The SAHARA Model for Service Composition Across Multiple Providers

Bhaskaran Raman, Sharad Agarwal, Yan Chen, Matthew Caesar, Weidong Cui,
Per Johansson[⋆], Kevin Lai, Tal Lavian[⋆⋆], Sridhar Machiraju, Z. Morley Mao, George Porter,
Timothy Roscoe[⋆⋆⋆], Mukund Seshadri, Jimmy Shih, Keith Sklower,
Lakshminarayanan Subramanian, Takashi Suzuki, Shelley Zhuang, Anthony D. Joseph,
Randy H. Katz, Ion Stoica

University of California at Berkeley[†]

Abstract. Services are capabilities that enable applications and range from basic network connectivity to sophisticated middleware functionalities. They are of crucial importance to enable pervasive computing in next-generation networks. *Service Composition* is the construction of complex services from primitive ones; thus enabling rapid and flexible creation of new services. The presence of multiple independent service providers poses new and significant challenges. Managing trust across providers under alternative business models, and verifying behavior and performance of the components in composition become essential issues. Adapting the composed service to changing network and user dynamics by choosing service providers and instances is yet another challenge. Providers must be concerned with provisioning and placing their service instances in the network. In SAHARA, we are developing a comprehensive architecture for the creation, placement, and management of services for composition across independent providers. In this paper, we present a layered reference model for composition based on a classification of different kinds of composition. We then present a discussion of the different overarching mechanisms necessary for the successful deployment of such an architecture through a variety of case-studies involving service composition.

1 Introduction

Pervasive computing demands the all-encompassing exploitation of services inside the network. By services, we mean the glue that interconnects components of distributed applications as they function across the network. Services range from providing basic network reachability to creating overlay networks with enhanced qualities like predictable and achievable latencies and sustained bandwidths. Services also include instances of application building blocks, requiring processing and storage, judiciously placed in the network to control connection latencies and to achieve scale through load sharing. Such services may be simple format translators, interworking functions, or major subsystems for content distribution or Internet search, which are often regarded as applications in their own right. Composition via interconnection of services allows more sophisticated services and applications to be constructed hierarchically from more primitive ones. Since economics makes it unlikely that any single service provider will be able to provide all of the connectivity, applications building blocks, processing, storage, and bandwidth resources to effectively deploy a globe-spanning application, the composition of services

⋆ Ericsson Berkeley Wireless Center
⋆⋆ Nortel Networks
⋆⋆⋆ Sprint Advanced Technology Lab
† This project is supported by Sprint, Ericsson, NTTDoCoMo, HRL, and California Micro Grant #01-042.

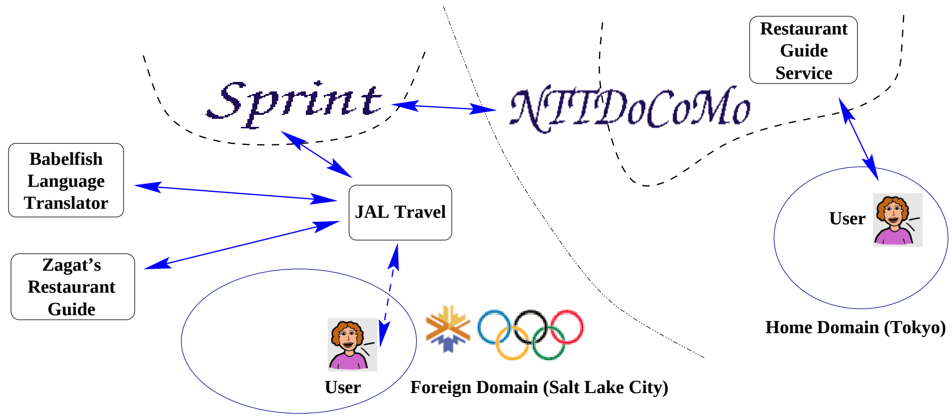


Fig. 1. Service Composition: A Multi-Provider Scenario

across independent service providers is essential. This paper proposes a comprehensive reference model for composed services in support of pervasive computing.

To illustrate our concept of service composition across service providers, consider the following scenario (Figure 1). Ms. Tanaka travels from Tokyo to Salt Lake City to attend the Winter Olympics. Her cellular provider, NTTDoCoMo, maintains roaming agreements with several foreign network operators, such as Sprint, so she can make and receive calls without any problems. But her new information appliance, built by Ericsson, is much more capable than just a phone – it is a gateway to extensive information, entertainment, and messaging services. In Japan, she subscribes to NTTDoCoMo's restaurant recommendation service, and would like to use the same application in Utah. In the U.S., Sprint has an arrangement with Zagat's Guide to present such information to its subscribers, but the text is in English and formatted for presentation on a different kind of display than Ms. Tanaka's. JAL Travel, a third party to NTTDoCoMo, Sprint, and Zagat, assembles a special new service for Japanese tourists at the Olympics: access to Zagat's Restaurant Guide, translated into Japanese, and reformatted for Japanese-style information appliance displays. Ms. Tanaka subscribes to this service for two weeks, and the usage charges appear on her NTTDoCoMo bill back in Japan.

This scenario exemplifies several key points about services composition. In next generation networks, users will demand enhanced services like restaurant recommendations, but they expect them to look and feel the same whether they are in Japan or the United States. A new service, *composed* from localized information sources, appropriate language translators, and content reformatters, makes the underlying differences transparent to the user. Such a service can be created quickly by simply connecting the components across the network. But many entities participate in the realization of this service, and they must be managed, authenticated and compensated in some way.

In the context of object-based systems, programming by composition across the network is hardly new. Yet there are critical new challenges. The first is compositions across independent service providers: Zagat owns the database and the web site, NTTDoCoMo maintains the relationship with the client and collects her monthly bill, and Sprint provides the local wireless access as well as the gateway access to local services. Creating the necessary mechanisms to support cooperative composition of services across independent service providers, each with its own authentication, authorization, and accounting mechanisms, is an essential challenge.

JAL Travel created a new recommendation service from pieces, some of which were provided by other third parties, such as Babelfish (babelfish.altavista.com), and all of which need

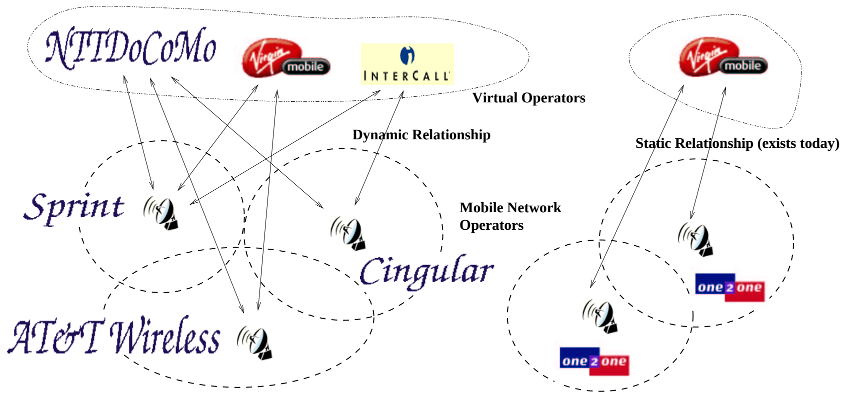


Fig. 2. Multi-provider Scenario in Wireless Connectivity Service

to run on machines interconnected across the network, spanning Internet Data Center providers and Internet Service Providers. Herein lies the second challenge: the ability for third parties to discover components and to broker new services from constituent pieces, some of which may not even be aware of the composition in which they are participating. As the qualities of a composed service are no better than its weakest component, an essential need is for brokers to be able to verify the performance and behavior of the assembled components, whether or not these underlying participants are aware of their role in compositions. If a component does not meet its performance or behavioral specification, it must be “composed out”, and a new instance from a different provider “composed in”.

A third challenge for service providers is the need for an extensive set of new service composition management tools. From a provisioning viewpoint, sufficient instances of the components need to be placed at locations within the network to ensure scalable performance and high availability even in the face of site failures or network outages. Such placement also needs to ensure appropriate network and processing latencies to achieve adequate responsiveness for the supported applications and their intended user community. Such tools include a policy management mechanism for service providers to inform service composers about how their instances in their network should be used for providing fault tolerant and load balanced behavior. A pervasive monitoring and measurement infrastructure is needed to detect changing access patterns and shifting workloads, to drive redirection to unloaded service instances or to change the number and placement of deployed instances. Network topology-awareness is important, for availability (to avoid correlated failures due to network outages) as well as performance (to achieve latency constraints and to obtain sufficient network bandwidth). Placement and connectivity issues are complicated since some service instances are anchored to fixed locations, like the Zagat web site or the Babelfish text translator, while others, such as screen reformatters, can be placed close to the user community.

Further challenges arise when we consider user dynamics. A large number of foreign roamers like Ms. Tanaka converge in Salt Lake City, yielding flash crowds and over-utilized spectrum. We need new ways of efficiently allocating resources in the context of new service provider business models. The 3rd Generation Partnership Project (3GPP) defines the concept of a Mobile Virtual Network Operator (MVNO) [13], an entity with subscribers but no network. The MVNO provides wireless connectivity service composed from the physical network resources of underlying wireless operators. In our example, NTTDoCoMo acts as a virtual operator using spectrum for its subscribers from the already established network operators such as Sprint, Cingular, and AT&T Wireless. We view such a multi-provider relationship as just another case of

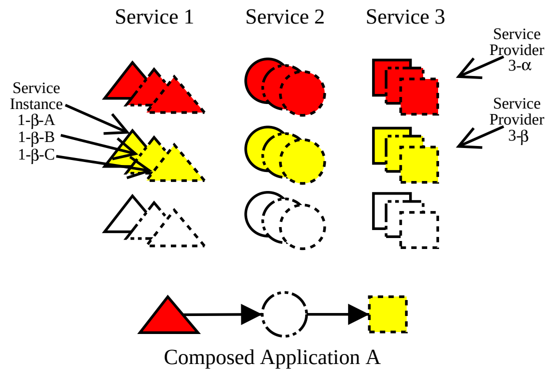


Fig. 3. Service Composition: Choices along Three Dimensions

service composition. Today, the relationships between the virtual operator and the Mobile Network Operator (MNO) are static and negotiated long in advance (e.g., between *Virgin Mobile* and *One2One* in UK [13], see Fig. 2). But this is inefficient when user dynamics are considered; and we expect much more dynamic formation and dissolution of relationships in the future.

Thus, as a fourth challenge, we address issues in efficient resource allocation across providers, considering the dynamics of user communities. In our MVNO scenario, this translates to dynamically selecting ("roaming") among co-located MNOs. Figure 2 illustrates such dynamic allocations in the MVNO context. We envision that dynamic relationships will last for short time-scales of minutes to hours, thus allowing for load balancing, and efficient resource usage. Dynamic, utility-based resource allocation amongst competing providers is an important challenge in service composition.

Our overall goal is to define *a comprehensive reference model able to describe the assembly from components of end-to-end services with desirable, predictable, enforceable properties, yet spanning potentially uncooperating service providers*. We are developing these concepts in the context of the SAHARA[1] project, which is also the name of our prototype architecture and system. The rest of this paper is organized as follows. The next section (Section 2) summarizes the discussion above with the technical issues in composition. Section 3 presents a classification of the different kinds of composition that we consider. Subsequently, we present a layered reference model for composition in Section 4. Section 5 describes the different mechanisms we employ to address the issues in the different models of composition. Section 6 discusses related work. We conclude in Section 7. Appendix A gives a glossary of SAHARA terminology.

2 Technical Issues

The scenario in our prior section can be understood along three dimensions (Figure 3) in the choices in service composition: (a) what set of services to use for composition, (b) which service providers' resources to use, and (c) which instances of each service to use for a particular client session. In addition, we also have the issue of who makes these decisions.

Considering these three dimensions, the technical issues that must be addressed in a reference model for service composition are:

– *Trust management and behavior verification:* When multiple providers interact, it is important to establish mutual trust. This is not only for the purpose of user authentication

[1] SAHARA: Service Architecture for Heterogeneous Access, Resources, and Applications

and billing, but also to verify the behavior of the components in composition. Does a component meet its promises in terms of functionality, protocols, performance, availability, or other properties?

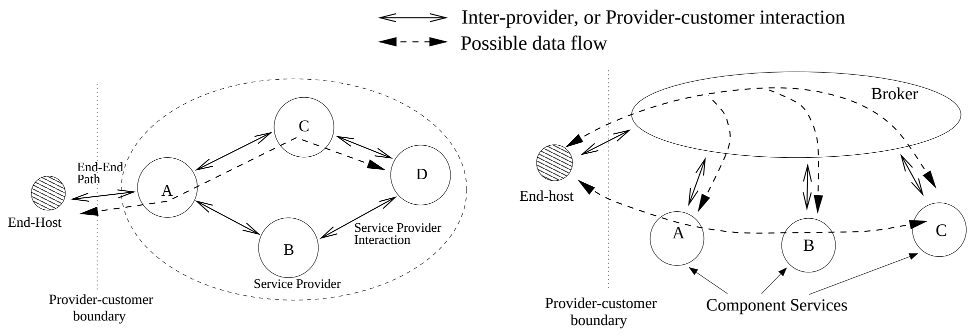
– *Adapting to network dynamics:* Where network dynamics are involved, workloads can shift, performance diminishing congestion can arise, and reachability to service instances can be lost. This problem implies the need for an accurate performance monitoring, modeling and prediction, combined with a performance-sensitive choice of appropriate service providers and service instances. Furthermore, such selections may change during client sessions. We term this the *service selection* problem.
– *Adapting to user dynamics:* When user dynamics is highly variable, different providers face varying demand for physical resources from their users. Allocation of physical resources (spectrum, network bandwidth, CPU, etc.) across providers, based on current demand is important to achieve fair/utility-driven resource allocation.
– *Resource Provisioning and Management:* For a given community of users and a given set of performance, availability, and administrative constraints, how many instances of a service are needed? How can this be optimized given a knowledge of the provisioning done by other service providers? Network topology will have a significant impact on the performance and availability of services. Given this issue, where should the instances be placed in the network? We call this the *service placement* problem. This problem needs to be addressed by a service provider when deploying a service. Also, how can service providers enforce their local policies of fault-tolerant and performance sensitive use of their service instances in a composition performed by a third party? We term this the *policy management* issue.
– *Interoperability across multiple service providers:* When composing services across providers, we have to deal with heterogeneity in protocols and data formats, as well as authentication and authorization mechanisms.

3 Service Composition Models

In this section, we classify service composition into two models and analyze their pros and cons. Our classification is based on the type of interaction between composed component service providers.

– **Cooperative Model:** Service providers interact in a distributed fashion, with distributed responsibility, to provide an end-to-end composed service.
– **Brokered Model:** A single provider, the broker, uses the functionalities provided by underlying service providers and encapsulates these to compose the end-to-end service.

In either case, the end-user subscribes to only one provider. However, the difference lies in the way the responsibility for the composed service is apportioned. The two possibilities represent different business models for composition. In the cooperative model, the properties of the composed service such as functionality, performance, and availability, are guaranteed by the design of the distributed interaction, and through service-level agreements between the interacting entities. Each service provider is only responsible for providing guarantees for the portion of the composed service within its domain. In the brokered model, the broker, assumes responsibility for the properties of the composed service. We can imagine a broker entering into contracts with service providers, and using these to construct end-to-end composed services. The broker verifies the functionality of the individual pieces in the service path. This is because individual component providers may not trust each other – they may limit the information about the state of their service they expose to the other providers in the composition, or may actively seek to cheat on the quality of service they provide.



(a) Cooperative Model of Composition (b) Brokered Model of Composition

Fig. 4. Service Composition Models

An example of cooperative composition is cellular roaming as a service, composed from the resources of multiple mobile network operators, as in Section 1. The distributed interaction between network operators (NTTDoCoMo and Sprint) enables roaming. Another example is end-to-end connectivity service in the Internet. This is provided through distributed interaction between the different autonomous systems (ASes). Cooperation between domains is assumed in the inter-domain routing protocol (BGP [23]). In these examples, there are long-term, static, negotiated contracts between the participant providers and there is actual value exchange outside the technical architecture.

An example of brokered composition is the restaurant guide service assembled by JAL Travel in our earlier scenario. It composes components such as Zagat's restaurant guide and Babelfish's language translator. It assumes responsibility for the functionality and performance of the composed service. Another example is the Yahoo portal service that composes third party services such as the Google search engine (www.google.com), Stock ticker, News, etc.

Figure 4 illustrates these alternate composition models. The models of composition say nothing about the data flow, only the nature of interaction, or the business model, between providers. As an example, in the brokered model, we could have data flow through the broker, who assembles it (shown by the dotted line on the top). Or, the broker could set up the data exchange and not be in the data path (shown by the dotted lines at the bottom of the figure).

The same composed service could be implemented in either model. Consider the provision of a connectivity service with QoS guarantees between two points on the Internet. In a cooperative model, ISPs enter into service level agreements (SLAs) that specify the mutual QoS guarantees. These may be stitched together in a distributed fashion using bandwidth brokers to offer end-to-end guarantees [3]. In a brokered model, a service provider like InterNAP (www.internap.com) purchases pipes with specified guarantees from individual ISPs and uses them to provide QoS to its customers. (Note the difference in our usage of the term "broker" – in [3] a broker only acts between a pair of peering ASes, whereas in our case, the broker is a separate provider entity that interacts with several providers).

Each of these models is suited to a particular environment. In the cooperative model, providers are only responsible for their portion of the composition, not the entire end-to-end service path. Since they work together, and can share performance information, they can more easily ensure end-to-end properties of the composed service. However, they must rely on each other, which leads to issues of trust. Since the responsibility for the composed service is distributed, each provider must continuously verify that the others with whom it has agreements meet their service specifications. These specifications are in terms of functionality, protocol, performance, and

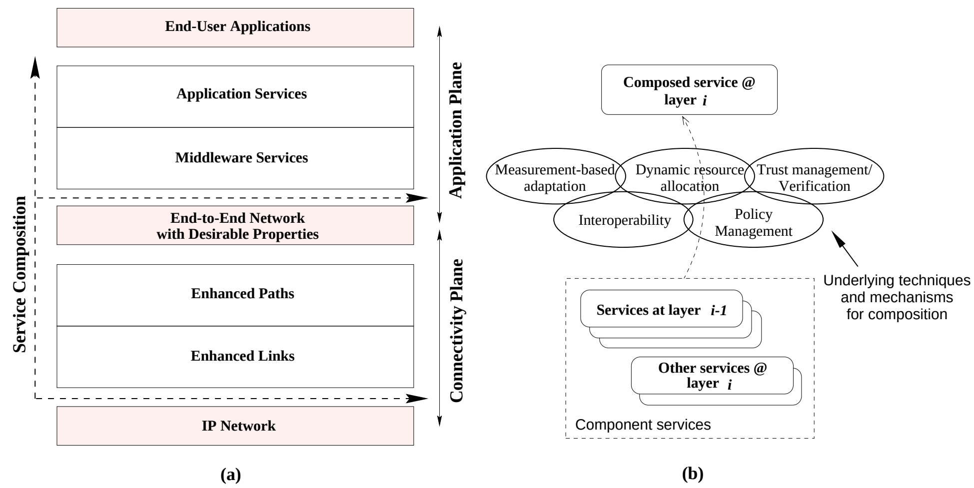


Fig. 5. A layered reference model for Service Composition

availability. Such comprehensive verification is absent in the cooperative composition of Internet connectivity service across ASes today. Malicious or mis-configured BGP route advertisements result in poor paths or loss of Internet reachability.

In the brokered model, the broker composes an end-to-end service by selecting individual services residing in different domains. This simplifies service deployment since the members of the composition need not agree among themselves, only with the broker. This also enables the composition of services across competing service providers. The broker can change the set of service providers it has chosen for composition. In exchange for this ability, the broker assumes responsibility for constructing the entire end-to-end service path. Brokering is a powerful tool to construct services from providers who are not necessarily aware that they are participating in a larger end-to-end service (e.g., in Figure 1, Zagat's restaurant guide service does not know it is being composed with Babelfish's language translation service). However, because the broker has limited visibility into the underlying provider resources, sub-optimal utilization of provider resources may result. The nature of the composition and the relationships of the underlying service providers will determine which model is most appropriate.

4 Service Composition: A Layered Reference Model

The previous section classified service composition by how responsibility is shared across providers. Here we present a layered reference model for composition across different layers of concern.

Composed services build on top of commodity connectivity provided by the IP layer, the bottom slice in Figure 5(a). We assume that IP provides reachability, and build several "desirable" properties in the *connectivity plane*. These include features such as performance guarantees (e.g., latency, bandwidth, loss-rate bounds), availability guarantees (e.g., available 99.99% of the time), as well as functionality or protocol guarantees (e.g. guaranteeing that a particular advertised Internet route is valid). We achieve them through composition at the connectivity plane, across multiple providers. This results in an "end-to-end network with desirable properties", as shown in the middle slice of the figure.

The connectivity plane is further divided into two layers: *Enhanced links*, and *Enhanced paths*. The enhanced links abstraction is built between two interacting entities – between two service providers, or between a provider and the end-user. This abstraction achieves desirable performance oriented properties, and verification of routing protocols between peering entities. Verification checks if the routes advertised by a peer are valid. An example of this composition is of the MVNO choosing between multiple MNOs, described in Section 1. Here the enhancement is improved performance through reduced call blocking rate, due to dynamic load sharing across multiple network operators [2].

Enhanced paths build on enhanced links, and provide desirable properties in an end-to-end path between points on the Internet. The path spans two or more service domains. Paths can be chosen adaptively via resource allocation across providers at this layer to meet performance or availability constraints.

Our reference model is independent of whether the performance guarantees are strict, or simply "enhancements" to the best-effort Internet. Alternative enhancements might be appropriate for different end applications.

The top half of the figure represents the application plane. These layers support end-user applications and are in turn supported by the end-to-end network. The two layers at the application plane build on top of the end-to-end network. The *middleware services* are enablers, thus they are not useful to the end-users. An example is the Babelfish language translation service in the scenario in Figure 1. Other examples include content-distribution networks, video/audio transcoders, databases, e-commerce components, etc. The *application services* layer consists of services useful to end-users. Zagat's restaurant guide in the scenario in Figure 1 is one such example. Other examples are: a search engine, a voice-mail service, or an e-commerce site. Composition at the application plane results in enhanced functionality. In our example, the enhanced functionality is that of the restaurant guide appearing in the user's native language (Japanese) and presentation style (NTTDoCoMo) in a foreign network.

Service composition can be applied within and across these layers, as illustrated in Figure 5(a). Figure 5(b) shows this more explicitly. A composed service at a higher layer is composed of multiple services in the same layer, or of services at the layers below. In Figure 1, composition takes place across the application (restaurant guide), middleware (language translation), and connectivity plane (roaming service enabled by Sprint) layers. In Figure 2, composition is at the enhanced link layer, using the component connectivity services offered by co-located MNOs.

We note that this layerization is only a reference model, and we can have compositions that do not strictly adhere to it. Some application services can be composed directly of enhanced links, without using the enhanced paths abstraction or middleware services. This will occur typically in performance sensitive applications, where the composer needs full visibility via a flat rather than a hierarchical composition. Figure 5(b) shows the different techniques and mechanisms that are used to enable composition. We discuss these next.

5 Mechanisms for Service Composition

The issues we listed in Section 2 appear in different flavors in the alternate models of composition. SAHARA (Service Architecture for Heterogeneous Access, Resources, and Applications) is our architectural prototype to explore the mechanisms required to address these issues. To understand the various mechanisms required, we are working on several specific case studies of service composition. These cover the dimensions of classifications in Sections 3 and 4. In this section, we describe the various mechanisms that we are designing to address the challenges we presented earlier. We also describe how each mechanism is used in the individual case studies.

[2] Although it is not obvious here how this example builds on IP technology, it will be clear after a discussion of the mechanisms for resource allocation, in Section 5.

The various mechanisms are summarized in Figure 5(b). We have performed in-depth evaluations of several of these mechanisms. Due to space limitations, we summarize the evaluations of only a subset of these mechanisms (Table 1).

Scenario	Result	Section
Availability of composed application service in the presence of Internet path failures	Failure detection and recovery within 3 seconds	5.1
Use of congestion-based pricing to allocate VoIP gateway resources	**Need text from Matt**	5.2
Inferring inter-AS relationship using BGP from multiple vantage-points	Able to infer over 99% of inter-AS relationships	5.4

Table 1. Summary of evaluation results for a subset of the mechanisms

5.1 Measurement-based adaptation

For service composition, it is often desirable to dynamically choose service providers, and service instances based on current network and server loads. Measurements can be carried out by a third party measurement service – a common element of the service infrastructure, or by the composer itself. This applies to both the cooperative and brokered models of composition. We now describe two services that we are developing involving measurement-based adaptation.

We are developing a general end-to-end Internet host distance monitoring and estimation service. Such a service is especially useful in a brokered composition because the broker does not have an insight into the network characteristics of individual service providers. Given the potential large numbers of service providers and instances, to scale the measurement service, we cluster the end hosts to be monitored based on the similarity of their perceived distance to the measurement points. The cluster center is then used as a single measurement target for future monitoring. Simulations with real Internet measurement data show that our scheme has good prediction accuracy and stability with a small communication and computation cost [11].

An application service that we have developed is the Universal Inbox [22] – a metaphor for any-to-any communication across heterogeneous devices and networks. Data transformation services such as audio/video transcoders and the conversions of text $\leftrightarrow$ speech are extensively used to adapt content between communicating devices. For example, in our earlier scenario, Ms. Tanaka's email service could be composed with a text-to-speech conversion service so that she can listen to emails over her cellular-phone. We use the brokered model of composition here. To adaptively choose the service providers and service instances for such composition, we design a middleware measurement layer [21] that exchanges network and server load using a link-state algorithm. This performance exchange takes place across service execution platforms, enabling a dynamic choice of service instances, possibly in the middle of the user sessions, to hide network and server failures from end-users.

A critical challenge that we address in the context of such an application is *availability*. When composed services span multiple providers, data could traverse the wide-area Internet. We detect and recover from Internet path failures *quickly* by using the middleware measurement layer to choose alternate service instances for the client session. Our measurements in [21] show that Internet path failures that happen due to congestion or other factors can be detected reliably within about *2 seconds*. Further, subsequent recovery by using alternate service instances can be completed in a *few hundred milliseconds*. Thus, network path failures lasting several tens of seconds to minutes [15] can be completely masked from the end client (Table 1, row 1).

In addition to these two services, we have also designed and implemented a measurement methodology [16] to improve DNS-based server selection, which is a common technique used by Content Distribution Networks (CDNs) today. Our technique enables the collection of client to local DNS server mappings to enable more accurate server selection based on a client's local DNS server IP address. Understanding the distribution of HTTP requests corresponding to local DNS servers also enable better load prediction given a DNS request and thus improved server selection mechanisms.

5.2 Utility-based Resource Allocation Mechanisms

In a multi-provider environment, different providers may experience different demands for resources due to user dynamics. Demand or utility-based resource allocation can be applied within a service provider to manage its instances. In the brokered model, it can be used to allocate resources across providers. In SAHARA, we are exploring two resource allocation mechanisms.

Auctions are one way of constructing a marketplace where a resource, such as bandwidth or physical spectrum, can be dynamically allocated. Auctions allocate resources to consumers based on their bids, which represent the value of the good to them. Furthermore, the resource can be subdivided into units, and multiple bidders can be allocated the resource until the resource pool is exhausted. Auctions can occur in rounds, where the allocation determined by each round can be for some future time period. Auction-based allocations in a multi-provider environment can provide the mechanism for demand-based resource allocation. For instance, in the MVNO scenario in Section 1, spectrum resources could be auctioned off every few minutes to competing virtual operators based on their current user-load in the area of coverage.

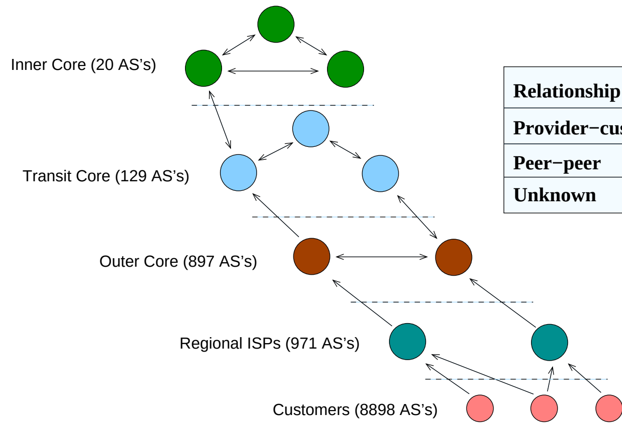
Congestion pricing is an allocation mechanism that assigns scarce resources to consumers using the abstraction of "price" as a means to moderate demand. During high demand, such a market ensures that the price increases. Only those consumers with the greatest need and having sufficient currency, will obtain the needed resource. During low demand, the price drops, and access to the resource with be cheap and plentiful. This approach should yield an assignment of resources (supply) to the need (demand) that adapts to instantaneous demands [25].

As an application that uses the congestion pricing mechanism, we have looked at the selection of Voice-over-IP (VoIP) gateways across multiple providers [7]. These gateways are deployed by independent entities, and exchange dynamic pricing information based on the TRIP protocol [24], as well as the peering relationships between the provider entities. The pricing is decided based on the load, or congestion, at each gateway; and the user gets to choose between several gateways based on the price and the required quality of service. This achieves pricing-based load-sharing among the gateways. In [8], we look at how the price-congestion curve can be best decided to better utilize provider resources and maximize provider revenue. We find that... **Need Matt's summary here** (Table 1, row 2).

5.3 Trust Management and Verification of Service/Usage

An important issue for service composition is the establishment and monitoring of trust relationships between inherently untrusting entities. This is important in cooperative composition where services providers have pairwise service agreements between them. This is also important in brokered composition where the agreements are with the broker. Typically a AAA (*Authentication, Authorization and Accounting*) server governs service instances and users within one administrative domain. However, in our scenario, we need to compose services across multiple domains governed by multiple, different AAA servers.

We are investigating an authorization control scheme with credential transformations to enable cross-domain service invocation [27]. Federated administrative domains form credential transformation rules based on established peering agreements. These are used by a AAA server



Relationship	# AS pairs	Percentage
Provider−customer	22,621	94.51%
Peer−peer	1,136	4.75%
Unknown	178	0.74%

Fig. 6. Internet AS Hierarchy and Inferred Relationships for 23,935 AS Pairs

to make authorization decisions for a service request from an affiliated domain. This liberates service providers from preparing authorization rules for each affiliated domain.

Another important issue in service composition is to verify whether the provided service adheres to the desirable properties advertised by its provider. Such properties can be specified in a bilateral Service Level Agreement (SLA) between provider and requester. We use parameter verification and usage monitoring as mechanisms to ensure that the properties specified in the SLA are being honored. For instance, in a case-study of connectivity composition across domains, we have border routers monitoring control traffic from different providers to detect malicious route advertisements.

5.4 Policy Management

An advantage of cooperative composition is that each provider has visibility into its network of services, while a broker does not. The disadvantage of this distributed form of composition is the lack of central control over the composition that the broker enjoys. This disadvantage can be minimized if some form of distributed policy management is in place. Specifically, in cooperative composition, the service composition policies of one service provider can be made visible to and applied at distant service providers further along the composed path. Such policies may include which service instances are for primary use and which are solely for use in various failure modes, and policies that govern load balancing between instances.

A case study involving this principle is the distributed application of policies in inter-domain routing on the Internet. BGP [23] was not designed to allow local policies to be imposed at distant points in the network. The Internet infrastructure has been plagued of late with pathological routes that attempt to impose such inter-domain routing policies by tricking distant service providers' route selections [14]. To address this issue, we are developing an architecture that allows distributed policy management between service providers, thereby allowing load balancing, performance and failure mode policies to be applied at various points in the network. By using an AS relationship topology map of the Internet and a measurement infrastructure, a policy agent for an AS can interact with distant policy agents to negotiate policy changes that will impact its incoming traffic [6].

A crucial piece of this architecture we have built so far is the mechanism to build an AS relationship map (Table 1, row 3). In [26], we use BGP measurements from multiple vantage points to build such a relation map. The map indicates the nature of inter-provider relationship that exists between neighbor ASes: peer-to-peer or customer-provider. Figure 6 shows a summary of the results of this study.

5.5 Interoperability through Transformation

Interoperability is important for service composition across different service providers. Data and protocols formats may need to be transformed for interoperability across heterogeneity. This is an important issue for composition in the connectivity plane and in the application plane where services have more complex interfaces. Defining the service interface and propagating it are two key challenges here. We have investigated one such interoperability service at the connectivity plane. In our *broadcast federation* work [9], a global multicast service is composed from the multicast implementations in different provider domains. We use protocol transformation gateways between administrative domains that have non-interoperable implementations of multicast.

5.6 Service Deployment

In any composition model, a service provider has to decide how many instances to place and where to place them. In SAHARA, we are exploring two mechanisms to enable such decisions.

First, we infer inter-provider relationships between Internet Autonomous Systems by passively monitoring inter-domain routes from multiple locations [26]. This allows us to understand traffic flows in the underlying network topology – inter-provider relations give more information about reachability between points in the network, than a simple graph of peering autonomous systems. This richer reachability information helps in service instance placement – the service provider can use the information to ensure that there are sufficient instances with good reachability from various points on the Internet.

Next, we have designed a dynamic replica placement protocol that enables an application level multicast tree for Web content distribution. This can meet client QoS constraints and server capacity constraints, while retaining *efficient* and *balanced* resource consumption [10]. To provide a scalable solution, we aggregate the access patterns of clients that are topologically close in the network. We use incremental clustering and distribution based on client access patterns to adaptively add new documents and purge old documents from the content clusters [12].

6 Related Work

Related work falls into two categories: (a) Architectures for seamless integration of devices and services, and (b) Internet-based web-services initiatives.

Architectures for seamless device/service integration: The UMTS model [17] admits of a sophisticated accounting, billing, and settlements architecture to support third-party brokering between subscriber needs for service and multiple service providers. However, there is no explicit consideration of where service provision and service mediation should exist in this architecture, other than in the core network that ties together various access networks.

The Virtual Home Environment (VHE) concept of IMT2000 [20] permits users to roam away from home, seeing the same service interface (service mobility). Nokia's MITA [19] also considers service mobility: seamless and network environment independent access to services such as rich calls (multimedia). ICEBERG [29] looks at extensible personal and service mobility. Nevertheless, these efforts do not consider composition of services across multiple providers, efficiency through network awareness, or resource allocation issues.

TINA [5] is a CORBA-based [18] service architecture. The TINA reference architecture contributes the conceptual separation of the business model, the informational model, and the computational model. Its three layer model of applications, distributed processing, and network environment has influenced our layerization of composition in SAHARA. Key differences are that SAHARA adds elements of composition across heterogeneous providers, with a greater awareness and management of the underlying network topology. SAHARA also considers resource management via placement, allocation, redirection to services and resources.

Internet-based Web-Services Initiatives: There are several industrial initiatives to enable web-services (WS) which "integrate PCs, other devices, databases, and networks into one virtual computing fabric that users could work with via browsers" [28]. These include HP's e-speak (www.e-speak.net) and web-services [1] platform, Microsoft's .NET [4], and Sun ONE [2]. These are based on a language for description of web services (WSDL), a common wire format for these descriptions (SOAP), and a registry to support service location (UDDI). Microsoft's .NET also defines a language independent software platform for easy and secure interoperation of applications [28]. However, these do not define a wide-area service architecture; they are complementary to SAHARA's goals of service placement, resource allocation, and network awareness aspects.

7 Conclusions

In this paper, we have presented a vision of distributed systems composed from services placed in the wide-area Internet, and spanning service providers at different levels. SAHARA is our evolving architectural prototype for the creation, placement, and management of services in next generation networks. Our goal is to enable end-to-end service composition with desirable, predictable and enforceable properties spanning multiple potentially distrusting service providers. We investigate two forms of service composition under different business models with varying degrees of cooperation and trust among providers. We classify component services and composed services into a layered hierarchy. The overarching themes in the various techniques and mechanisms that we use for composition include (a) measurement-based adaptation through dynamic choice among service providers and service instances, (b) utility-based resource allocation for demand-driven load sharing across provider resources, and (c) a trust-but-verify approach to management of trust and behavior verification when multiple providers interact to provide a composed service. We continue to develop these mechanisms through prototype distributed applications spanning wide-area networks, and constructed through service composition at various layers under different models.

References

[1] HP Web Services. http://www.hpmiddleware.com/products/hp_web_services/default.htm.

[2] Implementing Services on Demand and the Sun Open Net Environment (Sun ONE). http://www.sun.com/software/sunone/wp-arch/.

[3] Internet2-QBone Bandwidth Broker. http://www.merit.edu/working.groups/i2-qbone-bb/.

[4] Microsoft .NET. http://www.microsoft.com/net/.

[5] Telecommunications Information Networking Architecture Consortium. http://www.tinac.com/.

[6] S. Agarwal, C. Chuah, and R. H. Katz. An Overlay Policy Protocol to Augment BGP. Work in progress.

[7] M. Caesar, S. Balaraman, and D. Ghosal. A Comparative Study of Pricing Strategies for IP Telephony. In *GLOBECOM*, Nov 2000.

[8] M. Caesar, D. Ghosal, and R. H. Katz. Resource Management in IP Telephony Networks. In *International Workshop on Quality of Service*, 2002.

[9] Y. Chawathe and M. Seshadri. Broadcast Federation: An Application-layer Broadcast Internetwork. In *Intnl. Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2002.

[10] Y. Chen, R. H. Katz, and J. Kubiatowicz. Dynamic Replica Placement for Scalable Content Delivery. In *International Workshop on Peer-to-Peer Systems*, March 2002.

[11] Y. Chen, K. Lim, C. Overton, and R. H. Katz. On the Stability of Network Distance Estimation. Proceeding of ACM SIGMETRICS Practical Aspects of Performance Analysis Workshop (PAPA 2002) . Also to appear CMG Journal of Computer Resource Management , Spring Edition, 2002.

[12] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. H. Katz. On the Clustering of Web Content for Efficient Replication. Submitted for publication.

[13] F. Curley. Mobile Virtual Network Operators (Part IV). In *Eurescom Summit, 3G Technologies and Applications*, Nov 2001.

[14] Geoff Huston. Analyzing the Internet's BGP Routing Table. Cisco Internet Protocol Journal, March 2001.

[15] C. Labovitz and et.al. An Experimental Study of Delayed Internet Routing Convergence. In *Computer Communication Review, ACM SIGCOMM'00*, Aug/Sep 2000.

[16] Z. Mao, C. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers. In *USENIX Annual Technical Conference*, 2002.

[17] W. Mohr and W. Konhauser. Access Network Evolution Beyond Third Generation Mobile Communications. *IEEE Communications Magazine*, Dec 2000.

[18] T. J. Mowbray and R. Zahavi. *The Essential CORBA: System Integration Using Distributed Objects*. Wiley Computer Pub., 1997.

[19] NOKIA. *Inside MITA: Mobile Internet Technical Architecture*. IT Press, 2001.

[20] R. Pandya and et.al. IMT-2000 Standards: Network Aspects. *IEEE Personal Communications Magazine*, Aug 1997.

[21] B. Raman and R. H. Katz. Emulation-based Evaluation of an Architecture for Wide-Area Service Composition. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2002.

[22] B. Raman, R. H. Katz, and A. D. Joseph. Universal Inbox: Providing Extensible Personal Mobility and Service Mobility in an Integrated Communication Network. In *WMSCA'00*, Dec 2000.

[23] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4), Request for Comments: 1771*, Mar 1995.

[24] J. Rosenberg, H. Salama, and M. Squire. *Telephony Routing over IP*. Request for Comments (RFC) 3219, Jan 2002.

[25] J. Shih, R. H. Katz, and A. D. Joseph. Pricing Experiments for a Computer-Telephony-Service Usage Allocation. In *Proc. IEEE GLOBECOM*, November 2001.

[26] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy From Multiple Vantage Points. In *Proc. IEEE INFOCOM*, 2002.

[27] T. Suzuki and R. H. Katz. An Authorization Control Framework to Enable Service Composition Across Domains. In *ACM WWW2002 (poster)*, May 2002.

[28] S. J. Vaughan-Nichols. Web Services: Beyond the Hype. *IEEE Computer*, Feb 2002.

[29] H. J. Wang, B. Raman, C-N. Chuah, and et.al. ICEBERG: An Internet-core Network Architecture for Integrated Communications. *IEEE Personal Communications Magazine*, Aug 2000.

A Glossary of SAHARA Terminology

Service: a functionality provided by an autonomous entity (a business, a corporation, an institution, or other organization) to its clients.

Connectivity Service: a service that provides (possibly enhanced) connectivity between 2 or more points in the Internet. E.g. multicast delivery, QoS-assured connectivity, best-effort connectivity.

Middleware Service: a service that is an enabling function for other application or middleware services. E.g. content caches, data translation services, measurement services.

Application Service: a service visible to the end-user. E.g. live video broadcast, Internet telephony.

Service Instance: In the case when the same service functionality is available at different physical points in the Internet, each of these points is referred to as an instance of a service. E.g. Web-mirror sites.

Service Provider: The entity that owns service instances, and is responsible for making available each of the types of service specified above. Also typically the entity who gets paid for the service usage. Three types are possible: Application Service Provider (ASP), Middleware Service Provider, and Internet Service Provider (ISP).

Service composition: putting together primitive, component services from possibly different, competing, mutually-untrusting service providers, to form a performance and/or functionality enhanced, *composed service*.