Computer Networks 57 (2013) 3249–3266

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Realising an application environment for information-centric networking

Ben Tagger [a,c,*], Dirk Trossen [a], Alexandros Kostopoulos [b], Stuart Porter [c], George Parisis [a]

[a] Computer Laboratory, University of Cambridge, Cambridge, UK
[b] Athens University of Economics and Business, Department of Informatics, Athens 10434, Greece
[c] CTVC Ltd., London, UK

ARTICLE INFO

Article history:
Received 17 December 2012
Received in revised form 25 July 2013
Accepted 1 August 2013
Available online 29 August 2013

Keywords:
Information-centric networking
Application development environment
Middleware
Personalised delivery

ABSTRACT

It has been argued by many that the Future Internet should address information at the core of its operation. Prototypes have emerged to embody this new paradigm. Applications for such networks, however, are noted primarily by their absence. In spite of an appetite for Information-Centric Networking (ICN) applications, relatively little has come to fruition. We suggest that this is due to an unfavorable development environment, requiring applications to interface with the ICN substrate directly. This paper aims to answer this shortcoming by providing a middleware layer that aids the development of more advanced applications. We also present an application that leverages the middleware and answers a real-world problem concerning personalised media delivery. We argue that the development of this, and potentially other, application(s) is aided by the presence of such an application environment.

1. Introduction

In today's Internet, network operations are based on the exchange of opaque bits between explicitly identified endpoints. There exists no notion of information at an infrastructure level to assist in the storage, operation or otherwise management of networked content. In recent years, it has been suggested that *what* is being communicated is more important than *who* sent it or *where* it is going [1,2]. This has led us to question whether or not the current Internet architecture, with its focus on connection endpoints, is suitable in an information-centric world. As this debate continues [3], many research efforts have started to develop solutions for new networking architectures that place information at the focal point of design (see [4,5] for a small sample of efforts).

While considerable effort has been invested into designing and implementing new information-centric networking architectures, there has been virtually no emphasis on what we are going to do with these networks once we have them. We have already seen some simple applications for ICN that demonstrate the movement of bits or even media packets across a network [6]. These sample applications usually replicate a process that the current Internet does rather well and, as a result, are usually unimpressive as applications. The question remains: where are the *killer apps* for ICN? To answer this question, we return to the state of the current Internet. Very few applications are written over raw IP sockets; abstractions, middleware and toolsets have been developed for building and deploying web applications. As these development tools have matured, the resulting applications have become richer and more complex. Similarly using an information-centric network, we do not want to operate directly on the ICN substrate and this view is shared in related endeavors [7,8]. Given our network-level abstractions, we suggest that a middleware layer can be extremely thin compared to that

* Corresponding author at: Computer Laboratory, University of Cambridge, Cambridge, UK. Tel.: +44 7588338088.
E-mail address: bentagger@gmail.com (B. Tagger).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.08.015

required for an IP-based infrastructure, which often requires heavyweight solutions that are difficult to optimise. We present middleware that supports and enables the development of more advanced applications within an information-centric network. This middleware, which we optimistically refer to as *information-centric middleware* (*ICM*), models at its core the information space of a given application and automatically migrates this metamodel towards the network. We believe that the realisation of this middleware is key to enabling the development of large-scale applications for ICN.

In the next section, we provide a brief recap of our ICN prototype, *Blackadder*. In section three, we present our information-centric middleware that describes the process of translating the information space from the application to the network layer. As well as some generic middleware benefits, we describe two specific mechanisms enabled by the middleware; browsing of networked resources and distributed querying within the network. In section four, we present a real use case with commercial requirements that calls for the efficient and dynamic delivery of personalised content from distributed resources. We provide a solution to this use case, the Media Story Delivery Network (M-SDN), which demonstrates the increased flexibility of the new middleware as well as the decreased complexity of doing so compared to IP-based solutions. We suggest that the use of ICN in this case provides benefits with respect to a traditional *endpoint* implementation and we investigate why personalised delivery is so hard in the current Internet at the conclusion of section four. In section five, we present some evaluation of the middleware including some arguments for ICN and a description of some socio-economic aspects of the M-SDN. Finally, we present our conclusions of the work together with some thoughts for further work.

2. The architectural backdrop

Our design is grounded in a specific starting point for ICN that is described in [1] and elaborated in [9]. We provide a brief recap by outlining the five major principles of this architecture. These principles are realised in Blackadder [9], our prototype, and we describe their realisation within this implementation below. For a complete description of the prototype, refer to [9].

2.1. Identification of individual information items

We advocate the use of statistically unique, fixed size labels as a means of identification for individual pieces of information. These labels do not carry semantics and are meaningless to most network components and applications.

2.2. Contextualisation of information

The second principle places information items into a context, called a *scope*. A scope represents a set of information. It is an information item itself, identified with an individual identifier, and can be nested under other scopes. This leads to an information structure that forms a directed acyclic graph (DAG). It is this DAG structure that provides a simplistic but related form of many existing application concepts (e.g., ontological concepts, complex event processing, etc.), potentially promising an easy mapping of these higher-level concepts onto the abstractions provided by our architecture. Leaf nodes in the graph represent pieces of information, while inner nodes represent scopes. Fig. 1 shows an example for a possible information graph with SId denoting scopes and RId denoting items of information. Within the prototype, each node in the graph is identified with its full path, starting from a root scope. An individual node or leaf in the graph is currently implemented as a 64-bit flat label.

2.3. Interfacing of the information graph

The service model that operates on the information graph encapsulates the third principle. It follows a publish–subscribe semantic, like those provided by many application-level event systems. Table 1 presents the network-level interface exposed by the prototype design.

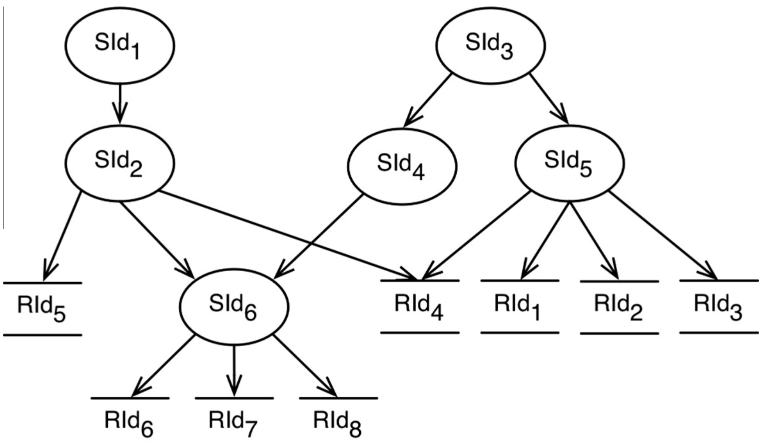


Fig. 1. The information space within blackadder.

Table 1
The interface for our networking prototype, Blackadder.

Method	Parameters	Description
publish_scope	string ID, string prefixID	Publishes a scope identified by ID under the scope previously published as prefixID
publish_info	string ID, string prefixID	Publishes an information item identified by ID under the scope previously published as prefixID. Note that data transfer does not occur at this point. Data transfer only occurs when at least one subscriber has been matched to this publication
unpublish	string ID	Deletes all references to a scope or item identified by ID
subscribe_scope	string ID	Subscribes to the scope identified with ID and implicitly subscribes to all information items under that scope
subscribe_info	string ID	Subscribes to an information item identified with ID. If at least one publisher exists for the item, it is matched to the subscriber. Otherwise, the interest for the information is registered until a publisher exists
unsubscribe_scope	string ID	Unsubscribes from the scope identified by ID
unsubscribe_info	string ID	Unsubscribes from the item identified by ID
publish_data	string ID, char* data, int len	Publishes data corresponding to the information item identified by ID. The publisher, upon receipt of a notification of subscriber interest, calls this method

These are the methods that are exposed to both network and application-layer components.

It should be noted that we have removed all mention of dissemination strategies from the above overview for the sake of simplicity. Each method call has an additional parameter that specifies the dissemination strategy to be used. We briefly describe dissemination strategies below but refer to [9] for a more thorough account of the available strategies.

2.4. Modularisation of the core network functions

Our fourth principle addresses the core network functions associated with the dissemination of information within a given scope; *rendezvous*, *topology management* and *forwarding*. The first, *rendezvous*, matches the availability of information to the interest in it. In other words, it creates a relationship between the publisher and subscriber of a particular information item. The locations of the publisher and subscriber are then used by the second function, *topology management*, to construct a suitable delivery graph for the transfer of data encapsulated by the information item. Finally, the transfer of data is executed by the third function, *forwarding*. The prototype implements these core functions in its node design, presented in [9].

2.5. Flexible information dissemination based on information scoping and well-defined strategies

The fifth principle addresses the methods used for implementing the aforementioned functions and also the issues regarding information space governance and management within the information space. For this, we define dissemination strategies associated to (parts of) the information structure, these strategies capturing the implementation details. Together with the scoping of information subspaces, these strategies establish a functional scoping through which the distinct functions can be optimised based on the requirements of communicating entities that access specific parts of the information graph.

Information-centric networking is an area of research that aims to bring applications *closer to the network* and this is achieved by providing information-centric abstractions that are accompanied by a publish–subscribe service model. As such, complex operations traditionally at the middleware layer that aid services such as resource discovery, resolution of middleware abstractions and mapping networking endpoint identifiers, are realised, in part, by this new internetworking layer. The promise of reducing the necessary complexity of middleware solutions goes hand-in-hand with the potential for optimising the operation of the infrastructure through an increased potential for caching information (instead of opaque packets) as well as optimising the core functions of the network.

3. Enabling an application environment: information-centric middleware

The purpose of many middleware systems is to hide low-level details from the application, providing instead a manageable abstraction of the underlying infrastructure. For many application environments, particularly dynamic ones, we cannot make *a priori* assumptions of the state of such infrastructure, requiring a middleware that can present context to the application layer while adapting to the required level of variability at the execution layer. Depending on the abstractions provided at the execution layer and the functionality required by the applications, the middleware layer can quickly become bloated. We suggest that many middleware efforts, including those offered for IP-based infrastructures, require developmental effort and additional complexity to achieve the information-centric starting point that already exists at our network layer. We present a middleware implementation below that, thanks to these abstractions, exists as a thin, shim layer for participating applications.

3.1. What is information-centric middleware?

Given that our networking infrastructure already maintains a significant complement of information semantics, we require a middleware layer that not only provides support for this but also ideally extends it. To this end, we present a design for *information-centric middleware* that operates directly on the ICN substrate, extending the idea of information-centricity towards the application layer with the use of semantic technologies and metadata.

Our middleware enables interaction with the network via a semantic layer that fulfils three key roles:

1. It presents networked data based on the associated metadata and relationships defined within a middleware metamodel. This metamodel is initially internalised as an ontology, which represents the application information space as a set of concepts and relations.
2. It aggregates heterogeneous networked data to provide a consolidated view of available resources. This allows the possibility to reuse common metadata to annotate publications and enables the transparent distribution of queries.
3. It allows us to attribute a degree of confidence about the consistency of networked data with the use of ontological tools such as reasoners [10].

The use of semantic metadata allows us to capture, not only context- but configuration-based information about the current state of the environment and the data therein. Furthermore, Semantic Web technologies such as ontologies and reasoners, allow us to infer implicit relationships from the available metadata. Semantic Web technologies have been shown to aid the execution of complex tasks while dealing with heterogeneity of the input [11]. They also have significant potential as a tool for solving problems of interoperability within ubiquitous computing [12] as well as providing *better automation of user's tasks*. The ease with which they deal with heterogeneity, promote interoperability and improve automation has caused them to be widely considered for middleware design (see [13–15] for a small selection of efforts). The construction of contextual ontologies to describe a domain of information is an important aspect of providing a shared view of an execution environment. For this reason, contextual ontologies are often used within middleware [16–18] and, furthermore, frameworks for such contextual ontologies [19] have emerged.

Technologies for the Semantic Web have predominantly aimed towards higher, application/user levels and much of the developmental focus has been to enhance machine readability [20] and overcome the limitations of *older* Internet tools, such as HTML. Technologies have evolved as tools and languages to support a growing need to understand the *meaning* of the data we use on a daily basis. Such tools have traditionally been expected to run over IP but how do these technologies relate to ICN?

Information implies an inherent understanding; *information* is interpreted *data*. The aim of the Semantic Web is to enable this interpretation, albeit at a higher level. ICN and its implementations, by definition, place information at the core design. We argue that, given this marriage of interest, semantic technologies are a natural consideration for middleware solutions for an ICN-based architecture. In fact, we go further and propose that an ICN implementation *facilitates* semantic (information-centric) technologies at higher levels by considering their requirements at the network level. This is important, as a future network should provide a useful abstraction for higher layers, as argued in [1]. We suggest that the ICN layer can facilitate the development and, specifically, lower the complexity and burden of a semantically aware middleware.

The remainder of this section is split broadly into two parts; Section 3.2 describes how to get data and metadata *in* to the system via the middleware and Section 3.3 describes mechanisms for getting data *out* of the system via the middleware.

3.2. Metamodelling in the middleware

We have already suggested that there is a mapping between the information space at the application layer to that at the network. In order to realise our application environment, we must understand the metadata requirements across these layers. More specifically, we ask the question: *What metadata do we preserve at the network layer* (*for processes at that level*) *and what metadata do we expose at the middleware layer* (*for users and applications*)? It is the role of the middleware to enable modelling of the metadata across these layers.

We now describe the process by which publications, annotated with metadata, enter the network via the middleware. We first present the idea of *stock ontologies*, a *pool* of available metadata with which we can annotate our publications. We go on to describe the mapping process by which publications are annotated with consistent metadata and subsequently placed within the network information space.

3.2.1. Stock ontologies

We have already asserted that our publications are to be annotated with metadata but where does that metadata come from? We use ontologies to encapsulate the semantics required within the middleware. Ontologies define terms and concepts and formally describe the relationships therein. We propose the assimilation of a set of *stock ontologies* (*SOs*) to enrich the publication semantics with which we can annotate publications to form a complete, descriptive publication. We propose the development of *stock ontologies* (SOs) for each definable domain, both within the application and network space; i.e., QoS, caching, etc. We provide no bespoke method for building a *stock ontology*; it is, to all intents, a *regular* ontology and can be built by any of the usual methods. What makes a *stock ontology* distinct is its purpose to provide a common platform upon which to build more application-specific metastructures. For example, there may be several vendors using the middleware to publish similar content (i.e. video catalogs). While the specific content might vary between vendors, it is likely that the video format, for example, will be unique. In such a case, it might be preferable for vendors to import a media format ontology from which to annotate their video catalog. Such an ontology, within our system would be referred to as a *stock ontology*. Going forward, the vendors might decide to use a third-party ontology[1]

[1] For example, they may use the Movie Ontology (MO) – http://www.movieontology.org/.

for their video annotations, allowing for an even slimmer application metastructure, and further optimise generic content-aggregation tools in this domain. In such a case, that third-party ontology would become a *stock ontology*. It is anticipated that, while application ontologies might be relatively numerous on a *per*-application basis, a *stock ontology* is defined by its use and efficacy between multiple applications.

The use of ontologies has several advantages. Firstly, publications are annotated with *consistent* metadata. We can use reasoners to ensure the consistency of the ontologies and, therefore, ensure that the descriptions of the publications are also consistent [10]. Furthermore, if we need to add a new feature (e.g. costing) or we want to describe publications of different content (e.g. library records), we need only add a single ontology describing the new feature and we can immediately begin annotating our publications with the new metadata.

It has been our assertion that the design of the network prototype can be leveraged to create a middleware layer that reuses much of the underlying architecture to (a) reduce the complexity of and (b) lower the computational burden on the required middleware. With that in mind, we present the implementation of the middleware that exists as a shim software layer for the publisher as well the subscriber (in certain circumstances). We suggest that this is a suitable approach within a network where performance is a key factor for success. Indeed, when services need to be delivered on the fly, it is impractical to employ tight integration of ontological reasoning [21]. It also supports our suggestion that an information-centric approach at the network layer, as in the prototype, simplifies the developmental complexity of the middleware.

3.2.2. Mapping the network metamodel

The following subsection, as illustrated in Fig. 2, describes the process by which a single publication enters the network via the middleware. This process can be placed in a batch mechanism in order to publish multiple publications at once although, for the sake of simplicity, we assume a single publication. In the following section, we will use the example of a video file that is to be added (published) to the network.

3.2.2.1. Annotation of the publication. We start with a publication payload. This is the raw data of the publication. To this data, we add metadata in the form of annotations from the SOs. The annotations are specified in the form of an expression, perhaps in description logic that is passed, along with the data, to the middleware during a publish call. As the ontologies are consistent, the annotations also benefit from this consistency. This does not mean that the annotations are *correct* (i.e., the annotations accurately describe the payload) but it does mean that the annotations are semantically valid; they do not contradict one another and are satisfiable. For our video file, we might attach metadata such as; *title* = "*Matrix*", *film_length* = "*136*", *actor* = "*Keanu Reeves*", *HD_possible* = *true*, and *UK_only* = *true*. This metadata will come from the stock ontologies where we might find a content ontology (containing *title*, *film_length* and *actors*), a Quality of Experience

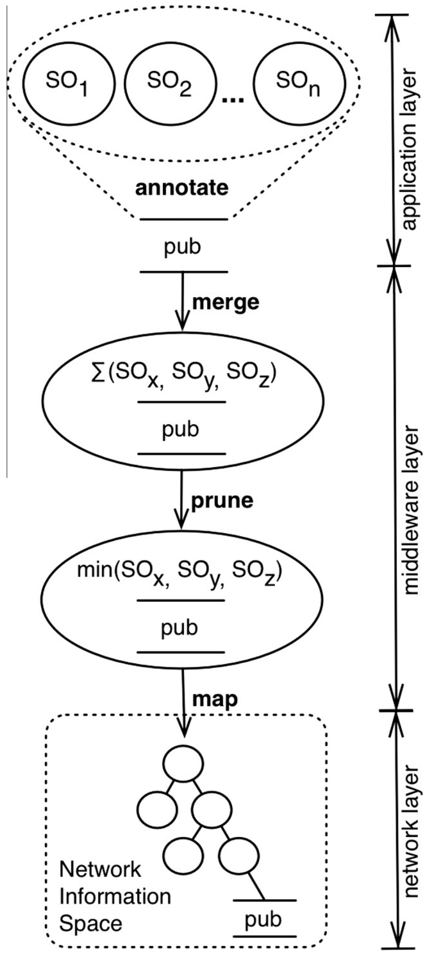


Fig. 2. The metadata mapping for a single publication.

(QoE) ontology (specifying whether we expect HD content for example) and any others depending on how we want to annotate our publication.

3.2.2.2. Merging of the stock ontologies. In order to move towards a single metamodel we merge the SOs into a single ontology, addressing (in part) the second challenge described above. The result is a single ontology that contains every concept in the SOs and every publication is specified exactly once.

3.2.2.3. Pruning the merged ontology. We find two opportunities for optimisation within the merged ontology described above. The first is unused concepts. We, therefore, prune any concepts that are not associated with a publication ensuring the resulting ICN metamodel will be as minimal as possible. We also prune concepts, whether attached to publications or not, that are *tagged* as non-network metadata. *Tagging* occurs during the build of the SOs in which developers specify which metadata (concepts) are to be preserved in the network and which are not. The pruning of tagged concepts ensures that the ICN metamodel contains the appropriate metadata. We use the *ontological toolset* (described below) to specify whether or not a concept or relation is to be preserved at the network layer, addressing the first challenge described above.

3.2.2.4. Mapping onto the ICN metamodel. The final stage is to map the pruned metamodel onto constructs used at the network layer, i.e., the scoping and labelling concepts for structuring information. Thanks to the preceding steps, the merged, pruned ontology is now much closer in terms of structure to the desired network metamodel. But there are still some significant differences; the network metamodel has a significantly simpler structure. Ontological sub-class relationships can be modelled directly by ICN sub-scope relations but ontological data and object properties cannot be directly modelled. At this stage of development, we provide an ontological toolset (OT) that aids the developer with this mapping process.

The OT is essentially a set of ontological super-properties to which we attribute bespoke ICN mappings. The OT is presently available as an ontology and is imported into the application ontology or SO to allow use of the OT properties. There are three standard OT properties; *hasScope*, *hasSubScope* and *hasItem*. By extending to one of these properties, we are expressing a requirement for that expression to be treated in some specific way by the middleware. These standard properties allow metastructures to express their properties in terms that can be understood in the network metamodel, in these cases flattening them into a simplistic inheritance-based metamodel.

Consider Fig. 3 as an example of using the standard OT property, *hasScope*. We have defined a new property, *hasFriend*, and we want the middleware to treat this new property in the same way as the *hasScope* property. In this simple example, we extend the *hasScope* OT property and this instructs the middleware that when it encounters the *hasFriend* property, it should be placed under the current scope as a new scope. While suitable for simple properties, we often need to express properties that cannot be expressed in terms of simple inheritance; for example, the expression that one person is the cousin of another requires traversal up and then back down the graph (assuming a traditional family tree-type graph). In these cases, we allow the OT to be extended using the additional property, *hasAssociation*. By extending from the *hasAssociation* property, we are asserting that there is an association from one concept to another but that association cannot be described using any of the standard OT properties. Having

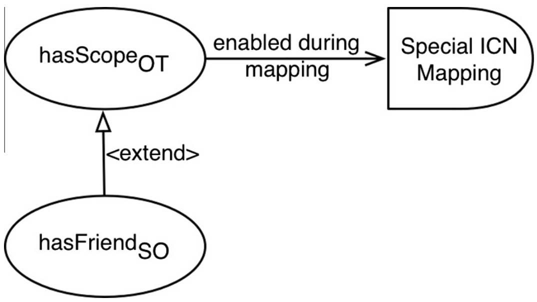


Fig. 3. Example of OT property extension.

made this extension, we must also tell the middleware how to deal with it. However, we have not yet developed a user-friendly way of doing this and it, therefore, requires direct handling of the middleware code in the form of a bespoke method.

The result of the ICN mapping described above is a network metamodel that maintains, to a significant degree, the original semantic structure defined in the middleware. But, why is this necessary or even preferable? The prototype, described in Section 2 and in [9], endeavors to classify information within the network layer with its abstractions of scopes and information items. These abstractions have, to a significant extent, enabled optimisations at the network layer. For example, routing processes rely on the network metastructure to route semantically similar items (i.e., grouped under the same scope, for example) in a similar fashion. Caching algorithms can use the network layer to opportunistically cache information items linked through the metamodel. It is therefore essential that we continue to support these and other optimisations, as well as enabling others.

3.3. Middleware services

The following subsection describes two middleware mechanisms that we use to get data out of the network. The first, browsing, allows scopes to become self-describing; something that is only possible with the *a priori* knowledge of the metamodel provided by the middleware. The second mechanism, querying, allows applications to specify search terms in order to request access to networked resources.

3.3.1. Mechanism: browsing

A basic requirement of a networking architecture is the ability to get a handle on the information within the network. In the absence of the middleware, the subscriber must know the ID of the media item they wish to retrieve. The only way that a subscriber can have this *a priori* information is by explicitly requesting it as a separate transaction; i.e., the subscriber subscribes to a pre-agreed catalog ID and the publisher publishes a catalog (in list form, for example) of the available media together with their associated IDs. There are two shortcomings of this approach. Firstly, the catalog may likely be very large and, if the media metadata is rich, complex too. This style of delivery is undesirable from the position of a subscriber, who may very well be a lightweight consumer application. The second shortcoming is that the catalog is sent as a static information item. It, therefore, cannot reflect changes that have occurred in the metamodel since it was last sent. The result is that the subscriber must (a) continually poll the publisher for updates to a potentially bulky catalog or (b) operate on catalog data that may be out-of-date.

In order to address the problems above, we have added to the middleware-published scopes the ability to self-describe. To each scope, we add a *special* data item that contains, by way of the payload, the scope's metadata. This metadata is created during the building of the ICN metamodel and contains the labels and associated IDs for the scope's children and parents if they exist. We can specify

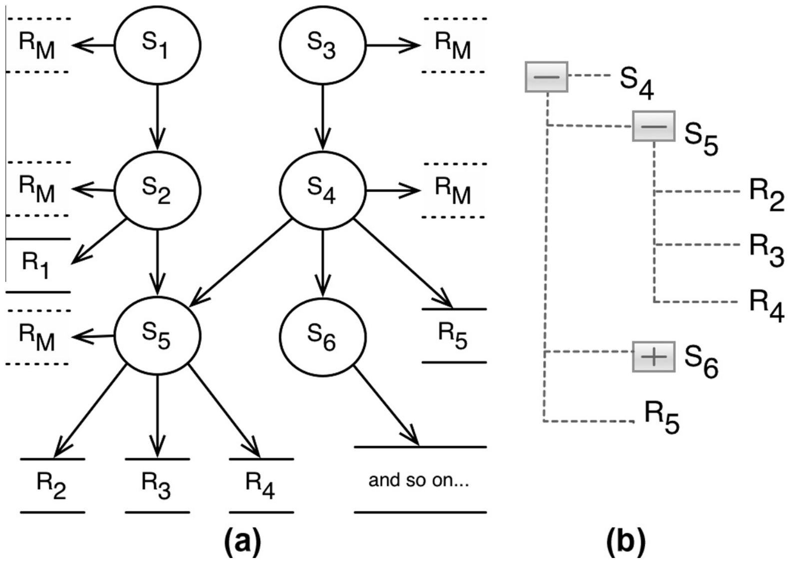


Fig. 4. (a) A sample network metamodel and (b) a view from the subscriber.

whether a scope, and therefore a branch of the network, is *browseable* by choosing to include it or not within the metadata data item. Now, a subscriber will subscribe to a scope in the network, not by subscribing to the scope itself but by subscribing directly to the metadata information item for that scope. In this way, we enable the inherent structure of our network to take the form of the catalog while providing the subscriber with a *view-of-interest* rather than the entire catalog at once.

Fig. 4(a) illustrates the network metamodel complete with scopes, S, data items, R, and metadata items, R_M. In this case, each scope has an associated metadata item. Fig. 4(b) illustrates the view from the subscriber when subscribing to the metamodel and illustrates two consecutive subscriptions (S_4–R_M followed by S_5–R_M). The subscriber subscribes to S_4 and, in reality, receives the metadata item published under S_4. The subscriber view is currently centered on S_4, receiving information on the children of that scope, S_5, S_6, and R_5. The subscriber then subscribes to S_5, which reveals the publications, R_2, R_3 and R_4 and so on. In this way, the catalog is revealed to the subscriber under a *spotlight* of interest from the point of view of that subscriber. The catalog requests are small and directed in contrast to a scenario that delivers the entire catalog.

3.3.2. Mechanism: querying

We often need to search the network for desired resources. Within a publish/subscribe platform such as ours, retrieving this data is non-trivial, as the information may not exist under any single scope within the ICN metamodel to which the subscriber can subscribe.

To address this issue, we can pass the query, an expression that is built at the subscriber, to the publisher to execute within the middleware. This query must first be sent to the publisher but there is a problem; it is not possible to subscribe to the ICN metamodel with a query, only an RID (refer back to Table 1 in Section 2). We have therefore implemented a method, illustrated in Fig. 5, which allows a subscriber to effectively query the publisher and retrieve results from that query using the ICN architecture and its native API.

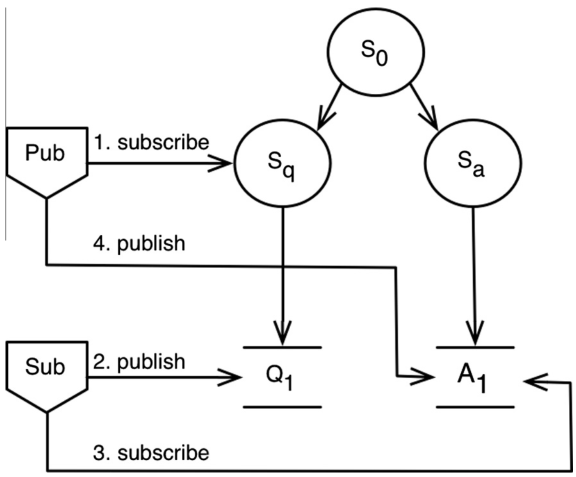


Fig. 5. Our mechanism for distributing queries.

In Fig. 5, scope S_0 represents the scope of interest for a particular domain, in other words our starting point. We begin by creating two additional sub-scopes under S_0; S_q and S_a that represent the scopes under which queries and answers are placed respectively.

1. The publisher starts by subscribing to S_q, indicating an interest in anything under that scope.
2. The subscriber publishes a data item (Q_1) under S_q, the payload of which contains both the query and a generated RID of the data item it expects the query results to be accessible from. In this case, it will be the RID of A_1 when it is eventually published.

3. The subscriber, knowing the RID of A_1, then immediately subscribes to it.
4. As the publisher is a subscriber of S_q and, therefore, implicitly a subscriber to all information items of S_q, it automatically receives the query Q_1. At this point, the publisher answers the query in the form of a published data item, A_1.
5. The subscriber receives this answer as it is subscribed to A_1 and, as it is already expecting A_1, knows the context of the answer (i.e., it can differentiate the answer from A_1 from any other answer).

In this way, we can use the middleware to answer complex distributed queries within the network without having to alter the structure of the network layer metamodel significantly. We contend that this mechanism also provides an elegant asynchronous method for distributed query answering as publishers are informed of queries as they arrive as subscribers are informed of answers as they are returned.

4. Story telling in the Internet

In the following section, we present a real commercial use case, with a scenario, that requires the management and delivery of personalised media to create media experiences for the end user. To answer this problem, we have developed the M-SDN that consists of two applications, the *catalog* and the *player*. These applications run over our ICN prototype, *Blackadder*, and leverage the services provided by our middleware (described in the previous section). These applications have been deployed over an international test bed of 40+ nodes across 10 sites.

It is our suggestion that the M-SDN requires significantly less design complexity, thanks to the services and abstractions provided by the middleware and, ultimately, the network. This is in contrast to a potential IP-based solution, which we suggest would require a considerably fatter middleware layer to achieve the same effect. We put some weight behind this argument in Section 4.4 by explaining why we believe this scenario is particularly difficult to address in the current Internet.

4.1. Current limitations for content production and delivery in the Internet

The Internet seeks to provide basic communication for the many users with services such as voice, video and basic access to important information. The mental model of two entities conversing for a certain purpose is directly reflected in the communication that underlies IP and its higher layer protocols, TCP and HTTP (the basic building blocks of today's web): two endpoints connect and retrieve appropriate information for the purpose of the communication.

Based on these basic building blocks, the Internet has become a playground for experiences, expanding on simple data exchanges and website visits in order to tell a story through multimedia (and often multimodal) experiences. For such story-telling, various pieces of information are assembled within the context of the experience and delivered transparently to the destination.

The key to such experiences is that they are often delivered as a mash-up of individual pieces of information instead of creating a monolithic blob of (new) experience; this leads to a distributed approach of experiencing information in the Internet. However, two significant challenges arise when moving to such a story-telling approach through mashing up individual parts of an experience, namely **control** over the individual pieces of information and **efficiency** of the overall experience delivery. What follows is a scenario based on real-world experiences that embodies these challenges.

4.1.1. A real-world problem, unanswered

It is common for media production companies like CTVC [22] to create different versions of their programs to cater for different markets. This is necessary because different broadcasters have different requirements with regard to the length of the film, scenes or imagery that can be included and use of rights-controlled content. Additionally, content is broadcasted in different ways via terrestrial (free to air and cable) signals, satellite signals or via the Internet as IPTV. All protagonists, except for CTVC, described herein are purely fictional.

CTVC has been commissioned by a UK broadcaster to make a short film about the turbulent life of a high profile media mogul called Philip Johnson. Johnson's company, InterNews, owns newspapers, magazines, television broadcasters, and radio stations across the globe and he is an internationally recognised figure.

During production of the film, the production team sources a range of video clips of Johnson from various periods throughout his life. In most cases, the license holders send links to low resolution versions of the footage for the director to view before he decides whether or not to include them in the final film. CTVC then reports its intended use of any content and the license holder usually invoices and issues full definition licenses accordingly. However, some of the footage has been found on the web and is owned by the InterNews archives who refuse to license it for inclusion in the film.

Under sections 29, 30 and 178 of the UK Copyright, Designs and Patents Act 1988, CTVC is entitled to follow a process called "Fair Dealing" which allows the footage to be included without the permission of the content owner so long as it is being used strictly within the context of non-commercial research or private study; review, criticism and news reporting [23]. Because the footage is being used to back up criticism of Johnson's business dealings with a French newspaper, CTVC and the director make the decision to include a lower resolution version of it anyway.

Because Johnson is such a famous figure, CTVC has also managed to secure several international sales deals in the USA, Australia, United Arab Emirates, Russia and Saudi Arabia for terrestrial, cable, satellite and IP-TV broadcast. Whilst US law and Australian law provide a Fair Use clause – similar to the UK's Fair Dealing clause – UAE, Russia and Saudi Arabia do not, so the director decides to change the footage in these versions for something which can be licensed in order to avoid any legal problems in these countries. Different broadcasters also have different slots

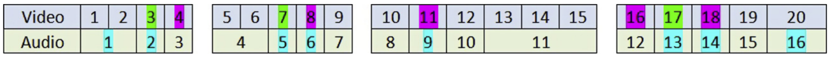


Fig. 6a. UK Version (47 min).

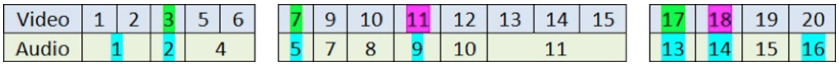


Fig. 6b. US and AUS Versions (38 min).

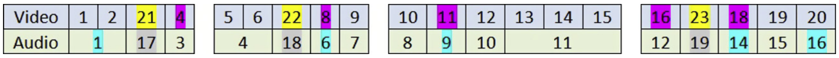


Fig. 6c. Russia Version (47 min).

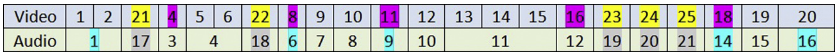


Fig. 6d. UAE and Saudi Arabia Versions (53 min).

available, which require the film to be cut to different lengths for different markets. The US version of the film must be cut to 38 min and 3 parts, whilst the version to be broadcast in UAE and Saudi Arabia needs to be 53 min and 1 part, and a scene with Johnson at a pool party with lots of bikini-clad young women also needs to be removed for the Middle Eastern markets. In all, 4 versions of the film are needed in order to comply with local demands and laws (see Table 2).

In the above scenario, the majority of the specially shot video and audio content remains, in the first instance, on the CTVC servers. The archive material and commercial music used in the soundtrack resides on the servers of the respective publishers. With this material, the publishers advertise their availability, providing an encrypted key (embedded in the license) to allow CTVC to subscribe to their content. During the editing process, the editor pulls chunks of full resolution media from the servers at CTVC and chunks of the high-resolution versions of the licensed footage from the external publishers into a single timeline. The editor orders and cuts the footage to length in order to create the film. He lays a soundtrack over the visuals using audio files from the CTVC servers.

The editor then creates the three international versions of the film, removing the necessary scenes and associated audio, and replacing them with other licensed or original material where necessary, before relaying the sound track and, finally, constructing a timeline for each version. Fig. 6 illustrates the four distinct timelines required to satisfy the six countries in which the film is to be distributed.

4.1.2. Significant challenges identified

In today's infrastructure, the additional work involved in creating the three *additional* versions is time consuming and expensive. It will usually involve several days of extra work in the edit suite, compiling multiple viewing versions onto digital tape formats and expensive international transportation of the tapes containing the final versions to the end-user/broadcasters. The scenario above presents the complexities of generating multiple versions but a real scenario can be more complex still. Considering the requirements for audio and language tracks, subtitles, description tracks for the aurally or visually impaired, the number of versions needed for each region can increase dramatically. In the case of digital delivery, the situation is further exacerbated. Each video track may need to be encoded at different rates for delivery depending on subscription or the media destination; i.e., HD TVs vs. cell phones. The possible combinations can quickly become unwieldy and the management of the situation becomes a content-management nightmare [24].

From a delivery perspective, the complexity is even more challenging. Different, albeit only slightly, storylines are represented within the network as "monolithic blobs". In other words, current IP networks are completely unaware that the data being transmitted is largely identical. As a result, the delivery of the story-based content cannot be optimised by, for example, caching common parts or pulling specific parts from remote repositories (instead of pooling these parts at the creator's side).

The aspect of pulling information together for delivering an experience to an end user often reveals efficiency

Table 2
Different versions of a film.

Market	Length	Version description
United Kingdom	47 min 4 parts	Including all licensed and "fairly dealt" material
United States and Australia	38 min 3 parts	Including all licensed and "fairly dealt" material
Russia	47 min 4 parts	"Fairly dealt" material replaced
UAE and Saudi Arabia	53 min 1 part	"Fairly dealt" and bikini clad women material removed/replaced

```
TITLE: EDLTEST1
FCM: NON-DROP FRAME

001  D1T1C3A  AA/V  C       01:04:51:16 01:04:57:15 01:00:00:00 01:00:05:24
* FROM CLIP NAME:  D1T1C3-GVS-B.MOV

002  D1T1C3A  AA/V  C       01:01:07:18 01:01:13:02 01:00:05:24 01:00:11:08
* FROM CLIP NAME:  D1T1C3-GVS-B.MOV

003  D1T1C3A  AA/V  C       01:06:34:13 01:06:41:14 01:00:11:08 01:00:18:09
* FROM CLIP NAME:  D1T1C3-GVS-B-1.MOV

004  D1T1C3A  AA/V  C       01:07:41:11 01:07:52:23 01:00:18:09 01:00:29:21
* FROM CLIP NAME:  D1T1C3-GVS-B-1.MOV
```

Fig. 7. A sample edit decision list.

as a major challenge too. Given that current IP-based networks are agnostic to the information transmitted, there are limits to the support for efficient content delivery and personalised information distribution. This is despite iterative attempts by the wider industry to improve on these limitations, for instance, through dedicated content delivery network (CDN) technologies.

Problems related to legal issues, such as InterNews' archives refusal for licensing some of the footage for inclusion in the film, require additional manual actions for stakeholders like CTVC, which cannot be automatically provided by current IP networks. This requires CTVC to cater to the specific needs of the individual license holders through producing individual content pieces, each of which reflects the particular situation with respect to the rights of contributing content. A solution where an overall storyline would be adapted based on (possibly dynamically) changing rights for parts of the content would require an object-based digital rights management being deployed; not easily implementable in IP networks.

The key questions we aim to answer are; how content distribution can be accomplished in a more efficient manner; how the involved stakeholders become accountable for the delivered data; and how the Internet might address such problems. We address these aspects in the following manner. First, we present an enabling networking paradigm that we suggest provides improved efficiency by making the individual parts of the storyline explicitly identifiable throughout the network. This is implemented using our prototype (described in section two) with a middleware layer (described in section three) that exists as a SHIM layer between the network and the M-SDN applications.

4.2. A real data model: the edit decision list

One of the key considerations for a story-telling service is the way in which the stories are modelled within the system. In the case of media stories, as in the M-SDN, we have been able to leverage an existing data structure that is used within media companies today. We refer to the Edit Decision List (or EDL)[2] that is used by producers of professional media content, such as CTVC, to specify the video footage and audio segments that make up a particular film or, in our case, a story. An example of an EDL is shown in Fig. 7.

Consider once again, the four film versions of our scenario film as depicted in Fig. 6. Each version is instantiated by a single EDL, a data document containing the order and time-codes of the various chunks of audio and video that make up the film. Once the final EDLs are completed in the offline edit, they are passed over to the online editor who, according to each version, compiles the corresponding film.

In the M-SDN, however, these EDLs enter the network and are represented directly as information items. It is these simple data files that are delivered to users of the M-SDN. Depending on their requirements, permissions or any other relevant metadata, users have the options of subscribing to their respective, personalised versions of the film. For example, a broadcaster may have a wider spectrum of permissions to view more versions of the film than an end-user, who can only subscribe to the version applicable to their particular end-user agreement. So, the content is retrieved along a storyline of personalised content pieces rather than as a single, monolithic blob of content. This moves content delivery from a centralised production and delivery process to a scenario of highly distributed editing as well as delivery of content pieces, each of which is tied into a larger story (all of which is under control of the production company).

4.3. Delivering personalised media over an information-centric network

Let us start by examining an overview of the system, illustrated in Fig. 8, in order to determine how the middleware and M-SDN applications fit together within an ICN environment. The M-SDN consists of two distinct applications; the catalog and the player. Largely as a result of existing within a publish/subscribe network, these applications are sometimes referred to as the *publisher* and *subscriber* to denote their primary function; the catalog publishes media and media stories and the player subscribes to them. However, as we shall see later, each application performs both publish and subscribe operations. In Fig. 8, for the sake of simplicity, the middleware is represented as a single function; the middleware, however, exists as a SHIM layer over both the catalog and player applications, allowing applications to either use

[2] The Edit Decision List – http://www.google.com/patents/US6871003.

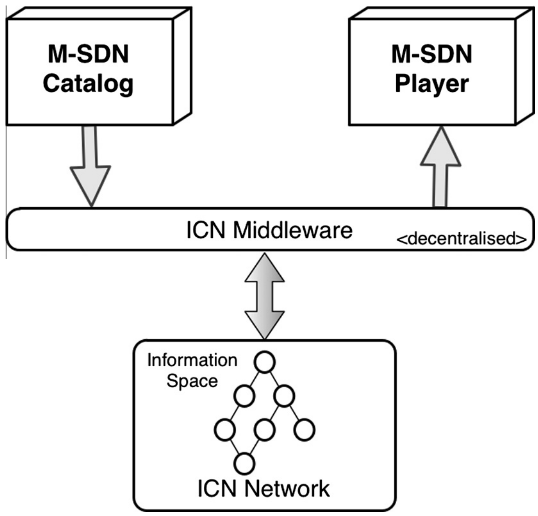


Fig. 8. An overview of the application environment of the M-SDN.

the middleware or not. In other words, applications are not forced to use the middleware and information published by the middleware is fully backwards compatible with existing non-middleware applications.

The M-SDN Catalog is responsible for modelling the application-layer information space, in this case a set of media items and stories (EDLs). Media items, stories and semantics (the structure of the information space) can either be added via the catalog itself (through its own GUI) or by directly manipulating the content ontology (one of the stock ontologies described in Section 3.2.1) via a third-party tool such as Protégé [25]. There are no *special* operations or data structures within the catalog application; the catalog consists of a set of ontological concepts that are annotated with concepts from the stock ontologies. It is this metamodel that is passed to the middleware, that performs the mapping described in Section 3.2.2, which then populates the information space within the network. The M-SDN player application leverages this information space using the mechanisms described in Section 3.3 and these are described in more detail below.

Fig. 9 illustrates the retrieval of a single media story using the M-SDN player and the middleware together with the Blackadder prototype. We assume that the media items (the source media) and the stories (the EDLs) have already been published in the network although the source media can exist in heterogeneous locations.

We begin with our actor – a broadcaster or any other subscriber of media – who has a metadata profile attached to them, this profile captured by the M-SDN player. Some metadata is implicit (modelled in Fig. 9 as rounded boxes) and is generally not directly controlled by the user, such as: geographic location, time of day, currently active subscriptions, and the ability to access adult content. There is also explicit metadata (modelled in Fig. 9 as clouds) that the user can directly influence; the title of the movie they want to watch, the actors they like, the subtitles they require and so on.

1. The metadata profile of the user, including all the explicit and implicit metadata, forms a query that is passed to the M-SDN player. This query is built using account, session and system information, together with search-like terms that are captured in a GUI.
2. This query is passed to the middleware and passed to the network using the mechanism described in Section 4.3.2.
3. Any publishers that have EDLs that match this query (the story annotations match) return them to the M-SDN player as information items. If there is more than one matching EDL, there may be a selection process. We will assume for the sake of simplicity, a single EDL is returned.
4. The EDL that is returned will contain a set of video and audio media items (that also reside somewhere in the network). The M-SDN player forms individual queries from these items and releases them into the network using the same querying mechanism. The queries are released according to an algorithm that takes into account the following; approximate order of the clip, the length of the clip, the amount of footage already buffered, and other information in order to ensure smooth playback of the media story and smooth clip handover.
5. As in 3, the clip queries are answered by any able publisher, which subsequently forwards the media information directly to the *M-SDN player*.
6. The received clips are re-assembled at the *M-SDN player* and played back to the user.

The *M-SDN* allows for the delivery of personalised and democratised content within an ICN infrastructure. Film and clip metadata (audio and video) is maintained and managed within the middleware and remains persistent throughout the lifetime of the media. Whereupon each media clip is played back at the application, it does so with the full complement of metadata with which it originated, including any kind of policy or DRM-like metadata.

4.4. Why is this so hard in the current Internet?

4.4.1. Fine-grained access to media

Key to the delivery of personalised media is an analysis of the receiver's credentials to ensure that the appropriate media is delivered to each user. This is a very important consideration as media delivery can have considerable ethical, and even legal, consequences. We see the following aspects as being most important:

- *Access control.* Specific information items (such as parts of a film, TV shows, news, replays, and advertisements), could be omitted or not with regard to the subscription time (i.e. for broadcasting before 10 pm).
- *Parental control.* Assuming that a video file may consist of several information items, each item may have different access policies and constraints for applying parental control. Parental control can be applied on a per-object basis, identifying adult content, violence or bad language and excluding or exchanging the relevant media object.
- *Ethic constraints.* Different kinds of content could be acceptable or banned in a certain territory (i.e. a shot that includes an alcoholic product or some lewd or otherwise culturally inappropriate behavior).

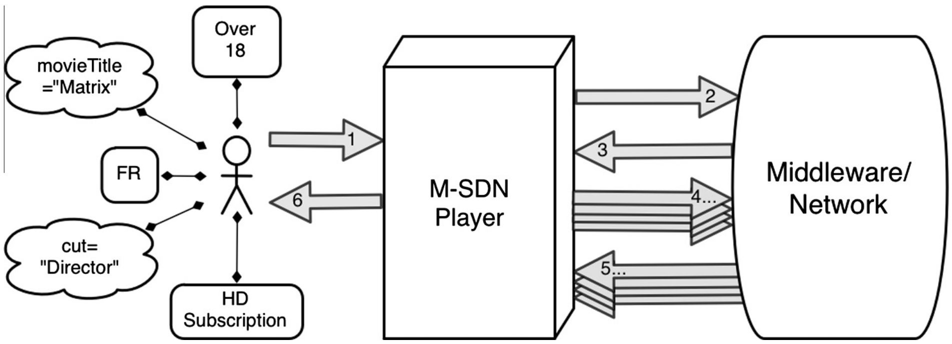


Fig. 9. The request and retrieval of a single media story within the M-SDN.

The assembly of the media experience from individual, objectified media following an agreed storyline lends weight to an ICN-like approach. Access mechanisms that reside at the level of rendezvous functions in the architecture enable the required flexibility as well as control over the final media delivery; a control that is hard to achieve in the current Internet without resorting to the monolithic blobs of content as we know them today. In addition, the focus on information in ICN allows for the provision of *information* security by securing the content independently from the channel over which it is transferred. This is contrast to *channel* security as provided at the network level of the current Internet (e.g. in the form of SSL encryption solutions). An overlay solution might provide similar flexibility in the current Internet but at the expense of likely inefficient delivery of the resulting media stream due to the network unawareness of that same overlay.

4.4.2. Enabling personalised, democratised content

The subject matter of the media determines its use within programming, e.g. through determining, to a significant extent, the viewer's interest in a piece of media. The matching of programming content and viewer interest is a key design feature of the M-SDN, which enables the delivery of personalised content in line with a receiver's expectations and appetites, while complying with well-defined access policies that govern the individual media objects. For example, news agencies could show varying subsets of news items with regard to the locality of the audience.

For the adaptation of an individual media experience, the exposure of individual media objects as well as domain factors at the network level (in the form of information items that are uniquely labelled and therefore identifiable) allow for democratising the delivery of the individual objects. We foresee the use of caching resources in hosted content delivery solutions where available as well as the utilisation of end user resources (e.g. where end users that have already consumed the media object at a prior time and are closely located to the current consumer). This is in stark contrast to the content delivery network solutions commonly used in the current Internet. These CDN solutions are based on dedicated hosting contracts, which centralise the use of storage resources avoiding the democratisation of the available resources (which are much larger than only CDN-provided resources).

4.4.3. Story creation and governance

The flip side of personalised content delivery is that of creating the necessary storylines for each of the individualised experiences as well as managing the delivery from the perspective of the content producer.

Such personalised story creation potentially opens up new revenue streams and markets for content producers, like CTVC, because they can guarantee that restricted content cannot be viewed by users who are not allowed to see it – e.g. minors or users in countries with strict Internet controls. For instance, in the example of Section 4.1, although the majority of the video and audio content is stored on the CTVC server, the archive material and commercial music tracks used in the soundtrack, still reside on the servers of the respective publishers. Such publishers may provide CTVC with an encrypted key (embedded in the license) in order to allow it to subscribe to the specific information items purchased by CTVC.

While flexible story creation has long been known in content production, it is the tight integration with access and delivery aspects, enabled by ICN, which opens the potential for new services. We assert that the simplified mapping onto networking concepts allows for reducing the overall complexity of solutions, making them easily affordable for a larger number of content producers, including end users, stimulating the market for user generated content. We also suggest that the tight integration with underlying delivery mechanisms increases the trust in the overall expected experience. This, in turn, will stimulate the development of personalised content from the content production industry since the perceived risk of failure is lower compared to today's solutions, which are often hampered by bad delivery experiences, lack of QoS and lack of content adaptation due to coarse-grained division of storylines.

4.4.4. Delivery

The state of the network will likely have a significant effect on the delivery of the media. Factors such as time of day, available bandwidth, end-user device, intermediate caches, will all have an effect on the quality and efficiency of the media delivery. As various broadcasters and viewers consume the material, the network becomes aware of the location of the individual chunks, which are then replicated across the network. Specific media objects can be replicated (for example only the German audio of a film) rather than the whole media file.

From a networking perspective, such information could be very useful, since network operators could apply their own optimisation criteria. For instance, the differentiation of caching functionality based on the nature of the content could possibly avoid extra unnecessary congestion within their networks. In addition, network providers could plan necessary capacities based on the availability of media objects that are relevant to given stories, potentially increasing the overall efficiency of the transport network when dealing with, for example, content virality.

The inherent network awareness of the ICN approach (from the perspective of content delivery) could allow network providers to offer better guarantees for media delivery at lower risk and higher efficiency. This is in contrast to the current IP model of delivery, which only allows guarantees for certain types of traffic at high, often-prohibitive cost. It also prohibits any fine-grained price differentiation for delivery of media, locking the network providers into a flat pricing scheme with little room for service and therefore price differentiation. It is here that the ICN approach proves beneficial by exposing individual information items, which allows for finer differentiation of the provided service. The authors in [26] discuss the problems of today's reservation schemes and the impact on network providers with respect to price differentiation as well as the potential for ICN to change the game yet again.

This information awareness within an information-centric network is contrasted by the approach taken in the current Internet, where such information awareness is only achieved at great cost. Approaches such as MPEG DASH [27] allow for assembling content through individual media assets, similar to our EDLs. However, the objects here exists at the level of HTML, requiring complex content delivery network (CDN) solutions to achieve an efficient distribution of the individual assets, effectively creating the information awareness within the IP network through a variety of architectural extensions such as DNS hijacking and others. Furthermore, replaying the assets from a CDN instead from a original content server might also lead to the playout of material to unauthorised users since the original HTTP-level authorisation mechanisms are circumvented (e.g., Facebook images retrieved from a CDN can be displayed despite possible contrary Facebook viewing policies).

4.5. Deploying on other ICN architectures

We have described how the M-SDN compares unfavorably in the current Internet, but what of other information-centric networking paradigms? The M-SDN makes exclusive use of the middleware (and its API) to interact with the Blackadder prototype. In that respect, a new architecture would only require a new or modified middleware layer. Hence, the real question is how would this middleware layer look for other architectures?

As we have stated before, our middleware layer directly utilises the metastructures provided by the architectural backdrop provided by our ICN solution [1], eliminating the need for complex mapping functionality in the middleware and therefore providing a relatively thin layer of adaptation. It stands to reason that architectures that use similar graph-like metastructures, such as [8], may require only a small modification of the middleware. Other architectures, such as DONA and NDN (those that use flat- or strict hierarchical naming schemes), may be less compatible. In such cases, we would anticipate that additional mapping functionality may be required in order to provide the same level of network expressivity and application functionality, ultimately increasing the complexity of the resulting middleware layer.

Alternatively, we can strive to provide compatibility through a universal naming scheme as defined through a sound analysis of the naming requirements [28]. By steering ICN towards such naming standards, the opportunities for middleware solutions are improved by providing a common persistent identification abstraction to applications such as M-SDN. Nevertheless, such persistent identification schemes in turn require functionality to exist in the middleware that maps onto naming schemes of specific ICN architectures; this mapping functionality similar to the functionality outlined above. It is worth noting that the author in [28] specifically singles out the architectural backdrop of Section 2 (i.e., the efforts described in [1]) as best supporting a generic persistent naming scheme; a naming scheme that would directly support our M-SDN use case.

One could argue that a different approach to the middleware itself would make the aforementioned additional mapping complexity unnecessary when considering other ICN architectures. However, we consider the ontological approach of our middleware core to the ability to reason over metastructures at the application level as well as to integrate the metastructures that are part of the network functionality, e.g., in the area of Quality of Service. Such an ontological approach, a W3C standard, could lead the way to more generic information structures based on directed acyclic graphs. While existing efforts in mapping such ontologies onto hierarchical naming schemes (such as provided by the current Internet) show that an ICM as proposed by us would generally work over any ICN architecture, we argue that the ICN architecture presented in this paper is particularly aligned due to its provision of a DAG abstraction at the network interface level. We argue that it is this alignment that reduces the complexity of mapping the metastructure compared to other ICN architectures, while not ruling out the use of alternative ICN approaches.

We have already asserted that this middleware has been designed to optimise functionality specifically for our own architecture. We also maintain that other architectures that use similar metastructures could also work efficiently with the middleware, while others might require additional effort. Therefore, while the middleware cannot be described as entirely generic, we do argue that it provides a solid starting point for information-based architectures.

5. Critical discussion

The following section provides some specific qualitative analysis of our contribution. We first describe the benefits of formally specifying the metadata that annotates our publications, a key aspect of our middleware. We then provide some discussion on the complexity and expressivity of the middleware in order to understand some of the constraints on the system. Finally, we present some socioeconomic issues for personalised story delivery in general and for the M-SDN in particular.

5.1. Formal metamodelling

We have proposed a formal specification of all metadata that enters the network. Building a metamodel using a structure such as an ontology presents several benefits. Ontologies are specifically designed to allow the representation of concepts within a domain and the relationships therein. We can use semantic reasoners to infer the logical consequences of explicit axioms to determine consistency, subsumption, and other factors that can affect the efficacy of the metastructure. In other words, we can demonstrate the correctness of the metamodel and, as importantly, we can demonstrate the incorrectness. We suggest that semantic validity can be preserved at the network level and, for the first time, we can attribute a degree of confidence to the consistency of the networking information space.

Every publication within our network has an associated formal description encapsulating its semantic meaning. Every piece of data is described with consistent metadata and, as this metadata is included in the metamodel, it benefits from the same demonstration of consistency. The level of description that can be attributed to each publication depends on the expressiveness prescribed by the metamodel.

In turn, subscriptions can now leverage the publication semantics in order to get a handle on the data. We can now subscribe according to any concept or relation we have modelled. The capabilities of the subscriptions depend on the expressiveness of the publications: the higher the expressivity of the publication, the more expressive you can make the subscriptions. In other words, we suggest that this service model significantly advances the simplistic model of scope/label-based operations that is exposed at the network layer.

5.2. Complexity of the middleware

We have suggested that middleware technologies that focus on information-centricity can provide natural and mutually beneficial relationships for ICN architectures. This is due to the significant similarities of some technologies, such as ontologies, to constructs used in ICN architectures. We claim that these similarities have a positive effect on the effort required to develop the middleware and also reduce the complexity of the resulting implementation. We can support this claim by making reference to the metadata transformation described in Section 3.2.2. The steps involved in this transformation are, in the authors' opinion, minimal and, furthermore, are achievable using existing ontological tools in a fully automatic way. We recognise, however, that a more systematic evaluation of this effort level should be performed for a final assessment.

Both the complexity and scalability of the middleware are intrinsically linked to the expressivity of the chosen meta-structure and supporting languages [29]. Given that these factors are typically at odds, we must find a metamodel that provides a balance to allow us to express the required types of information. For each metamodel, we have to determine the trade-off to select the appropriate level of expression. The ontological toolset, presented in Section 3.2.2, is one such effort to balance the expressivity differential between the middleware and network metamodels.

It is important to note that we are not advocating the complete or exclusive use of ontologies throughout the entire system. It is the choice of the designer whether to utilise the middleware structures, such as ontologies, while having the option to use simpler but possibly more efficient metamodels directly at the network level.

5.3. Expressivity of the middleware

It should be noted that by employing middleware, we add complexity to our system. Middleware adds a new layer within which to route traffic and this reduces the performance between the application and network. It has been noted that it is impractical to tightly integrate ontological reasoning with performance-intensive processes [21] and as such, the performance overhead needs to be investigated closely in particular for real-time scenarios.

Furthermore, we note that the expressivity of ontologies significantly affects the performance, for example when using reasoners to determine the consistency of the ontology. For our middleware to be effective, we will need to engage the appropriate ontological design in order to ensure scalability and adequate performance. We currently have not enforced any limits on the expressivity of the middleware ontologies in order to leave them open to whatever expressions are required by the ontological toolset at this prototypal stage. But we may have to think again when considering performance issues at the scale of a large area network and after a through analysis of the ontological requirements.

These critical points, however, apply generally to any deployment of (ontology-based) middleware developments. We assert, however, that basing such middleware within an internetworking layer that directly operates on information reduces the complexity that any such indirection through a middleware might cause. Furthermore, we contend that such complexity reduction will further favor the general benefits of introducing such middleware solutions through the higher-level abstraction that is offered to the application developer.

It is imperative to determining the meaning of complexity in this case. Apart from the complexity introduced through the various technologies, it is crucial to recognise that shifting functional parts, such as discovery as well as information-centric routing from the middleware into the internetworking layer, reduces complexity in terms of developing any middleware solutions since these functional components can now be assumed to exist in the lower layer. This is likely to have an impact on required processes for developing solutions in the middleware space, ranging from requirements engineering over standardisation to the development of a final solution. Therefore, apart from focussing on determining the overall performance of our proposed middleware at the runtime level, we aim to investigate the effort required for middleware development, based on well-defined metrics for the various phases of the development.

5.4. Quantitative analysis

We now provide some quantitative analysis and discussion of the M-SDN and the ICM. For a thorough analysis of the network layer, the Blackadder prototype, please refer to [30]. The M-SDN has been deployed, tested and showcased within an international deployment of the Blackadder prototype [6]. This is a testbed that spans several European, US and Japanese nodes, utilising the current Internet via an OpenVPN overlay. The M-SDN Catalog is published via a set of nodes housed at CTVC [22]. The M-SDN Player is available to any node connected within the testbed either via the desktop or a new GWT web-app, which allows the M-SDN player to be run on a testbed node without any specific ICN software installed (except for OpenVPN that allows the node to connect to the testbed).

While quantitative analysis of the middleware is ongoing, we have some preliminary results that convey the middleware application as a viable entity in terms of CPU and memory management. The following analysis uses a set of ontologies extracted from the caGrid infrastructure described in the work carried out in [31]. We chose these ontologies as (1) they were well ranged in terms of complexity and size (similar to what one might expect from an application ontology) and (2) they had already been evaluated within a comprehensive performance suite, described in [32].

One of the issues with ontological metrics is that size (or number of axioms) is not always the best (or even a good) metric for describing an ontology. Ontologies may differ in terms of attribute-, inheritance-, or class-richness, such terms are described in [32], and these often better reflect the complexity of the ontology. The problem, however, is that ontologies vary from one to another in these aspects and each will affect the results differently. Therefore, in the absence of a single, preferred metric for measuring the ontologies, we defer to size (number of axioms). The following results were obtained from a standard desktop PC computer (Intel i5, 8 GB Ram, Debian 6.0).

Fig. 10 illustrates the overhead of the middleware and, separately, the overhead of the reasoning process[3] of each ontology as it occurs during deployment within the middleware. The reasoning time represents the time taken to generate all inferred axioms of the ontology. The middleware overhead captures the time required to process the complete ontology and publish all scopes and items to the rendezvous. In reality, it is unlikely that this would happen on a regular basis but we use it here to denote a worst-case scenario. It would be far more likely that most changes to the ontology would be frequent and small, requiring even less overhead from the middleware. Even so, the overhead from the middleware is manageable and appears to increase linearly.

[3] The Pellet reasoner (http://clarkparsia.com/pellet/), an OWL 2 reasoner for Java, was used for the performance evaluation.

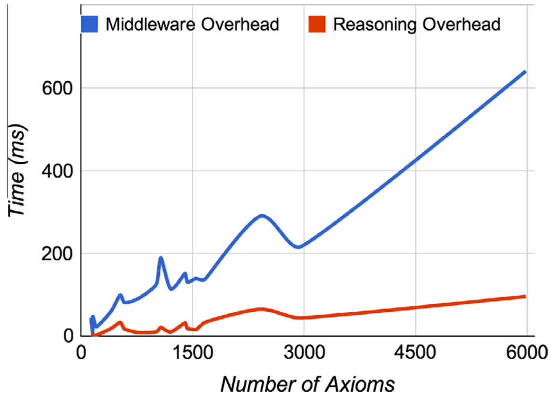


Fig. 10. Middleware and reasoning overhead during deployment.

Fig. 11 shows the load (megabytes in use on the heap) of a published ontology on the RV node relative to a baseline value (an empty RV node). Given that the load increases with apparent linearity relative to the size and also remains fairly lightweight (<25 Mb for 5000 axioms), it would suggest that such an implementation could be housed within a standard web server without requiring any specialised equipment.

We have looked at the deployment overhead of the middleware in Figs. 10 and 11 illustrate the deployment and runtime memory usage. We now discuss the middleware overhead during some of the runtime operations we have presented in this paper. To this end, we refer to three scenarios:

1. *Subscription by (R/S)ID.* This scenario assumes we already have the RID (or SID) of the required artifact. For example, when using the video catalog to subscribe to a static media item (i.e., not an EDL media story), we subscribe using the respective RID. In this scenario, there is no middleware overhead as the RID subscription is routed directly through the network-layer API (described in Table 1).
2. *Browsing* (as in Section 3.3.1). Browsing requires a search of the ontological metastructures in order to reveal the graph structure, including both direct subclasses and instances, to the participating application. This process, as it pertains to the middleware, consists of retrieval of the required metadata from the ontology, the bundling of that metadata into a network packet (at the publisher), and the unbundling of the same packet (at the subscriber). This as the middleware browsing overhead and an analysis of this overhead is presented in Fig. 12. For this analysis, we use a simulated metastructure that will guarantee a specific number of results to retrieve and then measure the time taken to answer the request. Measurements were taken for retrievals of up to 1000 results (increments of 10) and each experiment was repeated 100 times with the average values presented in Fig. 12.
3. *Querying* (as in Section 3.3.2). In order to answer a query within the middleware, we first build an axiom for that query, adding it to the ontology. The reasoner evaluates this new axiom and infers any potential individuals or

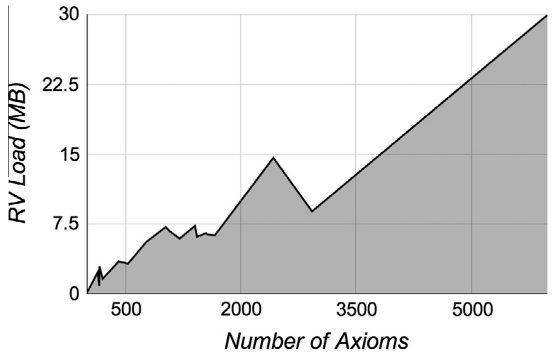


Fig. 11. Middleware load on the rendezvous.

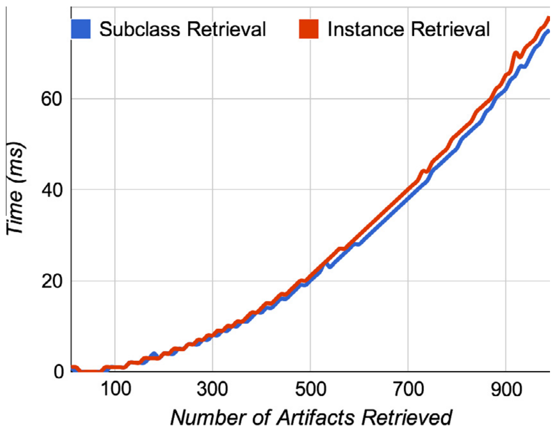


Fig. 12. Middleware overhead for subclass and instance retrieval.

classes satisfied by the query. These satisfied individuals, in our middleware framework, are the results of our query and it is those results that are returned to the application. In terms of measuring this overhead, we have already covered the required aspects in our previous analysis. Fig. 10 provides an indication of the overhead of running the reasoner over a given ontology and Fig. 12 shows the overhead of retrieving the results and returning them. The remainder of the process is the addition of the ontological axiom, which is a simple and quick process (even compared to the measured elements above) and, in the opinion of the authors, does not require specific analysis.

Given the issues mentioned above in selecting an appropriate metric against which to measure the ontological complexity, these analyses are limited in terms of what we can infer about the middleware's capability to handle such complexity. Rather, we have suggested that the analyses provide support that the middleware (and the use of ontological metamodels) is a viable option, not only for continued research but also for a potential real-world deployment and we make the following arguments in support of this.

1. Fully consistent application domain models can be deployed in a timely manner. All of our test ontologies could be fully deployed in well under a second.
2. The middleware exhibits a small footprint on system resources. Even our largest ontology occupied less than 30 Mb of memory in the middleware.
3. Run-time performance is admirable. Standard information retrieval of 1000 data items occurred in less than 80 ms.

The use of a reasoner is sometimes required (i.e. when answering a user-query) and, in such cases, the middleware can be adversely affected by the additional reasoning overhead. There are ways to improve reasoner performance. For example, if we were to settle on a particular level of ontological expressivity, which excludes unused expressions, we could select a reasoner specific to that level and achieve performance benefits from the resultant simplified reasoning process. We do, however, suggest that more analysis is required and plan to fully explore the performance of the middleware, particularly in terms of scalability and expressivity of the metamodels.

5.5. Socio-economic impacts of the M-SDN

Although the M-SDN aims at delivering independent contents that constitute a story, it will also have a significant impact on Internet economics as well as on the flow of money between the various stakeholders within the Internet ecosystem. Similar technical solutions, such as MPEG-4 or fMP4 [24], aim at improving the efficiency of data delivery, similar to the M-SDN. However, our proposal also incorporates the socioeconomic aspects for the delivered data. The metadata of each individual information item may contain information related to pricing, Quality of Service, copyright, etc. Contrarily, the metadata within MP4 or MPEG-4 files are usually related to multiple camera angles, or different language tracks and do not capture socioeconomic aspects. Out of the many potential socio-economic impacts that our proposition might yield, in this section, we focus on two potential key aspects, namely that of *price differentiation* and the evolution of *Internet business models*.

The current Internet does not provide sufficient market mechanisms for the involved stakeholders in order to express their preferences and therefore request a differentiation of the service they intend to consume. This has led to the best-effort service becoming the de facto standard service for an end-user to receive specific information within IP networks. As a result, IP end users usually open multiple TCP connections, which must all share the available bandwidth, regardless of potential user preference. An obvious example could be the case of giving lower priority for a software update, while giving a higher priority to incoming e-mails. Although reservation schemes for service differentiation do exist (see [26] for a discussion on this), the costs for realising these schemes for a wide range of services and content are prohibitive at scale. The result is that end-users have Service-Level Agreements (SLAs) only with their immediate ISPs, such contracts usually being based on either flat charging for connectivity or volume-based charging (especially in 3G networks).

Embedding our proposition within the context of an ICN architecture enables a platform under which end-users will be able to request differentiated services, in turn enabling network providers to optimise their story delivery in line with the expressed preferences of their subscribers. It is this expression of differentiation (and therefore choice) that could lead to fairer pricing and social welfare maximisation. We assert that the M-SDN proposition enabled in an ICN context can provide new market mechanisms for content delivery as well as to offer differentiated Internet SLAs. For instance, [26] suggests pricing mechanisms implemented per information item or per scope, which allows for aggregation of pricing strategies with the potential for accounting to be implemented at the appropriate points in the network. This scheme (pricing per content item, instead of per bit) will also reveal the owner of the QoE as expressed in [33]. We would expect a move from flat pricing models (for connectivity) to content-based pricing along the entire delivery chain potentially spanning many ISPs.

Our M-SDN proposition could also affect the current business models in the Internet value chain. Nowadays, ISPs have limited incentives to make new investments due to the inefficient business models in the current Internet. In particular, there is an exponential growth of Internet traffic due to the new applications and the increasing demand for video streaming (i.e. YouTube). As a result, ISPs are obligated to invest on capacity expansion of their networks in order to meet this new demand, although this does not have any directly positive effects on their revenue. In particular, the content providers increase their profits by providing new services but ISPs that transfer that content usually do not increase their profits and do not have incentives to make new investments to increase the QoE for their customers.

As discussed in [34], there is a need to change the current flow of money in the Internet content market, moving away from the business entity that transfers the content, towards the one that creates it. We can foresee business models in a M-SDN platform, where new SLAs define charging models for uploading an information item (with each of the individual charges being aggregated to individual stories that are delivered as an experience to end users). In practice, content producers could pay for publishing their content, whilst in turn being paid by the end-users who download this content. The ‘visibility’ of information throughout the network could enable the required money flow that will enable the desired Quality of Experience that the content owners intend to enable, e.g. through service differentiation at the network provider level, tying back into our discussion on price differentiation at the network provider level. Additionally, such information exposure would give end-users the power to monitor their SLAs on a *per information* basis, although this raises issues of privacy as network providers could conversely monitor users on a *per information* base.

We also see a potential impact on the interconnection market since individual network providers will have a more detailed view of both their network utilisation as well as transit needs given the exposure of information items at a network level. This change in view could increase the incentives for transit providers to provide additional caching services to stub providers rather than transparently transit requested traffic.

6. Conclusions

The current Internet has greatly benefitted from the availability of suitable application development environments that distance the developer from the details of IP-level programming of applications by providing more suitable abstractions for a particular range of applications. While IP provides the common ground at the network level for internetworking, it is this variety of application environments that drives the tremendous innovation we have seen in the current Internet so far. Such variety of application environments, however, is entirely missing for information-centric networking. In fact, any application environment is missing at the moment, limiting ICN applications to simple demonstrators of particular features and benchmarking tools.

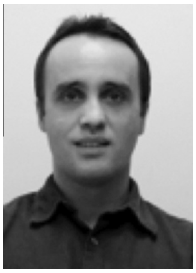
We argue that ICN has reached the maturity at architectural level to start the necessary work on how such application environment might look like, one that directly utilises the information-centric concepts exposed by the networking level. Our work suggests that the resulting middleware can exist as an extremely thin layer, given the compatibility of the middleware and network constructs as well as the generic middleware functionality that, thanks to the abstractions, has been subsumed by the network layer. We have exemplified this in the form of an advanced application, the M-SDN, which tackles a real-world problem requiring the personalised, democratised and distributed delivery of media experiences. The M-SDN leverages the tools, services and abstractions provided by the middleware and, ultimately, the network. While our work does not provide a final answer regarding the efficacy of an information-centric networking middleware, we provide first insight on how the design for an application environment in a new ICN world could look. We see our work as starting the necessary developments in this area.

References

[1] D. Trossen, M. Sarela, K. Sollins, Arguments for an information-centric internetworking architecture, SIGCOMM Comput. Commun. Rev. 40 (2) (2010) 26–33.
[2] V. Jacobson, A New Way to Look at Networking, 2006. <http://yovisto.com/video/19620>.
[3] L. Popa, A. Ghodsi, I. Stoica, HTTP as the narrow waist of the future Internet, in: Proc. of Hotnets, 2010.
[4] The Pursuit Project, 2012. <http://www.fp7-pursuit.eu/PursuitWeb/>.
[5] V. Jacobson, D.K. Smetters, J.D. Thornton, M. Plass, N. Briggs, R. Braynard, Networking named content, Commun. ACM 55 (1) (2012).
[6] G. Parisis, B. Tagger, D. Trossen, D. Syrivelis, P. Flegkas, C. Stais, C. Tsilopoulos, G. Xylomenos, Demonstrating an Information-Centric Network in an International Testbed, in: Proceedings of TridentCom 2012, Greece, June 2012.
[7] G. Tyson, A Middleware Approach to Building Content-Centric Applications, PhD Thesis, Lancaster University, 2010.
[8] G. Tyson, A. Mauthe, S. Kaune, P. Grace, T. Plagemann, Juno: an adaptive delivery-centric middleware, in: Consumer Communications and Networking Conference (CCNC), 2012.
[9] D. Trossen, G. Parisis, Designing and Realizing An Information-Centric Internet. Communications Magazine, Special Issue on Information-centric Networks, 2012.
[10] Ian Horrocks, Ulrike Sattler, Ontology reasoning in the SHOQ(D) description logic, Proceedings of the 17th International Joint Conference on Artificial Intelligence – Volume 1(IJCAI'01), vol. 1, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 199–204.
[11] S. Zhexuan, A.A. Cárdenas, R. Masuoka, Semantic middleware for the Internet of Things, Internet Things (IOT) (2010) 1–8.

[12] O. Lassila, Applying semantic web in mobile and ubiquitous computing: will policy-awareness help, in: Semantic Web and Policy Workshop, 4th International Semantic Web Conference, 2005.
[13] M. Klein, A. Schmidt, R. Lauer, Ontology-centred design of an ambient middleware for assisted living: the case of SOPRANO, in: AIM-CU, 2007.
[14] S. Verstichel et al., Ontology-driven middleware for next-generation train backbones, Sci. Comput. Program. 66 (1) (2007) 4–24.
[15] J. Wang, B. Jin, J. Li, An ontology-based publish/subscribe system, in: Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware, 2004, pp. 232–253.
[16] L. Gomez, A. Laube, Ontological middleware for dynamic wireless sensor data processing, in: proceedings of SENSORCOMM '09, 2009, pp. 145–151.
[17] X.H. Wang, D.Q. Zhang, T. Gu, H.K. Pung, Ontology based context modeling and reasoning using OWL, in: Pervasive Computing and Communications Workshops, IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.
[18] A. Devaraju, S. Hoh, Ontology-based context modeling for user-centered context-aware services platform, ITSim 2 (August) (2008) 1–7.
[19] D. Preveneers et al., Towards an extensible context ontology for ambient intelligence, in: proceedings of the Second European Symposium on Ambient, Intelligence, 2004.
[20] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web. Scientific American Magazine, May 17, 2001 (Retrieved 26.03.08).
[21] A. Agostini, C. Bettini, D. Riboni, Loosely coupling ontological reasoning with an efficient middleware for context-awareness, in: MOBIQUITOUS '05, 2005, pp. 175–182.
[22] CTVC. <http://www.ctvc.co.uk/> (accessed June 2013).
[23] The UK Copyright Service. <http://www.copyrightservice.co.uk/copyright/p09_fair_use> (accessed December 2012).
[24] T. Siglin, Unifying Global Video Strategies: MP4 File Fragmentation For Broadcast, Mobile and Web Delivery. A Transitions in Technology White Paper, November 2011.
[25] John H. Gennari, Mark A. Musen, Ray W. Fergerson, William E. Grosso, Monica Crubezy, Henrik Eriksson, Natalya F. Noy, Samson W. Tu, The evolution of Protége: an environment for knowledge-based systems development, Int. J. Human–Comput. Stud. 58 (1) (2003) 89–123.
[26] D. Trossen, G. Biczok, Not Paying the Truck Driver: Differentiated Pricing for the Future Internet, ReArch 2010 workshop at ACM Context, December 2010.
[27] Iraj Sodagar, The MPEG-DASH standard for multimedia streaming over the internet, IEEE MultiMedia 18 (4) (2011) 62–67.
[28] K.R. Sollins, Pervasive persistent identification for information centric networking, in: Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking (ICN '12), ACM, New York, USA, 2012.
[29] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web Ontology Language: Profiles. W3C Recommendation, 2009.
[30] G. Parisis, D. Trossen, D. Syrivelis, Implementation and evaluation of an information-centric network, IFIP Netw. (2013).
[31] A.G. Beltrán, B. Tagger, A. Finkelstein, Ontology-based queries over cancer data, in: Proceedings Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2010), Berlin, Germany, December 10, 2010.
[32] A.G. Beltrán, B. Tagger, A. Finkelstein, Federated ontology-based queries over cancer data, BMC Bioinform. 13 (Suppl. 1) (2012) S9.
[33] D. Trossen, A. Kostopoulos, Techno-economic aspects of information-centric networking, J. Inform. Policy 2 (2012) 26–50.
[34] AT Kearney Report, A Viable Future Model for the Internet. <http://www.atkearney.com>.

Ben Tagger is currently working on middleware aspects in the PURSUIT project, developing ontological solutions that directly operate on the abstractions provided at the network layer. Before joining the Computer Laboratory at Cambridge, he held a post-doctoral position at the Department of Computer Science at UCL, working with ontology-based biological data, implementing solutions for ontology-based queries, and the development of an online interface for building and rewriting ontology-based queries.



Dirk Trossen is a Senior Researcher in the Computer Laboratory at Cambridge University. He is the technical lead for the EU project PURSUIT. He held prior positions as a Chief Researcher at BT Research and as a Senior Principal Scientist at Nokia Research. He is currently a Research Affiliate with the Advanced Network Architecture group at MIT CSAIL. He holds a PhD from the Aachen University in Germany and published more than 65 papers and holds 27 international patents.

Alexandros Kostopoulos is currently a Ph.D. candidate at the Athens University of Economics and Business (AUEB), Department of Computer Science, under the supervision of Prof. C. Courcoubetis. ∗He is a member of the Network Economics and Services Group (NES Group) and the Mobile Multimedia Laboratory (MM Lab).

Stuart Porter joined CTVC in 2007 to co-ordinate the teams of film-makers making content for TrueTube. Stuart also manages CTVC's commitment to technical innovation. CTVC is currently a partner in two research projects SARACEN and PURSUIT which are funded by the EU's FP7 program, as well as one project funded by the UK's Technology Strategy Board.

George Parisis received a Ph.D. in Computer Science from Athens University of Economics and Business in 2009. He is now a Research Associate in the Computer Laboratory at Cambridge University. His research interests include information-centric networking, publish/subscribe systems and high-performance storage systems.