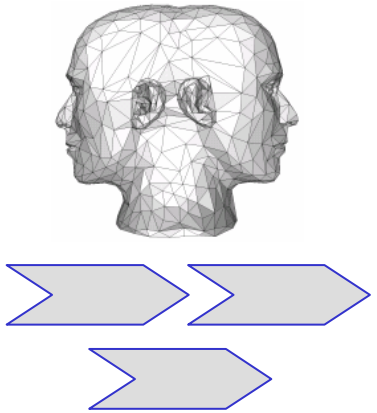


Modelling Generic Processes for the Identity- & Access Management

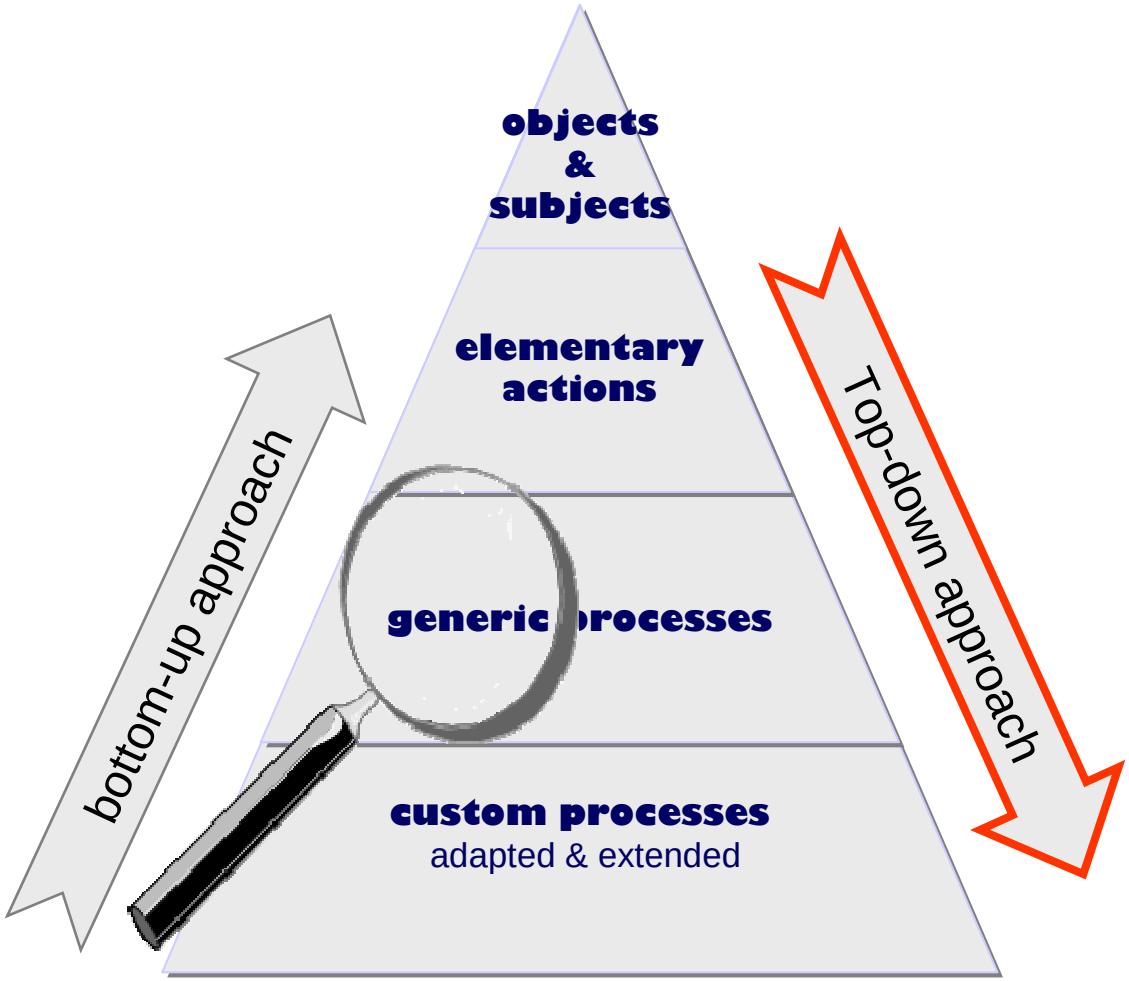
The top-down modelling approach

Version 0.23

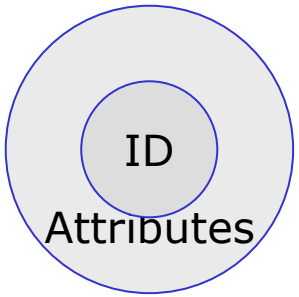
2008-01-25, Dr. Horst Walther

Modelling top-down

deriving genericity from interactions of generic objects

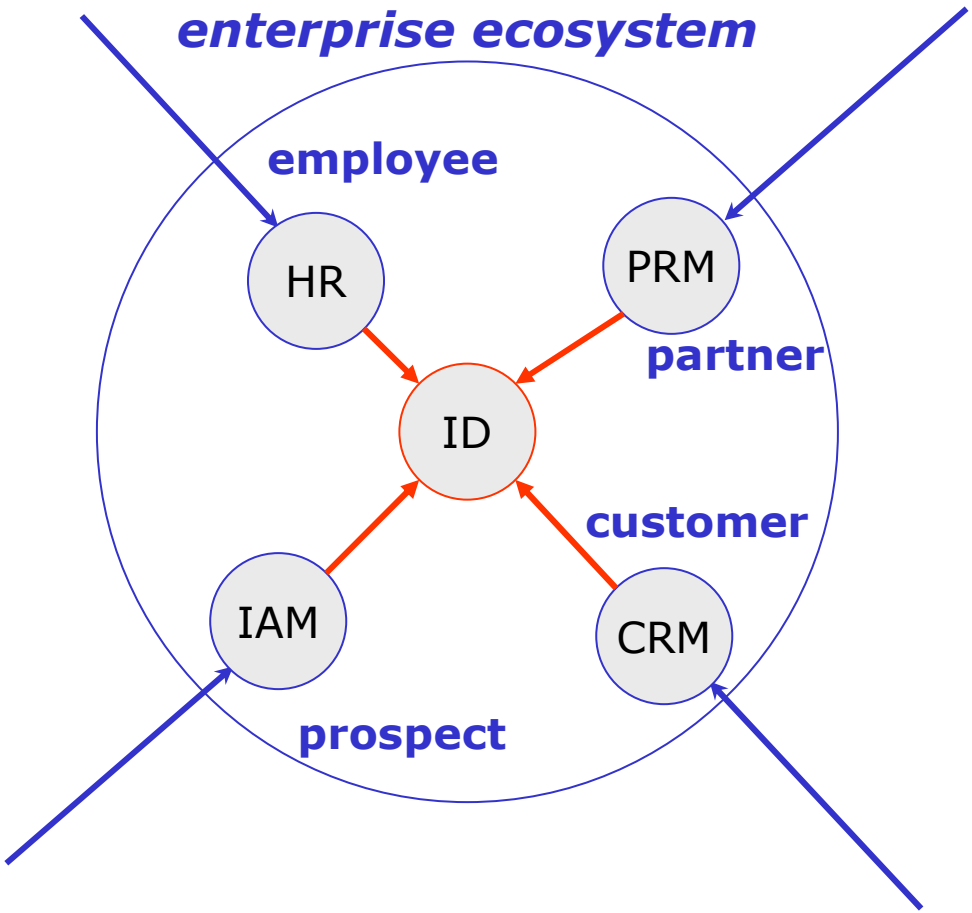


The Identity
and its “less rich” sibling the digital identity



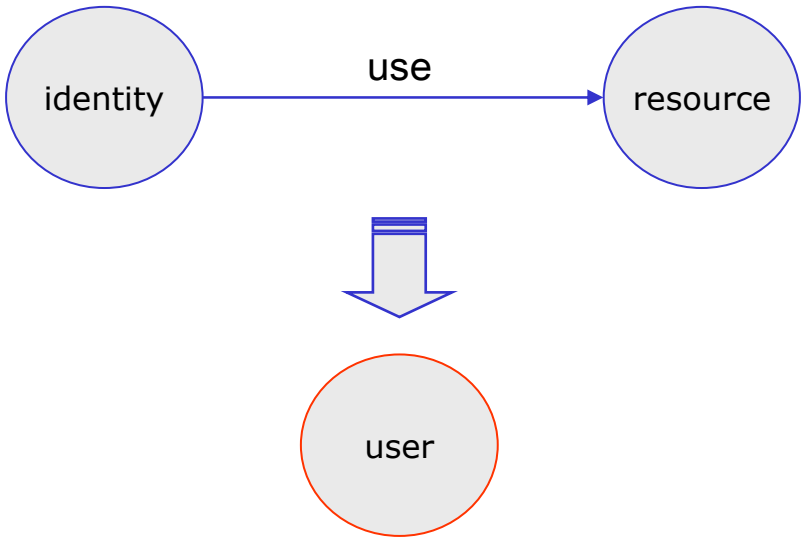
- Identity is the fundamental concept of identity management
- In philosophy Identity is the sameness of two things.
- In object-oriented programming Identity is a property of objects that allows the objects to be distinguished from each other.
- But in Identity Management …
  - “*We usually speak of identity in the singular, but in fact subjects have multiple identities*.”
  - “*These multiple identities or personas, as they are sometimes called, …*”.
- The sum of all these Personas makes up the identity.
- In turn personas are to be understood as its projection to the space of information demand in a specific context.
- Biometrics ties the digital identity to the real world physical identity.

The central digital identity
whenever an individual enters the enterprise ecosystem 1st …



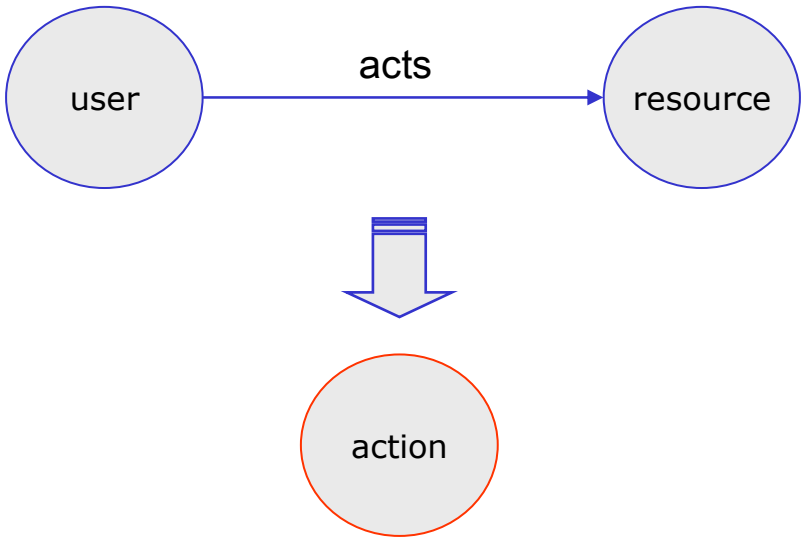
- Its digital identity is created whenever an individual enters the enterprise ecosystem 1st time.
- Regardless if it is a user or not
- Being a *user* represents a class of roles already
- The digital identity is the individuals digital sibling.
- Its lifetime is determined by the lifetime of the enterprises interest.
- The digital identity is global and unique
- It carries the minimal identifying attributes.

The User: Identity uses a Resource



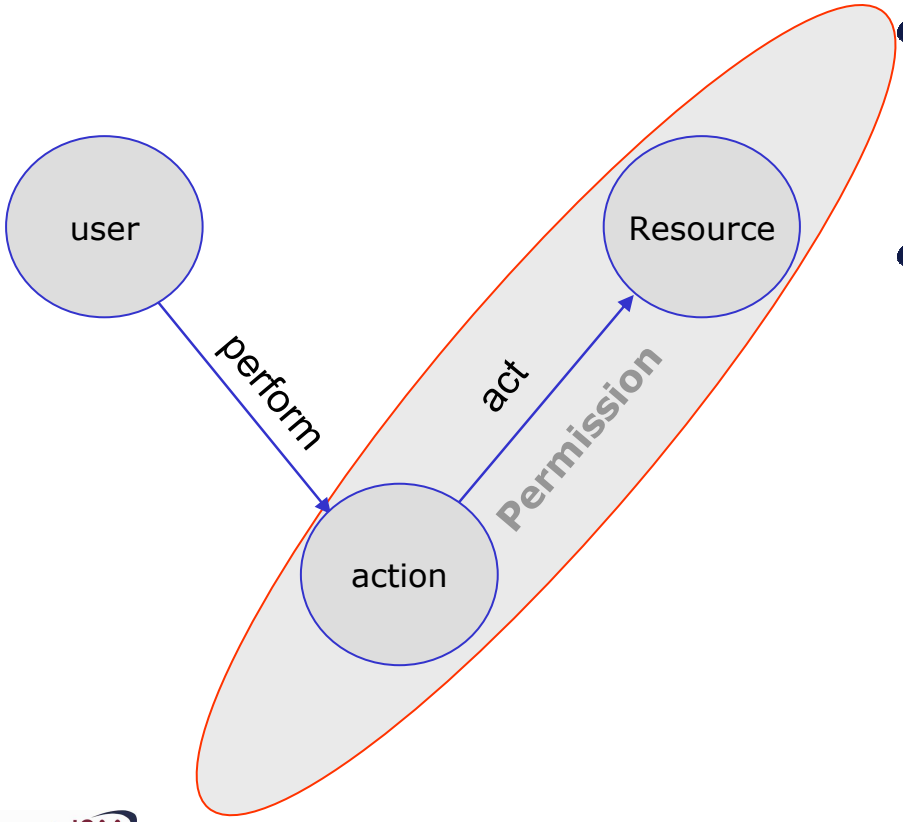
- Identities are often tied to resources
- They „use“ resources
- They do so by performing actions
- This relations may carry attributes
- It turns to a derived object: the user.
- Account is a synonym for user.
- The “user” is the identity's relationship to the resource.

The action: user acts on the Resource



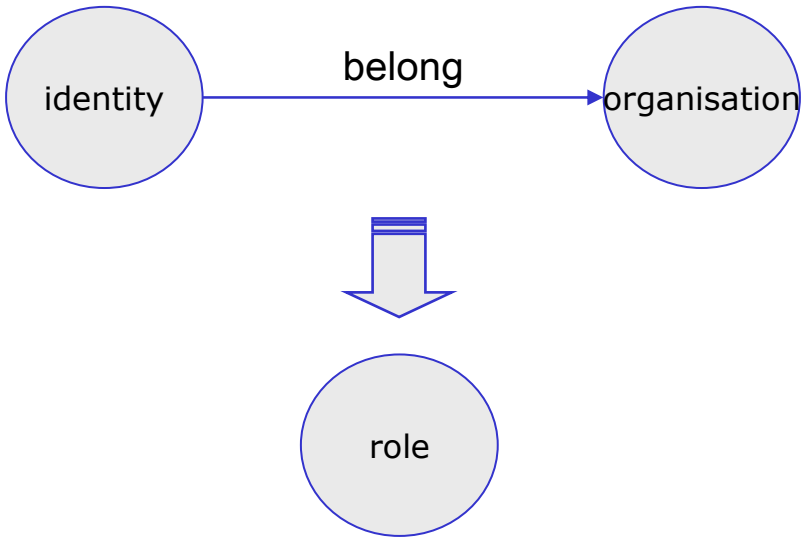
- users perform different actions on resources
- They „act“ on resources
- They do so by performing actions
- This relations may carry attributes
- It turns to a derived object: the user.

Permission = actions on Resources



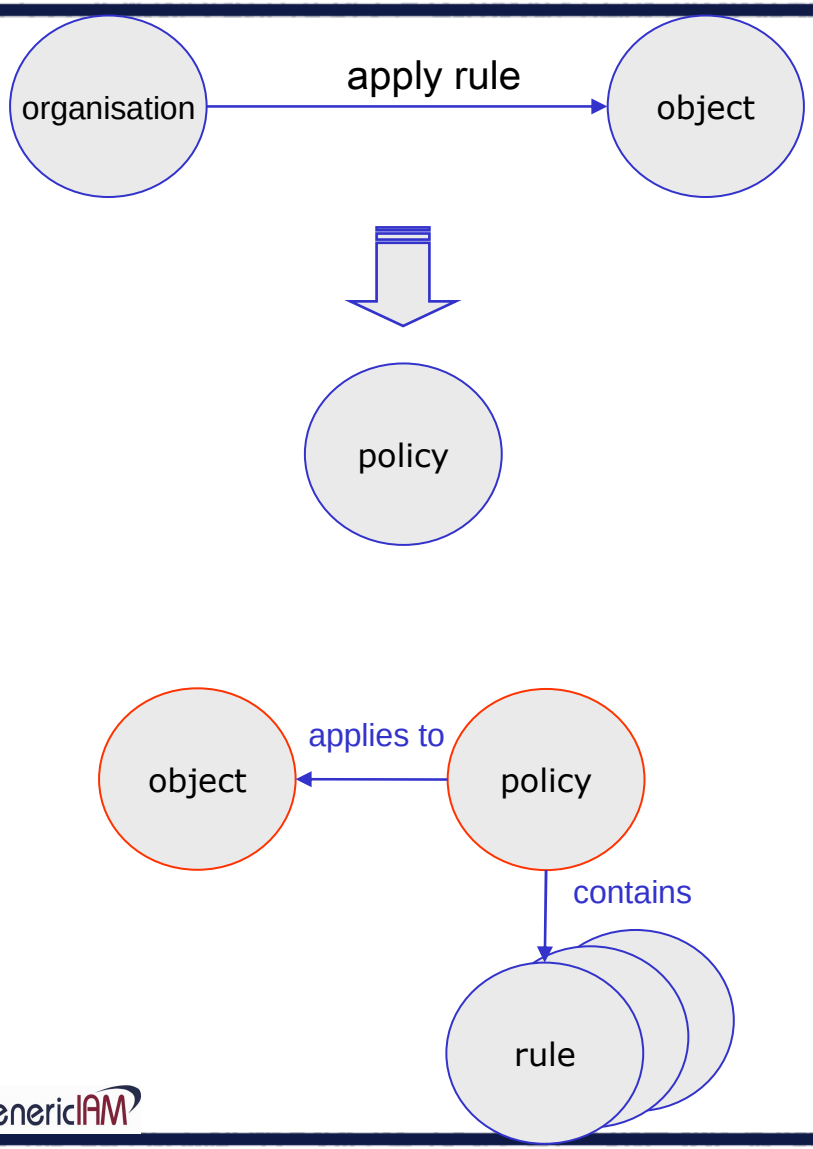
- The user performs an action
- The action acts on the resource
- actions on resources (objects) may be labelled with “permissions”.
- Permissions are elementary
  - They are simple by definition
  - There may be a large number
  - There is a limited set of permissions

The Identity belongs to an organisation



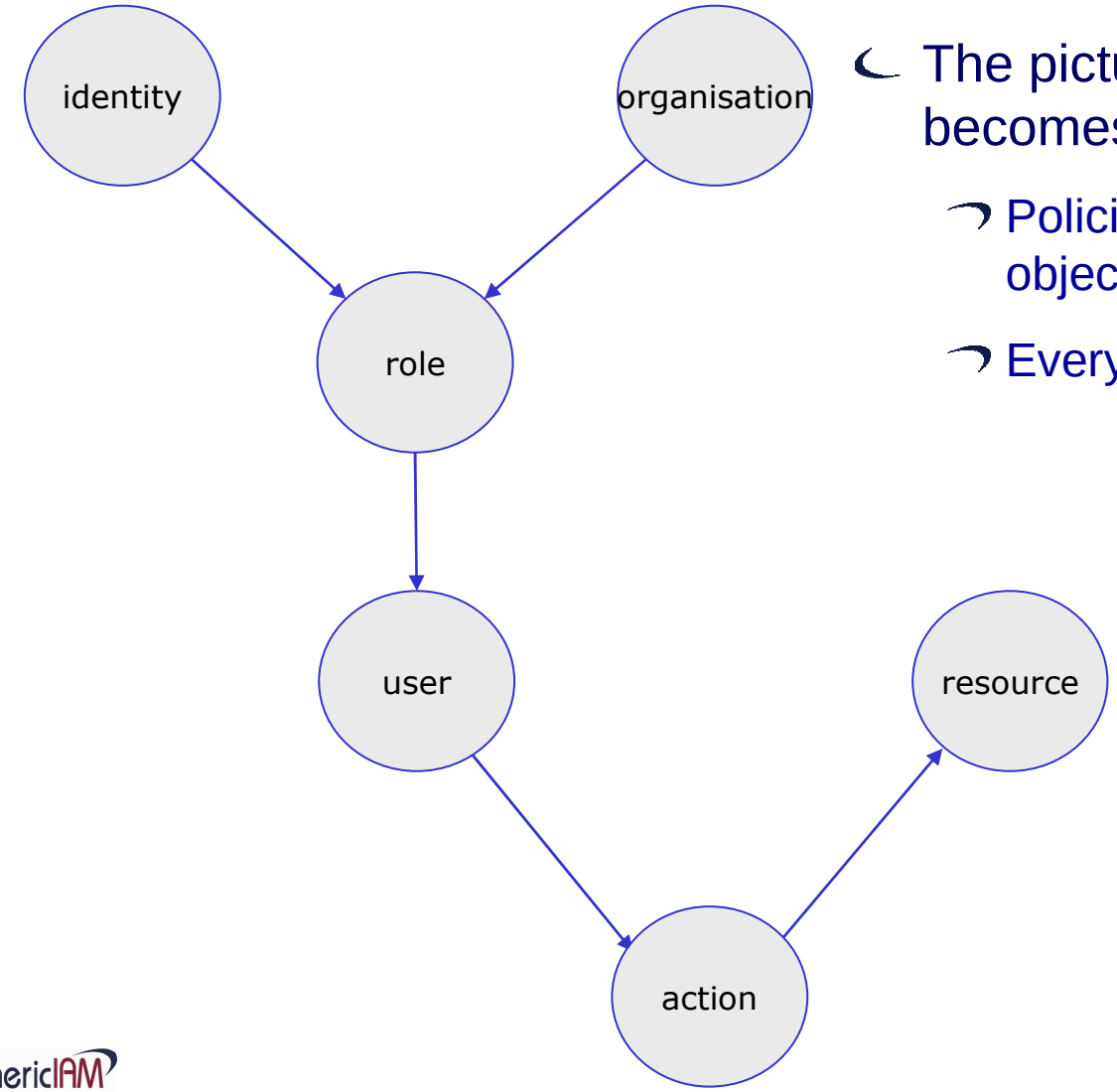
- The Identity has a **relationship** to an organisation
- It typically has an **owner** in this organisation
- There might be **more than one** relationship
- There are many **specialisations** to this relationship
- This relationship may carry attributes
- It turns to a derived object: the individuals **role**.
- The **role type** is a predefined class of relationships of an identity to an organisation.
- When the **role type** is assigned to an identity parameters are set to form the individual **role**.

To each object policies can be applied



- Policies are sets of rules
- The rules are generally applied on an objects state change
- Applying rules to objects needs to be expressed as an abject of its own.
- Policies can be attached to all objects
- Most obvious are SoD- (separation of duties) policies
  - A SoD policy applies to roles
  - A SoD policy contains several SoD rules
  - Only static SoD is considered here
  - Dynamic SoD requires the introduction of the object ‘session’

The Identity is linked …
to resources via other objects



- The picture grows and soon becomes complex
  - Policies may be applied to all objects
  - Every object has an owner

The role
actual vs. type

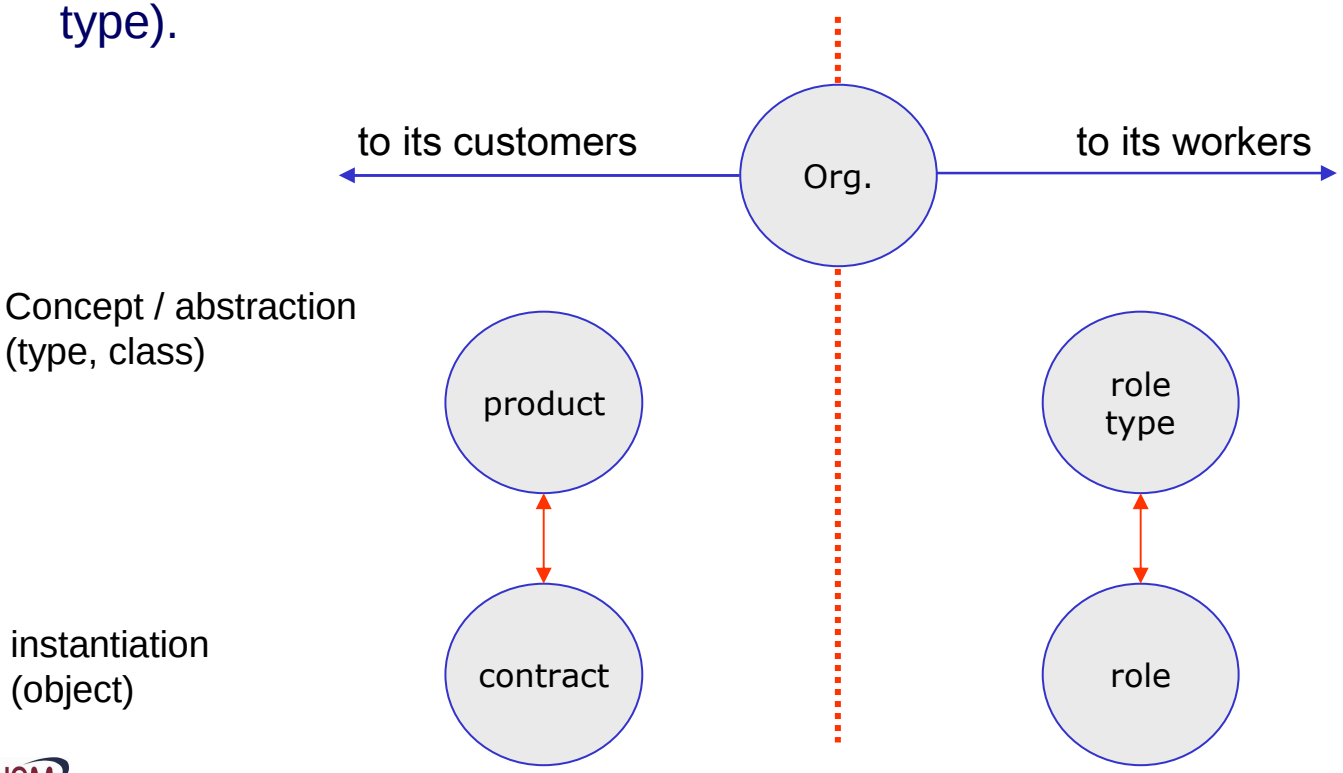
- The **role** is a bundle of privileges
- The **role type** is its abstraction
- Like the „**product**“ abstracts the „**contract**“

… the **role type** abstracts to **role**.

- The assigned individual role looks similar to an employee **contract**.
- Both may in fact may be one “**agreement**”.
- They may as well be left separate.
- A customer may receive a role as well.
- The role and the contract may well be one agreement (collapse to one).

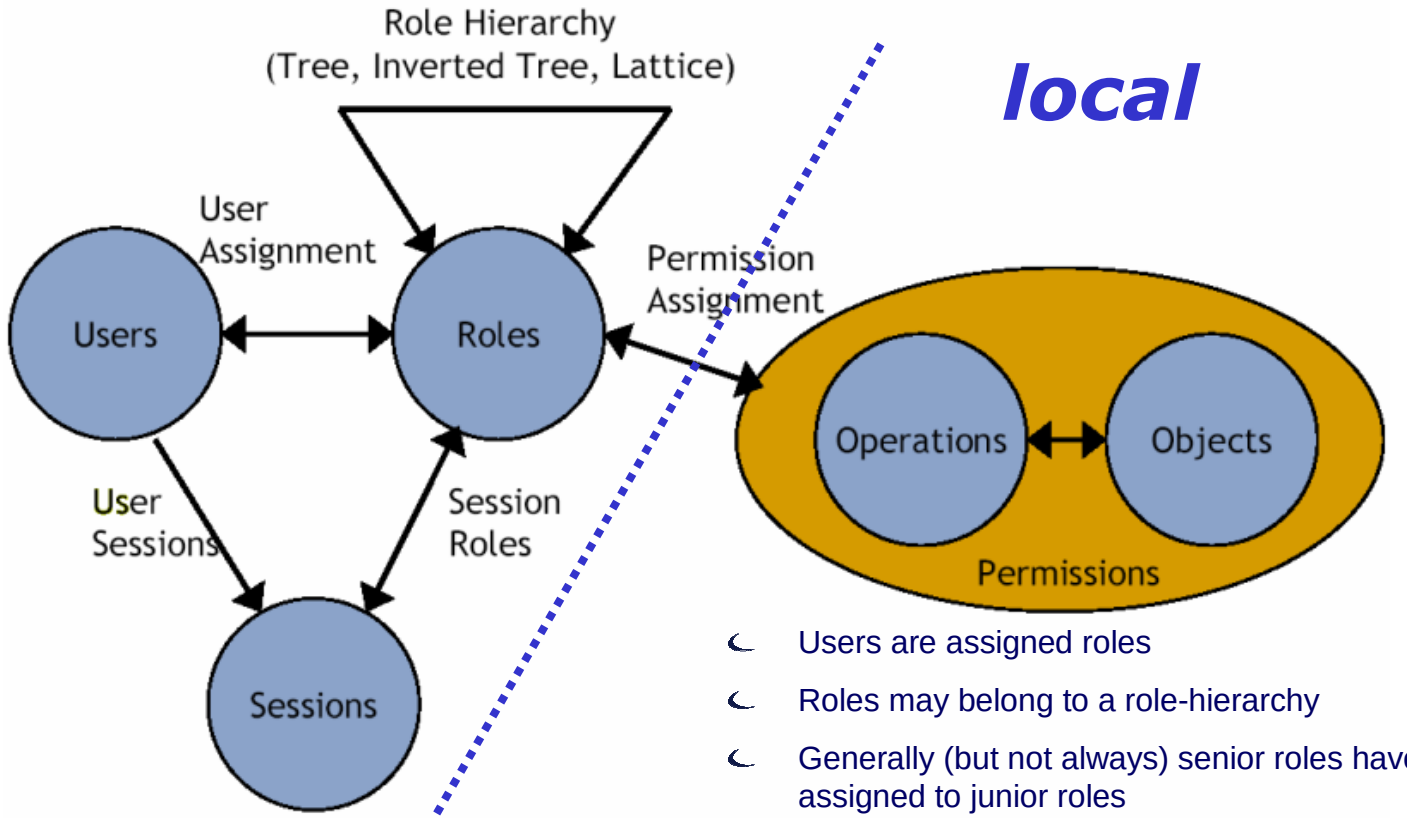
The concept of a role is an abstraction
like the product to the contract.

- The **product** generalises the contract. It is a **contract type**.
- The contract instantiates the concept of a product (or contract type).
- The **role type** generalises the **role**.
- The **role** instantiates the concept of a **role type**.



Central vs. Local

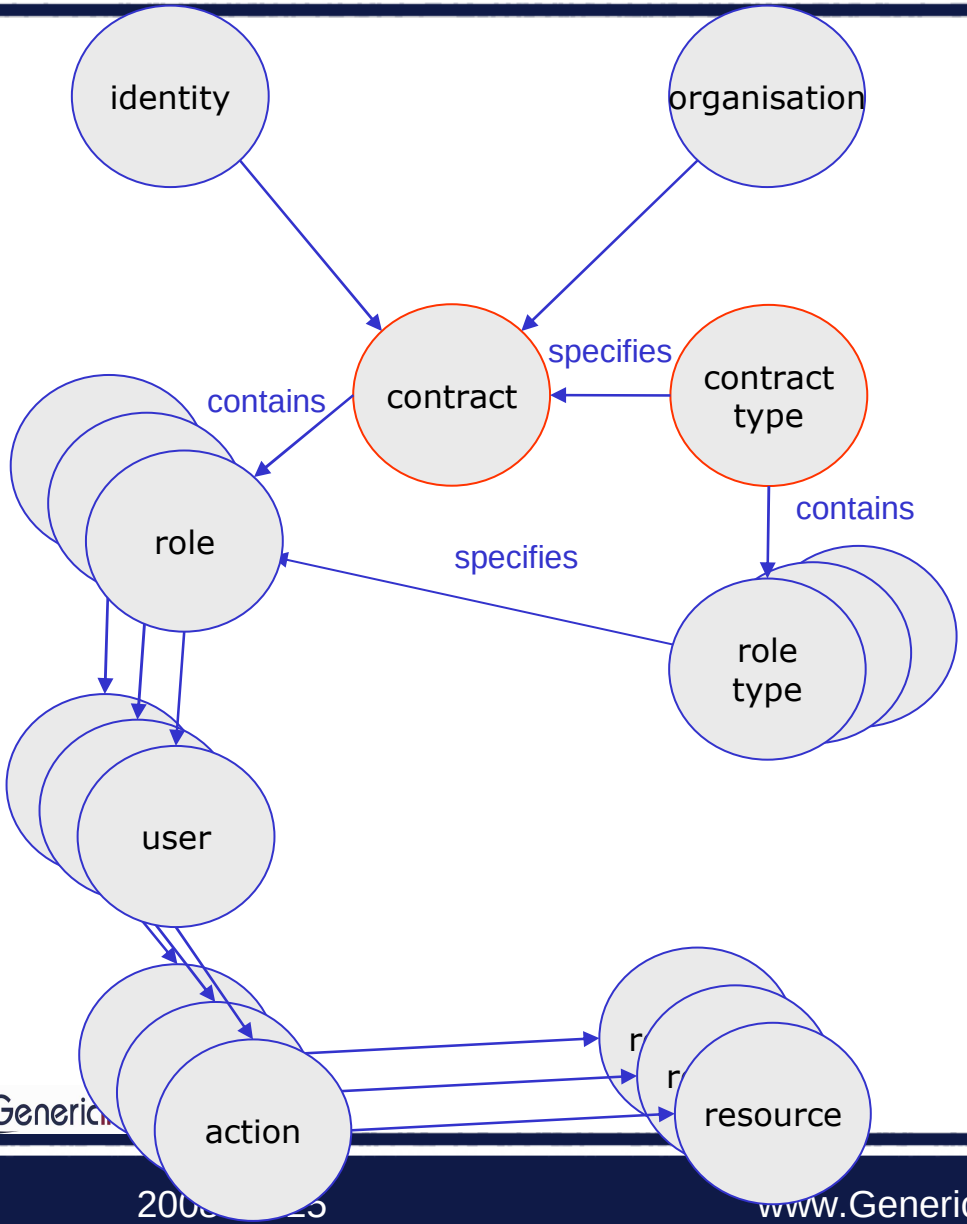
IDs & roles are central by nature, while permissions are local



- Users are assigned roles
- Roles may belong to a role-hierarchy
- Generally (but not always) senior roles have all permissions assigned to junior roles
- Permissions are actions on objects.
- Permissions can be assigned + (additional) or - (subtractive)
- Roles can be assigned temporarily per session

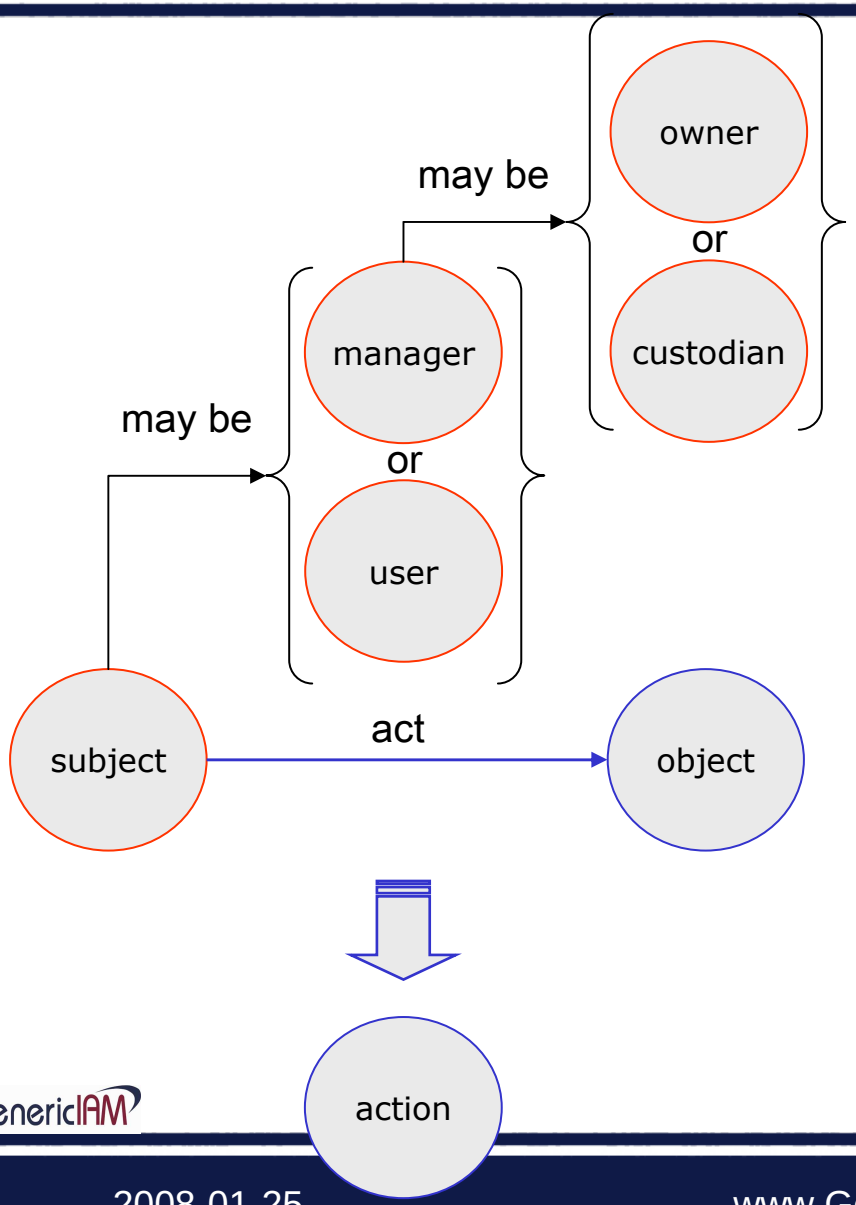
Source: Ferraiolo, Sandhu, Gavrila: A Proposed Standard for Role-Based Access Control, 2000.

Relationships are fixed in contracts



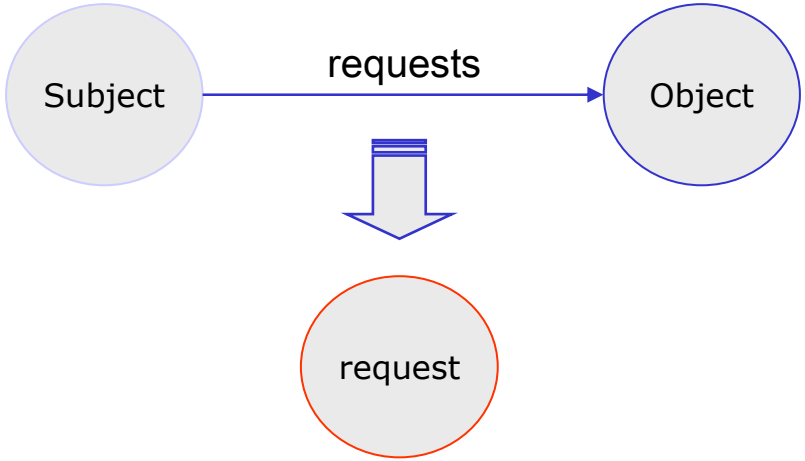
- The Identities role in an organisation performs actions on resources
- The role has a fine structure.
  - a contract defines the relationship
  - a role defines incarnation details
  - “the contract is expressed by several roles”

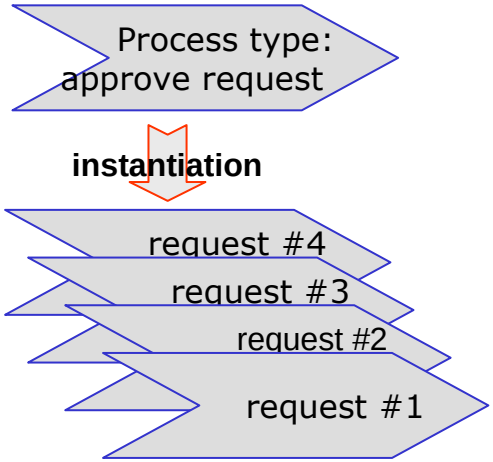
Subjects are acting on objects



- In processes subjects (actors) act on objects
- Subjects may be users or managers
- Managers are owners or a custodians
  - Owners are responsible
  - custodians act on behalf of owners
  - Owners delegate to custodians
- Subjects act or react
  - Their action triggers an event
  - Reactions often are approvals
- Time may act as a (virtual) subject
  - It acts on behalf of the organisation
  - They are driven by policy
  - Time-triggered events are common

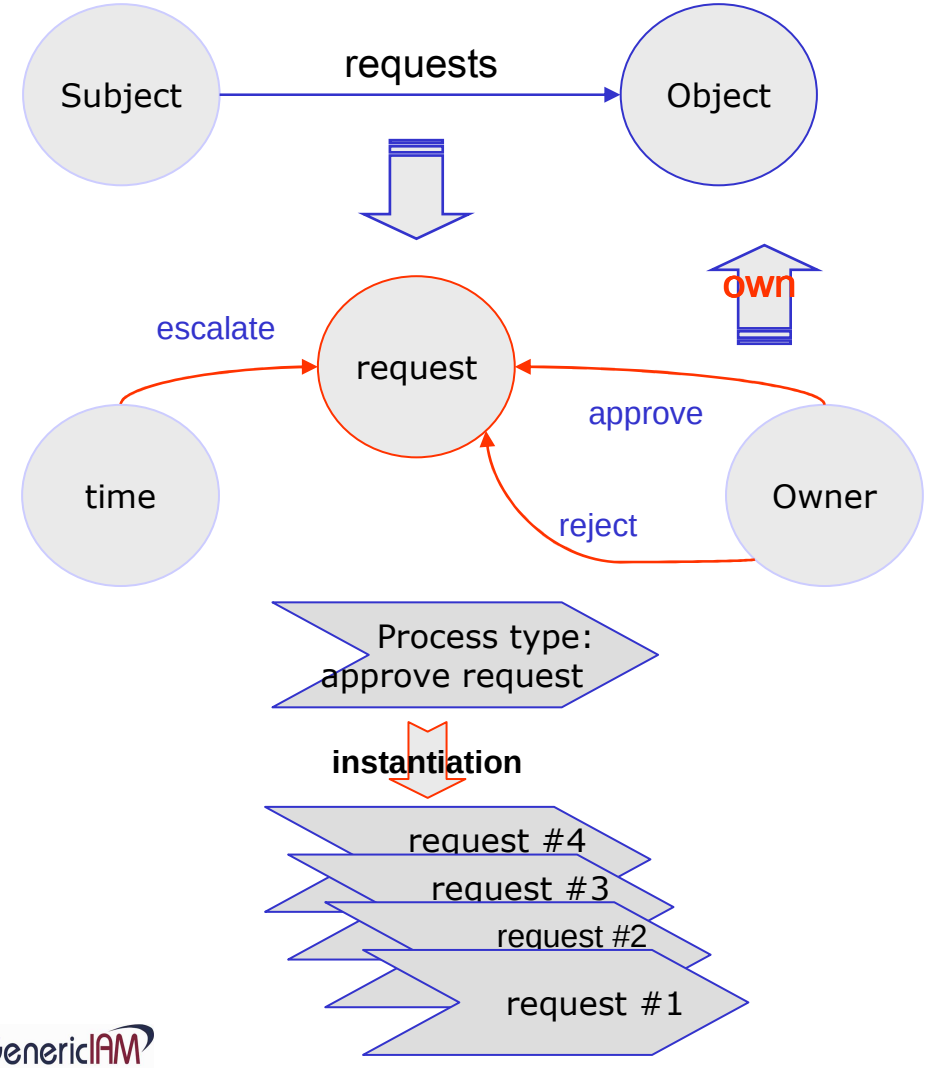
Request & approval





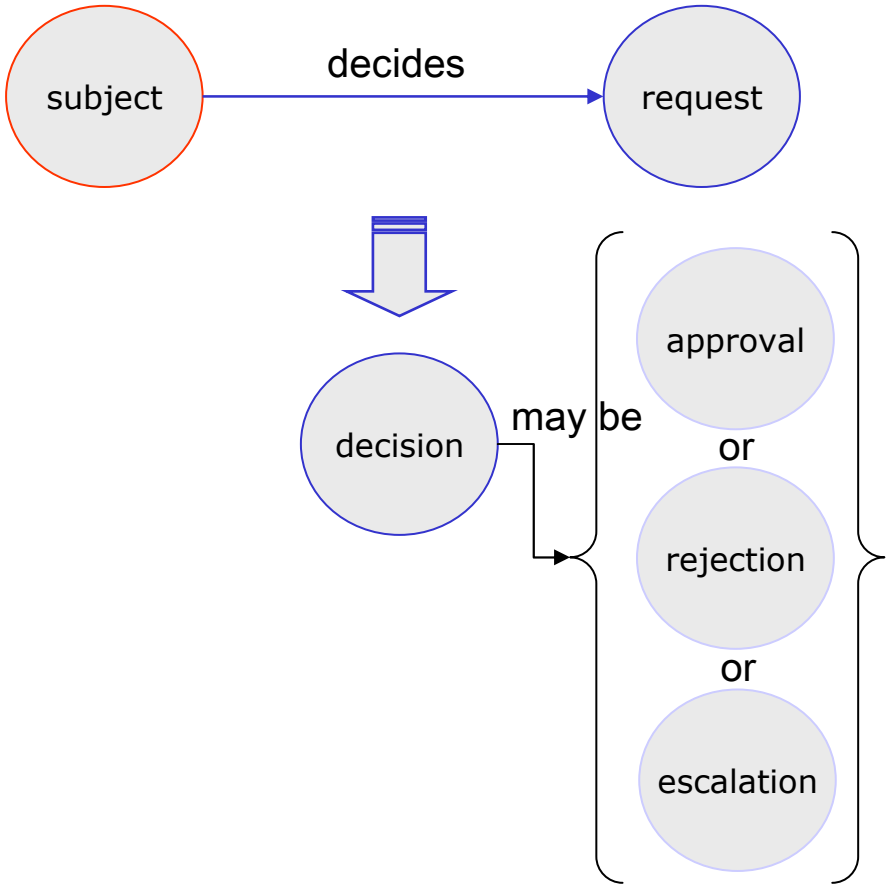
- The request is a **transient** object.
- It is the central **workflow** object
- It can be understood as the **instantiation** of a process type.
- The request is created by an **event**.
  - when a **subject** requests access to an object.
  - when **time** has come to re-validate a role / privilege.
  - when the defined response **period** has been passed without an action (escalation)

Request & approval



- The request is a **transient** object.
- It is the central **workflow** object
- It can be understood as the **instantiation** of a process type.
- The request is created by an **event**.
  - when a **subject** requests access to an object.
  - when **time** has come to re-validate a role / privilege.
  - when the defined response **period** has been passed without an action (escalation)
- The objects owner decides on the request
  - Changes its state
  - States are:
    - Approved
    - Rejected
    - Escalated
- As many requests as objects owners can be expected.

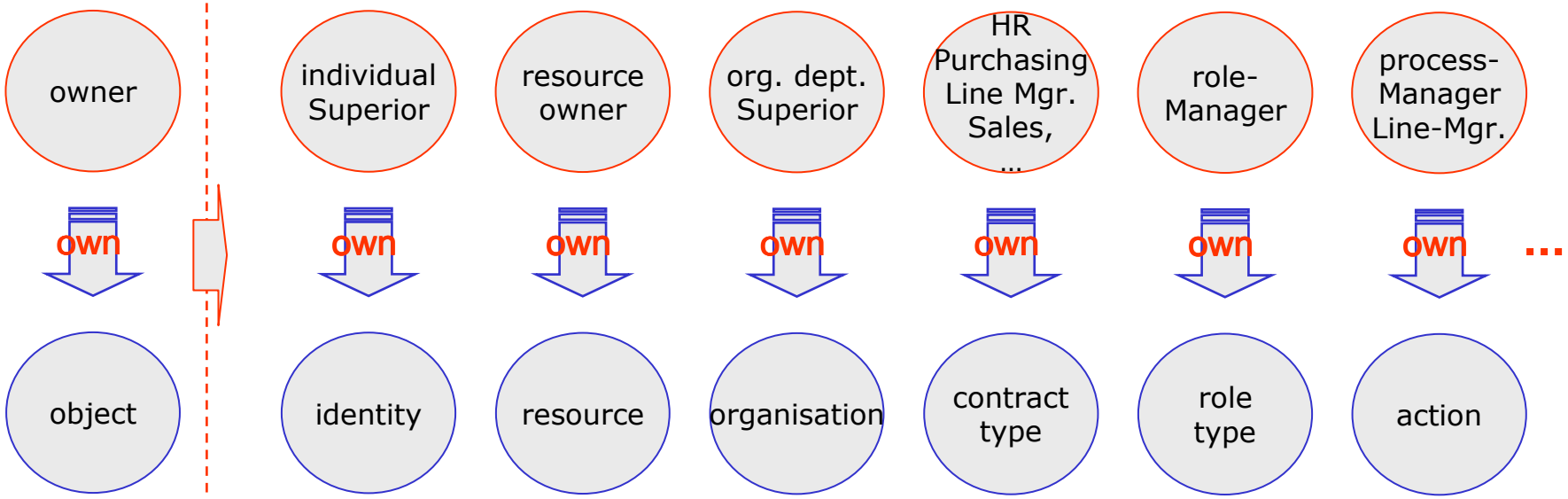
Subjects decide on requests



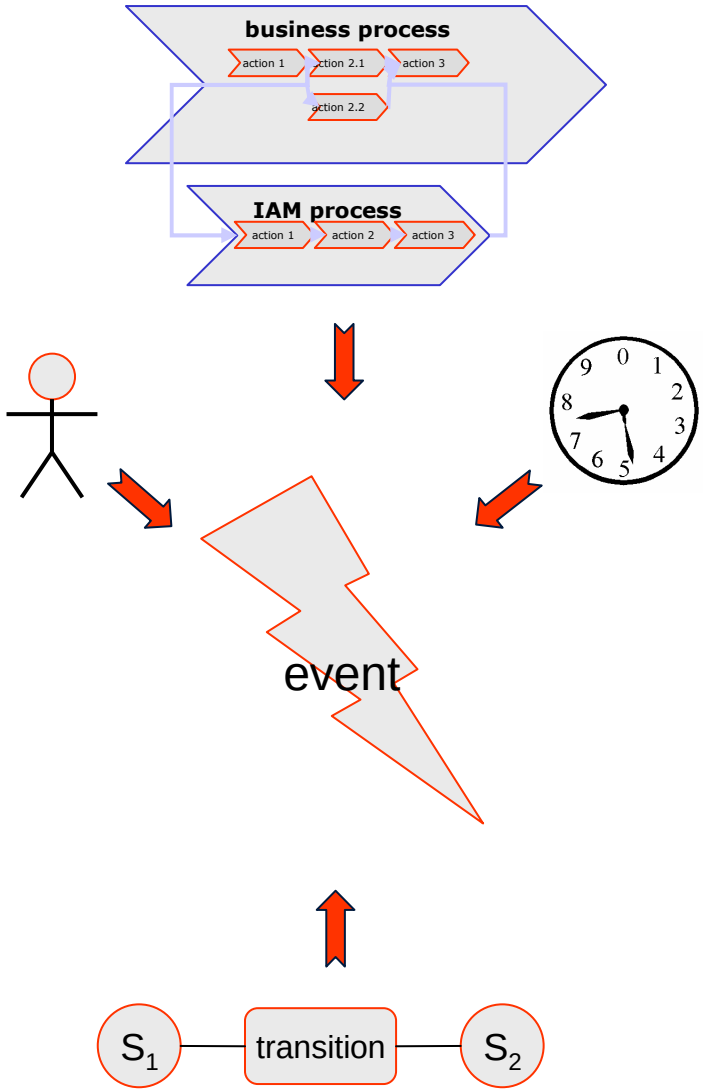
- In workflows subjects (actors) act on objects
- Subject may be owners or a custodian
- Owners are responsible
- custodians act on behalf of owners
- Owners delegate to custodians
- Subject act or react
- Their action triggers an event
- Reactions often are approvals

Every object has an owner

- Each object as one **owner**
- The owner is **responsible** for the object
- The owner may delegate object management to a **custodian**.
- The owner may temporarily **transfer** ownership (full responsibility) to delegate.
- Owners **differ** considerably from one organisation to another
- This apparent complexity is a result of **customising** a simple model



events

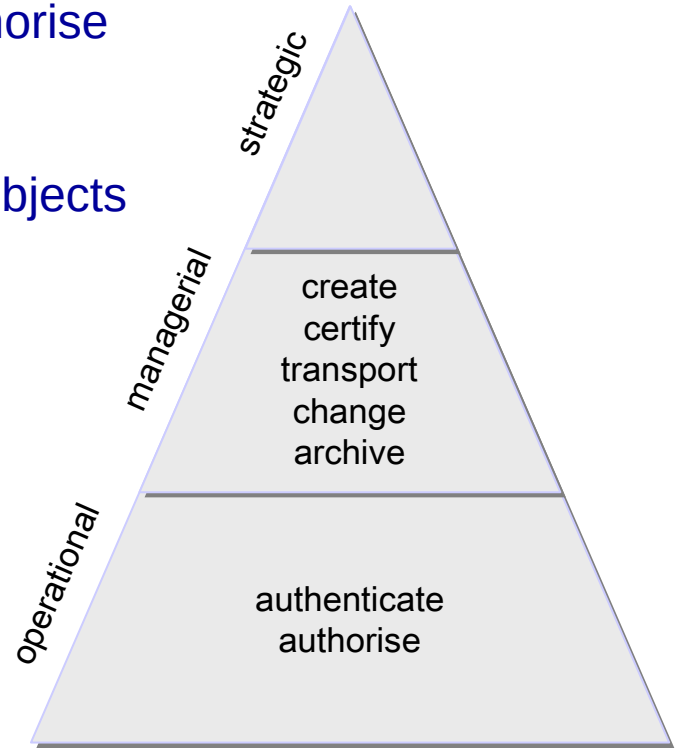


- Processes are triggered by events.
- There are events …
  - fired by embedding business processes.
  - created by an subject
  - triggered by time
  - fired by state transitions

Processes of the Identity Management

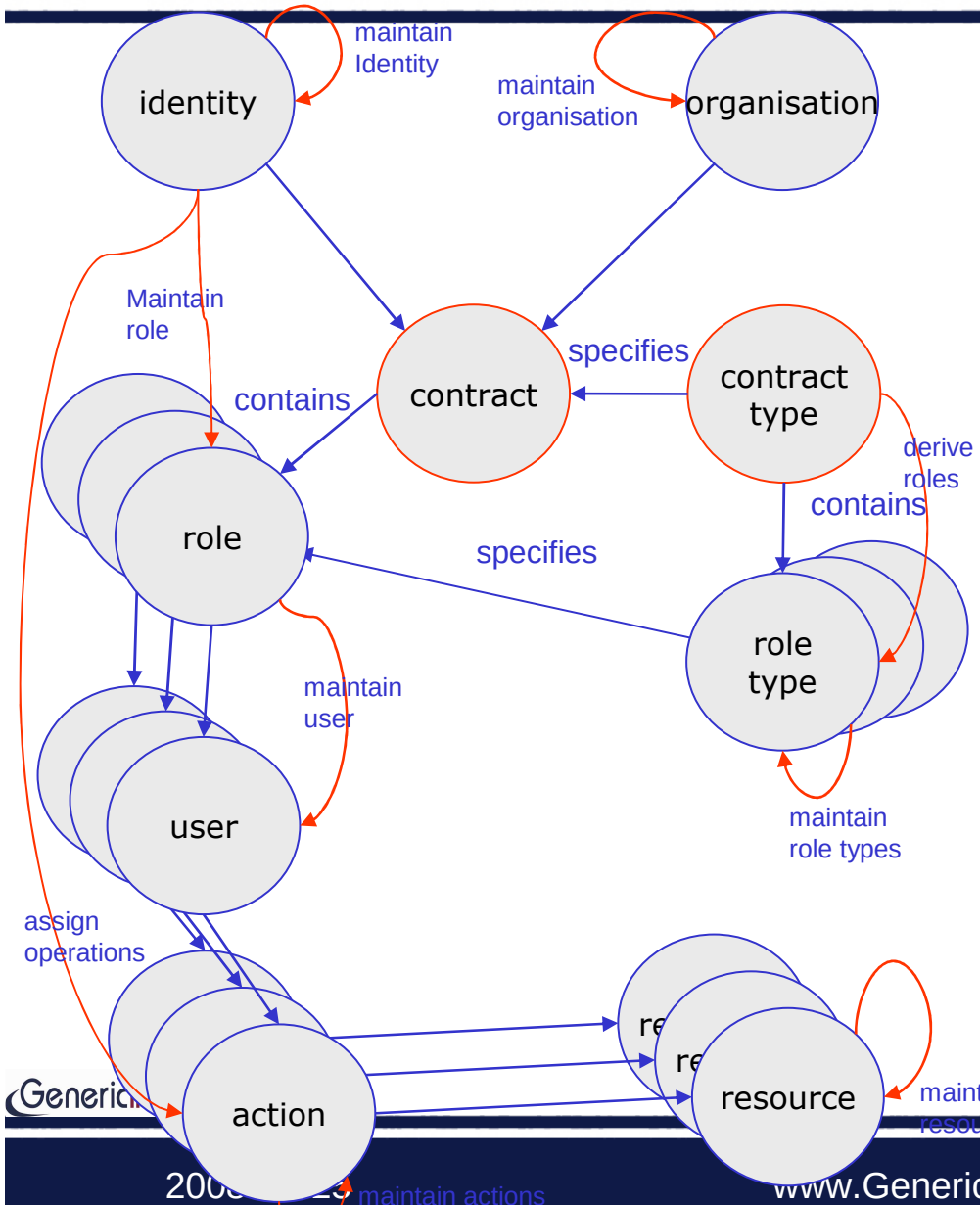
The Processes of the Identity Management may be grouped ...

- into operational, managerial and change
  - operational: identify, authenticate and authorise
  - managerial: administer digital Identities
  - Change: changing the implementation of objects
- into essential and physical
  - essential: administer and use
  - physical: integrate, transport, transform and “provision”
- By the leading objects, they act on



➔ each classification has its specific value.

elementary actions – changes on objects



The Identities role in an organisation performs actions on resources

- The Processes consist of >= 1actions.
- They are triggered by an event.
- They lead to a meaningful result to a subject.
- Process types (the class or definition) and process instantiations (incarnation, actual).
- Operational processes and managerial processes.
- Operational processes: *identification*, *authentication* and *authorisation*.
- The managerial:
  - *administrative processes*,
  - *audit processes* and
  - *change processes*.
- The administrative processes represent the “lions share” of all IAM processes.
- Its most prominent representative is the “*request & approval process*”.
- defines the relationship
  - a role defines incarnation details
  - “the contract is expressed by several roles”

Deriving elementary actions is an obvious process

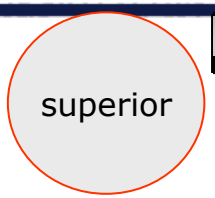
- Each object in the big picture needs a maintenance process
- Maintenance covers CRUD (create, read update, delete) and assignment

object	process	class	level	event	result
owner	maintain owner	administrative	essential	new owner, oor change	mantained owner
rule	maintain rule	administrative	essential	new or chaned policy	maintained rule
user	activate / deactivate user	administrative	essential	role change	(de-) activated user
operation	report operations on object for owned identities	audit	essential	request	report
policy	maintain authentication policy	change	essential	new or changed authentica	maintained authentica
policy	maintain federation policy	change	essential	new or changed federation	maintained federation
role	maintain role	change	essential	request	maintaned role
role type	maintain role type	change	essential	request, new system, new	maintained role type
system	maintain system assigments	change	essential	system change, policy char	maintained system as
system	maintain system	change	essential	new or changed system	maintained system
identity	autenticate identity	operational	essential	request	autenticated, rejected
identity	maintain role to identity assignment	operational	essential	request, contract change	maintained role type
identity	notify identity on request status	operational	essential	request fulfilled / rejected	subject notified
identity	maintain identity	operational	essential	new or changed identity	maintained identity
operation	maintain operation on object	operational	essential	new or changed object, ope	maintained operation
operation	reconciliate operations on objects	audit	physical	request, time	exception report
identity	federate identities	operational	essential	new or changed identity	federated identitties
system	login to system	operational	physical	login request	user logged in
system	(de-) provision operation on objects	operational	physical	request fulfilled / rejected	(de-) provisioned oper

Caveats
some not so obvious cases

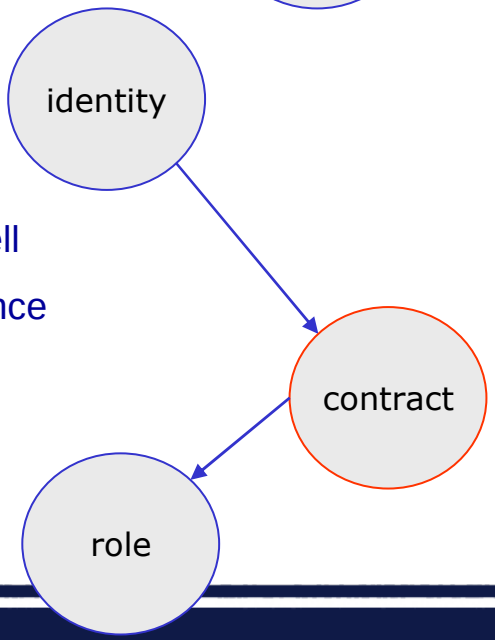
Who is the superior depends on the context

- To determine the superior it is important to ask …
  - For which task, process or project does the individual work
  - Who allocates the individuals working time there
- For different tasks, processes or projects the superior might be different
- The superior might be a project leader



Change role is more complex than it seems

- When an individuals role in an organisation changes
  - Often the old role need to be active for a defined period
  - Sometimes the new role needs to be active beforehand as well
  - Generally some actions on objects must be activated in advance
  - Generally some actions on objects need to phase out later (e.g. access to old mailbox)
  - Occasionally transfer actions must be inserted (e.g. migrate mailbox)



Applying the generic essential model
stepping from genericity down to earth

The model enables deriving a complete set of elementary actions.

- In order not to underestimate the number of ‘trivial’ maintenance processes.
- The majority of processes consists of only one elementary action.
- A minority of processes is more complex.
- Customising the model to a specific situation covers …
  - Naming all resources
  - listing all actions
  - Fleshing out the organisation
  - Naming the specific owners
  - Stating policies
    - e.g. for pre-approved role assignments
- Adding physical actions
  - transport, translate, transform protocol, check input quality, complement data, …
- Linking to embedding business processes

Applying the generic essential model
Adding physical actions

- Transport
- Translation
- Check controls (for GRC)
  - Preventive
    - avoiding an unwanted situation.
    - Policies and procedures
      - example.: change Management.
      - "all changes will go through a formal change management process“
    - Because 80% of computer errors are related to human error
    - Formally reviewed, tested and a rollback plan been developed if the change failed.
    - Monitoring can be used preventively.
    - preventive controls are not sufficient.
  - Detective
    - alert us when an unwanted event transpires.
    - as soon as possible, but it is after-the-fact.
  - Corrective
    - restoring the system to its expected state.
    - Having backup configuration files or hard drive images that can be reloaded to restore the state are both good examples.
    - predicated on an organization having effective change control and configuration management.

What is GRC?

Governance, Risk, and Compliance

- **Governance**.
  The culture, policies, processes, laws, and institutions that define the structure by which companies are directed and managed. Corporate governance includes the relationships among stakeholders and the goals for which the corporation is governed.

- **Risk**.
  The effect of uncertainty on business objectives; risk management is the coordinated actions to direct and control an organization to realize opportunities while managing adverse events.

- **Compliance**.
  The act of adhering to, and demonstrating adherence to, external laws and regulations as well as corporate policies and procedures; compliance management is the coordinated actions to stay within internally and externally mandated boundaries.

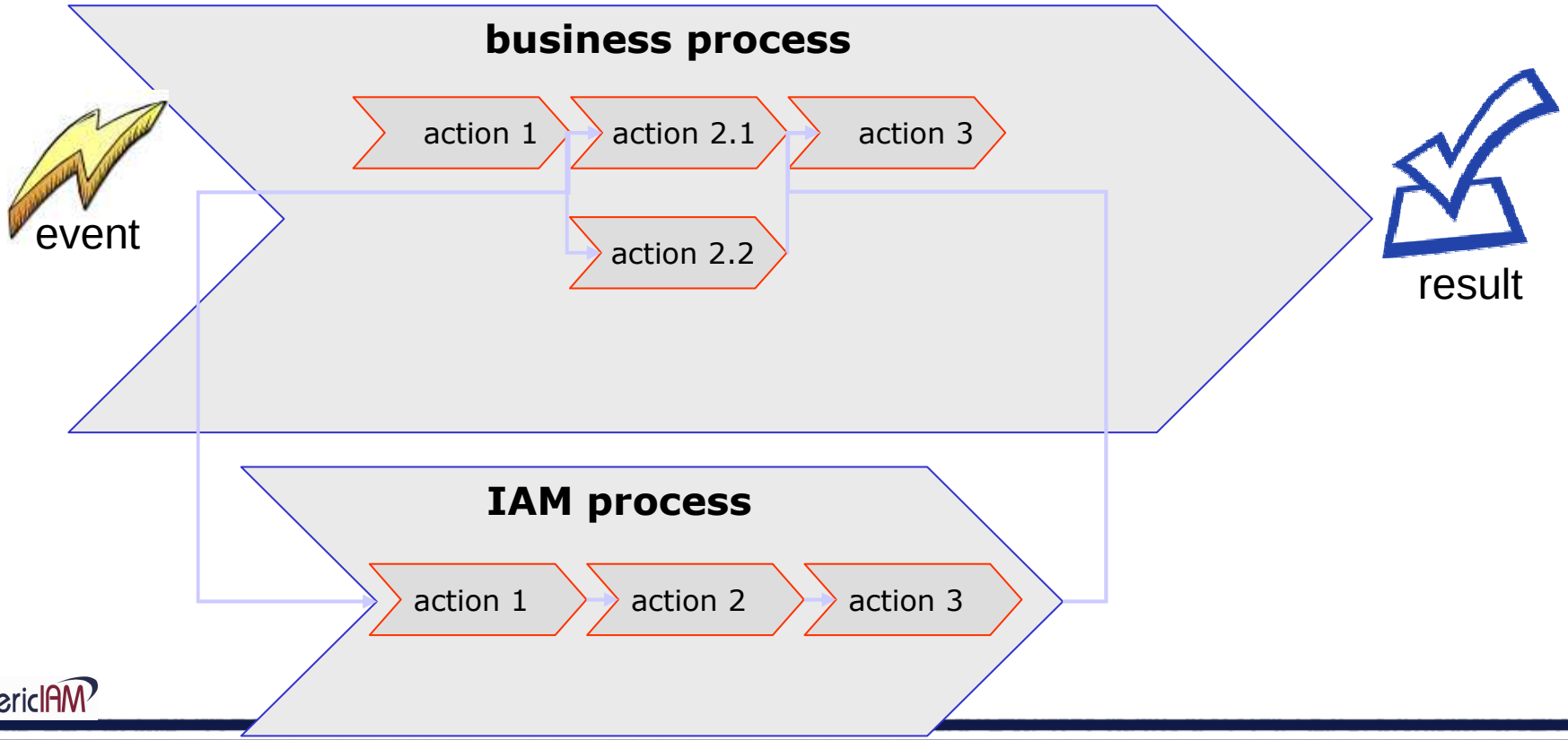
Typical GRC-controls
Evidence on users and privileges

- **Current User** accounts and privileges
  - Accounts and privileges applied for.
  - Report per user or per requester
  - Reports for business superiors
  - User accounts und privileges of users per organisational unit
- **Target system** specific Reports
  - Available base roles per target system
  - User accounts und privileges per target system.
- **Access** Reports
  - Who has accessed a system within a period?
  - Which systems has a user accessed within a period?
- **Reconciliation** with target systems
- Privileges **via roles** versus **direct** assignment.
- **Workflow** Reports
  - Weekly report on tasks that were not completed the previous week
  - Weekly report on provisioning actions by department, location, resource type, etc.
  - Were all of the accounts created on time? - How many times did we act late this month?
- **Licence tracking**
  - By user − which systems were not accessed by a particular user within a given period?
  - By system − which users didn't access a particular system within a given period?

Applying the generic essential model
Linking to embedding business processes

IAM-processes are triggered by business needs in a defined context.

- IAM-processes often are not stand-alone
- They are mostly part of embedding business processes
- This relationship has to be maintained seamlessly



Business processes
embedding business processes should be in place before hand.

typical business driven processes
which require the invocation specific IAM-sub-processes are:

- On boarding processes
  - hire employee
  - acquire customer
  - contract partner
  - contract temporary staff
- Off boarding processes
  - terminate employee
  - terminate customer
  - terminate partner
  - terminate temporary staff
- Change processes
  - change employees job description
  - de-activate contract (employee, customer, partner, temp. Staff)
  - re-activate contract (employee, customer, partner, temp. Staff)
  - changes identity attributes (marriage, name-, sex-change, …)

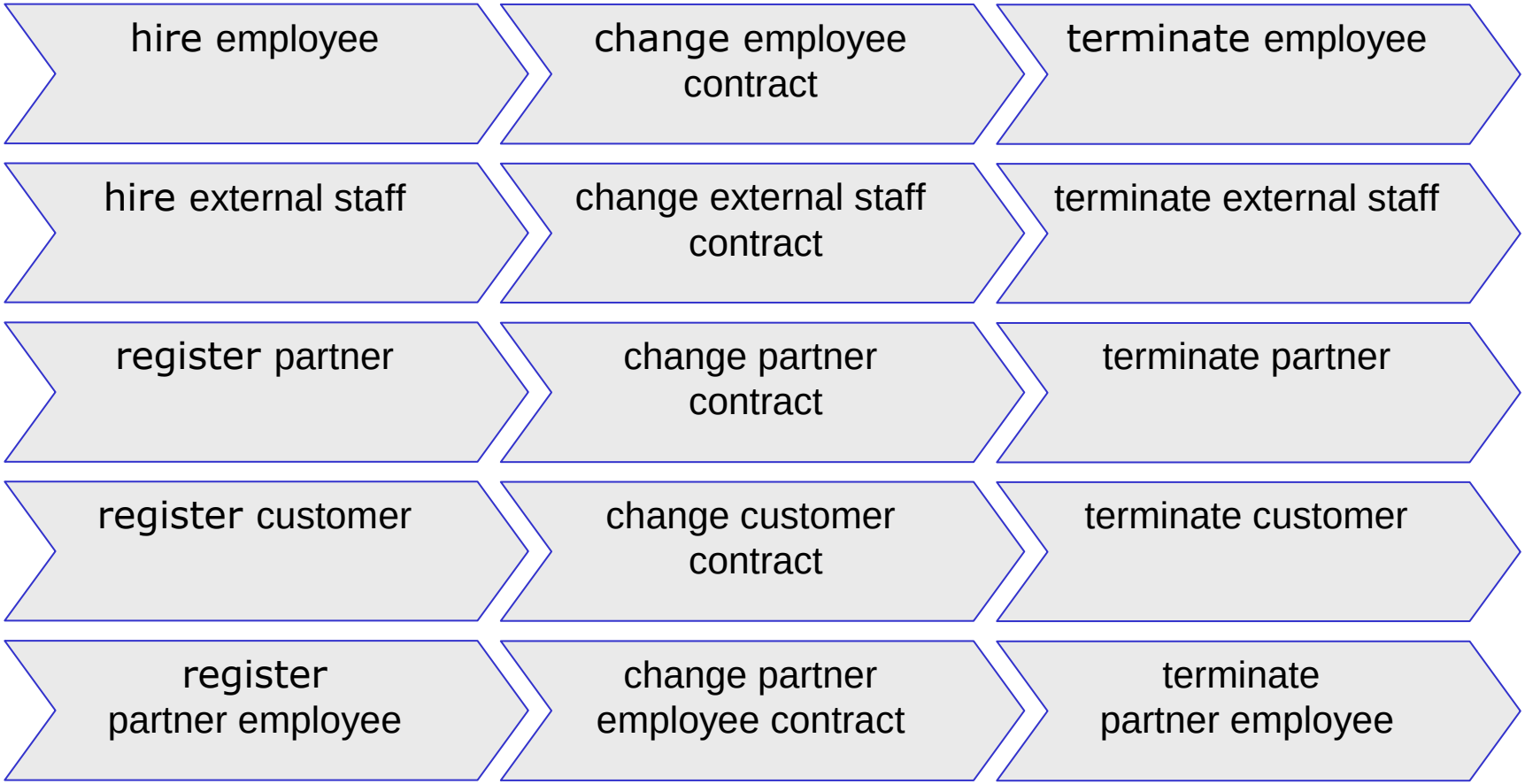
For these processes in an early stage it has to be determined, whether this individual is known to the corporation already (a digital identity exists).

Terminating means flagging as arched. Records are deleted when they are no longer of value or when data protection rules require deletion.

Change process tend to be complicated. Rarely there is a clear cut. Often an overlap of responsibilities requires phasing-in & phasing-out periods. Temporary responsibilities like project roles may be assigned additionally for a limited period.

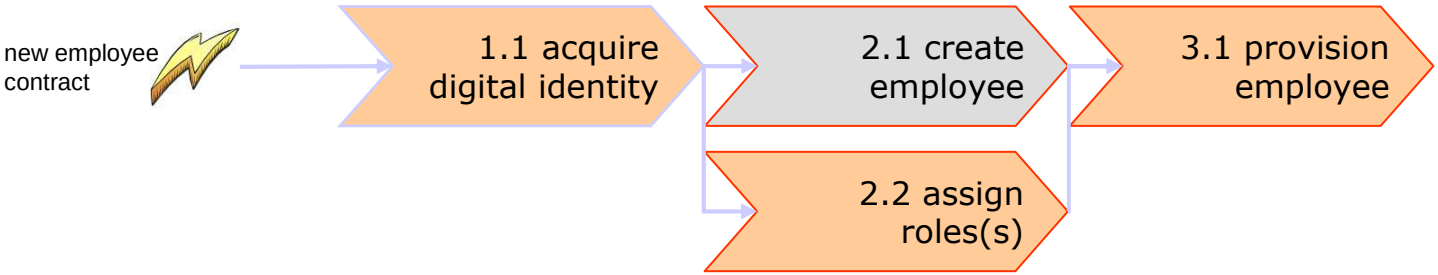
fundamental business process groups
and their variation by type of digital identity

- The 3 fundamental business process groups on-boarding, off-boarding & change processes split of by type of digital identity.



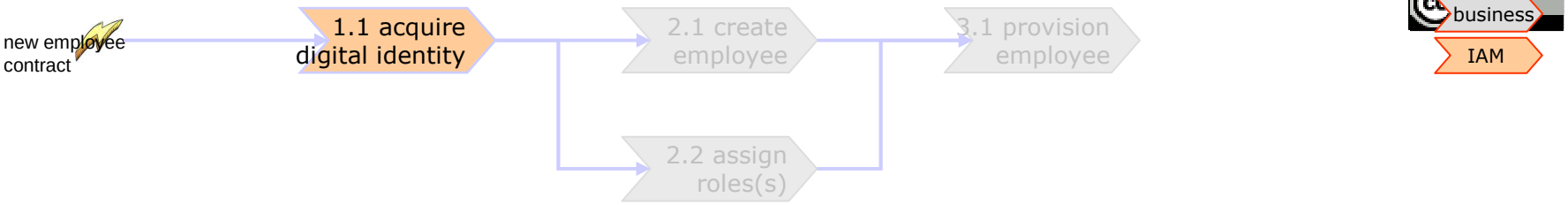
Business process “hire employee”
IAM- and business actions are interlinked

- The existing process “hire employee” …
  - needs to be split up into business- and IAM-actions
  - has to position the assignment of a digital identity up-front
  - must deliver 3 more attributes
  - Identifies employees (like external staff) by …
    - 1st assigned full name (birth name)
    - date of birth
    - location of birth



business
IAM

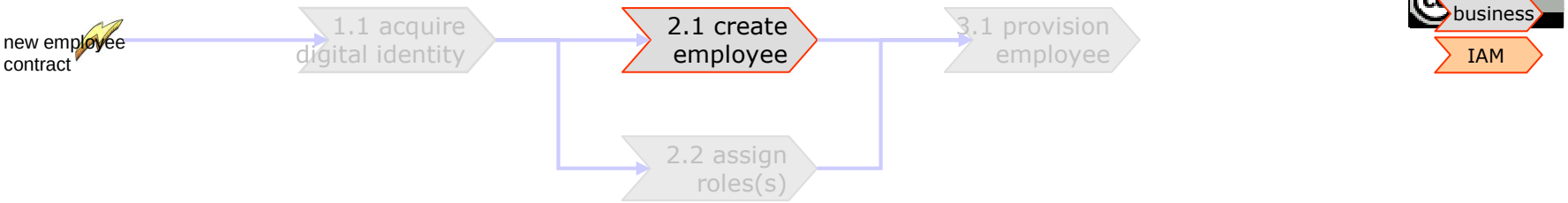
Business process “hire employee”
1.1 acquire digital identity



- The action “acquire digital identity” …
  - 1st looks up a central corporate directory for the employee
    - If it does not exist it will be created
    - If exits it will be reactivated
    - Search must include employee-, partner- & customer-identities
    - Duplicate entries must be detected during search
    - Non-obvious cases must be offered for manual selection
  - requires all identifying attributes to be provided at this point in time.
  - returns a unique ID, which ..
    - is non-disclosing
    - doesn’t carry any additional information
  - returns address data
  - returns it’s result in “real-time” (< 3 sec.)
- The “corporate identity manager” is responsible for this action.

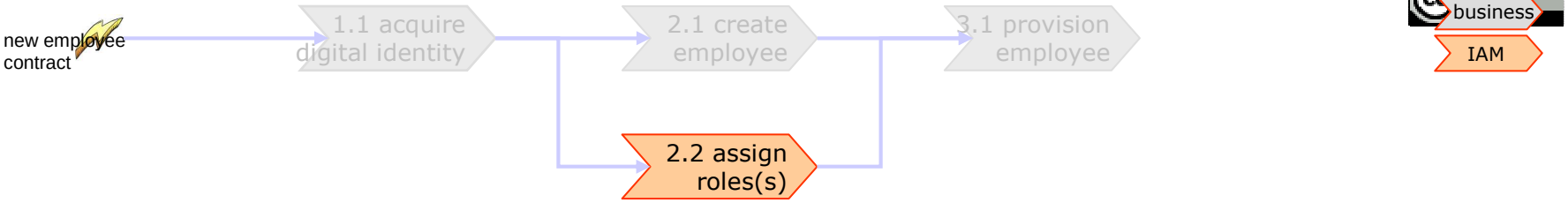
Business process “hire employee”

2.1 create employee



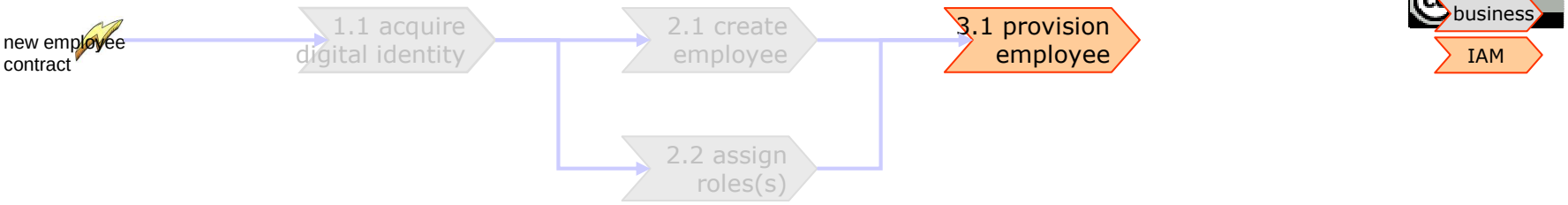
- The action “create employee” …
  - creates the personnel number
  - creates the personnel record in the HR-System
  - stores the ID delivered by action 1.1
  - in contrast to the current implementation stores the additional attributes …
    - contract type (e.g. “full time employee)
    - region (e.g. ASPAC)
    - location (e.g. Singapore)
  - may stay untouched besides these changes.
  - is a manual action and returns it’s result within one working day.
- The “human resource manager” is responsible for this action.

Business process “hire employee”
2.2 assign role



- The action “assign role” …
  - assigns a predefined “business role” to the employee
    - business roles are defined in terms of business entitlements
    - Role assignment doesn’t require detailed system knowledge
  - alerts the “corporate identity manager” in case of missing / ill-defined roles.
  - does not assign individual privileges
  - assigns roles for 85 % of all employees
  - may be skipped in 1st implementation increment due to project design.
  - is a manual action and returns it’s result within one working day.
- The “corporate identity manager” is responsible for this action.

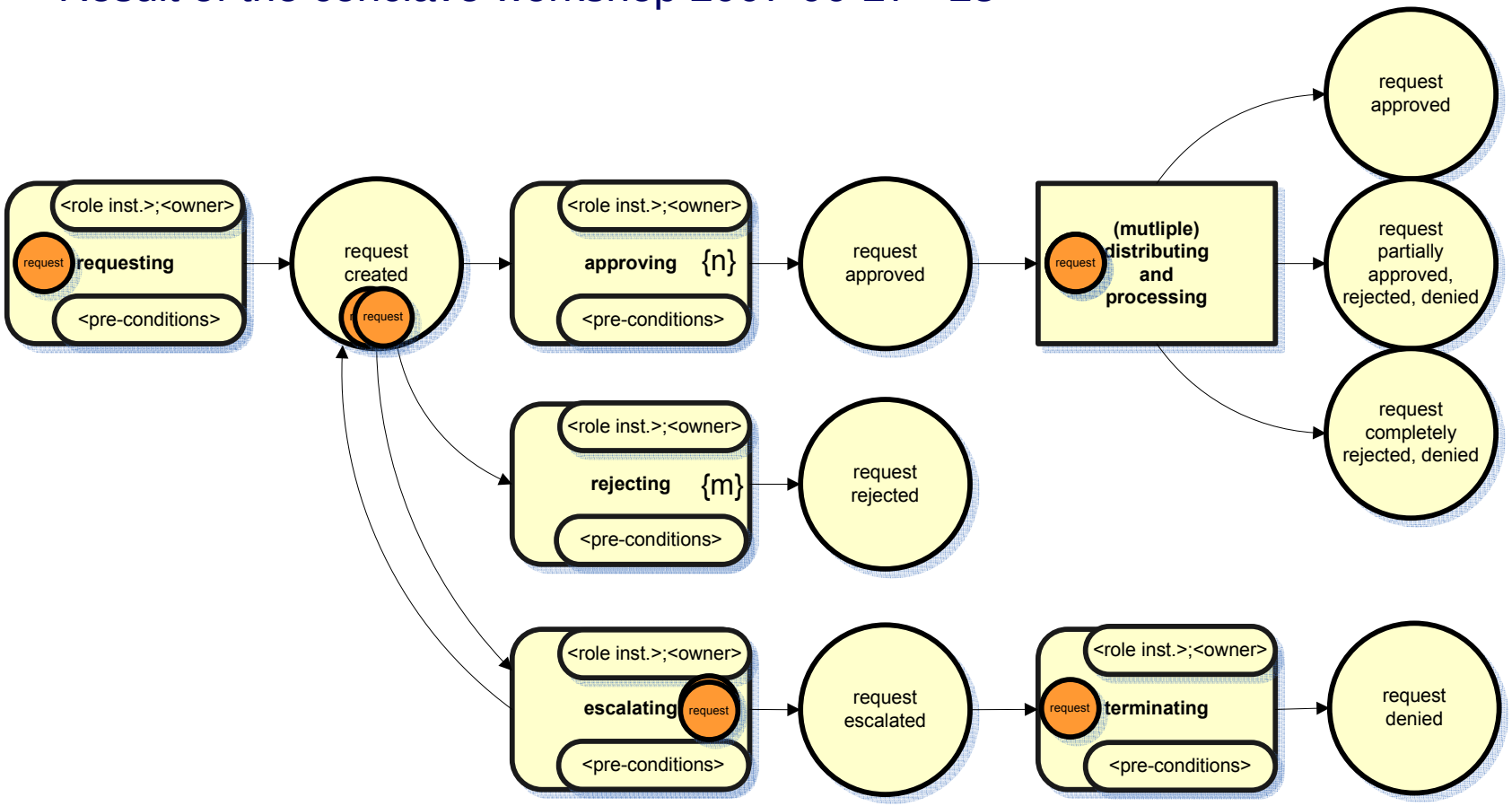
Business process “hire employee”
3.1 provision employee



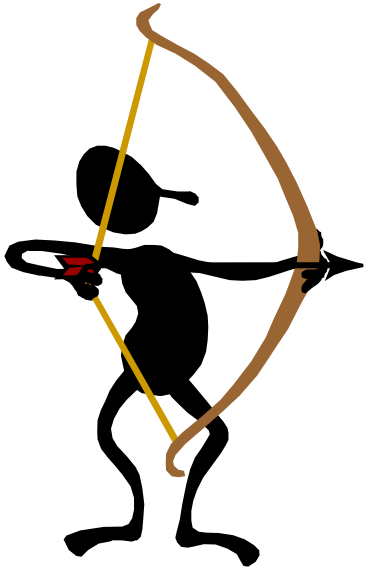
- The action “provision employee” …
  - provisions all the ID + all current attributes +
    - address data
    - contract type (e.g. “full time employee”)
    - region (e.g. “ASPAC”)
    - location (e.g. “Singapore”)
  - will later be complemented by
    - Translates roles to elementary privileges according to corporate policies.
    - provisioning {1..n} privileges
  - provisions the ISP in “real-time” (< 5 min.)
    - ISP needs adaptations to accept these data
    - This action is subject to change in later project phases
- The “corporate identity manager” is responsible for this action.

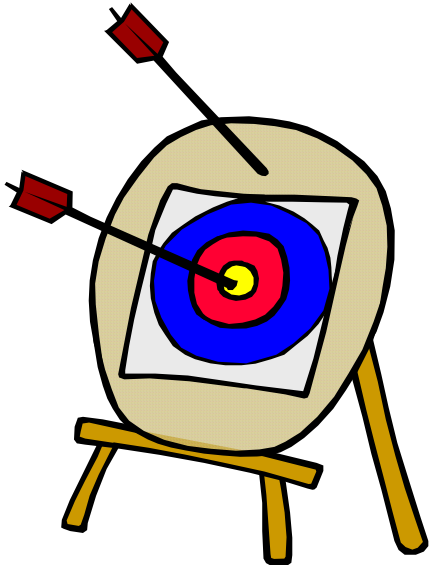
Approve request
generic process example using petri nets

- Result of the conclave workshop 2007-06-27 - 28



The end ...





Thank you very much for your attention!

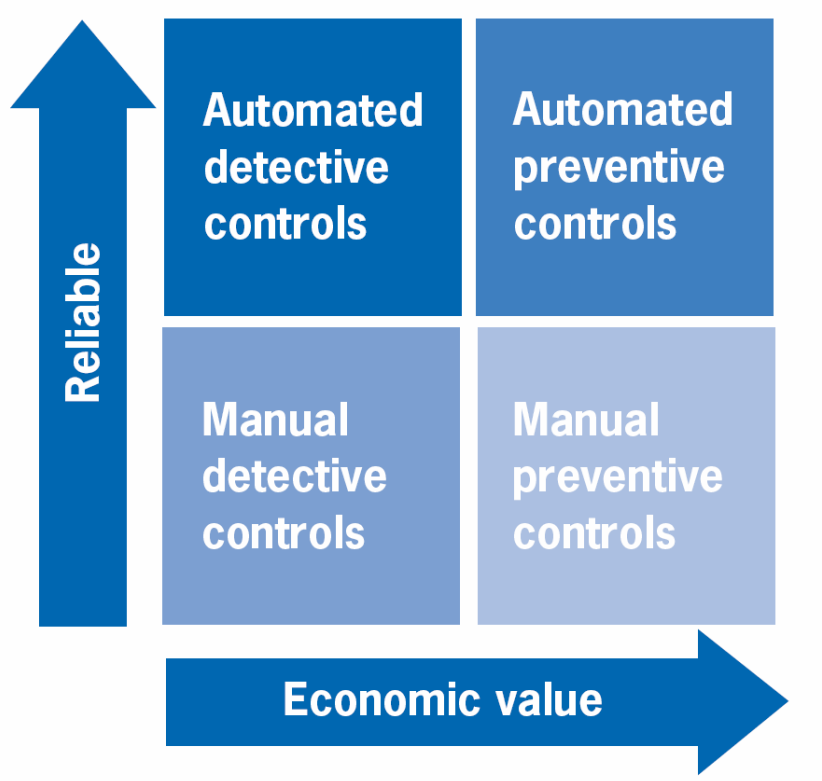
In case of any questions:
horst.walther@nifis.org,
skype: HoWa01

VoIP: +40 40 414314453

Attention

Backup slides

GRC controls
detective vs. preventive – manual vs. automated



- controls can be classified as preventive or detective.
  - They either prevent errors before they occur or
  - They detect errors after they have occurred but in time to correct them before they do real damage.
- Both types of controls are important.
- preventive controls are preferred to detective ones.
  - detective controls act after an error has occurred, this means that the undetected errors go on to have a negative impact on the business.
  - preventing errors is cheaper than to detecting and fixing them.
- Preventive controls generally have a higher “economic value” to an organization.
- detective controls may enable an acceptable control environment to meet minimal requirements.
- To improve the bottom line a proper balance of detective and preventive controls is necessary.

GRC controls
examples in 4 categories

Examples of **detective** and **preventive** controls

- Detective Controls are designed to identify an error or exception after it has occurred. Examples include:
  - Exception reports
  - Reconciliations
  - Reviews of operating performance
  - Periodic inventories
- Preventive Controls focus on preventing errors or exceptions. Examples include:
  - Use of checklists
  - Training
  - Proper segregation of duties
  - Authorization levels/approvals

Examples of **manual** and **automated** controls

- Manual Controls operate through human intervention. They are the most flexible but are also subject to human error. Examples include:
  - Comparison of amounts entered to source documents
  - Signatures/initials noted on completed documents
  - Budget-to-actual reviews
  - Re-performance of computations
- Automated Controls operate through and within information technology systems. They function systematically and work with a high degree of consistency. Examples include:
  - System access controls
  - Data entry requirements prior to transaction processing
  - Automated balancing and reconciliations
  - Automated flags that identify possible invalid or duplicate entries/data

Identity and Access Management Terminology

- **Access management.** Processes and technologies for controlling and monitoring access to resources consistent with governing policies. Includes authentication, authorization, trust, and security auditing.
- **Authentication**. A process that checks the credentials of a security principal against values in an identity store. Authentication protocols such as Kerberos version 5, Secure Sockets Layer (SSL), NTLM, and digest authentication protect the authentication process and prevent the interception of credentials.
- **Authorization**. The process of resolving a user's entitlements with the permissions configured on a resource in order to control access. Authorization in the Windows operating system involves access control lists (ACLs) on files, folders, shares, and directory objects. Applications such as SQL Server, SharePoint® Portal Server, and Exchange Server implement access control mechanisms on resources they manage. Application developers can implement role-based access control using Windows Authorization Manager or ASP.NET roles.
- **Credential**. Typically a piece of information related to or derived from a secret that a digital identity possesses, although secrets are not involved in all cases. Examples of credentials include passwords, X.509 certificates, and biometric information.
- **Digital identity**. The unique identifier and descriptive attributes of a person, group, device, or service. Examples include user or computer accounts in Active Directory, e-mail accounts in Microsoft Exchange Server 2003, user entries in a database table, and logon credentials for a custom application.
- **Entitlement**. A set of attributes that specify the access rights and privileges of an authenticated security principal. For example, Windows security groups and access rights are entitlements.
- **Federation**. A special kind of trust relationship between distinct organizations established beyond internal network boundaries.
- **Identity integration services**. Services that aggregate, synchronize, and enable central provisioning and deprovisioning of identity information across multiple connected identity stores. MIIS 2003 SP1 and the Identity Integration Feature Pack 1a (IIFP) for Active Directory provide identity integration services.
- **Identity life-cycle management**. The processes and technologies that keep digital identities current and consistent with governing policies. Identity life-cycle management includes identity synchronization, provisioning, deprovisioning, and the ongoing management of user attributes, credentials, and entitlements.
- **Identity store**. A repository that contains digital identities. Identity stores are usually some form of directory or database managed and accessed through a provider such as Active Directory or Microsoft SQL Server. Identity stores can be centralized, for example on a mainframe computer, or distributed; Active Directory is an example of a distributed identity store. They generally have well-defined schemas for what information can be stored and in what form it can be recorded. They usually incorporate some form of encryption or hashing algorithm to protect both the store and components of the digital identity, such as credentials. Older and custom identity stores may not have such strict security mechanisms and may store passwords in plaintext (with no encryption).
- **Identity synchronization**. The process of ensuring that multiple identity stores contain consistent data for a given digital identity. This process can be achieved using programmatic methods such as scripts or through a dedicated product such as MIIS 2003 SP1.
- **Provisioning**. The process of adding identities to an identity store and establishing initial credentials and entitlements for them. Deprovisioning works in the opposite manner, resulting in the deletion or deactivation of an identity. Provisioning and deprovisioning typically work with identity integration services to propagate additions, deletions, and deactivations to connected identity stores.
- **Security auditing**. A process that logs and summarizes significant authentication and authorization events and changes to identity objects. Organizations will differ in their definition of significant events. Security audit records can be written to the Windows Security Event Log.
- **Security principal**. A digital identity with one or more credentials that can be authenticated and authorized to interact with the network.
- **Trust**. A state that describes the agreements between different parties and systems for sharing identity information. A trust is typically used to extend access to resources in a controlled manner while eliminating the administration that would otherwise be incurred to manage the security principals of the other party. Trust mechanisms include cross-forest trusts in Windows Server 2003 and trusts between realms using the Kerberos version 5 authentication protocol.

Roles, tasks, and operations
according to Microsoft

- A **role** …
  is a set of permissions that a user must have to do a job.
  - Well-designed roles should correspond to a job category or responsibility (for example, receptionist, hiring manager, or archivist) and be named accordingly.
- A **task** …
  is a collection of operations, and sometimes other tasks.
  - Well-designed tasks are inclusive enough to represent work items that are recognizable (for example, "change password" or "submit expense").
- An **operation** …
  is a set of permissions that you associate with system-level or API-level security procedures like WriteAttributes or ReadAttributes.
  - You use operations as building blocks for tasks.

- **Role definitions**
  The role definitions that are appropriate depends on the structure and goals of your organization. Roles support inheritance from other roles. To define a role, you specify a non-arbitrary name, a friendly description, and some lower-level tasks, roles, and operations that are part of it. This provides a mechanism for role inheritance. For example, a Helpdesk role might include a Product Support role.
- **Role assignments**
  A role assignment is a virtual container for application groups whose members are authorized for the role. A role assignment is based on a single role definition, and a single role definition can be the basis of many role assignments.
- **Task definitions**
  A task definition is smaller than a role definition and can be used to define roles and other tasks.
  You associate tasks with roles in an intuitive way. For example, the Recruiter role might include the Interview task. Tasks, like roles, are defined in a way that is appropriate to the organization. To define a task, you specify a name, a friendly description, and some lower-level tasks and operations that are part of it.
- **Operation definitions**
  Operations are small computer-level actions that are used to define tasks and are not relevant to an administrator. You define operations only in developer mode.