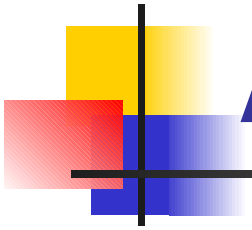
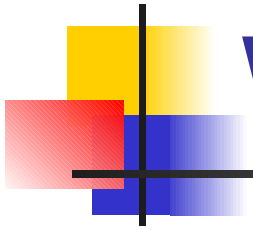
MPEG REL or XrMLv2
for UVCrypto

Tolga Acar
Microsoft



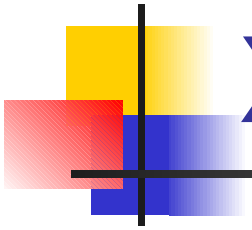
Agenda

- Introductions
- What is XrML?
- Authorization Model
- What is a Grant?
- Authorization Examples



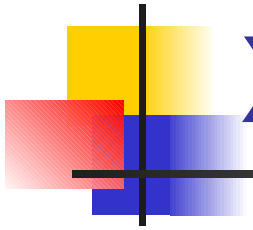
We need…

- Authorization Specification
- Issuance Specification
- Trust Specification
- Well Defined Semantics
- Extensibility
- Generality
- Integrity



XrML…

- Provides systematic, semantically unambiguous expression of authorization & trust policy
- Policies are interpreted by the Language Interpreter
- Relies on Security Contexts to enforce the policy



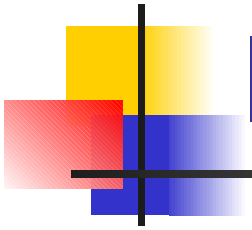
XrML Scope

a) Policy

b) Expression of Policy

c) Enforcement of Policy

Correct answer b)



Inference from Policy 1

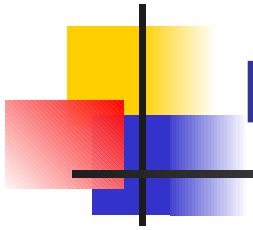
Policies

- *John grants Bill can print the book*
- *John grants Bill can read the book*
- *Tom grants John can pick anyone to read any book*
- *My system trusts only what Tom grants*

Conclusion

- *Therefore Bill may read the book but may not print the book*

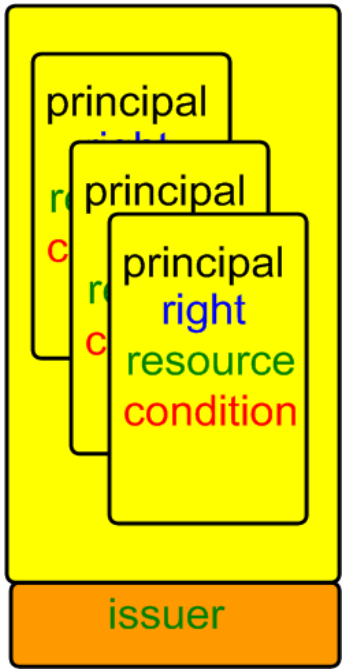
Well-defined authorization model

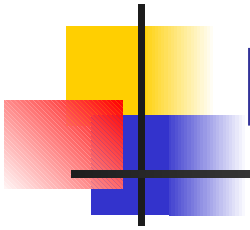


- Input
  - **Principal (“who”)**
  - **Right (“what”)**
  - **Resource (“object”)**
  - **Licenses (“policies”)**
  - **Root/Trust Policy (“who/what do we believe”?)**
  - **Time**
- Output
  - **Yes (with Conditions, contextual information, i.e. which grant etc.). Including “Proof”**
  - **No**

An XrML License

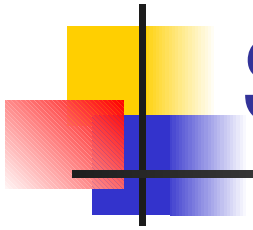
- A license contains one or more grants.
- License is authorized by the *issuer*





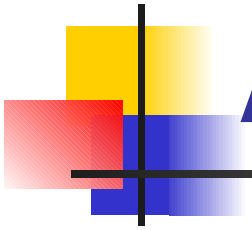
Essential Semantic of a Grant

- *Issuer* grants
  - *principal* the
  - *right* to
  - *resource* under
  - *condition (optional)*



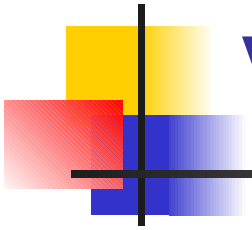
Simple Stuff

- *Acme.Com grants*
  - ***keyHolder Alice** can*
  - ***play***
  - ***A Clockwork Orange (License A)***



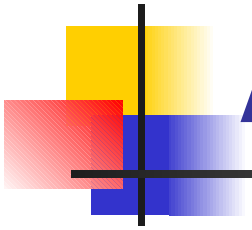
Authorization

- Input
  - ***keyHolder Alice***
  - ***Play***
  - ***A Clockwork Orange***
  - **License A**
  - **Believe whatever *Acme.com* says (Root Policy)**
- Output
  - Yes



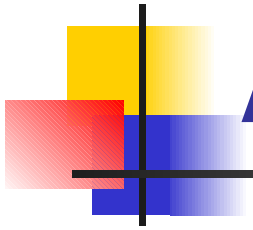
Variables

- *Acme.Com grants*
  - ***forAll "X"***
  - ***keyHolder Alice can***
  - ***right "X"***
  - ***A Clockwork Orange (License B)***



Authorization

- Input
  - ***keyHolder Alice***
  - ***Play***
  - ***A Clockwork Orange***
  - **License B**
  - **Believe *Acme.com***
- Output
  - **Yes**

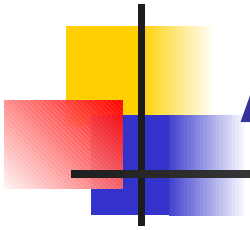


Adding Variables

- No relationship between Licenses
- Still no combination of policy
- Size of proof is still exactly one grant
- However, a single grant can authorize many different authorization requests

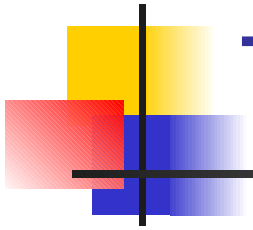
Variables & Patterns

- *Acme.Com grants*
  - ***forAll "X"***
    - ***startsWith("p")***
  - ***keyHolder Alice can***
  - ***right "X"***
  - ***A Clockwork Orange***



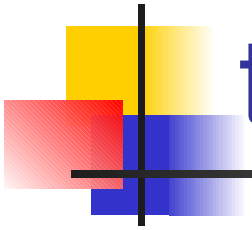
Authorization

- Input
  - ***keyHolder Alice***
  - ***Copy***
  - ***A Clockwork Orange***
  - **Licenses**
  - **Believe *Acme.com***
- Output
  - **No**



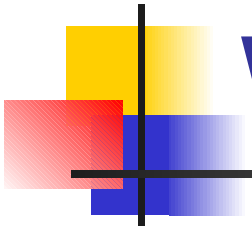
The “issue” right

- *Issuer* authorizes
  - *principal* to
  - *Issue*
  - *grant* under
  - *condition*



trustedRootIssuer

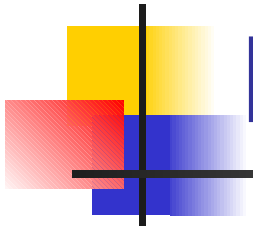
- *Issuer* grants
  - *principal* the
  - *right* to
  - *resource* under
  - *prerequisiteRight*
    - ***principal A***
    - ***right B***
    - ***resource C***
    - ***trustedRootIssuer****: Issuer I*
- *Issuer* grants *principal* the *right* to *resource* if *Issuer I* authorizes *principal A right B* over *resource C*



What's really in a license…

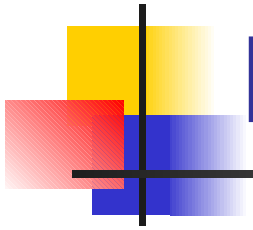
- **License**
  - **Grant (+)**
    - **DelegationControl (o)**
    - **ForAll (*)**
      - **Pattern (*)**
    - **Principal**
    - **Right**
    - **Resource**
    - **Condition (o)**
  - **OtherInfo**
  - **Issuer (+)**
    - **dsig:Signature**
    - **TimeOfIssue**

+ One or more
* Zero or more
o Optional



License A in XML

```
<license>
    <grant>
        <keyHolder>…</keyHolder>
        <play/>
        <digitalResource>
            <identifier> A Clockwork Orange </identifier>
        </digitalResource>
    </grant>
    <issuer>
        <Signature>…</Signature>
        <timeOfIssue>…</timeOfIssue>
    </issuer>
</license>
```

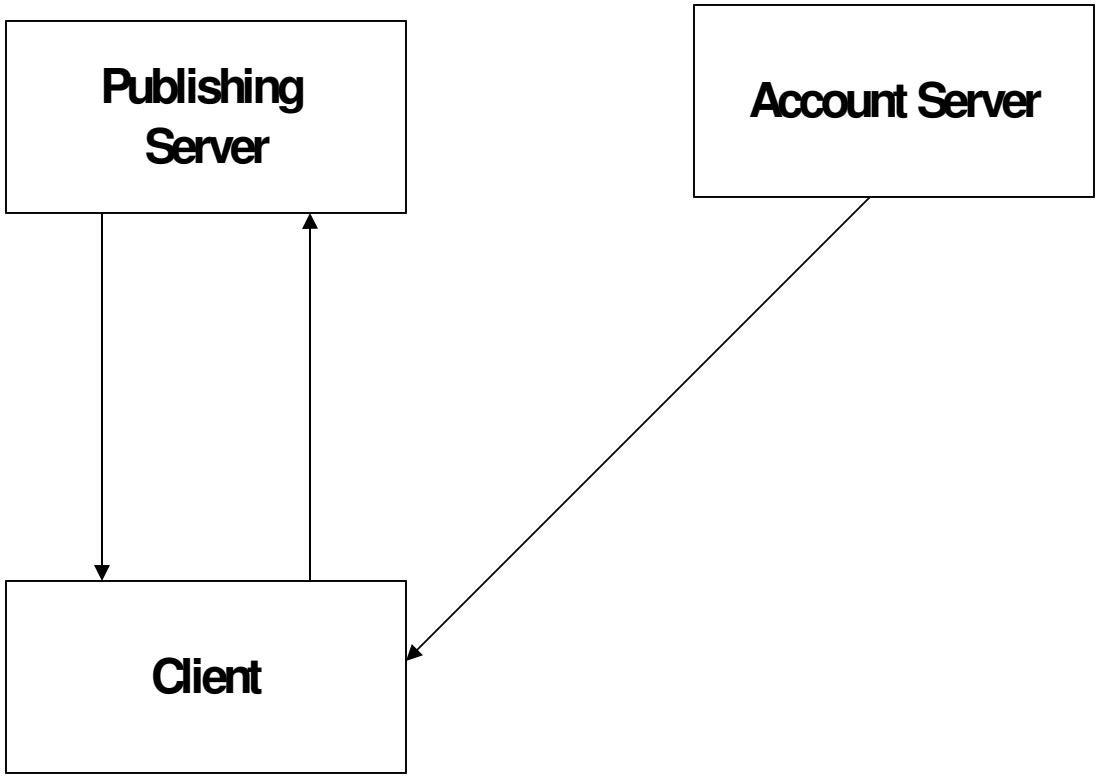


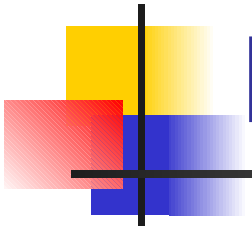
KeyHolder:Principal

- Uses xml dsig;KeyInfoType
- dsig:keyInfoType has built-in support to represent RSA, DSA key values, and embedding X.509 certificates etc.

keyHolder
  info
    KeyName keyHolder *tolga@uvcrypto.com*
    KeyValue
      RSAKeyValue
        Modulus wmbwKtO….
        Exponent AQAB

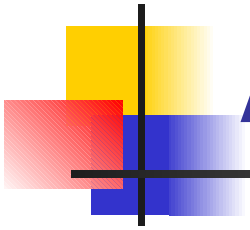
A Trust Model Slice





Publishing Policy

License
Grant
ForAll varName = “X”
KeyHolder PublishingServer
Issue
Grant
KeyHolder varRef = “X”
read
digitalResource
document *id1*
prerequisiteRight
KeyHolder varRef = “X”
possessProperty
emailName
ends in @uvcrypto.com
trustedRootIssuer : AccountServer
Issuer
YourTrustedRoot/DontCare



Account Certificate

license

- **grant**
  - **keyHolder**
    - **Tolga's RSA public key**
  - **possessProperty**
  - **emailName – tolga@uvcrypto.com**
- **issuer**
  - **Account Server**