Global Identity and Reachability Framework for Interoperable P2P Communication Services

Ibrahim Tariq Javed, Rebecca Copeland, Noel Crespi
Institut Mines-Telecom, Telecom Sud-Paris
Evry, France
ibrahim_tariq.javed|Noel.crispi@telecom-sudparis.eu

Felix Beierle, Sebastian Göndör, Axel Küpper
Telekom Innovation Laboratories,
Berlin Germany
beierle|sebastian.goendoer|axel.kuepper@tu-berlin.de

Marc Emmelmann, Ancuta Corici
NGNI, Fraunhofer FOKUS,
Berlin, Germany

Kevin Corre, Jean-Michel Crom
Orange Labs Products & Services
Cesson-Sévigné, France

Ahmed Bouabdallah
Institut Mines-Telecom, Telecom Bretagne
Cesson Sévigné, France

Frank Oberle, Ingo Friese
Telekom Innovation Laboratories
Berlin, Germany

Ana Caldeira, Gil Dias, Ricardo Chaves, Nuno Santo
INESC-ID, IST, Universidade de Lisboa
Lisboa Portugal,

***Abstract*—Advancements in web real time technologies are revolutionizing the way communication is taking place, where new emerging web services are confronting the traditional Telco operated communication services. This has led to the development of new web-centric service architecture that will allow service providers to provide their communication services globally. This architecture uses web based communication techniques that enables endpoints to become a mesh of live users, who can communicate in a peer-to-peer fashion. To distinguish such a service from current 'best effort Internet', it must be supported by appropriate governance, trust and security features. This distributed architecture presents new issues regarding authentication of non-service-bound user identities, finding users reachable addresses and providing a trustworthy communication environment. To provide these support services this paper introduces four different components namely: Identity Management, Policy & Governance, Graph connector and Directory services.**

Keywords—Identity management, Trust, Policy, Governance, Directory, Social graph, Registry, WebRTC, P2P communication

I. INTRODUCTION

Real time communication services deployed by Over-the-Top (OTTs) web companies are gaining momentum especially after the introduction of WebRTC standard [1] WebRTC technology enables real time communication capabilities between browsers for audio, video and data exchanges. This technology has brought significant advantages to the development of innovative services over the web due to the fact that it removes the need for standardized signaling and allows developers to choose any signaling method for peer to peer (P2P) communication [2].

In consequence, traditional telecommunication operators must face their current limitations that prevent them from competing with new communication services over the web, such as chat and social networks. Previously, Telecom service providers had complete control over the whole environment including user devices, service developers and network providers, but now they need to revise the key aspects of their operational structure, including signaling, identity management and QoS management. In [3] limitations of existing IP Multimedia subsystem (IMS) are addressed while enhancing existing telco infrastructure by providing a new service oriented control plain over the web using the underlying WebRTC technology. The major challenges are identified in [4] for the development and deployment of the new web based Telco architecture. The first is the development of communication framework for deploying real time communication services. The second challenge is concerned with identity management (IdM) techniques providing secure, trusted and privacy enabled identities whereas the third challenge is to provide end to end network QoS beyond best-effort.

The IdM challenge initiates the development of new, reliable and efficient ways to handle user identities in the new web based architecture. The main objective is to develop and deploy a trustworthy identity framework that is globally searchable; cross domain interoperable (search and contact user identified in other services); and portable (users able to easily migrate between service providers). To achieve this, unlike the traditional communication in Telco's, identity management is decoupled from the provided service, and third party identity provider (IdP) are used to independently provide, manage and verify user identities.

The European founded reTHINK project describes a new communication framework providing solutions to manage real time communication capabilities over the open internet while addressing the aforementioned challenges. This framework will allow Telco operators to compete with large OTT web companies by de-perimetrising their communication services. Moreover, interoperability between different service providers will be provided, unlike most web communication services. reTHINK not only focuses on the scientific value of the proposed concept for telecommunication service architecture but also aims to provide demonstrations and a technical proof of concept. For these reasons reTHINK pushes all the proposed novelties to the applicability area by providing software prototypes that can be used for validation and demonstration purposes. The main aim of reTHINK framework is to provide existing telecom operators and OTT players the incentive of free, open, global and interoperable communication flows over the web. reTHINK framework is non telecom centric but web centric architecture which does not require any formal legislation and regulations to operate globally as compared to traditional telecom frameworks.

This new framework proposed by reTHINK handles the governance, security, trust and identity management aspects of the P2P communication using four major components which are Identity Management, Policy and Governance, Graph connector and Directory services. These components are responsible for providing support services, such as user authentication, user's reachability and addressing, trust evaluation among entities and policies implementation. To manage global reachability and discovery these components make use of two new unique identifiers for users.

The rest of the paper is organized as follows: Section II gives a brief introduction of the new framework. The identifiers and components used in reTHINK framework with a call flow scenario are shown in section III. The entities and processes involved with user identities and trust management is explained in section III. Section V provides description of three of the major directory services involved in this framework namely catalogue, registry & discovery. Section VI highlights the use of graph connectors in linking known communication end points while section VII describes the Governance & Policy and VIII gives the conclusions.

II. RETHINK FRAMEWORK

The reTHINK project describes a new web based framework to provide real time communication over the Internet, in a secured way with enhanced QoS. The framework incorporates two software concepts: the Protocol-on-the-fly (ProtoFly) and the Hyperlinked Entity (Hyperty). The ProtoFly, as introduced in [5], allows protocol interoperability among distributed services. Implementation of ProtoFly involves a 'Protocol Stub', which is the protocol stack e.g. JavaScript file that can be dynamically loaded and used to support interoperability between distributed services. The reTHINK framework uses the ProtoFly concept to resolve the issue of inter-domain interoperability without creating new standard network protocols. The second concept which reTHINK framework relies on is the Hyperty. The Hyperty is a new concept that can be described as a module of software logic that is dynamically deployed in web runtime environments on end user devices, to execute session control and media flow management in a P2P manner. Hyperty represents a User in end-user device (Web Browser, Smartphone native app etc.) or in the network (Application Server). Hyperty can be securely associated either to a Human Identity, for Human to Human Communication use cases, or to an Identity of a physical object, for Machine to Machine Communication use cases. The Hyperty software module is maintained by the Communication Service Provider (CSP) and is downloaded to the user device where it represents the user for communication as shown in Figure 1. The CSP's catalogue service helps the user in selecting and downloading the appropriate Hyperty type. A running Hyperty is the Hyperty instance that operates in the runtime environments of user device. It represents a 'live' user who is available for incoming communications from the particular CSP. Each service provider's domain retains knowledge of its 'live' Hyperties and enables connectivity with other domain's running Hyperties. These Hyperties are terminated when the service is disconnected and new Hyperty instances are loaded whenever the service is initiated, or when necessary to update the software. Since the user is able to use services of multiple CSP's, multiple concurrent hyperties can be running in user device, as shown in Figure 1. The user is connected to all the CSPs and is available for incoming calls using the services of any one of them. If the user decides to terminate the services of any CSP, the corresponding CSP Hyperty instance is terminated.

The focus of this paper is to provide an overview of the support services encompassing the governance, identity management, global reachability and discovery aspects of the reTHINK framework while detailed description, implementation and utilization of the Hyperty concept are out of the scope of the paper. The support functions are provided by four components, Identity &Trust management; Directories; Governance & policy; and Graph connector. These components are implemented using various modules in reTHINK architecture. An overview of the functional modules involved in the support services are provided in Figure 2. As shown, the support functions reside in both the device base 'runtime environment' and in the CSP domain, but they

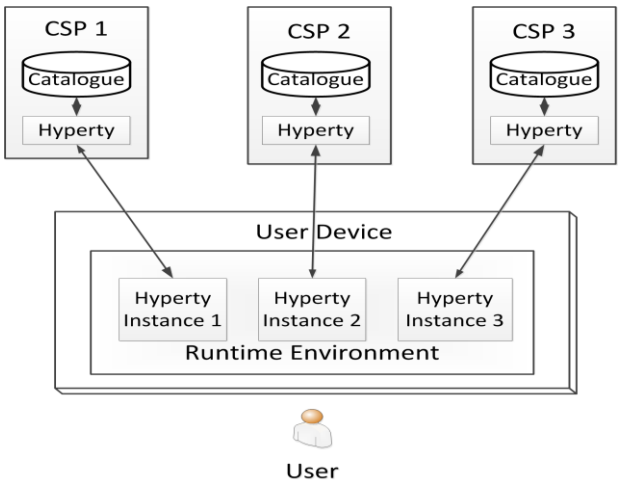

Figure 1: Hyperty Concept

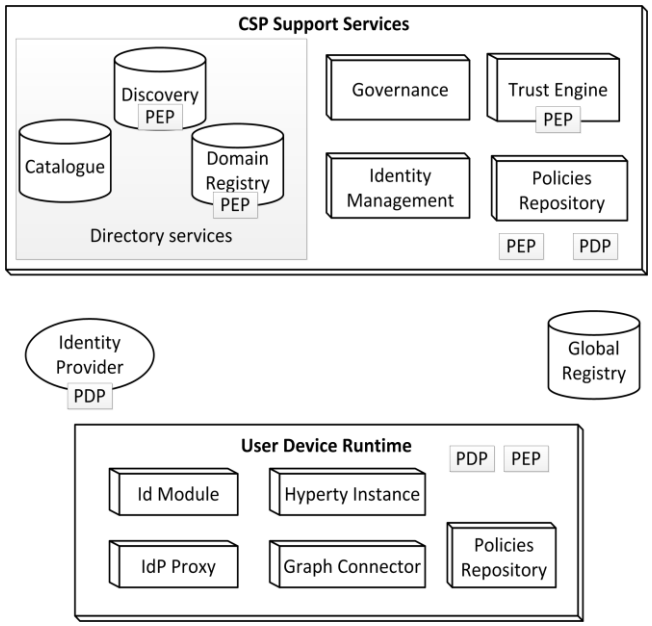

Figure 2: reTHINK funtional architecture

interface to external entities, such as Independent Identity Provider (IdP) and Global Registry.

The Identity management component is responsible for processes related to the lifecycle of user identities such as user registration, identity provisioning, user identifiers, identity authentication & verification. The identity management functions at the CSP level, which interacts with the IdP via two specialized modules (IdP proxy and IdModule inside the user device) is further explained in section IV. The Hyperty directory component include three types of directories: Registry for information about available Hyperty instances for communication, Catalogue for list of available service functions provided by Hyperty of various CSP's, and Discovery for finding users in various domains. The registry is further distributed in to global and domain registries. The domain registry contains the list of 'live' Hyperties, while the global registry converts user identifiers to their domain-based URLs. The governance component is required to manage rules and policies, obeying several CSPs, but coordinating them with the user's device environment. The separation of Policy Decision Point (PDP) and Policy Enforcement Point (PEP) is interpreted for the distributed architecture of reTHINK, where the positioning of these functions is different from the traditional one, as shown in Figure 2. Lastly the Graph connector component is described as a local address book implemented in user device runtime to manage list of known communication endpoints. The description and purpose of each of these four components and the functional elements involved is described in each of the following sections of this paper.

reTHINK goes beyond the concepts provided in this paper and will obtain a working platform capable of demonstrating the commercial and technological capabilities of this framework. The next step in reTHINK project is to implement the integrate the modules related to the support services described in this paper. The final target is to create a distributed, federated testbed integrating all the required components into a real computing environment and then validate this testbed by providing a first prototype to service providers suitable for being used in pre-commercial and exploitation preparation activities by the service providers. The vision of reTHINK project is to allow traditional Telco's to design, implement and operate innovative services overcoming the current challenges that restrict Telco's to compete with current OTT services.

III. RETHINK IDENTIFIERS & DISCOVERY

A. *Identity Management Essential Characteristics*

Identities in the reTHINK framework must be independent from any CSP, and are maintained by trusted IdPs. This certainly introduces some reliability risks as the framework depends upon the efficiency of a third party IdP which is not the case in traditional telecommunication architectures. Interoperability between various IdPs and between various CSPs is essential. The reTHINK framework allows two or more end users to communicate with each other using services from different CSPs, even if they are identified using different IdPs. Another essential feature is the portability, which allows users to switch between CSPs without losing their identity or contacts. The third feature is the global discovery of users, which enables callers to search and locate destination users globally, and not only within their own domain. These features are facilitated by the use of two main identifiers, namely Global User ID (GUID) and UserID. The GUID is a unique and domain independent identifier that remains the same, irrespective of the CSP. However the UserID is a domain specific identifier which is used to locate the actual location of user/device, by discovering its Hyperty instance within CSP domain.

B. *Identity Functions Logic and Call Flow*

To explain how the components and identifiers operate in reTHINK framework work, a simple example of Alice and Bob call scenario is presented in Figure 3. Alice wants to call Bob, so she decides to log on to CSP A services. Bob is already using services of CSP B. The steps involved in the process are explained as follows:

- Step 1-3: When Alice attempts to log on for the first time CSP A needs to authenticate and confirm the identity of Alice. CSP A requests Alice's runtime software to provide her own IdP's authentication (Identity assertion). The IdP proxy module in Alice device provides an interface between her device and her IdP for an IdAssertion (identity assertion) generation or verification. The IdP checks the Alice's credentials and after the authentication process, it provides Alice's runtime with her IdAssertion.
- Step 4-5: The IdModule inside Alice device manages and stores IdAssertions for Alice, provided by her IdP. The IdModule then sends the IdAssertion to CSP A, which in turn, gets it verified by Alice's IdP. This allows Alice runtime to use CSP A's services.

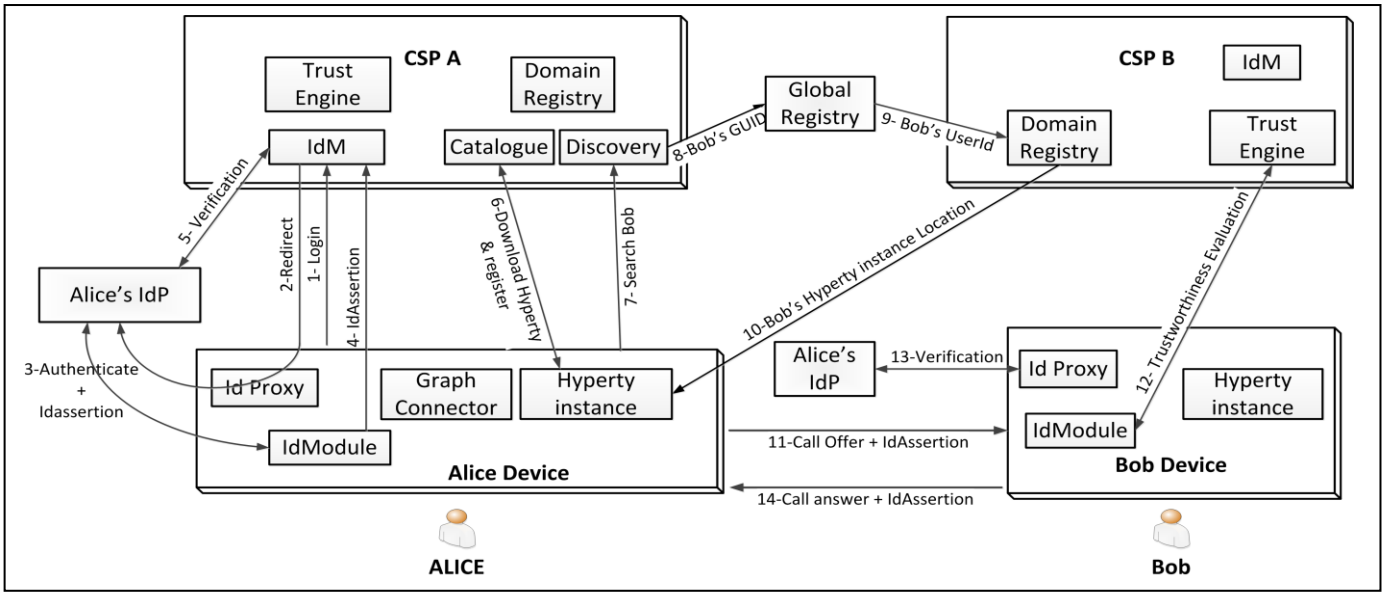

Figure 3: reTHINK call flow sequence

- Step 6: Using the catalogue service of CSP A, the type of Hyperty (by device OS or application) is selected and downloaded into the Alice's runtime, enabling P2P communication.
- Step 7-10: Alice then uses Discovery and Registry services to locate Bob who is already registered to CSP B using his own IdP. If Alice knows Bob and has contacted Bob previously, she can use the local address book with the Graph Connector module to locate Bob. Otherwise Alice uses discovery service of CSP A to search for Bob, using information she already knows about Bob, such as name, place of work etc. Using this information, the discovery service provides the GUID of Bob. As the GUID is domain independent identifier, it does not show which CSP Bob is currently using. To locate the current Bob's CSP, the global registry component resolves the GUID into Bob's UserID, which is a domain dependent identifier that shows that Bob is currently connected to CSP B. To find the actual location (IP address) of Bob's running Hyperty, the domain registry component at the CSP B platform resolves Bob's UserID to the actual location of Bob's running Hyperty, which is sent to Alice's runtime Hyperty Instance.
- Step 11-14: Now Alice's Hyperty A knows the actual location of Bob's Hyperty B, so her Hyperty initiates a call request to the Bob Hyperty instance. To provide her identity, Alice's Hyperty also attaches her IdAssertion, as stored inside the IdModule. Before accepting or rejecting the call, Bob checks the identity of Alice by verifying the IdAssertion with Alice's IdP. After the verification process, Bob checks the trustworthiness of Alice, using the trust engine service. Depending upon the level of trust and the authentication, Bob can accept or reject the call. If Bob accepts the call, a call answer request with his IdAssertion is sent back to Alice. Depending on the level of trust, Alice's Hyperty may verify Bob's IdAssertion before connecting the call.

IV. IDENTITY AND TRUST MANAGEMENT

Trust is an important part of Human to Human relations when faced with uncertainty. This is especially true on distant virtual communications such as on the Internet as the uncertainty factor is much higher. In this case the affiliation of each party to a single trust domain or a circle of trust usually supports trust. But in a typical reTHINK scenario involving users from different domains (i.e. not part of a circle of trust) trust between correspondents must be supported by some other means. The very first step to establish trust between parties is usually asserting their identities. Along this reTHINK also relies on external recommendation sources to estimate the trustworthiness of a caller. Thus the reTHINK framework involves two components for identity and trust management cooperating to provide trustful call services. The identity management component is involved with the identification, authentication and verification of users using identity related security tokens while trust management component is concerned with estimating the trustworthiness of the user.

A. Identity and Authentication

Identity management relies upon managing lifecycle of identity-related security tokens. The Identity assertions can come from various sources and may use different protocols and formats. These identity assertions are provided by IdP which can be on the network or possibly on the device being used. Various identity assertion standards have gain popularity over the internet but the main concern when it comes to communication services is the audience data which refers to other user's involved in the call. For this the reTHINK identifiers can be used. Indeed GUID usage is the zero level, commonly named "self-asserted" identity that would need to be upgraded in most cases (but not when anonymous calls are considered).

In WebRTC, identity assertions are generated and verified by the IdP Proxy mechanism. The IdP Proxy allows dynamic

IdP discovery independently of the CSPs. In this model, an Identity Assertion (implemented for instance by a JSON Web Token [6] like in OpenID Connect [7]) is joined to the call offer or answer. As the Identity Assertion contains verification endpoint URL (used to retrieve some code capable of verifying the assertion, on the IdP Proxy), it is possible to verify a user's claimed identity without knowing beforehand the IdP for that user.

However, since the browser does not (yet) knows its user's identities, it is thus not capable of verifying properties such as the intended audience of an Identity Assertion (i.e. the party which this assertion has been issued for). This may cause difficulties or even security breaches when implementing IdP Proxies for standard protocols such as OpenID Connect [8]. reTHINK solves this issue by using the IdModule component in its architecture. In its most simple form the IdModule is storage for user's identities and a platform for IdP Proxy execution. In a standard WebRTC fashion these IdP Proxies can generate Identity Assertion or verify a received one. Functionalities are provided over the network as each IdP Proxy is capable of exchanging messages with an endpoint on its IdP.

This can also be enriched by local functionalities: the IdP Proxy could act as a local IdP running on the device, able to deliver trusted identity tokens to installed native applications, which improves user experience with single sign on, security and privacy enhancement. Such a solution has already been designed by the Liberty Alliance [9] and an OpenID Connect-based open source implementation has been published [10] as a Trusted Identity Module (TIM) [11]. Currently TIM is a smartphone module (Android and JavaCard) that operates as an OpenID Connect server that requests from native applications needing to access user personal data. The TIM works in offline and online modes and provides many benefits among with:

- Usage continuity when in offline scenario (or in a roaming situation)
- Privacy improvement for the end-user as the online IdP is not contacted and therefore unable to track the user's activity
- Improved security with the use of a combination of Trusted Execution and Secure storage.

B. Trust management

The reTHINK architecture involves a new component called trust engine that can be seen as a service providing information about trust in a communication partner. Trust is a rather complex term and might consist of many different quantities and vectors. The different vectors could be combined or grouped to certain levels. But the decision of who is trusted or not is subjective and should be up to the user or delegated to the CSP. That's why trust vectors would be visible to the users in the beginning of a conversation. These vectors could be indicated in plain text or by symbols to the user. The trust vectors involved in reTHINK trust engine are as follows:

- The communication partner is authenticated
- The communication partner is already known by former transactions/calls/communications
- The communication partner is already known by other communication partners or in a certain community
- The communication partner is not on a black list (for spammer etc.)

The trust engine consists of two basic modules the authentication validation module and white/black list module. As a result of the authentication validation process and the white/black list lookup gets information whether the caller is trustworthy or not. The information is indicated in an appropriate way to the user e.g. by showing dedicated symbols in a display.

1) Authentication Validation

Before a trust engine is able to provide any kind of information about a communication party it needs an identifier as an input parameter. As this identifier must be authenticated the IdModule does this verification as explained in section IV A. The decision, whether the IdP is trustful or not is up to the user or can be delegated to the CSP. The IdP might have direct relationship with the CSP or it could be certified as a part of a Trust Framework.

2) White and Black list

The lists contain all identities of known communication partners based on previous call or any kind of communication sessions. The list can be filled from various sources. The first source is the user client in which all directly known communication partners can be white listed .Second sources are other user clients of the same community, company, family or group. In this case identities are marked as trusted when they had a conversation with other communication partners or in a certain community. In cases of malicious behavior or certain attack patterns the regarded identity is listed in the black list. Thus this module indicates whether the caller identity is well known (whitelisted) or already known for malicious behavior (black listed). A third status may be given back in case that there is no information available about the caller.

V. DIRECTORY SERVICES

Registry, Catalogue and Discovery are the three components in reTHINK framework collectively called as Directory services. The registry stores information on how to reach a Hyperty instance, the catalogue stores descriptors of Hyperties available for users whereas discovery services provides means for users to find other users to initiate communication.

A. Registry services

In order to initiate a connection to a specific Hyperty Instance, it is required to know the destination Hyperty's current network address. The reTHINK architecture proposes to allow frequent changes of the location and also of the domain (CSP) of the device running the Hyperty. For that the information about the Hyperty and running Hyperty Instances is published and updated in a Registry. All the information

required to initiate a connection to Hyperty is published in this registry upon the initiation of a Hyperty Instance and is removed when the instance is terminated. If the network address of the device running the Hyperty Instance changes during runtime, the information is updated automatically to provide a seamless way to connect to the Hyperty Instance. This makes the Registry into a directory of 'live' users who are available to receive communications.

As the reTHINK architecture proposes to allow seamless migration of users between different CSPs, a globally unique identifier (GUID) is assigned to each user. This identifier is domain agnostic and derived from a user's public key pair and therefore can be used regardless of the CSP's domain even when changing the CSP. However UserID is domain dependent identifier which indicates the CSP's domain of the user. This identifier changes when user shifts to another CSP domain.

Registry services in reTHINK comprise two main components, Global Registry and Domain Registry. While the Global Registry is built on P2P technology similar to the Global Social Lookup Service [12][13] in order to provide a distributed and domain-independent service, the Domain Registry follows a traditional client-server approach to facilitate fast response times as well as frequent Write operations due to updated client information. The Global Registry resolves a user's GUID to the CSP specific UserID, whereas the Domain Registry translates UserID to the information about the Hyperty instances of this user, i.e. the IP location of a reachable endpoint for this user. This allows other users and network devices to initiate a connection to the actual Hyperty location of the user.

Figure 4 outlines the connection between Hyperty, Global Registry, and Domain Registry. When initiating a connection to a Hyperty instance, the Global Registry is queried to resolve the targeted user's GUID to UserID and its current Domain Registry server. In the following step, the Domain Registry is used to resolve the user's UserID to his Hyperty Runtime's current IP address.

B. *Catalogue service*

The Catalogue Service conceptually acts as a software repository that contains information and the executable code for Hyperties and ProtocolStubs. For each, descriptors are treated as a resource with all relevant information in form of resource attributes. Attributes in-common for all descriptors are a globally unique identifier, a (verbal) description of the resource, a flag denoting if the resource is a Hyperty or ProtocolStub, and the actual executable source code or a web-link to retrieve the code from. Treating Descriptors as a resource with associated attributes has notable advantages, closely fitting the reTHINK requirements:

Use case agnostic bootstrapping – ReTHINK targets at supporting H2H, H2M, and M2M use cases. As the Catalogue is the initial entry point providing components to be executed at end user devices, access to the Catalogue has to be provided via standard protocols widely and commonly used in either use case domain. The resource-based view of catalogue entries allows representing them according to OMA-TS-LightweightM2M [14]. Create, Read, Update and Delete

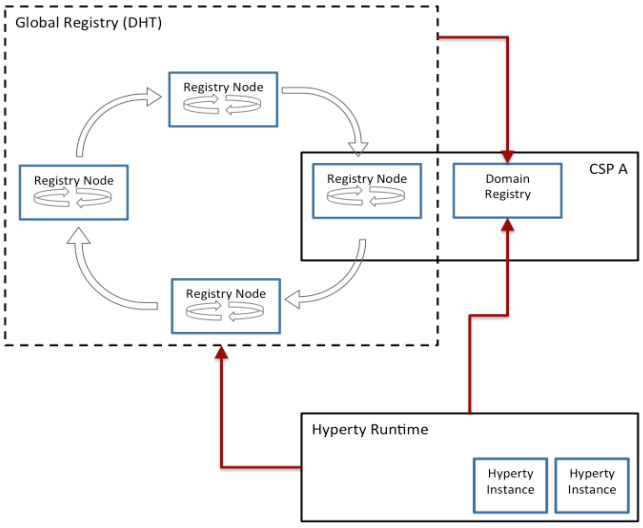

Figure 4: Global and Domain Registery

(CRUD) based access can be directly mapped to http [15] or lwm2m/coap-based [16] operations.

Discovery – reTHINK employs for its Catalogue RFC 6690 [17] specifying URIs to Descriptors as entries of a "well-known" core, which allows standard-compliant discovery of all stored resources via an http-get operation.

C. *Discovery services*

In todays fragmented communication landscape with closed domains, the address book is a common way to find a communication partner within a domain. However, how would you get in touch with other people across networks or domains in a peer-to-peer architecture, when no address book entry is available?

A web-centric peer-to-peer architecture such as reTHINK architecture requires a natural and intuitive way to initiate communication beyond the address book, which benefits from contextual information like neighborhood, location or interests and new technologies like social graphs and semantic interpretation. The provider of such a discovery service has to be a reliable actor, so that customers are willing to leave their sensible data. This is an opportunity for Telco operators to offer another trusted web service. This service could also be an extension to the offerings of trusted IdPs.

The idea is to provide a discovery service which enables searching for your conversational partners and initiates your conversation based on what you already know from the search for content on the internet. The discovery service receives natural written/spoken search requests like e.g. "John Miller from Orange in Paris" or "all people interested in playing football in my neighborhood". A semantic interpretation component within the discovery analyses the natural speech and returns the registry key(s) of those discovery-repository entries that match the search request description. The registry keys are used for a lookup of the communication endpoints in the registry. In order to be detectable for others, each user is free to create his own entry in the registry and publish data like e.g. his full name, pseudonyms, location, profession or hobbies.

These data items should be protected according to the user's policies.

The discovery service is intended to be used directly by users, to search and communicate with other people without any need to be a subscriber of a certain communication service. On the other hand, the discovery service might also be an enabler to enrich existing communication services with cross-domain connectivity. In that case, the user might benefit from cross domain connectivity together with service specific features like previously used contacts or call history.

VI. GRAPH CONNECTOR

In the reTHINK framework, Hyperties manage the communications between users. The connections formed by the communication patterns between users form a (social) graph. When each communication endpoint manages a list of previously contacted users, similar to the idea of a local address book or contact list, each endpoint stores a part of a social graph. This distributed graph information can be used to estimate the trust level between users that are not connected. It can also be used, for the definition of policies, for search and recommendation, or for the distribution of content along relevant paths.

In current, centralized online social networking systems like, e.g., Facebook, the social graph – indicating the connections between users – is stored with the service provider. In reTHINK, the concept is that the graph is stored in a distributed manner on the users' devices. Each user stores information about her connections to other users. , There is no central service provider that has all the users' data. Furthermore, in current online social networks, mostly explicit connections, usually indicating a friendship relation, form the social graph. In reTHINK, we have a graph that can have named and weighted edges between vertices. Edges could also be directed – indicating a unidirectional relationship between two users – or be aging – indicating the decay of an edge after some time.

The idea is to have different applications build on different edges of the graph. Thus, in the end, there is a distributed graph that not only indicates friendship relations but also relations like similar taste in music, similarity in location traces, etc., thus, forming additional tiers in the social graph based in common contexts and locations [18].

In order to be able to reliably connect to other users in the graph, globally unique identifiers will be employed. In order to fully utilize the graph, each node will have to know more than just its neighbors. The local view of the graph, consisting of the direct connections to other users, must be expanded, e.g., by comparing common contacts or profile data with unknown users to establish a bigger graph at each node. Such calculations should respect privacy and could be done when establishing a new edge in the graph and at certain regular intervals. In order to respect the users' privacy, hashing algorithms can be employed in order to mask identifiers or profile data. Existing research utilizing Bloom filters when comparing user profiles while preserving privacy seems most promising [19]. Using a minimal data structure such as for Bloom filters, enables calculations to be made on smartphones with bandwidth and battery constraints.

Overall, regarding online social networks and (social) graph data, privacy and data ownership are important questions to address. In a recent report, questions of data ownership and data control were discussed [20]. The results of the report can help design a privacy-aware system that leaves the user in control over her data. By distributing the graph among the users' devices, each user is in control of its own data. Hashing and Bloom filters further ensure privacy when users exchange information about other users. Regarding search, recommendation, and content distribution, each node in the P2P network, i.e., each user (or an application on behalf of her), can decide whether to use or share private information.

VII. GOVERNANCE & POLICY MANAGEMENT

The management of Hyperties is determined by the CSP administrative domain that issues this Hyperty. CSP's governance operations consist mainly of two tasks: the CSP interaction with other CSPs, and the life-cycle of the communication services provided by the CSP. The CSP takes the responsibility of several supporting services that are involved in the Hyperty life-cycle. The implementation of these services is distributed between backend servers in the CSP infrastructure and in the end users devices. Hyperty governance involves global management of the defined rules, expressed through policies. The main supporting services involving security considerations in a broad sense are identity management with Authentication, Authorization and Accountability (AAA) issues, discovery management with privacy issues, trust management, as well as QoS (Quality of Service), as depicted in Figure 5.

Since the Hyperties are created and received from remote CSPs, to ensure the correct governance of the Hyperty at the endpoint runtime environment, the downloaded Hyperties need to obey the policies defined by their respective CSPs. On the other hand, their execution behavior must also be controlled and coordinated by the user's runtime, in order to ensure that the Hyperty does not compromise the user's device. As such, the local execution of Hyperty has to be controlled by the device. To guarantee this, the runtime environment includes several mechanisms, such as sandboxes and user defined policies. These policies determine access to local resources and

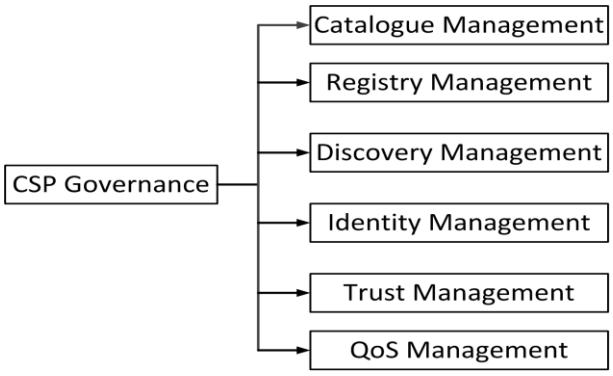

Figure 5: CSP governance impacts

communications with other entities of interest. The Policies involved in the end devices are of two types. Polices defined by the CSP and accepted by the end user in the subscription downloading the Hyperty. The second type are specific to the device, the goal of which is to maintain its native security level by controlling the communications of the Hyperty together with the allowed access to local resources.

Using a classical PDP/PEP/Policy repository structure [21], the management of the policies involved in the Hyperty life cycle can be summarized in the schema in Figure 2. The specification of these policies can be done using existing access control and messages exchange policy languages, in particular the XACML and WS-Policy [22], both expressed using XML specification language.

VIII. CONCLUSION

This paper describes IdM aspects of a new web communication framework that is developed by the reTHINK project. The main goal of the reTHINK framework is to devolve the communication session management to the endpoints, where the Hyperties operate as autonomously as possible. Such a P2P architecture requires a fresh approach to governance, IdM and trust in the communicating parties who download software, and so on. This paper explores aspects of peering identities, user registry and discovery, social graph based facilities and a trust engine that supports the P2P approach, and governance of the edge based processing, where multiple CSPs share a single runtime environment.

The reTHINK framework treats dynamic user registration from their devices as the means of establishing user's availability for incoming calls. User addressability is facilitated by globally unique and routable GUID identifiers that are associated with users' known attributes. The level of trust in callers is presented to the called parties, based on previous conversations or same-circle membership. Social network techniques are also used for creating meaningful linked address books, utilizing social circles and common interests. The framework also includes procedures to manage multiple CSPs communication modules (Hyperties) in a single endpoint and safeguard the operation in the distributed environment.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 645342, project reTHINK.

REFERENCES

[1] A. Bergkvist, D. C Burnett, C. Jennings,"WebRTC 1.0: Real-time Communication Between Browsers," W3C Working Draft, 10 February 2015. [Online]. Available: http://www.w3.org/TR/webrtc/.

[2] C. J Jennings, T. Hardie and M. Westerlund, "Real-time communications for the web," *IEEE Communications Magazine,* vol. 51, no. 5, pp. 20-26, April 2013.

[3] E. Bertin, S. Cubaud, S. Tuffin, N. Crespi and V. Beltran, "WebRTC, the Day After: What's Next for Conversational Services?," in *Intelligence in Next Generation Networks (ICIN)*, 15-16 Oct. 2013.

[4] S. Becot, E. Bertin, J.-M. Crom, V. Frey and S. Tuffin, "Communication services in the Web era: How can Telco join the OTT hangout?," in *Intelligence in Next Generation Networks (ICIN)*, 2015.

[5] P. Chainho, K. Haensge, S. Druesedow and M. Maruschke, "Signalling-On-the-fly: SigOfly," in *Intelligence in Next Generation Networks (ICIN)*, Feb. 2015.

[6] M. Jones, J. Bradley and N. Sakimura, "JSON Web Token (JWT)," Internet Engineering Task Force (IETF), [Online]. Available: https://self-issued.info/docs/draft-ietf-oauth-json-web-token.html. [Accessed May 2015].

[7] N. Sakimura, J. Bradley, M. Jones, B. d. Medeiros and C. Mortimore, "OpenID Connect Core 1.0 - draft 17," 3 February 2014. [Online]. Available: http://openid-foundation-japan.github.io/openid-connect-core-1_0.ja.html.

[8] V. Beltran, E. Bertin and S. Cazeaux, "Additional Use-cases and Requirements for WebRTC Identity Architecture," March 9 2015. [Online]. Available: https://tools.ietf.org/html/draft-cazeaux-rtcweb-oauth-identity-00.

[9] C. Cahill, "Liberty ID-WSF Advanced Client," [Online]. Available: http://www.projectliberty.org/liberty/content/download/3905/25642/file/liberty-idwsf-adv-client-v1.0.pdf.

[10] J. Vossaert, P. Verhaeghe, B. Decker and V. Naessens, "User-Centric Identity Management Using Trusted Identity Module -Binding Mobile Phone Secure Elements to the OpenID Connect Protocol," in *SARSSI 2014, 9th Conference on Network Security Architectures and Information Systems*, Saint-Germain-au-Mont-d'Or (Lyon), France, 2014.

[11] "Trusted Identity Module," Orange, [Online]. Available: https://github.com/Orange-OpenSource/TIM.

[12] S. Göndör, F. Beierle, E. Küçükbayraktar, H. Hebbo, S. Sherhan and Küpper, "Towards Migration of User Profiles in the SONIC Online Social Network Federation," in *International Multi-Conference on Computing in the Global Information Technology (ICCGI)*, 2015.

[13] S. Gondor and H. Hebbo, "SONIC: Towards seamless interaction in heterogeneous distributed OSN ecosystems," in *Proceedings of the IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*, 407-412

[14] "Lightweight Machine to Machine Technical Specification Draft Version 1.0," Open Mobile Alliance, 19 Jun 2014. [Online]. Available: http://dev_devtoolkit.openmobilealliance.org/IoT/LWM2M10/doc/TS/index.html.

[15] M. Belshe, R. Peon and E. M. Thomson, "Hypertext Transfer Protocol Version 2 (HTTP/2)," Internet Engineering Task Force (IETF), May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7540.

[16] Z. Shelby, K. Hartke and C. Bormann, "The Constrained Application Protocol (CoAP)," Internet Engineering Task Force (IETF) , June 2014. [Online]. Available: https://tools.ietf.org/html/rfc7252.

[17] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format," Internet Engineering Task Force (IETF), August 2012. [Online]. Available: https://tools.ietf.org/html/rfc6690.

[18] F. Beierle, S. Göndör and A. Küpper, "Towards a Three-tiered Social Graph in Decentralized Online Social Networks," in *Proceedings of the 7th International Workshop on Hot Topics in Planet-scale mObile computing and online Social neTworking (HotPOST '15),ACM*, 2015.

[19] M. Alaggan, S. Gambs and A.-M. Kermarrec, "BLIP: Non-interactive Differentially-Private Similarity Computation on Bloom filters," in *Stabilization, Safety, and Security of Distributed Systems*, Springe, 2012, p. 202–216.

[20] P. Bosesky, P. H. Deussen, A. Quandt, S. E. Schulz and L. Strick, "Datenhoheit in der Cloud," Fraunhofer Fokus, Berlin, 2013. [Online]. Available: http://publica.fraunhofer.de/dokumente/N-281950.html.

[21] R. Yavatkar, D. Pendarakis and R. Guerin, "A Framework for Policy-based Admission Control," IETF, January 2000. [Online]. Available: https://tools.ietf.org/html/rfc2753.

[22] M. Karusseit, T. Margaria and H. Willebrandt, "Policy expression and checking in XACML, WS-Policies, and the jABC," in *Proceedings of the 2008 workshop on Testing, analysis, and verification of web services and applications. ACM*, 2008.