What are the Services of an Information-centric Network, and Who Provides Them?

Anders Eriksson, Börje Ohlman, Karl-Åke Persson
Ericsson Research
Ericsson
Stockholm, Sweden
{*Anders.E.Eriksson, Borje.Ohlman, Karl-Ake.Persson*}*@ericsson.com*

***Abstract*—Various Information-centric Network (ICN) services have been proposed in the literature, such as content distribution, publish-subscribe, event notification, and search. Such services have traditionally been described as separate from one another, and little attention has been given to the issue of a common ICN architecture within which they all can interact efficiently. In this paper, we describe how information-centric services can interact within one architectural framework to provide a rich ICN service offering related to Information Objects, such as content and data objects. We give examples of how the architecture can support various application domains, for example content distribution, machine-to-machine communication, and interactive and live streaming applications. We also propose the business role of an ICN Service Provider, which adds value by composing a service offering of a variety of ICN services. The contribution of the paper is to highlight the need for efficient interaction between multiple ICN services in order to provide an attractive and complete ICN service offering, to describe an architecture for such interaction, and to identify the business role of an ICN Service Provider.**

***Keywords*-Information-centric; architecture; service model.**

I. Introduction

Information-Centric Networks (ICN) is a relatively new research field. There are currently a number of approaches being developed, e.g., NetInf [1], CCN/NDN [7], PSIRP [8]. There have also been some attempts to define common mechanisms and characteristics that should be generic to ICN [2]. In this paper, we make an attempt to define ICN at a mechanism-independent service level. The core idea of ICN is to move the focus from devices and network entities to the Information Objects (IOs) themselves, i.e., content, data files, RFID tags, web pages, sensor data, application instances, etc. ICN should thus be defined by how we can interact with these IOs, i.e., by defining the services needed to interact with them.

The service offered by an IP network is primarily store-and-forwarding of IP packets. What are, then, the services offered by an Information-Centric Network? Judging from the papers published on this topic summarized in [2], an ICN primarily offers an optimized transfer service for IOs, similar to that of http. Using this service, clients can retrieve named IOs from storage or streaming servers in an anycast fashion. Some network architectures also offer a subscribe and event notification service. Other services needed for a full-blown ICN, like an advertisement service, are not, as far as we have seen, described in the proposed approaches.

Much of the ICN research focus on an http-like transfer service which results in a research agenda with topics and mechanisms well-known from TCP/IP research, such as routing, congestion control, mobility management, etc., although with an information-centric touch. In this paper, we argue that there is more to Information-Centric Networks than this. An ICN allows for a significantly richer service offering in addition to that of transfer of IOs. A key ICN service is search for IOs and metadata associated with IOs. Traditional Internet search engines offer unpredictable performance in terms of when or whether a published resource will be reachable via search. An ICN search service should offer timely reachability of published IOs. One of the things we discuss in this paper is how the interaction between an ICN Search Service and other ICN services, such as Event Notification, Advertisement, and Storage allows for a more predictable Search Service.

When discussing ICN services, it is of course relevant to describe business roles, service providers, and interdomain interfaces. The paper outlines a business model for ICN using open and non-proprietary interdomain interfaces.

The remainder of this paper is organized as follows: Section II describes a horizontal application-independent ICN infrastructure. Section III describes the semantics of the interaction between ICN services on the one hand, and the interaction between these services and the ICN Publishers and Subscribers on the other hand. Section IV describes how this interaction model applies to different use cases. Section V describes business roles and interdomain interfaces between ICN services and service providers, as well as ICN metadata relevant for these interfaces. Section VI discusses related work. Section VII draws conclusions and proposes next steps in the development of an ICN service model.

 ISBN: 978-1-61208-238-7

II. Application-Independent Horizontal ICN Services

As mentioned in the introduction, a number of ICN approaches are currently being proposed. Which of them will be successful only the future can tell. Potentially they could prove to be successful in different parts of the network, e.g. the core network, highly dynamic edge networks or sensor networks. Irrespective how this will turn out we see that they will offer a very similar set of services. By defining these services, including standardized Application Programming Interfaces (APIs) and interdomain interfaces we can provide a stable environment for application developers. Likewise, an interdomain interface which hides domain-internal ICN mechanisms will enable interoperation between ICN domains employing different domain-internal mechanisms.

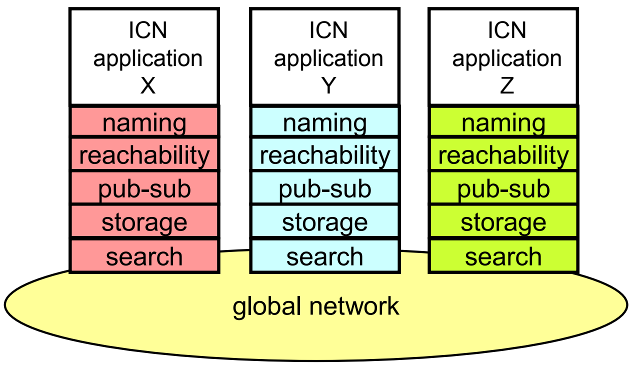

Figure 1. Information Object lock-in per vertical application.

Figure 1 shows several vertically integrated ICNs, each dedicated for a single application and operating as an overlay on a global network. Such vertically integrated ICNs have been employed for content distribution, social networks, peer-to-peer communication, and machine-to-machine communication, e.g. Skype, Spotify and BitTorrent. Each of the overlay ICNs has a separate name space, and a separate name resolution system. This has several implications:

- Each vertically integrated ICN requires dedicated systems for management of information object names, name resolution, search, and publish-subscribe.
- The IOs are locked-in within one vertical ICN and cannot be reached from other verticals. As a consequence, it is hard to develop an application that communicates with IOs belonging to different verticals. The limited access to such IOs diminishes their usefulness.
- The barrier of entry for a new vertical ICN with global coverage is rather high, since it requires deployment of new systems for management of IO names, name resolution, reachability, search, and publish-subscribe.

The lock-in effect on IOs ultimately results in a lock-in effect on end-users, which become tied to specific applications, since only those applications can communicate with the favourite IOs of the end-users. Likewise, the renaming effort needed to port a large number of IOs from one vertical to another may be prohibitive.

The issues associated with vertically integrated ICNs can be overcome by means of a horizontal infrastructure for management of IO names, name resolution, reachability, publish-subscribe, storage, and search as shown in Figure 2. The infrastructure is horizontal in the sense that it can serve as a global and open platform for a multitude of applications.

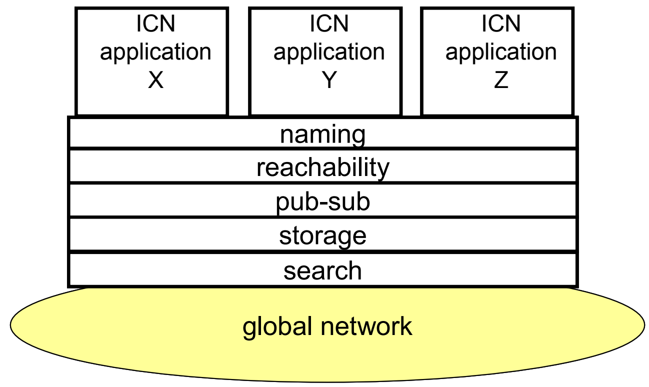

Figure 2. Horizontal infrastructure allowing multiple applications to communicate with Information Objects in order to avoid object lock-in.

With the horizontal approach, each IO has an application-independent name which can be resolved into a globally unique locator by a Name Resolution Service (NRS). Any application can thereby reach any IO. This allows for a variety of applications communicating with a specific IO, which adds to the usefulness of the IO. In addition, an application can communicate with a variety of IOs of different types. This facilitates innovation of novel applications which cannot be supported by rigid vertical ICNs dedicated for a specific application and set of IO types.

To allow for internetworking across a variety of access technologies, the horizontal infrastructure should not only be independent of applications, but also of access technologies. Moreover, the application-independent horizontal ICN infrastructure should be designed so that it can handle arbitrary types of networked objects, such as data files, RFID tags, and persons. This will enable novel applications that can interact seamlessly with such objects.

III. Interaction model for ICN services

We claim that the emergence of a monolithic ICN service, in analogy with the best-effort store-and-forward service of an IP network, is unlikely due to the multitude of ICN use cases. Instead, we propose that an ICN will provide a set of services. Figure 3 shows a set of services provided by an ICN, and the semantics of the interaction between these services, as well as with the Publishers and Subscribers of the ICN. Interaction between the services allows for a more

ISBN: 978-1-61208-238-7

powerful ICN than a set of non-interacating services. The set of services amd their interactions are selected to handle a range of use cases, see section IV.

In the model shown in Figure 3, the Publisher and the Subscriber are end users of the ICN. A Publisher publishes IOs, which can subsequently be accessed by a Subscriber. The various services shown in the figure support different aspects of this interaction between the Publisher and the Subscriber as will be described below.

The IO has a URI and can be accessed over the ICN. The Publisher assigns a URI to the IO. To inform a potential Subscriber about the URI and the metadata associated with the IO, the Publisher uses an Advertisement Service. Subscribers use the Advertisement Service to find information about an IO before it is eventually published via the Storage Service or via streaming. The Advertisement Service can use any ad-hoc mechanism external to the ICN, such as random web pages or e-mail lists. This is currently a rather common way of advertising IOs. Alternatively, the Advertisement Service can be included in the ICN, and be accessed via an ICN API.

When a Publisher assigns an URI to an IO, this URI may not include any information about the location of the IO. For example, if the IO is nomadic or mobile, location information may not be included in the URI. In this case, there is a need for a NRS to resolve a location-independent URI into a URL, which includes information about the locator of the IO. The Publisher registers the binding between the URI and the URL in the NRS. The Publisher may also register metadata associated with an IO in the NRS.

When a Publisher advertises an IO with its URI and metadata via the Advertisement Service, it also registers the IO with the Event Notification Service. The Subscriber can then subscribe to the IO via the Event Notification Service to receive notifications about events related to the IO, for example when it is published via the Storage Service or via streaming.

A Storage Service allows for access to stored IOs. When a Publisher publishes an IO, it also stores the IO in the Storage Service. The Storage Service returns a URL for the stored IO to the Publisher, which can use this URL when registering a URI-URL binding in the NRS. The Storage Service may be associated with the Publisher, or it may be a third-party service.

There is a Processing Service associated with the Storage Service. The Processing Service processes stored IOs, and thus produce new IOs, which in turn are stored by the Storage Service. The Processing Service notifies the Publisher when a new IO has been stored, and the Publisher can decide to publish this IO and register it in the Name Resolution Service in order to make it available for Subscribers.

A Search Service can keep track of all advertisements, and decide to subscribe to all subscription names learnt via the Advertisement Service. The Search Service will then be notified about all events related to all URIs registered with the Event Notification Service.

In addition, the Search Service can index information stored in the Storage Service, and also index URI-metadata bindings in the Name Resolution Service. This type of Search Service will be able to immediately detect when IOs are published, or when they change state. This allows for shorter and more deterministic response times compared to traditional Internet search systems based on web crawling.

A key aspect of ICN is the use of caching. Caches in the ICN store IOs which are retrieved from the Storage Service. This reduces access times and network load.

The abstract service model described above can be instantiated in various fashions, with different sets of protocols and mechanisms to invoke the services. Of course, the implementation details of an instantiation, including the selection of protocols, depend on the concrete set of use cases at hand.

IV. ICN Use Case Examples

In this section, we present some examples of how our service model for ICN can be applied to specific use cases.

A. *Sensor networks*

An example of an information flow in an information-centric sensor network is shown in Figure 4. Sensors collect data from Entities of Interest (EoI), for example the temperature at a specific location. Each Machine Device (MD) has a Publisher function, a network interface, and hosts one or many sensors. The sensor data are stored and processed by a Storage and Processing (S&P) function in the MD. As a result, the sensor data is transformed into an IO which is stored in the MD, and also published by the Publisher function in the MD.

Dedicated S&P nodes in the information-centric network subscribe to IOs published by MDs, or published by other S&P nodes. The S&P nodes process these IOs and produce new IOs. For data sets, e.g. sensor data, S&P nodes can create new IOs with data for, e.g., average, sum, or time series. Finally, application nodes subscribe to IOs published by S&P nodes or an MD.

A motivation for separate S&P nodes is that sufficient processing capacity may not be available in an MD, so processing and storage needs to be off-loaded. It is also possible that the application nodes at the top of Figure 4 are lightweight devices which off-load processing and storage to separate S&P nodes.

B. *Content distribution*

In the interaction model shown in Figure 3, content distribution starts with the Publisher storing a content IO in the Storage Service, and then registering the IO and its metadata with the Advertisement Service and the Name Resolution

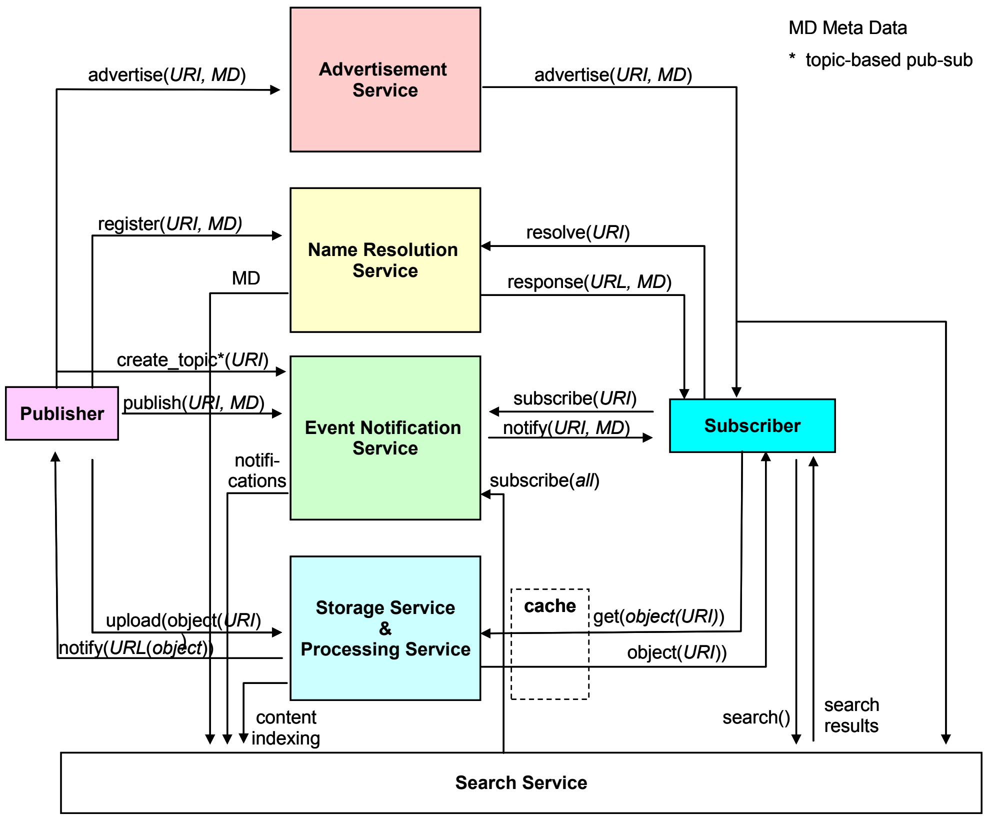

Figure 3. Service semantics of an ICN and interaction between services.

Service. This will trigger the Advertisement Service or the Storage Service to register the IO and its metadata with the Search Service; see the figure. The Subscriber can now search for the IO using the Search Service, which returns a URL for the IO. This URL enables the Subscriber to either retrieve the IO directly from the Storage Service, or make further queries for the IO using the Advertisement Service.

Alternatively, the Publisher can advertise the IO with the Advertisement Service and the Event Notification Service well before it is stored in the Storage Service. This will trigger the Advertisement Service to register the IO and its metadata with the Search Service. The Subscriber can now find the Advertisement for the IO using the Search Service, and then subscribe to the IO via the Event Notification Service. When the Publisher eventually stores the IO in the Storage Service, this will trigger the Event Notification Service to send event notifications to inform all Subscribers of the IO about the URL of the stored IO.

C. *Interactive and live streaming use case*

The first thing to happen in the interaction model is that the Publisher advertises the upcoming service (e.g. telephony call or live football game) through the Advertisement Service. Next, interested Subscribers will subscribe through the Event Notification Service. As content becomes available, the Publisher will make it available by uploading it to the Storage Service, register it with the NRS, and publish it through the Event Notification Service.

Many interactive services have more of a push than a pull character. This is in conflict with one of the key features of ICN, the receiver-oriented operation, which gives the receivers better control of the traffic flows and helps to curb DoS attacks. In the interaction model shown in Figure 3, this can be dealt with in a receiver-oriented fashion by handling a media stream as an IO which is requested by making a subscription for the IO ID that represents the stream. The sender can then push the stream by sending updates to that subscription, which will result in an event notification reaching the Subscriber.

V. INTERDOMAIN OPERATION AND FEDERATION OF ICN SERVICES

In ICN, what constitutes a domain will differ very much from context to context. The minimal ICN interdomain operation might only consist of two ICN nodes that belong to different domains. At the other end of the spectrum, very large ICN providers could offer services like name resolution or storage on a global scale in a federated fashion. The service model in Figure 3 can be applied regardless of the size or number of ICN providers.

 ISBN: 978-1-61208-238-7

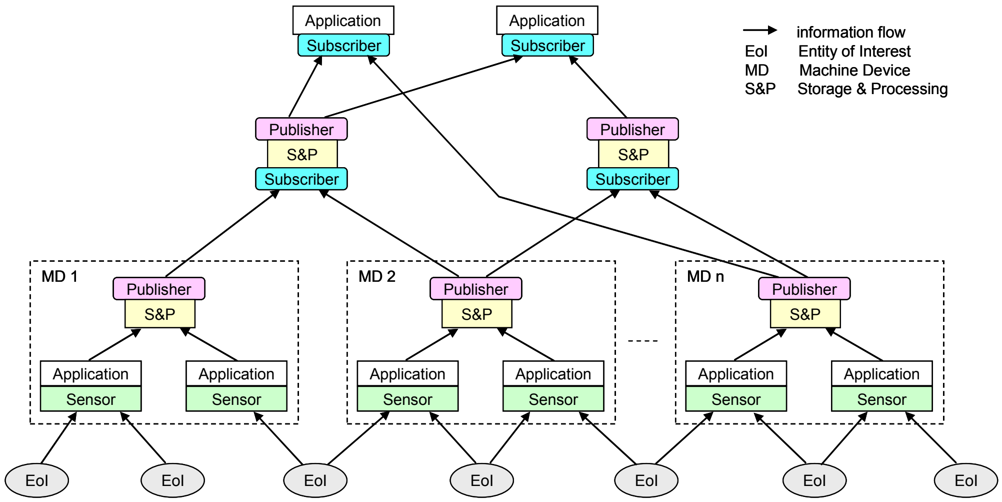

Figure 4. Information flow in an ICN for sensor data

A. Business Roles and Interdomain Interfaces

Figure 3 describes a set of services and the interactions between these. The figure is agnostic with regard to the business roles of the actors implementing these services. In this section we discuss the issue of mapping these services to business roles and actors, as well as describing interdomain interfaces between different business roles and actors. We focus the discussion on open and non-proprietary interfaces between the different ICN services shown in the figure, as well as between the actors implementing the services. The intention of such interfaces is to foster competition in the evolution of various ICN services, and to enable an efficient horizontal infrastructure with open interfaces as shown in Figure 2.

The interdomain interfaces between actors implementing a Name Resolution Service is strongly dependent on the architecture of the NRS. For example, a global and federated NRS based on a distributed hash table architecture will result in each actor managing resource records belonging to other actors. By contrast, a global NRS based on a DNS-like architecture will result in each actor managing its own resource records. This will lead to very different interdomain interfaces between the actors for the two types of architectures with regard to signaling semantics for update and retrieval of resource records, as well as for authentication and authorization.

An ICN Search Service can be implemented by traditional search providers, each offering a separate service having a proprietary internal architecture, but offering open and non-proprietary interdomain interfaces to allow for interaction with other ICN services as shown in Figure 3. Alternatively, a federated Search Service can be considered, where different search providers join in a search federation and use open and non-proprietary interdomain interfaces between each other, as well as between the federated Search Service and other ICN Services.

As illustrated in Figure 3, an ICN Search Service has interdomain interfaces to the NRS, the Advertisement Service, the Event Notification Service, and the Storage & Processing Service.

Contrary to Name Resolution Services or Search Services, there are no well established global Event Notification Services. There are several reasons for this, among which are scalability and real-time performance issues. While waiting for the ultimate event notification service, there may be a need to use a variety of Event Notification Service types, depending on the use case at hand.

There is a possibility for new business roles which provide the ICN services in the model shown in Figure 3. An Access ICN Service Provider adds value by composing a service offering of a variety of ICN services by acting as a broker between end users and Interdomain ICN Service Providers.

B. Metadata

Metadata is in ICN used for a number of purposes including access and security policies. Information contained in metadata is an important part of the interdomain interface. This can be compared with the metadata for Content Delivery Networks Interconnection [4]. In traditional host-centric networking, this type of information is normally stored on the same node-complex that stores the data the policies relate to. In ICN this type of information needs to be tied to the individual IOs (including all copies). How metadata is best stored is still a research issue under investigation. Alternatives include having metadata stored as records in the NRS, or stored together with the IO itself in the IO storage.

Metadata can include security data such as signatures, delegation of rights, and the public key of the publisher. When it comes to ensuring the confidentiality of an IO, ICN can offer a wide range of alternative security levels, starting with fully public IOs which anyone can retrieve if they only know the IO ID, or if they just search for IOs that are matching a set of attributes. A bit more of confidentiality can be achieved by only storing objects in private NRSs. The highest levels of confidentiality is achieved by cryptographic means. By using strong crypto mechanisms and advanced key distribution schemes, very high level of confidentiality can be achieved. There are two main drawbacks of these cryptographic methods. Firstly, the complexity of administrating key distribution schemes, which is well-known. Secondly, when the highest level of access control with individual keys are used for each user and each object, one of the main advantaged of ICN is forfeited, since caching will be useless due to the individually encrypted IOs.

VI. Related work

As mentioned in section I, Information-centric Networking puts retrieval of information objects in focus instead of traditional packet routing and forwarding between nodes. A key idea in ICN is the naming of the information objects with globally unique names, in contrast to traditional networking, where the focus is on naming of nodes and interfaces. ICN can also, in general, be said to be receiver driven, i.e., to initiate the transfer of an information object the receiver requests an object by name from the network. In traditional networks the sender pushes data into the network and the network forwards it towards the receiver.

The ICN approach allows for a network architecture that can exploit a multitude of alternative infrastructures and mechanisms to retrieve the desired information objects. These include, but are not limited to, locally cached copies at nearby nodes, use of multiple access networks in parallel, and information retrieval over pure broadcast networks such as FM, TV, or satellite networks. Also, networks with intermittent connectivity and data mule networks can be utilized in this information-centric paradigm.

A number of information-centric architectures have been described in the literature, such as DONA [9] , NetInf [1], CCN/NDN [7] and PSIRP [8]. These architectures are based on a set of concepts and mechanisms for the retrieval of information objects, such as secure naming of IOs, routing and forwarding of IOs, transport of IOs, and an API between the application layer and the Information-centric Network layer. Key differentiators between these approaches, are if they use hierarchical names or flat names; if they route directly on names or if they rely on some type of name resolution service to map names of information objects into network locations. For a more extensive overview of different ICN architectures and their detailed mechanisms we refer the reader to the IEEE ICN survey paper [2]. That paper as well as [6] are trying to decompose and find the basic components of a generic ICN architecture with the focus on one specific network service, i.e., retrieval of information objects. By contrast, in this paper we propose a set of services and interactions between services to enable a more powerful ICN.

Another important aspect of ICN is the service model of the API, from the application perspective ICN looks more like a database than a traditional end-to-end communication network. In ICN the application makes a request for an information object to the network, not a specific host, by name. The requested information object is then delivered to the application through the API, without any information from which host it was retrieved. This makes ICN resemble Remote Invocation Method (RMI) systems. It can therefore be interesting to discuss to what extent distributed system technologies such as CORBA [11], Jini [3], or web services could support the ICN services and interactions shown in Figure 3. Both CORBA and Jini natively support name resolution and event notification mechanisms, which are key in the ICN architecture. However, these mechanisms do not necessarily scale to the number of IOs that are envisaged for an ICN, i.e. at least a few orders of magnitude above the number of objects accessible in the current Internet.

In addition, the ICN architecture should support highly dynamic network topologies where IOs, hosts, and networks can be mobile. CORBA and current web service technologies do not have native support for such dynamic topologies. Even though Jini is designed to handle object mobility, this support is not sufficient to handle highly dynamic network topologies with strict requirements on short handover latencies [10]. By contrast, mobility and multihoming mechanisms have been developed to handle such mobility use cases for ICN [5].

VII. Conclusions

In this paper, we argued that a key feature of ICN is the support of a variety of use cases by providing a set of services such as name resolution, reachability, storage, event notification, search, etc. These services make minimal assumptions about the applications, and can therefore be used by a multitude of application types. To describe these services and the interaction between them, there is a need for a model that is independent of the mechanisms used to implement the services. We have made a first attempt to describe such a service model. Having such a service model will make it possible to define APIs and interdomain interfaces to allow interoperation between ICN approaches with different domain-internal mechanisms, as well as providing a mechanism-independent environment for application developers. There is a possibility for a new business role which provides the ICN services in the model, i.e., an ICN Service Provider.

References

[1] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, M. Marchisio, I. Marsh, B. Ohlman, K. Pentikousis, R. Rembarz, O. Strandberg, and V. Vercellone. Design considerations for a network

of information. In *Proceedings of ReArch'08: Re-Architecting the Internet*, Madrid, Spain, Dec. 9, 2008.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A Survey of Information-Centric Networking. *IEEE Communications Magazine*, 50(7):26–36, July 2012.

[3] Apache. Jini Specification. Available online at http://river.apache.org/about.html, retrieved: August, 2012.

[4] M. Caulfield and K. Leung. Content Distribution Network Interconnection (CDNI) Core Metadata. draft-caulfield-cdni-metadata-core-00, October 2011.

[5] A. Eriksson and B. Ohlman. Scalable object-to-object communication over a dynamic global network. In *Proceedings of Future Network and MobileSummit 2010*, June 2010.

[6] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox. Information-Centric Networking: Seeing the Forest for the Trees. In *Tenth ACM workshop on Hot Topics in Networks (HotNets-X)*, Cambridge, MA, USA, Nov. 2011.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM.

[8] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander. LIPSIN: Line Speed Publish/Subscribe Internetworking. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 195–206, New York, NY, USA, 2009. ACM.

[9] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of SIGCOMM'07*, Kyoto, Japan, Aug. 27-31, 2007.

[10] J. Newmarch. *Jan Newmarch's Guide to Jini Technologies*. 2006. Available online at http://jan.newmarch.name/java/jini/tutorial/Jini.xml, retrieved: August, 2012.

[11] OMG. CORBA Specification. Available online at http://www.omg.org/spec/CORBA/index.htm, retrieved: August, 2012.

 ISBN: 978-1-61208-238-7