# Rules for Identity and Access Control

## IFIP/FIDIS Summerschool 2007

TNO | Knowledge for business

Rieks Joosten

August 2007

# Dutch Organisation for Applied Scientific Research

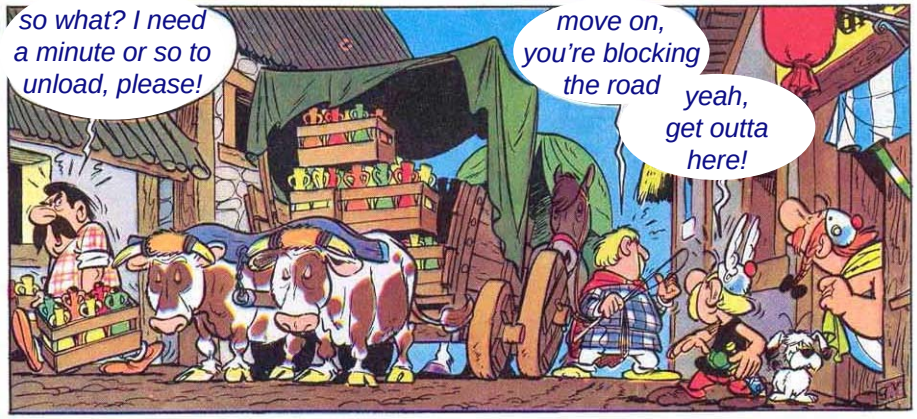
- **Dutch R&D and consulting organisation**
  - Founded by law to help industry access and apply academic knowledge
  - Independent source of knowledge & innovation
  - Some 5000 employees, active in many fields – construction, healthcare, nutricion, space…
  - ICT area (350 people, EUR 40M turnover )

- **TNO ICT Security department**
  - 25 full-time security specialists
  - Broad range of expertise
    - Telecommunications security
    - IT & Network Security
    - Information Security Management
    - Cryptography & PKI
  - Consulting, contract R&D for businesses and government

## IAM within IT is like traffic:

you need RULES
(and COMPLIANCE)

varying across different
automated CONTEXTS





*Images taken from comic book "Asterix"*

# Observations

- The meaning of terms such as ‘Identity’, ‘Identifier’, ‘Identification’, ‘Identify’ and others, when used in human contexts (i.e.: not within a computer application), is highly ambiguous:
  - different people mean different things at a given point in time
  - depending on the time of day, different meanings are attributed by the same person

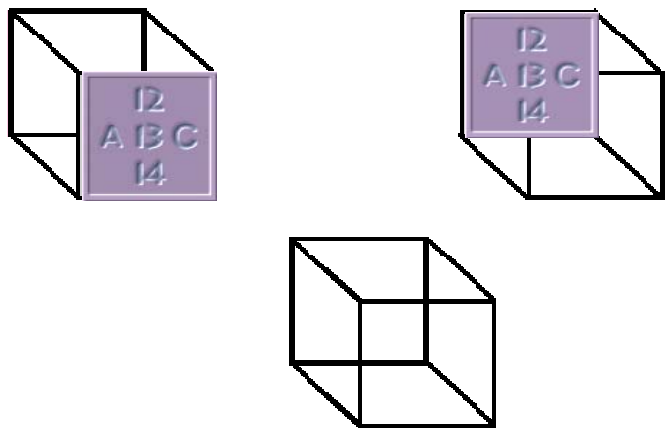
- Houston, we have a problem
  when unambiguous meaning is necessary, e.g. for specifying automated systems, especially and more so as the scope of such systems grows larger.

- Don’t blame the systems architects (designers) for failing to build such systems
  - It is not his task to define meaning, or to create a committed consensus between various contexts. He wasn’t educated to do such things.
  - His task is to build systems, not to define their use (or what it shouldn’t do)
  - We need to help this poor thing so that he no longer needs to worry about two many things at the same time (note that you start making mistakes)

- Will we be able to solve identity and privacy issues before the aforementioned issues are largely resolved?

# Human Contexts and Automated Contexts

- If notions such as ‘Identity’, ‘Identifier’, ‘Identification’, ‘Identify’ and others are to be used in automated contexts (by computer (applications)), the business must define appropriate policy (rules, principles, constraints) that constrain the use thereof within automated systems
- The same is true of any other notion that the business considers relevant
- Doing so requires every concept (notion, term) to be
  - unambiguous, consistent and coherent w.r.t. its intended meaning (i.e.: can be expressed formally)
  - business relevance
- People that do this should be competent in
  - things like Relation Algebra and Set theory
  - in automated contexts
  - as business consultants (human contexts)




**Be aware of and handle all these different contexts appropriately**

# I will be talking about

- the kind of IAM rules we might want

confined to **AUTOMATED** contexts

- suppose we have rules, then what?

- our experiences with this kind of work
- the impact this may have on businesses / governments
- future work

# An example set of IAM rules, for automated contexts, expressed in a Natural Language:

1. Every domain, i.e. a named set of responsibilities, has at least one domainmanager bearing all responsibilities.
2. Everything that happens within one session is the responsibility of precisely one domain.
3. Every session is of precisely one type.
4. Sessions of a given type may only run within a domain if there exists a valid approval within that domain for running this type of sessions.
5. A tokenadministration consists of entries, each of which is uniquely characterized by a token, the type of that token and the token's issuer.
6. Each entry in a tokenadministration has 1 userid.
7. Each entry in the tokenadministration has 1 domain that bears all responsibility for every use of the token.
8. Userid's associated with multiple tokenadministration entries must have the same responsible domain.
9. Logging into a session means providing a token, its tokentype and its issuer to that session.
10. A sessiontoken is a login-token where the provided token, tokentype and issuer identify an entry in the tokenadministration (authentication)
11. A sessionCoactor is the userid associated with a sessiontoken.
12. A sessionCodomain is the domain that is responsible for every use of a sessiontoken.
13. Every session shall have at most one sessionCoactor and one sessionCodomain at any time.
14. Whenever a token, tokentype and tokenissuer combination is presented in a session that already has or has had a sessiontoken, this token shall only become a sessiontoken if its associated userid is the sessionCoactor.
15. If a dataobject contains a list of Codomains, it shall only be accessed in a session if this session's sessionCodomain appears in said list.
16. If executing an action in a session implies that the sessiondomain is taking a risk, then a permission shall be required for executing this action
17. An action shall only be executed within a session if all permissions it requires, exist in that session.
18. A permission exists within a session if it is associated with a sessionrole.
19. A sessionrole for a session of a certain type is any role that (1) has been assigned to the sessionCoactor, and (2) has been defined as a role that may be activated for sessions of this type.
20. Roles shall only be assigned to existing userid's.
21. A token can only become a sessiontoken (i.e.: you may only login) in a session of a certain type if the userid associated with that token has been assigned at least one role that is relevant for sessions of that type.

# An example set of IAM rules, for automated contexts, expressed in a Natural Language:

1. Every domain, i.e. a named set of responsibilities, has at least one domainmanager bearing all responsibilities.
2. Everything that happens within one session is the responsibility of precisely one domain.
3. Every session is of precisely one type.
4. Sessions of a given type may only run within a domain if there exists a valid approval within that domain for running this type of sessions.

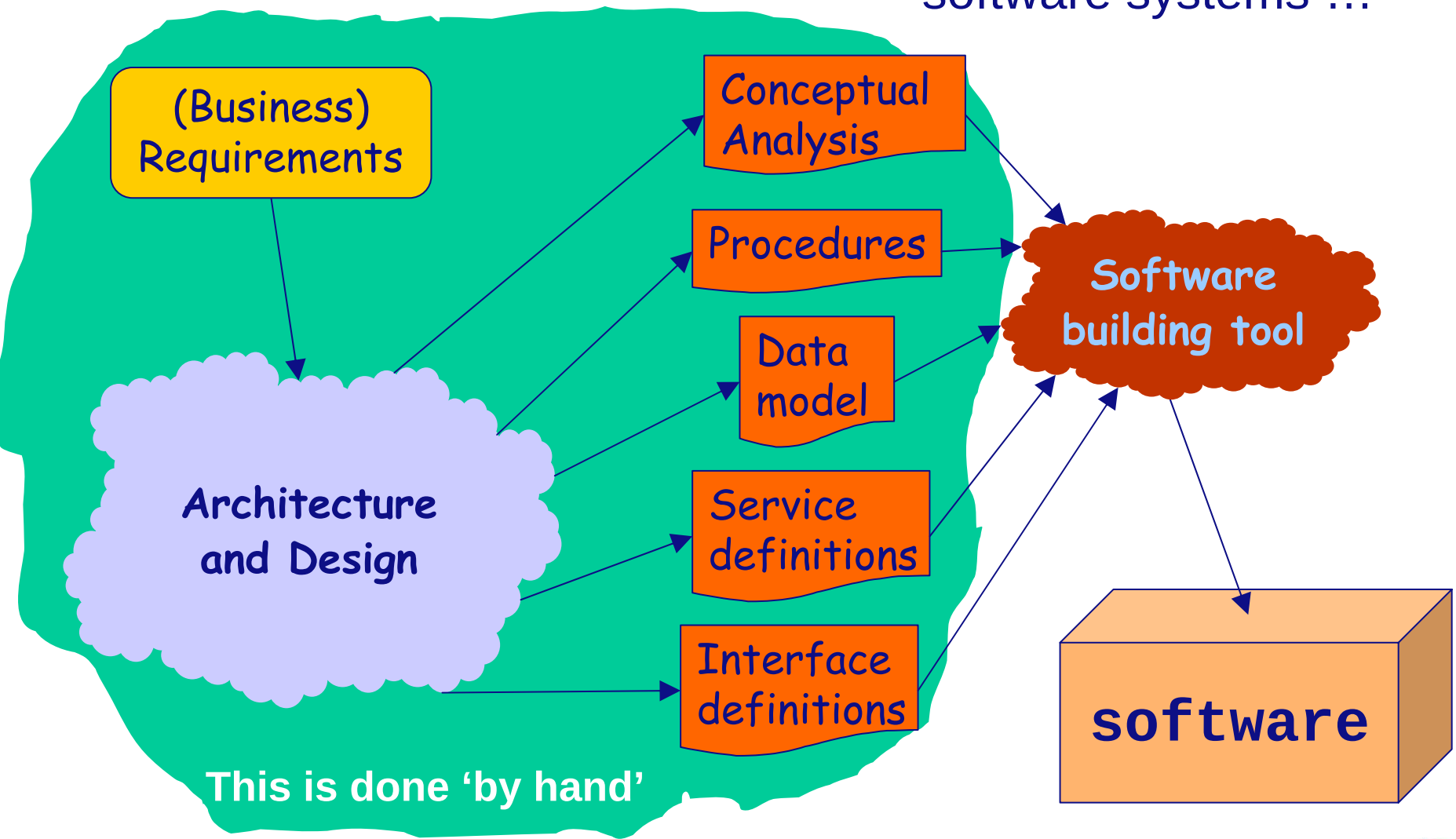
16. If executing an action in a session implies that the sessiondomain is taking a risk, then a permission shall be required for executing this action
17. An action shall only be executed within a session if all permissions it requires, exist in that session.
18. A permission exists within a session if it is associated with a sessionrole.
19. A sessionrole for a session of a certain type is any role that (1) has been assigned to the sessionCoactor, and (2) has been defined as a role that may be activated for sessions of this type.

# I will be talking about

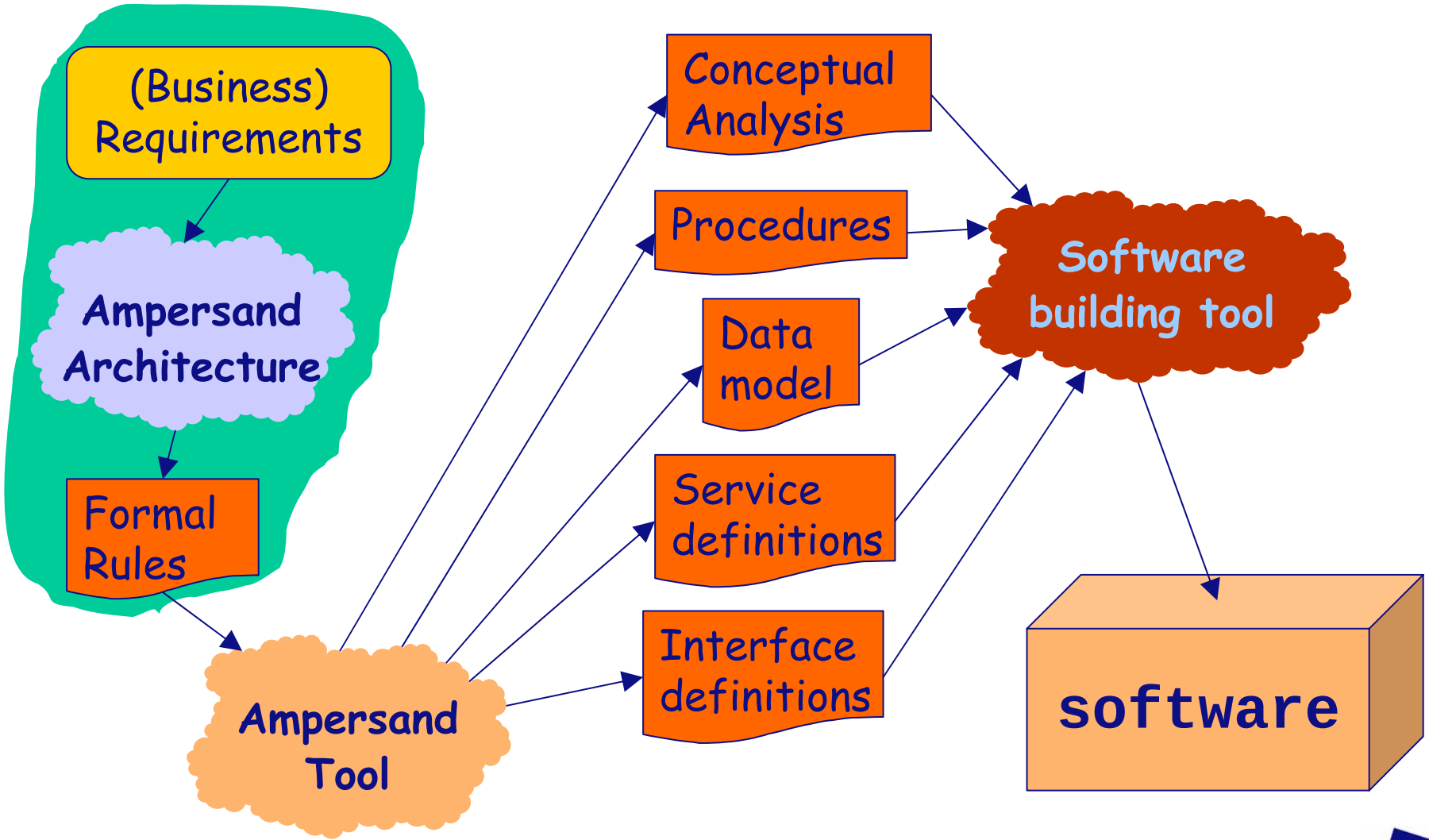
- the kind of IAM rules we might want
- suppose we have rules, then what?
- our experiences with this kind of work
- the impact this may have on businesses / governments
- future work

still, for **AUTOMATED** contexts

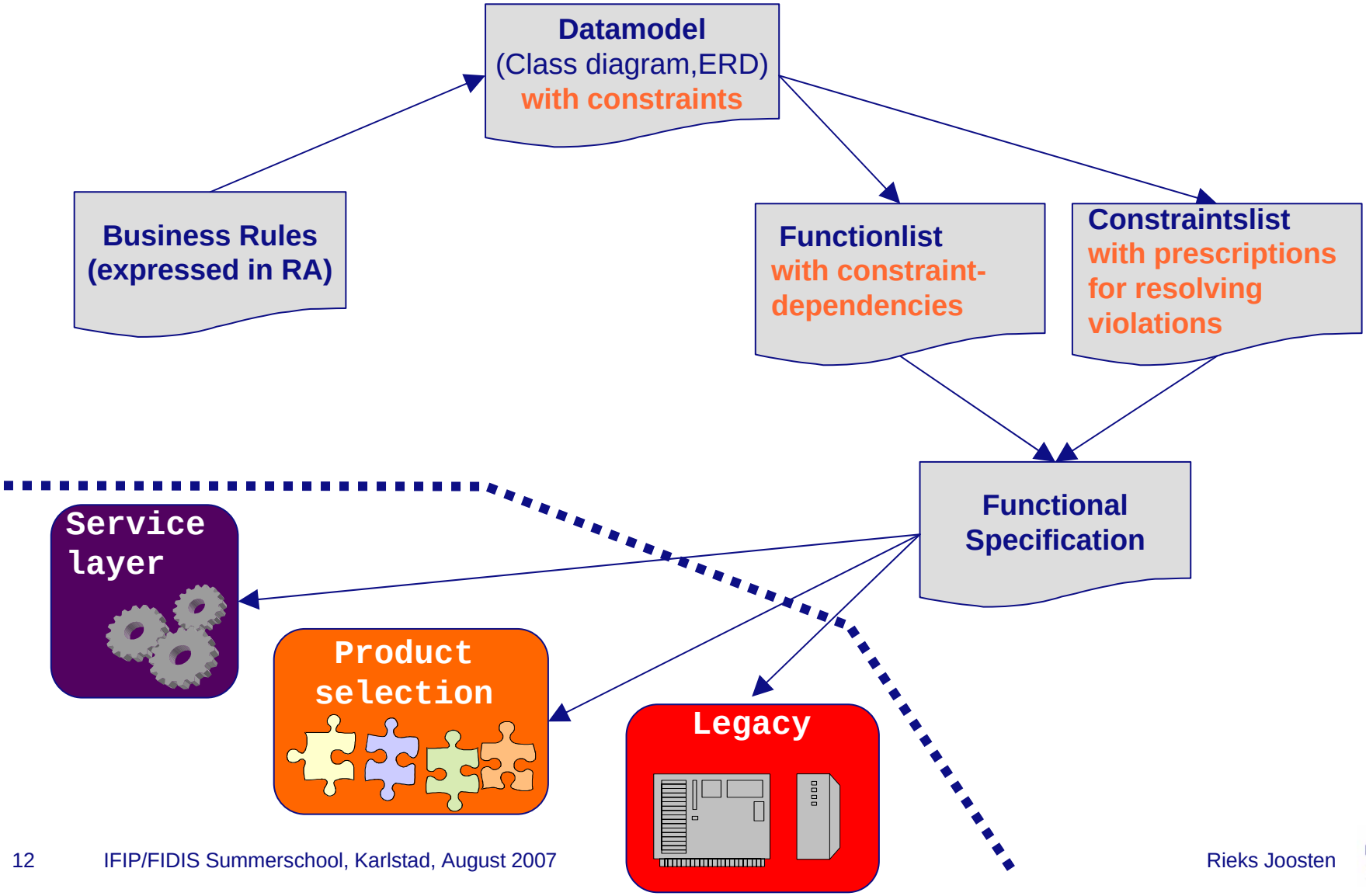
# A new way to enhance compliance and integrity of software systems …



# A new way to enhance compliance and integrity of software systems …



# Tooling (Courtesy OpenUniversiteitNederland)



# I will be talking about

- the kind of IAM rules we might want
- suppose we have rules, then what?

- our experiences with this kind of work
- the impact this may have on businesses / governments
- future work

# Experience / Results

- There's a demo!
  - with a (generated) IAM service layer (PHP, with a MySQL database)
  - we found it easy to build compliant (demo) applications
- Operationalizing IAM business rules in terms of the demo
  - gives a short feedback cycle the IAM rules
  - provided us with a better understanding of the underlying problems
  - made it easy for the business to 'understand' what we are doing
- There seems to be a basis of IAM rules that we use over all contexts and that we use as our 'conceptual standard for IAM'
- Ability to organise IAM over multiple contexts facilitates application reuse (as shown by the demo).
- Applications using IAM services are provably compliant with the ruleset used to generate such services
- Created (parts of) functional architecture for various parties including a Dutch Telco and Defense Dept.
  - Many IT-architects and businesses have trouble grasping these ideas

# Business impact / Effects we anticipate:

- **the business is in the lead instead of the IT department**
  as every specified IAM function is traceable to (business) rules
- **no more database pollution**
  as IAM service layer is provably correct
- **guaranteed compliance and enhanced governance**
  as (business) rules are systematically monitored
- **novice programmers can write compliant applications**
  as they use the service layer that guarantees compliance
- **costs of defining and building systems will be slashed**
  as more of the automation process itself is automated, and the feedbackloop to the business is shortened
- **path towards standardization at higher conceptual levels, and**
- **good starting point for public awareness campaigns and support**
  as formal rules allow for precise comparison in different contexts

# Future work

- Development of tools
- Development of demo's, e.g. for
  - identifier translation
  - claim based authentication
  - IAM in personal networks (PNP2008)
- Further development of formal models
  - we have some (identifiers, authentication, authorization, etc.)
  - we need to refine them for additional contexts
  - we need new models (service provisioning, process development etc.)

I'm looking for people

# Thank you!

Rieks Joosten
+31 50 585 77 44
rieks.joosten@tno.nl