The Secure Multiserver Operating System (SMOS) Framework

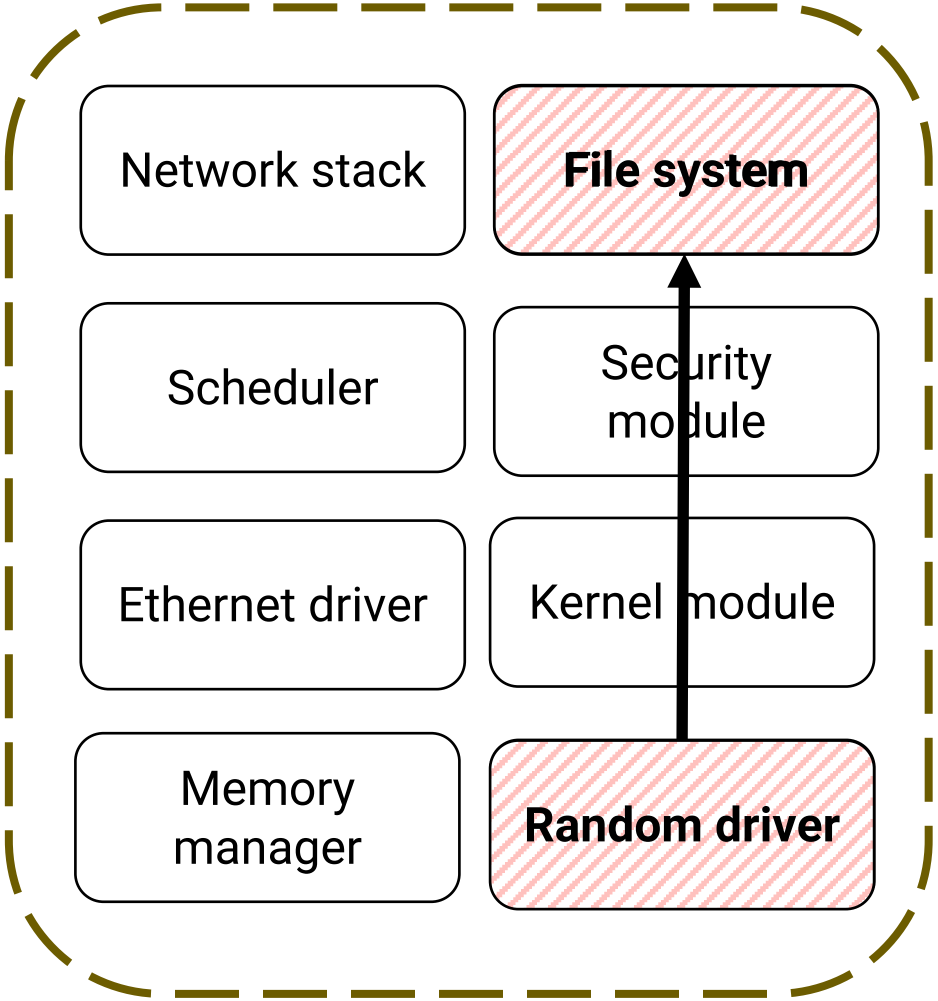
Alwin Joshy, Kevin Elphinstone, Gernot Heiser, Craig McLaughlin

Motivation

Current operating systems are **not** secure

Linux > 25 million SLoC

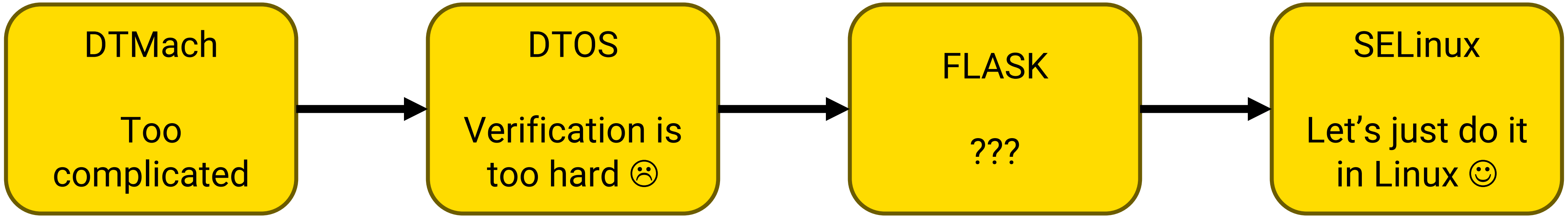
Inevitable trajectory of monolithic kernel design



We need secure operating systems

Current operating systems are **not** secure

People have been trying to fix this problem for half a century



DTMach

Too complicated

DTOS

Verification is too hard ☹

FLASK

???

SELinux

Let’s just do it in Linux ☺

Enter SMOS

Current operating systems are **not** secure

=> seL4 **is provably** secure

How can we use seL4 to build a dynamic, general-purpose, provably secure operating system?

Microkernel design → better for security, but does the past continue to haunt us?

Verification **Performance**

Goals of SMOS

Dynamic Architecture

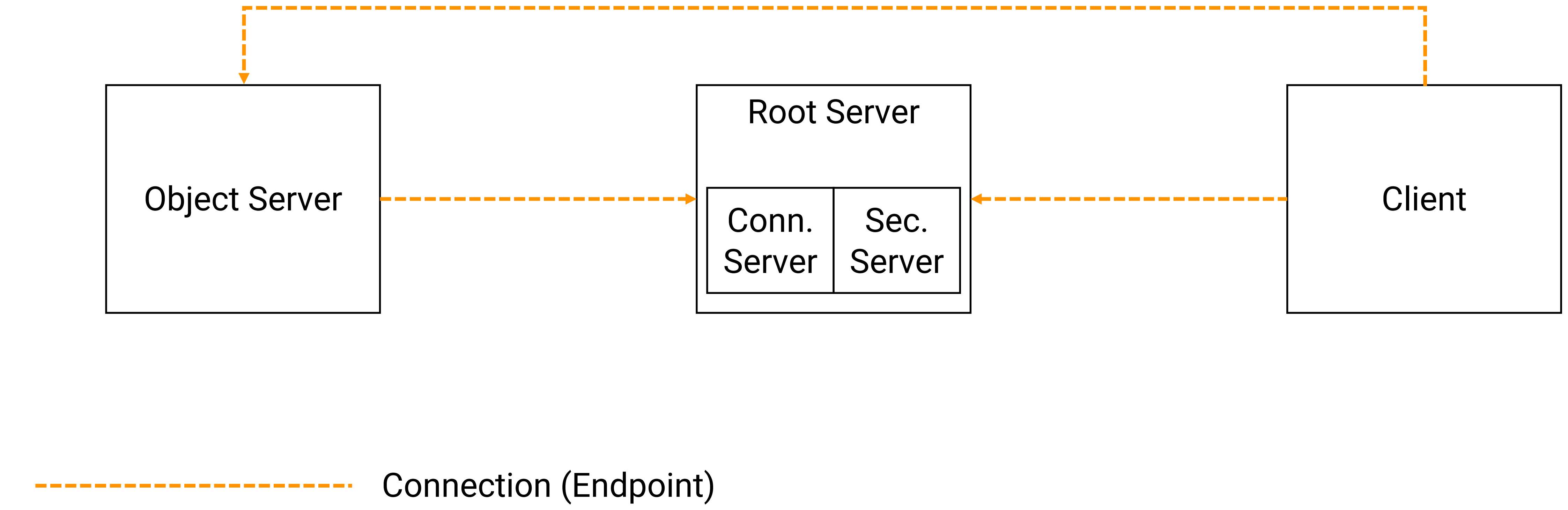
Flexible Policy

Verifiable Enforcement

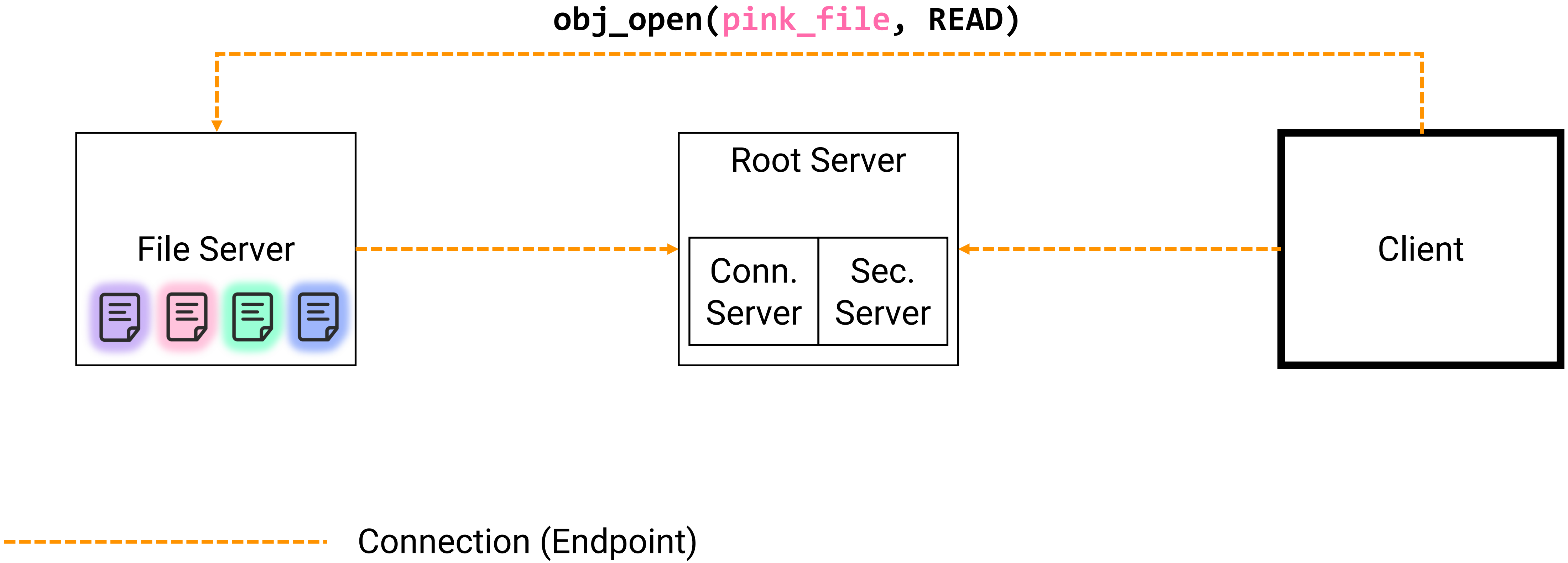
Uncompromising Performance

What does a SMOS system look like?

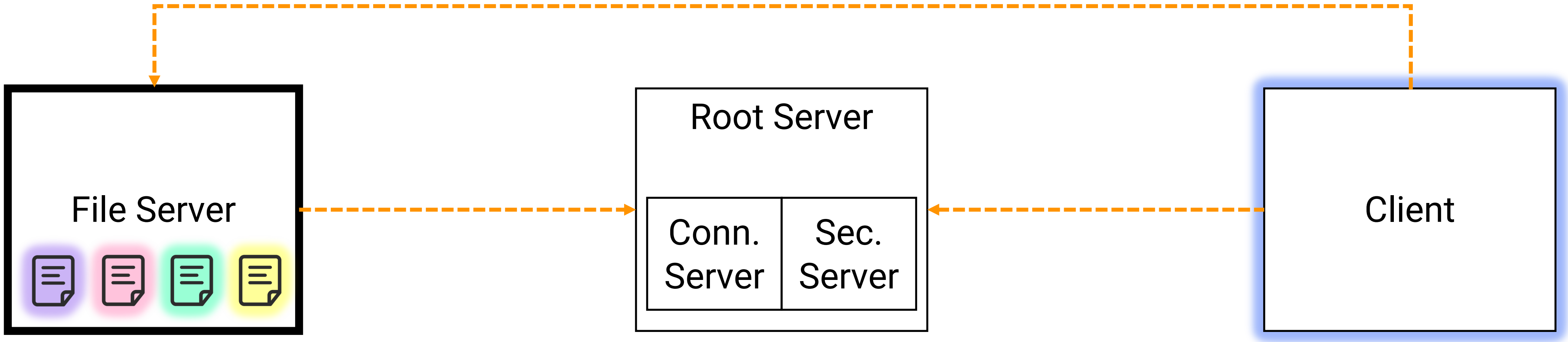
What does a SMOS system look like?



Policy enforcement mechanism



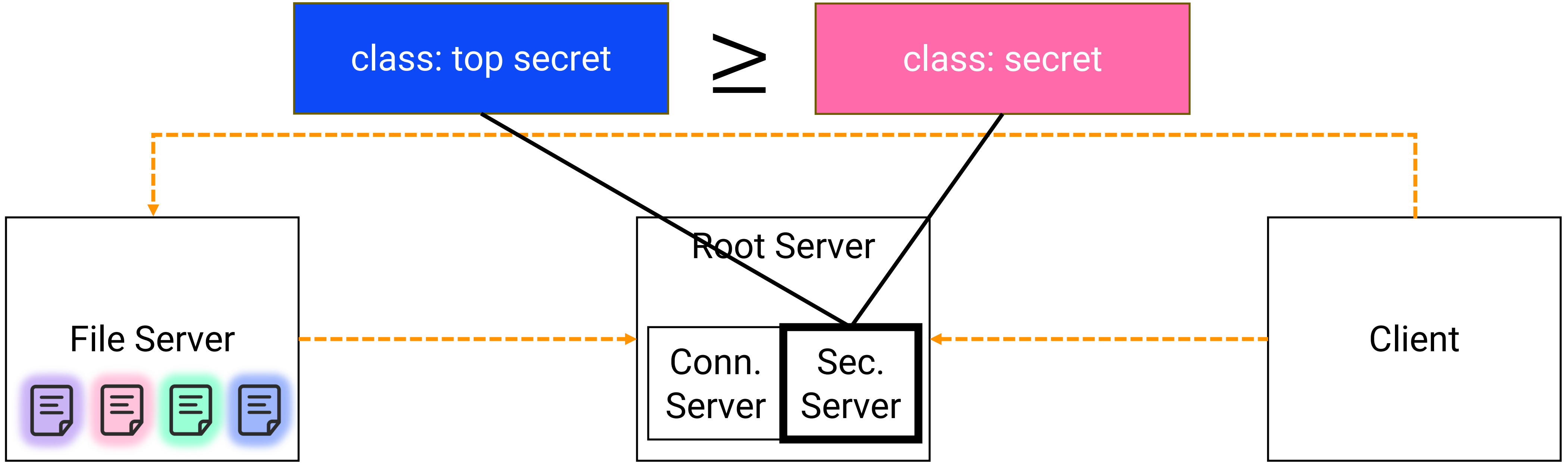
Policy enforcement mechanism



```
smos_auth(cli_sid, file_sid, obj_open, READ)
```

- - - - - - - - Connection (Endpoint)

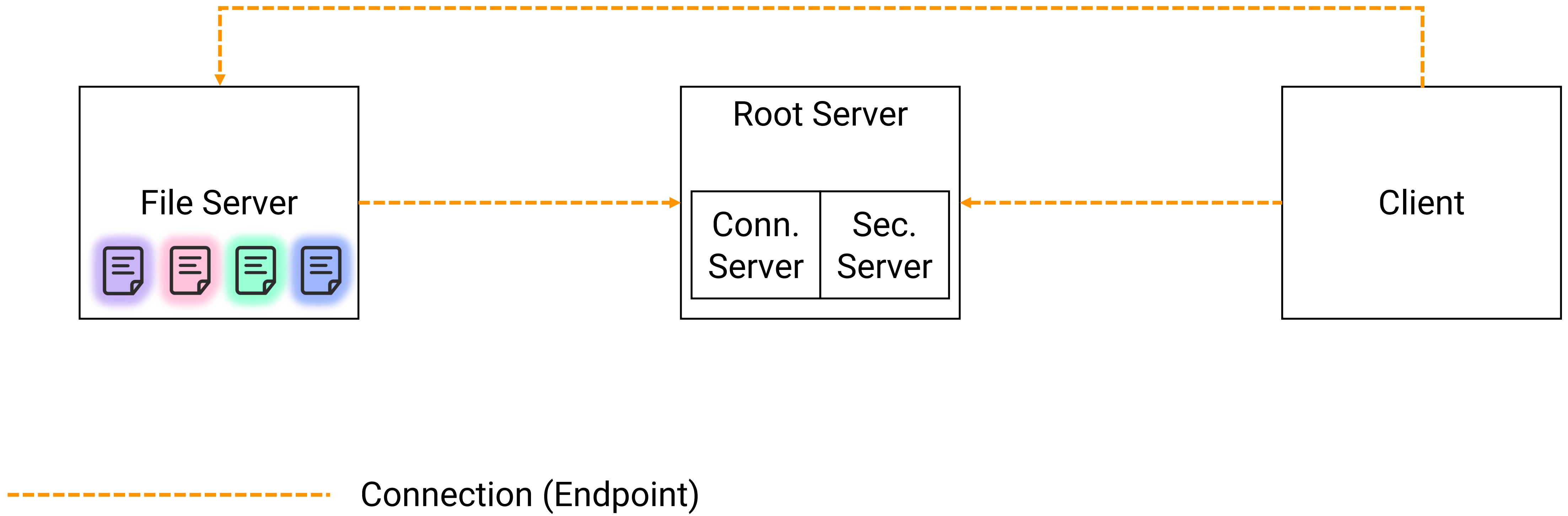
Policy enforcement mechanism



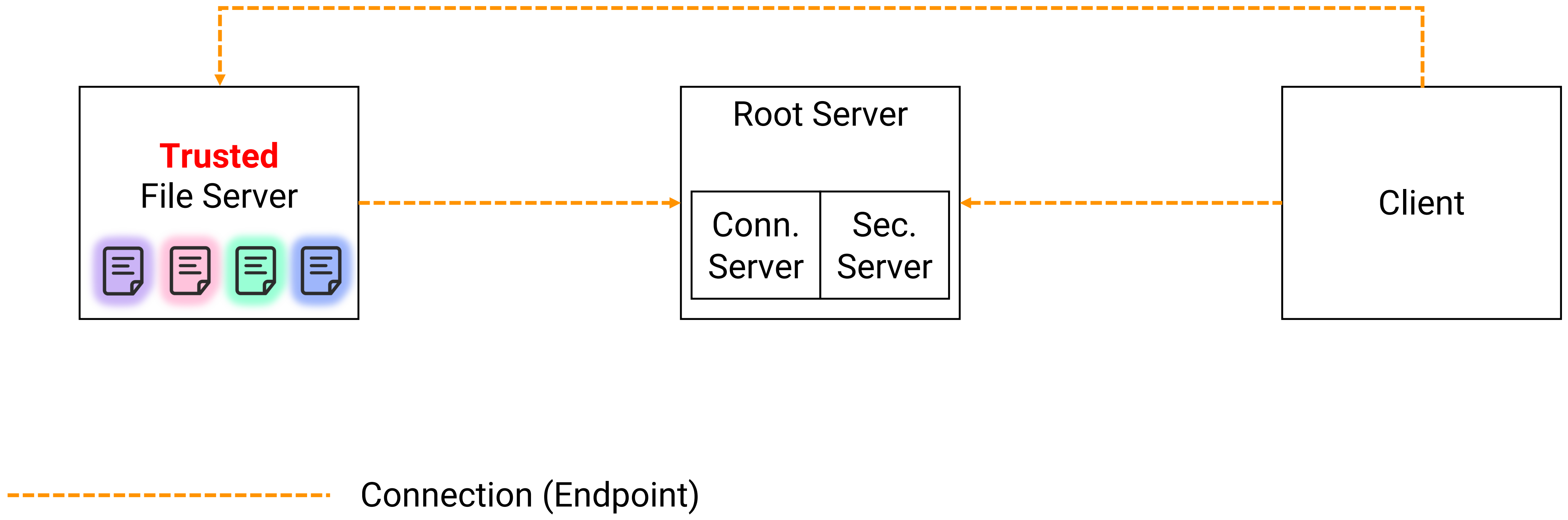
```
smos_auth(cli_sid, file_sid, opj_open, READ)
        <= permit/deny
```

Policy enforcement mechanism

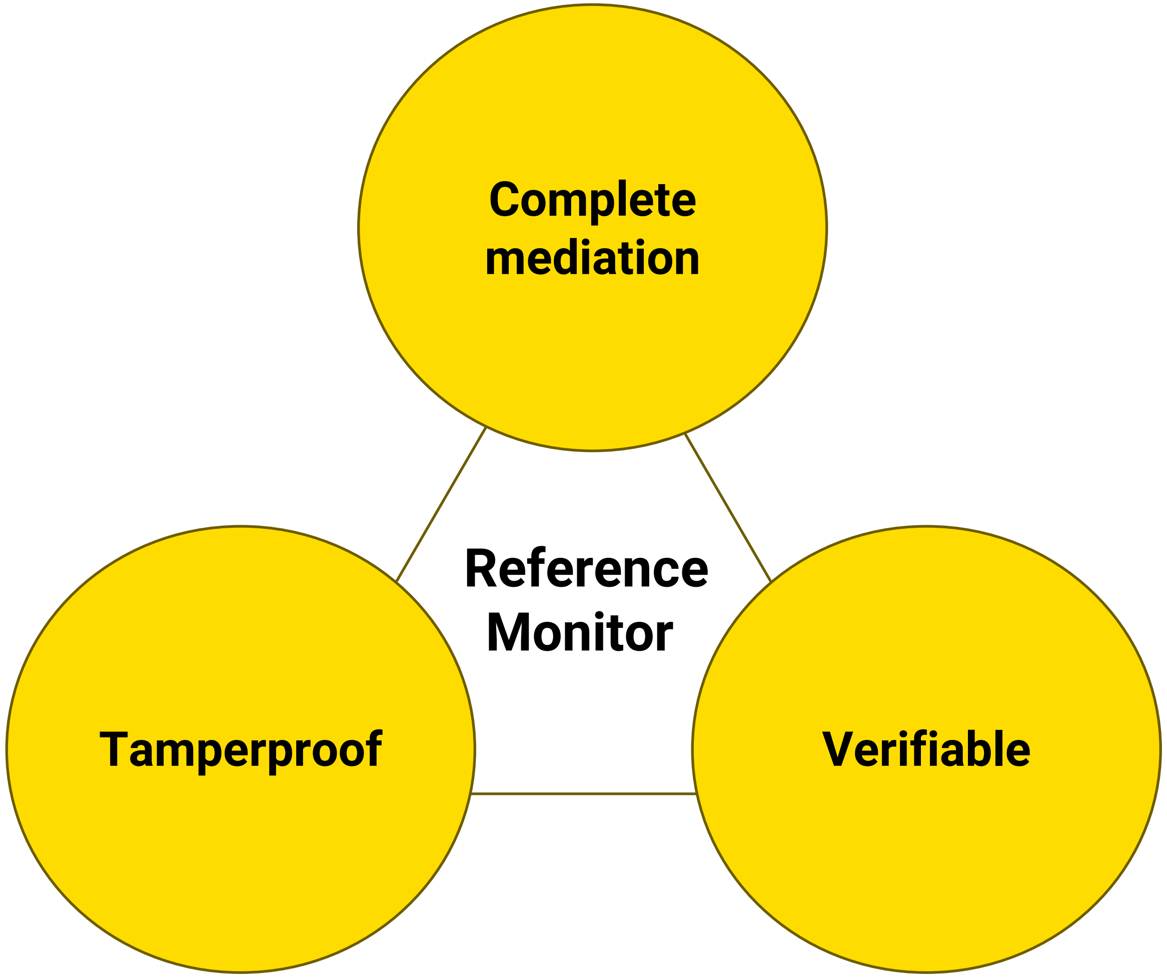
```
file_open(pink_file, READ)
     => success/deny
```



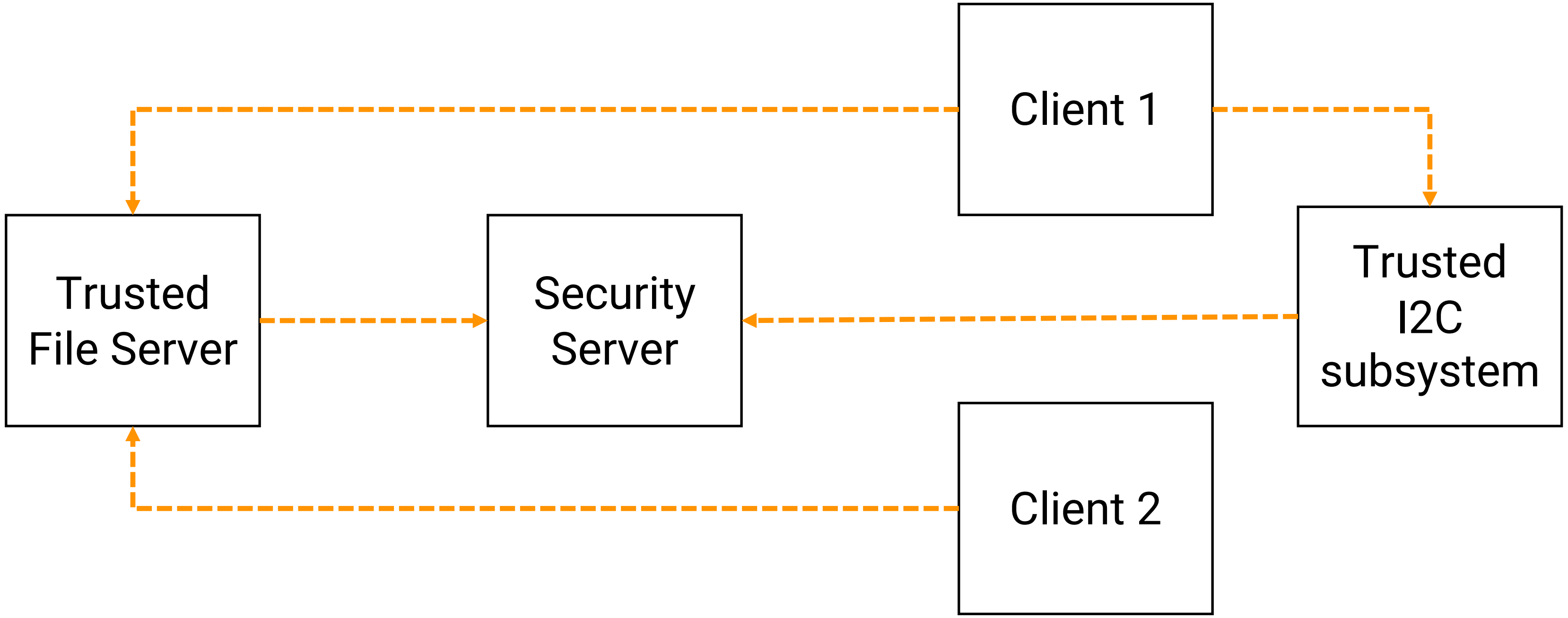
Policy enforcement mechanism



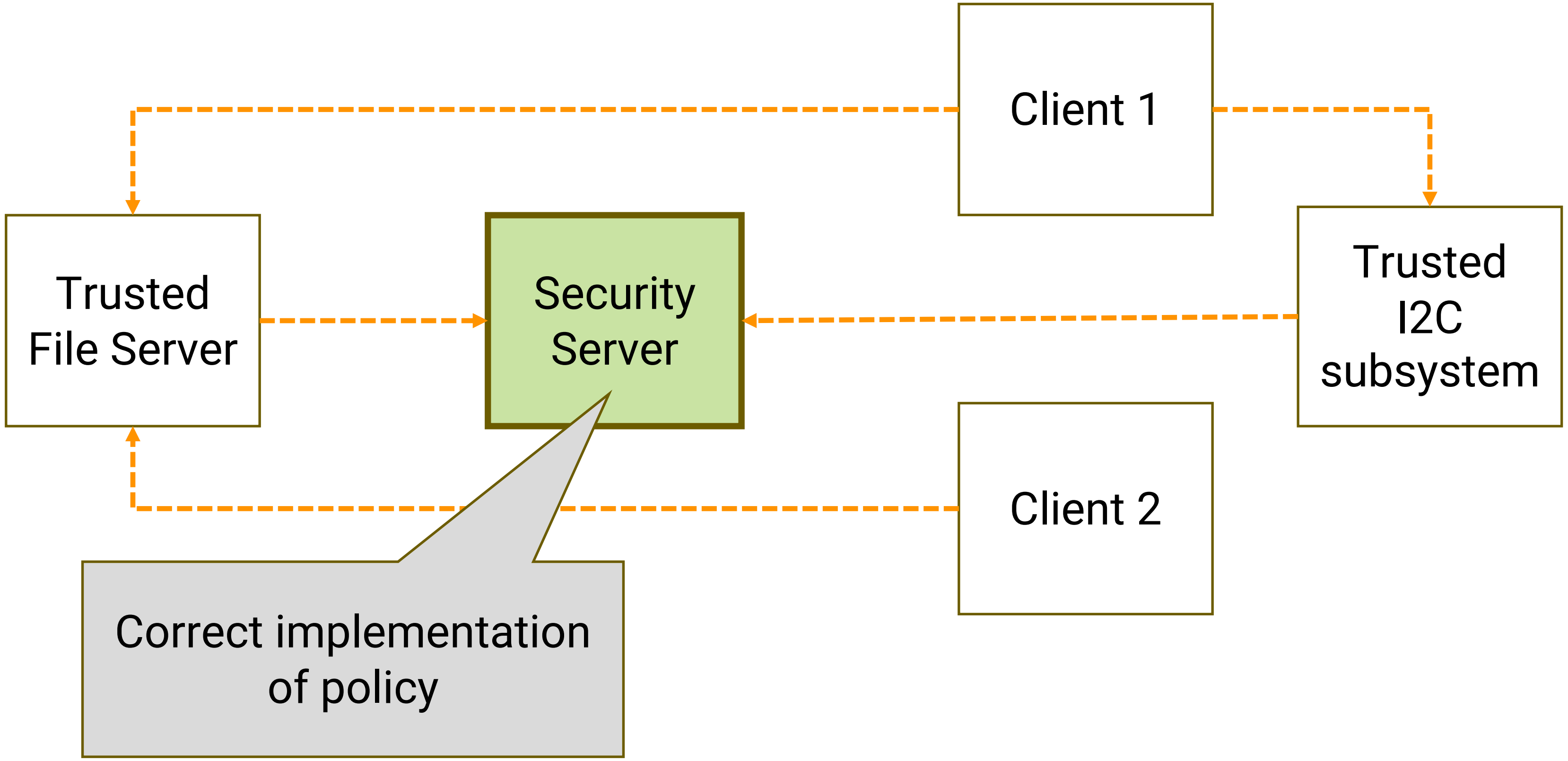
What makes an OS “secure”?



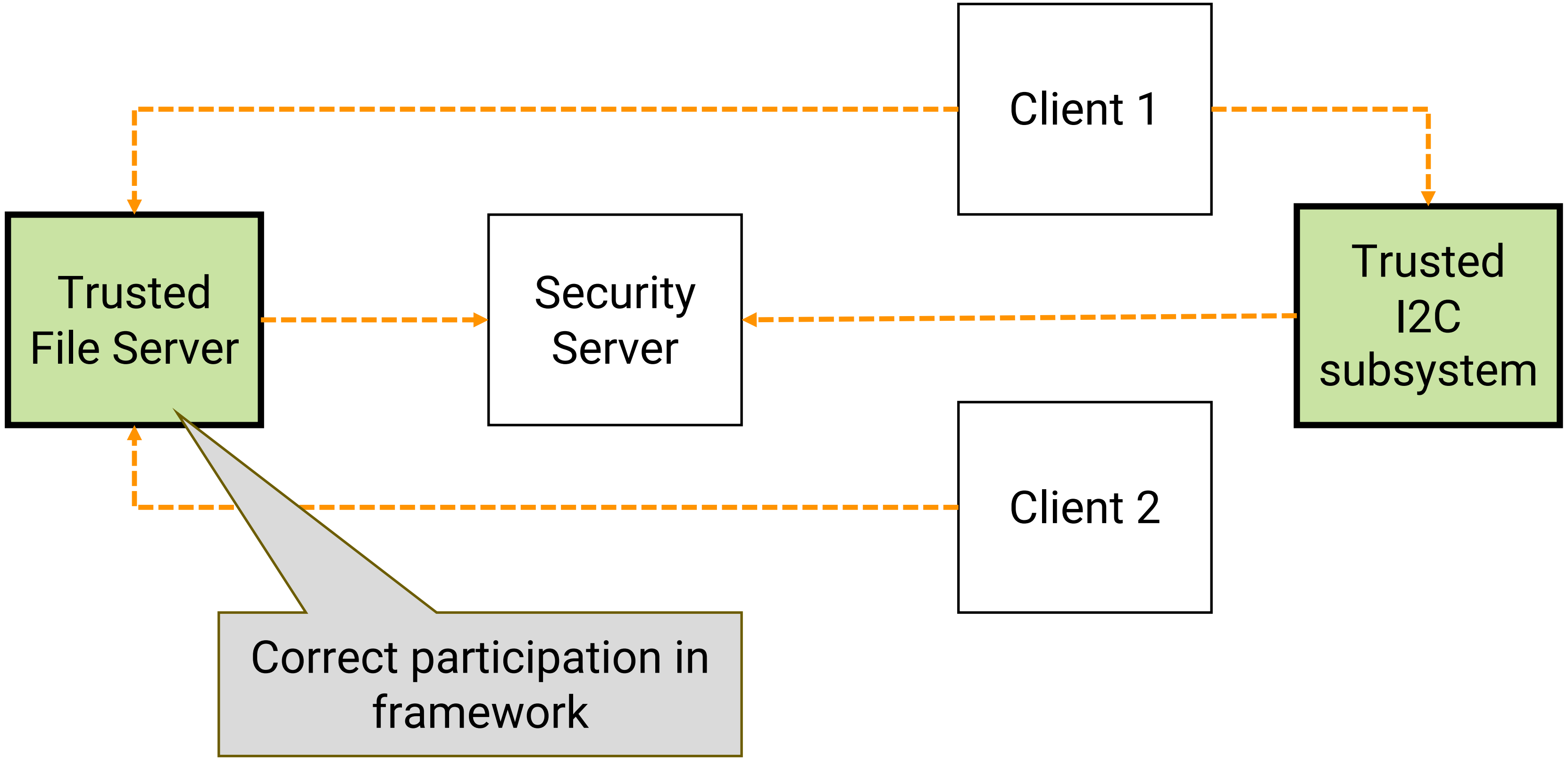
How can SMOS satisfy the reference monitor concept?



How can SMOS satisfy the reference monitor concept?

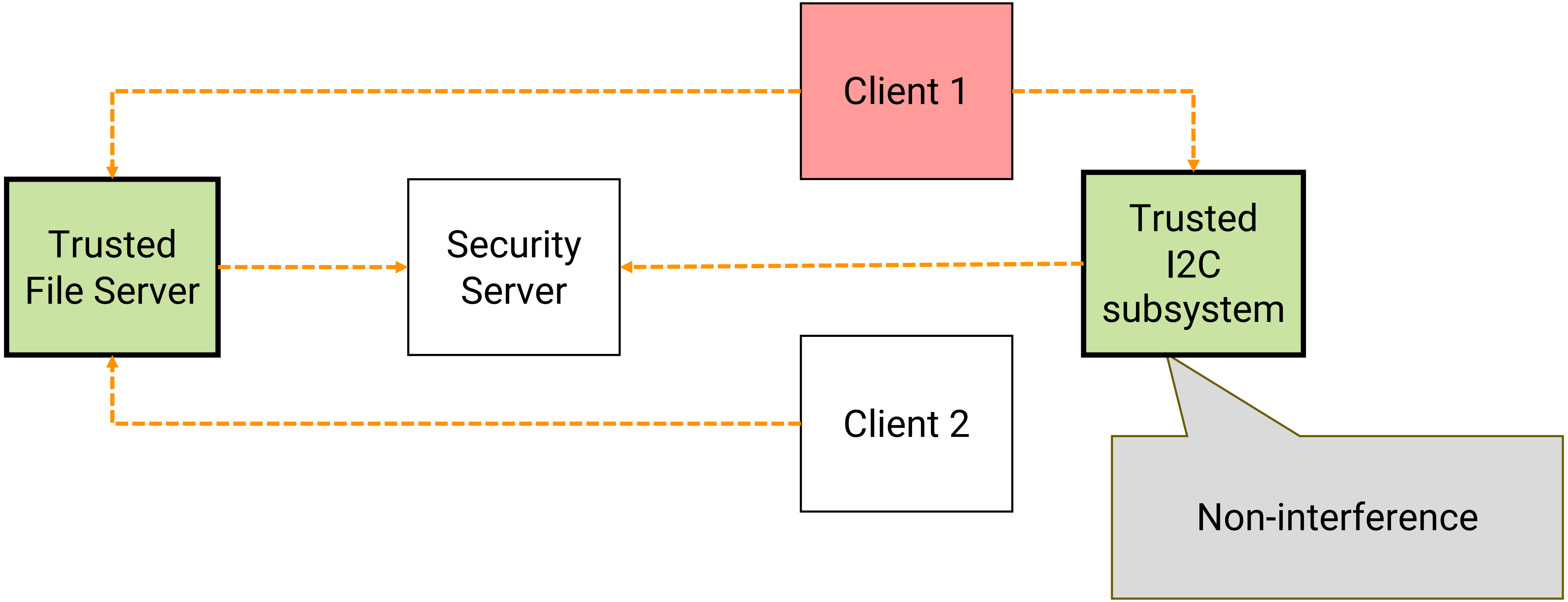


How can SMOS satisfy the reference monitor concept?



Correct participation in framework

How can SMOS satisfy the reference monitor concept?



Implementation progress

Engineering

Initial C prototype for exploring concepts/designs

Rewrite in Rust (using rust-seL4) – ongoing

Verification

Formal modelling in Lean4 of a general class of access control-based systems

Policies mandate sensitive information leakage is within certain acceptable bounds

Aim to connect SMOS instances to instances of the general class of AC systems

Next steps

Extend sDDF for dynamic systems

Verified interface generation

Implementation of non-trivial security policies

Thanks for listening!

Any questions/comments?

Image credits

Some images were taken from

rawpixel.com / Freepik

'Flaticon.com