Design Rules for Interoperable Domains

Controlling Content Dilution and Content Sharing

Dr. Gwenaël Doërr – University College London, UK

Ton Kalker – Hewlett Packard Labs, Palo Alto, CA, USA

UCL

hp invent

Outline

- Incentives for interoperability
- Domains of devices
- Domain configuration parameters
- (Controlling content dilution)
- Allowing content sharing

Incentives for Interoperability (1/2)

- Legal pressure
  - Class action threat based on consumer protection and anti-trust legislation
  - New legislation to force interoperability
- Consumers switch from illegal copies to protected content
- Monopoly does not maximize profit
- Spread the security risk across several systems
- Business becomes DRM-agnostic

Incentives for Interoperability (2/2)

- Lowers entry barriers for new players and promotes innovation

Strong pressure on DRM vendors to “open” their proprietary DRM technologies

CORAL CONSORTIUM

DECE LLC
(Open Market)

dlna

HANA

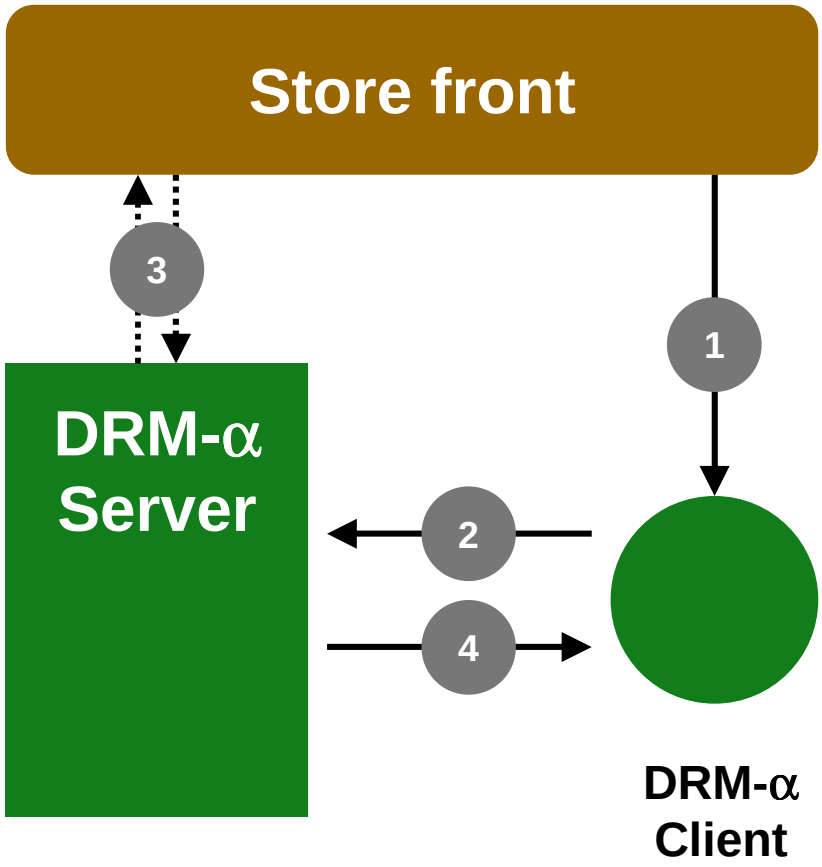
Approaches to Interoperability

- Single DRM standard
  - ☹ Well established DRM providers are reluctant
  - ☹ Prohibitive licensing cost
- Bilateral agreements e.g. translation services
  - ☹ Loss of information (incompatible REL)
  - ☹ Scalability
  - ☹ Consumer confusion
- Scalable and multilateral interoperability framework
  - ❖ Domain model

Domain Model

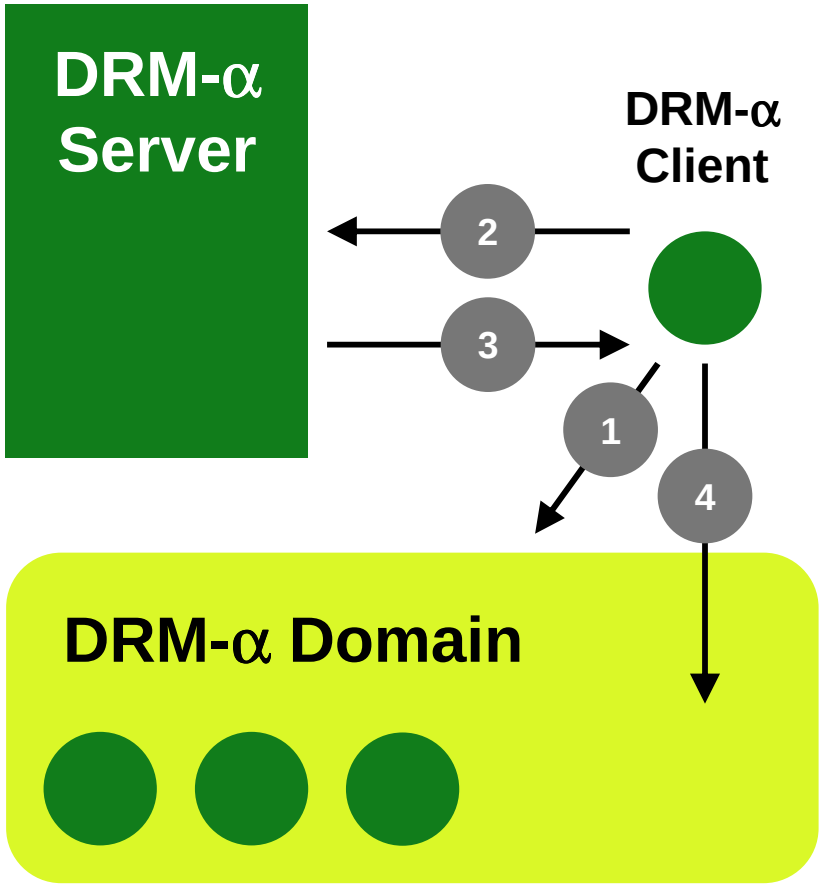
- Domain = group of devices
  - Content can freely flow between devices (content dilution)
- Coral's philosophy: do NOT re-invent the wheel
- Most deployed DRM system share the same architecture
  1. Scrambling layer: bulk encryption of digital content
  2. Management layer: key management and access control (*licenses*)
  3. Communication layer: defines communication between DRM services (*trigger*)

Trigger-enabled License Acquisition



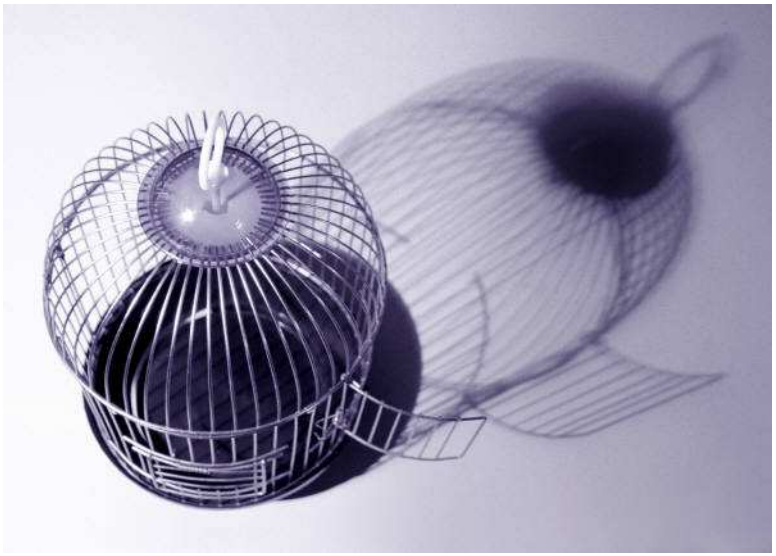
1. Upon purchase, the store front sends a trigger to the client.
2. The client forwards the trigger to the DRM server indicated in it.
3. The DRM server checks with the store front that the client is entitled to get a license for the content indicated in the trigger.
4. The DRM server issues a license and sends it to the client.

Joining a Domain



1. A new device “discovers” the Authorized Domain.
2. The device sends a join_request to the DRM server.
3. The DRM server checks its policy and possibly authorizes the device to join.
4. The device is now part of the domain.

Breaking Through Walled Gardens

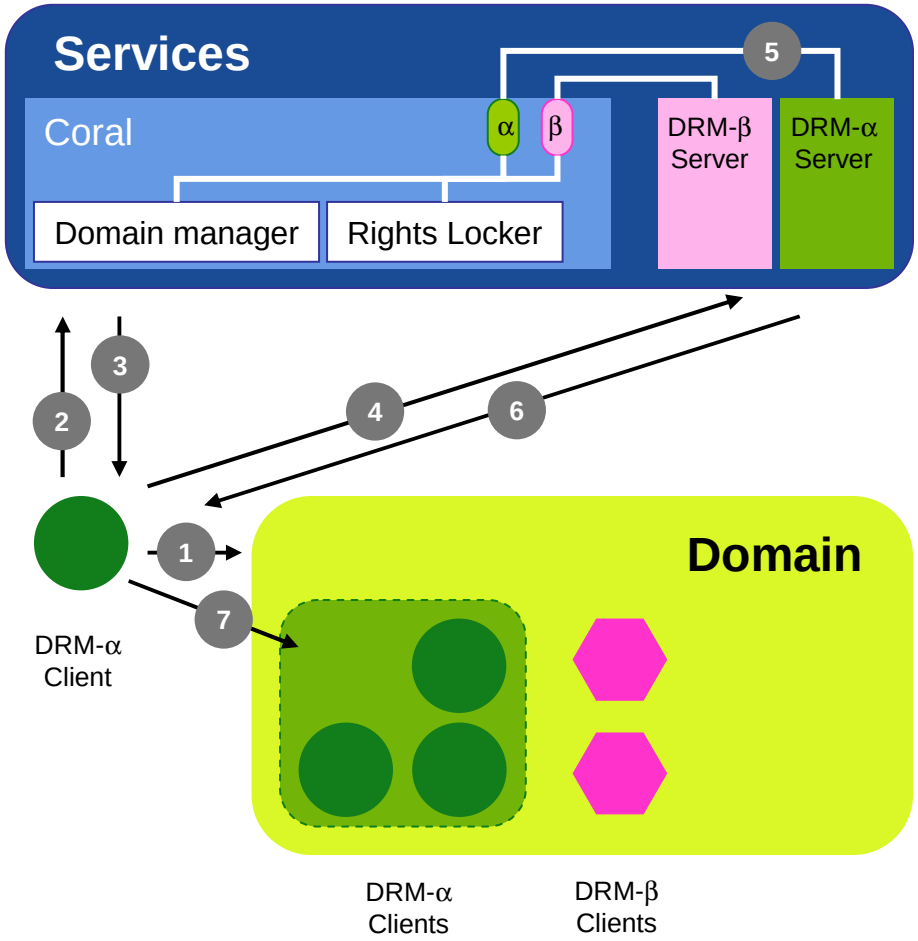


- Shortcoming of native domains
  - All devices are required to implement the same proprietary DRM technology (lock-in)

No interoperability across DRM

- Interoperable domains with Coral
  - Separate *authorization* from *enforcement*
    - DRM agnostic standardized authorization
    - Proprietary DRM enforcement

Interoperable Domains

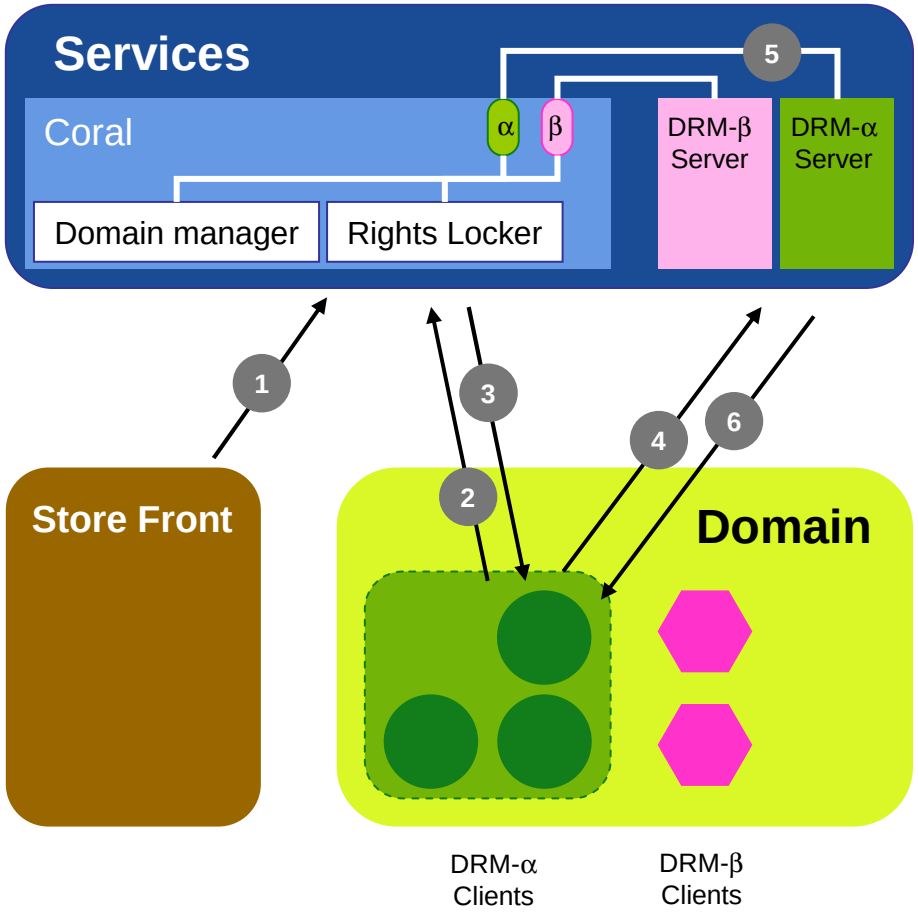


1. A new device “discovers” the domain.
2. The device sends a `join_request` to the interoperable domain manager, which checks whether it is allowed by its policy.
3. The domain manager returns a trigger with information to locate the relevant native DRM server.
4. With the trigger, the client formulates a request to the native DRM server to join the native domain. The server checks if it is possible or not.
5. If agreed, the native DRM server notifies the Coral domain manager.
6. The DRM server delivers a native domain membership token to the device.
7. The device is now part of the domain.

Rights Tokens

- DRM agnostic data structure holding the authorization to access content
  - Who (user, domain, device)
  - What (content)
  - How (usage model)
- Rights Tokens do **not** rely on any REL
  - Usage rules are decided at ecosystem level and stipulated with an index e.g. usage rule #5

From Rights Tokens to Native Licenses



1. When the consumer buys content, the store front delivers a DRM agnostic rights token to the rights locker.
2. A device of the domain want to access the content and sends a request to the rights locker.
3. If the request is allowed, the rights locker creates a DRM-specific trigger and sends it to the device.
4. The device forwards the trigger to the relevant DRM server to get a native license.
5. The DRM server checks the rights locker to verify that a license can be issued.
6. The DRM server generates a native DRM license and dispatches it to the device.

Configuring Domains

- Content can flow freely within the domain ⇒ content dilution
  - Need to define boundaries
    - Too loose = unlawful sharing and loss of revenue
    - Too tight = does not accommodate consumers needs
- Three main primitives
  - Counters and cardinality limits
  - Clocks and time-outs
  - Proximity detection

Counters and Cardinality Limits

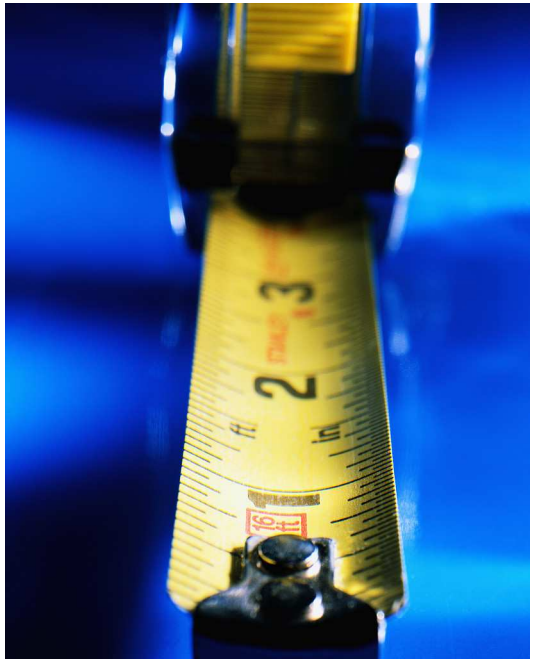
- Simplest and most used DRM parameters
  - clients-per-domain, domains-per-client
  - accounts-per-domain, domains-per-account
  - account-flipping-limit-number
- Technical challenges
  - Maintain accurately and consistently concurrent counters
  - Flexible mechanism to update counters e.g. device de-registration

Clocks and Time-outs

- Limit the lifetime of entities within domains & restrict the frequency of certain actions
  - valid-until, member-client-timeout
  - account-flipping-limit-time
- Use to enforce business models and for damage control
- Technical challenges
  - Synchronization
  - Tamper-proof

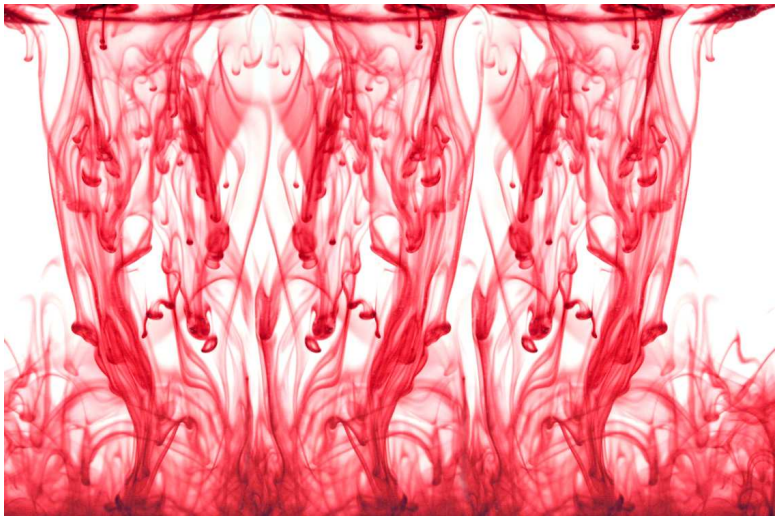
Proximity Detection

- Estimate physical proximity (to approximate household)
  - DVD Regional playback control
  - Wire connectivity
  - DTCP, Cardea, etc.
- Key questions
  - When: registration, acquisition, playback
  - Between what: anchor devices
- Technical challenge
  - Accuracy



Controlling Content Dilution

- Typical setup
  - Conservative cardinality limit e.g. a handful of devices
  - Sensitive information bond
    - ⇒ Ends up in a very frustrating situation for the consumer



Is it possible to relax such conservative choices?

Fill-and-Resell Business

- ❑ Content stored on digital devices is more valuable than the device itself
  - ❖ iPod classic: $350 / 160GB ~ $40.000
  - ❖ Buy devices, fill them up and resell them with a profit
- ☒ Cardinality limit
- ☒ Proximity detection
- ☑ Time-out domain membership
  - ❖ Damage control

Sharing Between Distant Consumers

- ❑ Sharing a domain = sharing credentials to register
  - ❖ Users who have never met can join the same domain
- ☒ Cardinality limit
- ☒ Time-out
- ☑ Proximity detection
- ❑ Alternative: bind sensitive information to credentials
  - ❖ Does not solve the problem
  - ❖ Strong disincentive to sharing

Densely Populated Buildings

- ❑ Typical example: student dorm
- ☒ Proximity detection
- ☒ Time-outs
- ☑ Cardinality limit

Summary

- Conservative cardinality limits are not the only way to define domain boundaries
  - Proximity detection ↔ guarantee that devices physically meet the domain before joining in
  - Time-outs ↔ damage control for devices leaving the physical boundaries of the domain
  - Cardinality limits ↔ densely populated buildings

All is best… but may be not necessary for all business models

Throwing Bridges Between Domains

- Is interoperability within “personal” domains enough?
  - Content sharing is not a right… but consumers got used to this usage and value it highly
- Domain sharing may be necessary to accommodate for some real life social situation

How to open up domains and still keep control over content?

Content Sharing

- Split and merge social events
  - Example: divorce, wedding, room/flat sharing
  - Deviates from the underlying assumption that a household is a fixed group of persons based at a fixed location
    - Still manage as outliers in current systems
- Social sharing
  - Example: a friend pay you a visit for 2 weeks and would like to enjoy your content library on his portable device

Resolving Content Sharing

- Allow a device to register to more than one domain
- Keeping control over the flow of content
  - Proximity detection
    - Registration: the device has to meet the domain
    - Content acquisition: device can load content only if within the domain
    - Rendering: optional
  - Domain membership time-out
    - Damage control if rendering does not require proximity detection
  - Cardinality constraint to limit domain membership aggregation
  - Relax the domain update policy

Conclusion

- 1st generation DRM systems heavily focused on copy protection
  - Upset consumers
- Domain model
  - Grants the possibility to re-introduce highly valued usages
  - Still not perfect
    - First sale doctrine, gifting
  - Each actor along the content value chain focus on its on business

Acknowledgements

The Royal Academy of Engineering