Imperial College London

Sharing Data Through Confidential Clouds: An Architectural Perspective

Daniele Sgandurra, Francesco Di Cerbo, Slim Trabelsi, Fabio Martinelli, and Emil Lupu

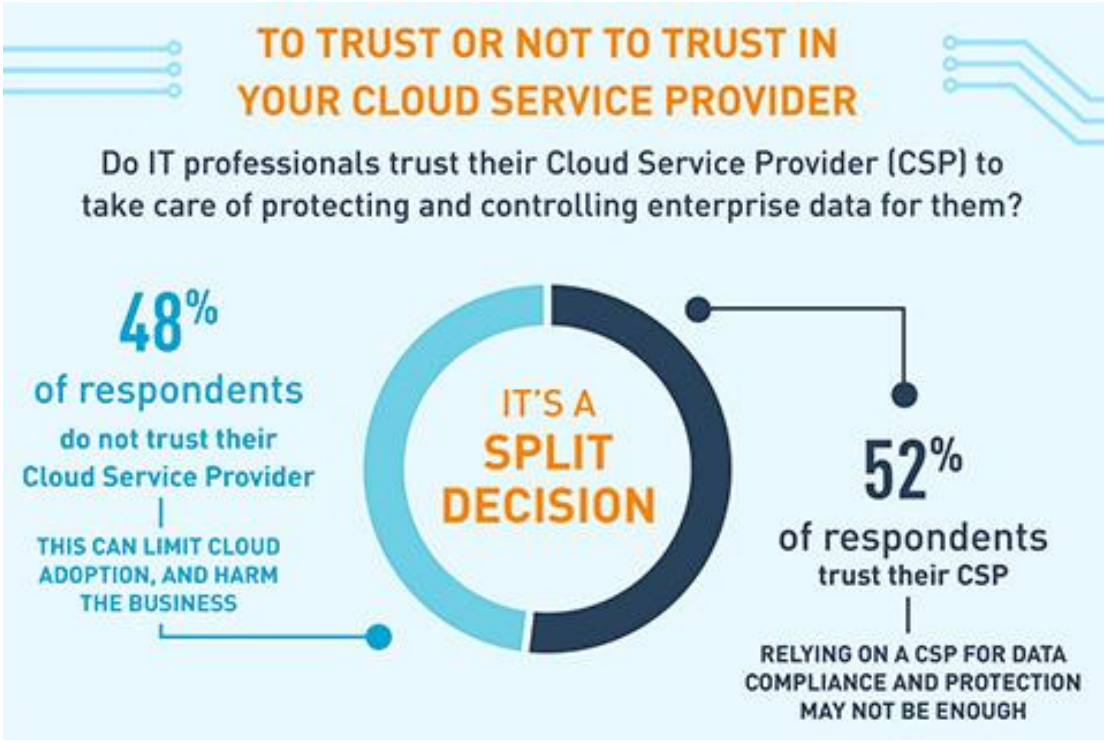
1st International Workshop on TEchnical and LEgal aspects of data pRIvacy and SEcurity

Florence, 18/05/2015

Moving to Cloud

Outsourcing **data storage** and **management** to the Cloud, usually for improved availability and sharing, entails an inherent **loss of control** by Cloud tenants**.**

To Trust or Not to Trust?



Source: http://www.net-security.org/secworld.php?id=18369
(Survey taken at RSA Conference 2015, 125 attendees)

Additional means of **data protection** and **usage control** must be provided to allow organizations to safely **share** data on the Cloud**.**

Goals of our Project

Goals of our Project

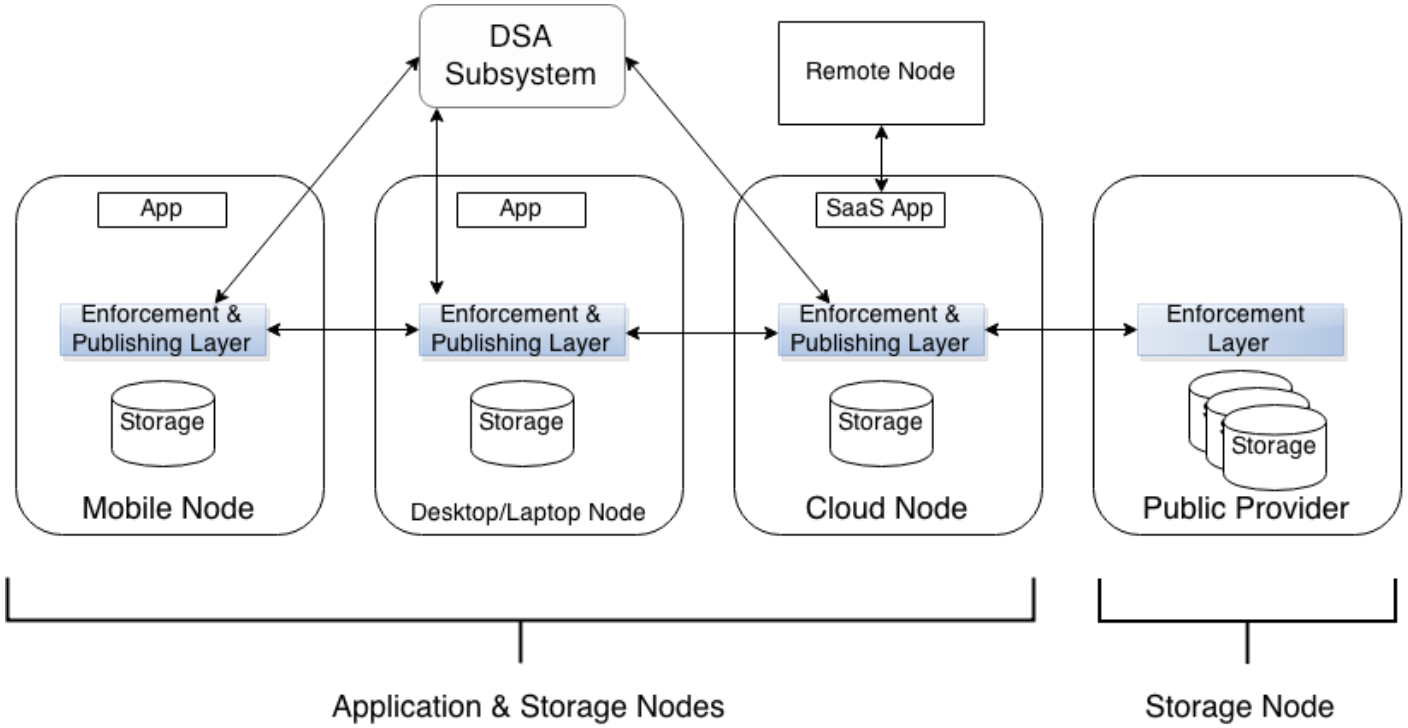
- To provide a framework for **information sharing** among organizations:
  - that permits **confidential** and **secure** operations.
- Strong emphasis on **legal** and **regulatory** aspects:
  - allow organization to effectively deploy **contracts** to share data with users, and other organizations;
  - e.g., Data Protection Directive (**Directive 95/46/EC**).
- Automatically **enforce** the requirements and constraints over their **subsequent** use.
- http://www.coco-cloud.eu/

Contract

- A **Contract**, or **Data Sharing Agreement** (**DSA**), is a formal document that regulates how organizations and/or individuals share data.

Enforcing the DSA

- We introduce an **abstraction layer (Enforcement & Publishing Layer)** as an uniform enforcement component that enforces DSAs through Cloud services and mobile devices.

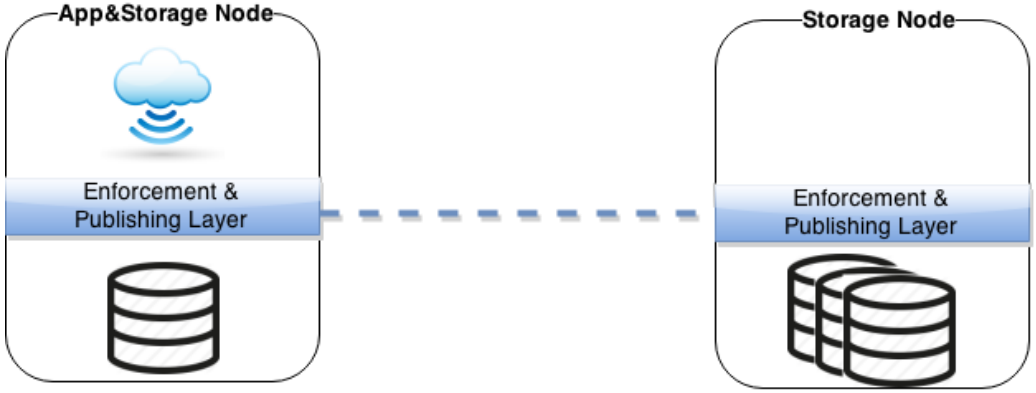


Enforcement & Publishing Layer

- Such a layer focuses on **the enforcement of the policies** resulting from the DSA:
  - **access rights** to the data,
  - **obligations** associated with data usage.
- **Different implementation**:
  - some of the services may be implemented **locally**;
  - others may be implemented by having recourse to **remote services** (e.g., SaaS).
- **Different scenarios** of use:
  - **deployment**: consumer, enterprise, Cloud, multiple;
  - **Pilots**: Mobile, PA, healthcare;
  - **trust domains**: trusted/untrusted/partially trusted;
  - **node types**: Cloud/desktop/mobile/storage nodes.

Cloud Nodes

- **Cloud as application and storage nodes**, for access and use of the data:
  - instantiations of a generic node that includes both applications to access the protected data and storage space.
- **Cloud as a storage service nodes**:
  - instantiation of a provider storage node where the content of protected data is not processed nor accessed by provider applications, and data are only stored and replicated.



Cloud as Application and Storage nodes

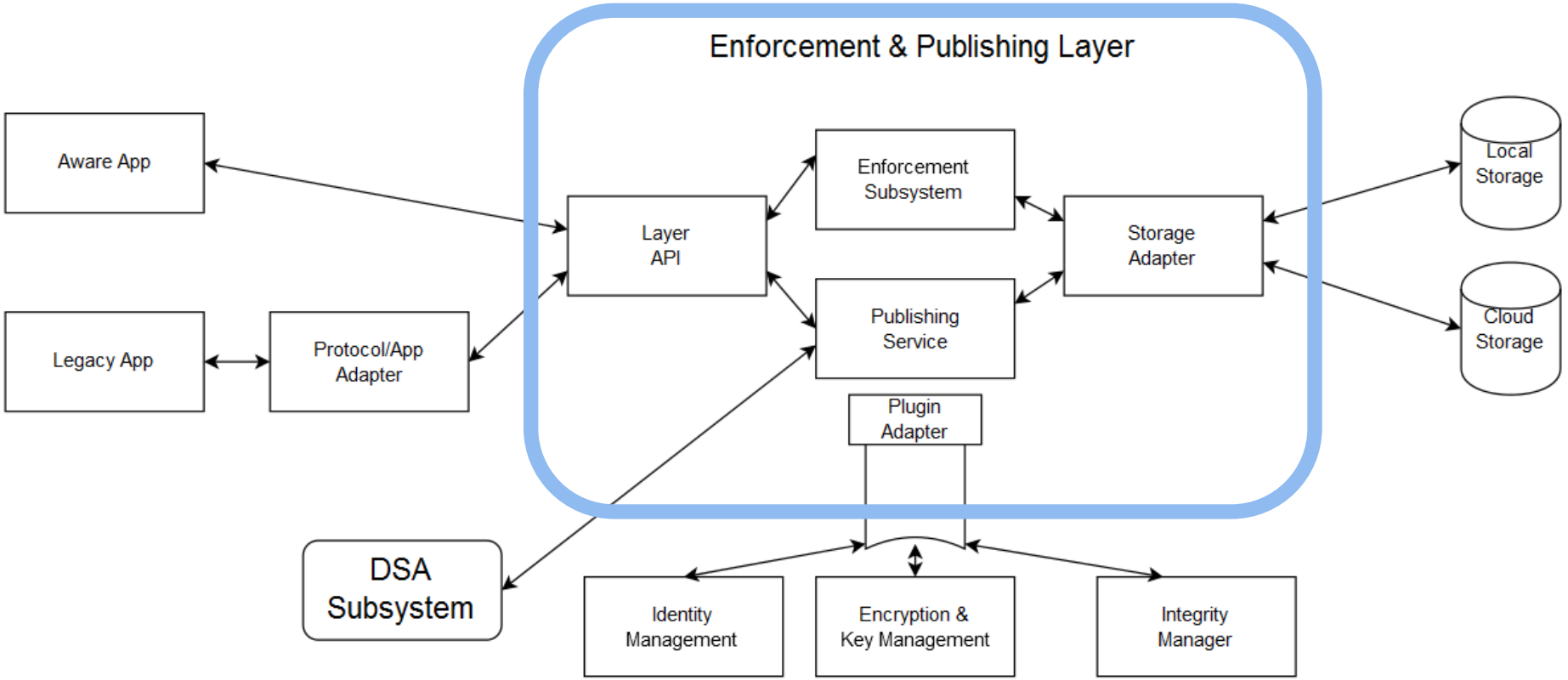
- **Mobile Node**: this is any mobile device that includes the Layer.
- **Cloud Node**: this is a SaaS node, accessed remotely by an user device, and that includes the Layer to access its functionalities.
- **Desktop/Laptop Node**: this is a standard desktop or laptop computer that includes the Layer.

- From an architectural point of view, all the previous nodes are an **instantiation** of a **generic node**:
  - they offer the **same set of functions and features**.

Different Type of Access

- Node type is related to the **different types of access**, **enforcements** and **threats**.
- A node containing **applications** needs to access the content of the data exchanged, may change it, create new data, share or disseminate them further:
  - the layer mediates between these applications and the protected resources by **enforcing** all policies on **content** access specified in the DSA.
- **Storage providers** do not need to access the information content, and they are exploited for storage capabilities only.

Enforcement & Publishing Layer Functions



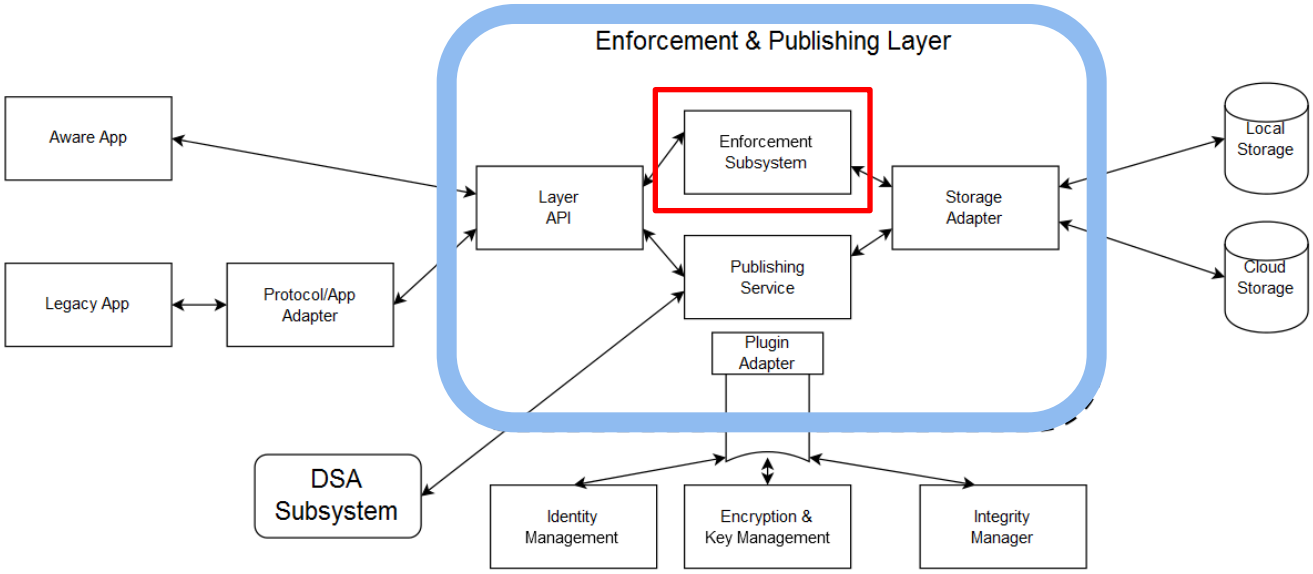
Enforcement & Publishing Layer Functions

The enforcement and publishing layer is **responsible** for:

1. the **enforcement** of DSA (through the enforcement subsystem);
2. the **publishing** of new object(s) under DSAs (through the publishing service);
3. the protected objects can be stored locally or remotely through the **storage adapter**, or sent to another node of the architecture;
4. defining the **layer API** (aware or legacy apps).

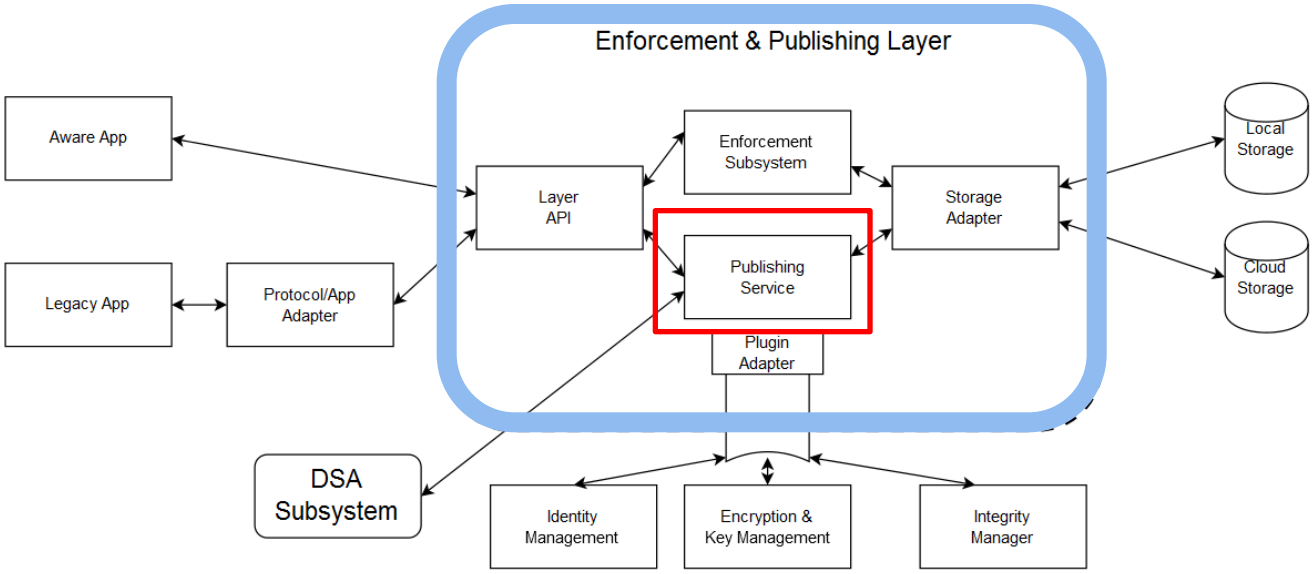
Enforcement Subsystem

- The enforcement subsystem is responsible of **enforcing the attached DSA policies**.
- Examples are enforcement of **obligations**, **continuous authorizations** and managing **mutable attributes** (e.g., attribute which counts a number of data copies in the system).

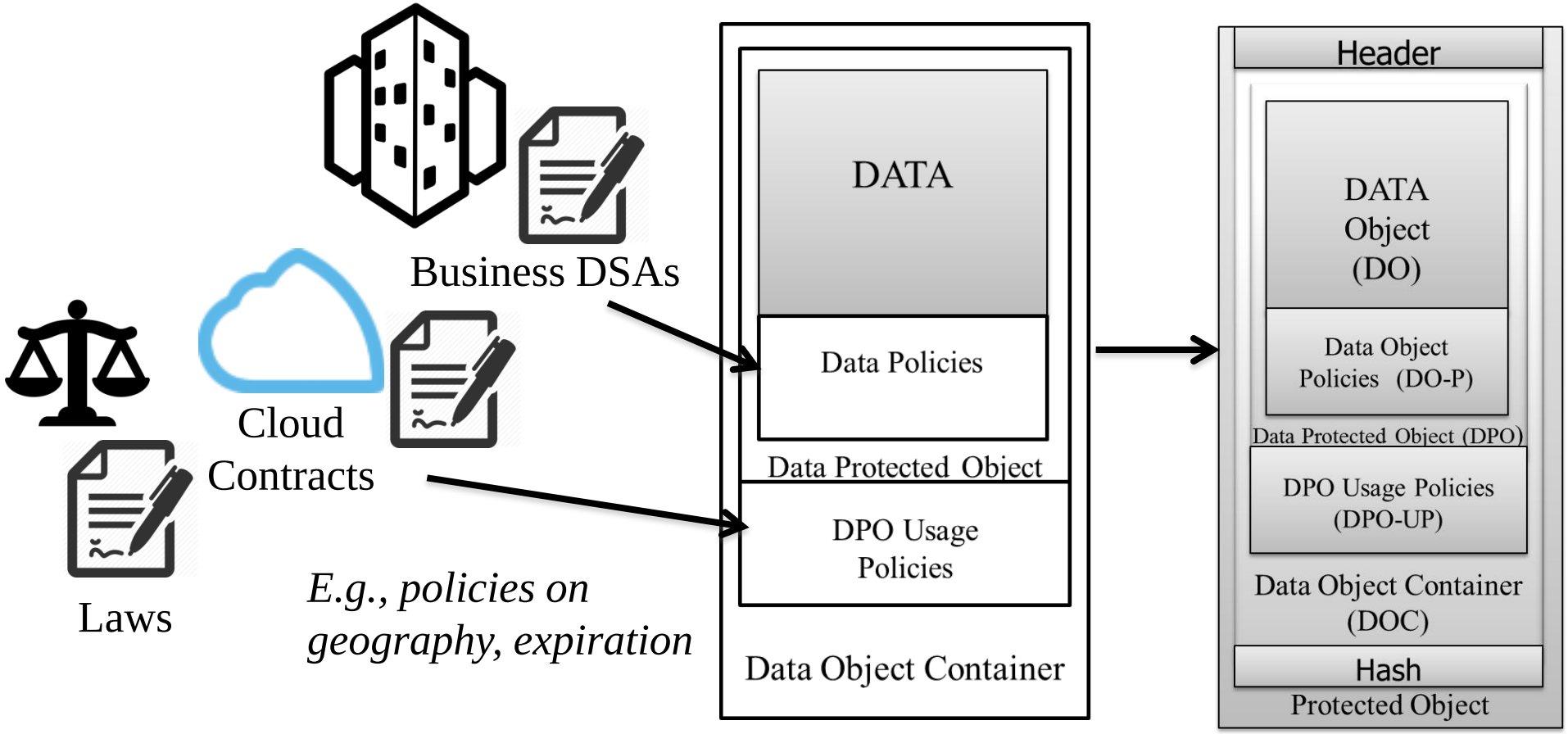


Publishing Service

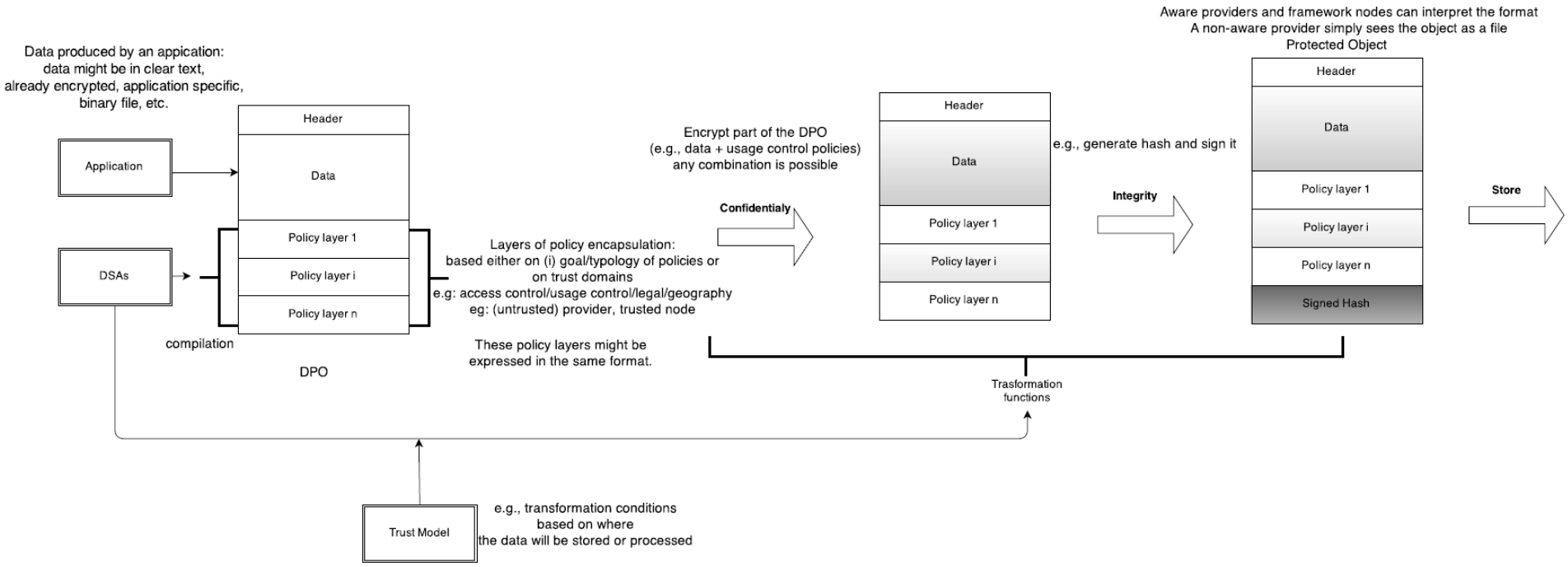
- Used to **publish new objects** (files) **protected** under the relevant **DSAs.**
- When such a protected object is created, it is **ready to be consumed** on any other enabled node.



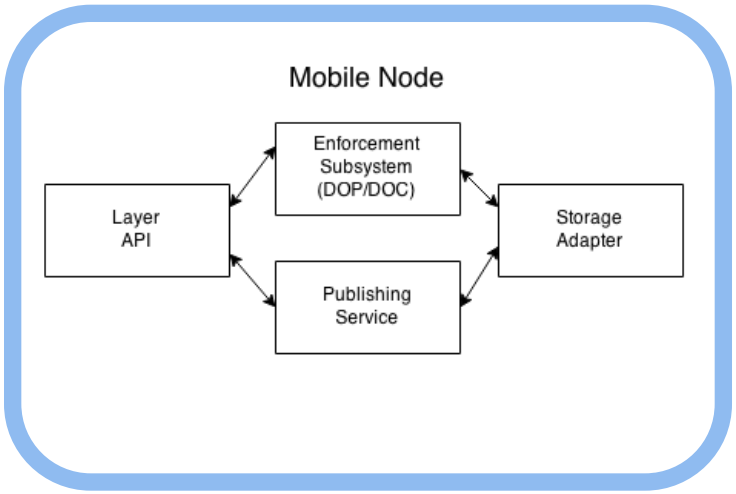
Creating and Securing the Protected Object

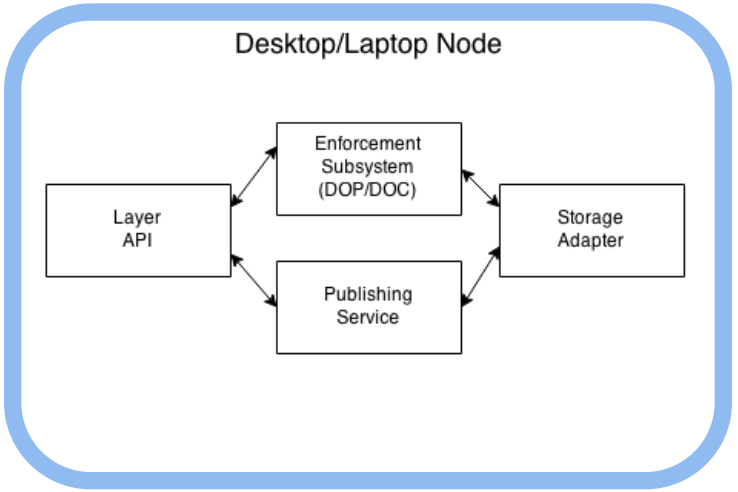


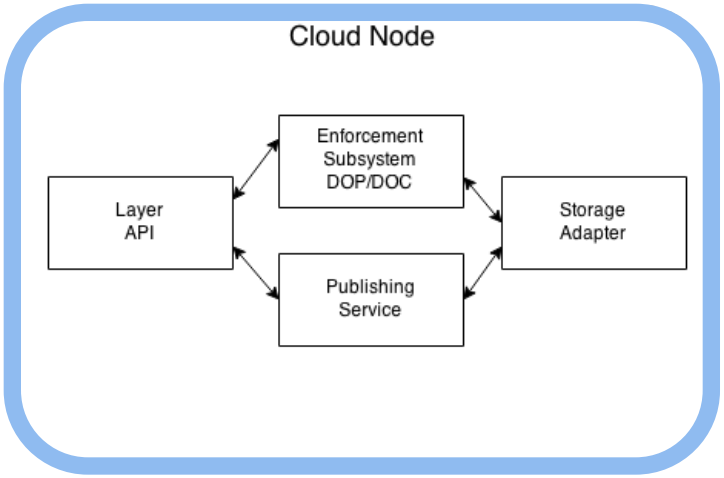
Creating and Securing the Protected Object

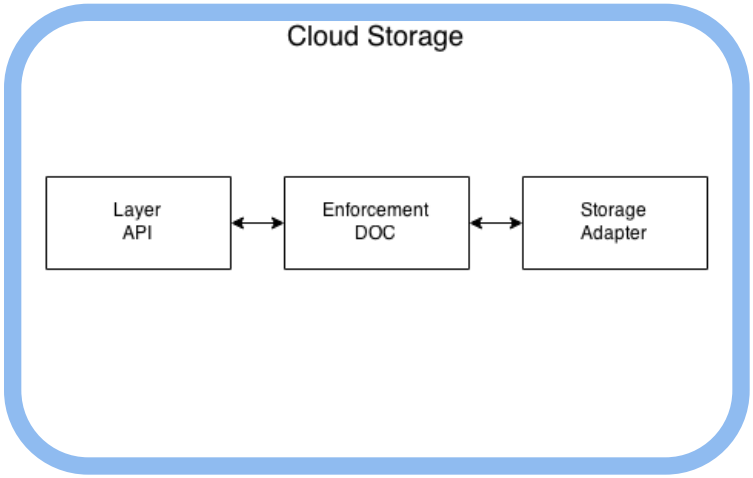


Mapping of Enforcement Functions

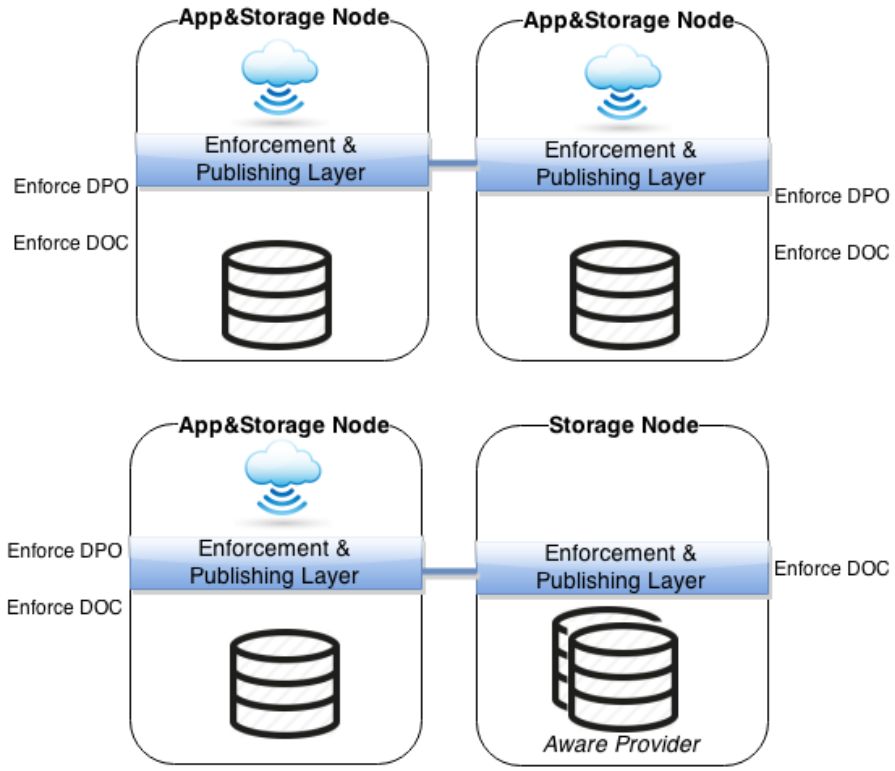


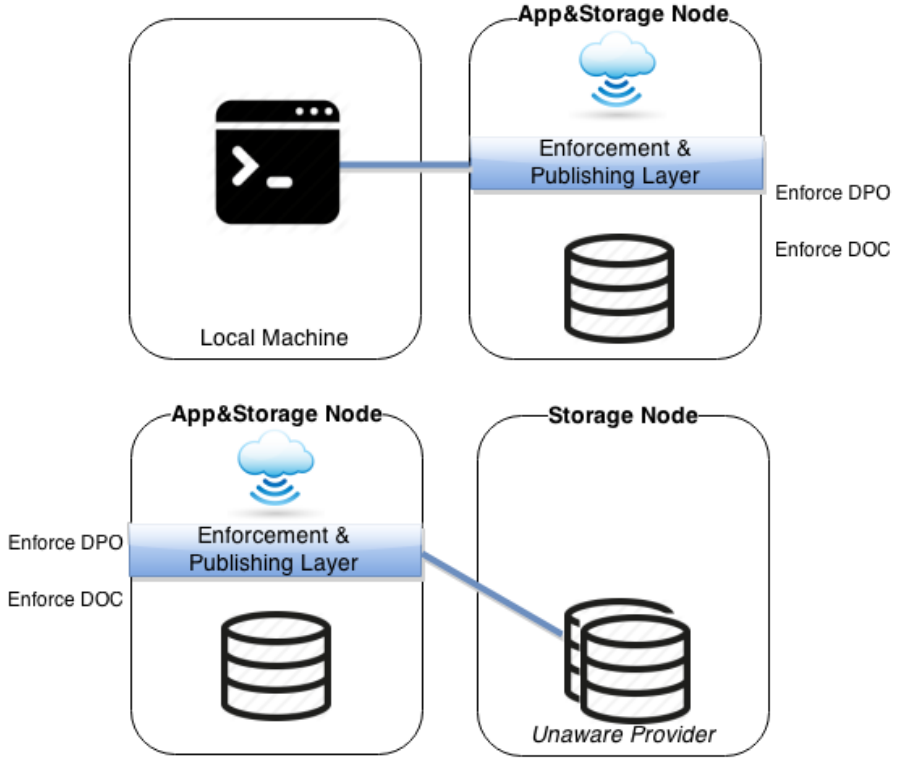




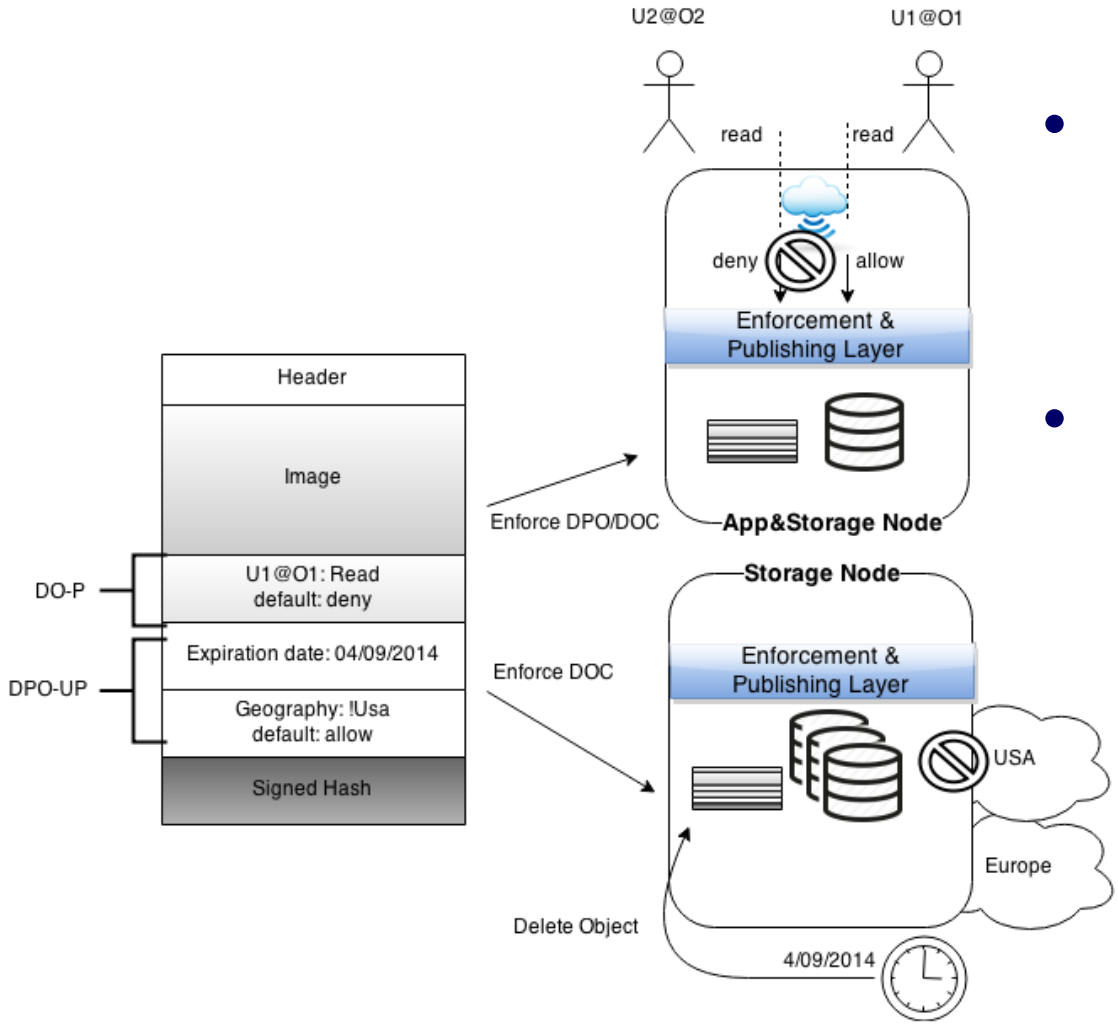


Enforcing the Policies





Protected Object Usage Example



- **Different policies**:
  - data policies for app&storage nodes only;
  - usage policies for any node.
- **Nested layers of encryption:**
  - only data;
  - data and policies;
  - whole container.

Concluding Remarks (I)

- **Automatically enforceable contracts** are needed to establish the trust relationships between the parties, their expectations, rights and duties:
  - translated into the usage control policies to be enforced.
- A **common middleware layer** is necessary, not only to hide the distribution aspects, but also to hide how services and functionality is brought to the point of use:
  - **transparently** offer all the functionalities on all devices even if some of these may be offered remotely,
  - **abstract** from how the services are provided.

Concluding Remarks (II)

- Data protected within **containers** and associated with policies.
- Different policies may apply
  - to the **management** of an opaque (i.e., encrypted) container,
  - to the **usage of the data**, each needing to be enforced by a different provider.
- This leads to the **encapsulation of policy layers**:
  - each layer contains (only) those policies that apply to the management of its immediate contents.

Ongoing and Future Work

- **Trust models for the Layer:**
  - definition of the **conditions/assumptions** and **verification procedures** under which the Layer can be trusted.
- **Threat modelling** for the whole architecture:
  - e.g., **STRIDE** (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege).
- **Key management:**
  - different scenarios based on **trust** and **user requirements**.

Thank You! Questions?