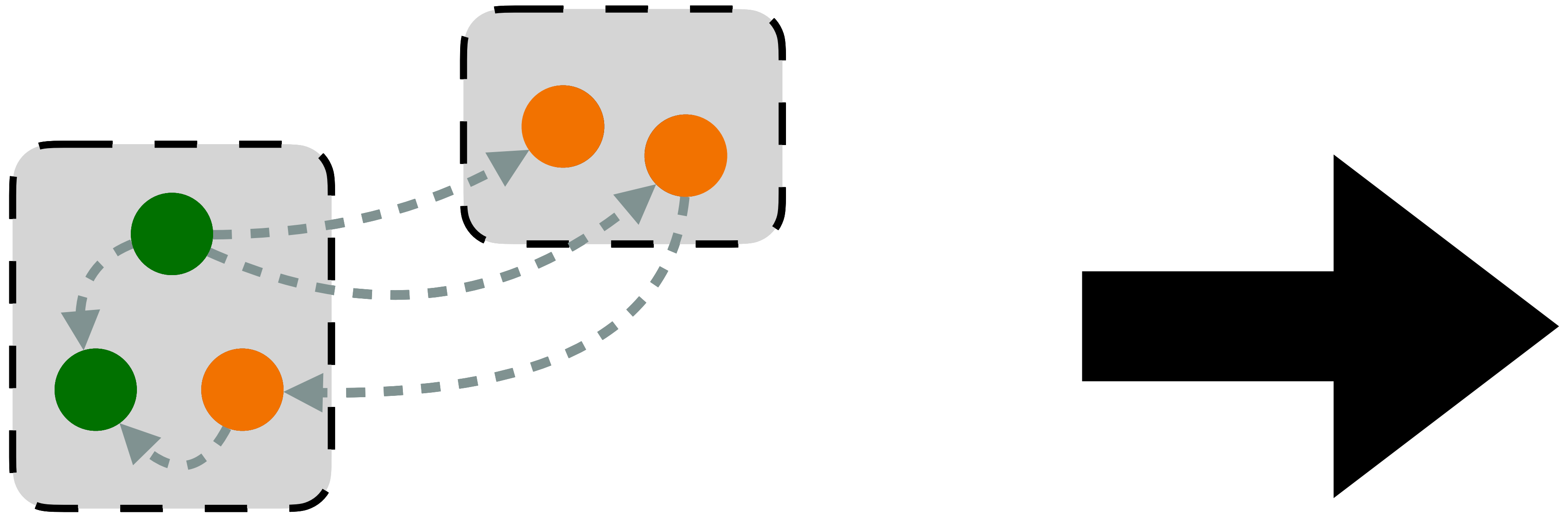

execve
src/os/unix/ngx_process.c
src/core/nginx.c
src/http/ngx_http_write_filter_module.c
src/os/unix/ngx_udp_sendmsg_chain.c
src/http/modules/ngx_http_ssi_filter_module.c
src/os/unix/ngx_process_cycle.c
src/core/ngx_output_chain.c
src/os/unix/ngx_writev_chain.c
src/os/unix/ngx_linux_sendfile_chain.c
src/core/ngx_cycle.c
src/core/ngx_slab.c
src/http/modules/ngx_http_gzip_filter_module.c
src/core/ngx_module.c
src/http/modules/ngx_http_proxy_module.c
src/http/ngx_http_copy_filter_module.c
src/http/modules/ngx_http_try_files_module.c
src/http/modules/ngx_http_fastcgi_module.c
src/core/ngx_conf_file.c
src/http/modules/ngx_http_index_module.c
src/http/ngx_http_variables.c
src/core/ngx_rbtree.c
src/http/ngx_http_script.c
src/http/ngx_http_header_filter_module.c
src/http/modules/ngx_http_upstream_keepalive_module.c
src/http/ngx_http_upstream.c
src/http/ngx_http_postpone_filter_module.c
src/event/ngx_event.c
src/core/ngx_log.c
src/http/modules/ngx_http_limit_conn_module.c
src/http/modules/ngx_http_userid_filter_module.c
src/event/modules/ngx_epoll_module.c
src/event/ngx_event_accept.c
src/http/ngx_http_request_body.c
src/http/ngx_http.c
src/http/modules/ngx_http_upstream_zone_module.c
src/http/ngx_http_core_module.c
src/http/ngx_http_request.c
src/os/unix/ngx_channel.c
src/http/ngx_http_file_cache.c
src/event/ngx_event_udp.c
src/core/ngx_palloc.c
src/event/ngx_event_pipe.c
src/http/modules/ngx_http_limit_req_module.c
src/core/ngx_open_file_cache.c
src/http/modules/ngx_http_not_modified_filter_module.c
src/http/modules/ngx_http_scgi_module.c
src/http/modules/ngx_http_log_module.c
src/http/modules/ngx_http_headers_filter_module.c
src/http/modules/ngx_http_uwsgi_module.c
src/event/ngx_event_connect.c
src/event/ngx_event_timer.c
src/core/ngx_resolver.c
src/http/modules/ngx_http_charset_filter_module.c
src/core/ngx_connection.c
src/event/ngx_event_posted.c
src/http/modules/ngx_http_rewrite_module.c
src/http/modules/ngx_http_chunked_filter_module.c
src/http/modules/ngx_http_range_filter_module.c
src/http/ngx_http_parse.c
src/os/unix/ngx_recv.c
src/http/modules/ngx_http_upstream_ip_hash_module.c
src/core/ngx_radix_tree.c
src/http/modules/ngx_http_memcached_module.c
src/http/ngx_http_upstream_round_robin.c
src/core/ngx_list.c
src/os/unix/ngx_daemon.c
src/http/modules/ngx_http_access_module.c
src/core/ngx_array.c
src/core/ngx_shmtx.c
src/http/modules/ngx_http_mirror_module.c
src/os/unix/ngx_send.c
src/core/ngx_file.c
src/os/unix/ngx_shmem.c
src/os/unix/ngx_udp_recv.c
src/http/modules/ngx_http_referer_module.c
src/core/ngx_array.h
src/os/unix/ngx_user.c
src/os/unix/ngx_setproctitle.c
src/os/unix/ngx_errno.c
src/os/unix/ngx_readv_chain.c
src/http/modules/ngx_http_empty_gif_module.c
src/http/modules/ngx_http_upstream_random_module.c
src/os/unix/ngx_files.c
src/event/ngx_event_timer.h
src/http/modules/ngx_http_static_module.c
src/http/modules/ngx_http_upstream_hash_module.c
src/http/modules/ngx_http_auth_basic_module.c
src/http/modules/ngx_http_autoindex_module.c
src/os/unix/ngx_setaffinity.c
src/core/ngx_syslog.c
src/os/unix/ngx_alloc.c
src/http/modules/ngx_http_geo_module.c
src/core/ngx_string.c
src/core/ngx_inet.c
src/core/ngx_hash.c
src/core/ngx_crypt.c
src/http/modules/ngx_http_map_module.c
src/core/ngx_list.h
src/core/ngx_regex.c
src/os/unix/ngx_posix_init.c
src/http/ngx_http_special_response.c
src/os/unix/ngx_udp_send.c
src/http/modules/ngx_http_upstream_least_conn_module.c
src/core/ngx_buf.c
src/http/modules/ngx_http_browser_module.c
src/os/unix/ngx_linux_init.c
src/http/modules/ngx_http_split_clients_module.c
src/core/ngx_proxy_protocol.c
src/core/ngx_parse.c
src/core/ngx_times.c
src/core/ngx_crc32.c
execve
Distance: 1
#Files: 1
Distance: 2
#Files: 10
Distance: 3
#Files: 47
Distance: 4
#Files: 50
Distance: 5
#Files: 3

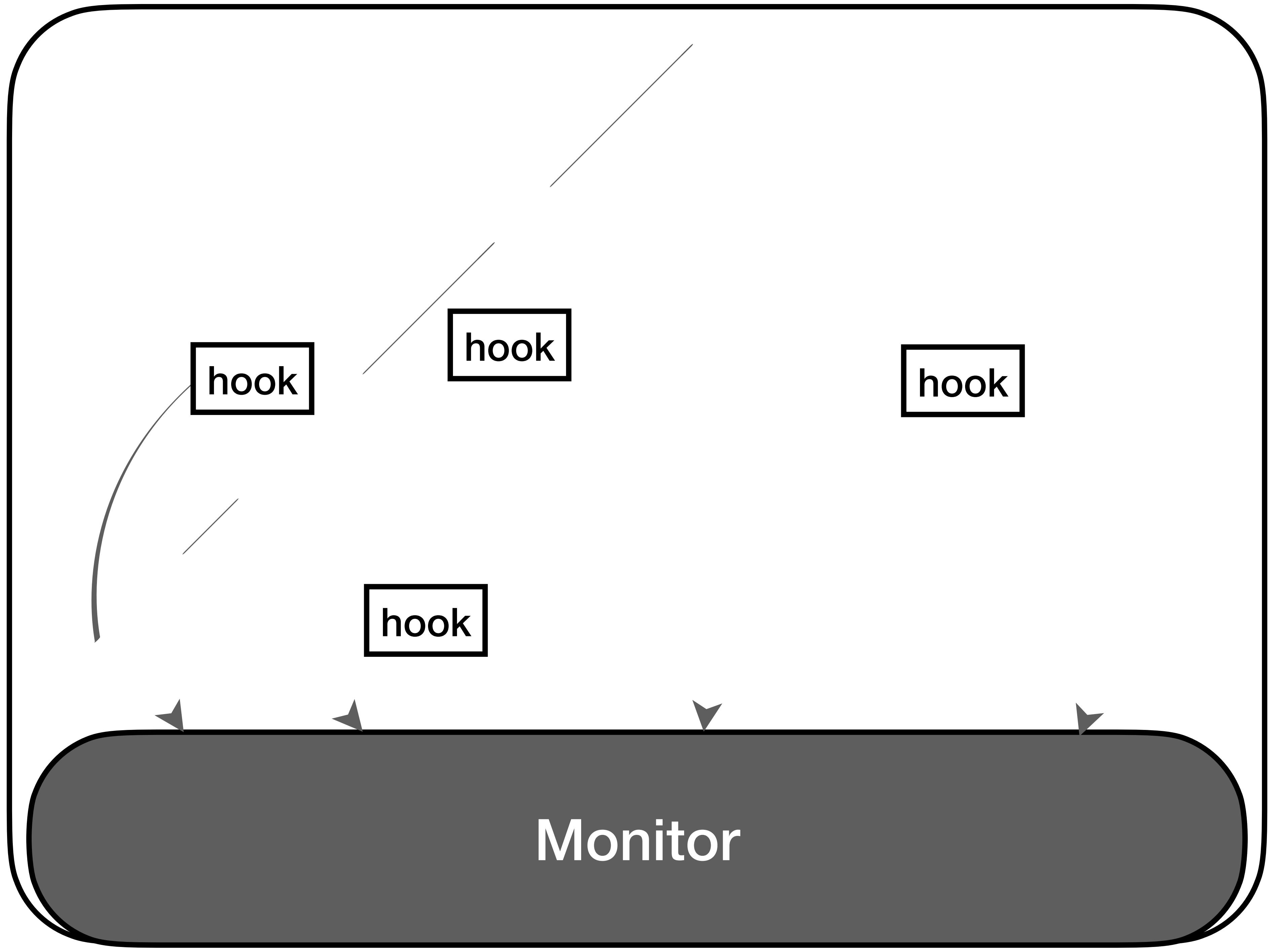
Exposure Reduction Metric

System	System Interfaces										
	`system`	`fork`	`exec*`	`popen`	`open*`	`read*`	`write*`	`ioctl`	`dup`	`mmap`	`mprotect`
libexpat	0 / 16	0 / 16	0 / 16	0 / 16	1 / 16	1 / 16	0 / 16	0 / 16	0 / 16	0 / 16	0 / 16
libroxml	0 / 40	0 / 40	0 / 40	0 / 40	2 / 40	1 / 40	1 / 40	0 / 40	1 / 40	0 / 40	0 / 40
jansson	0 / 37	0 / 37	0 / 37	0 / 37	3 / 37	2 / 37	1 / 37	0 / 37	0 / 37	0 / 37	0 / 37
json-c	0 / 56	0 / 56	0 / 56	0 / 56	2 / 56	2 / 56	1 / 56	0 / 56	0 / 56	0 / 56	0 / 56
facil.io	0 / 102	1 / 102	0 / 102	0 / 102	4 / 102	5 / 102	4 / 102	0 / 102	0 / 102	1 / 102	0 / 102
libxml2	0 / 141	0 / 141	0 / 141	0 / 141	3 / 141	3 / 141	4 / 141	0 / 141	1 / 141	0 / 141	0 / 141
Nginx	0 / 260	2 / 260	1 / 260	0 / 260	10 / 260	4 / 260	4 / 260	4 / 260	2 / 260	2 / 260	0 / 260
CPython	1 / 600	1 / 600	1 / 600	0 / 600	4 / 600	6 / 600	4 / 600	2 / 600	1 / 600	2 / 600	1 / 600
MuPDF	1 / 613	0 / 613	0 / 613	0 / 613	8 / 613	8 / 613	5 / 613	0 / 613	0 / 613	0 / 613	0 / 613
PHP	0 / 1056	2 / 1056	1 / 1056	4 / 1056	16 / 1056	8 / 1056	7 / 1056	0 / 1056	2 / 1056	5 / 1056	1 / 1056
Gimp	0 / 4086	1 / 4086	1 / 4086	0 / 4086	4 / 4086	4 / 4086	1 / 4086	0 / 4086	0 / 4086	0 / 4086	0 / 4086



Why is this useful?





Objective: can we build tools to help refactor?

Several research challenges obstruct

- **Representation and Abstractions**: What is a privilege?
- **Flexible**, **Scalable**, **Systematic Policy Synthesis**:
  - What are the subjects? objects? operations?
  - How to get as non-expert?
  - How to protect all resources not just known critical?
- **Security**: How to measure the security of the system?

Today

Program Capability Graph

Object-Encapsulation Model: automated program reasoning through object-oriented grouping

An end-to-end compiler for automated privilege and compartmentalization analysis

Fully automated program partitioning and privilege reduction measurement with **Inductive Approach**

Security measurement using deductive sensitive and suspicious type labels

Key results: 50% private, 75% for linux, syscall exposure very low, automated library analysis, privilege locality arises in both good and bad cases

FINI!

RICE

TRAIL OF BITS

Appendix

Nginx External RO/RW Global Variable Rank

#	Encapsulated-Object	Ext. R/O ↑
1	os/unix/ngx_process_cycle.c	47
2	event/ngx_event_timer.h	43
3	event/ngx_event.c	41
4	http/ngx_http_core_module.c	40
5	http/ngx_http_upstream.c	39
6	http/ngx_http_request.c	38
7	core/nginx.c	30
8	event/modules/ngx_epoll_module.c	27
9	http/modules/ngx_http_proxy_module.c	27
10	http/ngx_http_variables.c	25

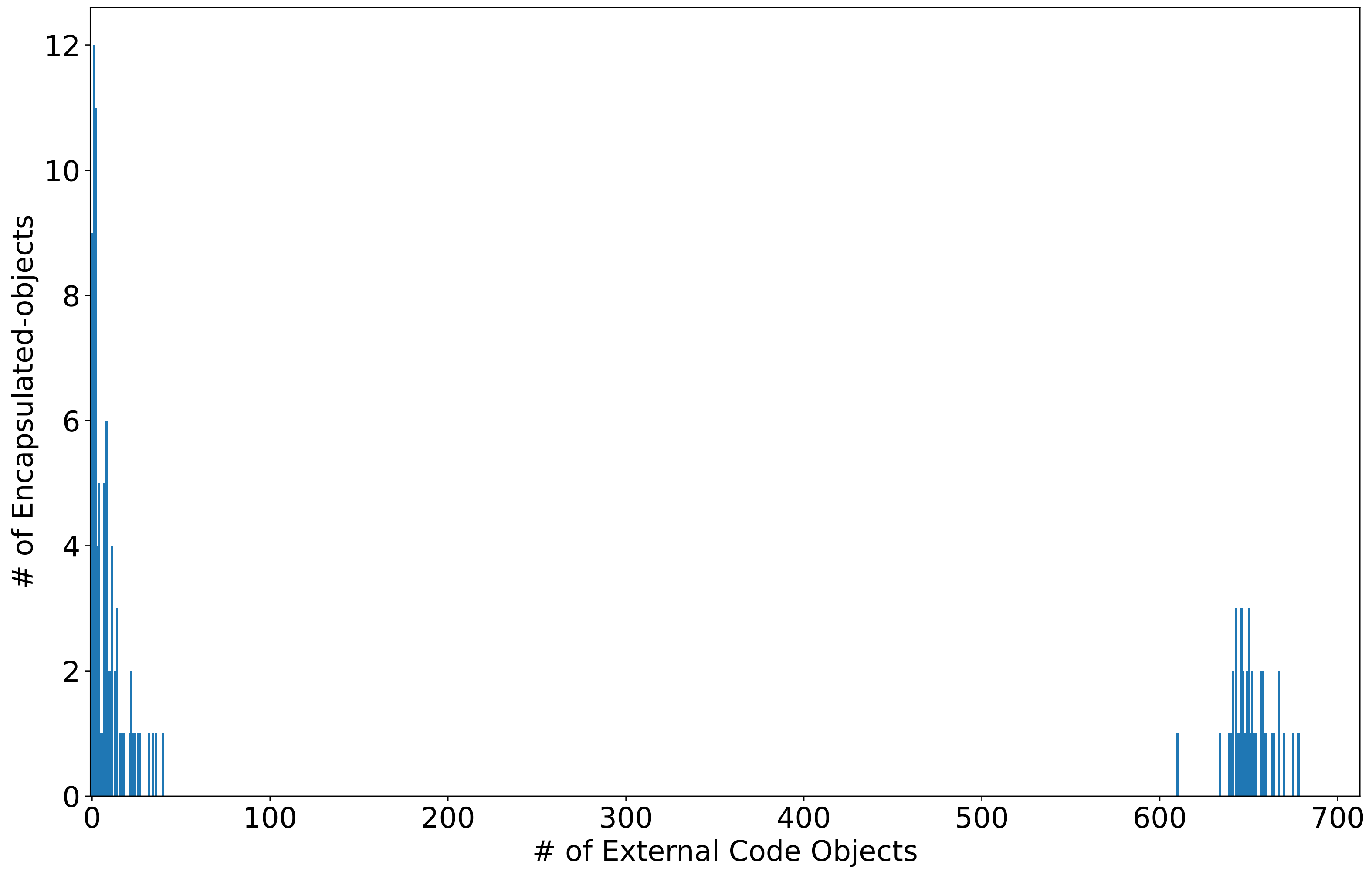
#	Encapsulated-Object	Ext. R/W ↑
1	os/unix/ngx_process_cycle.c	36
2	core/ngx_times.c	30
3	core/nginx.c	26
4	os/unix/ngx_process.c	20
5	event/ngx_event.c	17
6	event/modules/ngx_epoll_module.c	16
7	core/ngx_regex.c	9
8	core/ngx_cycle.c	7
9	event/ngx_event_accept.c	6
10	http/modules/ngx_http_ssi_filter_module.c	6

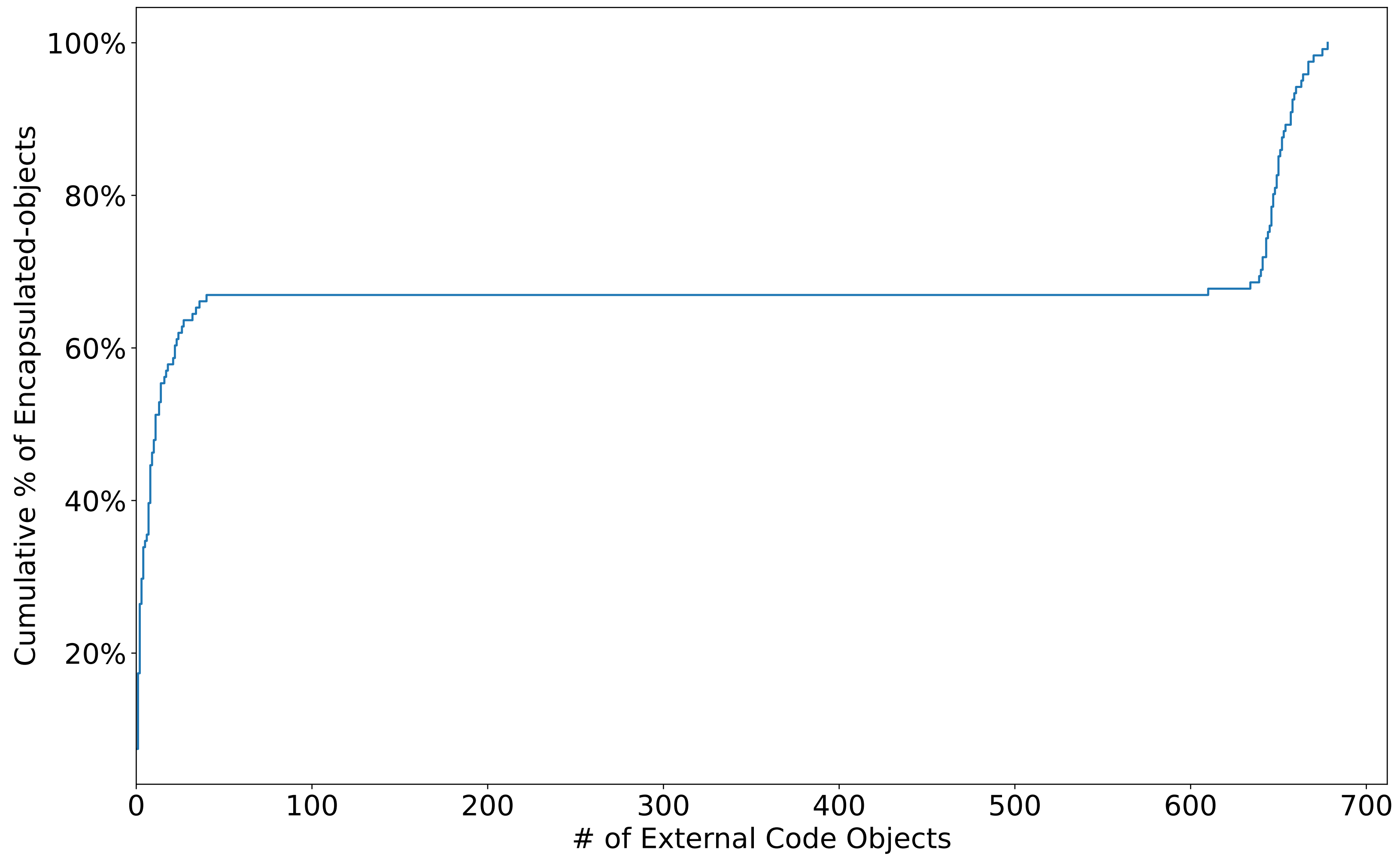
External Access Ratio Metric (Cont.)

Private Objs.
Public Objs.
100%
75%
50%
25%
0%
Percent of Total Function Count
read
write
close
newstat
newfstat
newlstat
poll
lseek
mmap
mprotect
munmap
brk
rt_sigaction
rt_sigprocmask
rt_sigreturn
ioctl
pread64
access
pipe
select
mremap
dup
dup2
alarm
getpid
socket
connect
sendto
recvfrom
sendmsg
recvmsg
bind
getsockname
setsockopt
getsockopt
clone
execve
wait4
kill
newuname
fcntl
flock
fsync
fdatasync
ftruncate
getcwd
chdir
rename
mkdir
rmdir
creat
link
unlink
symlink
readlink
chmod
fchmod
chown
fchown
lchown
umask
sysinfo
getuid
getgid
setuid
setgid
geteuid
getegid
getppid
getpgrp
setsid
getgroups
setgroups
setresuid
getresuid
setresgid
getresgid
getpgid
getsid
capget
capset
sigaltstack
statfs
fstatfs
prctl
arch_prctl
gettid
getxattr
lgetxattr
futex
getdents64
set_tid_address
exit_group
utimes
openat
mkdirat
fchownat
newfstatat
unlinkat
symlinkat
readlinkat
faccessat
pselect6
ppoll
set_robust_list
sync_file_range
utimensat
dup3
pipe2
prlimit64
sendmmsg
renameat2
getrandom

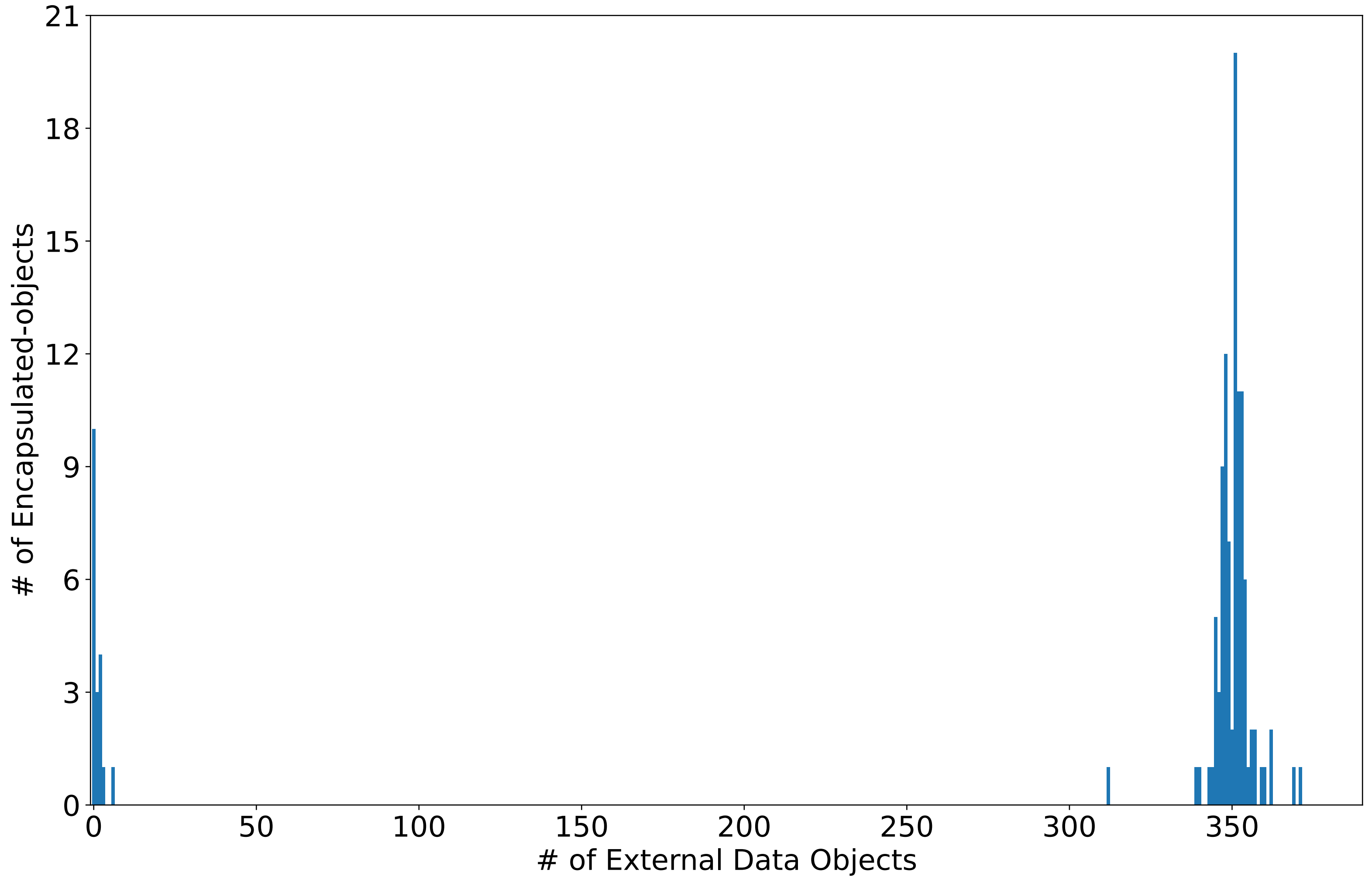
External Access Distribution per Object

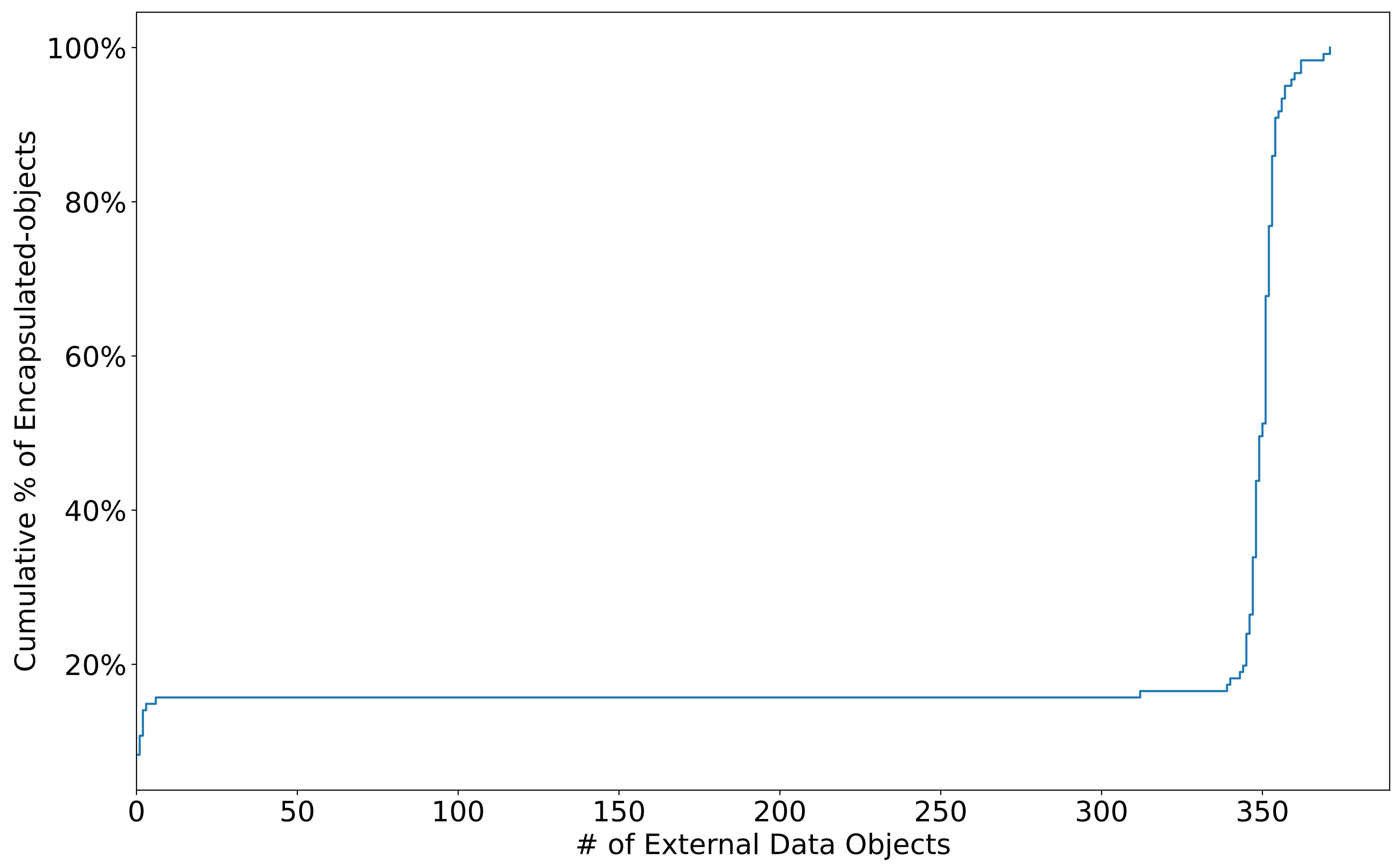
Code Objects





Data Objects





PDF **CDF**

Read / Write Relation for LLVM Instructions and Operations

Instruction	Read-Only / Writable
LoadInst	RO
StoreInst	RW
GetElementPtrInst	Depends on forward uses
CallInst	RO
CallBrInst	RO
ICmpInst	RO
AtomicRMWInst	RW
AtomicCmpXchgInst	RW
ReturnInst	RO
SelectInst	RO
PtrToIntInst	RO
PtrToIntOpr	RO
GEPOperator	Depends on forward uses
BitCastOperator	RO
PHINode	RO
Constant	N/A

Evaluated Programs and Libraries & Generation Time

Parsing Libraries	Software System
libroxml	Nginx
jansson	CPython
facil.io	PHP
libxml2	MuPDF
json-c	gimp

Target Program	Generation Runtime
libroxml	0.854s
jansson	0.824s
facil.io	8.290s
libxml2	2m58.852s
json-c	0.884s
Nginx	6m3.503s
CPython	119m24.766s
PHP	180m44.243s
MuPDF	60m43.136s
gimp	1m26.717s