In-the-know: Recommendation Framework for City-supported Hybrid Cloud Services

Manav Singhal*, Jay Ramanathan+, Prasad Calyam*, Marjorie Skubic*
University of Missouri-Columbia*, The Ohio State University+
Email: *msinghal@mail.missouri.edu*, *jayram@cse.ohio-state.edu*, *calyamp@missouri.edu*, *skubicm@missouri.edu*

***Abstract*—Hybrid cloud architectures are particularly attractive to leverage city-level investments for building customized clouds, and for extending them to leverage public clouds. A successful design of the hybrid cloud architecture should facilitate the provisioning of scalable and secure services suited to a variety of communities such as residential homes and high-tech business incubators. In this paper, we present a novel "In-the-know" recommendation framework for provisioning of cloud resources in the form of 'on-demand contracts' to address the challenges in delivering the hybrid service variations for different community and individual needs. Our recommendation framework uses knowledge of the city's socio-economic goals/values as well as user preferences in terms of cost, performance and mobility. Using such knowledge, it recommends dynamic decisions by choosing from various provisioning alternatives in order to: (a) ensure optimal user Quality of Experience (QoE) in service delivery, and (b) effective utilization of hybrid cloud resources. We validate our recommendation framework using service composition experiments to satisfy an exemplar collaboration use case in an actual city-supported hybrid cloud testbed involving citizen consumers.**

***Index Terms*—Context-sensitive service recommendation; Smart cities with cloud-based utilities; Hybrid cloud architectures; Service composition usability**

I. Introduction

Cities have long-partnered with service providers of common utilities such as electricity and water. These partnerships have fostered the building of market ecosystems that provide cost-effective and high-quality service choices to their citizen consumers. Emerging concepts such as "smart city" [1] have led to cities creating information and communication technology (ICT) infrastructures. These infrastructures provide value-added services for citizen services such as for e.g., access to public libraries, collaborative transportation planning, and construction of smart buildings. There is now a rise in data centers and fiber accessibility for businesses and residents through new services such as Google Fiber [2]. As a result, many cities have started including "high-speed broadband access" and "cloud compute" to their list of *utility investments*. In addition, city councils are making investments that support subsidized cloud computing services for community programs that provide public benefit, as well as business growth that is heavily data-driven in today's 'knowledge economies'.

This material is based upon work supported by the National Science Foundation under award numbers CNS-1347889, IIP-0753710, CNS-1346789 and VMware. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or VMware.

There are several benefits in transforming the traditional desktop-server paradigm to hybrid cloud service architectures. The transformation leverages a city's local ICT investments for building private clouds, and extends them 'on-demand' to utilize the benefits of public clouds. It is important for cities to invest in private clouds that are managed by trusted non-proprietary service providers. This can support latency-sensitive and privacy-sensitive application tasks that require cloud resources to be close to the user. Also, having private cloud infrastructures with high-speed and programmable networking devices allows for overlay networks to be constructed with VLAN extensions from consumer sites to remote sites. Such approaches to overcome Internet bottlenecks, and seamlessly stitch networks across multiple domains are being increasingly adopted. Particularly, we see uptake of OpenFlow protocol based 'slices' within academia [3], and commercial services such as Amazon Direct Connect [4].

Further, with the availability of virtualization management technologies such as OpenStack [5] and VMware Horizon [6] that use Federated Identity and Access Management (Federated-IAM), there are opportunities for unified management of user credentials and application policies. For instance, passwords of users need not leave city boundaries. Moreover, users can be provided 'pay-as-you-go' access to diverse application choices in virtual desktops through Desktop-as-a-Service (DaaS) and Software-as-a-Service (SaaS) offerings. These offerings are particularly convenient for mobile user access, and synchronized/secure data access across user devices.

To realize the transformation of application provisioning, the design of a city's hybrid cloud architecture should facilitate the provisioning of subsidized, scalable and secure services [7]. In this paper, we present a novel "In-the-know" recommendation framework for provisioning of hybrid cloud resources in the form of 'on-demand contracts' to address the design challenges within city-supported service delivery. Our framework name is derived from the fact that it builds upon the knowledge of the socio-economic goals/values reflected in the city's investments and subsidy policies. It maps socio-economic goals/values to hybrid cloud resource reservations (i.e., network, compute, storage) in an on-demand manner for citizen consumers. An example of a socio-economic value is when a city invests in building fiber laterals to buildings with high-tech companies to enable them to have access to ICT resources that help them grow faster, and thus contribute to the local economy. Alternately, a public library in a city has long been a critical socio-economic value proposition.

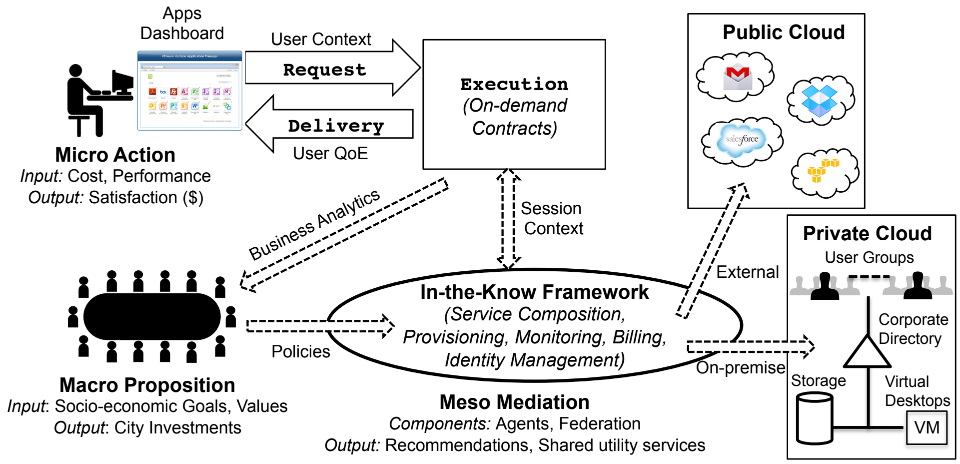


Fig. 1: In-the-know Ecosystem showing mediation between micro (citizen/consumer), meso (service provider) and macro (city) perspectives

In addition, our framework orchestrates the network and resource configurations in order to customize them to meet the diverse user preferences when they access domain (e.g., manufacturing, healthcare) software. The user preferences can be known in terms of 'cost' that they are willing to afford, 'performance' they are expecting from applications, and 'mobility' factors (i.e., device type and end-to-end network bandwidth) that can be inferred when users are accessing the applications.

This ambitious goal requires a way to deal with complex systems that is provided by interlinking stakeholders at the macro, meso and micro levels as illustrated in Figure 1. Research towards this is by authors in [10] who report that socio-economic capital (e.g., knowledge, value) at the macro level is itself built as a result of many interactions at the micro level. The micro level focuses on the success of a single interaction, and our aim is to use these as 'on-demand' contracts in our recommendation framework. These micro interactions are in turn facilitated by operational or meso level network connections and the city infrastructure. Our recommendation strategy for interlinking falls under the category of content-based recommendation systems [12]. It is based upon the concept of cosine similarity between the user preference and the candidate service vectors that are sparse by nature.

We validate our In-the-know recommendation framework using service composition experiments to satisfy an exemplar collaboration use case in an actual city-supported hybrid cloud testbed involving citizen consumers. We use empirical evidence from the perceptible performance comparison of a healthcare application under different network conditions. With this approach, we study the extent to which they provide access alternatives for service composition to satisfy user preferences of cost, performance and mobility.

The remainder of the paper is organized as follows: Section II discusses related work. Section III describes the In-the-know recommendation ecosystem. Section IV presents our mediation approach and our dynamic decision processing logic within the recommendation framework. Section V details our service composition validation experiments for an exemplar collaboration use case. Section VI concludes the paper.

II. Related Research Areas

Cities typically invest in ICT and advanced networking technologies for enhanced governance and participatory processes that define appropriate public service [1]. In [8], the authors presented a cloud-mediated architecture and a social media platform for game-type interactions with citizens. Significant work has been conducted in the EU FP7 COCK-PIT project [9] where participatory design of services meets conflicting requirements. Our work is comparable to the smart city ICT policy work in [19], where hybrid cloud frameworks facilitate decision making in sustainable services through citizen engagement, participatory governance and collaborative decision-making. Our study directly validates how a smart city that is ready to offer easy-access cloud services to citizens can leverage novel hybrid cloud platforms such as VMware Horizon, and broadband infrastructures such as Google Fiber.

In our framework, the consumer perceptible factors at the micro interaction (session) level are based on the Application Performance Index (Apdex) [11]. This is an industry standard to report, benchmark and track the web application performance by assigning Apdex score based on the web response time as perceived by the user in different network conditions. However, since Apdex is not suitable to benchmark the performance of individual productivity applications such as Microsoft Office, we extend it in our study to mediate recommendation of SaaS and DaaS to consumers.

In order to mediate the consumer perceptible factors and the provider services, we build on the work in recommender systems [12] that predict users and items. Such recommendations can be broadly classified into content-based and collaborative categories. In a content-based recommender system, Cosine and Jaccard similarity measures are used on item properties for recommendations. On the other hand, in collaborative type recommenders [12], the items are recommended to the users based on the ratings/similarity measures from diverse users with similar preferences. We use the content-based recommender approach because we are trying to recommend the service based on individual preferences.

In [13], the authors proposed a cloud-based mobile multimedia recommendation system, which can reduce network overhead and speed-up the recommendation process. Users are classified into several groups according to their context types and values for collection of user contexts, user relationships. Another recent collaborative recommendation work is by [14], where they proposed a cloud-based framework viz., *OmniSuggest* that uses Ant Colony algorithms, social filtering and hub scores to generate optimal location recommendations addressing real-time issues such as traffic and weather conditions.

III. City Scenarios and Recommender Characterization

The goal of this research is to start with real-world citizen communities and scenarios as overall context. With this, we show how to move to a residential future based on 'center-out' innovation ('center' being economic development and job growth, and 'out' being the support of residential, library, high-tech startups). Below, we outline how we develop our recommendation framework that can optimize user QoE leveraging user participation and needs from three different scenarios.

A. Scenarios

1) Residential Scenario: Providing a healthcare solution to elders through virtual desktops and thin-clients is a use case where we can integrate and validate our recommendation framework. The macro perspective of the collaborating Healthcare Hospitals/Medical Institutions is to deliver hosted virtual desktops for use in different Kansas City homes. These virtual desktops provide interactive physical therapy interface to elderly people using 1 Gbps Google Fiber to provide high-definition video conferencing with 3D sensing. At the micro level, there are residential elder homes in Kansas City connected to Google Fiber who are potential users that will pay for an on-demand 'Eldercare-as-a-Service' (ECaaS) App.

2) Business Scenario: The macro perspective here is that of the City of Dublin, Ohio utilizing resources in the GENI project [15] with an ultimate goal of fostering local high-tech job growth. The meso infrastructure perspective is to leverage the unique potential of cloud platforms to provide on-demand high performance computing (HPC) resources at utility computing pricing for the startups in advanced manufacturing. For the users at the micro interaction level, a key need is to share a 'Simulation as a Service (SMaaS)' App [15] between remote collaborator sites for delivery of new manufacturing related model designs to customers via a SaaS delivery medium.

3) Community Associations Scenario: There are efforts in the Kansas City region to leverage broadband connectivity and data centers to support virtual desktop pools as part of a software lending library [16]. This concept allows expensive software application licenses to be 'checked-out' by concurrently by certain number of library members. Programs such as US Ignite and Mozilla Ignite that are encouraging efforts in [15] and [16] mentioned above, are targeted to create synergies between service providers, App developers and technology vendors. The purpose here is to develop next-generation Apps that leverage high-speed broadband, low-latency networks and cloud networking technologies such as OpenFlow.

B. Problem Summary

In-the-know recommendation framework design is aimed to work for different city scenarios having distinct needs and requirements based on a 'Subscription' model. Figure 2 characterizes the provisioning options for different user applications by choosing amongst device/network/compute/storage alternatives. Subscriptions need different options in their hybrid cloud configuration on top of the same underlying complex infrastructure, however they will pay only for the options they use. For example, the ECaaS App needs to be delivered with a thin-client (DaaS) with cloud computing versus using a physical desktop that has performance and mobility constraints, as well as scalability limitations.

We can formalize our recommendation framework problem to address the above city scenarios as follows: Let G1 and G2 be different group of Apps. Let A = Offline App and A* = Online App. Let A_i and A_i^* be Apps that belong to G_i. Let

$$V_i = \{C_i, P_i, M_i\} \tag{1}$$

Where i = A1, A1*, A2, A2* and C_i = Cost Score, P_i= Response Performance Score, M_i = Mobility Score

We design the recommender to use the citizen perceptible factors in the context of an exemplar collaboration use case under different network conditions. For this, we conduct scoring experiments that underlie the provisioning factors and predictions. Ultimately, our recommender system design is to take user attributes and recommend pertinent services in a large space of possible options. More specifically, for a recommender to mediate:

- It must explicitly know (or be able to sense) the consumer perceptible factors and preferences of cost, performance and mobility, and
- It must be able to dynamically control the service factors that cater to those preferences using 'on-demand contracts'.

We mediate user preferences and service provider propositions to guide the service provisioning with 'on-demand contracts'. Our work is novel in that for each application, we are making use of a *cost score* (C_i), a *performance score* (P_i) along with a *mobility score* (M_i) for providing context-aware recommendations to the user.

IV. Recommendation Framework: Mediation of Consumer-Provider Factors

A. Consumer-Provider Agents and Similarity Computation

Figure 3 illustrates the In-the-know agent architecture as envisioned and the overall workflow for context-aware mediation. We can see that the platform (including methods, agents, and middleware) has three types of agents: consumer agent, recommender agent and provider agent. Each agent has a different set of responsibilities and co-operates with other agents. We assume the end-user obtains the security token (label 0) to access the service using Federated IAM. With this, the user can request the services using various devices such as desktops/thin-clients using wired (LAN), smart phones or tablets using wireless networking (GPRS or Wi-Fi).

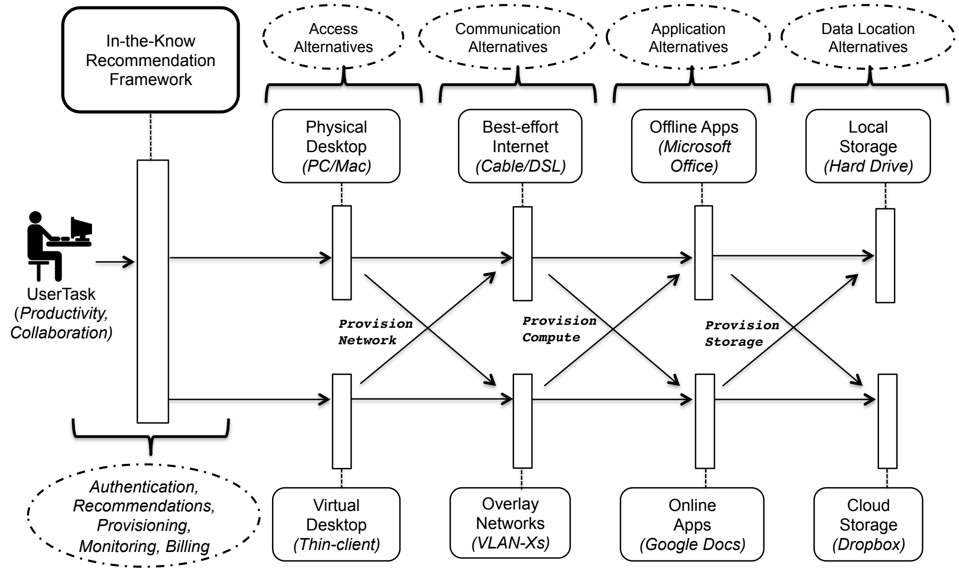


Fig. 2: Hybrid Cloud options illustrating alternatives for interaction, communication, application and storage options

The consumer agent is responsible for collecting the user's preferences like cost and performance (labels 1, 4). The mobility factor is configured automatically without user interaction by accessing device type and bandwidth. Similarly, the provider agent is responsible for maintaining rules for alternate provisioning options using the infrastructure at different cost, performance and mobility levels.

The Recommender agent uses its own rules based on similarity as elaborated below to perform the context-aware mediation of matching the various user preferences (label 2) and service propositions of providers (label 3). In addition, end-users take final decisions based on the recommendation provided by the framework (label 5) and tradeoff their virtual desktop rental bill with perceived QoE. Finally, the recommender agent predicts the service options with the highest matching score, obtains feedback from the consumer if needed, and sends directives to hybrid cloud resources (label 4).

Next, we illustrate how the Recommender agent collects the consumer's preferences as the vector Consumer Preference Vector $\overrightarrow{C_P}$:

$$\overrightarrow{C_P} = \{\textit{weight cost}, \textit{weight performance}, \textit{mobility}\} \tag{2}$$

We rate all these factors (cost, response performance, mobility) on a scale of 0 to 5 (with 0-lowest and 5-highest), however any other arbitrary scale can be used to the same effect. The user can also give the weight for cost and performance based on application nature or budget constraints. Higher weight would mean lower intent to compromise with consumer preference. Weight is defined in terms of percentage. For instance, if the user enters 60% preference for Cost, then the weight becomes 0.6 for Cost and 0.4 for Performance. Thus, if user enters W_i for 'Cost', (1- W_i) would be the weight for 'Performance'. Alternately, if the user is connected through a tablet with 4G and wants access his/her email service cost-effectively, then for the user $\overrightarrow{C_P}$ becomes $\overrightarrow{C_P}$ = {2, 3, 3}

The cost score defines the relative price preference; the performance response score defines the user performance preference in terms of better response time, and finally the mobility preference. By default, the cost score is 0 and performance response score is 5 i.e., the user wants service at lowest cost with best performance. The user's preferences are stored in the system and can be changed by the user at any point of time.

The *Provider Application Vector* ($\overrightarrow{P_A}$) is also similar, and scores (on a scale of 0-5) are given based on real-world experiments and objective performance measures. This is also expressed as:

$$\overrightarrow{P_A} = \{\textit{cost}, \textit{performance}, \textit{mobility}\} \tag{3}$$

For instance, $\overrightarrow{P}_{SaaSGoogleDocs}$ = {2, 3, 5} at mobility level 5. This vector denotes that at very high mobility (i.e., less bandwidth), the Google Docs is scored as 3 based on response time with the associated cost scored as 2. The cost score is based on the service provider's price, and we normalize the cost on a scale of 0-5.

Thus, we obtain the *consumer preference* vector $\overrightarrow{C_P}$ and the alternate provider application vectors $\overrightarrow{P_A}$. With these vectors, we can estimate the degree to which a user would prefer a specific type of service provisioning by computing the cosine similarity between these vectors. Cosine Similarity finds the similarity between two n-dimensional vectors based on cosine angle between them. It does not work for null vectors and is used in positive space. It is formally defined as:

$$CosineSimilarity = \overrightarrow{A}.\overrightarrow{B}/(|\overrightarrow{A}|.|\overrightarrow{B}|) \tag{4}$$

where $\overrightarrow{A}.\overrightarrow{B}$ is dot product of vectors $\overrightarrow{A}$ and $\overrightarrow{B}$, and $|\overrightarrow{A}|.|\overrightarrow{B}|$ is product of magnitude of vector $\overrightarrow{A}$ and $\overrightarrow{B}$.

By measuring the cosine angle between the vectors, we get a good idea of their similarity. A larger positive cosine

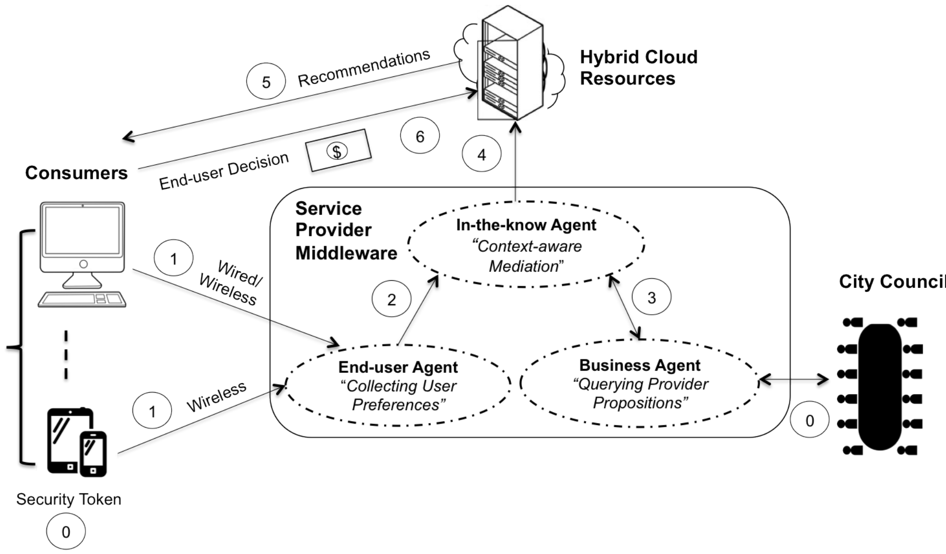


Fig. 3: In-the-know framework agents and workflow for context-aware mediation

value implies a smaller cosine angle and vice versa. The resulting similarity value ranges from -1 to 1. The value (-1) indicates exactly opposite, (1) indicates exactly same, (0) indicates independence and mid-values indicate intermediate similarity or dissimilarity. The smaller angle we have between vectors, the bigger (closer to 1) cosine value will result, and bigger will be the similarity.

B. Cost, Performance and Mobility (CPM) Model

In this section, we detail the Cost, Performance and Mobility (CPM) model used in our In-the-know recommendation framework to effectively capture the user preferences.

1) Cost: Our cost model provides various cost elements to define the cost to be charged for each computing/networking resource and fixed costs based on a user's service usage.

A chargeable computing/networking resource is any resource that must be considered when calculating the Information Technology (IT) operational costs. We referred to the VMware Chargeback manager [17] for defining our cost model. We considered the resources of CPU usage, Network (transmitted and received), Storage and vCPU number. Network usage will vary from user-to-user based on their available last-mile bandwidth of their Internet connection. Hence, we conducted experiments to measure network usage for defining cost with different scenarios explained in Section V.

Base Rate is the configurable overall rate that can be charged for a unit of chargeable computing/networking resource used for a particular duration. Table I shows the sample base rates (hourly duration) for different resources. We refer these sample base rates from a VMware vCenter and VMware Horizon setup on a Server.

A fixed cost is a definite cost that can be charged to the users, and can be accounted as either recurring costs or one-time costs. For instance, Google [2] charges \$300 as a construction fee for installation setup in Kansas City, Missouri. Another instance of costs involve VMware Licensing cost and

TABLE I: Sample Base Rate for Computing Resources

No.	**Chargeable Resource**	**Unit**	**Base Rate (\$)**
1.	CPU	GHz	0.0399
2.	Network (Received and Transmitted)	Gb/hour	0.08
3.	Storage	Gb	0.0013
4.	vCPU	Number	0.04

the hardware cost for setting up VMware Horizon [18]. We analyzed the total cost per user for accessing a virtual desktop using VMware Horizon 5.2 supporting 174 users. At the time of this writing, the cost estimate is \$483. However, we remark that this calculation will vary with time and the number of users supported. Hence, the cost function can be written as:

$$TotalCost = FixedCost + \sum_{i=1}^{n} B_i.t \tag{5}$$

where n = number of computing resources, B_i = Base Rate for i^{th} computing resources and t = usage time (in hours).

2) Performance: Performance measure is difficult to generalize for all the applications since every application has different requirements and features. For instance, video quality (in terms of frame rate) can be used as performance measure for the video conferencing application i.e., better video quality has better performance. For the Eldercare-as-a-Service (ECaaS) App, we analyzed the video quality (shown in Figure 4) running on virtual desktops and physical desktops both connected to Kinect sensors. The frame rate was analyzed at different network conditions defined in categories.

$$Performance(ECaaS) \propto VideoQuality \tag{6}$$

Another example for performance measure for SaaS and DaaS applications is response time. For instance, we analyzed load time of Google Docs (SaaS application) and open time for Microsoft Office (DaaS application) on virtual desktops with

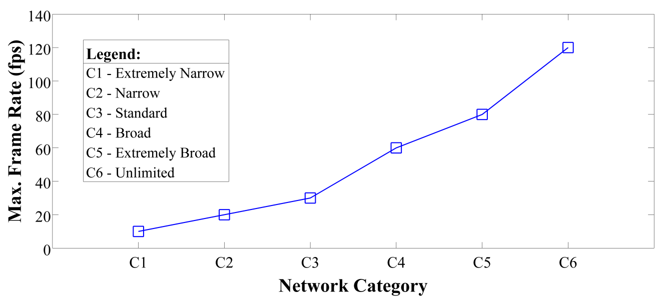


Fig. 4: Max. Frame Rate (fps) for ECaaS App under different network scenarios that are configurable with PCoIP tuner

various network settings and normalized them on a scale of 0 to 5 (shown in Figure 5). For low bandwidth up to 50 Kbps, the performance score for Microsoft Office is better than Google Docs since at very low bandwidth, time for webpage loading is greater than open time for Microsoft Office. However, as we keep on increasing the bandwidth, the load time for Google Docs improves at a relatively better rate and open time for Microsoft Office does not improve at a significant rate.

$$Performance(SaaS - DaaS) \propto ResponseTime \quad (7)$$

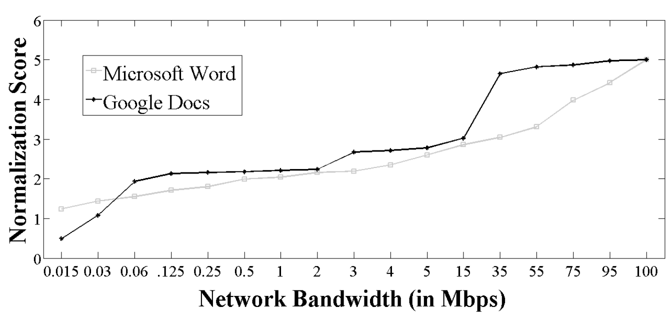


Fig. 5: Normalized Performance Score for Google Docs (SaaS) and MS Office (DaaS)

3) Mobility: The user mobility factor is automatically sensed based on context to minimize intervention by the user within our In-the-know framework implementation. To achieve this automatic sensing of context, based on the user's device and the network bandwidth from which the user gains access to the system, we define the mobility factor in the range of 0-5 where '5' denotes highest mobility and '0' denotes the least mobility. We remark that a different range can be chosen as desired to obtain the same effects.

As shown in Figure 6, Mobility is divided under Category X (Wired/Wireless; thin-client) and Category Y (Wireless; Tablet) based on the delivery medium of access. Each category is further sub-divided into Grades A, B and C based on network bandwidth in respective category, and we use scores in the range of 0-5 for these grades in each category. For instance, if the user is accessing the system using his/her tablet through a 2G connection, the user's mobility will be defined as 5. In order to determine whether the request is coming from a mobile device or a PC, the Consumer agent inspects the header that the browser sends whenever the user makes a HTTP request to a service mediated by the In-the-know framework. By parsing the different parameters, the Consumer agent detects the category of the device that the user is using to get connected such as a desktop, phone or tablet. After detecting user device, the agent further senses the network bandwidth consumption rate at the user's side.

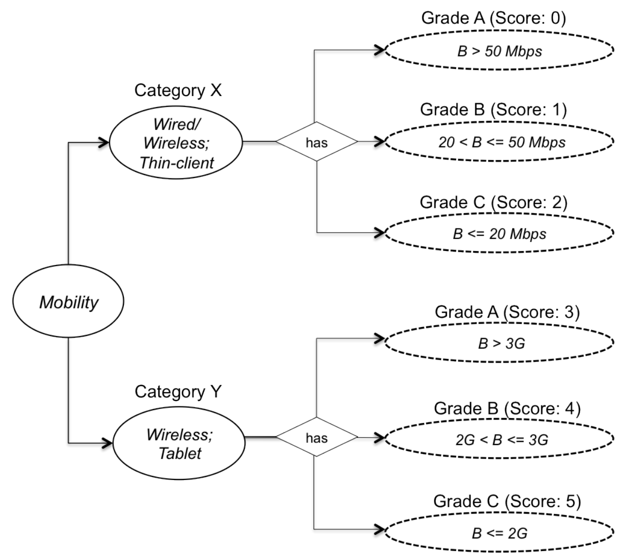
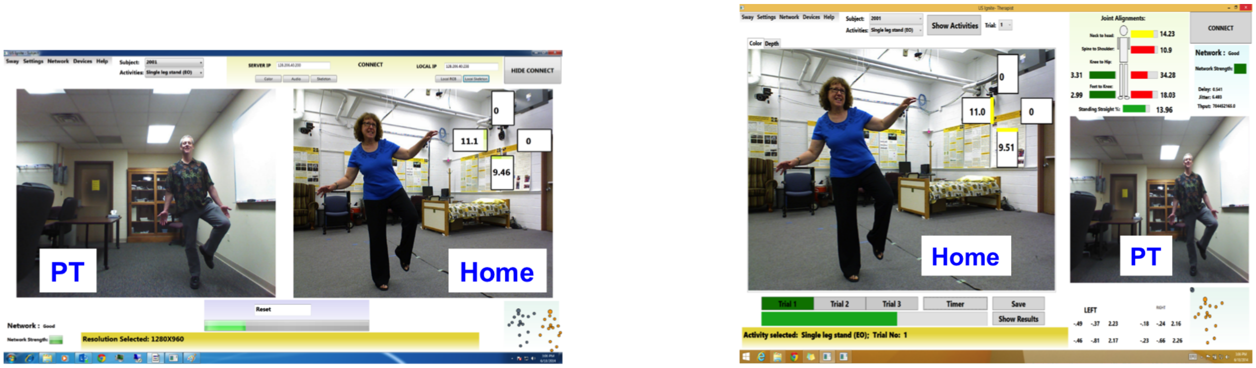


Fig. 6: In-the-know mobility configuration context logic

If the user's device is mobile/tablet, we assume that it is connected via a mobile telephony provider's network that support technologies such as 2G, 3G, 4G or Wi-Fi. With 2G GPRS (General Packet Radio Service), we have a maximum theoretical transfer speed of 50 Kbps but get 40 Kbps in practice; with 2G EDGE (Enhanced Data Rates for GSM Evolution), we can expect a theoretical transfer speed of max. 250 Kbps but get 150 Kbps in practice. Similarly, with 3G or UMTS (Universal Mobile Telecommunications System), we have a maximum theoretical transfer speed of 384 Kbps but get 360 Kbps in practice. And with 4G or LTE (Long Term Evolution), we have a maximum theoretical transfer speed of 150 Mbps but get 20-to-30 Mbps in practice. Similarly, if a user is connected through his/her desktop/PC/thin-client, he/she will be classified based on their connection's network bandwidth.

C. In-the-know Recommender Mediation for the Collaboration Use Case

We consider the Eldercare-as-a-Service (ECaaS) collaboration use case to demonstrate the In-the-know recommendation mediation. ECaaS App includes an interactive physical therapy interface that needs to be supported by high-definition video conferencing using thin-clients connected to Kinect devices. It also requires real-time processing of data from multiple in-home sensors across flexible, high-speed Google Fiber connecting patients in the home (Kansas City), remote care coordinators (MU Medical Experts), other clinicians. Figures



(a) In-home patient view

(b) Physical Therapist view

Fig. 7: Eldercare-as-a-Service App interface showing a user-centered interface design

7(a) and 7(b) show the in-home patient and physical therapist views, respectively of the ECaaS App interface, which has a user-centered design. We can observe that the App relies on 3D sensing data which augments the reality of the patient exercise postures to the physical therapist. Furthermore, we can observe that the overall interface experience is more richer than a typical videoconference session. Given the features and benefits of such a ECaaS App, a health care provider could offer physical therapy and other telehealth interventions as part of a subscription service with on-demand contracts of required hybrid cloud resources, and per-session pricing.

Figure 8 demonstrates the architecture for the proposed system with context-aware recommendations to serve the ECaaS App use cases. Basically, there are two use cases for collaboration in ECaaS that dictate the need for high-speed bandwidth - “Synchronous” Monitoring and “Asynchronous” Analytics.

- “Synchronous” Monitoring - Two-way high definition videoconferencing and 3D sensing between Kansas City and the Physical therapy clinic in Columbia, MO by elder care providers for e.g., Interactive Physical Therapy
- “Asynchronous” Analytics - To monitor the data logged in from Kansas City homes using in-home sensors for analyzing health of elders and providing real-time services such as Fall Alarm notifications to Care Givers

During the synchronous videoconferencing, the Kinect can generate the highest volume, with uncompressed depth, color and skeletal images. For instance, 1 second of video at a resolution of 1280 x 960 pixels for depth video at 30 frames per second generates approximately 140 MB of data. Collectively for depth, color and skeletal data with audio data, total bandwidth of approximately 500 Mbps is required for high definition two-way videoconferencing with enhanced synchronous sensing.

For this use case, we can assume the user preferences for cost and performance as a Consumer Preference Vector and design experiments (as detailed in the following section) to get the performance and cost scores for the ECaaS App, as well as suitable service composition recommendations under various network health conditions.

V. Service Composition Validation Experiments

In the previous section, we showed how our In-the-know recommender mediates the consumer preferences with the provider vector with objective scoring of options. In this section, we describe the closed-network setup and experimentation to validate the feasibility of our framework using the ECaaS App.

A. *Eldercare-as-a-Service (ECaaS) Cloud Testbed*

We actively conducted several experiments in collaboration with different Kansas City homes (patients) and University of Missouri, Columbia (UMC) (physical therapists, other clinicians, etc.). The experimental setup is organized in 2 layers: thin-client systems, and server-side virtual desktops running on VMware Horizon platform setup at UMC on a GENI Rack.

The thin-clients are connected to Kinect sensors that capture the user video (raw, depth and skeletal data). Our physical therapy interface runs on thin-clients at both ends - patients and physical therapists. At the thin-client sites, the testing environment with the closed-network is used for our offline experiments to measure the performance for the ECaaS App. We setup virtual desktop environment with VMware VDI [18] whose default thin-client protocols for VD access are RDP and PCoIP, respectively. We used Dell Wyse P25 thin-clients with a network emulator ‘Netem’ connected to virtual desktops.

For network monitoring and analysis of the impact of bandwidth on video quality, we use ‘PCoIP Tuner’ software [18] available for tuning the PCoIP parameters. We tune PCoIP to particular network scenarios, to cope with limited bandwidth or abundant bandwidth alternatives and thus extract the best performance out of the available resources. We analyze the video quality based on different bandwidth settings seen between the patients and physical therapists.

B. *Experimental Data*

For the ECaaS application, we measure the performance of the application in terms of video quality (frames/second or fps). We conducted various experiments to analyze the amount of bandwidth consumed at various fps. We define Mobility based on the minimum and maximum link-rate that

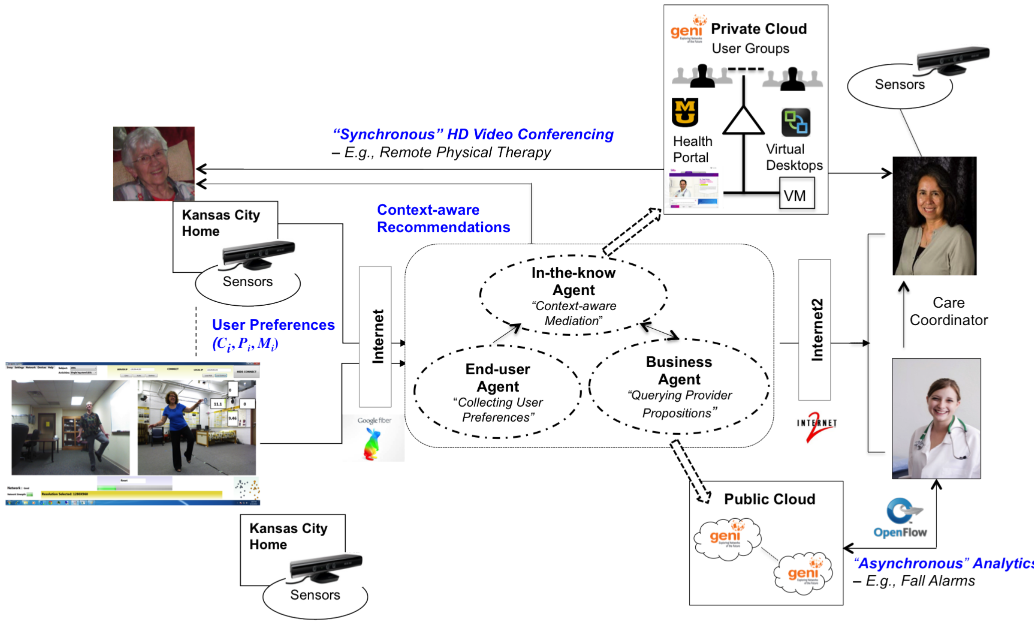


Fig. 8: "Eldercare-as-a-Service" System Architecture with 'In-the-know' recommendation framework

was available. Thus, we indexed Mobility into 3 classes and normalized the values on a scale of 1-3, 1 being least mobile while 3 being highest. We remark that an alternate scale can be chosen for the study purposes to the same effect as desired.

Mobility Score = 3, if Link rate is <= 10 Mbps = 2, if 10 Mbps <Link rate <= 50 Mbps = 1, if 50 Mbps <Link rate <= 90 Mbps

Next, we define the Performance score based on the fps value. We know that higher the fps value, higher is the image quality. While conducting the experiments, we tuned the PCoIP tuner to check the change in the image quality at different fps values. We observe that there is a maximum fps we can achieve at each Mobility level due to available bandwidth at each mobility level. We, thus defined 3 performance scores at different fps values for each Mobility level and analyzed average bandwidth consumption at those fps levels. With the above understanding, we define the following rules:

- When Mobility score = 1, Performance score is defined as 1 at 10 fps, 2 at 40 fps and 3 at 70 fps.
- When Mobility score = 2, Performance score is defined as 1 at 10 fps, 2 at 30 fps and 3 at 50 fps.
- When Mobility score = 3, Performance score is defined as 1 at 10 fps, 2 at 20 fps and 3 at 30 fps.

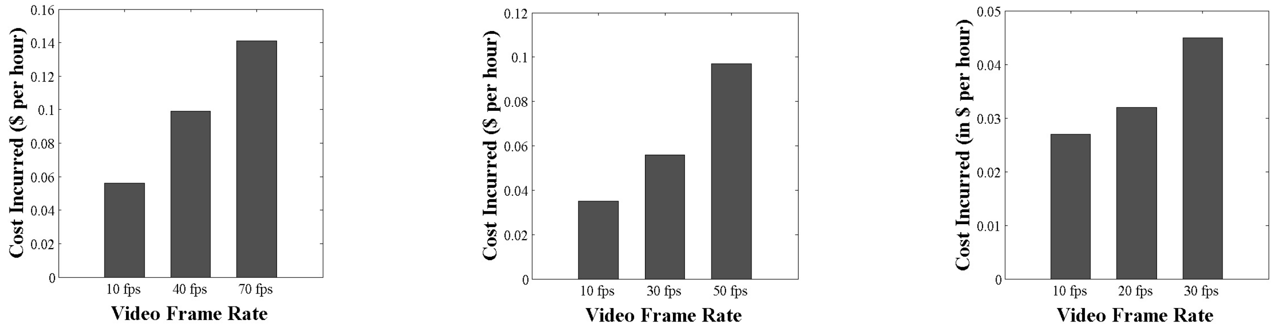
Next, we compute average bandwidth consumption per hour. By using the base rate for network (transmitted and received) in Table 2, we plot the graph of costs incurred at various performance levels (fps) under different mobility levels as shown in Figures 9(a)-(c). We can observe that the cost will vary with time, and with the number of users simultaneously accessing the resources. We considered a sample base rate of $0.08 Gb/hour for network usage of the virtual desktop that hosted the ECaaS App. Based on the above cost calculation for network usage, we calculated the normalized cost score on a scale of 1-3 as follows:

Cost Score = 1, if cost is <$ 0.035 = 2, if $0.035 <cost <$0.06 = 3, if cost >$0.06

Hence, based on different cost score, performance score and mobility score factors, we recommend the best alternative that fits with user requirements within the ECaaS App in terms of Good, Acceptable or Poor performance grades. Based on the cost score that we get using our Cost model, we tried to analyze the relationship between the three CPM vectors in the experiment context. We find that Performance is directly proportional to Cost, while it is inversely proportional to Mobility. Thus next, we attempt to rate the feasibility of the recommendation made by our recommendation system. Feasibility Index (FI) can be defined as the level to which our In-the-know system implementation satisfies user requirements at different CPM inputs. We compute FI for each mobility score as -

$$FeasibilityIndex = (C_u.P_u)/(C_r.P_r) \tag{8}$$

Where C_u and P_u are the weighted Cost score and weighted Performance score preferences provided by a user; whereas, C_r and P_r are weighted Cost and Performance scores that are recommended to the user by our In-the-know framework. We calculated the feasibility index for the 27 possible test cases, and assume equal weights for cost and performance score. As it can be seen from Figure 10, out of the 27 test cases, our recommendation system provided the most feasible recommendation for 23 cases where FI is less than 1. For the remaining 4 cases, the request was not feasible due to the FI being greater than 1. Thus, our In-the-know system implementation recommended the best possible solution given any of the user preferences, and we see that our recommendation system gives the most feasible solution in 85% cases.



(a) Mobility Level 1 (b) Mobility Level 2 (c) Mobility Level 3

Fig. 9: Cost incurred at various performance levels (fps) for different Mobility Levels

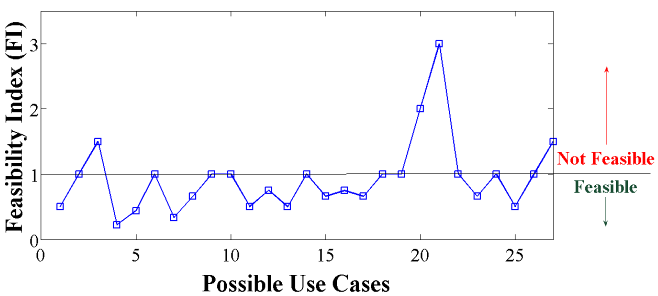


Fig. 10: Feasibility Index (FI) for various 27 possible cases in ECaaS scenario

VI. Conclusion

In this paper, we present a novel In-the-know framework and recommender that meets user perceptible factors while guiding the selection of provisioning options based on objective scoring of cost, performance and mobility factors. To validate our framework, we characterize an exemplar collaboration use case using cosine similarity between user preference factor vector and provider service (application/service) factor vectors that were mediated by rules to adapt the hybrid cloud environment.

Our future work involves extending our recommender framework for other city community scenarios, and collecting additional measurements related to different App use cases that require high-bandwidth and low-latency network paths to remote public cloud resources. Comparison of our content-based recommender method with other approaches such as collaborative recommenders to study the user context and scalability issues is also part of our future work. Thus, our work lays the foundation for a new smart city paradigm that meets the needs of citizen consumers, who benefit from a city's investments for utility-model based cloud computing services.

References

[1] Z. Khan, S. Kiani, "A Cloud-based Architecture for Citizen Services in Smart Cities", *Proc. of IEEE UCC*, 2012.
[2] Google Fiber - http://fiber.google.com
[3] N. McKeown, T. Anderson, H. Balakrishnan, et. al., "OpenFlow: Enabling Innovation in Campus Networks", *ACM SIGCOMM CCR*, 2008.
[4] AWS Direct Connect - http://aws.amazon.com
[5] OpenStack: Open Source Cloud Computing Software - http://www.openstack.org
[6] VMware Horizon: Virtual Desktop Infrastructure Management Software - http://www.vmware.com
[7] D. Ludlow, Z. Khan, "Participatory Democracy and the Governance of Smart Cities", *Proc. of AESOP Congress*, 2012.
[8] J. Ramanathan, R. Ramnath, N. Singh, Z. Xu, Y. Xu, "Sense-respond Cloud Mediator Architecture for Services Evolution", *Proc. of ACM SAC*, 2011.
[9] A. Hartman, A. Jain, J. Ramanathan, A. Ramfos, et. al., "Participatory Design of Public Sector Services", *Proc. of EGOVIS*, 2010.
[10] I. Falk, K. Sue, "What is Social Capital? A study of Interaction in a Rural Community", *Sociologia ruralis*, 2000.
[11] Application Programming Index - http://www.apdex.org
[12] P. Lops, M. Gemmis, G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends", *Recommender Systems Handbook*, 2011.
[13] Y. Mo, J. Chen, X. Xie, C. Luo, L. Yang, "Cloud-Based Mobile Multimedia Recommendation System With User Behavior Information", *IEEE Systems Journal*, 2014.
[14] O. Khalid, et al., "OmniSuggest: A Ubiquitous Cloud based Context Aware Recommendation System for Mobile Social Networks", *IEEE Trans. Service Computing*, 2013.
[15] Simulation-as-a-Service for Advanced Manufacturing - https://mozillaignite.org/apps/443
[16] Mozilla Ignite Software Lending Library - https://mozillaignite.org/apps/460
[17] VMware vCenter Chargeback Manager - http://www.vmware.com/products/vcenter-chargeback
[18] VMware View 5 PCoIP Tuner Network Optimization - https://www.vmware.com
[19] Z. Khan, D. Ludlow, R. McClatchey, A. Anjum, "An Architecture for Integrated Intelligence in Urban Management using Cloud Computing", *Journal of Cloud Computing: Advances, Systems and Applications*, 2012.