Willkommen zur Vorlesung

Sicherheit:
Fragen und Lösungsansätze

im Wintersemester 2012 / 2013

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Vorlesungswebseite (bitte notieren):
http://www-jj.cs.tu-dortmund.de/secse/pages/teaching/ws12-13/sfl/index_de.shtml

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems
Fundamentals of Control and Monitoring

Themen der Vorlesung

Part I: Challenges and Basic Approaches

1) Interests, Requirements, Challenges, and Vulnerabilities
2) Key Ideas and Combined Techniques

Part II: Control and Monitoring

3) **Fundamentals of Control and Monitoring**
4) Case Study: UNIX

Part III: Cryptography

5) Fundamentals of Cryptography
6) Case Studies: PGP and Kerberos
7) Symmetric Encryption
8) Asymmetric Encryption and Digital Signatures with RSA
9) Some Further Cryptographic Protocols

Part IV: Access Control

10) Discretionary Access Control and Privileges
11) Mandatory Access Control and Security Levels

Part V: Security Architecture

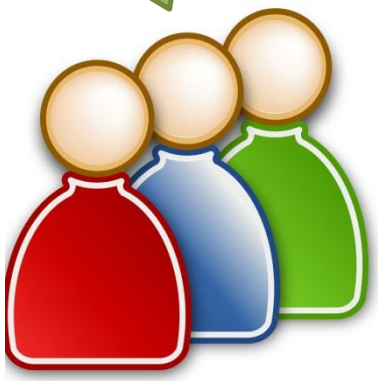
12) Layered Design Including Certificates and Credentials
13) Intrusion Detection and Reaction

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Problem Example

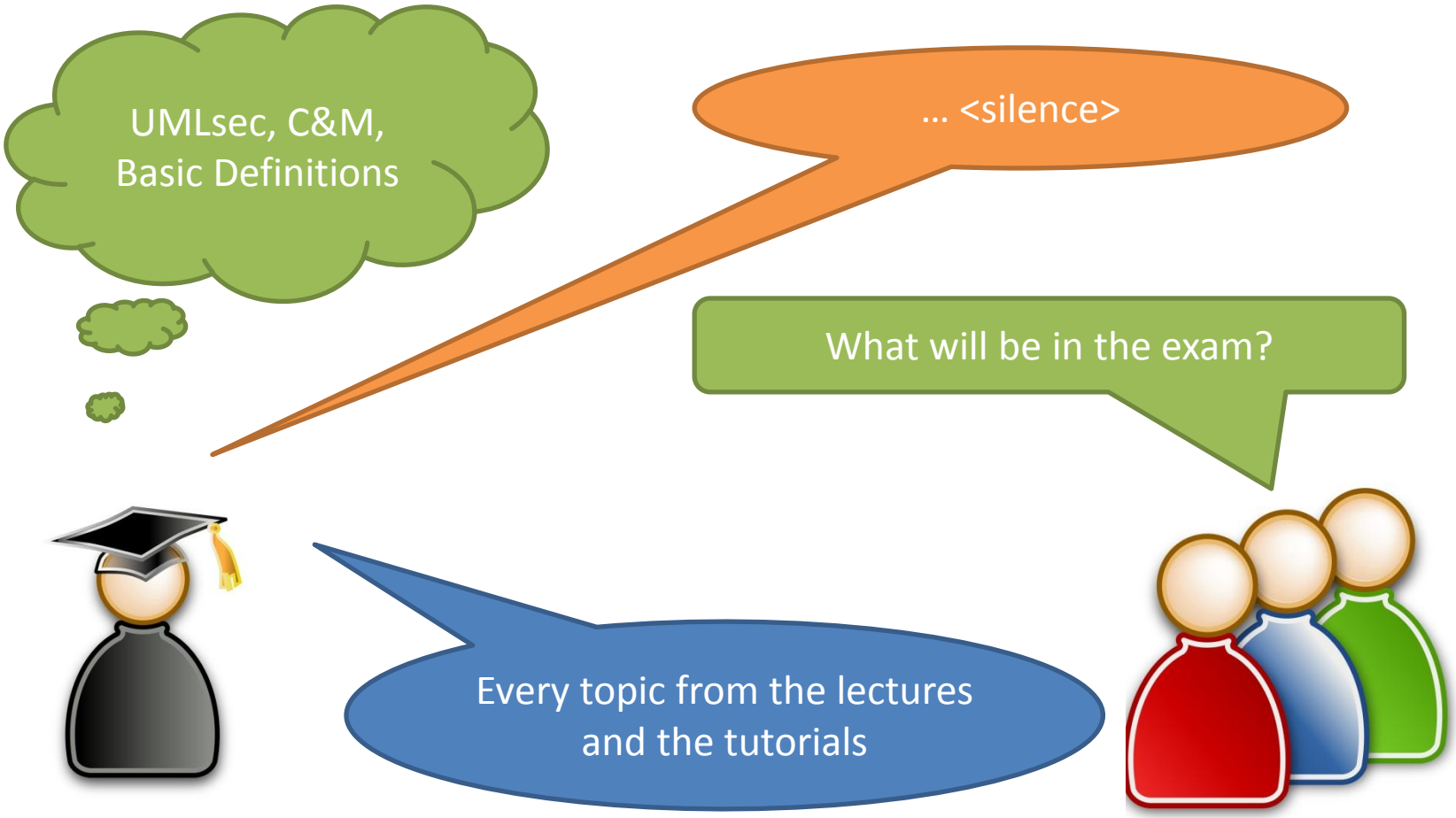
UMLsec, C&M,
Basic Definitions

What will be in the exam?



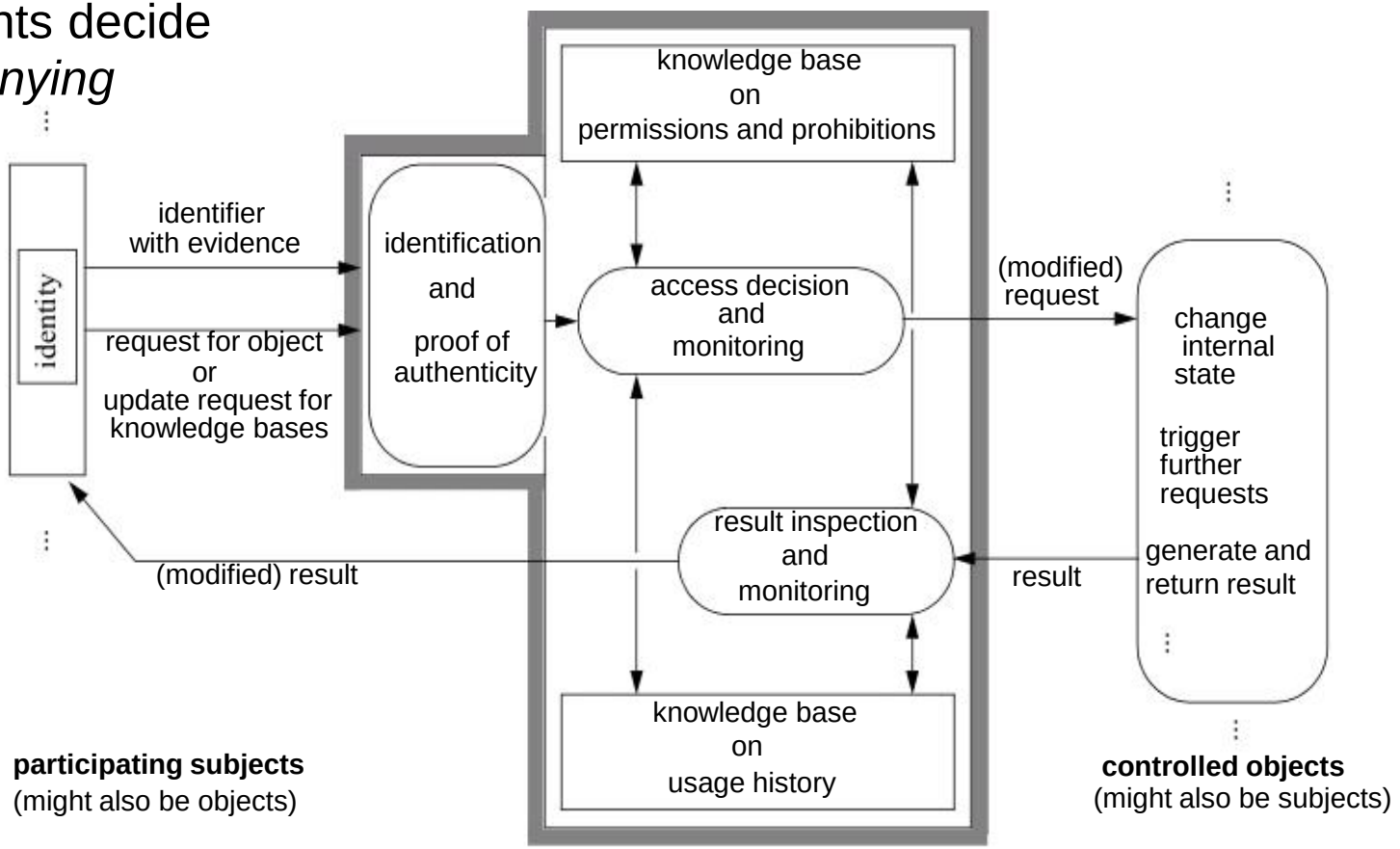
©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Problem Example



Control and monitoring

- identifiable agents can have *access rights* granted and revoked
- access requests of authenticated agents are intercepted by *control components*
- control components decide on *allowing* or *denying* an actual access



©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Essential parts

- declaration of permissions and prohibitions
- control operations
- isolation, interception and mediation of messages
- proof of authenticity
- access decisions
- monitoring

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Declarations: subjects, objects, and kinds of access

- conceptualize and denote the ***subjects***: carriers of permissions and prohibitions
- where appropriate, treat *collectives* of subjects in a uniform way

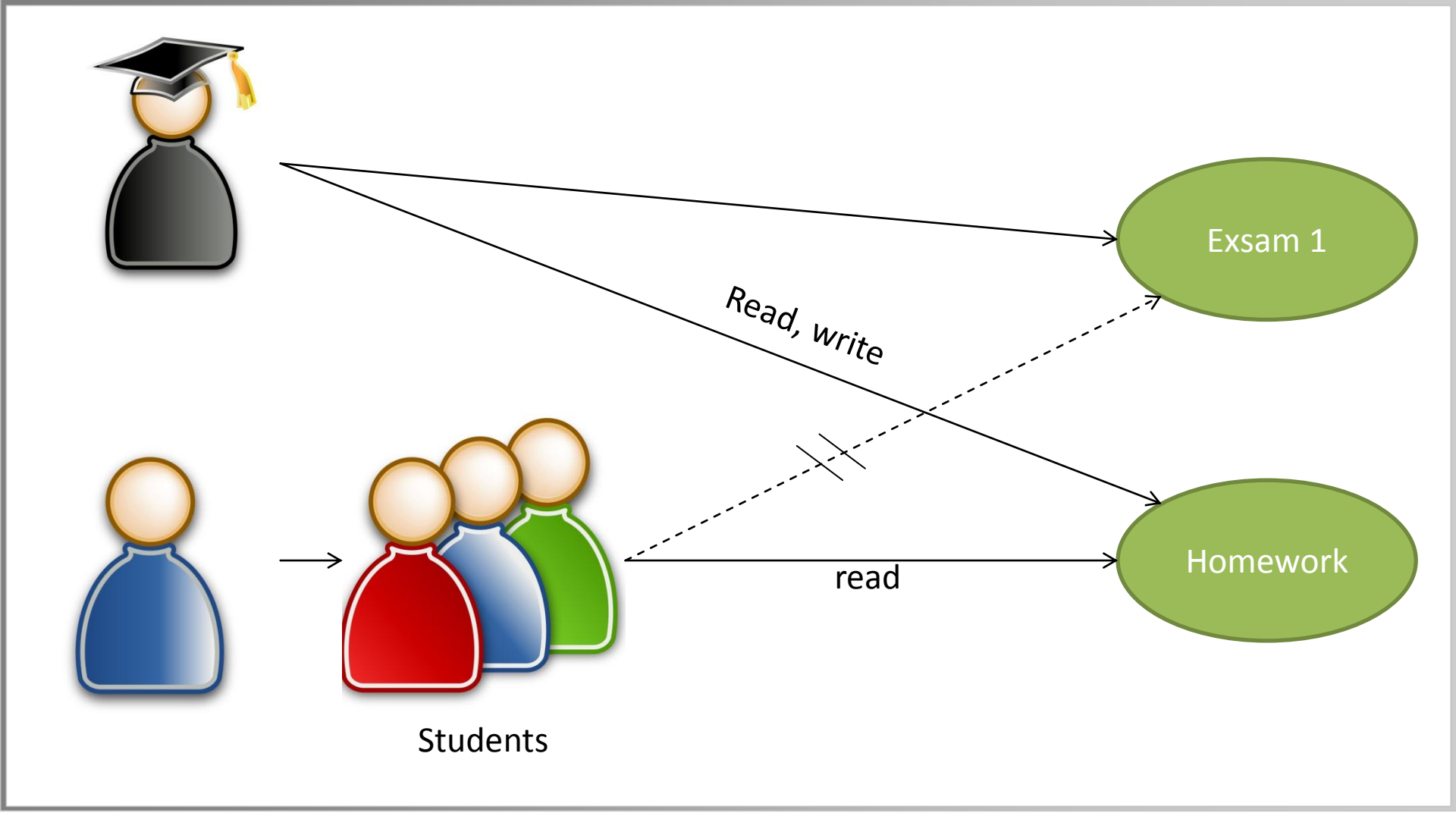
- conceptualize and denote the ***objects***: targets of permissions and prohibitions
- where appropriate, collect objects into
  classes, *domains* or related *aggregates* for uniform treatment

- conceptualize and denote the ***kinds of access*** offered:
from generic *reading* and *writing*
  to application-specific *methods*
- where appropriate, abstract from concrete accesses and
instead refer to their (*operational*) *modes*

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Declarations: subjects, objects, and kinds of access

LEHRSTUHL 14 SOFTWARE ENGINEERING



©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Declarations: expressive means

- a permission or a prohibition can be *directly* expressed
  by *explicitly* naming the respective subject, object and operational mode

- preferably, the needed items are expressed in a more *indirect* way,
employing a wide range of techniques of computer science
(programming languages, knowledge engineering, ...):

- in particular, *syntactic* means for
  - collectives of subjects (e.g., *hierarchies*),
  – aggregates of objects (e.g., *complex compositions*)
  – modes of access (e.g., further *method invocations*)
  must be suitably handled at the *semantic* level

- in general, techniques for deriving
  implicit properties of the items considered from explicit properties
  might be exploited
  (e.g., *inheritance* rules, *first-order logic reasoning*, ...)

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Declarations: positive, negative, and mixed approach

- positive approach:
  only **explicit permissions** expressible,
  and, by default, prohibitions defined as the absence of a permission

- negative approach:
  only **explicit prohibitions** expressible,
  and, by default, permissions defined as the absence of a prohibition

- mixed approach:
  both **explicit permissions and explicit prohibitions** expressible,
  with a need for the *resolution of conflicts* and for *completions*

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Required completeness property for declarations

for any request

of a subject *s*

to access an object *o*

in an operational mode *m*,

the declared permissions and prohibitions entail

a unique and definite access decision

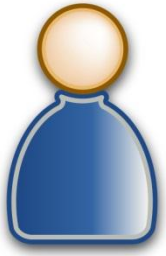
©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Declarations: subjects, objects, and kinds of access

for any request
of a subject s
to access an object o
in an operational mode m,
the declared permissions
and prohibitions entail
a unique and definite access
decision

Read, write

Exam



read

Homework

Stude
nts

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Control operations

- first level:
  permissions and prohibitions for the *functionality* of a system, i.e., for *functional operations*

- second level:
  permissions and prohibitions for the ***control operations***
  that manipulate the first-level functional permissions and prohibitions,
  including ***granting*** and ***revoking*** of *functional* permissions and prohibitions;
  more advanced control operations deal with, e.g.,
  transferring or ***delegating*** permissions and prohibitions
  to declare functional permissions and prohibitions

- further levels:
  possible, but rarely employed

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Grantors and owners

- need to define which subjects may grant permissions and prohibitions
  - initially
  - by means of some special qualifications

- example:
  a (nearly) omnipotent *administrator*,
  known as *root* or *super-user*, is permitted
  - to manage *any kind* of permissions and prohibitions
  - to assign each subject that generates a new object
    the *ownership* of the creation, coupled with the permission
    to manage the permissions and prohibitions *for that creation*

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Control states

- the granting of permissions should be done with great care

- an administrator or owner planning some control operations has to *analyze* the potential consequences regarding
  which subjects can *eventually* acquire which permissions

- more generally, for any *control state* resulting from control operations, such an analysis should be performed
  (unfortunately, in general computationally infeasible or even impossible)

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Required analysis property for control operations

for any control state resulting from control operations,

the analysis problem regarding

which subjects can eventually acquire which permissions

should be computationally feasible

or at least admit a computational approximation

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Isolation, interception and mediation of messages

- effective enforcement of declared permissions and prohibitions relies on an appropriate system *architecture*:
  - it strictly *isolates* subjects from objects
  - it considers that some entity might act *both* as a subject *and* as an object
- a subject should *not* be able to *directly* access any object
- a subject can send a message containing an *access request* that will be *intercepted* by a separating *control and monitoring component*
- the control and monitoring component mediates the request, basically in three steps:
  - *identification* and *proof of authenticity*
  - *access decision* and forwarding
  - further *monitoring*

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Required complete mediation property

each request of a subject

to access an object

is intercepted and mediated

by a control and monitoring component

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Proof of authenticity

- declared permissions and prohibitions refer to well-conceptualized *subjects*
- the control and monitoring component must relate
  the sender of any request message, an actual *requestor*, to a pertinent subject
- given a request message,
  the control and monitoring component must *recognize* the requestor
  as one of the conceptualized subjects,
  being aware of the possibility of a maliciously cheating agent
- the requesting agent must provide some further *evidence* regarding itself;
  the control and monitoring component can then base a *proof of authenticity* on
  - the *freshly* communicated *evidence*
  - suitably maintained permanent *verification data*

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Required authenticity property

any mediation of an access request

is based on a proof of authenticity

of the requestor and,

as far as needed,

of the target object as well

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Access decisions

- once a requestor has been recognized as a conceptualized subject,
the control and monitoring component takes an *access decision*
by evaluating the request with respect to
  the previously declared permissions and prohibitions

- the declarations constitute a *knowledge base on permissions and prohibitions*,
from which the access decision is derived as a logical consequence

- such *derivations* might vary
  from simple lookup procedures to highly sophisticated reasoning

- such *reasoning* might additionally consider
  the *dynamic evolution* of the controlled system,
  as conceptually represented by a *knowledge base on the usage history*

- such a knowledge base must be appropriately maintained
by *logging* all relevant events

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Requirement for architecture of control

the control and monitoring component

maintains suitably isolated knowledge bases

on permissions and prohibitions and

on the usage history

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Monitoring: inspecting results

- an accepted and forwarded request might produce some *results* that should be *inspected* afterwards

- if the *results* are to be *returned* to the original requestor: the inspection might retain all or some parts of them:
  - totally *block* the forwarding to the requestor, or
  - suitably *modify* the results before forwarding

- if an *internal state* of an accessed object might have been changed or *further requests* to other objects might have been triggered:
  the options for undoing such effects depend strongly on additional mechanisms such as *transactions*, seen as atomic actions that can be finally either completely *committed* or *aborted*

- in case of an *abort*, the effect should be (largely) *indistinguishable* from the situation where the access has not occurred at all

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Monitoring: auditing and intrusion detection

- complementary to access decisions and result inspection, the control and monitoring component can analyze all messages and possibly further audit data regarding an *intrusion defense policy*

- such a policy assists in classifying the activities actually occurring as either *semantically acceptable* or *violating*

- the notions of *permissions* and *prohibitions* should be semantically related to the notions of *acceptable behavior* and *violating behavior*, respectively

- in general, however, these notions will not fully coincide because of
  - inevitable shortcomings of the preventive access control mechanisms
  - efficiency considerations (leading to an *optimistic* approach)

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Requirement for architecture of monitoring

complementarily to access decision and result inspection,

the control and monitoring component

audits and analyzes all activities regarding potential violations

defined by an intrusion defence policy

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Imagined ideal and real world

- **ideal world:**
  - all subjects behave as expected
  - all informational devices actually operate as completely specified
  - correct and complete knowledge is available whenever needed

- **real world:**
  - such an imaginary scenario is not met with at all
  - security aims at managing the imperfections, including:
    - potentially maliciously behaving subjects,
    - failing implementations of inadequate designs,
    - decision making regarding remote subjects

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Root of trust

- there always remains the need
  to base at least small parts of an overall computing system on *trust*

- trust in a *technical part* usually means,
  or at least includes the requirement,
  that the *participant* controlling that part is trusted

- as security is a *multi-lateral* property
  that respects potentially conflicting *interests*,
  trust is essentially *context-dependent*, i.e.,
  subjectively assigned by one participant
  but refused by another one

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems

Issues of trust raised when the following problems are investigated

- does the *control and monitoring component* actually work as expected, intercepting and suitably mediating each access request?
- does it support *availability* by accepting permitted requests, and does it preserve *integrity* and *confidentiality* by denying prohibited accesses?
- do participants permitted to execute *control operations* behave appropriately and honestly when granting, revoking, transferring or delegating permissions?
- do shown *evidence* and maintained *verification data* reflect the actual *peculiarities* of remote communication partners?

©2009 Springer-Verlag Berlin Heidelberg / ©2010 Joachim Biskup TU Dortmund / Jan Jürjens : Security in Computing Systems