| | |
|---|---|
| **Project Number:** | **FP7-257123** |
| **Project Title:** | **CONVERGENCE** |
| **Dissemination Level** | **Public** |
| **Deliverable Number:** | **D4.3** |
| **Contractual Date of Delivery to the CEC:** | **31.07.2012** |
| **Actual Date of Delivery to the CEC:** | **22.10.2012** |
| **Title of Deliverable:** | **Final Definition of the Versatile Digital Item** |
| **Workpackage contributing to the Deliverable:** | **WP 4** |
| **Nature of the Deliverable:** | **Report** |
| **Editor:** | **Giuseppe Tropea** |
| **Authors:** | **Nicola Blefari Melazzi, Andrea Detti, Giuseppe Tropea (CNIT), Maria Teresa Andrade, Helder Castro (INESC), Angelo Difino, (CEDEO), Angelos-Christos Anadiotis, Aziz Mousas, George Lioudakis, Dimitra Kaklamani and Iakovos Venieris (ICCS)** |
| **Abstract:** | **This deliverable describes the design of naming schemes, at all levels of the CONVERGENCE architecture, for the purpose of identification, location and verification of our information units.** |
| **Keyword List:** | **Information Centric Networking; caching; digital signature; naming schemes; performance evaluation.** |

# Executive Summary

Given the layered architecture of CONVERGENCE (network, middleware), we need to identify two naming schemes, one for the network and one for the middleware. For the sake of generality, in previous deliverables (D4.1 and D3.2), we assumed to have two different schemes for naming middleware-level information units (VDI identifiers), and network-level information units (Named-Data CIUs identifiers, or NIDs).

As regards the middleware, the VDI identifiers, must follow MPEG-21 Digital Item Identification rationale and schemes. However, ISO/IEC 21000-3 DII does not specify an identification system, but provides a standard mechanism to transport industry identifiers within the context of MPEG-21. Specifically, DII does not interfere with the governance of identifier standards and systems, and IDs and their governance are domain-specific. This allows a great flexibility in choosing a naming scheme.

As regards the network layer, the functionality of CoNet raises new requirements for the structure of the names of the network transacted objects (which are generated once their VDI is encapsulated by CoNet). In the CoNet layer naming, routing and security interplay. The structure of names has an impact both on network security properties and on network routing properties. To satisfy security requirements, we have the choice between self-certifying and human-readable names. To satisfy routing requirements we have the choice between names with a hierarchical structure and flat names. The naming scheme has a direct impact on network's efficiency, on the security of content caching functionality, and on the network's scalability.

The first part of this deliverable addresses design choices related to naming issues at the network layer. We present a model for the naming scheme, considering the four possibilities mentioned above (possible combinations of self-certifying/human-readable names and hierarchical/flat names). We analyse the properties of the four possible schemes, combining each of them with matching digital signature features. We study possible choices for digital signature techniques, analysing their performance and assessing in which measure each of them satisfies our requirements in terms of protection of cached content. Then, we make a choice between flat and hierarchical schemes, considering also technological requirements.

In the second part of the deliverable, we discuss the implications of our choice at the network level on the middleware. We find out that it is possible to use the chosen CoNet network identifiers also for the CoMid level, using them to identify VDIs. At the beginning of the project we devised a solution based on a global, static VDI_ID <---> NID mapping. Now we propose a **complete equivalence** between VDI ids and NIDs which we find that it is possible, provided that we introduce some specific implementation steps when deploying the Identify Content service protocol of MPEG-21. This protocol is the service responsible for assigning an identifier to a VDI.

The third part of the deliverable explores the details of the CONVERGENCE-specific data structures that compose the VDI, and gives examples of usage of VDIs (P-VDIs and S-VDIs) as containers of system signalling messages at the CoMid level, as well as Resource descriptors.

An Annex with complete XML schemas concludes the deliverable.

# INDEX

## Glossary

| TERM | DEFINITION |
|---|---|
| IBS | Identity Based Signature is a cryptographic signature scheme where the public key employed for data signing is also the signer's identity. |
| ICN | Information Centric Network |
| RSA | Rivest, Shamir and Adleman's asymmetric cryptography algorithm. |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| Versatile Digital Item | A structured, hierarchically organized, digital object containing one or more resources and metadata, including a declaration of the parts that make up the VDI and the links between them. |

# 1 Goals and structure of this document

Deliverable 4.3 (Final Definition on the Versatile Digital Item) is the third deliverable from WP4: Definition of the Versatile Digital Item. The general goal of WP4 is to define **the Versatile Digital Item**, extending the scope of the MPEG-21 DI by including new classes of objects, including Real World Objects, services and people and supporting new classes of operations. The VDI is the basic unit for transaction used within CONVERGENCE.

This deliverable focuses on the design of the most appropriate naming schemes of the CONVERGENCE architecture, for the purpose of identification, location and verification of our information units: VDIs at the CoMid level, and network data units at the CoNet level. Chapter 2 introduces the problem, and anticipates the solution that we adopt. Chapter 3 presents our model and shows how the naming scheme impacts both on the routing and on the security functionalities of the system. Chapter 4 presents the requirements for the secure operation of caching procedures in our system; it introduces and discusses three reference scenarios and our choices to satisfy said requirements. Chapter 5 discusses inter-layer relationships originating from our choices of naming schemes. Chapter 6 presents additional details and examples of how the design of the VDI supports typical CoMid operations of content description, publication and subscription. Annex 7 presents the final XML schema of CONVERGENCE-specific enhancements to the MPEG-21 Digital Item, which in essence define the VDI. Section 8 lists all the scientific references we studied to produce this deliverable.

# 2 Introduction

Given the layered architecture of CONVERGENCE (network, middleware), we need to identify two naming schemes, one for the network and one for the middleware. For the sake of generality, in previous deliverables (D4.1 and D3.2), we assumed to have two different schemes for naming middleware-level information units (VDI identifiers), and network-level information units (Named-Data CIUs identifiers, or NIDs).

The VDI identifiers must follow the MPEG-21 Digital Item Identification rationale and schemes. On the other hand, the functionality of CoNet raises new requirements for the structure of the names of the network transacted objects (which are generated once their VDI is encapsulated by CoNet). These requirements may be in contrast with the requirements of the VDI identifiers.

To pro-actively face this potential impasse, we had specifically included mapping/translation functionality, from VDI identifiers to NIDs and vice-versa, in the CONVERGENCE architecture. However, we have left the details of such translation mechanism un-specified so far. In previous deliverables (D4.1 and D3.2), we also anticipated that the design of a mapping functionality enables an independence of the two levels but is difficult to deploy. Indeed, any such deployment would resemble a large, distributed DNS-like system, which needs to embed extended cross knowledge of both network topology and location of named-content. Hence, we proposed a global, static VDI_ID <---> NID mapping.

Now we propose a **complete equivalence** between VDI ids and NIDs which we find that it is possible, provided that: i) we introduce some specific implementation steps when deploying the Identify Content service protocol of MPEG-21; this protocol is the service responsible for assigning an identifier to a VDI; ii) we adopt a naming scheme for the network level, which is compatible with some constraints of the middleware identifiers.

In the following, we show how we can design a naming scheme that satisfies the requirements of the network layer and how we can exploit the flexibility of the MPEG-21+MPEG-M standards so that our network choices become compatible with middleware level requirements.

# 3 Naming in ICN Systems

Throughout the life span of our project, and especially in the last year, many Information-Centric Networking architectures have been proposed. The research community is trying to harmonize these proposals, to convergence on a common architecture. A first significant result of this effort is the newly born IRTF Research Group on Information-Centric Networking (ICNRG, http://irtf.org/icnrg), which CONVERGENCE contributed to create. Indeed, the first achievement of the ICNRG has been to decide a common codename (i.e. ICN) for all initiatives that aim at improving the Internet towards a more efficient distribution and manipulation of **named** information. From the ICNRG webpage:

> "Information-centric networking (ICN) is an approach to evolve the Internet infrastructure to directly support this use by introducing **uniquely named** data as a core Internet principle."

The most important research challenges listed in the charter of the ICNRG include “naming schemes for ICN, including scalable name resolution for flat names”.

In today’s Internet, core networking functionalities are disjoint from content security and naming: data routing and forwarding is done at the IP layer, content security is usually dealt with on top of IP, in an end-to-end fashion, while content naming is an issue of the Domain Name System, at the application layer.

In case of an ICN network layer, such as our CoNet, networking, security and naming should interplay in a unique framework: **names are addresses**, hence a bond of naming and networking functionalities; **network nodes cache contents**, hence a bonding of security with networking; and finally **users trust content by name** rather than by server, hence a relationship between naming and security.

Advantages, and possible shortcomings, of ICN, have been discussed at length in [5][11];, Other papers [13][14] study the impact of different naming schemes on the security properties of an information-centric network. An interesting position paper [12] reminds us that the adopted naming scheme has a direct impact on the networks’ efficiency in caching content, which presumably ranges from a simple caching-along-the-default-path, in case of hierarchical names, to a fully distributed cache, in case of flat names.

Our work builds on these studies. To satisfy security requirements, we have the choice between self-certifying and human-readable names. To satisfy routing requirements we have the choice between names with a hierarchical structure and flat names. We present a model for the naming scheme, considering the four possibilities deriving from the combinations of self-certifying/human-readable names and hierarchical/flat names. We analyse the properties of the four possible schemes, combining each of them with matching digital signature

features. We study possible choices for digital signature techniques, analysing their performance and assessing in which measure each of them satisfies our requirements in terms of protection of cached content. Then, we make a choice between flat and hierarchical schemes, considering also technological requirements. Furthermore, we discuss the implications of our choice at the network level on the middleware.

## 3.1 Naming Model

We classify naming schemes for ICN along two main independent axes: impact on network security properties and impact on network routing properties. Along the first dimension we have names that can be self-certifying or human-readable. Along the second dimension we have names that can be hierarchical or flat. See the following Table.

| *Routing* / *Security* | **Flat** | **Hierarchical** |
|---|---|---|
| **Human-readable** | foo.com.videos.video1.mp3 | foo.com/videos/video1.mp3 |
| **Self-certifying** | 0x3fb889fffaddd98 | 0x65de3/videos/video1.mp3 |

**Table 1 — Naming schemes.**

## 3.2 Naming Scheme's Impact on Routing

In order to take shortest-path routing decisions at packet speed, both the size of routing tables and the computational and signalling burden of the distributed routes-computation algorithm must be manageable by today's technology. With current hardware and cost constraints, routing tables of DFZ[1] routers have to be in the order of $10^6$ entries at most [4][5], in order to operate at line speed. On the other hand, the analysis of scalability properties of BGP in terms of algorithmic stability and signalling overhead, confirms that its communication overhead is exponential [6]. Deployment of large MPLS VPNs has introduced higher scalability requirements for BGP: some large Provider Edges hosting many VPNs carry around 2 million routes. Hence, a figure around $10^6$ entries seems to be today's limit, both in terms of size constraints and computational constraints.

Staying within such technological constraints is feasible for today's Internet's IPv4 routing plane, with its hierarchical addressing scheme and small address space, although limits are approaching fast. But we wonder if a route-by-name architecture is still manageable. As of today, Google has indexed more than $10^{12}$ URLs; other estimates [1] of the number of individual content pieces to be handled in ICNs, have figures in the order of $10^{15}$.

1 The default-free zone (DFZ) refers to the collection of all Internet autonomous systems (AS) that do not require a default route to route a packet to any destination. Conceptually, DFZ routers have a "complete" BGP table, sometimes referred to as the "Internet routing table".

In the case of flat names, the whole name uniquely represents each object/content published in the network. Users give the full name to the "fetch primitive" of the network layer, which uses it as a routing address. Routing tables would need to handle a huge number of names/addresses, making the usage of flat names unfeasible, unless:

- Route-by-name lookup is **not** performed at packet speed. It is handled by external, slower, DNS-like resolution systems. Once located, data is then acquired at full-speed. In this case the risk is to under-utilize the transport infrastructure, if data packets are small and need frequent lookups.
- Non-shortest paths are computed, incurring in very high path-stretch, and using DHT-like approaches. This is the classic case where one can freely assign addresses (a perfect fit for flat names that serve the purposes of the upper-layer applications only), and then **topology follows addressing**. Routing tables can be kept very small and lightweight distributed implementations are easy. But a path-stretch of *n* generates n-times more traffic for each and every packet, with respect to a patch stretch equal to one (shortest path).
- Someone invents a truly scalable compact routing algorithm for the Internet, which did not happen up to now [15]. A scalable compact routing algorithm should have the following properties: distributed implementation able to cope with network dynamics; name-independence[2]; reasonable average path stretch (around 1.5) for the first packet name lookup; logarithmic scaling of the rate of signalling messages as the network size increases. Such a routing system would **guarantee scalability** (independently of technology constraints) on top of a totally flat, location-free namespace.

As a consequence, deploying a working ICN with today's technology requires some form of aggregation of the name/address space and a drastic reduction of the number of routes.

It is known that the topological aggregation of today's IP address space allows saving 3 - 4 orders of magnitude of the size of the routing tables of the DFZ (reducing the number of entries from about $10^8$ hosts to about $10^5$). Unfortunately, topological aggregation is not a viable option for named contents, as it would raise a case of mutable semantics applied to permanent names. Location, for example, is a mutable semantic (both in terms of endpoint mobility as well as path changes).

To limit the adversities of topological aggregation of named content, the first, most obvious, idea is to have a "conceptual" aggregation of content, based on the concept of content ownership. Content is grouped under a (vast) number of common top-level roots, each representing the owner of content published under that root. All content produced by the same

2 A detailed discussion of compact routing algorithms is outside of the scope of this document, but suffice to say that in name-dependent algorithms addresses/names are assigned following the topology, while in case of name-independent algorithms addresses/names can be freely assigned, hence can be flat and free of any location semantics.

owner is grouped under the same freely chosen identifier, which uniquely identifies the root, and uniquely represents, at the network level, the *principal* of such content. These identifiers are location free and do not represent addresses related to network topology, or network nodes. Many different principal identifiers can be hosted on the same physical network node. Each and every content assigned to the same principal identifier is, in turn, distinguished by a unique *label* assigned to such specific content. A full name has then the form of Principal/Label. The routing plane disregards the label and considers only the principal identifier. The principal is the only portion of the name used to route-by-name. All content owned by the same principal must be located in the same physical node, as far as the network level is concerned.

Today's Internet is already using a similar aggregation: webpages grouped under a domain name. There already exists a global, unambiguous, location-independent set of names that we can use as prefixes: DNS names. They are in the order of $10^8$; this means a reduction from $10^{12}$ possible objects of all different contents to $10^8$ entries in the routing tables. This number, $10^8$, seems to be the bare minimum for any name-centric design that relies on today's routing algorithms, which ensure stretch-1, at the cost of keeping all possible destinations in the routing tables and of a high signalling overhead (again, see [15]).

In [16] we propose a technique, named lookup&cache, to reduce the number of routing entries, but this is done by distributing routing entries in different systems: a centralized routing engine, that runs on a server named Name Routing System (NRS), logically serves all the nodes of a domain and has all routing entries. Routing tables in each router has only a subset of all possible routes and work as "route cache". If the router does not have the routing entry necessary to route-by-name a given request of content, then it lookups the routing information in the NRS and caches the routing entry in the FIB. Therefore, lookups occur only in cases of route-cache misses, which should be a rare event. This limits the size of routing tables in all routers but require a centralized node with complete knowledge and increases the signalling traffic.

To summarize, scalability of routing protocols depends on two different issues: the first one concerns size of the routing tables; the second one concerns rate of signalling messages, known as the *communication cost* of the protocol, i.e. the number of control messages needed to converge after a topology change [15]. The lookup&cache architecture copes with the first issue and ensures **feasibility** of a wide-scale ICN Internet, although it does not guarantee theoretical scalability in terms of bounded logarithmic behaviour. The design of an efficient ICN routing protocol that limits the rate of routing messages remains an "orthogonal", open issue. But it is worth noting that our lookup&cache architecture does not impose the use of any specific ICN routing protocol: for instance name-based versions of BGP [7] or OSPF [8] could be viable candidates.

In the remainder of this deliverable we assume that a P/L 1-level hierarchical naming scheme, coupled with lookup&cache technology, is a viable solution, with today's constraints, for a deployable ICN system that supports distributing content in a web-like fashion, but with all the benefits typical of the ICN approach.

In the following section we discuss security implications of naming.

## 3.3 Naming Scheme's Impact on Security

Since content is delivered to users not only from the original server but also from distributed caches, in order to avoid cache Denial of Service (DoS) and pollution attacks, all nodes of an ICN are required to verify, **at line speed** in the most general case, each piece of cached content in terms of its:

- **integrity**: data has not been modified;
- **provenance**: source is authentic, i.e. the principal identified by P has indeed provided the data;
- **relevance**: data corresponds to the name employed, by the user, to fetch it.

Obviously, the identifier of the principal and the name given to content, play a central role in such verification operations[3].

These are both DoS and cache pollution issues, not an endpoint correctness issue: the endpoint consumer can verify the correctness of data, too, because it knows the key. But if the ICN cannot verify in real-time integrity, provenance and relevance, then the ICN system may repeatedly deliver false data (and thus not be able to reliably deliver the correct data).

If we assume that each publisher, who is the principal of the corresponding network content, has a real-world identity and a public key that she uses to sign published content, then digital signature verification is the solution to the above requirements.

There are two main naming systems proposed in the ICN literature. The first one [14], which resembles today's DNS names, uses human-readable names. A variety of techniques can allow users to know the public key (ranging from personal contacts to webs-of-trust to PKIs), but, for the network to be aware of this key, it requires a global, human-managed PKI, dedicated to network operation, which binds names to keys.

The second naming system [13] uses self-certifying names. Here, the key is bound to the name itself (the name being the key itself, or a hash of the key), so the network needs not use

3 It is worth noting that confidentiality (data not read by others) remains an end-to-end issue, dealt with by data encryption, which we do not consider here.

a PKI. These names are not human-readable, so consumers must use other techniques (e.g., search engines, personal contacts, webs-of-trust) to determine the name of the content they want.

As clearly explained, again in [13], there is a duality between the two approaches. Both of them use external mechanisms for one binding; however, for the first approach (where human readable names are integral to network functioning) the external binding is between a human readable name and its crypto key; for the other approach, (where self-certifying names are used as addresses for network functioning) the binding is between human names/descriptors of content and its self-certifying name.

In the following chapter we propose a unified data-packaging scheme to evaluate meaningful combinations of digital signature techniques and corresponding choices of name types (human-readable, self-certifying) for the principal identifier P.

# 4 Signature Verification for Named Content

In our model, every data *D* is packaged together with a header *N*, which represents the name of that content. *N* is composed by a hierarchical concatenation of a **unique**, fixed-size, identifier of the principal, and a Label describing the content, in the form P/L, such that P represents a unique routing prefix. The resulting content *C*={*N*, *D*} is then digitally signed by the principal by means of her public key. The signature *S*, plus any other information needed by the router in order to verify on-line *C* against *S*, is included at the tail, in a verification block *V*. Each network packet *P* is then {*C*, *V*}, where *C*={*N*, *D*} and *V*={*S*, *INFO*}.

| | | |
|---|---|---|
| ***P***acket | ***C***ontent | ***N***ame = P/L |
| | | ***D***ata |
| | ***V***erification block | ***S***ignature of Content |
| | | additional ***INFO*** |

**Table 2 — Data-packaging model.**

In this model, the name is part of the digitally signed material; hence, checking the **relevance** of the packet is always **automatically guaranteed** because it is not possible to change the packet's name without a failure of the digital verification.

## 4.1 Scenario

Let us now assess three different solutions for the same scenario. The scenario is:

- The owners of Foo wish to use an ICN to distribute their video content, identifying their contents in the form of *foo.com/video1.mp3*, and placing contents in servers run by Telco.
- The ICN has to route requests, beginning with *foo.com*, to the Telco servers, and additionally serve replicas, from intermediate caches, whenever possible.
- Customers trust Foo by name (something like *foo.com*) and are willing to enjoy its video material, if it comes from such trusted a source.

The key requirement of the scenario is the "trademark" value that the name *foo.com* assumes: owners of Foo know that their customers trust the name of the brand, and they want to trademark it, so that the name is used in telecommunications networks to fetch their content, i.e. they want to buy exclusive and certified usage of such a name for Internet purposes. The ICN approach guarantees, to the company, that *foo.com* itself represents the network address of any content published by Foo.

We have investigated the usage of Identity Based crypto techniques [17] for the ICN cache verification scenario, depicted above. ID-based signatures seem extremely promising in this context since in ID-based cryptography, a publicly known string representing an individual or organization is used as a public key. The public string could include an email address, a domain name, or a physical IP address, which would allow users to verify digital signatures by using only public information, such as the user's identifier.

Hence, in the following, we compare three solutions that differently combine naming schemes and "traditional" verification vs. ID-based digital signature verification.

#### *4.1.1 Human-Readable & Traditional Verification*

Owners of Foo go to a centralized Trademark Authority TA and buy the unique trademark *foo.com*, receiving a signed receipt that certifies that they can use such a name. They then go to one, of several, trusted Certification Authorities CA, and, by showing the receipt, receive a digital certificate that links *foo.com* to a public key. When they publish *video1.mp3*, they bind *foo.com/video1.mp3*, as name *N*, together with the video data *D*, sign such content, and include signature *S*, and the previously received certificate, in the verification block *V*. The signature is a traditional one, constructed on the widely employed RSA [19] or ECDSA [18] algorithms.

Whenever a router needs to cache content in transit, it needs a digital certificate to verify integrity, relevance and provenance of *C* using it together with *S*. It extracts them both from *V*. See **Figure 1**, which illustrates this first case.

It is worth observing that, in this case, secure operation of in-network caching requires the presence of a CA infrastructure.

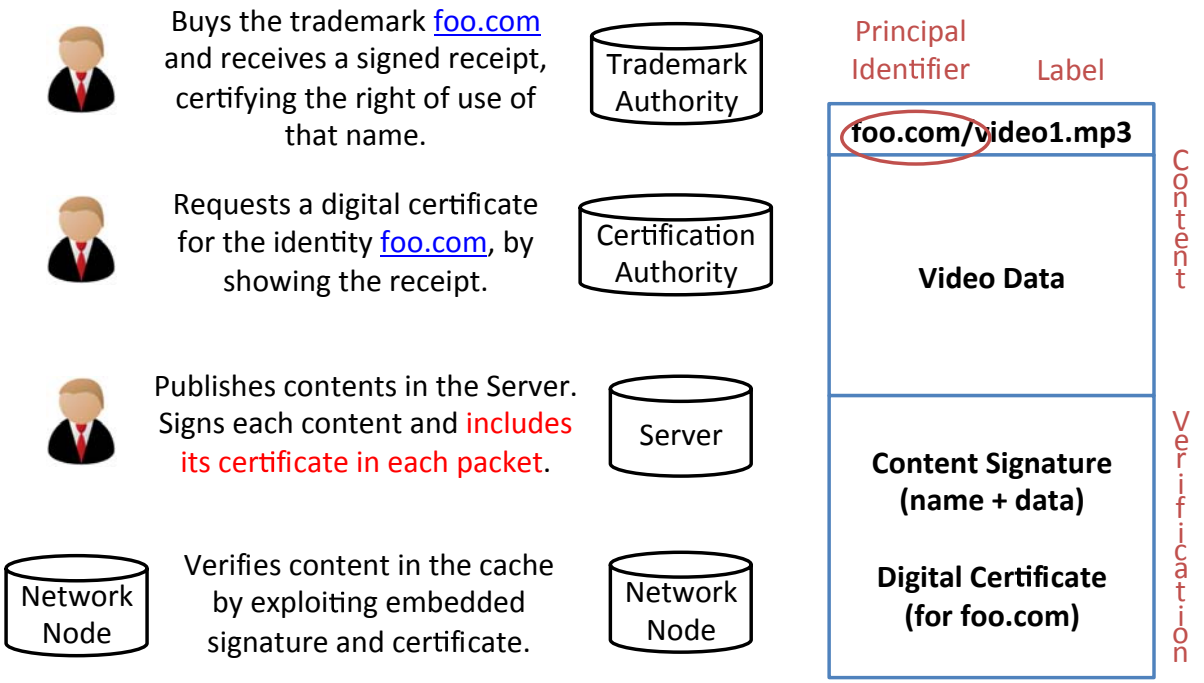


**Figure 1 – Human-readable names with traditional signature verification.**

#### *4.1.2 Human-Readable & ID-Based Verification*

Owners of Foo go to a centralized Trademark Authority TA, and buy the unique trademark *foo.com*, receiving a signed receipt that certifies that they can use such a name. They then go to one, of several, Key Generators (KG), and, by showing the receipt, receive a {public, private} key pair where the public key is exactly equal to the string *foo.com*. When they publish *video1.mp3*, they bind *foo.com/video1.mp3* as name *N* (and in this case the principal identifier P, effectively is Foo's public key), together with the video data *D*, sign such content, and include signature *S* and a KG identifier in *V*. The signature algorithm is usually based on the discrete logarithm problem and elliptic curves [10]. Each router must carry some additional verification information (cryptographic material and parameters which are globally known), specific to the "almighty" KG. Please notice that there can be many global and competing KGs, and it is up to the user to choose a preferred one.

Whenever a router needs to cache content in transit, it verifies the integrity, relevance and provenance of *C* by extracting *S*, the public key (from name *N*), and the KG id, using the latter to point to the correct crypto parameters of the KG chosen by the user. See **Figure 2**, which illustrates this second case.

It is worth observing that the secure operation of in-network caching in this case requires the presence of a PKG infrastructure that, however, could also be distributed [20], since there exist schemes that allow a certain degree of distribution in the KG architecture. Description of the functioning of such crypto schemes is beyond the scope of this deliverable.

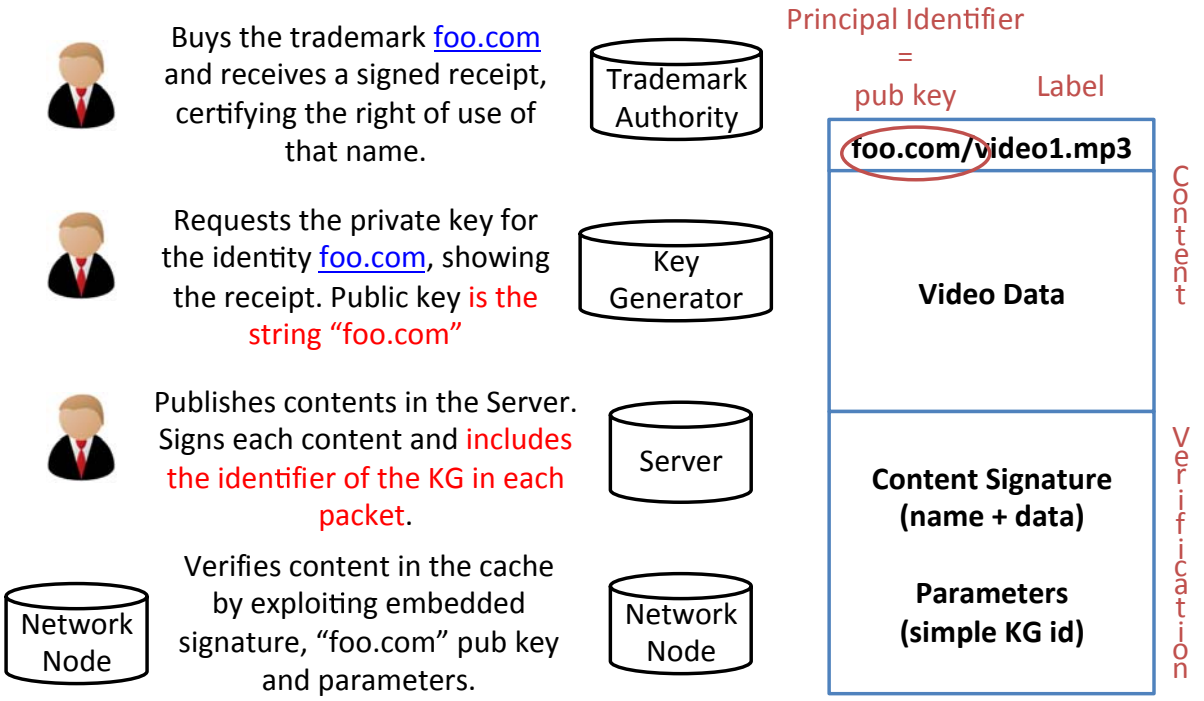


**Figure 2 – Human-readable names with identity-based signature verification.**

#### *4.1.3 Self-Certifying & Traditional Verification*

Owners of Foo self generate a unique, random key pair {public key, private key}, where, just as an example, PUBKEY=0xA00F. When they publish *video1.mp3*, they bind *0xA00F/video1.mp3*, as name *N* (i.e. they use the non-mnemonic public key as principal identifier), together with the video data *D*, sign such content, and include signature *S* in *V*. No additional information, other than the signature *S*, is needed in the verification block in order for the router to verify the integrity, relevance and provenance of the packet. It just needs to extract signature *S* from *V* and use it in conjunction with the public key (extracted from name *N*).

Owners of Foo may, optionally, go to one, or more, of the several external mapping services, which they know users trust, to register a mapping between the public key *0xA00F* and the string *foo.com*, which represents their trademark in the network, possibly authenticating, to the service, with signed challenges. See **Figure 3**, which illustrates this third case.

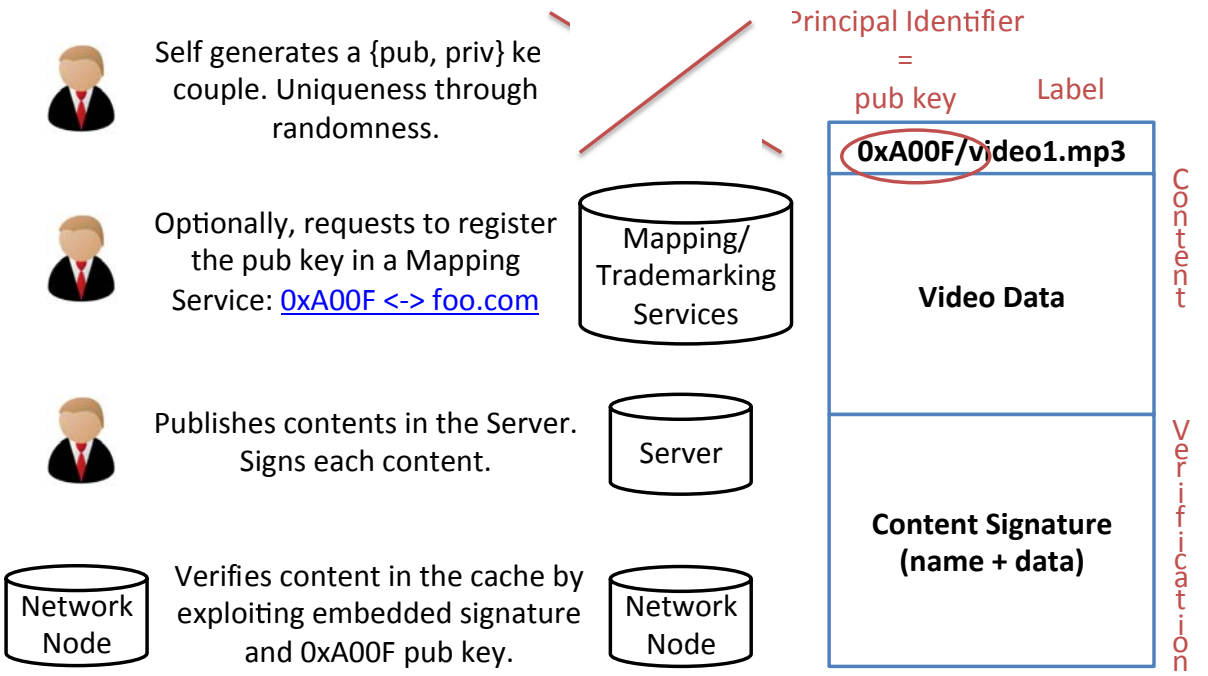


**Figure 3 – Self-certifying names with traditional signature verification.**

To ensure certified naming resolution from *foo.com* to *0xA00F*, users need to do a DNS-like resolution step, before accessing any content by its trademarked name. They accomplish this by employing one, of several, mapping services that they trust (or they trust Foo having indicated). There can be several such services, both certified or not, ranging from user bookmarks to complex hierarchical Attribute-Based certification authorities.

## 4.2 Discussion

The model that we have introduced in the above section clearly shows how, by employing digital signature verification, one can fulfil the requirements for a properly operating ICN system that secures named content at the network caches, thus avoiding DoS attacks and pollution of caches with invalid content. Still, just saying than the solution is to "digitally sign each named-data CIU" is not enough. Which naming scheme, and which signature verification best solve the challenge?

Traditional digital signature verification is based on RSA or ECDSA technologies. Solid implementations of these schemes are readily available for testing and assessment. On the other hand, novel techniques that rely on ID-based crypto techniques (e.g., pairing-based), are much more difficult to assess, in terms of performance, and are both very experimental, and not readily available or openly implemented.

To further help making a choice, we conducted a comparative quantitative performance analysis on the involved techniques; and we tested ID-based signature (IBS) verification

algorithms that employ Schnorr-like signatures concatenation (again, see [10]), instead of pairing cryptography, making this scheme very competitive compared to "traditional" signatures (i.e. RSA, DSA, ECDSA). It is worth noting that CONVERGENCE invested effort in developing, from scratch, the crypto algorithms, when an open-sourced implementation was not available.

The following Table reports sizes of the verification block *V* in the different cases. Comparison is done by using configuration parameters that provide the same level of security. Specifically, for RSA signature we use keys of 1024 bits; for ECDSA signature, we use an elliptic curve over a 160bit prime field (NIST secp160k1 [9]); as said, for ID-based signature we use the same elliptic curve and the approach proposed in [10], where the (slow) pairing computation is not required. We evaluated verification speed using OpenSSL API and an Intel I7 processor @1.8Ghz. We point out that, in case of combinations 1a and 1b, it is needed to verify the digital certificate, too. Accordingly, we assume that such certificate verification consumes a time equal to the verification of the signature *S*. For instance, in case of combination 1a, we have 0.06 ms to verify signature *S* and 0.06 ms to verify the digital certificate, which sums up to 0.12 ms as shown.

| Combination | Add. INFO | Sign. (bits) | Add. INFO (bits) | Verification time (ms) |
|---|---|---|---|---|
| 1a- H.R. with RSA | Certificate | 1024 | 2048 | 0.12 |
| 1b - H.R. with ECDSA | Certificate | 320 | 408 | 0.58 |
| 2 - H.R. with I.B. | PKG id | 506 | 2 | 0.33 |
| 3a - S.C. with RSA | None | 1024 | 0 | 0.06 |
| 3b - S.C. with ECDSA | None | 320 | 0 | 0.29 |

**Table 3 — Sizes of the verification block fields and verification time in case of Human Readable (H.R.) and Self-Certifying (S.C.) names, and RSA, ECDSA and Identity-Based (I.B.) signature schemes.**

Our model shows that, when discussing different possible solutions depending on the combined name structure and crypto algorithms, a trade-off choice emerges. The trade-off is between bandwidth overhead, speed of verification, and requirements in terms of external infrastructures, which characterizes each solution.

A detailed discussion on the implications of these figures, in terms of the capability to validate incoming packets at line speed, will be included in future WP8 deliverables. The aim

is to decide what to keep in cache and what to discard. For the time being, it suffices to say that within the "traditional" techniques (RSA and ECDSA), a trade-off exists between size (length of signatures and names), and verification speed.

Techniques based on elliptic curves (ECDSA and ID-based) have a lower bandwidth overhead with respect to RSA, but have a quite slower verification performance. We observe that overhead introduced by the combinations 1a, 1b and 3a is rather high, considering that an ICN chunk has a relatively small size, e.g. 4kB. In particular, RSA implies a waste of bandwidth totally un-acceptable in real-life deployments, and is not considered as viable. Any large overhead may vanish the ICN benefits of traffic reduction. Consequently, architectural choices seem to favour human-readable names with identity-based signature (2) or self-certifying names with ECDSA (3b). However, in the Identity-Based scheme the network cannot operate as a self-standing entity, but requires an external, human-managed PKG infrastructure, and this may complicates network deployment.

The following Table summarizes all the important indicators discussed and collected so far, in a comparative way, with the goal of motivating a decision of the best trade-off in the different cases.

| | Network address uniqueness | Provenance | Mnemonic | Transmission overhead | Stand-alone operation | Trademarking |
|---|---|---|---|---|---|---|
| **Human Readable + ECDSA** | Trademark Authority | Certification Authority | YES | High | CA needed for new Principal | IN NETWORK |
| **Human Readable + IBS** | Trademark Authority | Key Generator | YES | Low | KG Needed for new Principal | IN NETWORK |
| **Self-Certifying + ECDSA** | Automatic | Automatic | Mapping Service, Bookmark, etc. | Low | YES | EXTERNAL |

**Table 4 — Comparison.**

The first case shows how embedding digital certificates in each packet results in high utilization of bandwidth for the purpose of verification. Usually this is unacceptable in Internet-scale deployments.

The only two viable alternatives appear to be a coupling of human readable names and IBS techniques, or self-certifying with ECDSA. For the first alternative to take off, ID-based

crypto techniques need a wider acceptance in the real world of networking. As of today, issues or misconceptions concerning the architecture of KGs limit their applicability.

The ICN system, in the third case, is not able to guarantee trademarking of names/addresses as a network primitive, but, on the other hand, it can be deployed and can function without the need of an external PKI infrastructure managed by humans.

Essentially, we are trading the possibility to have Trademarking for the possibility to operate the network without key infrastructures governed by humans (see Stand-alone vs. Trademarking columns).

In all cases, verification speed is, probably, not high enough to check all content that enter routers at line speed. Advanced caching policies for improving this specific problem are currently the subject of our (and of ICN networking research's) study. Our major finding in this specific area, which at a first glance may appear counter-intuitive, is that *the limited speed of practical signature verification algorithms does not seem to be a critical problem for an ICN*. Indeed we found that overall caching performance, in terms of cache hit probability, may even *increase* when the usual LRU caching policy is employed, if not all content that enters a router is cached. Details of this result will be given in forthcoming WP8 deliverables.

# 5 From Network to Middleware Identifiers

"Digital Item Identification" (ISO/IEC 21000-3, a.k.a DII) is part 3 of the MPEG-21 standard. The principle of ISO/IEC 21000-3 is to be compatible with existing and new identification schemes, and to enable the use of such identifiers in the context of MPEG-21 applications. Thus ISO/IEC 21000-3 does not specify an identification system, but provides instead, a standard mechanism to transport industry identifiers within the context of MPEG-21.

This is done by means of the following tags:

- `dii:Identifier` – associates the ID of a Digital Item with the Digital Item itself;
- `dii:RelatedIdentifier` – enables the association, to a specific Digital Item, of an ID of another DI, which is "related to" the earlier Item.

Amendment 1 to ISO/IEC 21000-3 further clarified the RelatedIdentifier concept, and defined a method to make the nature of the relationship between an Identifier and a RelatedIdentifier explicit. The terms for typing the relationship are provided by ISO/IEC 21000-6.

CONVERGENCE has innovated the current status of the standard, by devising two extensions, that contribute to render the DI concept much more "versatile", in the context of identification of digital resources.

First, we have proposed a solution (see deliverable D4.1) to the acknowledged fact that another consumer of the same DI may need to distinguish between "variants" (or "updates") of a single DI. Such a solution takes the form of a **sequence identifier** for VDIs.

Second, our project has introduced a novel mechanism for declaring semantic relationships between digital items. Said mechanism, whilst adopting the MPEG-21 principles (in line to what has been done for Amendment 1) semantically enriches DIs through **the use of RDF/OWL ontologies**. The devised mechanism has been submitted as Amendment 2 to ISO/IEC 21000-3 [AMD2], and is in the later stages of acceptance.

These innovations do not alter the original principles of MPEG-21 part 3, though (and do not make the syntactic structure of a Versatile DI any different: a VDI is a DI, still). It is of special importance, to our present discussion, to highlight which principles drive ISO/IEC 21000-3 and the **governance** of the identifiers. Specifically:

- DII does not interfere with the governance of identifier standards and systems
- IDs and their governance are domain-specific
- The identifier systems themselves:
    - Provide their own semantics

- Provide their own conformance scheme
- May provide the means for resolving an Identifier to its reference metadata.

Thus, it would seem that transporting CoNet network identifiers up to the CoMid level, and assigning them directly to VDIs, is not only feasible but perfectly inline with MPEG-21 DII principles.

Although this is true, we need to revise the detailed mechanisms for VDI identification, (i.e. the procedure any digital item goes through in order to get its own identifier), to understand how it is possible to directly use network identifiers as VDI identifiers.

Specifically, the MPEG-M service, responsible for assigning identifiers to VDIs, is the Identify Content Elementary Service. This Elementary Service enables a User to obtain an Identifier to a Digital Item or any of its component elements. The response should be an identifier, an identified DI or a reference to it. The Service Provider, providing the Identify Content Service, keeps a store of Digital Items, after their proper registration is done by the Identify operation. The Identify operation may be used in two different situations: to register new content, or to obtain the already existing identification.

For the purposes of the current discussion, we shall focus on the first situation. Steps are as follows:

1. A User sends an `IdentifyContentRequest` message. This message should contain a DI to be identified.
2. The SP sends back a message containing an identifier, or a hash of the identified data (or an identified DI or a reference to it, in case of situation two) using an `IdentifyContentResponse`.

Let us introduce a simple example: a digital item, representing a book, is authored. The VDI is sent to an Identify Content Service owned by an ISBN Agency or any other company responsible for issuing ISBN numbers in that country. The service scans the VDI, and returns a valid ISBN number to be assigned to the book. Such an ISBN number, staying perfectly inline with the spirit of digital item identification standard, can be used as the unique DI identifier for the VDI.

But what happens when the VDI is published, and exchanged, over a telecommunication network? A concrete need arises to route messages to the nodes where the VDI is stored, and to fetch it.

Either a translation functionality (external to the network), exists, which returns the location of the item, given its identifier, or the identifier itself contains sufficient hints about the item's location, for the network to be able to route (by name) messages to that point.

As anticipated in the above sections, we want to exploit the route-by-name nature of the CoNet, and reject the overly complex hypothetical "translation component" of the architecture.

For this to be possible, at identification time, the VDI id must be composed of two parts:

- a routing prefix (the network part);
- an application specific identifier (the application part).

In our example, the application specific identifier is the ISBN number. The routing prefix shall be the principal identifier *P* we have introduced in the previous chapter. It is the unique identifier of the real-world hosting service of the data, which represents the principal of such data **in the network**. In our case it might, very well, be something like *www.barnes&noble.com, www.telecomitalia.com* or even *www.johnsmith.com*.

Eventually, in accordance to the P/L schema introduced above, we would like our VDI's id to resemble, for example, something like: *www.barnes&noble.com/ISBN817525766-0*.

During packaging, all of the important metadata about the book are bound together inside the VDI, with the exception of the identifier, which is assigned by the Identify Content service. Of course, we have the problem, here, that the Identify Content service in question can never be in position to assign the network part by just scanning the VDI of the book. It knows nothing about network principals. It may be able to infer that the author of the book is John Smith and the copyright holder is Wonderful Editions Ltd., but not that they want to place it in the network under the *www.telecomitalia.com* or *www.barnes&noble.com* names.

Fortunately the MPEG-M protocols have been designed with sufficient flexibility and a future-proof approach, which allows accommodating for such a case.

Just like any other MPEG-M protocol, the `IdentifyContentRequestType` extends `ProtocolRequestType`, which in turn extends `ProtocolBaseType`. If we examine the latter:

```
<complexType name="ProtocolBaseType" abstract="true">
      <sequence>
            <element name="TransactionIdentifier" type="string" />
            <element name="Timestamp" type="dateTime" minOccurs="0" />
            <element name="Entry" type="mpegmb:KeyValueDataType"
                  minOccurs="0" maxOccurs="unbounded" />
            <element name="ApplicationSpecificData"
type="mpegmb:ApplicationSpecificDataType"
                  minOccurs="0" maxOccurs="unbounded" />
            <element ref="dsig:Signature" minOccurs="0" />
      </sequence>
</complexType>
```

We see that it is possible to carry application specific data in any MPEG-M protocol request!

```
<complexType name="ApplicationSpecificDataType">
      <sequence>
            <any namespace="##other" processContents="lax" />
            <!-- May contain any application-specific XML element. -->
      </sequence>
</complexType>
```

It is thus sufficient to embed the network part of the identifier in the identification request, in order to have the Identify Content Elementary Service use it and construct a complete identifier that includes both parts.

This basically means that the CONVERGENCE Application, running on the peer, is configured for publishing content “under” a known (to it) network principal. Each publishing operation, which produces VDIs, tells the Identify Content what the network principal identifier is, and the service incorporates it in the issued-back VDI id. The VDI is then packaged and secured at the CoMid level using said VDI id, and when it is transferred to the CoNet level, the very same NID is employed by the ICN system to store it in cache and route fetch messages to it.

# 6 Convergence Metadata

The Convergence Metadata is the core XML structure inside every VDI that holds system information about publications and subscriptions, necessary to perform CONVERGENCE workflows for propagating, matching and revoking them. The ConvergenceMetadata schema is appended in the Annex and it imports the DIDL-MSX schema and the MPQF schema. The paragraphs below describe the basic elements of the Convergence Metadata schema and their usage.

## 6.1 ConvergenceMetadata element

This is the root element of the ConvergenceMetadata schema and contains the following information:

- **Sequence identifier**, this identifier is used in case of VDI updates
- **VdiKind**, the type of VDI, namely R-VDI, P-VDI, S-VDI
- **Start date**, the start date when the VDI should be accessible
- **Expiry date**, the end date after which the VDI should be invalidated

The ConvergenceMetadata element also contains one additional element, according to the type of VDI: ResourceMetadata for a R-VDI, PublicationMetadata for a P-VDI, and SubscriptionMetadata for a S-VDI. These elements are presented in the following sections, while the example below presents a generic ConvergenceMetadata element:

```
<ns21:ConvergenceMetadata>
<ns21:SequenceIdentifier>urn:sequence_identifier:temp:56aceb27-6db7-4c98-
bd11-caf75c5bfa8f</ns21:SequenceIdentifier>
<ns21:VDIkind>RVDI</ns21:VDIkind>
<ns21:StartDate>2012-08-22T17:35:06.300+01:00</ns21:StartDate>
<ns21:ExpiryDate>2012-10-22T17:35:06.300+01:00</ns21:ExpiryDate>
…
ResourceMetadata/PublicationMetadata/SubscriptionMetadata
…
</ns21:ConvergenceMetadata>
```

## 6.2 ResourceMetadata element

Inside the ResourceMetadata element it is possible to define all kinds of metadata about the VDI.

- **Keywords**
- **Tags**
- **Field/Value**
- **StructuredData**

The information inside this element is exploited for matching publications with subscriptions. The StructuredData element is imported by the didl-msx schema and allows the definition of any kind of XML based metadata, including RDF/XML.

The example below presents a ResourceMetadata element containing "field, value" pairs of metadata:

```
<ns21:ResourceMetadata>
<ns21:Keyword>Lens</ns21:Keyword>
<ns21:Keyword>Canon</ns21:Keyword>
<ns21:FieldValue>
    <ns21:Field>PRODUCT_NAME</ns21:Field>
    <ns21:Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns25="http://www.w3.org/2001/XMLSchema" xsi:type="ns25:string">Canon
24-70mm f/2.8 L</ns21:Value>
</ns21:FieldValue>
</ns21:FieldValue>
<ns21:FieldValue>
    <ns21:Field>WEIGHT</ns21:Field>
    <ns21:Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns25="http://www.w3.org/2001/XMLSchema"
xsi:type="ns25:integer">909</ns21:Value>
</ns21:FieldValue>
</ns21:ResourceMetadata>
```

## 6.3 PublicationMetadata element

This element contains the actual information that is used for the matching procedure and extends the ResourceMetadata element by providing two more mandatory elements:

- **R-VDI id**, the id of the R-VDI that it is being published by this P-VDI
- **Dimension**

The dimension element specifies the fractals where the publication must be announced. Please note that it is possible to define a union or an intersection of fractals using the FractalAlgebra element (see Annex).

The example below presents a PublicationMetadata element containing RDF metadata:

```
<ns21:PublicationMetadata>
<ns21:Dimension>
    <ns21:FractalAlgebra>
        <ns21:Fractal>analysis</ns21:Fractal>
    </ns21:FractalAlgebra>
</ns21:Dimension>
<ns21:RVDIid>urn:eu:convergence:server1/a6f9118c-fdf3-4f87-935f-
0b1606c73e64</ns21:RVDIid>
<ns21:StructuredData ref="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
```

```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:videodescription="http://escom.msh-paris.fr #"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
        <rdf:Description rdf:about="http://escom.msh-paris.fr/c248b5e2-
002d-4990-8bfe-ec235f8c7b41_DescriptionPattern">
            <videodescription:Sign rdf:resource="http://escom.msh-
paris.fr/00f435dd-8309-476f-a0be-c30e381bb8e5"/>
            <rdf:type
rdf:resource="http://escom.mshparis.fr/videodescription.owl#DescriptionPatt
ern"/>
  </rdf:Description>
        </rdf:RDF>
</ns21:StructuredData>
</ns21:PublicationMetadata>
```

## 6.4 SubscriptionMetadata element

This element contains the subscription query of a S-VDI. It has, just like the PublicationMetadata element, a Dimension element defining the fractals where the subscription should be announced and the additional:

- **Query**

The query element is imported from the MPEG Query Format schema and is of InputQueryType. The MPQF schema is a standard query wrapper and can contain any kind of queries. It completely covered our needs for subscription queries as it offers a lot of flexibility to define complex disjunctive or conjunctive queries of any type e.g. SPARQL, XQuery, or field/value.

The example below presents a SubscriptionMetadata element containing a SPARQL query:

```
<ns21:SubscriptionMetadata>
      <ns21:Dimension>
            <ns21:FractalAlgebra>
                  <ns21:Fractal>analysis</ns21:Fractal>
            </ns21:FractalAlgebra>
      </ns21:Dimension>
      <ns21:Query>
            <ns22:QueryCondition>
                  <ns22:Condition
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns22:QueryBySPARQL">
                        <ns22:SPARQL>
SELECT ?x
WHERE
{
?x &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#type&gt;
&lt;http://escom.msh-paris.fr/videodescription.owl#MetaDescription&gt;
.
?concept &lt;http://escom.msh-paris.fr #Value&gt;
&lt;http://escom.msh-paris.fr/49f5a94d-04c0-4e69-ae47-f26e970690b1&gt;
.
```

```
?actor &lt;http://escom.msh-paris.fr #Name&gt; &quot;CRANSAC&quot; .
?subject &lt;http://escom.msh-paris.fr #hasSubjects&gt;
&lt;http://escom.msh-paris.fr/20d6c8e4-76cd-4d8f-ab7f-3b0781a0a4e8&gt;
.
?videoLang &lt;http://escom.msh-paris.fr #Language&gt; &quot;fr&quot; .
?kind &lt;http://escom.msh-paris.fr #Kind&gt;
&lt;http://escom.msh-paris.fr/bd3628f3-3a51-4535-9204-ef0228e532d3&gt;
.
?analysisLang
&lt;http://escom.msh-paris.fr #hasLanguages&gt; &quot;fr&quot; .
}
                        </ns22:SPARQL>
                    </ns22:Condition>
                </ns22:QueryCondition>
            </ns21:Query>
</ns21:SubscriptionMetadata>
```

# 7 Annex - Convergence Metadata Schema

This is the complete specification of the CONVERGENCE-specific information inside VDIs.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:conv="urn:conv:metadata:schema:2011"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:mpqf="urn:mpeg:mpqf:schema:2008"
            xmlns:didl-
msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
            xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
            jxb:version="1.0"
            targetNamespace="urn:conv:metadata:schema:2011"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">

<xsd:import namespace="urn:mpeg:mpqf:schema:2008"
schemaLocation="mpqf_new.xsd"/>
<xsd:import namespace="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
schemaLocation="mpegm-schemas/mpeg/didl-msx.xsd"/>

<xsd:element name="ConvergenceMetadata" type="conv:ConvergenceMetadataType"
/>
<xsd:element name="ResourceMetadata" type="conv:ResourceMetadataType" />
<xsd:element name="PublicationMetadata" type="conv:PublicationMetadataType"
/>
<xsd:element name="SubscriptionMetadata"
type="conv:SubscriptionMetadataType" />
<xsd:element name="FieldValue" type="conv:FieldValueType" />
<xsd:element name="Dimension" type="conv:DimensionType" />
<xsd:element name="FractalAlgebra" type="conv:FractalAlgebraType" />

<xsd:complexType name="ConvergenceMetadataType">
      <xsd:sequence>
      <xsd:element name="SequenceIdentifier" type="xsd:anyURI" />
      <xsd:element name="VDIkind" type="xsd:string" />
      <xsd:element name="StartDate" type="xsd:dateTime" />
      <xsd:element name="ExpiryDate" type="xsd:dateTime" />
      <xsd:choice minOccurs="0">
            <xsd:element name="ResourceMetadata"
type="conv:ResourceMetadataType" />
                  <xsd:element name="PublicationMetadata"
type="conv:PublicationMetadataType" />
                  <xsd:element name="SubscriptionMetadata"
type="conv:SubscriptionMetadataType" />
            </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DimensionType">
  <xsd:sequence>
    <xsd:element name="FractalAlgebra" type="conv:FractalAlgebraType" />
  </xsd:sequence>
  <xsd:attribute name="usage" type="xsd:anyURI" use="required" />
```

```
</xsd:complexType>

<xsd:complexType name="FractalAlgebraType">
  <xsd:choice maxOccurs="unbounded">
            <xsd:element name="Fractal" type="xsd:anyURI" />
            <xsd:element name="FractalAlgebra"
type="conv:FractalAlgebraType" />
  </xsd:choice>
  <xsd:attribute name="operand" type="xsd:string" use="required" />
</xsd:complexType>

<xsd:complexType name="ResourceMetadataType">
  <xsd:sequence>
    <xsd:element name="Keyword" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Tag" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="FieldValue" type="conv:FieldValueType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="StructuredData" type="didl-
msx:StructuredDataType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PublicationMetadataType">
  <xsd:sequence>
    <xsd:element name="Dimension" type="conv:DimensionType" />
    <xsd:element name="RVDIid" type="xsd:anyURI" />
            <xsd:element name="Keyword" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Tag" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="FieldValue" type="conv:FieldValueType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="StructuredData" type="didl-
msx:StructuredDataType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SubscriptionMetadataType">
  <xsd:sequence>
    <xsd:element name="Dimension" type="conv:DimensionType"/>
            <xsd:element name="Query" type="mpqf:InputQueryType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FieldValueType">
  <xsd:sequence>
    <xsd:element name="Field" type="xsd:string"/>
            <xsd:element name="Value" type="xsd:anySimpleType"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

# 8 References


[1] Matteo D'Ambrosio, Christian Dannewitz, Holger Karl, and Vinicio Vercellone. 2011. MDHT: a hierarchical name resolution service for information-centric networks. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking (ICN '11). ACM, New York, NY, USA, 7-12. DOI=10.1145/2018584.2018587 http://doi.acm.org/10.1145/2018584.2018587

[2] CONVERGENCE project. Deliverable D4.1. Preliminary Definition of the Versatile Digital Item. http://www.ict-convergence.eu/deliverables

[3] CONVERGENCE project. Deliverable D3.2. System architecture. http://www.ict-convergence.eu/deliverables

[4] Diego Perino, Matteo Varvello, A Reality Check for Content Centric Networking, ACM SIGCOMM Workshop on Information Centric Networking (ICN), Toronto, Canada, August 2011.

[5] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, "Supporting the Web with an Information Centric Network that Routes by Name", Elsevier Computer Networks, vol. 56, Issue 17, p. 3705–3722.

[6] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. Transactions on Networking, 9(3), 2001.

[7] A. Narayanan, S. Previdi, B. Field "BGP advertisements for content URIs", INTERNET-DRAFT draft-narayanan-icnrg-bgp-uri-00, July 2012.

[8] Lan Wang, A K M Mahmudul Hoque, Cheng Yi, Adam Alyyan, Beichuan Zhang "OSPFN: An OSPF Based Routing Protocol for Named Data Networking", NDN Technical Report NDN-0003, July 2012.

[9] Standards for efficient cryptography, "SEC 2: Recommended Elliptic Curve Domain Parameters", Certicom Research available at www.secg.org/collateral/sec2_final.pdf

[10] D. Galindo, F. D. Garcia, "A Schnorr-Like Lightweight Identity-Based Signature Scheme", in International Conference on Cryptology in Africa: Progress in Cryptology (AFRICACRYPT '09), Bart Preneel (Ed.). Springer-Verlag, Berlin, Heidelberg, 135-148.

[11] IRTF - Information-Centric Networking Research Group (ICNRG); http://irtf.org/icnrg

[12] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, J. Wilcox, "Information-Centric Networking: Seeing the Forest for the Trees", 10th ACM Workshop on Hot Topics in Networks, HotNets 2011.

[13] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures", ACM SIGCOMM workshop on Information-centric networking, ICN 2011.

[14] D. K. Smetters, V. Jacobson, “Securing network content”, PARC TR-2009-1; 2009 October.

[15] Dmitri Krioukov, k c claffy, Kevin Fall, and Arthur Brady. “On compact routing for the internet”. SIGCOMM Comput. Commun. Rev. 37, 3 (July 2007), 41-52. DOI=10.1145/1273445.1273450; http://doi.acm.org/10.1145/1273445.1273450

[16] A. Detti, M. Pomposini, N. Blefari Melazzi, and S. Salsano: “Supporting the Web with an Information Centric Network that Routes by Name”, Elsevier Computer Networks, August 2012.

[17] Adi Shamir, “Identity-Based Cryptosystems and Signature Schemes”, Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 7:47--53, 1984.

[18] D. Johnson and A. Menezes, “The elliptic curve digital signature algorithm (ECDSA)”, Technical Report CORR 99-34, University of Waterloo, Canada, February 24 2000.

[19] Rivest, R.; A. Shamir; L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the ACM 21 (2): 120–126, 1978.

[20] A. Kate and I. Goldberg, “Distributed private-key generators for identity-based cryptography”, International conference on Security and Cryptography for Networks (SCN'10), Springer-Verlag, Berlin, Heidelberg, 2010.