

A Software Architecture to Facilitate the Creation of DRM Systems

koen.buyens@cs.kuleuven.be



Joint work with Kristof Verslype, Sam Michiels,
and Wouter Joosen
DistriNet research group, K.U.Leuven, Belgium

Overview

- Context
 - Challenges
 - Objectives
 - Approach
- Architecture
 - Architectural overview
 - Scenario
 - Architectural decisions
- Conclusion and future work

Context: challenges

- DRM field is complex
 - Diversity of devices, users, platforms, media
 - Wide variety of system requirements
 - Security, flexibility, ...
- Complexity poses 4 major challenges
 - Rapidly evolving field (extensibility)
 - Varying business policies (modifiability)
 - Consumers should be able to use different services with same consuming device (interoperability)
 - Do not reinvent the wheel (reuse)

Objectives and Approach

- Research objective
 - Propose a blueprint (software architecture) for DRM that tackles these challenges
- Approach (bottom up):
 - Identify common parts in existing DRM systems
 - What is lacking in existing systems?
 - Develop blueprint (architecture), based on [8]
 - Has common fixed set of components
 - Support for different requirements
 - Different applications, context, environment, policies

[8] Jamkhedkar and Heileman. DRM Interoperability Analysis from the perspective of a layered framework

A Software Architecture to Facilitate the Creation of DRM Systems

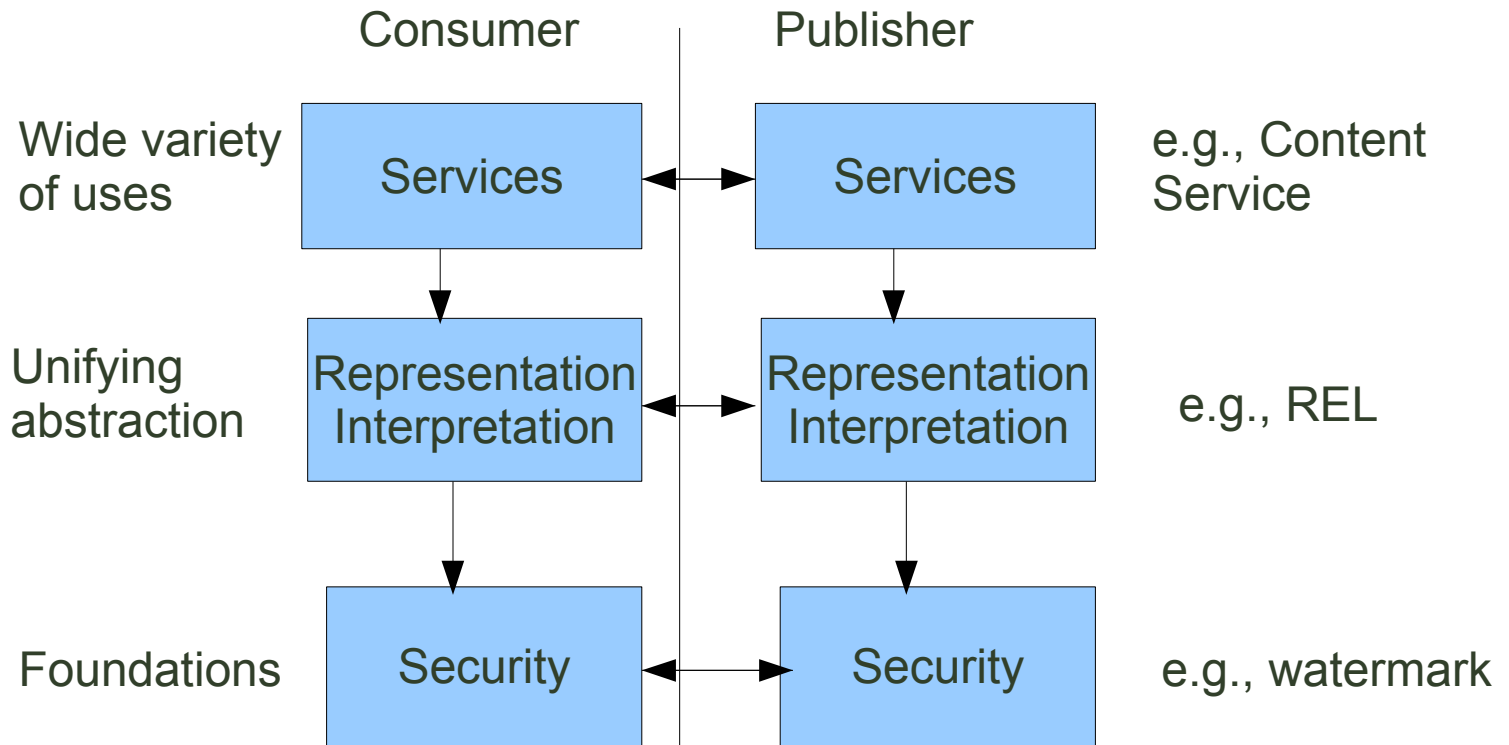
Approach (continued)

- Approach (continued)
 - Software architecture based on architectural patterns
 - structural organization schema
 - predefined subsystems
 - specifies responsibilities and relationships
 - Implement prototype
 - Evaluate

Overview

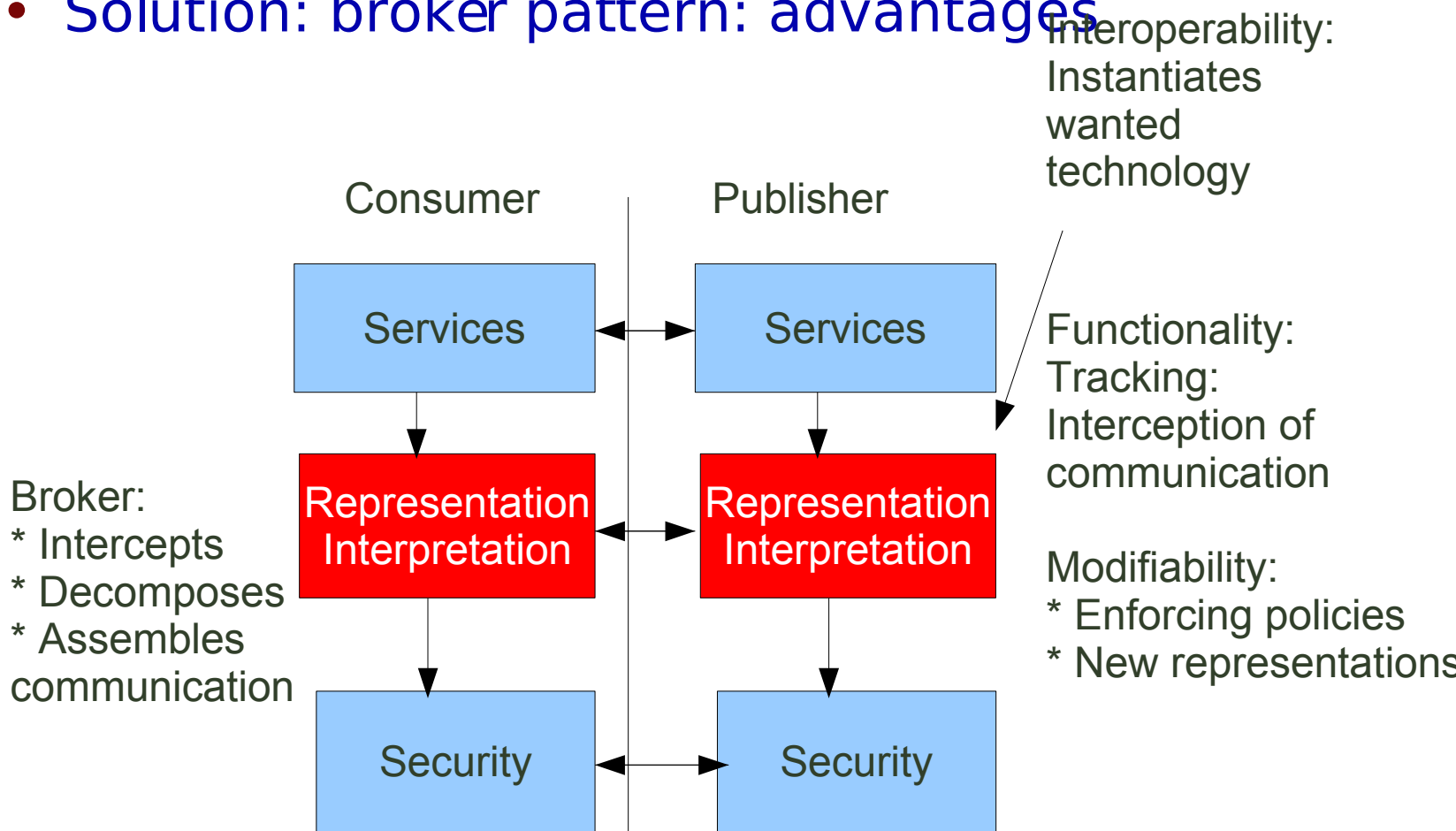
- Context
- Architecture
 - Architectural overview
 - Scenario
 - Architectural decisions
- Conclusion and future work

Overview of architecture



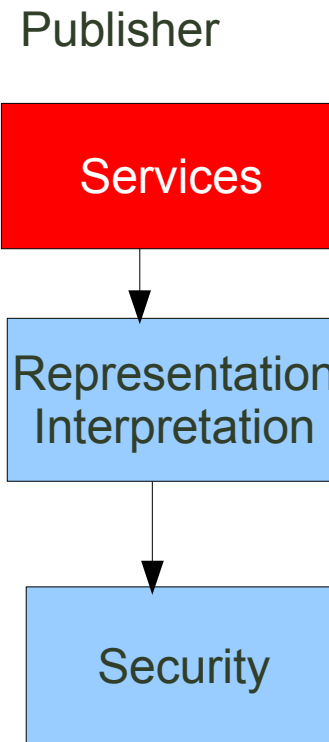
Overview of architecture

- Solution: broker pattern: advantages



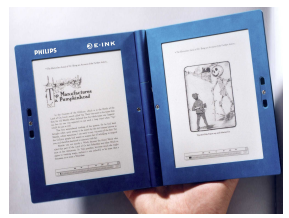
Overview of architecture (2)

- Zoom in on each part



Identified service components

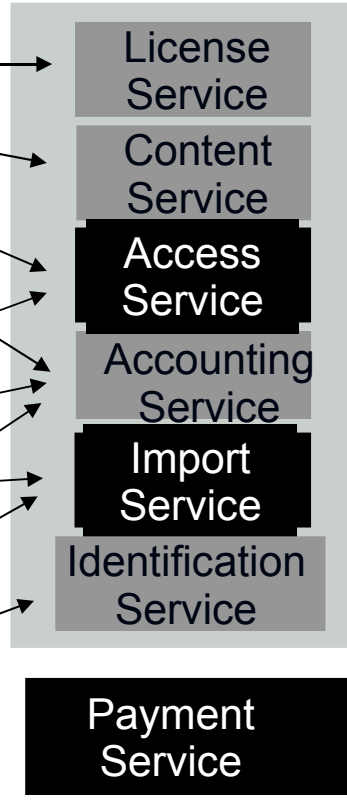
Consumers



Producers



Publishers



Major DRM Service components

External Services

Legend

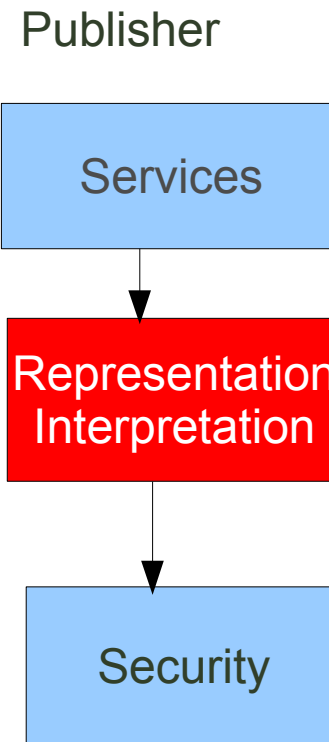
Publishing context

Service architecture: needed

- Lots of complexity
- Modifiability and extensibility
 - Choose services which you are interested in
 - Add new services
 - Replace services
- Reusability
 - Reuse/buy services from other companies and plug them in

Overview of architecture

- Zoom in on each part

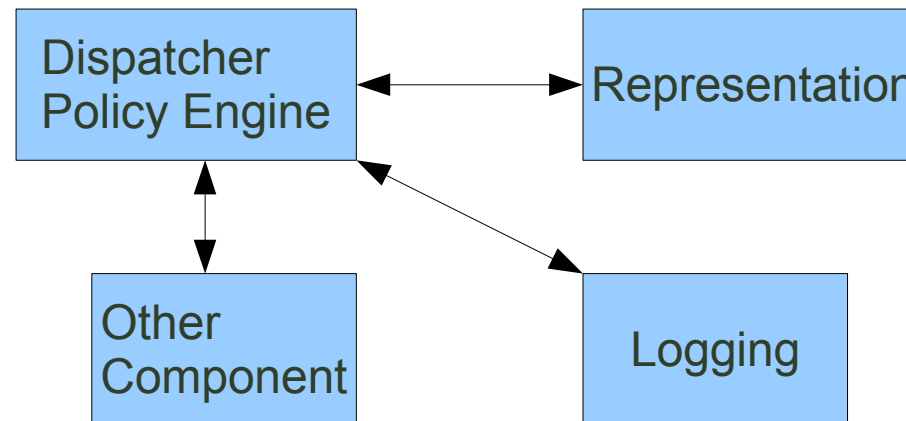


Representation and Interpretation: needs

- **Functionality**
 - Represent content and licenses
 - Interpret policies
- **Modifiability**
 - Another policy, ...
- **Extensibility**
 - Add new REL, content format, policy, ...
- **Interoperability**
 - Instantiate into specific technology, e.g. Windows media DRM
- **Reuse**
 - Reuse policies, components, ...

Representation and Interpretation: solution

- Dispatcher pattern: advantages



Interoperability:
Interfaces

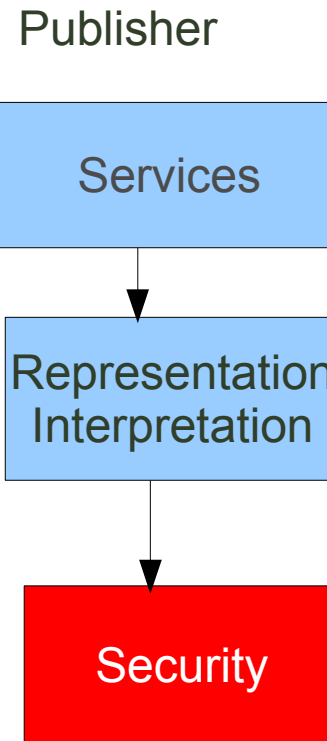
Modifiability
Extensibility
* Plug-in own
components

Representation and Interpretation: solution

- Representation Component: allows different representations of
 - RELs: e.g. XrML, ODRL, ...
 - content formats, e.g. AAC, Windows Media DRM, ...
 - Identifiers, e.g. for users, devices, domains, ...
 - policies
- Policy Engine: interprets policies (including licenses) and gathers required context
- Logging Component: logs several requests

Overview of the architecture

- Zoom in on each part



Security layer : needs

- *Extensibility*
 - *New cryptographic algorithms*
- *Modifiability*
 - *Changing algorithms*
- *Interoperability*
 - *Different versions of same algorithm*
- *Reusability: crypto library or components*

Security: solution

- Security library
- Used patterns:
 - Layers
- Advantages
 - Extensibility: new algorithms
 - Modifiability: new default algorithms
 - Interoperability: various existing algorithms

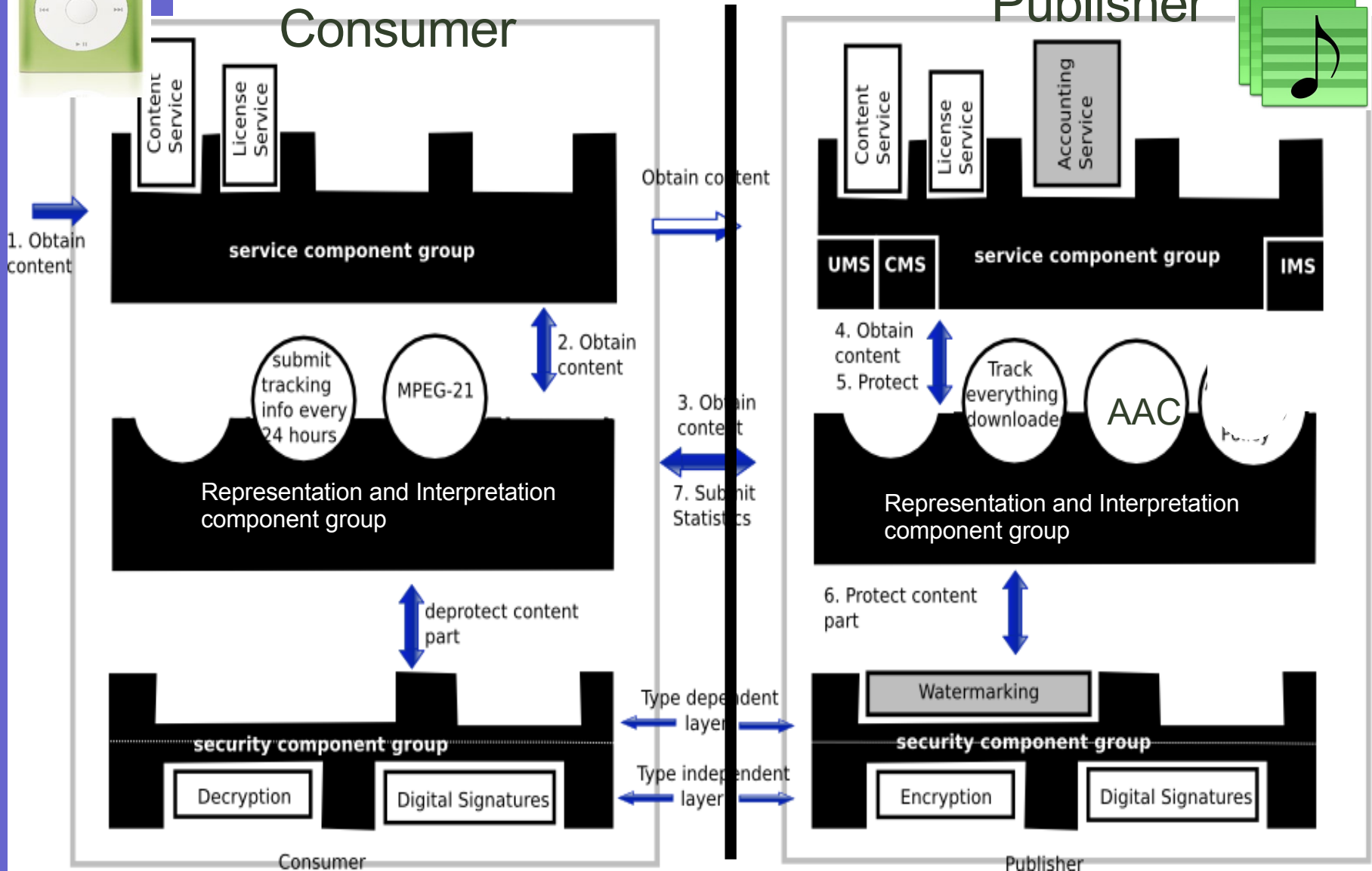
Security: identified components

- Identifying security components
 - Identified components organized as layers
 - Upper layer type dependent components
 - Watermarking Component: watermarks data
 - Fingerprinting Component: makes fingerprint
 - ...
 - Lower layer type independent components (crypto)
 - Encryption Component: envelops data
 - ...

Overview

- Context
- Architecture
 - Architectural overview
 - Scenario
 - Architectural decisions
- Conclusion

Architectural Overview: scenario



Overview

- Context
- Architecture
 - Architectural overview
 - Scenario
 - Architectural decisions
- Conclusion

Architectural decisions

- Broker pattern
 - Pros: Modifiability, extensibility
 - Cons: Performance, single point of failure
- Dealing with the cons
 - In deployment of architecture
 - E.g., components on individual machines
 - E.g., duplication of machines

Overview

- Context
- Architecture
- Scenario
- Architectural decisions
- Conclusion

Conclusion: contribution

- Proposed a software architecture for DRM that supports
 - Reuse
 - Interoperability
 - Modifiability
 - Extensibility
- Designed and implemented key parts of the architecture's security component group, service component group
- Future work
 - Validate architecture in proof of concept prototype

A Software Architecture to Facilitate the Creation of DRM Systems

Koen Buyens, Sam Michiels and
Wouter Joosen

DistriNet research group
Koen.Buyens@cs.kuleuven.be

KATHOLIEKE UNIVERSITEIT

LEUVEN

