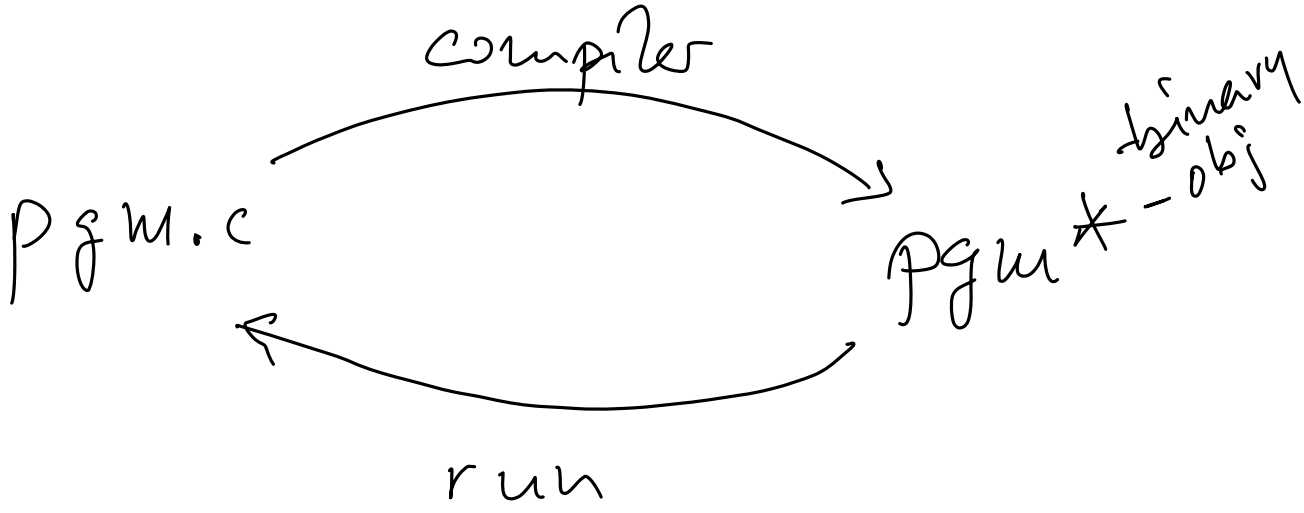
3) Principles of Computer Security

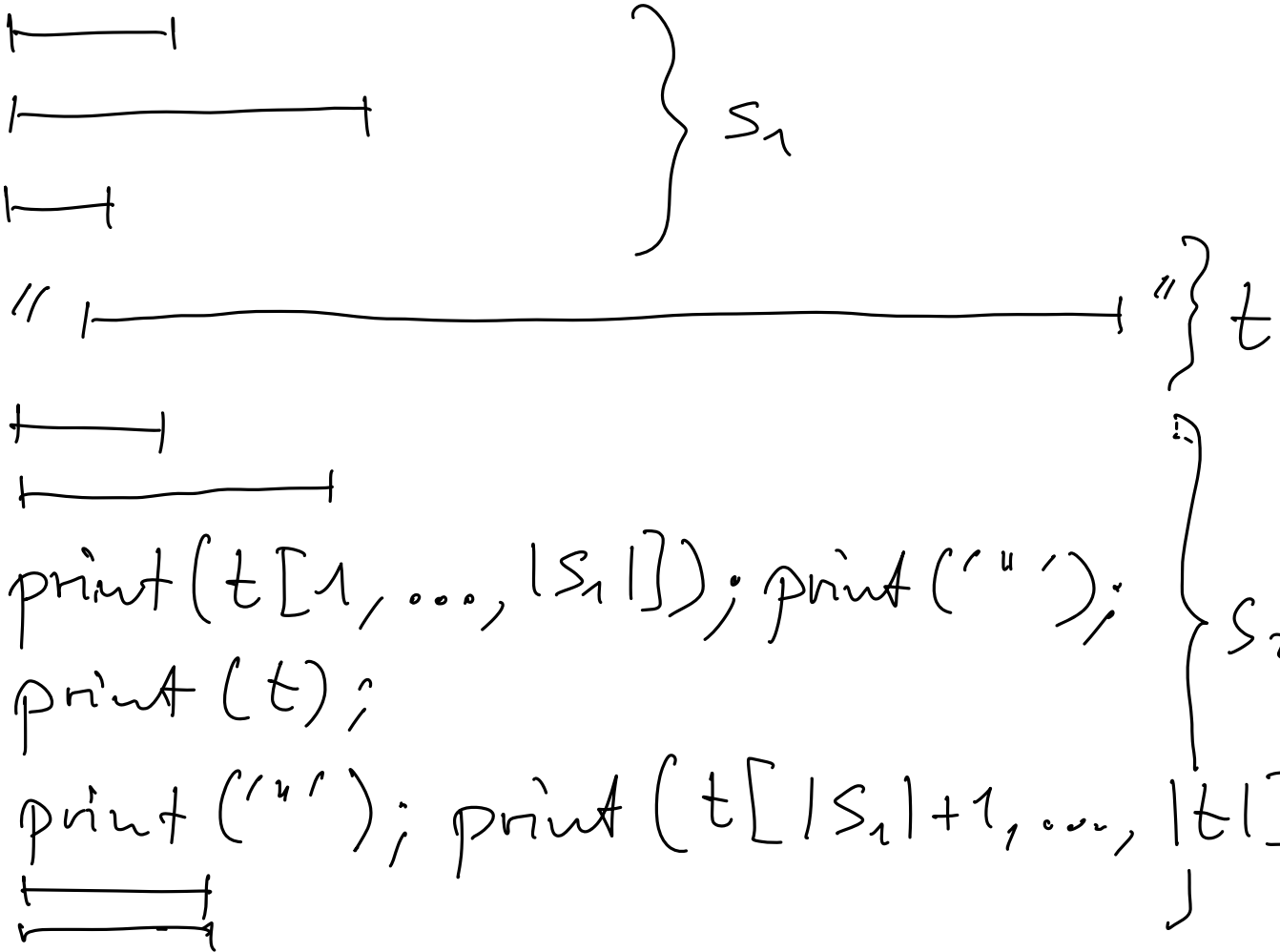
3.1) Trust and Trojan Horses

Q: How to write a program that prints a copy of its own source when it runs?

Q: How to write programs that replicate themselves?



pgm.c



using $s_1 \,//\, s_2 = t$

t string const

A program that prints its source is called a "quine."

Quines exist in all programming languages, due to Kleene's Recursion Theorem in computability theory.

Hide code in a compiler

Today one cannot produce (manufacture) new computing software or hardware without using existing ones.

Compiler (c): written in C

How was the first compiler written?

- by hand

a) Example of extending compiler

Characters:

'\n' ⟶ 0xA (10)
'\r' ⟶ 0xD (13)
...

Compiler:

```
if (c == '\n') return('\n');
if (c == 'r') return('\r');
```

b) Learn how to compile char. '\v':

Once insert line

```
if (c == '\v') return (char) 0xB;
```

From then on, source is simply

```
if (c == '\v') return ('\v');
```

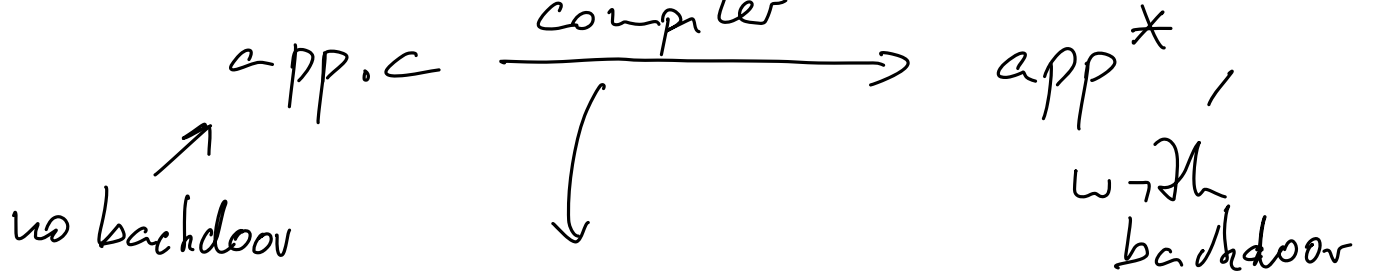
$\Rightarrow$ Compiler contains '\v' $\Rightarrow$ 0xB

Extending a compiler with a backdoor

Consider any app or pgm. distributed in binary form

Claim: Even if source is open, the binary compiled by someone untrusted may contain a backdoor.

Step 1:



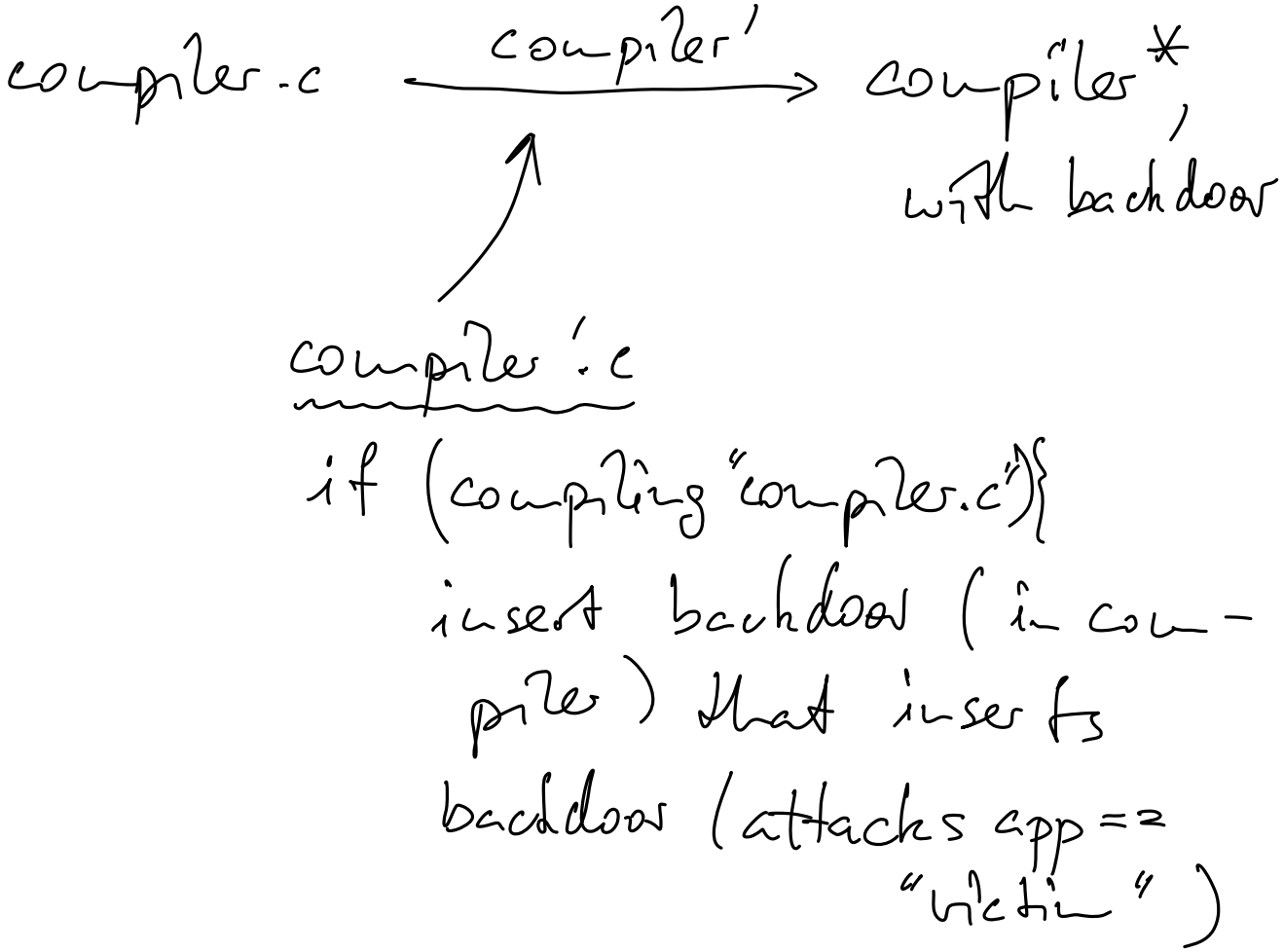
↓

compiler.c

```
if (app == "victim") {
    insert backdoor
        (code that leaks secrets)
}
```

... backdoor is visible

Step 2:



app.c —compiler*→ app with backdoor

... but app.c and compiler.c are clean!

Such backdoors could exist also in hardware.

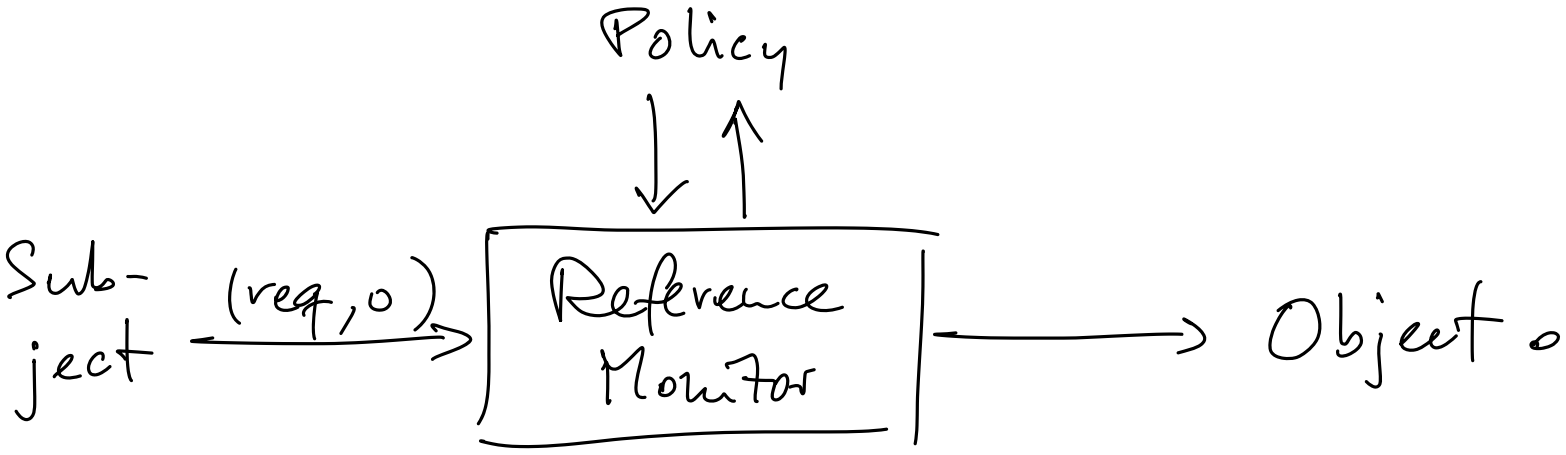
Self-replicating code has had a lot of success in the form of malware or viruses.

Defences are similar:

- start from clean systems
- physically secure storage
- verify authenticity of software
- Trust the process and the people behind your software

Ken Thompson: "Reflections on trusting trust", Turing Award, 1983.

Access Control

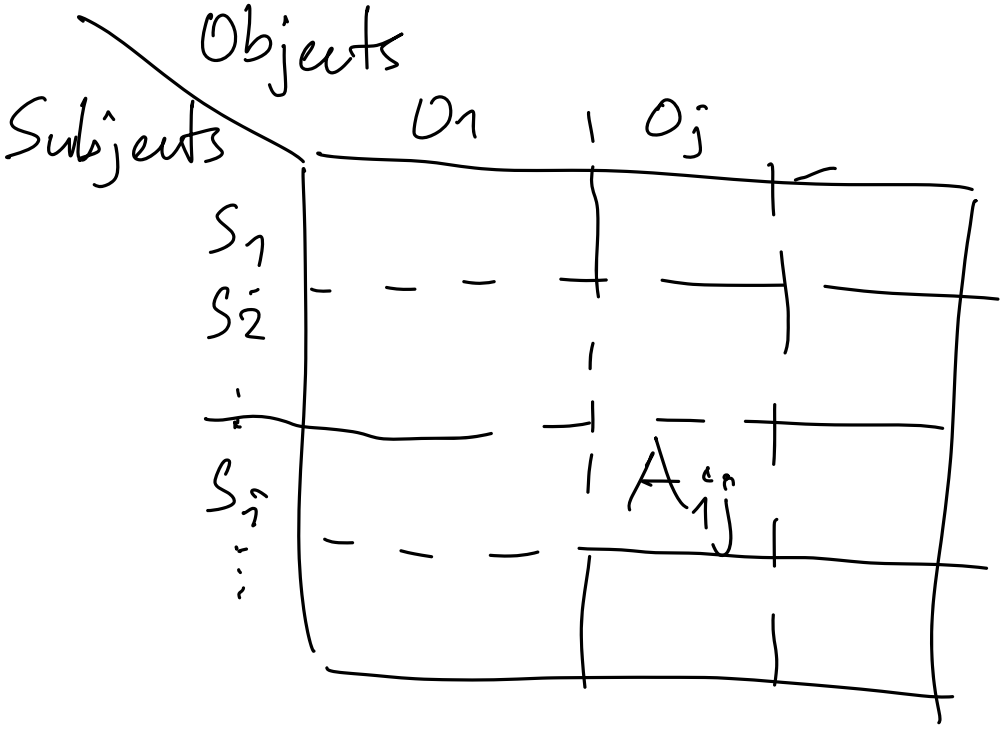


Reference monitor is logical concept to model if requests from subjects (principals) to objects (resources) are permitted.

Subjects are any active entity, typically a process ID, user ID, ...

Objects are all resources of the system

Policy defines for each pair of subject – object pair all permitted operations.



Access-control matrix

A_{ij} contain all permitted operations from S_i on O_j.

Typically, A is represented

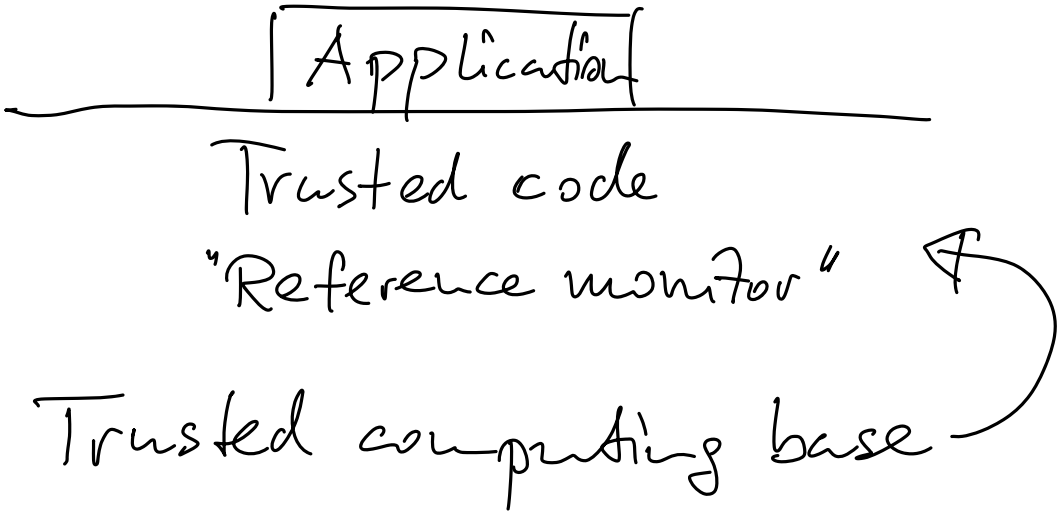
- by columns : list of access permissions by user for each object

  Access-control list (ACL)

- by row : list of capabilities by object for each user

  ↑ not often seen

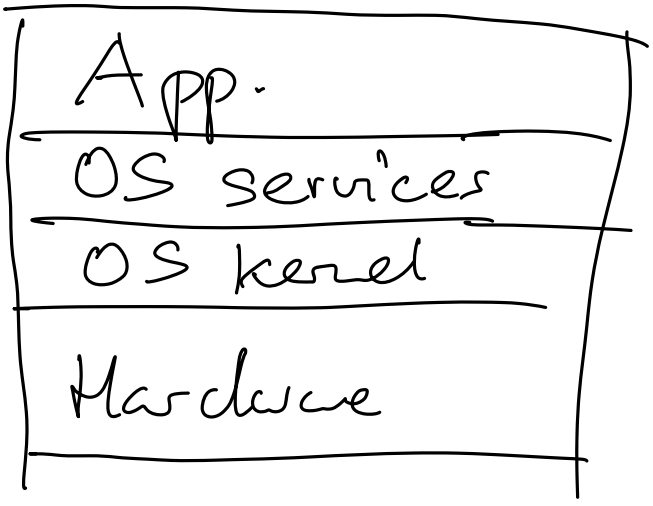
  ↖ must be protected



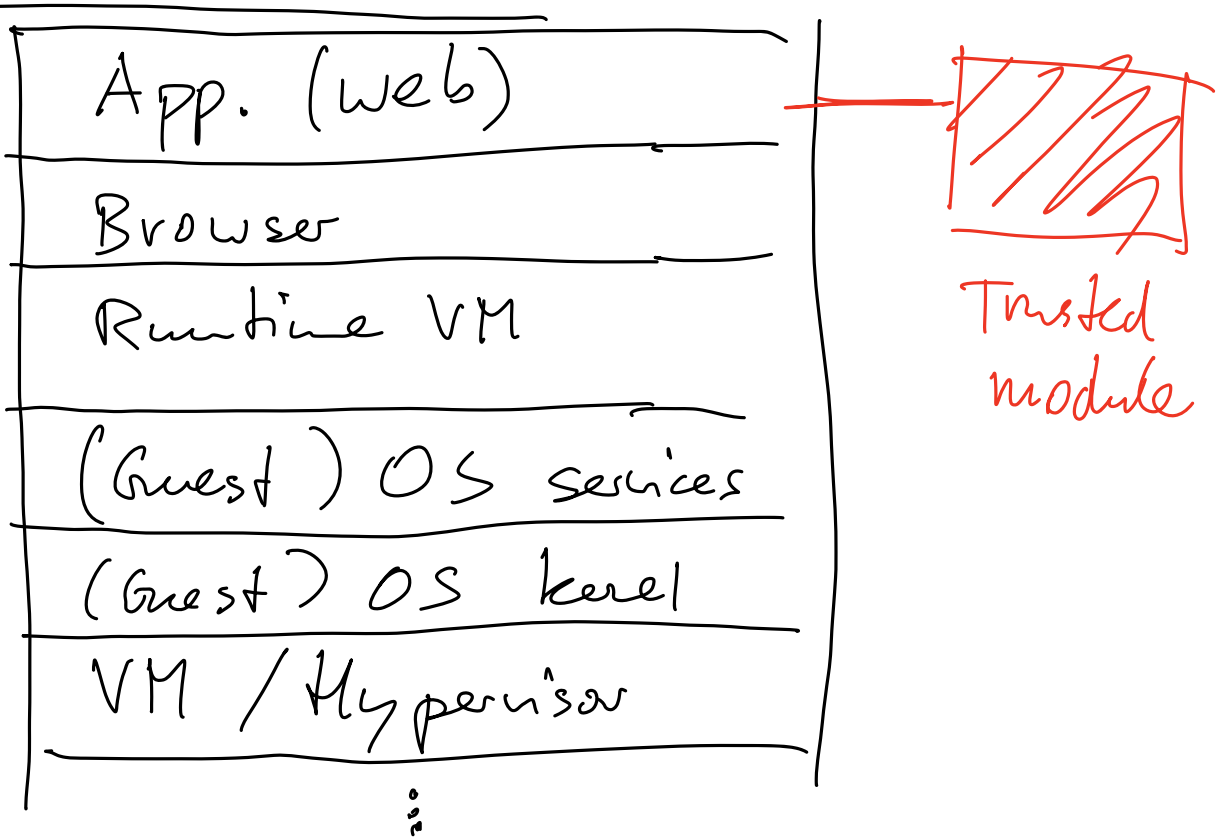
Layers

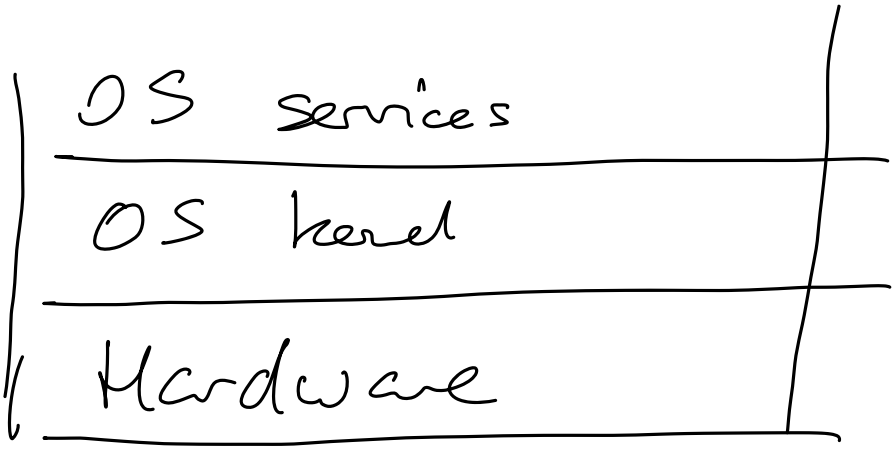
Computers are organized in many layers

- Textbook



- Real-word view





Trusted execution environments (TEE)

Smart cards

Hardware Security Modules (HSM)