

Centre for
Telematics and
Information
Technology

24CQET



University of Twente
The Netherlands

The LicenseScript Project

Sandro Etalle

joint work with:

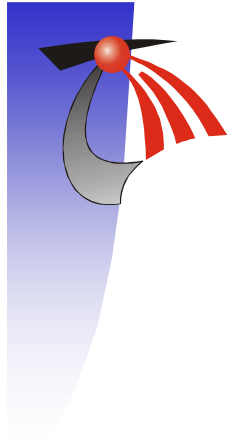
Jordan Chong, Ricardo Corin, Pieter Hartel, Yee Wei Law
EEMCS – University of Twente

in cooperation with
Telematica Instituut, Philips Research

13 June 2003

SAFE-NL





Centre for
Telematics and
Information
Technology

24CQET



University of Twente
The Netherlands

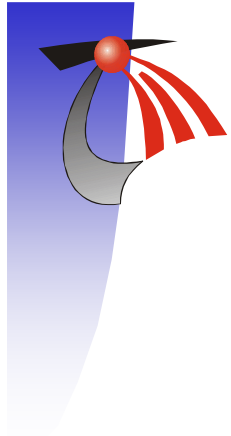
Overview

- Background
- DRM basics
- The LicenseScript project:
 - Objectives
 - Project description
 - The LicenseScript language
 - Some examples
- Conclusions

13 June 2003

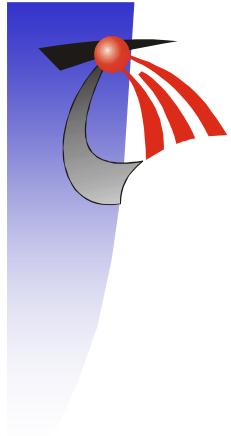
SAFE-NL





Background

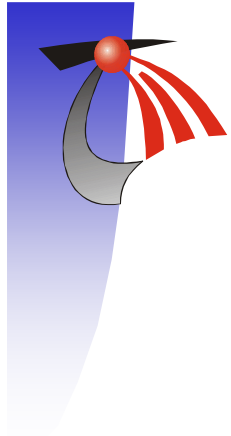
- Technology advances in content distribution form both a threat and an opportunity
- Content providers have lost billions
 - Napster, Kazaa, etc.
- Consumer Electronic (CE) manufacturers face now competition of the computer industry



The problems (some of)

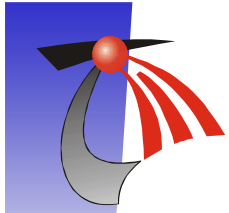
- Music and videos
 - can be easily duplicated
 - can be shared via P2P networks
 - difficult to manage and control
- Content providers were the first to be hit
 - most of the present “solutions” come from the CP industry
 - initially: copy protection.
 - they often joined forces with computer industry





Some opportunities

- For content providers and distributors
 - all sorts of new business models are possible
 - e.g., pay per view
 - the scope of the application broadens enormously
 - the whole distribution chain can be re-thought
- For computer manufacturers
 - stepping into the huge CE world



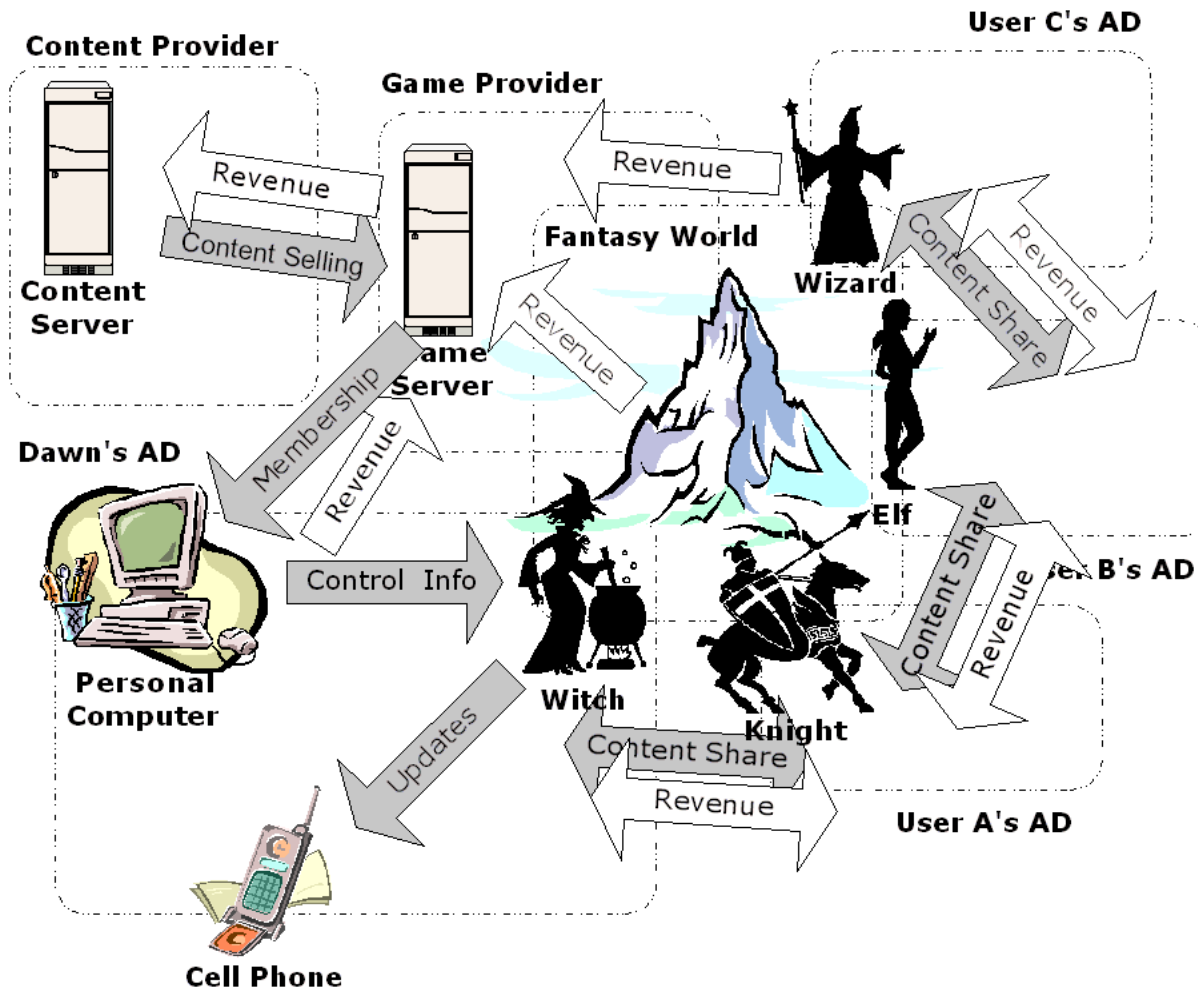
For instance, games

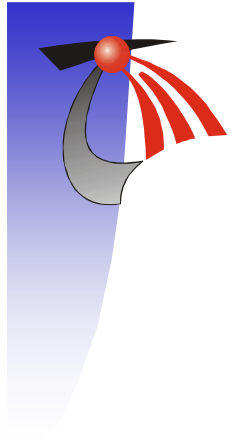
- massively multiplayer games are a multi-million \$ industry in Korea
 - millions of users
 - <http://www.mpogd.com/games/massively.asp>
- the user can define his/her own characters
- trading characters has begun already
 - what about the rights and duties of the author?
 - revenues, liability, ownership protection...
- other example: DJ





Content: from monolithic to patchwork





DRM basics 1

➤ Wanted:

A way of describing user rights & right attributes

➤ Type of rights (Stefik)

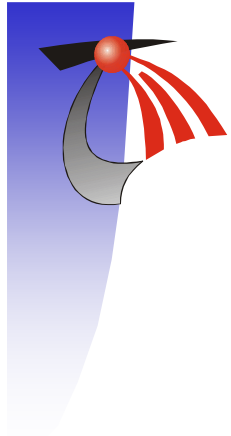
- render
 - print, view, play
- transport
 - copy, move, loan
- derivative
 - extract, edit, embed
- utility
 - backup...

➤ Right attributes

- you might use it 30 days before registering
- use only in NL
- not for public display
- ... for non-commercial use only ...
- ...

➤ Depend on *circumstances*

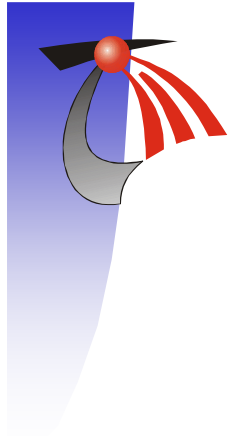




DRM basics 2

- Even more wanted:
 - a way of enforcing the correct application of user rights & right attributes**
- Context-aware
- Allowing to check (later?) that the actual use corresponded to the intended one
- Allowing **fair use**
- We are talking about **dreams** here...

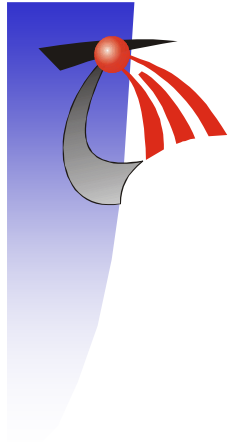




DRM basics 3: a taste of fair use

- Fair use is *by definition* an exception to the rule
- In the US legislation depends on
 - the purpose and character of use
 - the nature of the copyrighted work
 - the amount and the substantiality of the portion used in relation to the whole work
 - the *effect* of the use
- Example: making a copy of a book chapter





Centre for
Telematics and
Information
Technology

24CQET



University of Twente
The Netherlands

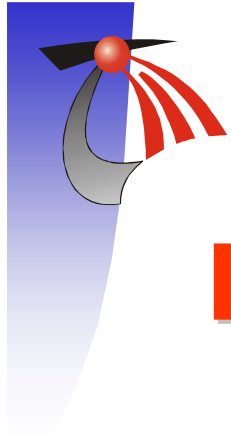
Reasons enough to get worried, or crazy

So, after this *reality check* let's get
back to concrete stuff: the
LicenseScript project

13 June 2003

SAFE-NL

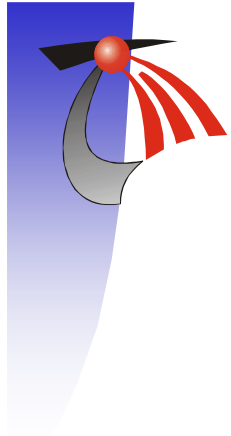




Principles of licensescript

- every piece of information should be equipped with a license
 - music, videos, but also personal data
 - anyone should be able to issue a license
 - one should be responsible of what he/she says
- licenses should be verifiable
- licenses should evolve with the data.





Project's goals

➤ a framework for the

- specification
- enforcement
- verification

of usage policy

➤ e.g also

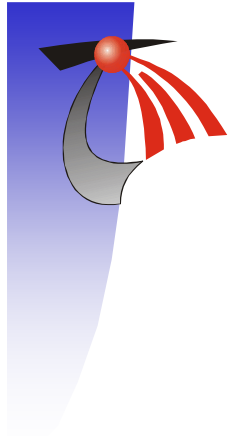
- privacy policies,
- software rights, etc

➤ implementation

- on trusted components
- the starting point is the architecture
 - like meta-S

➤ verification

- preventive (model checking)
- recovery (e.g., intrusion and fraud detection)



Constraints

➤ Home environment

- music should be playable on different devices
- devices may be *off...*

From device (or user)
centered to domain-
centered

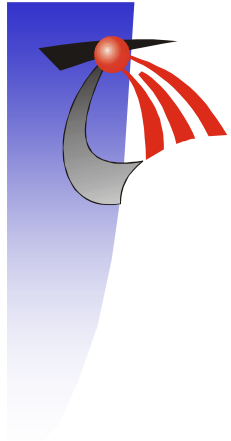
➤ Cosumer's 3A:

- content access anytime, anywhere, anyhow.
- ease of use is crucial for success

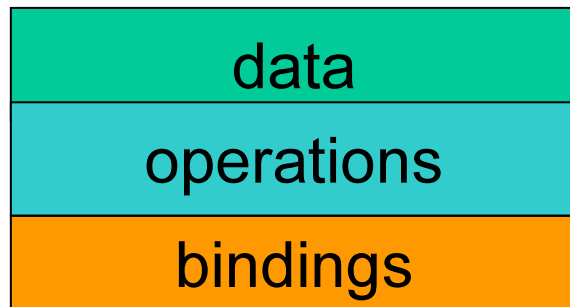
➤ Provider's 3F:

- Flexibility: for all business models.
- Financial beneficial.
- Fine-grained control over the content.





license structure



➤ data

- (pointer to) the music

➤ operations

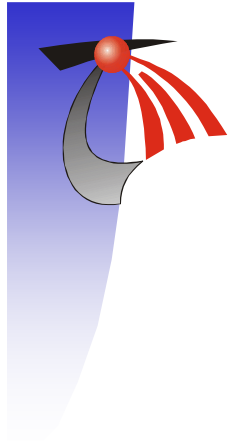
- program
- stating which operations are legitimate

➤ bindings

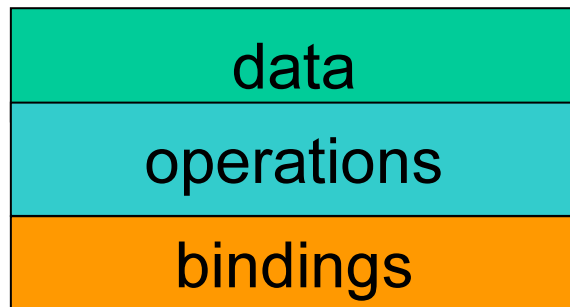
- the “data” in the program

bindings may change,
operations don't





a simple example



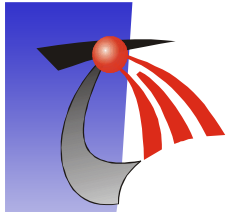
➤ operations example:

```
canplay(B, B) :-  
    today(D),  
    get_value(B,expires,Exdate),  
    Exdate > D.
```

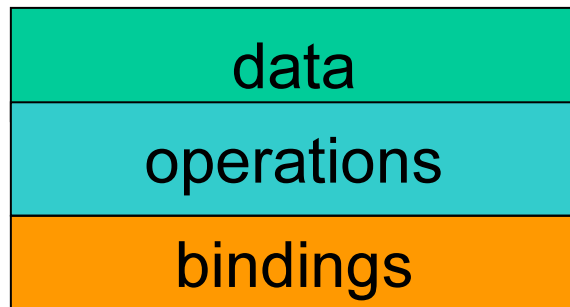
➤ bindings:

```
expires = 10/10/2003
```





another example



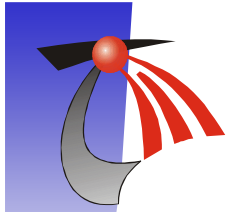
➤ operation:

```
canplay(B, B') :-  
  get value(B, played times, R),  
  R < 10,  
  set value(B, played times, R+1, B').
```

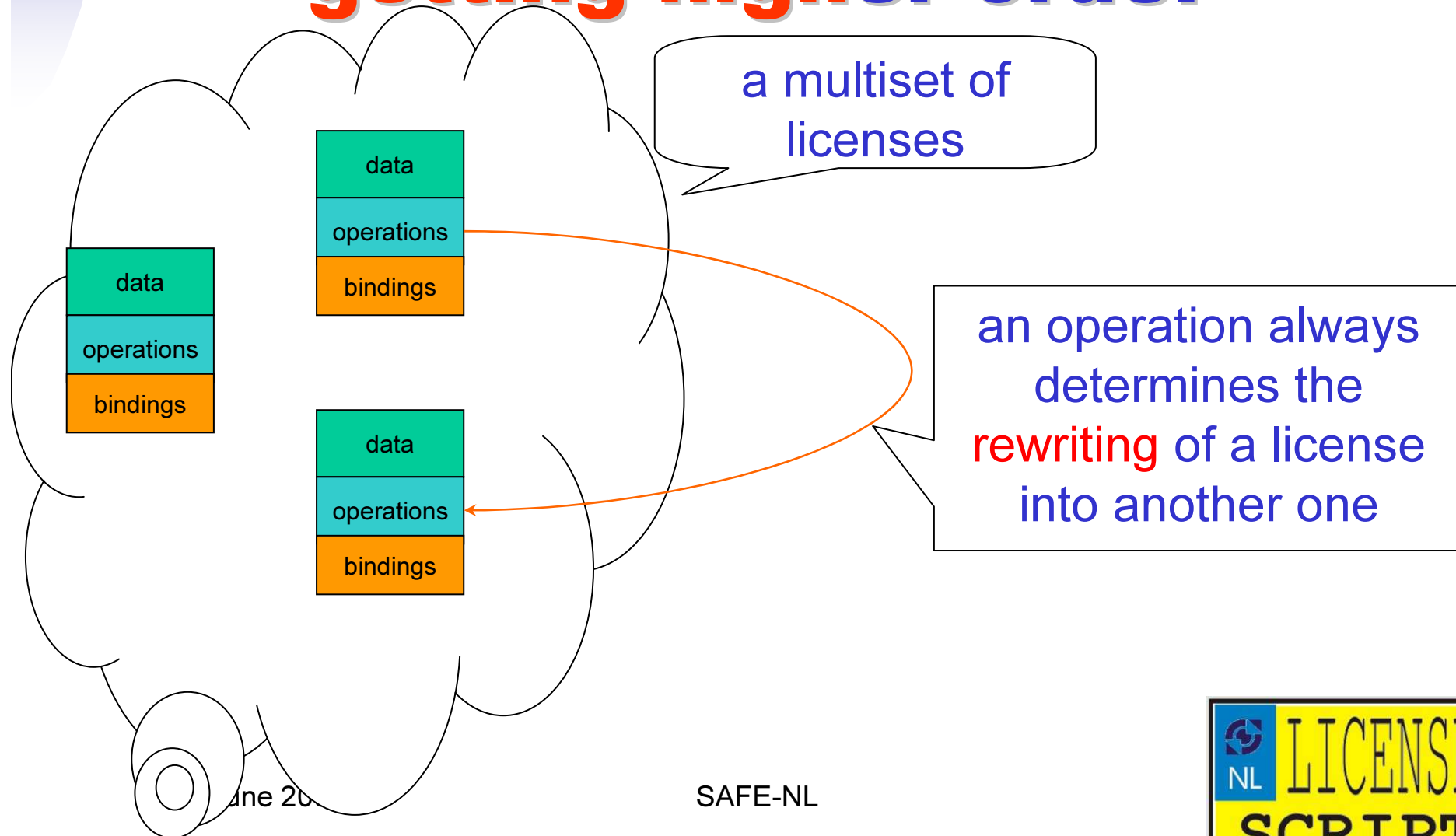
The play operation has
changed the bindings

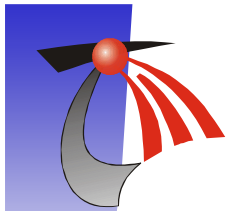
➤ bindings:

```
played_times = 3.
```

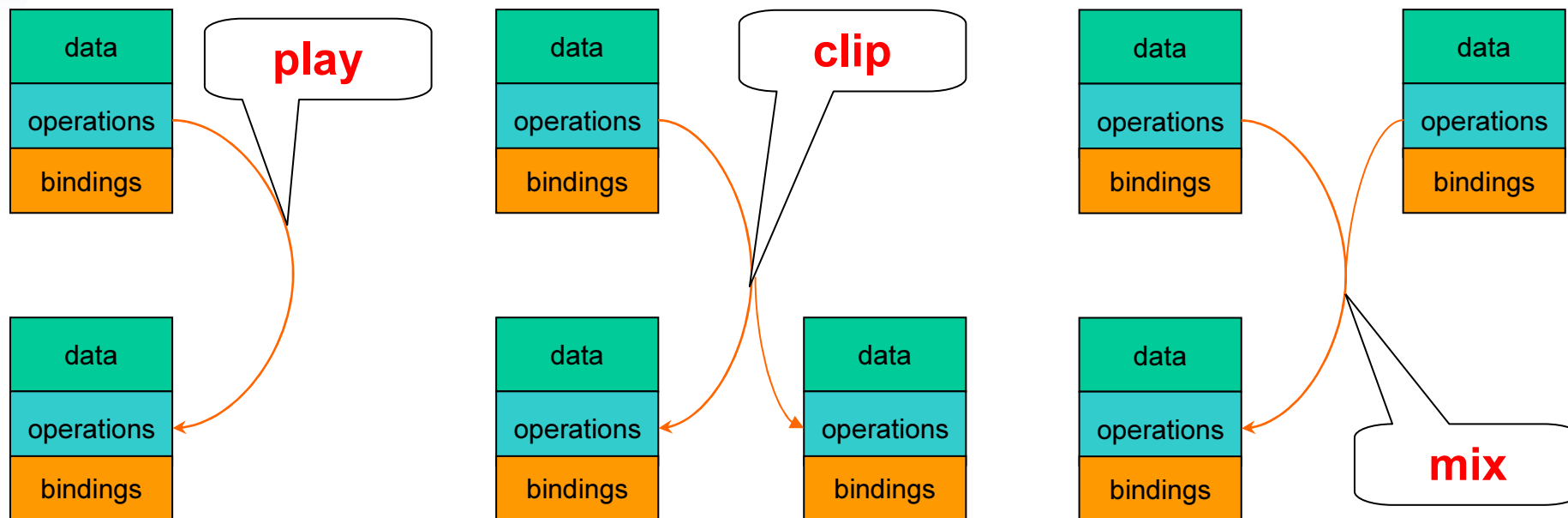


getting higher order



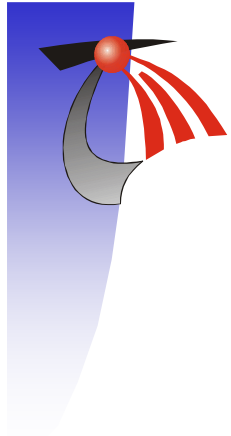


playing around



the license follows the content



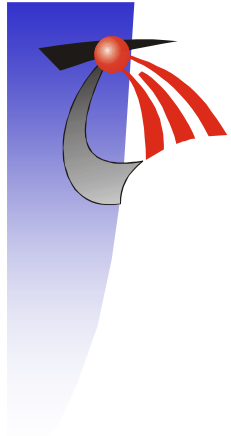


Rules: the missing step

- *Rules* allow to manipulate licenses
- *Rules* are the firmware of the devices
- E.g.

$\text{play}(X) : \text{lic}(X, \Delta, B) \rightarrow \text{lic}(X, \Delta, B') \Leftarrow \Delta \vdash \text{canplay}(B, B')$

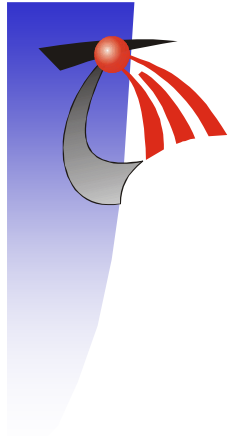
- Necessarily higher-order



Some conclusions

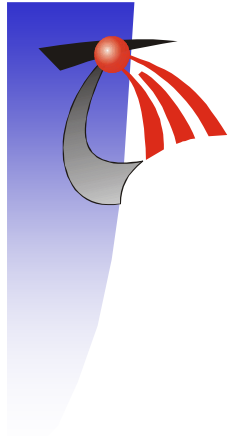
- a license is a basically a program
 - it is declarative – thanks to PROLOG
- licencescript has an architectural approach
 - we don't say only what it is supposed to hold
 - we check it explicitly
 - it is a *script*, think of an applet





why PROLOG

- Procedural and declarative at once
 - procedural: simple to implement
 - declarative: clear
- Recognized best language for modeling legal reasoning
- Ideal language for answering yes/no (maybe) queries
 - which is what we need here: cfr: XrML's semantics (XrML specification, part 2: Authorization algorithm)
 - queries can be complex
- Can handle higher order



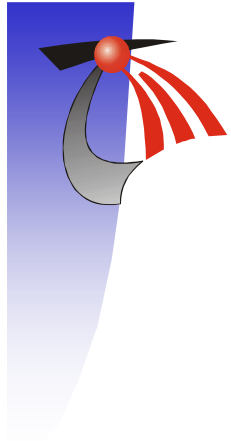
what about other DRLs

➤ Semantics

- XML-based rights language
 - no formal semantics...
 - *much more* complex (tried a number of examples)
- Gunter et al. and Pucella et al.
 - too static

➤ Architecture

- not quite taken into account,
- here: view of CE industry



Centre for
Telematics and
Information
Technology

24CQET



University of Twente
The Netherlands

References

➤ Media-S:

- SideSpace Solutions: www.sidespace.com

13 June 2003

SAFE-NL

