Secured Routines:

Language-based Construction of Trusted Environments

Adrien Ghosn, James R. Larus, Edouard Bugnion
EPFL, Switzerland

TEEs

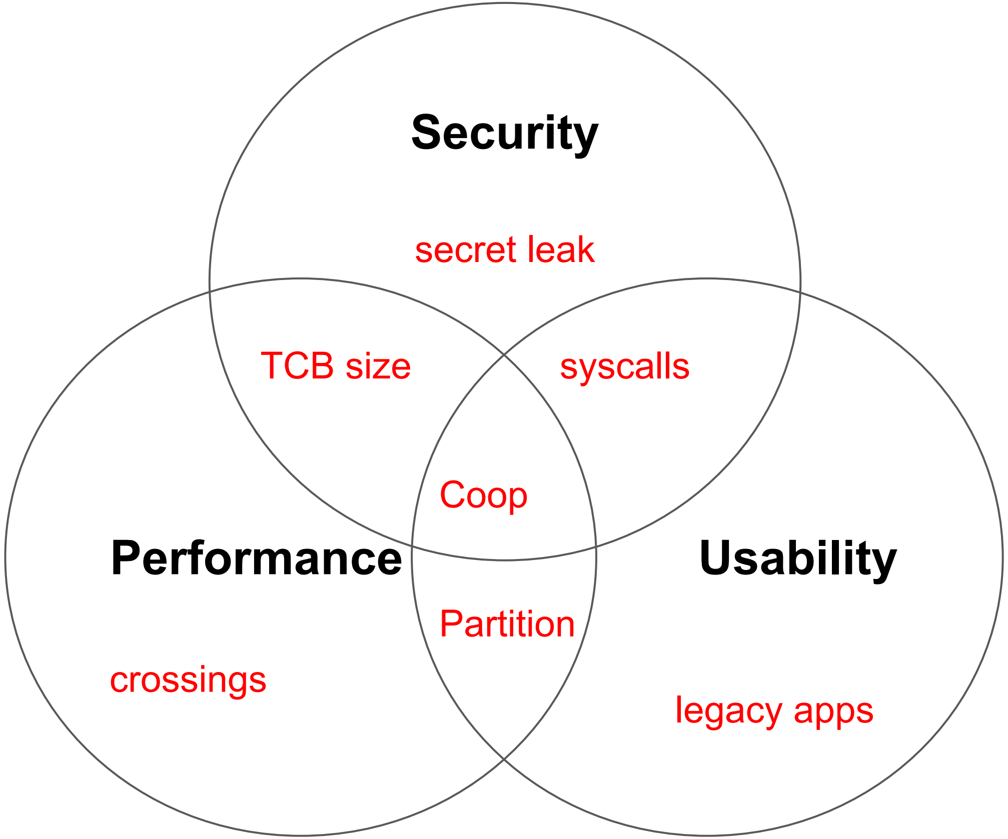
The solution to the problem of trust in the Cloud.

Confidentiality & Integrity.

Intel SGX enclaves.

… **SGX is pretty hard to use!**

TEE-Support Challenges

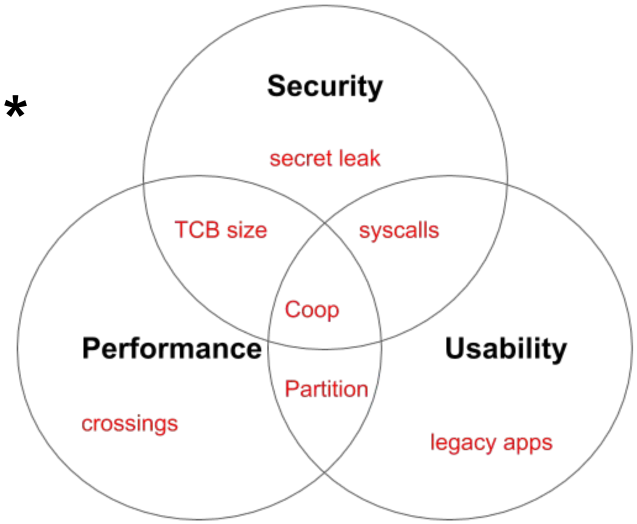


TEEs-Support Challenges

Guess what…

… A compiler can do that for you! *

*(given the right programming abstraction)



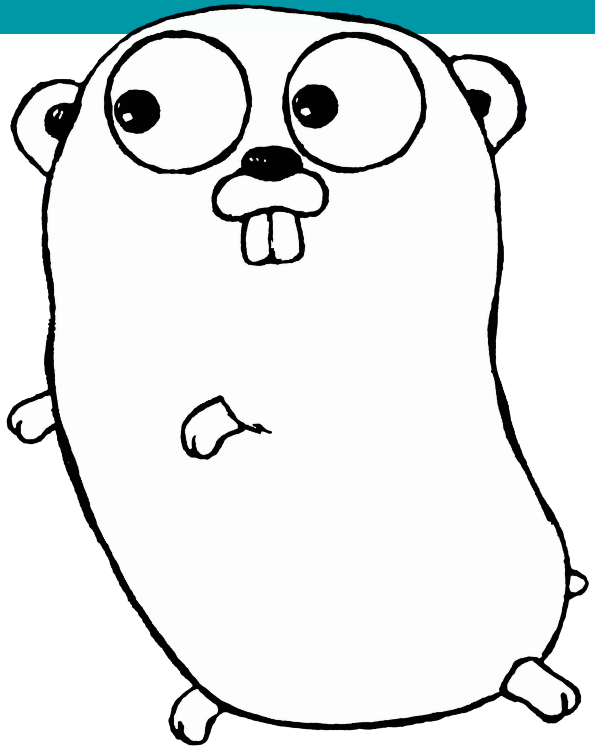
Secured Routines

A language-level approach for TEEs.

A familiar programming abstraction.

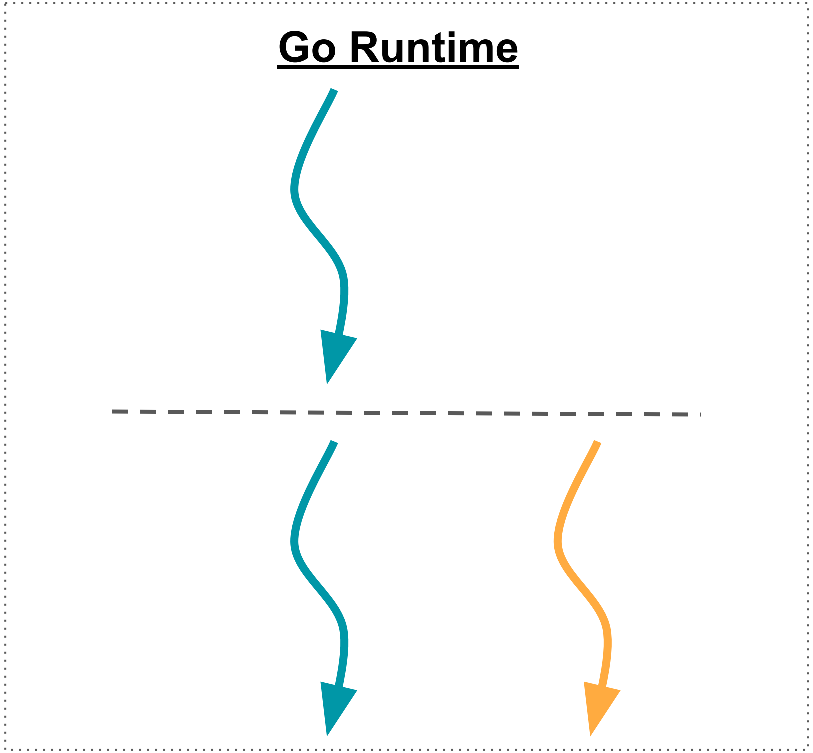
Implemented **GOTEE,** our fork of the Go compiler.

Adding a single keyword, **`gosecure`**…



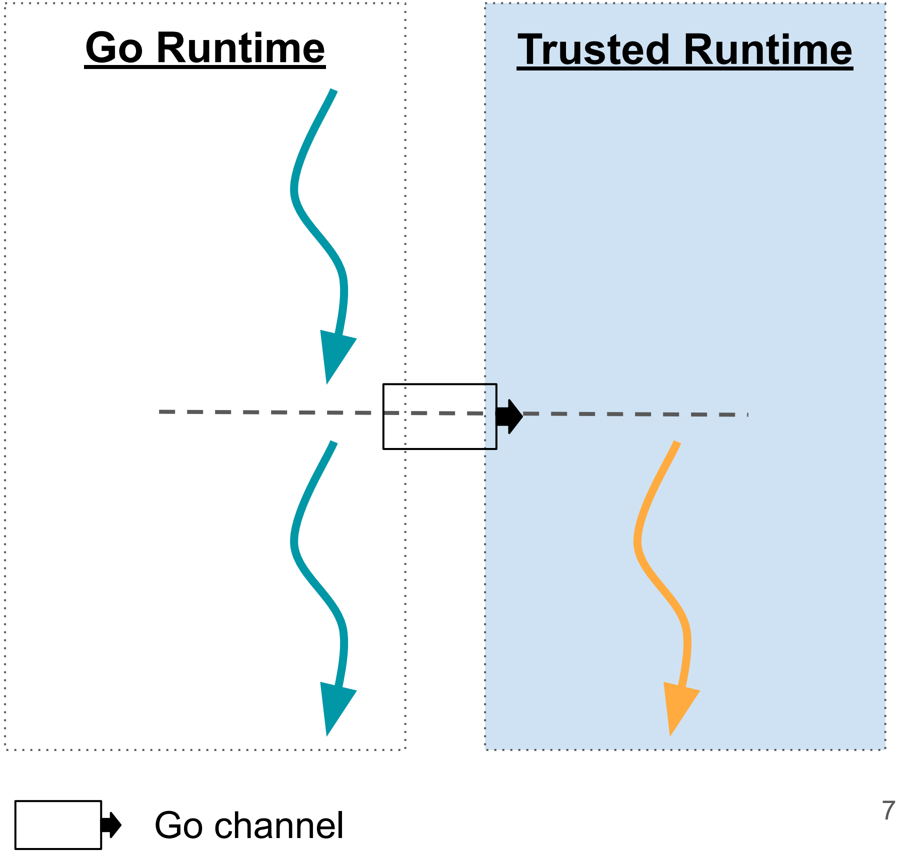
Go Execution Model

```
func main() {

    go foo()

    ...

}
```

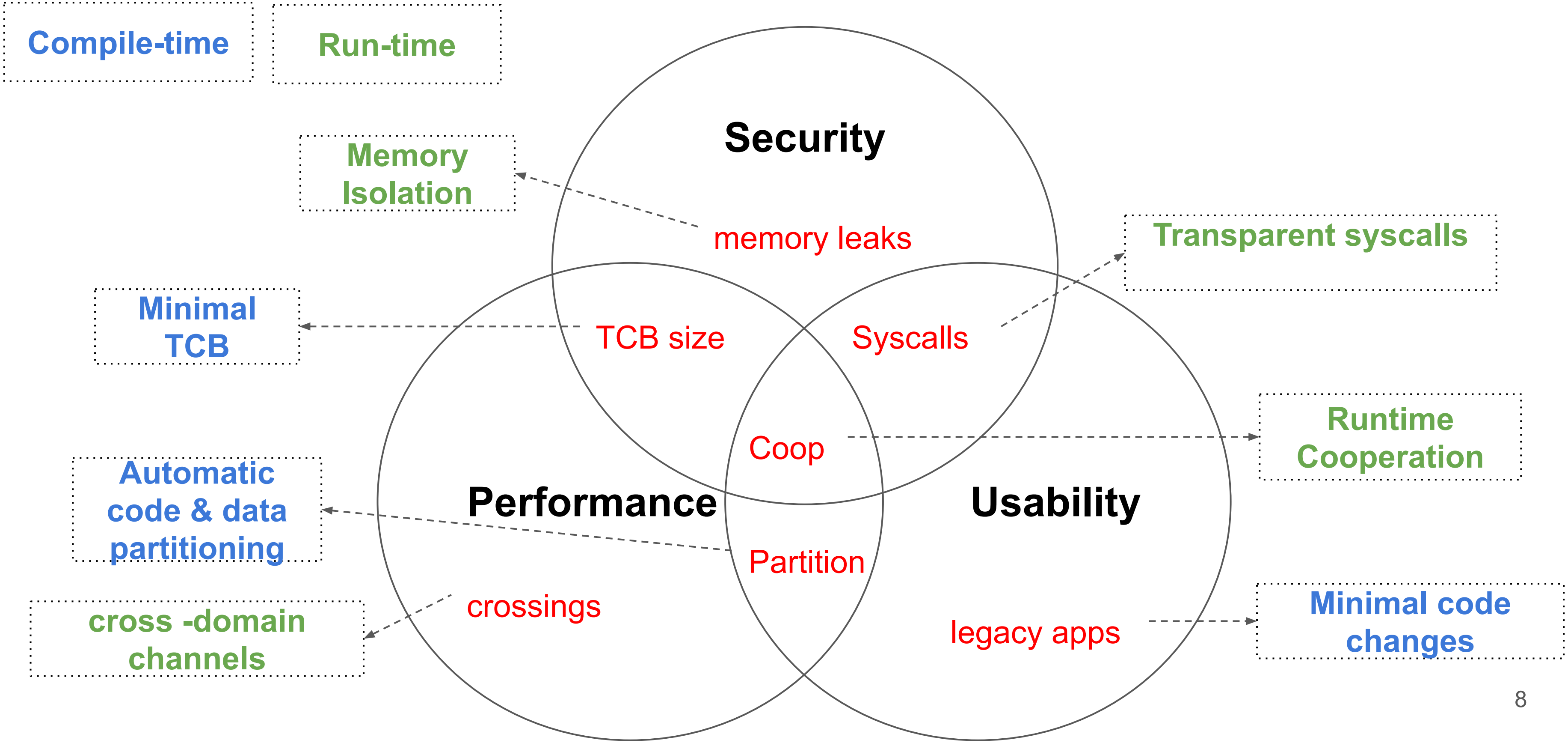


GOTEE Execution Model

```
func main() {

    gosecure foo()

    ...

}
```



GOTEE



Compile-time

Run-time

Security

Performance

Usability

memory leaks

Memory Isolation

TCB size

Minimal TCB

Syscalls

Transparent syscalls

Coop

Runtime Cooperation

Partition

Automatic code & data partitioning

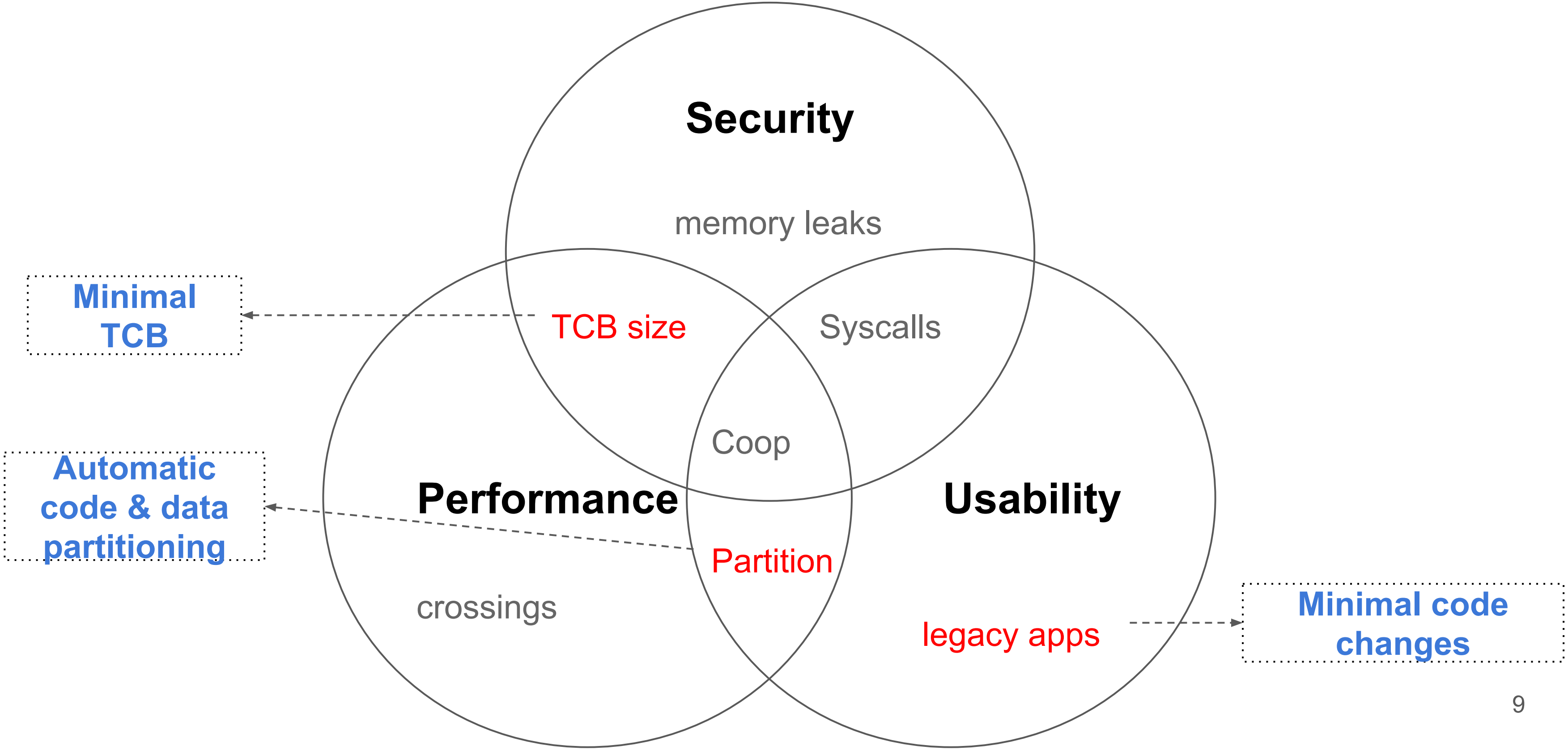
crossings

cross -domain channels

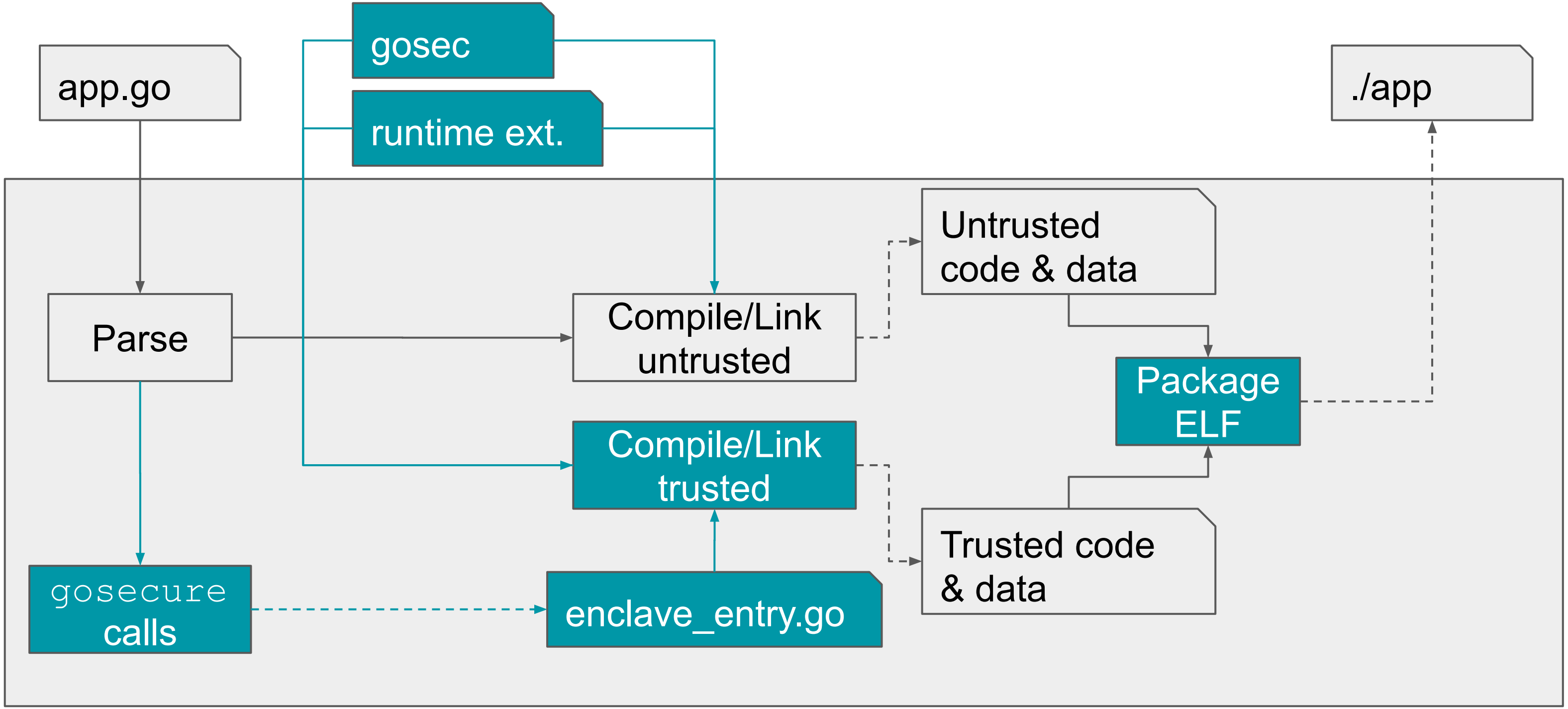
legacy apps

Minimal code changes

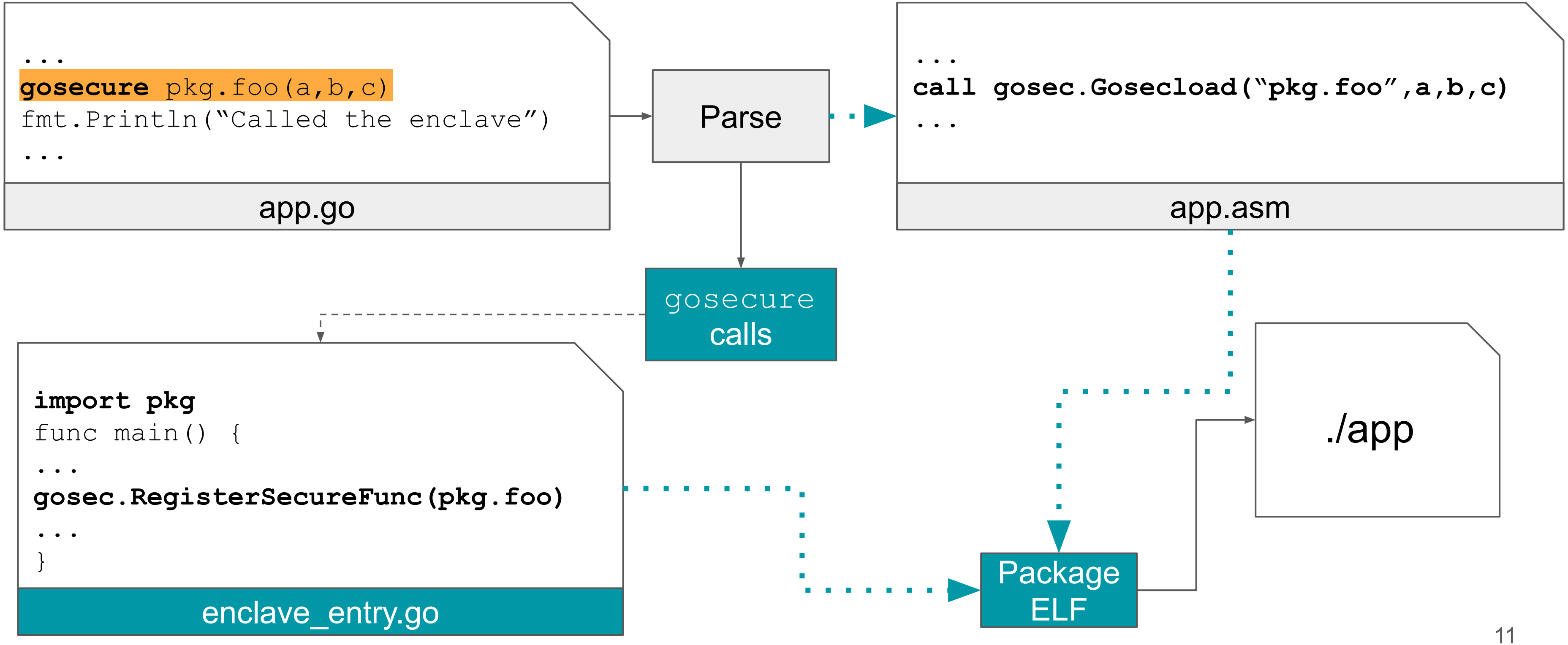
@Compile-time



GOTEE compile-time overview



Partitioning Code & Data



```
...
gosecure pkg.foo(a,b,c)
fmt.Println("Called the enclave")
...
```

app.go

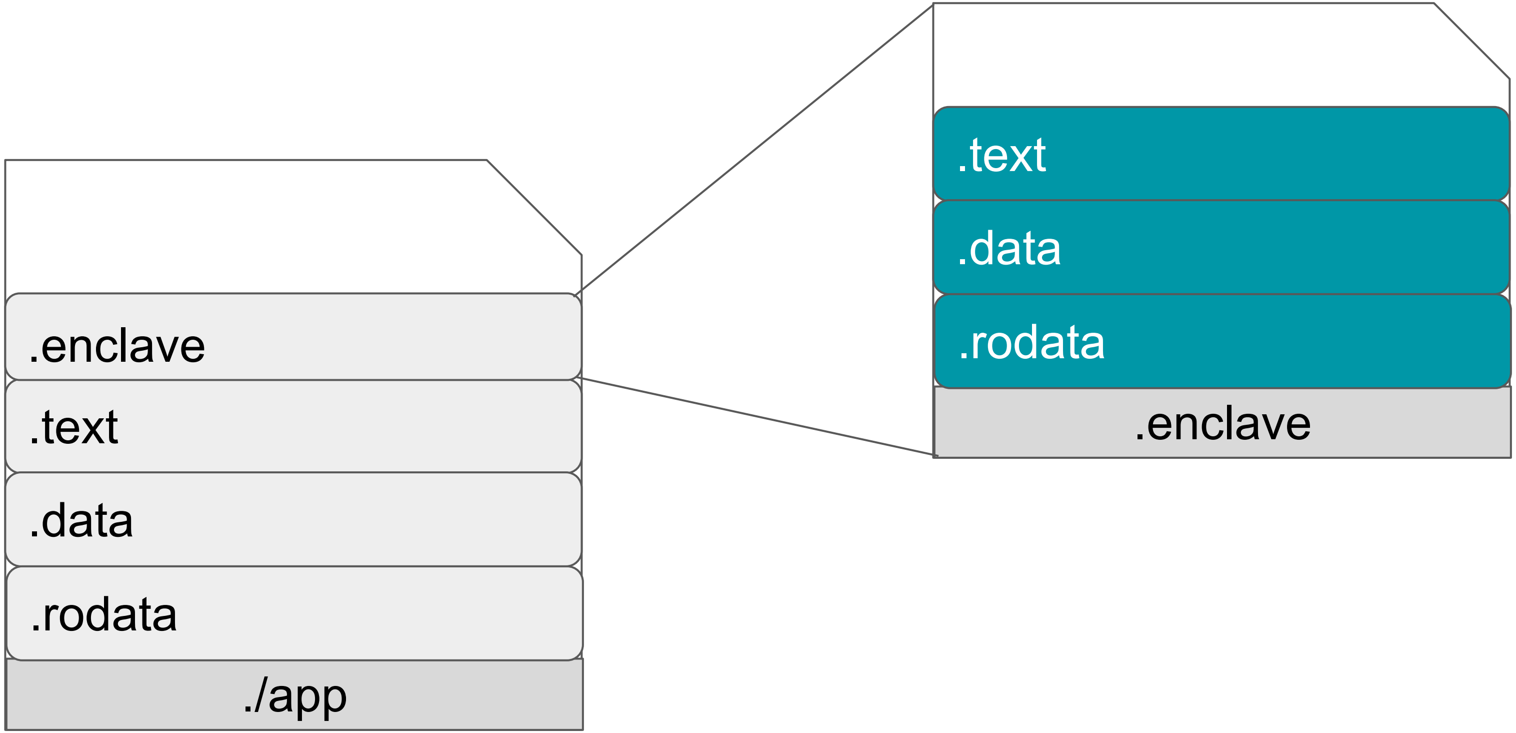
```
...
call gosec.Gosecload("pkg.foo",a,b,c)
...
```

app.asm

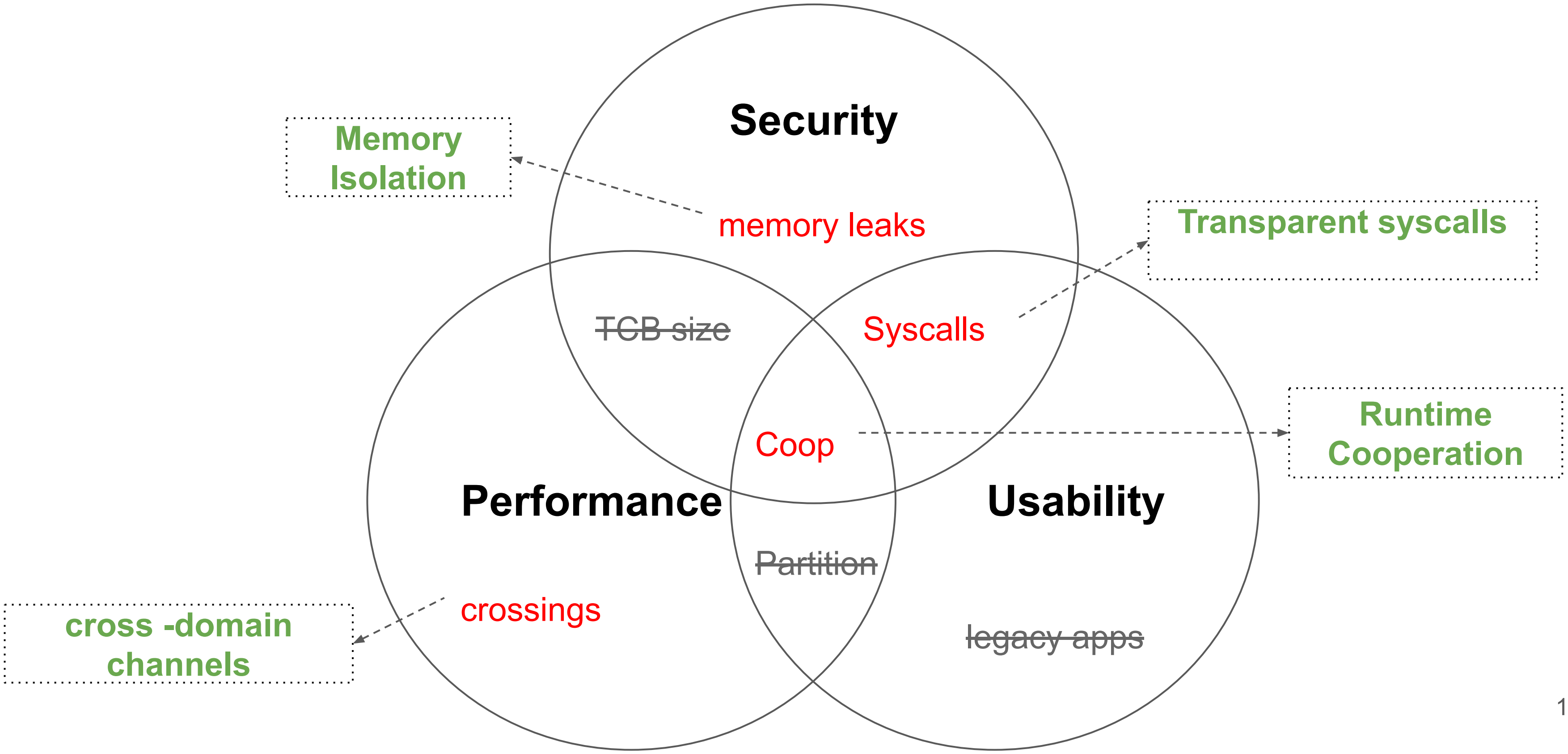
```
import pkg
func main() {
...
gosec.RegisterSecureFunc(pkg.foo)
...
}
```

enclave_entry.go

Final executable



@Run-time



Security

Performance

Usability

memory leaks

~~TCB size~~

Syscalls

Coop

~~Partition~~

crossings

~~legacy apps~~

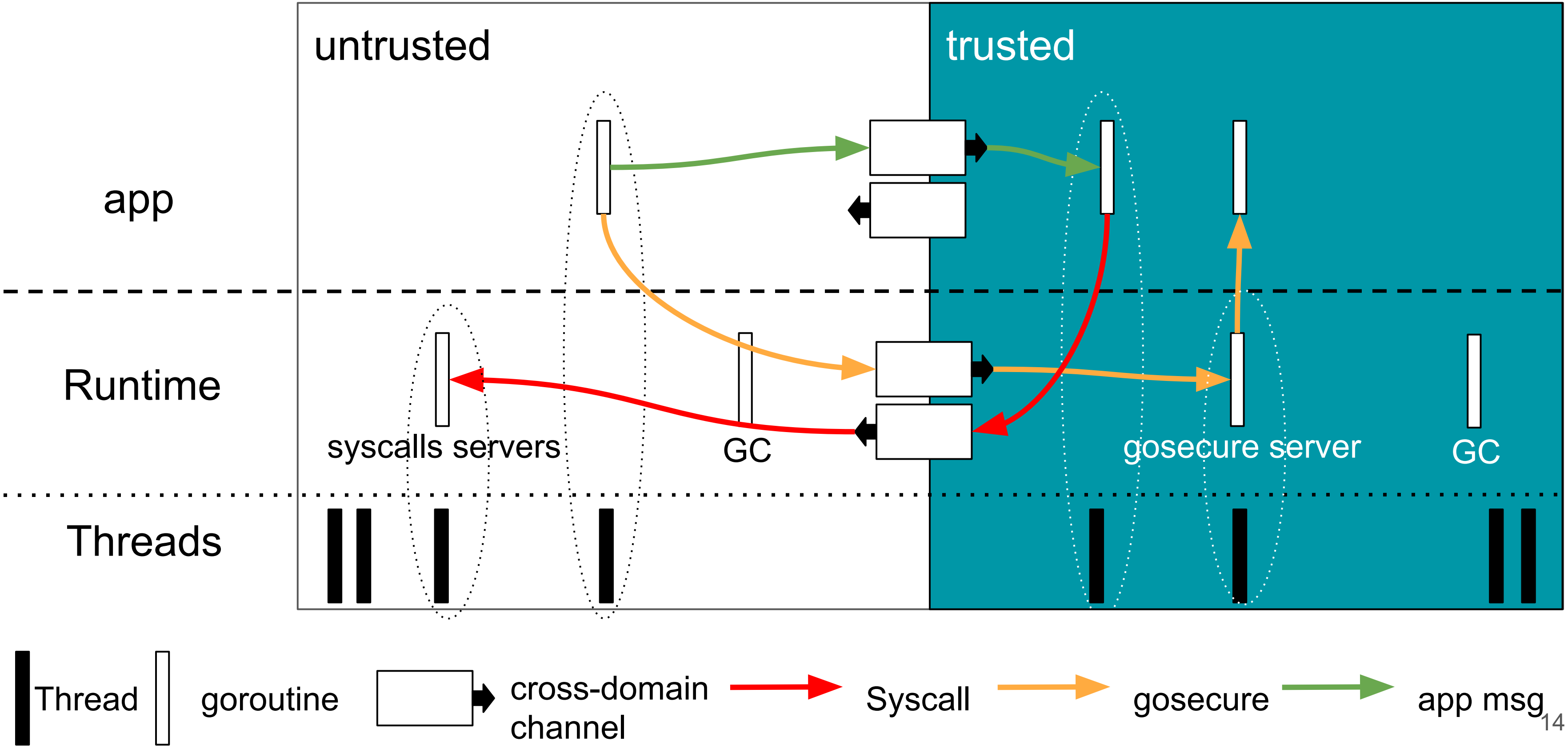
Memory Isolation

Transparent syscalls

Runtime Cooperation

cross -domain channels

GOTEE run-time overview



Crossings

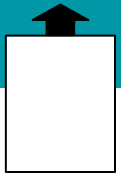
These □➡ are Go typed & synchronized channels extended by GOTEE.

Automatic deep-copy for cross-domain communication.

Single point of interaction between the domains…

Get rid of **crossings**!

Deep-copy



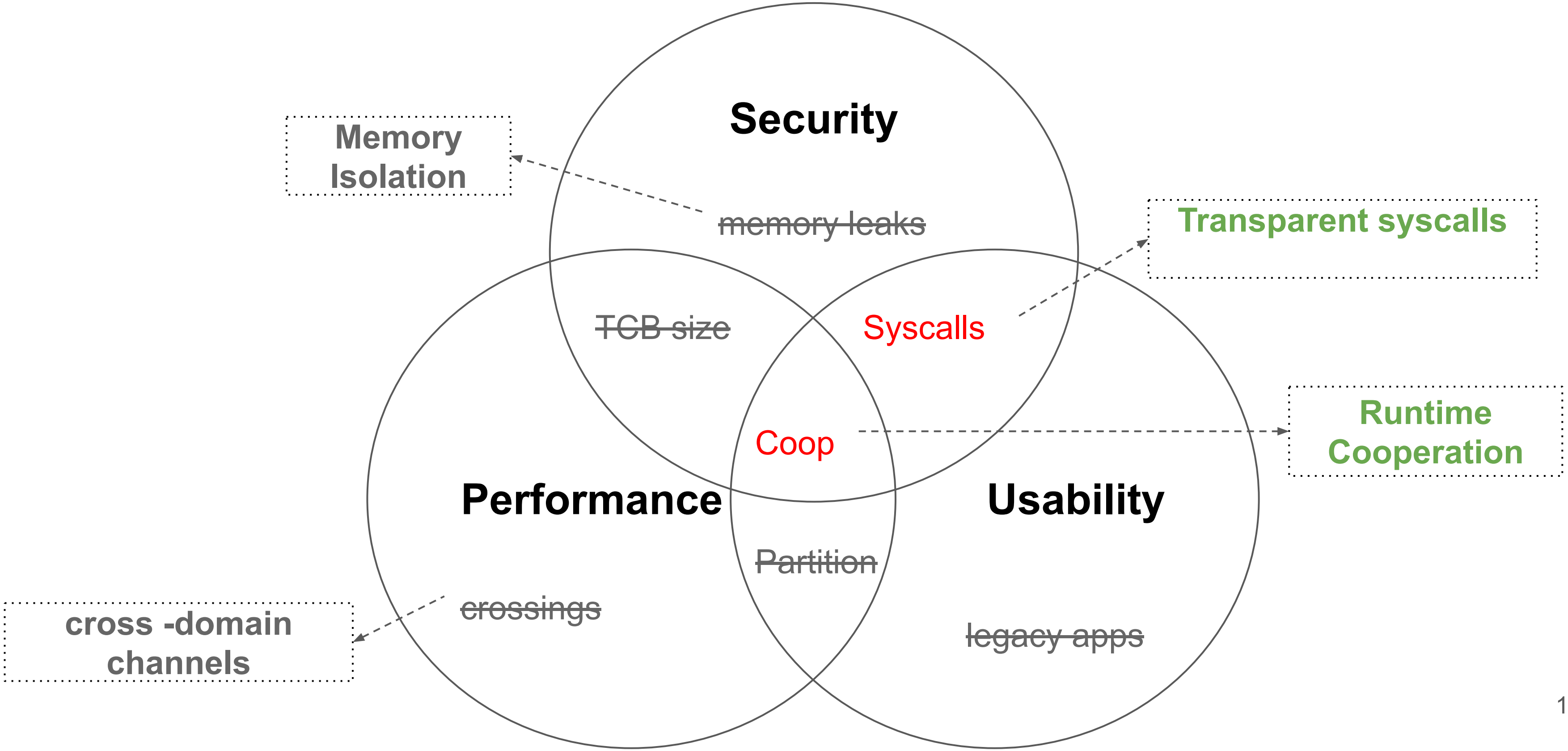
No cross-domain memory references.

Similar to network marshalling.

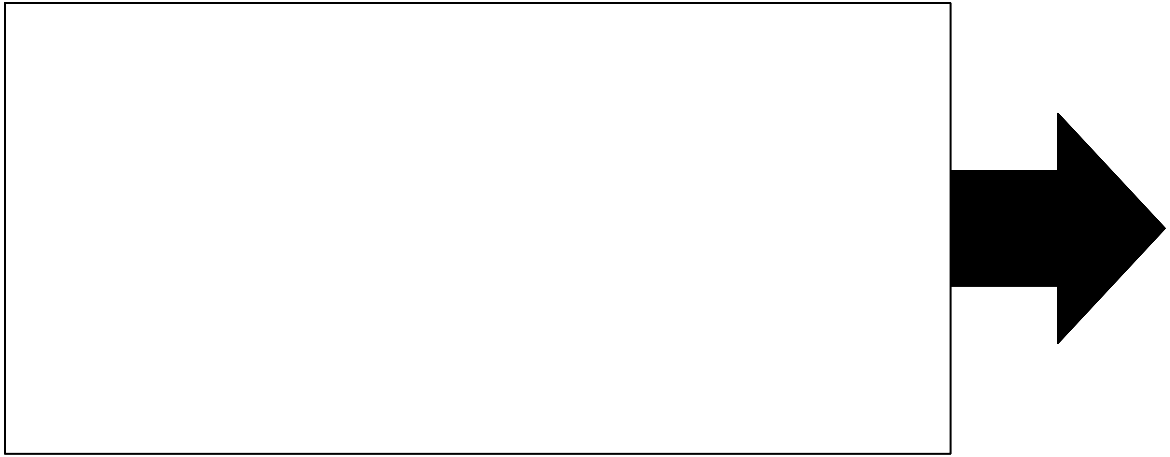
Implemented via reflection.

Independent GCs & enhance **memory isolation**!

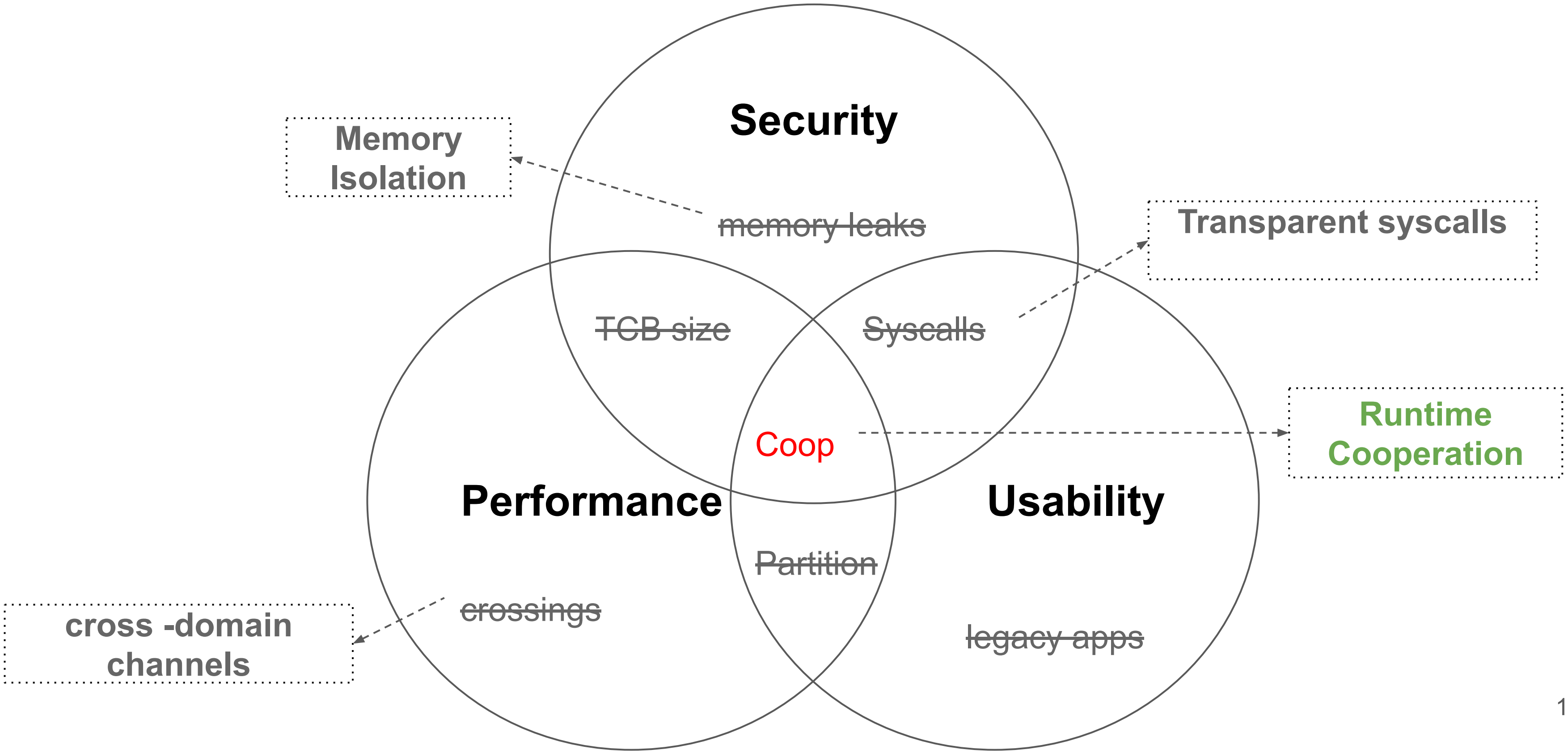
@Run-time(2)


Memory Isolation
Security
~~memory leaks~~
Transparent syscalls
~~TCB size~~
Syscalls
Runtime Cooperation
Coop
Performance
Usability
~~Partition~~
~~crossings~~
cross -domain channels
~~legacy apps~~

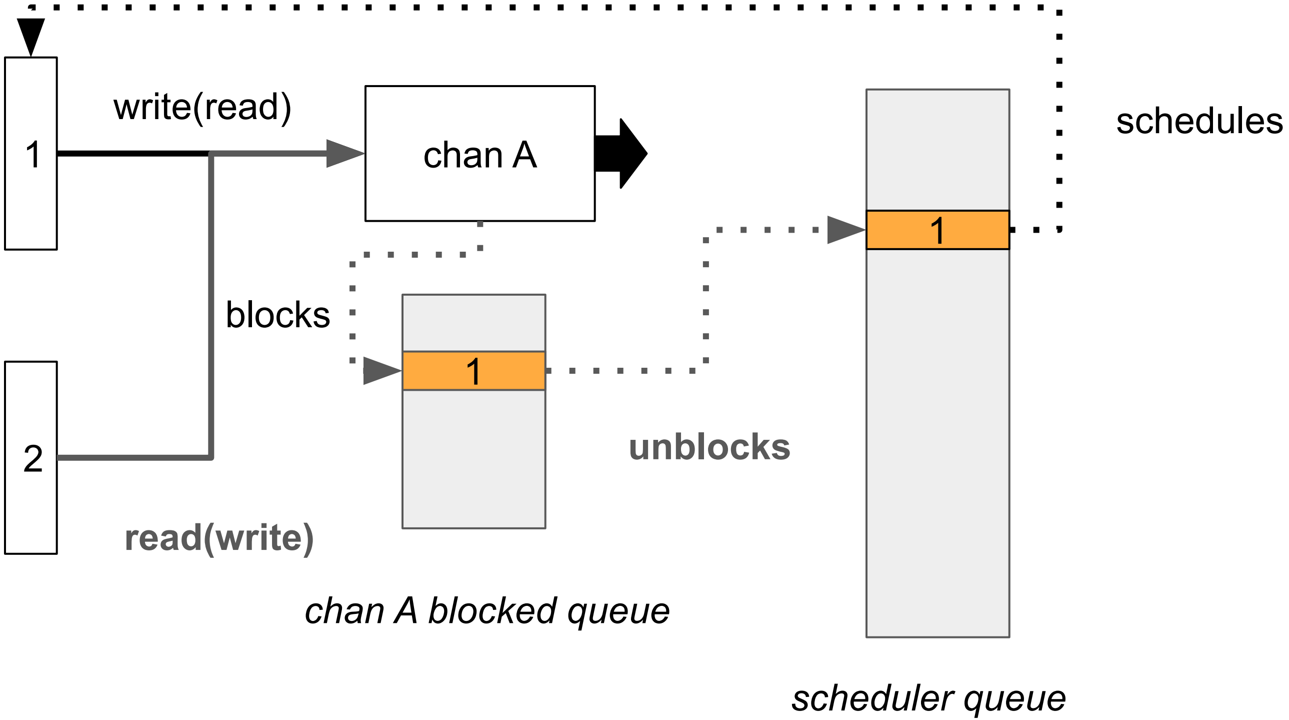
Syscalls



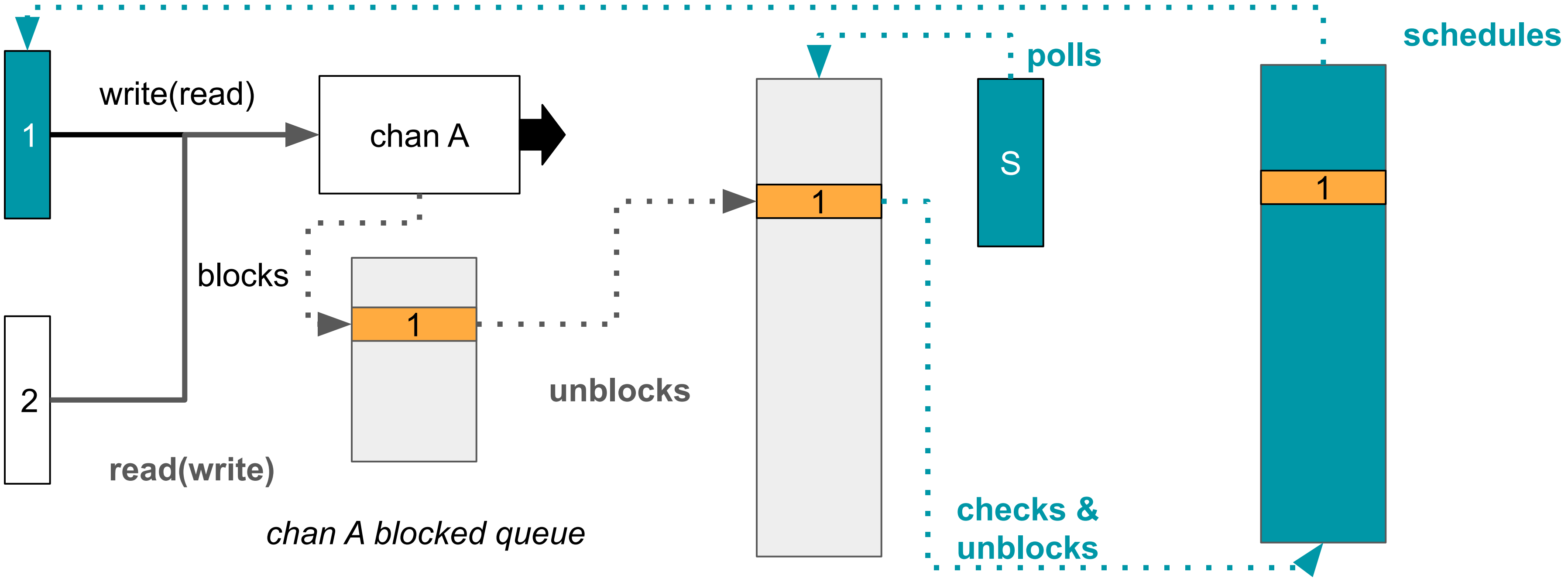
@Run-time(3)


Memory Isolation
Security
~~memory leaks~~
Transparent syscalls
~~TCB size~~
~~Syscalls~~
Runtime Cooperation
Coop
Performance
Usability
~~Partition~~
~~crossings~~
cross -domain channels
~~legacy apps~~

In Vanilla Go



In GOTEE



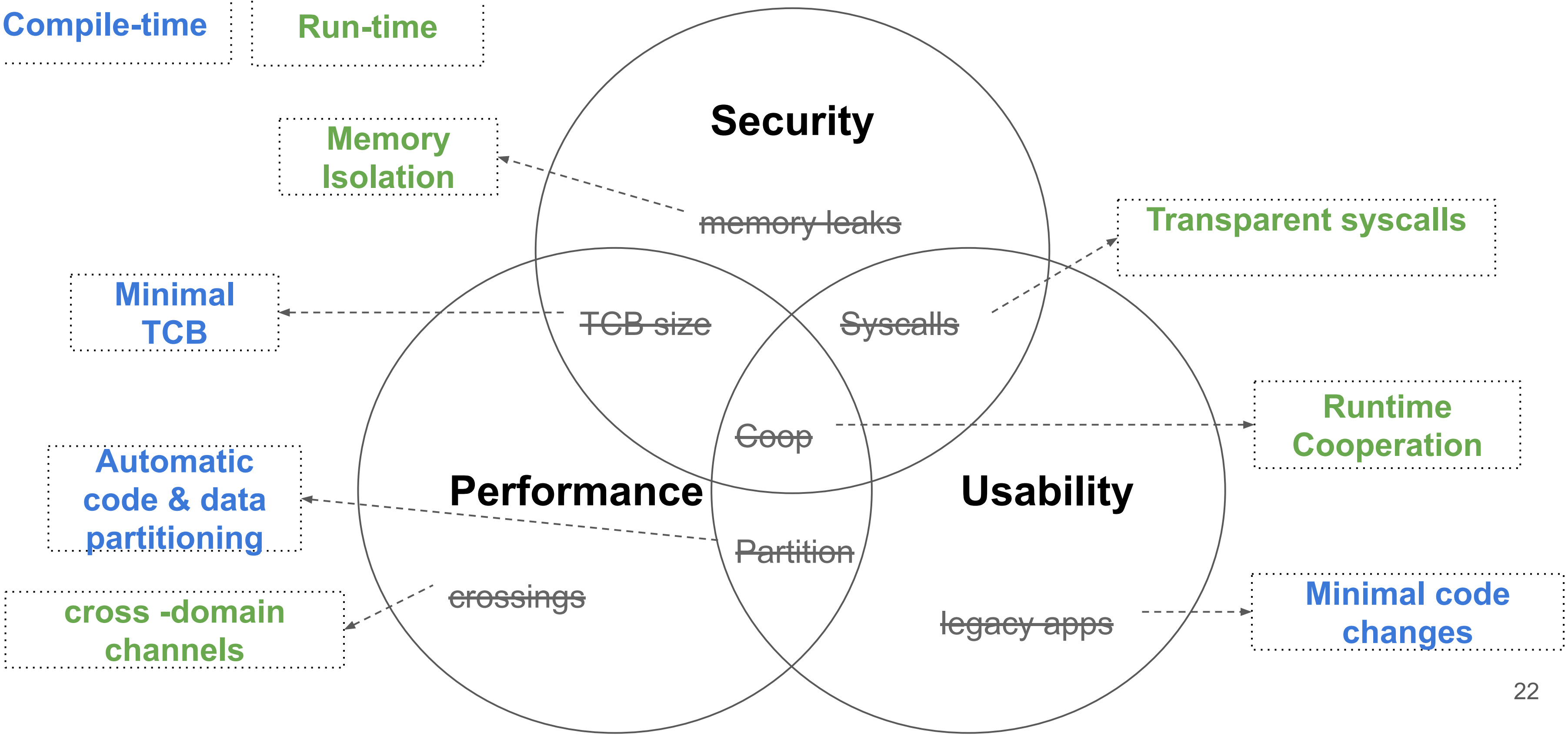
~~scheduler queue~~
TrustedReadyQueue

(Trusted) scheduler queue

GOTEE

Compile-time

Run-time



Security

~~memory leaks~~

Memory Isolation

~~TCB size~~

Minimal TCB

~~Syscalls~~

Transparent syscalls

~~Coop~~

Runtime Cooperation

Performance

Usability

~~Partition~~

Automatic code & data partitioning

~~crossings~~

cross -domain channels

~~legacy apps~~

Minimal code changes

Evaluation

Micro-benchmarks.

Macro-benchmarks.

Code & Data sizes.

Evaluation @micro-benchmarks

	Go	GOTEE	Intel SGX SDK	Ratios GOTEE/SDK
Syscall (µs)	0.23	1.35	3.69	**2.7x**
Gosecure+block (µs)	0.30	1.5	3.50	**2.3x**
Throughput (KOPS/trusted thread)	6000	1460	281	**5.2x**

Evaluation @macro-benchmarks

Trusted SSH-server

- Everything in the enclave, no application-level code modification.

Fine-grained TLS

- Private key only available inside the enclave.
- **9 LOC** in `tls` and **35 LOC** of new code in enclcert.
- 88% of native throughput (handshakes).

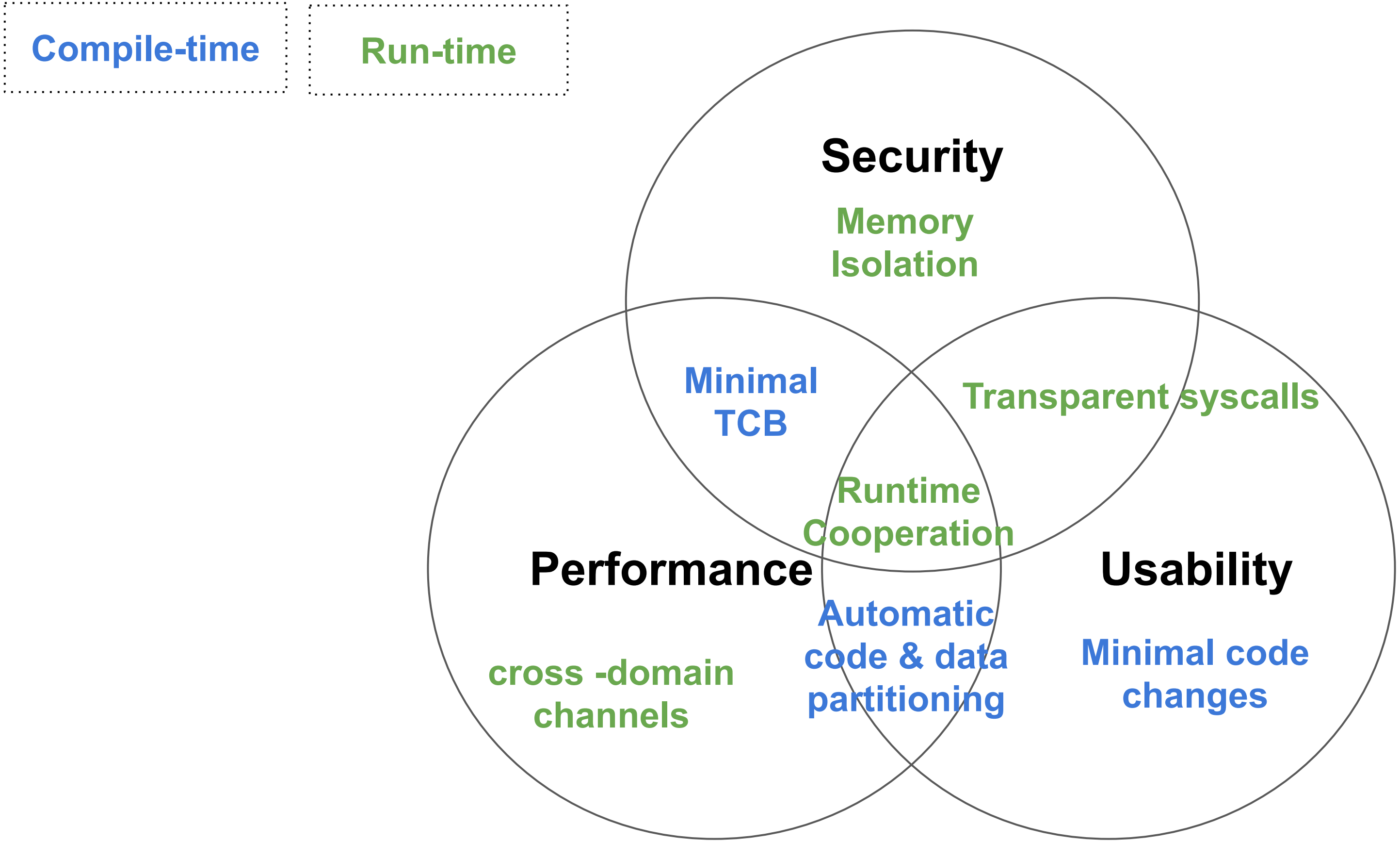
Go-ethereum trusted keystore

- 1 day, 500 lines of code.

Evaluation @Code & Data sizes

	TCB (KB)	Pkg Deps	Application LOC
runtime GOTEE	793	runtime	-
Hello World	884	++ fmt, syscall, io, unicode...	13
SSH-server	2437	++crypto, golang.org/crypto/*, net, encoding...	71
Keystore	**3936**	++crypto/ecdsa, crypto/elliptic, crypto/aes...	474
SGX SDK runtime	75	-	-
SGX SDK Hello	166	-	355

This was GOTEE. Merci Beaucoup !


Compile-time
Run-time
Security
Memory Isolation
Minimal TCB
Transparent syscalls
Runtime Cooperation
Performance
Usability
Automatic code & data partitioning
cross -domain channels
Minimal code changes