Role Based Access Control

Permissions and roles

- A role is a set of permissions
- A permission is the ability to perform an action. Used also as synonym of right

- A group is collection of users that maintain their identity in the system

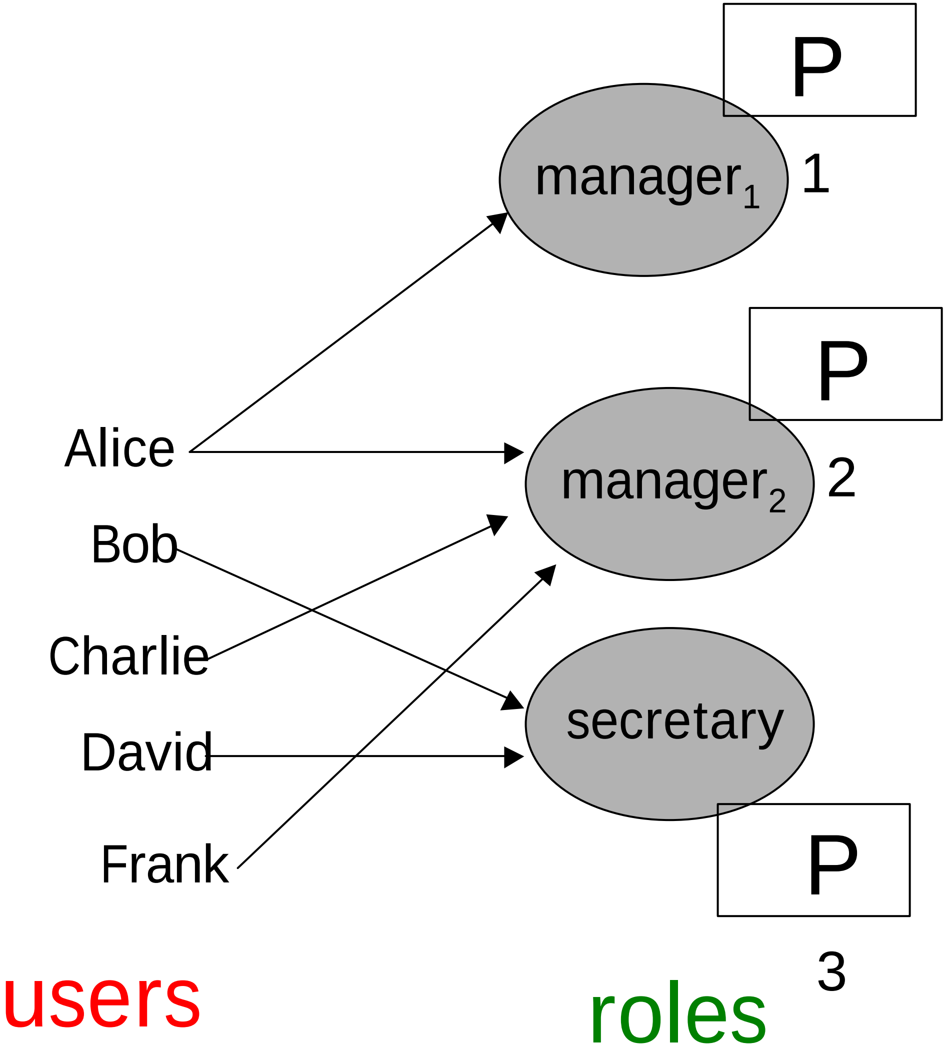
- User assigned to a role, assume role’s identity within the system

RBAC: Role Based Access Control

- RBAC model the authorization model in use in many commercial organizations

- A role maps a working activity/job function
  - ex. manager, secretary, sys adm., etc.

Roles Assignment



P_1

manager$_1$

P_2

manager$_2$

secretary

P_3

Alice

Bob

Charlie

David

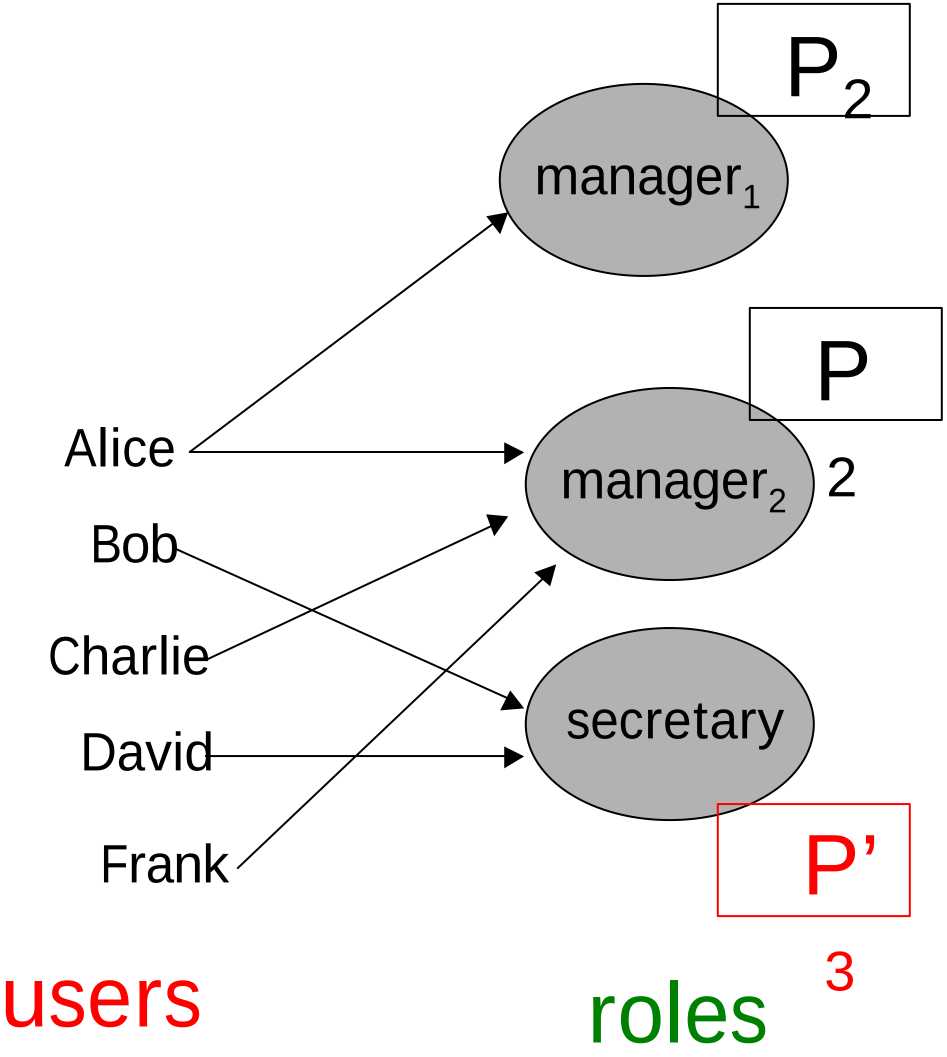
Frank

users

roles

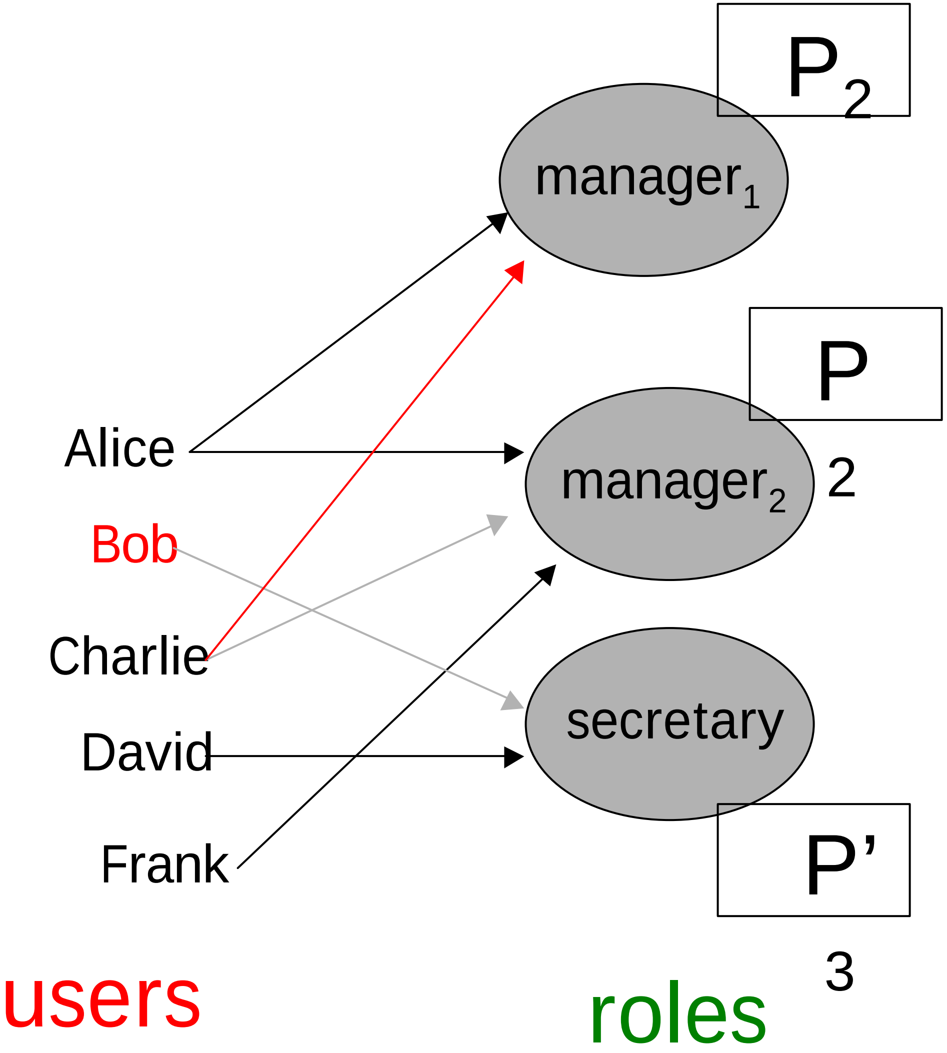
P_x = set of permission

Authorization Management



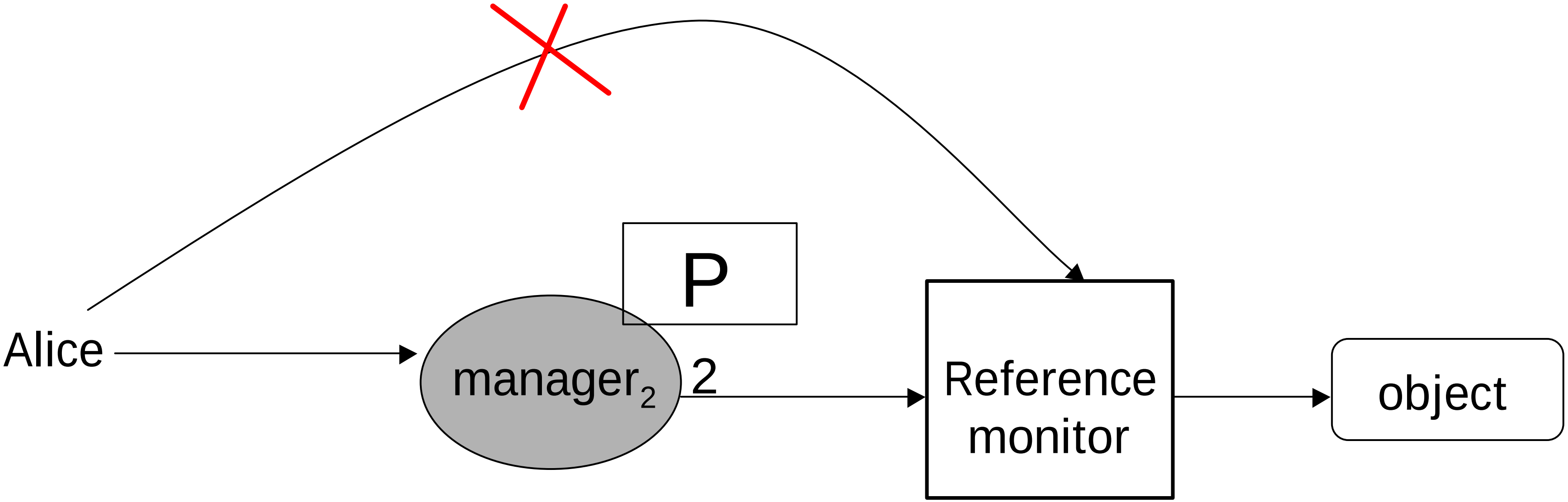
Change role definition instead of changing Bob and David's authorization

Authorization Management



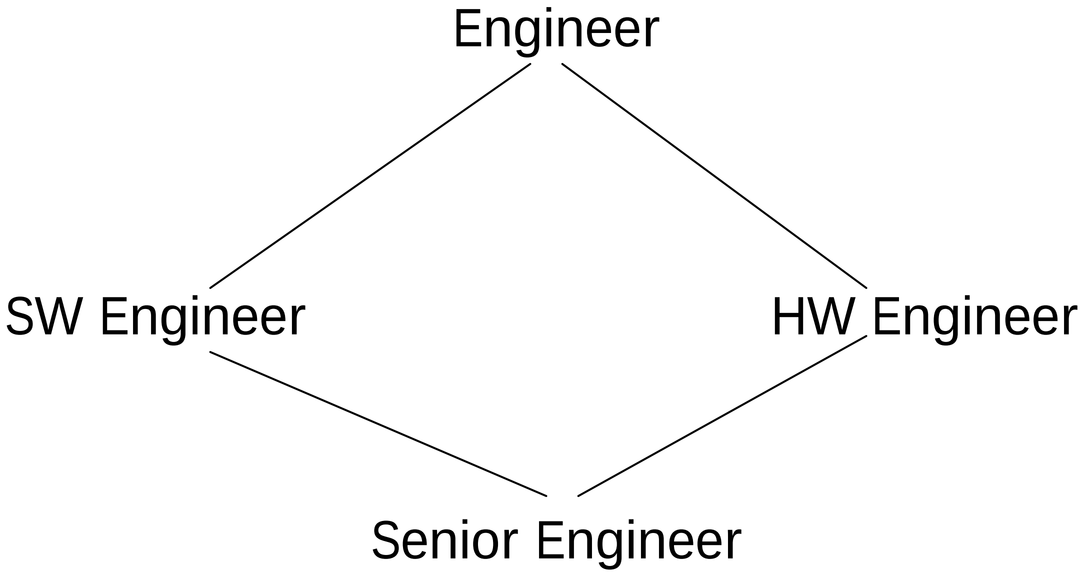
Easy to remove users or to assign new roles without rewriting permissions

Access requests



- Requests as role never as user

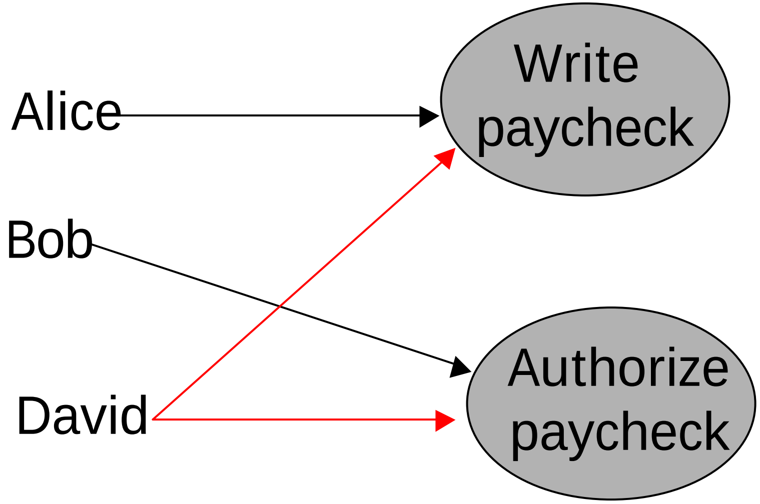
Hierarchical Roles



Inheritance of permissions

Senior Engineer ⊇ SW Engineer *and* HW Engineer

Separation of Duties



Static checks: at definition of roles

Dynamic checks when users are assigned to roles

Otherwise David could abuse the system by writing and authorizing his own checks

Object Classes

- Also object can be categorized

- Instead of enumerating the objects that can be accessed by a role is more convenient to specify classes of objects

  – Ex. Commercial letters, private letters, etc.

Administration of Authorizations

- Admin. Policies determine who is authorized to modify given permissions

- With Mandatory Access Control security labels determines permissions. Labels are assigned centrally by a security administrator

Alice.TS can write any object.S

Administration of DAC

- Centralized
  - i.e. security officer
- Hierarchical
  - i.e. org. chart
- Cooperative
  - separation of duties
- Ownership
  - Owner of the object grant/revoke rights to it
- Decentralized
  - Delegation and distribution

Administration of RBAC

- Similar in theory to DAC. In practice because of the critical task of defining roles, usually beforehand, only central, hierarchical and cooperative approaches are used

Intrusion Detection

Intrusion

- A successful attack. A set of actions that attempt to compromise the security of the system (confidentiality, integrity, availability)

- An intrusion is always referred to the access control policy of the system.

Intrusion

- Misuse: attack originated inside the organization

- An attacker is assumed to have succeed to masquerade as a legitimate user to the system

Intrusion responses

- Prevention
  - Authentication, firewall, etc.
- Detection
  - Second line of defense. Intrusion detection systems
- Tolerance
  - Ability of the system to provide services also under attack

Auditing

- Crucial to intrusion detection is the ability to record the activity of the system in audit files.
  - Host/Application based data collection
    - Sys op. log functions, apps specific, etc.
  - Network based data collection
    - Network sensors to collect local network traffic

Detection

- All audits are then cross analyzed and processed by automatic tools in order to detect:
  - Intrusion based on unusual behavior of users anomaly detection model
  - Pattern of commands that has been recognized as an attack misuse detection model

Detection

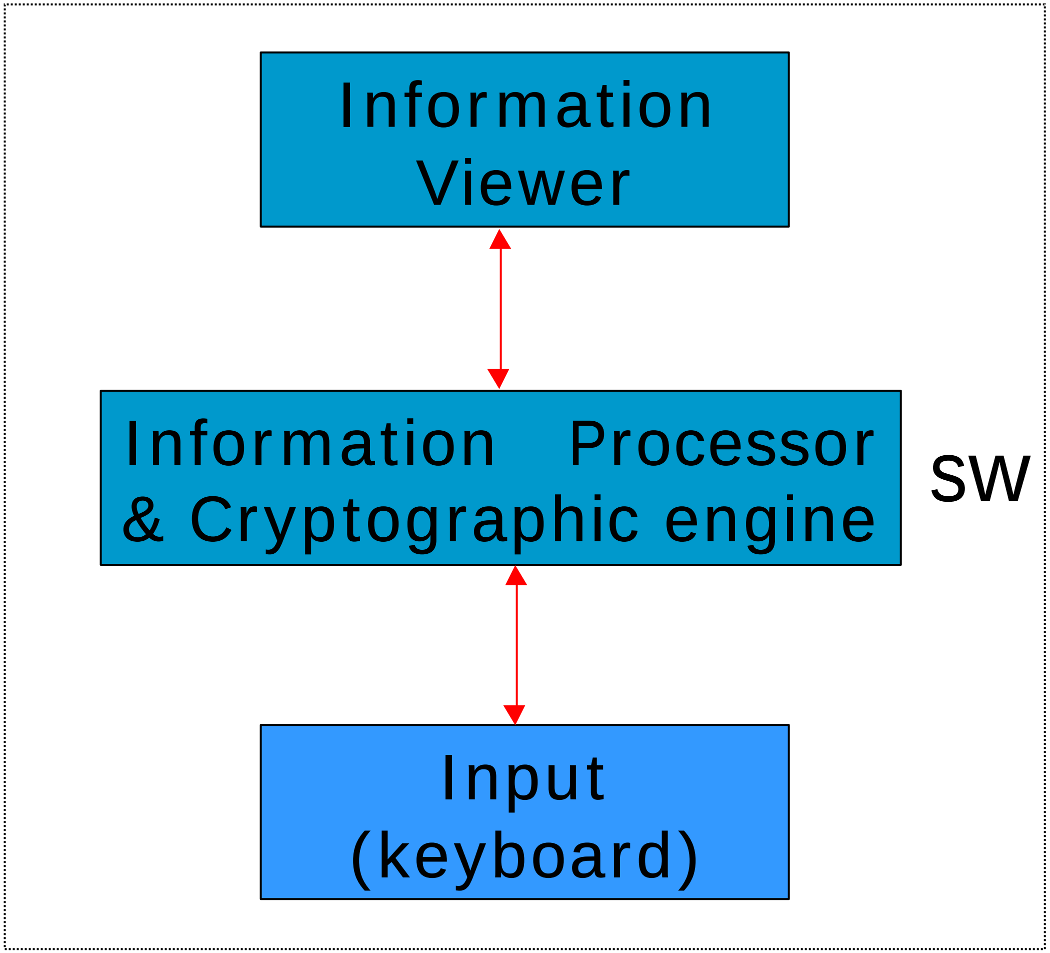
- Anomaly based detection
  - Based on “legitimate” behavior of authorized user
  - Difficult to characterize legitimate behavior
  - Annoyance of false negatives
- Misused detection
  - Signature of the attack. Similar to anti-virus
  - *A posteriori* defense

Trusted UI

Trusted User Interface

- Trusted User Interface: an interface that user can trust or at least he can detect if it has been tampered with.
- Today we are far from that

The Local System



Information Viewer

Information Processor & Cryptographic engine

SW

Input (keyboard)

All components and channels need to be trusted

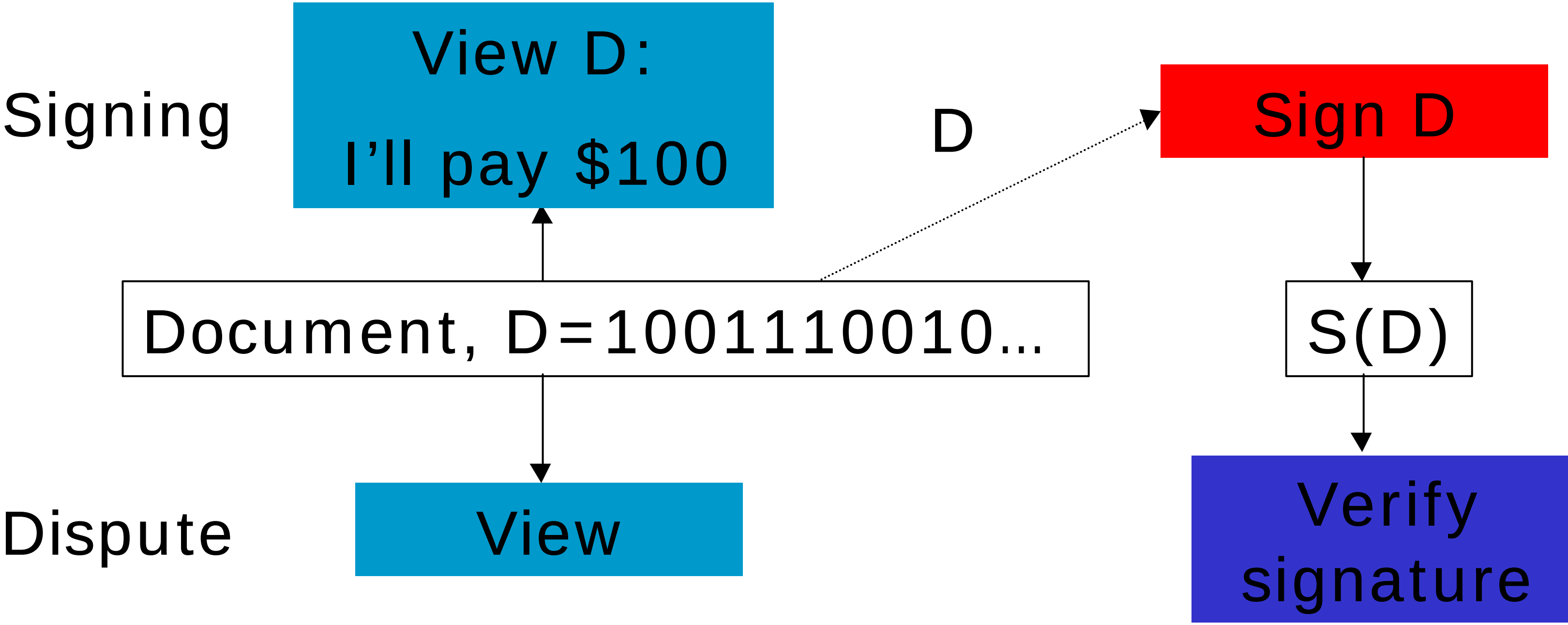
Problems

- What user type is the input of the processor?
- What user see is what is really processed?
- Does the cryptographic engine behaves correctly?

Existing legislation

- DigSig, privacy and e-commerce European laws require:
  - WYSWYS (What You Sign is What You See) principle
  - Explicit consent and authorization

Signing and Verification



Signing

View D:

I'll pay $100

D

Sign D

Document, D=1001110010...

S(D)

Dispute

View

Verify signature

Signing Process

- Representation
  - Handwritten signature is attached to **unique** representation of document (printed)
  - Digital signature:
    - A particular digital version of the document is signed
    - document shown in viewer
- Secure Signing
  - Handwritten signature: Signer is in control of the pen
  - Digital signature: Crypto engine signs arbitrary bits

Still outside TUI

Insecure local system

- operating system
- hostile applications
  - virus
  - downloaded application (unknown consequences)

Problem with security application

- errors in application
- user errors (bad user interface)
- limits of cryptographic engine (CSP)
- access control to keys (weak link in the chain)

Representation

- Word
  - binary representation
  - printed version may change (e.g. automatically update of dates)
  - macros

- HTML (XML)
  - links
  - images (e.g. hiding text)
  - java/javascript

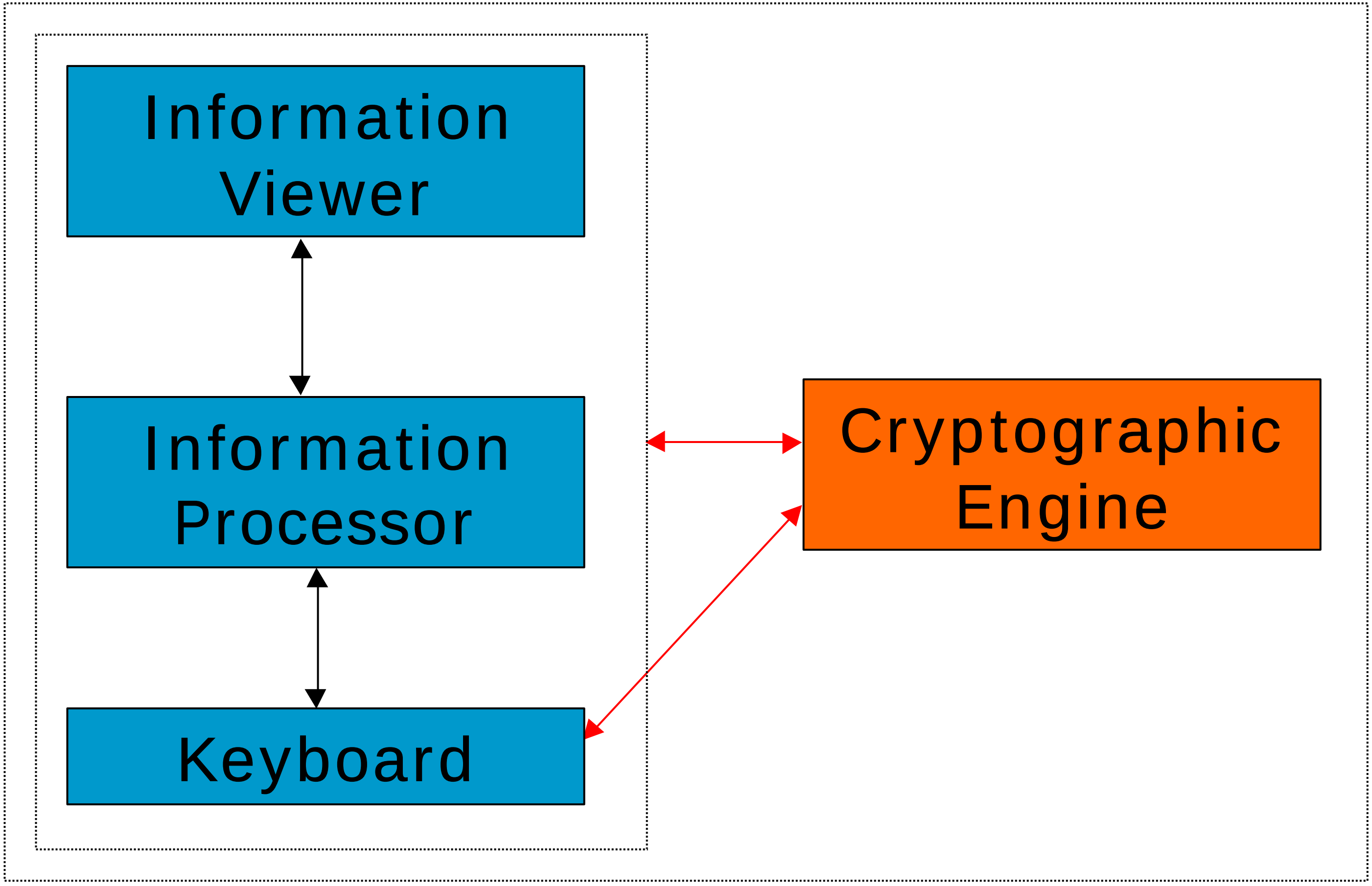
Representation

- Signed part of document must remain unchanged
- One application may shown the same bits differently depending on time, locality, scripts,...
- Links: Inclusion of external information (files, images, ...)

Secure Signing

- Tamper resistant device for
  - storage of/operation with key
  - signing correct data
- Only sign when authorized
  - biometrics
  - PIN
- Viewer/application must be trusted to
  - show/maintain unique representation
  - send right document to crypto engine (secure channel)

State of the Art



Information Viewer

Information Processor

Keyboard

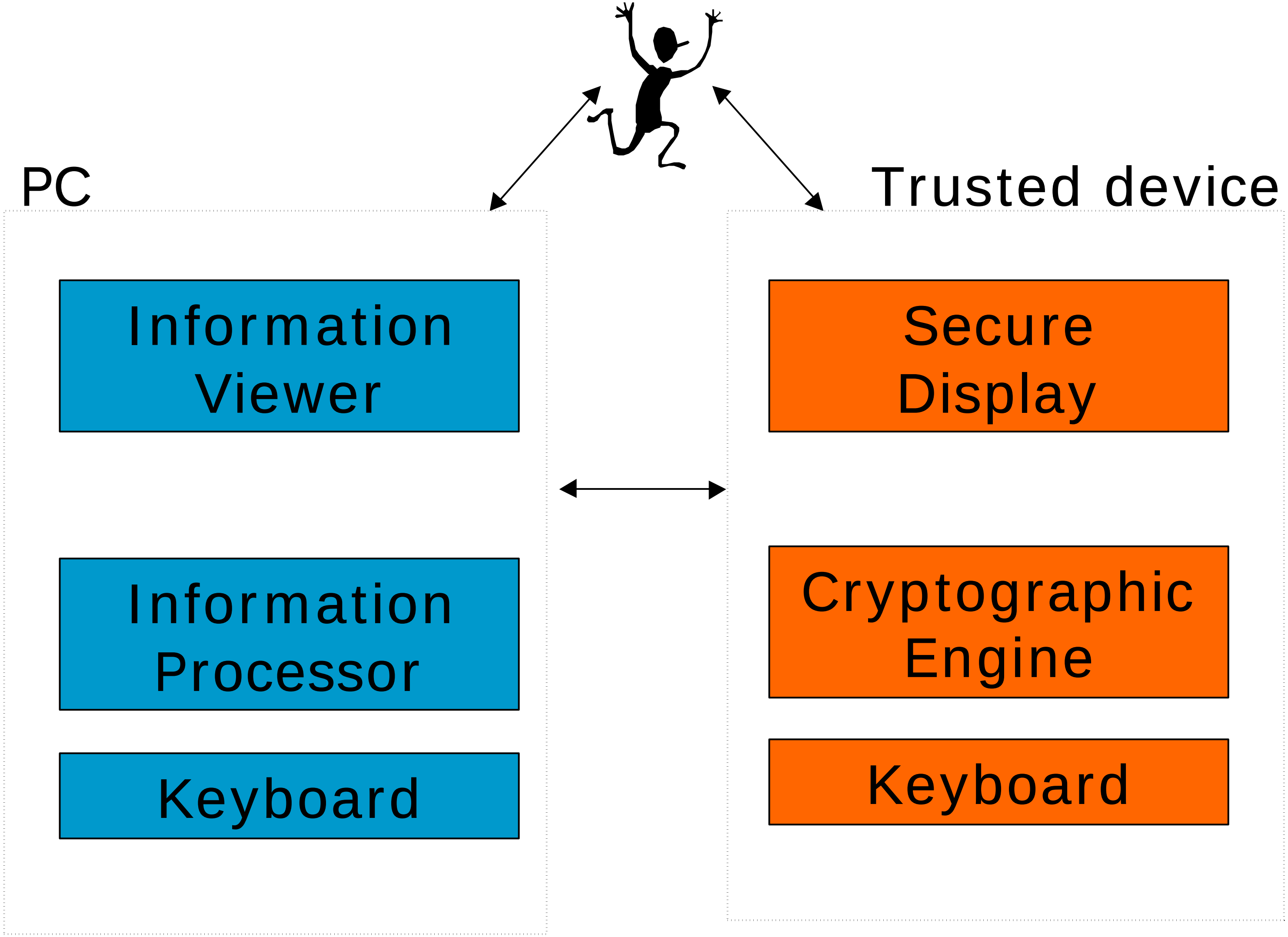
Cryptographic Engine

Ex. Smart-card

Requirements

- Unique representation of documents
- No hidden parts of the document
- Tamper-resistant crypto engine (secure signing)
- Secure channels between
  - viewer and crypto engine (WYSWYS)
  - user and crypto engine
- User must be able to recognise viewer
  - personal configuration (securely)
  - same window for all security applications
- User friendly design

Trusted Devices



PC

Information Viewer

Information Processor

Keyboard

Trusted device

Secure Display

Cryptographic Engine

Keyboard

Representation

Too expensive with PC but things are a bit different with mobile phones or PDAs

- Condensed version must be complete, so..

References (links) to registered information in condensed version.

Example

Offer/order to repair house

- List of things to be done
  - change window
  - make new wall
- Time period
- Date
- Price
- Reference to standard conditions, materials
- Detailed information about conditions and materials

Conclusion

Local system for secure dig. signatures

- Requirements
  - Representation of document (unique, complete)
  - Secure, personal display (split representation)
  - Secure crypto engine
  - Secure channels between these and user
  - User friendly (UI, secure device)

- Usual smart cards + reader insufficient
- Smart card + reader + secure keyboard is still not enough
- Usual smart cards + reader + keyboard + display