

# The Fluid Internet: Service-Centric Management of a Virtualized Future Internet

Steven Latré\*, *Member, IEEE*, Jeroen Famaey†, *Member, IEEE*,  
Filip De Turck†, *Senior Member, IEEE*, and Piet Demeester†, *Fellow, IEEE*

\*Department of Mathematics and Computer Science, University of Antwerp – iMinds, Belgium  
E-mail: steven.latre@uantwerpen.be

†Department of Information Technology, Ghent University – iMinds, Belgium  
E-mail: {jeroen.famaey, filip.deturck, piet.demeester}@intec.ugent.be

**Abstract**—The Internet was originally designed as a best-effort packet forwarding substrate. However, since its inception, its purpose has shifted towards a rich, service-centric delivery platform. Its underlying infrastructure, as well as the number of connected devices, have taken immense proportions, and sharing of capabilities has come into widespread use. Moreover, the Internet’s services are becoming increasingly interactive, context-aware, and content-oriented. This has led to stringent delivery requirements being imposed on the underlying infrastructure. Despite these evolutions, management of the Internet has not evolved significantly. It remains largely static and is unable to provide dynamic end-to-end service delivery guarantees in a cost effective manner. In this article, we propose the *Fluid Internet*, a novel paradigm aimed at tackling these management challenges. The Fluid Internet seamlessly provisions virtualized infrastructure capabilities, adapting the delivery substrate to the dynamic requirements of services and users, much like a fluid adapting to fit its surroundings. As such, the Fluid Internet gives a service provider the ability to manage its services end-to-end and elastically. Our vision is achieved through a unification of concepts from network virtualization, cloud computing, and service-centric networking. The relevant stakeholders, as well as their functions and interactions are described. Additionally, we identify the major technical challenges that remain to be tackled for the Fluid Internet vision to become a reality.

## I. THE INTERNET (R)EVOLUTION

The Internet has evolved far beyond its inventors wildest dreams. Originally a simple packet forwarding substrate, it has grown into a ubiquitous delivery platform for complex services. This transformation is driven by a plethora of changes and innovations in terms of the Internet’s infrastructure and services.

Recently, several *infrastructure*-related trends have emerged. A huge number of mobile and capacity constrained devices, have become part of the Internet, providing increased immersion and new levels of interactivity. Additionally, more and more everyday objects and appliances are becoming part of the Internet. This trend, collectively called the *Internet of Things*, paves the way for novel and ubiquitous services in a wide range of application domains. Finally, sharing of device, server, and networking capabilities (e.g., the *cloud computing* paradigm) has become a reality, giving users access to virtually unlimited amounts of storage and computing

power. The increasing popularity of *network virtualization* further strengthens this evolution. In the future, these trends are expected to persist, resulting in immense amounts of seamlessly connected static and mobile devices, appliances, servers, and network elements, offering their virtualized capabilities to users. In this context, the boundaries between the network and the cloud are clearly fading. Also in edge networks, computational capabilities are gradually being introduced (e.g., telco clouds such as the recently presented Radio Application Cloud Server by Nokia Siemens<sup>1</sup>).

The Internet’s transformation into a global service delivery platform has given rise to a wide spectrum of novel *services*, which are becoming increasingly interactive, context-aware and content-oriented. The explosive growth of Internet-based interactive multimedia services in the last decade clearly demonstrates this evolution. In turn, this has led to increasingly stringent delivery requirements being imposed on the underlying network. Moreover, such requirements are becoming more versatile: a broadcast multimedia service requires fundamentally different delivery guarantees than an interactive game. Finally, the requirements of a single service can also fluctuate significantly over time due to external stimuli (e.g., a user switching from an audio call to a video call). With the growing attention to Over The Top (OTT)-based services and successful examples of monetizing them (e.g., Netflix), the importance of these services and their delivery with guarantees will only become more important in the future.

These trends introduce three important management challenges, which need to be resolved for the Internet to maintain its role as complex service delivery substrate:

- 1) It must be possible to dynamically *manage and provision services end-to-end*, effectively breaking the boundaries between different management domains. Despite many attempts (e.g., SLA-based management), it is still not possible to provide complex delivery guarantees from service originator up to the customer. The reasons for this are both technical (i.e., lack of automated and dy-

<sup>1</sup>Nokia Siemens Networks Radio Application Cloud Server  
<http://www.nokiasiemensnetworks.com/portfolio/liquid-net/intelligent-broadband-management/liquid-applications>

dynamic SLA management and negotiation) and business-oriented (i.e., lack of incentive for cooperation).

- 2) Current static management approaches, overdimensioning of resources, and classification of traffic in only a few traffic classes (e.g., Diffserv's best effort vs. assured forwarding) contrast greatly with the service-centric nature of today's Internet. Management of the Future Internet must be dynamic enough to *cope with the services' versatile and fluctuating quality requirements*.
- 3) As devices are becoming smaller and more portable, Future Internet management should place *larger emphasis on mobility support*. Users must be able to roam between networks, without experiencing major impact on service quality during and after handover.

To tackle these challenges, we introduce the *Fluid Internet* paradigm, which envisions the dynamic and seamless provisioning of virtualized infrastructure capabilities, in order to continuously satisfy the ever-changing requirements of services and users, in face of environmental dynamics. The paradigm's name is inspired by the behaviour of fluids, which continuously adapt their shapes to fit their surroundings. We introduce two main differentiating points compared to the state-of-the-art. First, services are treated as first class citizens, allowing service components to be dynamically placed by decoupling them from their location. Second, the Fluid Internet facilitates the elastic provisioning of virtualized end-to-end service delivery infrastructures, in which leased (virtual) capabilities are dynamically configured and scaled as a function of user demand and service requirements. This vision goes beyond the traditional elasticity of clouds, as in addition to computational and storage resources, the managed capabilities also include network resources (e.g., bandwidth) and high-level service and device functionalities.

The need for an integrated virtualization approach encompassing computational, storage, and networking resources has recently been proposed in the context of *data center network virtualization* [1]. This novel paradigm allows customers to lease full-fledged virtual data center networks, consisting of virtual machines as well as a virtual network that interconnects them. In contrast, traditional cloud computing supports only the leasing of a set of virtual machines, without any guarantees concerning their connectivity. Although the Fluid Internet vision and data center network virtualization both strive to combine aspects from cloud computing and network virtualization, there are some key differences. First, data center virtualization is limited to a single data center network domain. In contrast, the Fluid Internet facilitates end-to-end virtual service delivery networks, federating multiple physical networks. Second, existing data center virtualization solutions are purely driven by low level network and computing requirements (e.g., bandwidth, delay, storage, CPU), while we take a service-centric approach driven by high-level service and user requirements.

## II. THE FLUID INTERNET

We envision the Fluid Internet as a unification of concepts from several networking and management principles: cloud

computing [2], network virtualization [3], and service-centric networking (SCN) [4]. With cloud-related computational resources being introduced in the network as well, these parameters should also be taken into account in management. Current network virtualization approaches focus mainly on the virtualization of pure networking capabilities such as bandwidth. We argue that network virtualization should go much further and also support the elastic allocation of storage, and computing resources as well as high-level capabilities, both in the cloud and in the network. The service-centric networking paradigm is an extension of information-centric networking (ICN) [5]. By treating not only the content (as is the case with ICN) but also the service as a first class citizen in the network, it allows service components to be dynamically placed, moved and duplicated. Only by combining all three aspects, can dynamic end-to-end quality requirements of future services be fully and continuously satisfied. The Fluid Internet combines these three aspects. Therefore, in contrast to the traditional management state-of-the-art, it allows setting up end-to-end delivery paths owned by multiple physical infrastructure providers and providing varying Quality of Service (QoS) guarantees, which are inherently linked with the service requirements.

The Fluid Internet is constructed by setting up a delivery chain, called a Virtual Service Infrastructure (VSI), that can deliver the service given the requested quality guarantees. Components of this VSI can for example be network resources (e.g., links, or router queues), network protocols (e.g., retransmission protocols), computational resources (e.g., in-network components that replicate a service), and storage resources (e.g., caches). With the term *capabilities*, we refer to all of the above components. Access to capabilities is virtualized and the service provider has complete freedom to use them in construction of a VSI. If the initial leasing of capabilities does not suffice at some moment in time (e.g., due to an increase in users or user mobility) additional capabilities can be leased or released. This can lead to a complete reconfiguration of the constructed VSI. For example, if the service provider experiences a peak of user requests in a certain region of the network, it can decide to replicate the service closer to the edge of that region by leasing additional computational and storage capabilities.

The Fluid Internet has a number of major advantages compared to the traditional Internet management architecture, giving it the potential to resolve the challenges defined in the previous section:

- 1) *Delegation of management responsibilities*: As capabilities are virtualized, the owner must not necessarily be in charge of their management and configuration. It can choose to delegate this to another party, which may provide more differentiated management functions (e.g., tailored to specific services or building a virtualized infrastructure that optimizes one particular capability). This is becoming a reality due to the popularity of the Software Defined Networking (SDN) concept and introduces the opportunity of new business cases as novel stakeholders may arise.
- 2) *End-to-end manageability*: Coupled with the ability to

remotely manage physical networks comes the opportunity to lease capabilities belonging to different administrative domains. As the Fluid Internet allows coupling them into one end-to-end VSI, end-to-end service management becomes possible.

- 3) *Dynamic management*: Similar to the scaling feature offered by today's clouds, the Fluid Internet paradigm supports elastic scaling of capabilities based on service demands.

A number of important stakeholders are involved in the Fluid Internet. This is illustrated in Figure 1. We discuss stakeholder responsibilities and interactions in the remainder of this section.

#### A. Infrastructure providers

Infrastructure Providers (IPs) own the Internet's physical infrastructure and offer virtualized access to it. This means that they open up their physical infrastructure to remote parties. They are responsible for mapping VSI requests to the physical reality *within the administrative domain they manage*. As shown in Figure 1, we call this mapping vertical embedding. Note that this role is broader than the traditional concept of *network provider* and also comprises providing access to cloud infrastructure, storage and end-user device capabilities (e.g., gateway, set-top-box).

Note that our approach does not imply that all physical infrastructure must be virtualized. There may be infrastructure, which the IP chooses not to virtualize as there is no need for management (e.g., part of the core network). Such infrastructure will then be traversed in the traditional best-effort manner. In this situation, it is only possible to provide end-to-end guarantees if the non-virtualized infrastructure has enough available capacity to accommodate all traversing traffic.

#### B. Virtual Service Infrastructure Providers

Virtual Service Infrastructure Providers (VSIPs) provide inter-domain virtualized infrastructure by coupling the virtualized infrastructure offered by different IPs into an end-to-end VSI. As this coupling deals with crossing different administrative domains, we call this horizontal embedding. Additionally, VSIPs provide management functionality in which they can optimize the end-to-end VSI towards one or more resource capabilities. For example, as illustrated in Figure 1, a VSIP may choose to offer two different end-to-end VSIs: one focusing on an optimization of bandwidth and storage, and one on the optimization of delay.

In order to deploy a service, the VSIP must determine the most optimal deployment of service components (e.g., caches, replication of services) by constructing a VSI configuration, based on the services' requirements. This mapping from requirements to a VSI is called *requirement translation* and is discussed in more detail in the next section.

Note that there can be a hierarchy of different VSIPs, where each higher layer VSIP offers additional functionality (e.g., additional optimization features, increased coupling of VSIs). This additional functionality is then provided by relying on the lower layer VSIPs. The latter thus allows recursive

virtualization. This is similar to cloud computing models such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS), where SaaS is often based on PaaS, which is in turn based on a IaaS platform.

#### C. Service Providers

Service providers (SPs) offer value-added services to users, and comprise both OTT and X-play (e.g., Triple Play, Quadruple Play) providers. Note that, as network virtualization is being adopted, there is a motivation for removing the – often tight – connections between today's infrastructure and service providers [6]. To offer a service, an SP will lease a particular VSI from the VSIP. For example, for providing an interactive videoconferencing VSI, it will lease resources from the VSIP's delay optimized VSI.

#### D. Users

Users consume services by interfacing with one or more service providers. In the Fluid Internet, SPs maintain the primary connection with the users. Users access a service from an SP, which promises to deliver it under pre-agreed delivery guarantees. How this delivery is performed (i.e., through which physical infrastructure) is of less importance to the user. As such, users will no longer need to care about the indirect factors that influence their Quality of Experience (QoE), instead focusing directly on their QoE expectations.

The user should have the *perception* that he/she has its own VSI when accessing a service. This means that this infrastructure is able to elastically scale and adapt to its needs caused by factors such as mobility or changes in the consumption pattern of the service. Obviously, maintaining a single VSI for each user does not scale. Hence, in practice similar VSIs are grouped: first by the SP who maps one service type into a particular VSI, and afterwards by the VSIP who again groups several SP VSIs to a VSIP VSI with similar QoS requirements.

#### E. Multi-role stakeholders

Although we presented different roles, this does not mean that each role corresponds with a separate entity. As illustrated in Figure 2, it is possible that one company fulfils multiple roles. For example, traditional ISPs now already fulfil the role of IP, SP and, in some cases, VSIP by offering limited shared access. Similarly, a large SP (e.g., the current Google model) may choose to perform the management of VSIs by leasing directly from an IP. Finally, large infrastructure providers (e.g., the current Akamai model) may choose to virtualize their own infrastructure and map it to different VSIs.

### III. SERVICE DELIVERY IN THE FLUID INTERNET

Figure 1 depicts the plethora of interactions that take place between the stakeholders of the Fluid Internet. This section further elaborates on the details of these interactions. They are split into two major steps. First, the dimensioning, planning, and provisioning of a virtual service infrastructure is discussed. Second, we focus on the more dynamic elastic management concepts.

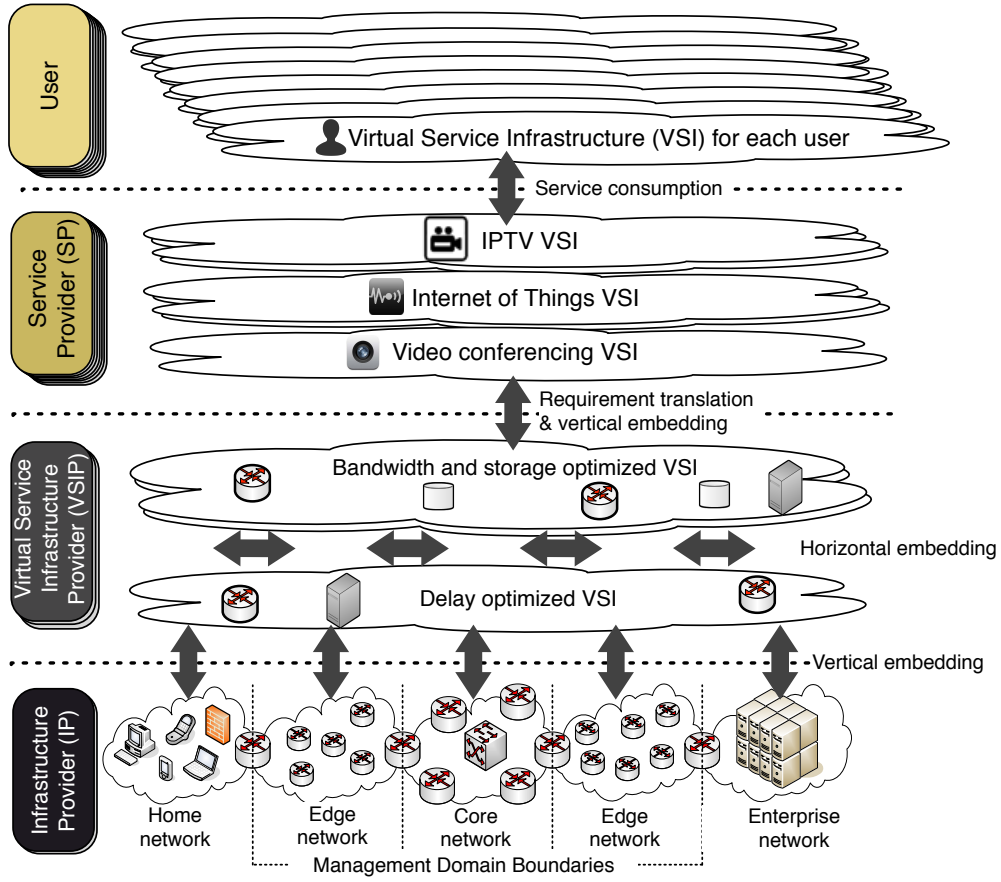


Fig. 1. Architectural overview of the Fluid Internet introducing different stakeholders and their interactions

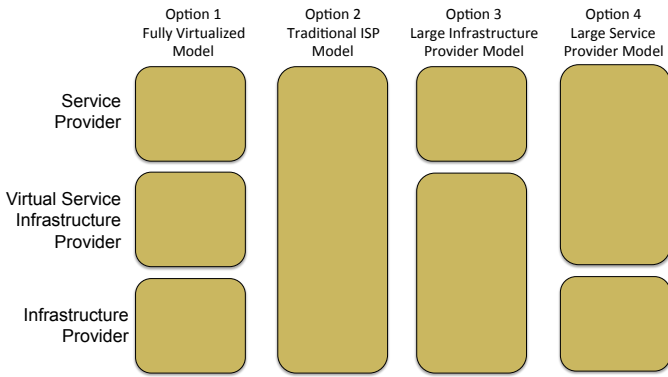


Fig. 2. Overview of models defining how the different roles can be mapped to different entities.

### A. Leasing the Internet

Figure 3 depicts the interactions and other steps involved in dimensioning, setting up, and configuring a fluid virtual service infrastructure. Every VSIP pre-configures one or more end-to-end VSIs, based on different types of quality guarantees. For example, it can operate a VSI optimized for low end-to-end delay services on one hand, or delay-tolerant, high-bandwidth applications on the other. Each VSI is configured for a specific capacity, which is expected to suffice for its predicted near-future demand.

As depicted in the figure, at some point, an SP may request

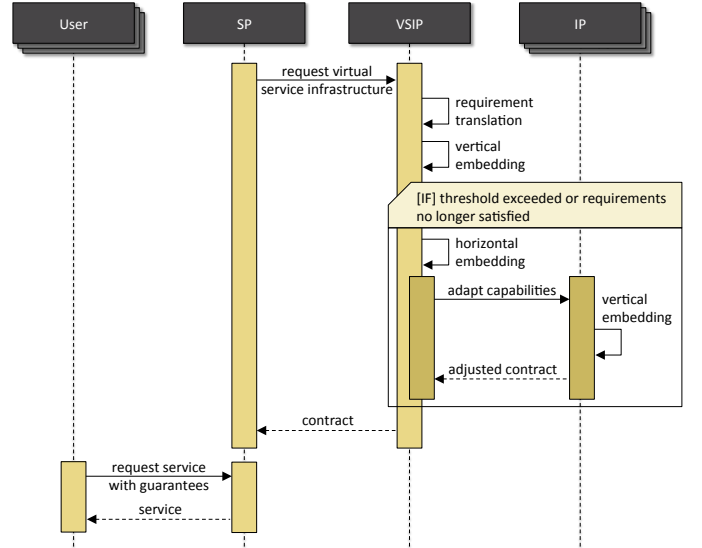


Fig. 3. Sequence diagram detailing the steps and stakeholder interactions involved in setting up a virtual service delivery network

an end-to-end VSI for delivering its services under specific guarantees. Based on this request, the VSIP subsequently performs two or more actions. First, it translates the received end-to-end service requirements into a VSI configuration. Second, it executes a *virtual network embedding* (VNE) step.

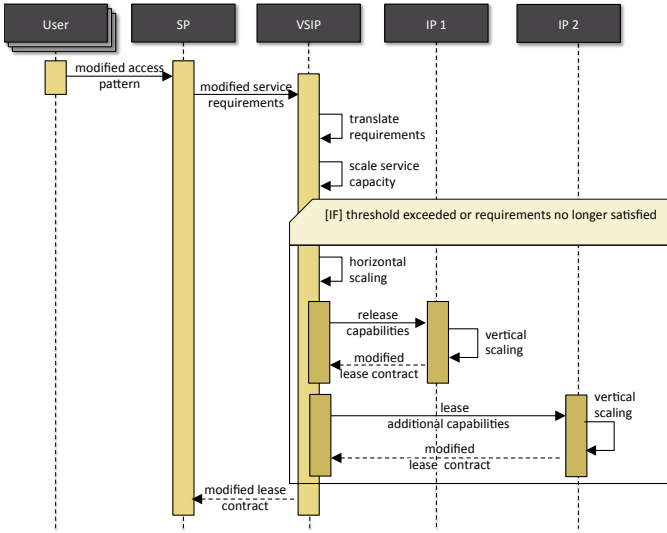


Fig. 4. Illustrative sequence diagram detailing the steps and stakeholder interactions involved

We refer to this specific step as *vertical VNE*, as it is concerned with assigning the VSI configuration requested by the SP to the virtual resources in one of the VSIP's VSIs. This is a complex process that also involves previously embedded VSIs of other services (likely from multiple SPs). As such, the VSIP should calculate the optimal mapping of all SP VSIs to its own VSIs. Embedding the newly requested VSI configuration may fail, due to a lack of available resources (i.e., the resource capacity threshold is exceeded) or the fact that the VSIP does not operate a VSI that can satisfy the requested service guarantees.

When such a failure occurs, several other actions need to be performed. Based on the changed resource capacities or service requirements, the VSIP will adjust one or more of its own VSIs. Subsequently, it performs a *horizontal VNE* step. It is referred to as horizontal, as it is concerned with embedding the VSI across multiple physical IPs. Concretely, the horizontal VNE problem consists of partitioning the end-to-end VSIs in multiple intra-domain VSI components. Subsequently, the VSIP maps all intra-domain VSI parts of all its VSIs unto compatible physical infrastructures.

Subsequently, the VSIP sends requests to the relevant IPs to adjust its leased capabilities, in line with the results of the horizontal VNE step. Each IP then solves the vertical VNE problem for all intra-domain VSI components that are provisioned on its infrastructure, mapping virtual capabilities unto concrete physical resources. This is similar to the vertical VNE performed by the VSIP, except that it maps virtual infrastructures unto a physical one, instead of virtual unto virtual.

Finally, the lease contracts are forwarded to the VSIP and SP, which can then start delivering services to its users under the pre-requested service guarantees.

### B. Elastic Management

Dynamic management in the Fluid Internet goes beyond traditional cloud-based elastic management, both in scale and capabilities. The SP is expected to reserve spare capacity in

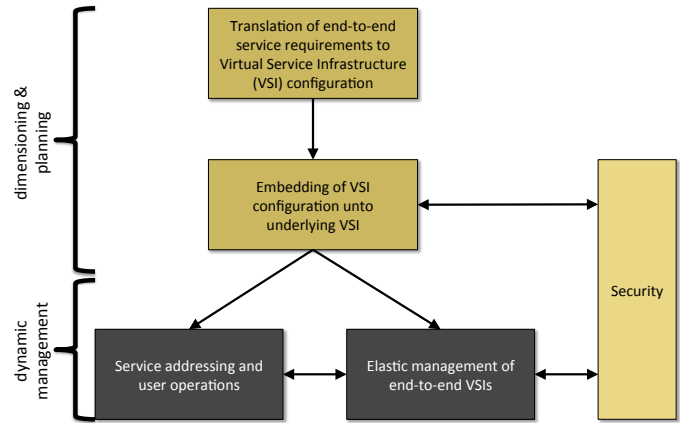


Fig. 5. Overview of the major technical challenges that remain to be tackled before the envisioned service-centric virtualized Future Internet becomes a reality

its VSIs to be able to cope with minor fluctuations in user behaviour and demand, caused by service popularity changes, user mobility, or user quality requirement variations. However, providing too much spare capacity would result in inefficient resource utilization and unnecessarily high costs. As such, major deviations in user access patterns, might cause the SP's VSIs to become inadequate in offering users the expected service guarantees.

Figure 4 presents such an example. The SP continuously evaluates the deployed services' requirements in order to assess if its VSIs still satisfy them. In the example, the SP detects a modified access pattern, and decides that more capabilities are needed to continue supporting the service adequately. It forwards the modified service requirements to the relevant VSIP, which translates them into a modified VSI configuration. Much like the SP, the VSIP is expected to provide some spare capacity in anticipation of minor service requirement fluctuations. If the spare capacity is sufficient, the VSIP can merely scale the capacity allocated to the SPs VSI. If not, or if the modified requirements cannot be supported by the VSIPs current set of VSIs, several other actions need to be taken.

First, the VSIP performs the *horizontal scaling* step. It is similar to the horizontal VNE step it performs when provisioning new services. However, as changes in requirements are expected to occur on much shorter time scales than the provisioning of new services, the problem is solved in an online fashion. Consequently, this step requires fast heuristics that iteratively adapt existing solutions rather than computationally intensive VNE algorithms that calculate the optimal solution from scratch. Based on the modified VSIs resulting from the horizontal scaling step, the VSIP releases, scales, and/or leases capabilities from different IPs. In turn, those IPs perform vertical scaling operations, which are again a faster, online version of the vertical VNE step. In the example, the VSIP releases its leased capabilities from IP 1, while leasing new capabilities from IP 2.

#### IV. THE FUTURE OF THE FLUID INTERNET

The way towards the Fluid Internet is riddled with many remaining challenges as the current state-of-the-art in research is unable to tackle the scientific problems as outlined above. This section presents the most important open issues, the current advances in the state-of-the-art and the progress beyond that needs to be achieved. Figure 5 outlines these challenges and their relationships.

##### A. Requirement translation

State-of-the-art algorithmic research in network virtualization has focused solely on the network embedding problem. However, in order to embed a VSI configuration onto the underlying infrastructure, this configuration should first be constructed. We argue that the creation of a VSI configuration, based on a set of QoS requirements and user locations, is a non-trivial problem. Specifically, there is a need for algorithms that are capable of translating a wide range of QoS requirements into a set of virtual capabilities (e.g., storage, computing, link capacities, router queue configurations, protocol parameters) needed to satisfy them. Moreover, the algorithms should be able to connect those capabilities in an end-to-end VSI, and incorporate the dependencies between services, user access patterns, and their associated content and other data. Finally, there is a strong synergy between the translation problem on one hand and the network embedding problem on the other. The translation step could result in several alternative configuration solutions, of which some may prove infeasible in the embedding step. As such, this interaction needs to be taken into account in the design process of both algorithm types.

The translation of QoS requirements into VSI configurations is related to the more general problem of automated software requirements refinement, a sub-area of requirements engineering [7]. Requirements refinement is concerned with translating high-level customer business requirements into concrete software component specifications. Although some progress has been made in this area, existing solutions are limited to semi-automated tools for assisting humans in performing refinement tasks, rather than fully automated algorithms. In the area of network and service management, requirements refinement principles have been applied to the policy refinement [8] problem, which aims to translate high-level business policies into low-level device configurations. In line with requirements refinement, no fully automated approach has been proposed to date. However, Fluid Internet-based requirement translation has a potential for success for two reasons. First, the translation is less complicated than pure policy refinement as it only needs a translation to a VSI configuration, which can be seen as an intermediate language in the continuum of policy languages. Second, with the increased popularity and adoption of SDN, there is an increased attention towards the design of network programming languages [9]. Research towards network programming languages also include formal verification methods, which are key in proving the feasibility and accuracy of a requirement translation.

##### B. Service-centric network embedding

Network embedding has been a much researched topic in recent years. Nevertheless, a recent survey by Fischer *et al.* [10] has shown that state-of-the-art algorithms focus mainly on solving the problem in a centralized way. Obviously, centralized algorithms do not scale well to a huge network of networks, such as the Internet. As such the designed algorithms need to calculate an embedding solution for a huge number of co-existing VSIs, based on partial and incomplete knowledge in a highly distributed fashion.

Moreover, the study showed that very few algorithms can cope with the intricate complexities of federated virtual networks, spanning across multiple infrastructure providers. However, to be able achieve the discussed horizontal embedding, such a federated approach is necessary. The algorithms should be able to cope with inter-provider difficulties, including operator selfishness, trust issues, and pricing model differences.

The network embedding problem has, in the past, been tackled from a pure network virtualization perspective. However, the Fluid Internet approach proposed in this article, warrants the need for including computing-, service- and content-related aspects as well.

As such, there is a need for distributed network embedding algorithms that operate on partial knowledge and scale to immense amounts of capabilities and co-existing VSIs. They should incorporate support for inter-provider VSIs, for example by using game theoretical principles to model the intricate relationships between infrastructure providers. Moreover, they should transcend traditional embedding algorithms that are purely focused on network capabilities and also incorporate embedding of storage and computing resources.

##### C. Elastic management

In order to support elastic scaling of deployed VSIs, in accordance with service and network dynamics, both requirement translation, as well as virtual network embedding need to be able to adapt calculated solutions. Fully recalculating and re-embedding end-to-end VSIs whenever a change occurs in service requirements or infrastructure conditions is obviously infeasible. As such, algorithms are needed that are capable of incrementally fine-tuning existing configurations. They should focus on quick tweaking, in order to ensure the continuous satisfaction of quality guarantees, which could result in sub-optimal configurations in terms of for example deployment costs or energy efficiency. In order to alleviate this, an integrated approach is necessary, combining computationally-intensive optimal algorithms with incremental fast-responding heuristics.

Fischer *et al.* [10] stated that, in addition to being centralized, state-of-the-art virtual network embedding algorithms focus mostly on static embedding. Moreover, the problem space that combines distributed and dynamic embedding remains virtually unexplored. This introduces a gap in state-of-the-art research, as well as an opportunity for novel heuristics that can quickly adapt existing configurations in a fully distributed manner. The requirement translation problem remains, even in the static case, an open issue. Consequently, there is a need for

dynamic heuristics, capable of adapting existing VSIs based on changes in requirements. Note that the above approach of elastic management, very much complements current cloud management approaches. Similar to a cloud environment, the users are given an elastic network that can grow and shrink with its resource requirements.

#### D. Service addressing and protocol signalling

The Fluid Internet introduces an important paradigm shift to the current addressing and delivery of services: service components can be replicated, fragmented and dynamically migrated to support delivery requirements. Instead of identifying a service by location, users should be able to send out a location agnostic request, coupled with service delivery requirements. The above means that the service and not the location should be the primary entity in the Internet. It should be noted that the decoupling of services and content from their location is already happening to some extent in the current Internet. For example, CDNs dynamically move content around the Internet, based on request patterns and user requirements, and cloud computing supports the dynamic migration of services. However, such solutions are provider-specific, limiting the migration to locations within the same CDN or cloud domain.

The ICN [5] paradigm also aims to move away from traditional host-centric addressing on the Internet. However, we argue that ICN alone is not sufficient to grasp the intrinsic relationships present between the roles. Although content is important, not all services are focused on delivering content, but might instead serve to process data or perform other types of calculations. Therefore, the ICN concepts should be generalized to serve as a basis for a service-centric Internet. As the Fluid Internet paradigm can reuse ICN concepts (e.g., the removal of host-centric communication) it has ICN-related challenges [11] but also adds considerably different ones. For example, location agnostic requests and their reply must also take into account the VSI and the services' requirements. Additionally, in delivering and routing a service, algorithms must take into account the volatility of services in terms of bandwidth.

Moreover, to allow the leasing of capabilities between IPs, SPs and VSIPs, communication protocols between the stakeholders are needed. Current SDN-inspired network programming language approaches [9] form a good basis for such communication protocols as they allow remote manageability of networking devices. However, they are too focused on networking programmability and should be extended to support configuration of service-centric concepts, which are key in the Fluid Internet paradigm.

#### E. Security

Although the Fluid Internet introduces important new opportunities, it also entails some non-negligible security concerns. Remote parties are given access to the management of local network and computational capabilities, which can lead to security issues such as data breaches, non-isolation of configuration management between different VSIs, and authentication and authorization challenges. Many of those security risks

stem from the fact that capabilities are virtualized and shared access to the same physical infrastructure is provided. Similar security challenges are faced by cloud computing platforms. Subashini *et al.* [12] present a survey of security issues in cloud computing in which the aforementioned concerns are described for a cloud platform. Moreover, the survey provides an overview of current security solutions in cloud environments, discussing the need for developing security solutions and standards for the cloud. Besides a set of best practices for information assurance in the cloud, they also argue that an integrated security model targeting different levels of security is required and still under research. The recent move towards hybrid and federated clouds further complicate the problem. Centralized solutions no longer suffice, due to scaling limitations and lack of centralized ownership. For the same reasons, a Fluid Internet should target a distributed security model, which incorporates both cloud-originated (i.e., data-centric security) as network-originated security concerns (i.e., access-centric security).

## V. CONCLUSION

In this article, we present a disruptive but evolutionary view on the future of network and service management. Given today's evolutions, management should focus on providing end-to-end quality guarantees in a highly dynamic environment with mobile users and versatile service requirements. We argue that a holistic management approach is necessary, combining aspects of cloud computing, network virtualization, and service-centric principles. We call this the Fluid Internet in which network-, service- and cloud-centric resources are freely assigned to leased infrastructure and can elastically scale in the network according to the users' needs. A key advantage of the Fluid Internet is that it provides the ability to dynamically manage services end-to-end. We argue that network virtualization, which currently receives a lot of attention, should not be limited to the virtualization of pure network resources but also support the virtualization of computational and storage resources. We presented the main interactions between the stakeholders by illustrating how a Fluid Internet can be constructed and dynamically managed. Moreover, we identified the main open research challenges to be tackled; including the translation of service delivery requirements to virtual network configurations, service-centric network embedding, elastic management of virtualized service infrastructures, addressing and signalling, and security. Through elastic management, a virtualized network can be achieved that scales with the users' and applications' demands, similar to scaling features in clouds, but which goes beyond cloud-based elastic management both in scale and capabilities.

## ACKNOWLEDGMENT

This work was partly funded by FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

## REFERENCES

- [1] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, 2013.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50–58, 2010.
- [3] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: State of the art and research challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, 2009.
- [4] Q. Duan, Y. Yan, and A. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Transactions on Network and Service Management*, vol. 9, no. 4, pp. 373–392, 2012.
- [5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.
- [6] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 61–64, 2007.
- [7] A. van Lamsweerde, "Requirements Engineering: From system goals to UML models to software specifications," Wiley, 2009.
- [8] A. K. Bandara, E. C. Lupu, A. Russo, N. Dulay, M. Sloman, P. Flegkas, M. Charalambides, and G. Pavlou, "Policy refinement for IP differentiated services Quality of Service management," *IEEE Transactions on Network and Service Management*, vol. 3, no. 2, pp. 2–13, 2006.
- [9] N. Foster, A. Guha, M. Reitblatt, A. Story, M. Freedman, N. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, D. Walker, and R. Harrison, "Languages for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 128–134, 2013.
- [10] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, 2013.
- [11] M. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–53, 2012.
- [12] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

## BIOGRAPHIES

**Steven Latré** is an assistant professor at the University of Antwerp, Belgium and the Future Internet Department at iMinds. He received a Master of Science degree in computer science from Ghent University, Belgium and a Ph.D. in Computer Science Engineering from the same university. His research activity focuses on autonomous management and control of both networking and computing applications. His recent work has focused on Quality of Experience optimization and management, distributed control and network virtualization.

**Jeroen Famaey** is currently a post-doctoral researcher at Ghent University, Belgium and is affiliated with the Future Internet department of iMinds. He obtained a masters degree in computer science from Ghent University in June 2007 and a Ph.D. degree in computer science engineering from the same university in June 2012. His main research interests include autonomic management of network federations, virtualized networks, and multimedia services.

**Filip De Turck** is a professor and leads the network and service management research group at the Department of Information Technology of Ghent University, Belgium and is a member of the Future Internet Department of the iMinds research center, Flanders. He is a senior IEEE member and

author or co-author of more than 350 refereed papers published in international journals or in the proceedings of international conferences in the area of network and service management and design of new communication services.

**Piet Demeester** is professor in the faculty of Engineering at Ghent University. He is head of the research group "Internet Based Communication Networks and Service" (IBCN) that is part of the Department of Information Technology (INTEC) of Ghent University. He is also leading the Future Internet (Networks, Media and Service) Department of iMinds. He is Fellow of the IEEE.