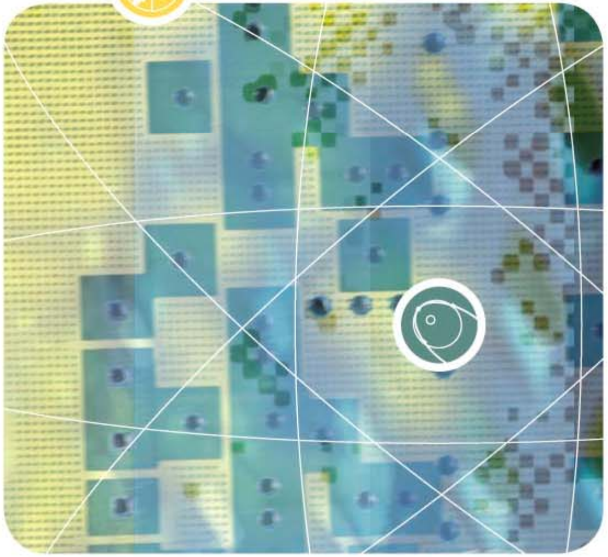
FFI Forsvarets forskningsinstitutt

Security model for resource availability – Subject and object type enforcement

Ole-Erik Hedenstad

Norwegian Defence
Research Establishment



Overview

Security model for resource availability – Subject and object type enforcement (SOTE)

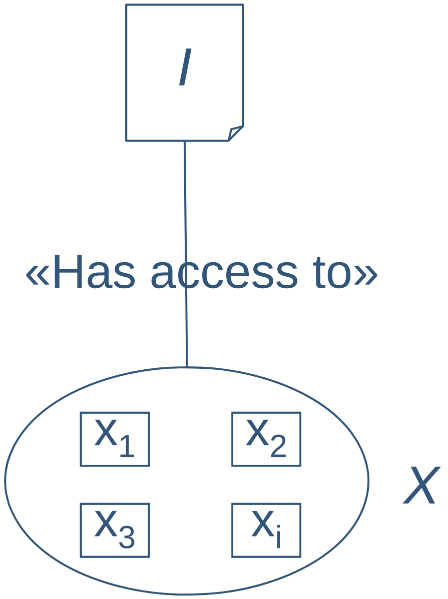
- What resource availability is
  - and some other terms
- ”Subject and object type enforcement”
  - why
  - proposed new model
- Composite policy for cross-domain information flow
- Related work

A *security model* is a model that represents a particular policy or set of policies. (Bishop)

Availability

Let *X* be a set of entities and let *I* be a resource. Then *I* has the property of *availability* with respect to *X* if all members of *X* can access *I*. (Bishop)

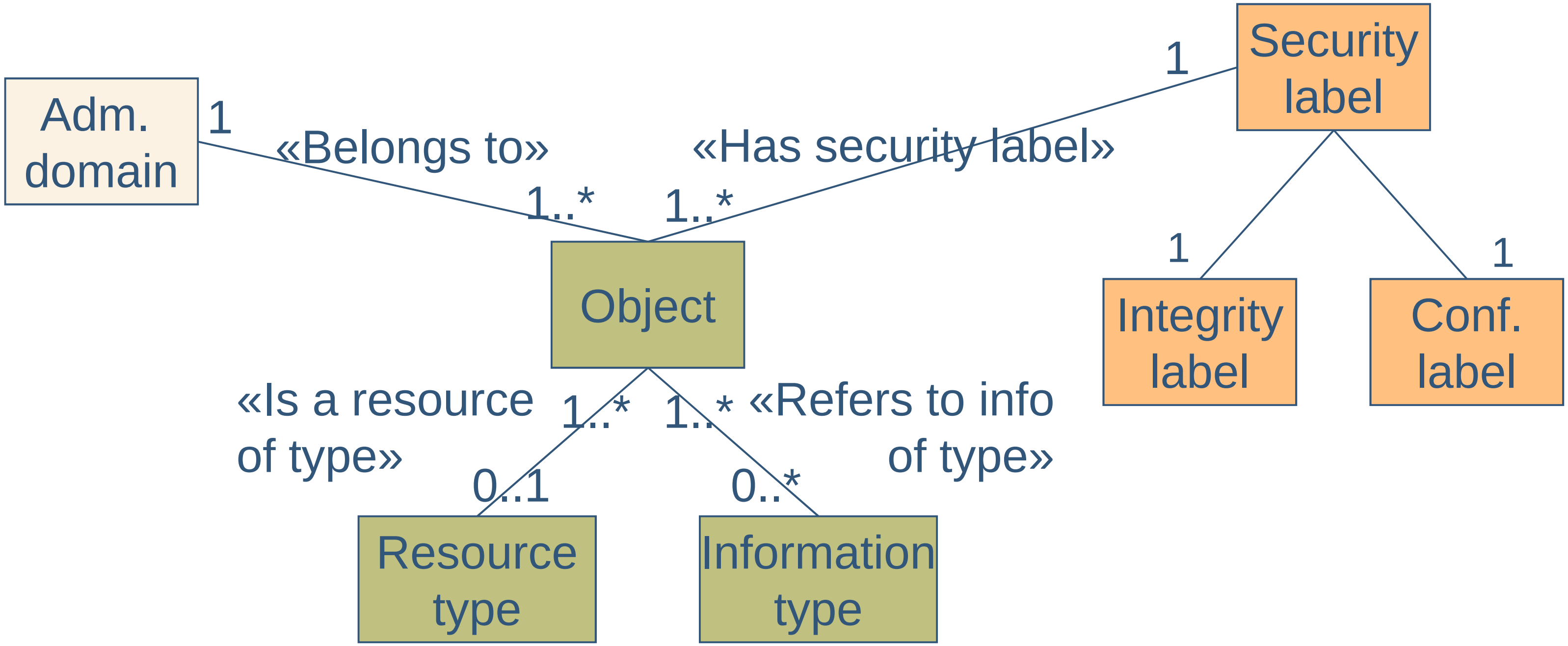
Resource availability regulates the access to resources in order to get timely, reliable and secure access to services and data



- Availability is associated with requirements on throughput, redundancy, backups etc.
- We also include restrictions and conditions resources must fulfil in order to be available
- We make a distinction between *information* and *resource* availability

Administrative domains

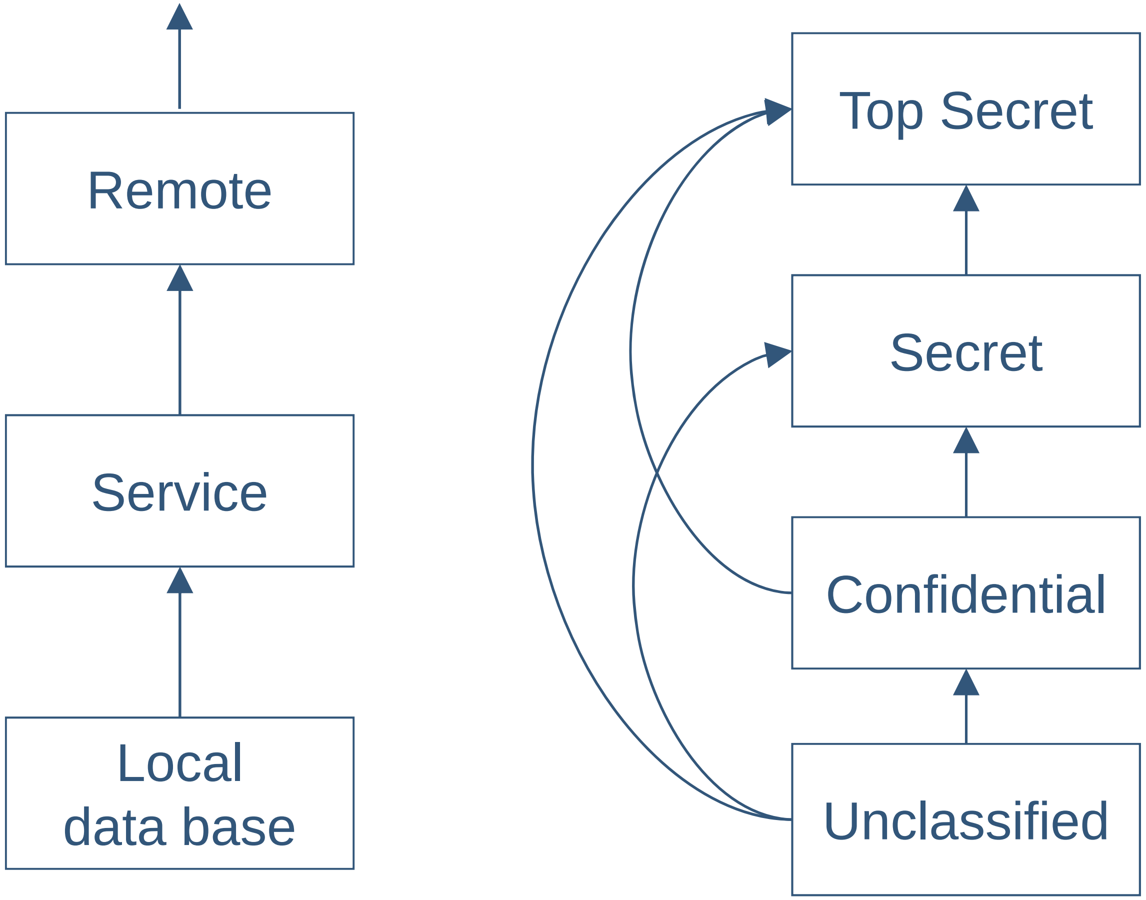
An *administrative domain* is a collection of computer systems to which applies the same set of security policies and security levels, executed by a single authority.



Rationale for SOTE

- Basic idea: to define the permitted information flows between resources of different types, typically between types of program components.
- Heterogeneous environments. The administrative domains do not implement the same set of security policies and security levels.
- The domains have requirements to control and confine the interaction with resources of the other domains:
  - express fine-grained restrictions on information flow, supports the principle of least privilege
  - express conditions a resource must fulfill
  - express intransitive (indirect) information flows

Types of information flow



Intransitive

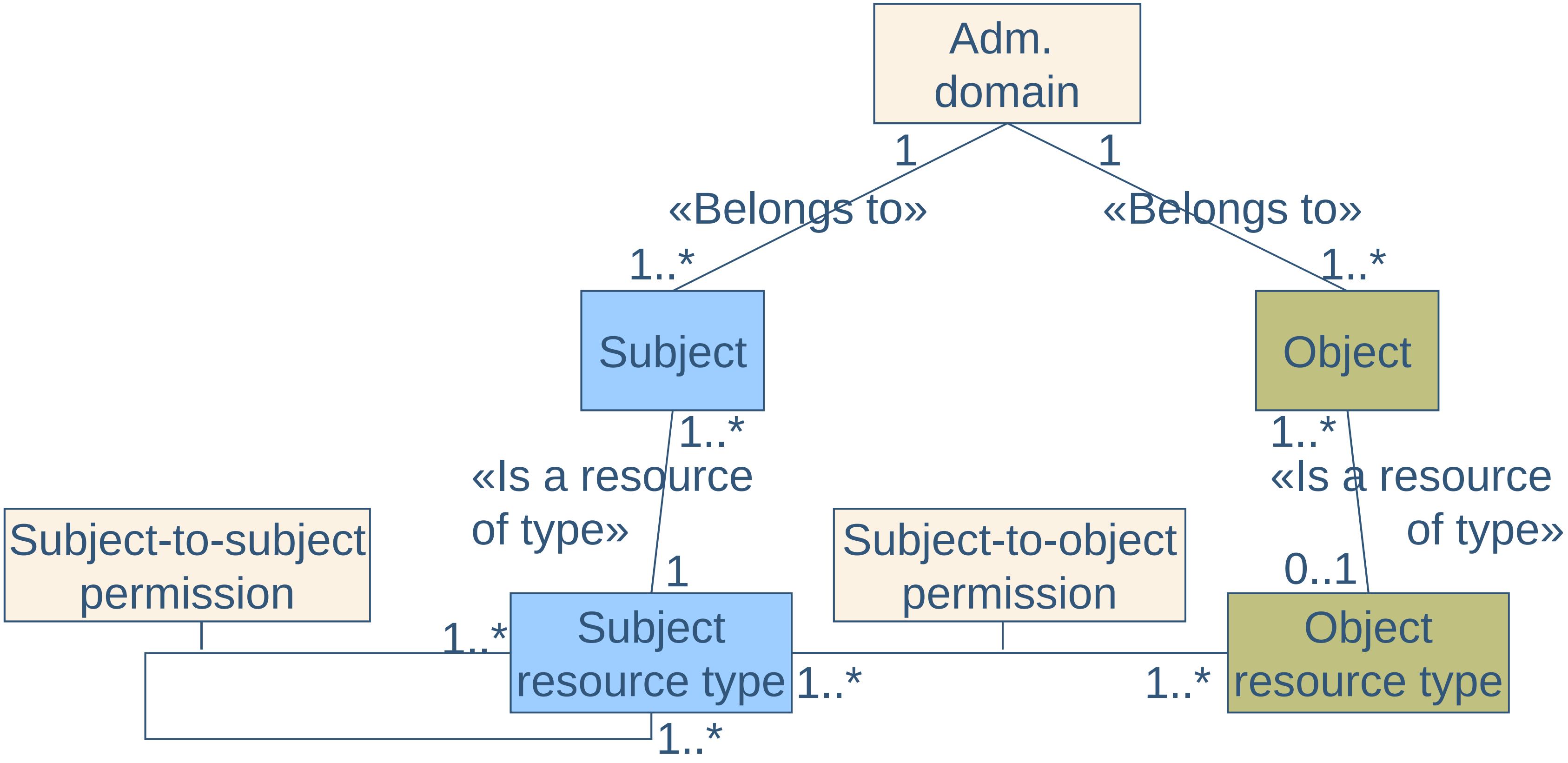
Transitive

Assumptions

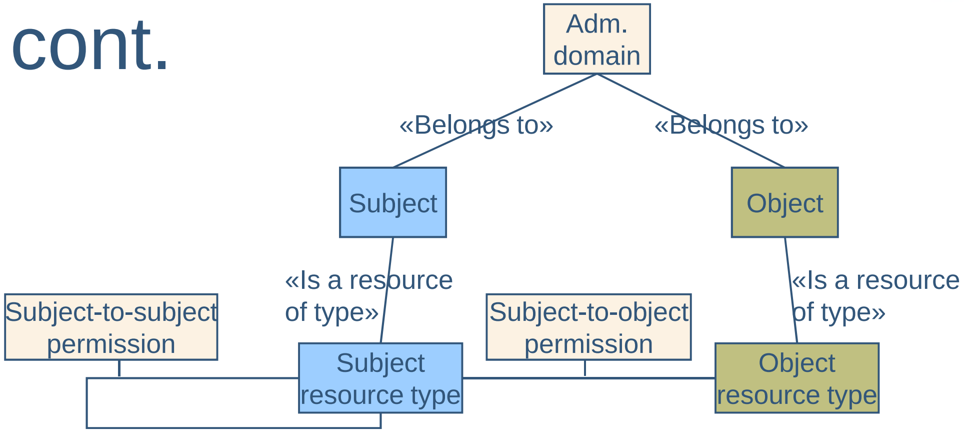
- Computers of the different administrative domains are connected to a common network.
- The computer systems within a domain implement the same set of security policies and levels.
- The cooperating parties (administrative domains) implement a *common* set of confidentiality, integrity and information availability policies, e.g. a set of NATO policies. However, the implemented security levels may vary from domain to domain.
- The SOTE resource availability policy is implemented in all actual administrative domains.
- Trust between cooperating parties has been established, and the cooperating parties have knowledge of the security policies and levels of the other part.
- Confidentiality, integrity and availability are independent security properties.

SOTE proposal

Information flow is controlled by defining the permitted interactions between types of *subject resources* and *object resources*.



SOTE proposal cont.

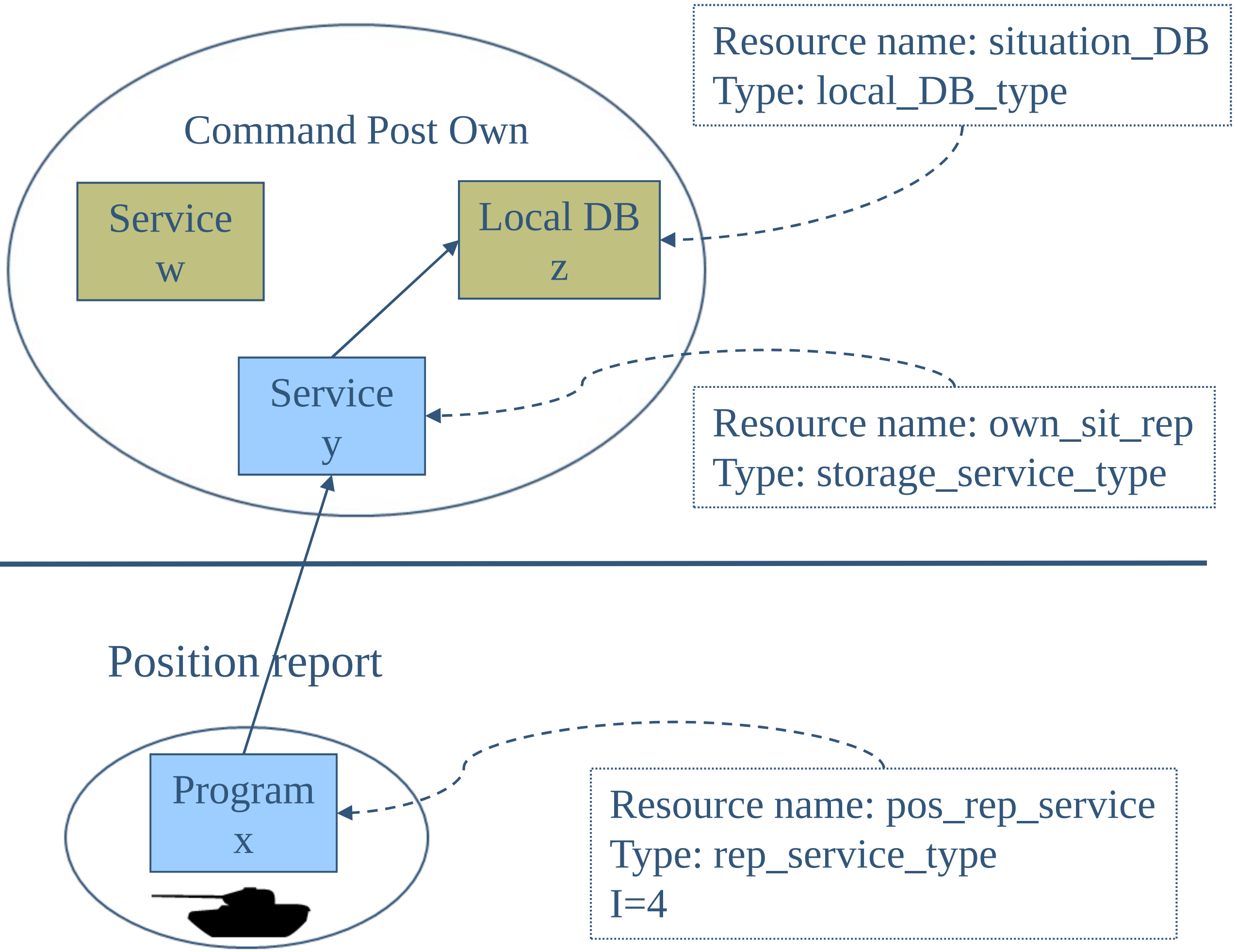


- Permitted **subject-to-object** interactions are specified for pairs of *subject resource type* and *object resource type*.
- The permission modes are none, read-related or write-related.
- Permitted **subject-to-subject** interactions are specified for pairs of *subject resource types*.
- In addition a set of security **requirements** and **conditions** can be associated with a *subject resource type* and an *object resource* type.
- Generalizations are used to define a *resource type* **hierarchy**.

Composite policy for cross-domain information flow - example

Own: adm. domain tactical

Multilevel security
policy (C & I).
C = {Unclass, .. ,Secret}
I = {1, .. ,7}
A = SOTE configuration



Resource name: situation_DB
Type: local_DB_type

Resource name: own_sit_rep
Type: storage_service_type

Resource name: pos_rep_service
Type: rep_service_type
I=4

D1: adm. domain combat

Single level confidentiality,
multi level integrity.
C = Restricted
I = {3, 4}
A = SOTE configuration

Related work

- Domain and Type Enforcement (DTE) is an enhanced version of type enforcements. Badger et al (1995), "Practical Domain and Type Enforcement for UNIX“
- DTE has been integrated with network services in a UNIX-based research prototype. Sherman et al (1995), "Controlling network communication with domain and type enforcement“
- The type enforcement security model is implemented in Security-Enhanced Linux (SELinux).

Summary

- A new security model for *resource* availability has been proposed, called SOTE.
- The SOTE model can express policies for information flow between resources of different administrative domains. It controls the *types* of resources that are allowed to interact.
- Type enforcements can express intransitive information flows.
- The model can express information flow policies at a fine-grained level.
- The ability to express the conditions a resource must fulfill, is also part of the model.
- Also a data model that describes SOTE and related security elements, using UML notation, has been proposed.