A Service-Oriented Network Infrastructure for Crossover Service Ecosystems

Bangpeng Zheng
Zhejiang University

Jianwei Yin
Zhejiang University

Shuiguang Deng
Zhejiang University

Zhaohui Wu
Zhejiang University

Schahram Dustdar
Technische Universität Wien

***Abstract*—Crossover service networks have become a trend promoting cross-enterprise, -field, and -industry cooperation. As an innovation process for enterprises, crossover services can provide users creative, novel, and amalgamated services by breaking traditional organization, business, and domain boundaries. However, business and interface inconsistencies in today's crossover service ecosystems make crossover cooperation difficult and time-consuming. We designed a crossover service ecosystem-oriented network that provides a sound framework for efficiently deploying, publishing, discovering, composing, monitoring, and optimizing access on crossover services to fully deliver the potential of them. Furthermore, a prototype of the service network, namely JTangYdrail, is introduced, where service switches and service routers are designed to provide the transparency between the complex crossover service environment and the convenient user service access.**

Digital Object Identifier 10.1109/MIC.2020.2972121
Date of current version 25 March 2020.

The open business model has grown rapidly with the development of the Internet. The API economy is defined by IBM as "speed to market, reaching new markets/customers, innovation, and improved sharing of assets across the enterprise/domains."[1] Organizations across all

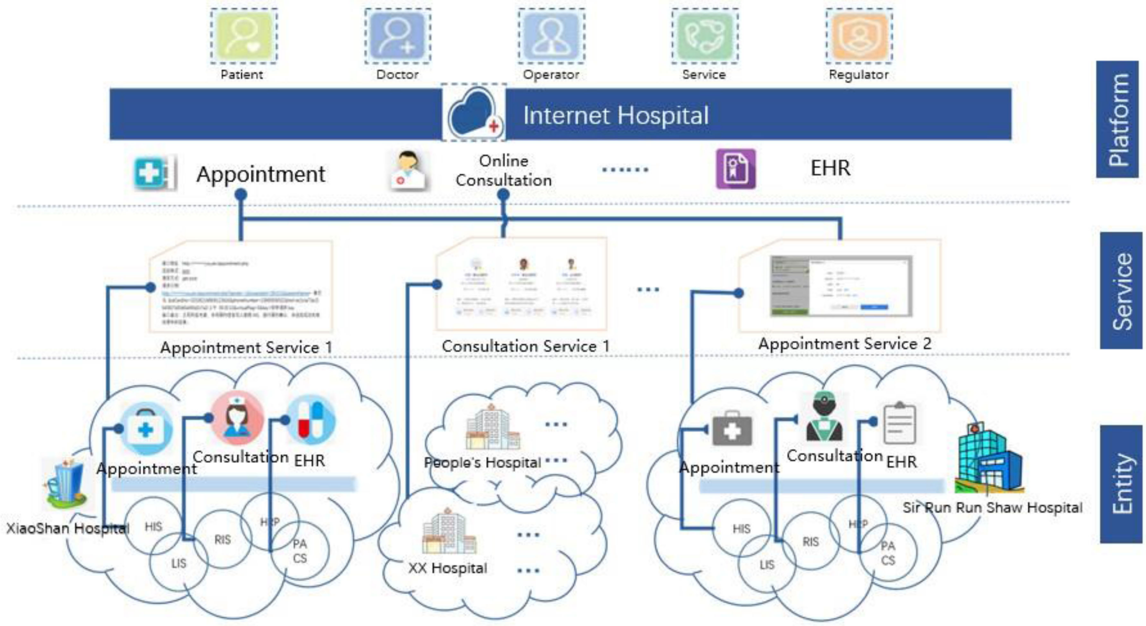


Figure 1. Service Integration in E-health. Different services (service layer) from different hospitals (entity layer) have different formats. Internet hospital (platform layer) collects all services and provides a portal for patients.

spectra have already moved businesses onto the Web, which has brought about a fast growth of various Web services. However, most of the services are in an isolated status of the "service island" and the function of services from a single field cannot meet the complex requirements of users.[2] With the fusion of the Internet and traditional industries, crossover has become a trend to promote the cross-enterprise, -field, and -industry cooperation, and the creation of innovative products, markets, and business models.[3] As an innovation process for enterprises, crossover service can provide users creative, novel, and amalgamated services by breaking traditional organization, business, and domain boundaries.[4] Crossover services can create value that a single-domain service cannot provide, thus achieving the value emergence effect of "$1 + 1 > 2$."[5] Crossover services are proliferating and being widely used in several realistic scenarios, such as e-commerce, e-health, e-finance, amongst others.

However, inconsistencies in business offerings and interfaces make service cooperation challenging and time-consuming in crossover service ecosystems. Using the example of e-health, as Figure 1 shows, the Internet hospital ecosystem consists of three layers: entity, service, and platform. The first layer contains all the entities providing various services. In the Internet hospital, the entity layer is mainly composed of medical institutions, such as hospitals, pharmacies, and logistics. The second layer contains all the services provided by the entities, each of which has the ability to fulfill a task in a particular field or enterprise. However, these services from different institutions usually have different formats. The third layer, platform, collects all the services and arranges these services according to business processes of hospitals to meet the needs of users.

Assume John, a professor who had a bad headache, needs to visit to the hospital tomorrow. He decided to make an appointment in the department of neurology of The Seventh People's Hospital, which is nearest from the campus, by using the online-appointment service from Wuzhen Internet Hospital, the first Internet hospital in China. However, The Seventh People's Hospital is not providing the online-appointment service on its web-based platform. In fact, there are only 17 online-

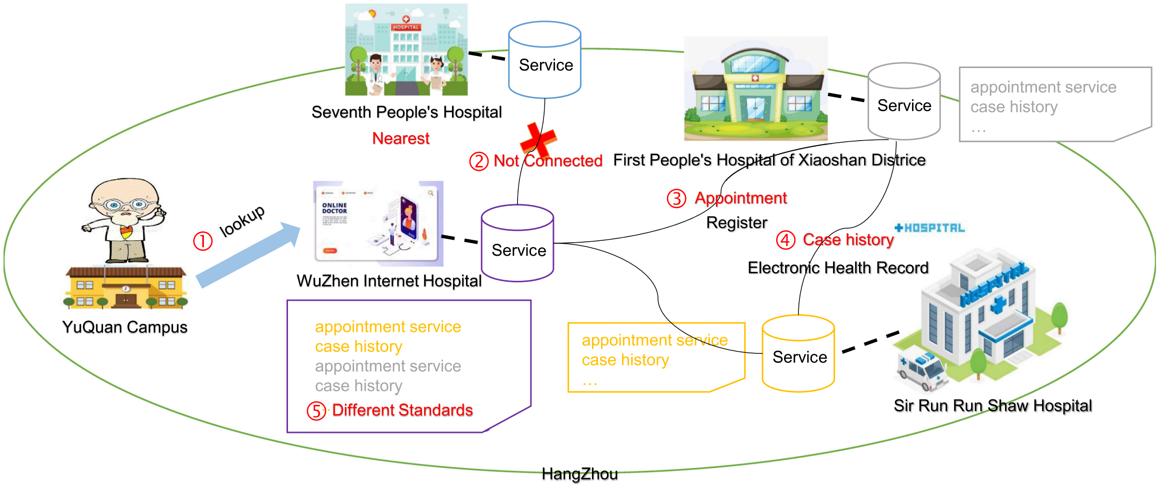


Figure 2. Medical Example. There are very few hospitals connected to Wuzhen Internet Hospital in Hangzhou, China. Different colors mean different services from different platforms. Different languages, APIs, and styles might cause trouble while using these services from users' view.

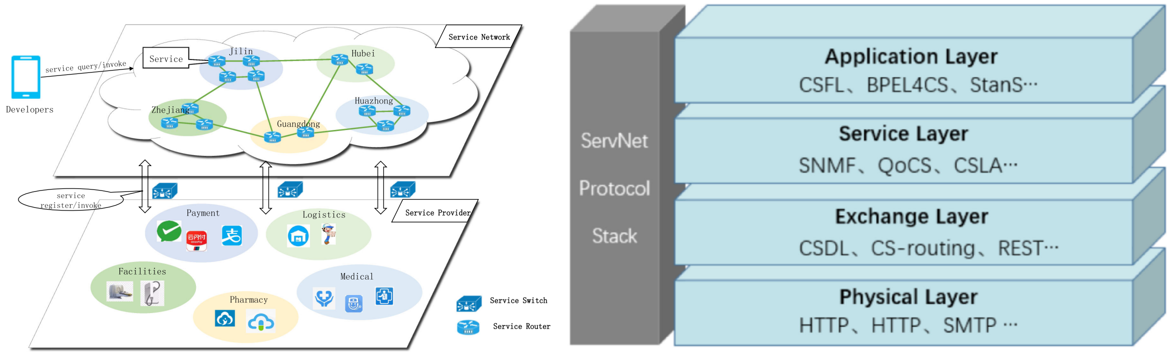
appointment-support hospitals in Hangzhou connected to the platform. Hence, John can only choose one of the 17 hospitals. For example, the First People's Hospital of Xiaoshan District of Hangzhou. Then, John should provide his ID number and phone number to make an appointment. The next day John went to the hospital and visited the doctor, but his condition and medical history was in another Hospital, Sir Run Run Shaw Hospital. He had to retire the medical record by using the case history service from Sir Run Run Shaw, which required a refill of information in another platform. It is quite troublesome while these two hospitals have their own systems.

As shown in Figure 2, there are several challenges in such a crossover service ecosystem. a) *Environmental heterogeneity*, because there are a large number of services that come from different platforms, languages and communication styles. Such dynamic environments increase the pressure on deployment in a crossover service ecosystem. b) *Distributed separately*. Services are widely distributed and it is complicated for developers to integrate services from different enterprises. c) *Service heterogeneity*. There are services, which have the same or similar functions but different operations. For service consumers, it is difficult to find the suitable services. It is quite troublesome to choose the suitable services and refill the required information again and again.

In this article, a network infrastructure named JTangYdrail (JTang is an open source community supported by Zhejiang University, and Ydrail is from old Norse Yggdrasil, an immense mythical tree that plays a central role connecting nine worlds, which means the bridge of all services) is proposed to cater to the above problems. To deploy and manage crossover services, we put forward a service network model, and design an approach combining hardware with software, which compose of two novel service infrastructures: service switch and service router. As for services that have the similar functions, standardized service is proposed to represent the multiple services by mapping function in this system. So, consumers can use services free from confusion, which initially caused by different registries. Besides, an optimal service extraction method named service wrapper is proposed to support efficient service discovering.

JTANGYDRAIL: SERVICE NETWORK MODEL AND INFRASTRUCTURE

In this section, we first introduce how service network works in the perspective of model and architecture, then the service infrastructures and key techniques.



(a) (b)

Figure 3. (a) Conceptual architecture of ServNet. (b) Protocol stack of ServNet

Service Network Model

In crossover service ecosystems, services will be connected to each other and produce the service network (ServNet). As an overlay network, ServNet plays the role of broker in SOA solutions based on the traditional network [shown in Figure 3 (a)]. It is more difficult for service interoperation with the increase in the number of services. Just as the requirement of getting information expedites the birth of web search engines, like Google and Baidu, ServNet is such an infrastructure to carry out service interoperation and crossover cooperation in an open and dynamic environment.

As shown in Figure 3(b), the protocol stack of a service network consists of four layers. The physical layer provides the way to share and transport data from one node of the network to another, serving as the traditional network. The exchange layer in the protocol stack is based on REST and responsible for describing a specific service [Crossover Service Description Language (CSDL)], providing the common communication method between different service nodes (e.g., CS-routing). The service layer is responsible for providing easy deployment/management functionality, such as Service Network Management Framework, Quality of Crossover Service, and Crossover Service Level Agreement. The application layer is responsible for service integration to provide great value, in which Standardized Service is designed to provide a transparent logic view for heterogeneous services.

Formally, ServNet is defined as 8-tuple

$$\mathrm{ServNet} = \langle Nw, Nr, S, St, Rwr, Rws, Rss, Rst \rangle$$

where *Ns*, *Nr*, *S*, *St*, *Rn*, *Rs*, *R*, *Rt* are, respectively, service switch node, service router node, actual service, standardized service, relation between service switch node and service router node, relation between service switch node and actual service, relation between actual services, and relation between standardized service and actual service.

The two service nodes can be defined as

$$Nw = \langle \mathrm{NP}, \mathrm{SL} \rangle$$
$$Nr = \langle \mathrm{NP}, \mathrm{RT} \rangle$$

where

- NP (Node Property) is the basic information of node, such as name, description, and address;
- SL (Service List) is the repository, which stores all the local services information;
- RT (Routing Table) stores the routing information records of near router nodes.

The key element of ServNet is service. In this model, a service is defined as a 4-tuple

$$S = \langle \mathrm{SP}, \mathrm{SI}, \mathrm{QoS}, \mathrm{SC} \rangle$$

where

- SP (Service Property) is the basic information of service;
- SI (Service Interface) is the operation information of service;
- QoS describes the quality of service;
- SC (Service Category) stores the category and classification information, which contains the semantic features of service.

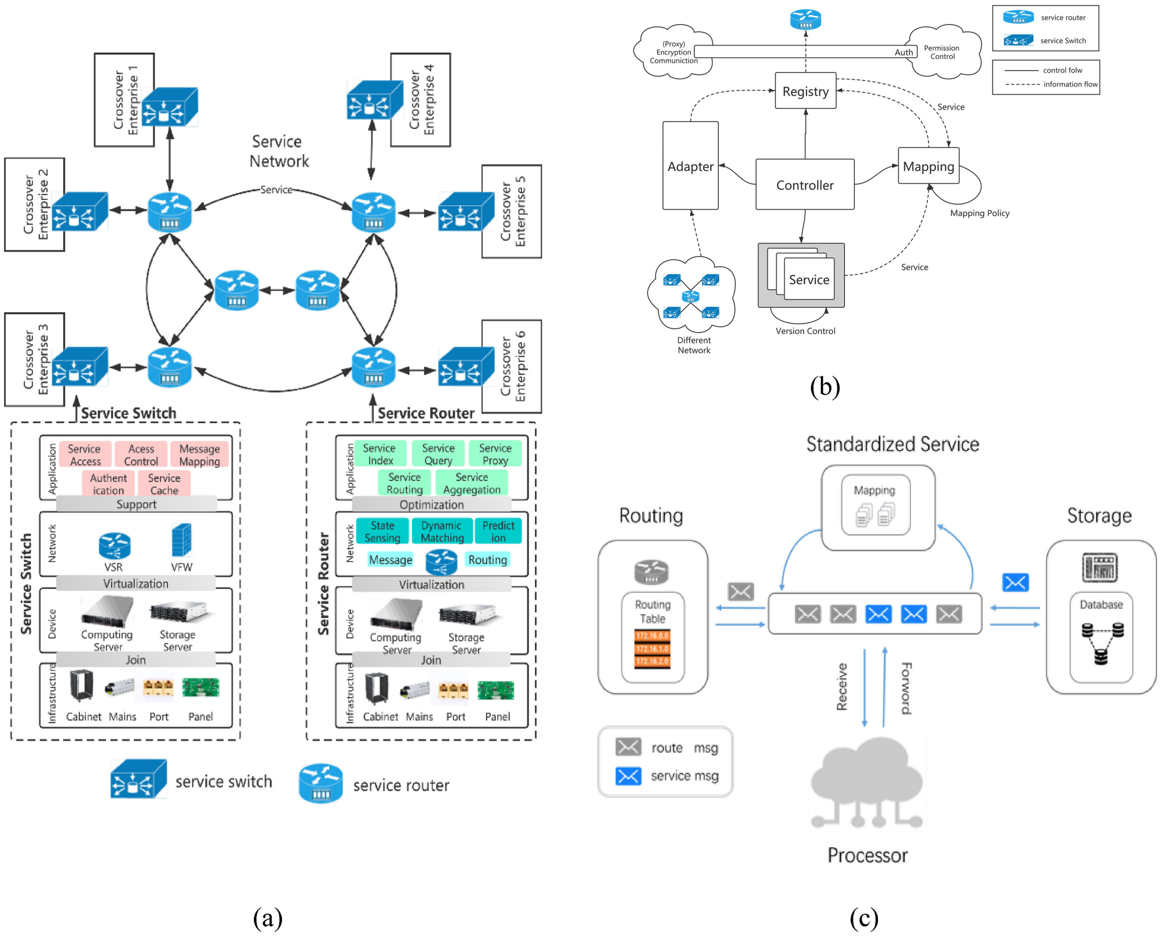


Figure 4. Architecture of ServNet and Infrastructures. (a) Architecture of ServNet. (b) Architecture of Service Switch. (c) Architecture of Service Router.

Standardized service is the abstract service in functionality. *St* is defined as a 3-tuple: *St* = <SP, IO *(Input/Output)*, SC>. And R*st* is the relation between standardized service and actual service, which will be described in this article later.

In relation R*wr*, each service switch node can have one and only one service router node as the parent node. R*ws* can help to locate an actual service from the corresponding service switch node. Generally, each service switch node represents an enterprise in the network.

R*ss* is service relation, which is the logic relationship among actual services, such as call back, sequence cooperation, and asynchronous message. The complex processes require the cooperation of services. For example, we need to define a process to satisfy the requirement of an appointment.

Service Switch and Service Router

ServNet Architecture As shown in Figure 4(a), ServNet is a service-based network allowing sharing services, and is composed of service switches and service routers.

While it is difficult for most enterprises to publish their own services efficiently and safely, the *service switch* is designed as the entrance to the network to realize the functions of service publishing, access controlling, and service management. Enterprises can access the ServNet and publish services through service switches. Moreover, they can search and apply for other services for ease of use from different companies all over the network. The framework of service switch consists of the following four layers: Infrastructure Layer (cabinet, mains, panel, and so on); device layer (computing server and storage server); network layer (virtual services router and virtual firewall); and application layer (authentication, publishing, management, and so on).

Service router is another service networking key device that forwards service requests between service nodes to realize the functions of service discovery, service integration, and service routing. Moreover, service routers store the routing information of the service network, which can adjust the network scale, peer status, and connections among peers. The framework of the service router also consists of four layers, and the infrastructure layer and device layer are the same as service switches. The network layer provides reliable delivery of services between different nodes, including proxy, caching, and routing. The application layer is responsible for service discovery, standardized service, service analysis, and so on.

Service Switch: Gateway to Publish and Access the Open Service for Enterprise A service switch is the hardware that connects enterprise on a service network to receive, and forward data to the inter- and exterservice. A service switch is the service bridge that uses service addresses to forward service at the exchange layer. As illustrated in Figure 4(b), there are several key functional modules.

- The registry module (ReM) maintains the service information and business data of the enterprise. Once a service registered successfully, the service information will be stored locally and its metadata will be broadcasted to the whole network by flooding or other ways. The service can be searched and used by other service switches shortly thereafter.
- The adapter module (AdM) is designed to support environmental heterogeneity. An AdM provides a variety of adaptors that can be implemented by different languages to enable communication with heterogeneous services. The SOAP WS, RESTful WS, and database adaptors translate data formats between services within the invocation process. Assuming semantically compatible, the various services connected to different adaptors become connected and integrated, despite different hardware and software platforms, languages, and APIs.
- The policy module (PoM) uses several policies to improve the performance. The proxy policy enables users to invoke the services via the proxy. The load balance policy of the service switch allows for scalability and load balancing by increasing the number of processes. The service caching policy enables some nodes to cache the result of a service invocation, which can make full use of the resources of edge nodes in the network.
- The mapping module (MaM) aims to bring convenience for users to invoke the service or receive the result in the particular format. An intuitive graphical interface is provided for users to generate rules files.
- The authentication module (AuM) is designed based on key cryptography and using role-based access control policy to protect the safety of data.

Service Router: Bridge for Service Connection Between Enterprise A service router is a networking device that forwards service in the service network. On the one hand, service routers collect service information from service switches and broadcast the data in the whole network. On the other hand, service routers would forward the request to the corresponding service nodes if services were invoked by the users. The structure of the service router is shown in Figure 4(c), which contains the routing component, the storage component, the processor component, and the standardized service component.

- The storage component (StC) consists of three parts: the cache of services, includes some service metadata and some query results; the routing information, which is the basic of the whole network; and the structure information of the service network.
- The routing component (RoC)'s function is to forward the service request based on the service ID, which is quite different from the IP address in the traditional network.
- The processor component (PrC) provides a communication mechanism between internal and external resources. There are two types of messages: service message and route message. Once receiving a message, it will be

preprocessed by PrC, then transmitted to the routing component or the storage component.

- The standardized service component (SsC) is designed for homogeneous services with similar functions in the service network. When any fault discovered for a web service, it can dynamically replace services with suitable ones by using standardized service, which has greatly improved the availability and reliability of services.

Key Techniques and Functionalities

Service Wrapper: Converting Web Data into Services Webpages are an important open resource for enterprise. We develop a tool called the Service Wrapper, which consists of two main components, the Service Extractor and Service Invoker, to facilitate the servitization of the webpage. The former is responsible for dynamic webpage handling, block segmentation, and service information generation. The latter is used to invoke the services generated by the Service Extractor through the corresponding APIs. Finally, we can obtain the data whose structure is predefined by the wrapper user with the Service Extractor.

The Service Extractor is used for service generation. It receives a web page and encapsulates its internal specified information into a Web service. There are two cases we consider when extracting data from web pages: dynamic and static web pages. For dynamic web pages, we need to find possible fillable forms on the page and provide them to our users to decide which form to fill in. After that, we can directly segment the page into blocks and sort the blocks by analyzing the structure of pages and data.

The Service Invoker receives an API request from the caller and returns structured data designed by the wrapper user. After a user sends an API request with the method GET, the Service Invoker accesses the corresponding service information in the database, handles the request with its modules. Analogously, the Service Invoker contains four modules to complete the service invocation: The Parametric Analyzer is responsible for analyzing the request parameters; The System Request Parameters Processor is used to deal with the system request parameters; The Service Information Parser maps the keys of input parameters to the actual fields of a specified form; The Result Filter helps the caller to filter out some items from the output of the last module with some filtering conditions.

Standardized Service: Providing a Transparent Logic View for Heterogeneous Services Several services with the same or similar functions are common in ServNet; for example, there are many appointment services in internet hospitals, which have different API formats. Therefore, it is difficult for service consumers to choose the suitable service. The services with similar functions but different operations cause the high cost of migration from one to another when the former is disabled. Standardized service is proposed to solve these problems.

A standardized service is the abstraction and unification of homogeneous services with similar functions in ServNet. The generation of standardized services is to identify the homogeneous services that already exist in ServNet through data mining methods, which can generate a blueprint for standardized services automatically. On this basis, experts from relevant fields will be invited to determine the final standardized service manually before publishing to ServNet.

For service providers, they can choose to associate their own services to map to the standardized services in this system. Specifically, the service switch will recommend relevant standardized services for the current service actively. A visual tool is developed to bind services and generate configuration files automatically by using natural language processing, in which the description of service, parameters, interfaces, and other semantic information will be considered. In this way, the service consumer can create an association between the standardized services and the local services. For service consumers, you can look up and use the standardized service directly. The routing policy of standardized service will route to the appropriate service and return the result. The routing policy will make use of the various information of the context to make suitable choice. The users can also prompt to customize their own strategies.

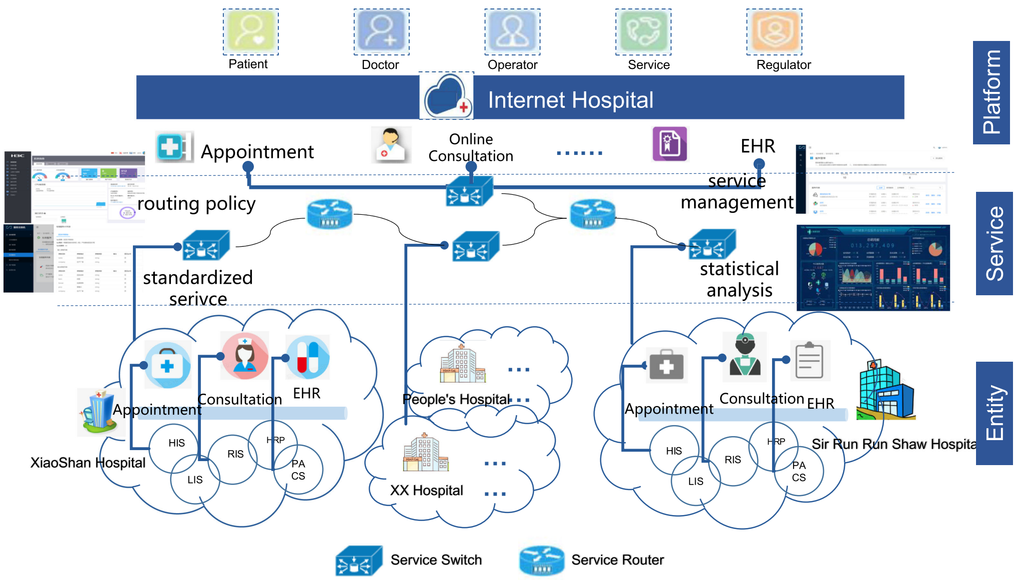


Figure 5. Service Integration in E-health with Service Network. Service switch is provided for hospitals to publish services. Service router is the device that forwards service requests between service nodes. In addition, it provide a rich user interface and advanced functionality for users to operate conveniently.

Service Routing: Locating and Discovering the Service Efficiently Service routing, in particular, refers to the problem of how to search for and locate services.

The descriptions of services are usually considered lying in well-defined service repositories. A substantial progress has been done in this area thanks to several research and industrial efforts.[6,7] However, these solutions are typically limited for two reasons: First, they are usually centralized where there is a central server (e.g., UDDI registry) that keeps track of all available services. As a result, those solutions perform unsatisfactory in scalability and fault tolerant. Second, they usually offer limited search capabilities.

To address the above issues, the Hierarchical Service Routing based on Network Resource (HRNR) that supports a rich set of search operations, is proposed. Chord is a P2P system that implements Distributed Hash Table(DHT), which supports efficient exact key lookups. However, dynamic node join and departure might be costly due to finger table updates and key transfers. Additionally, Chord does not consider the heterogeneity of the peers and treats each peer equally. HRNR uses a super peer-based overlay built on top of Chord. Instead of inserting all participants into the Chord ring, only a subset of the participants are assigned as super peers and join the Chord ring. In this architecture, the super peers do all the indexing and query routing. The remaining peers are called the client peers and they use the system by connecting to a super peer. Each client peer forwards its requests to its super peer who processes the requests on its behalf.

APPLICATION CASE FOR E-HEALTH

We develop the prototype of crossover service system named JTangYdrail based on the architecture of ServNet. With JTangYdrail applied as shown in Figure 5, service switch is provided to access the network for each hospital. Once the service switch is connected, hospitals can publish their medical services easily based on the various adaptors, unified integration semantics, and environment-aware management mechanisms. The Internet Hospital can also apply for and provide these integrated services to patients. In addition, JTangYdrail provides a rich user interface and advanced functionality for users to operate conveniently. For example, the webpage of statistical analysis, configuring route policy, standardized services and so on.

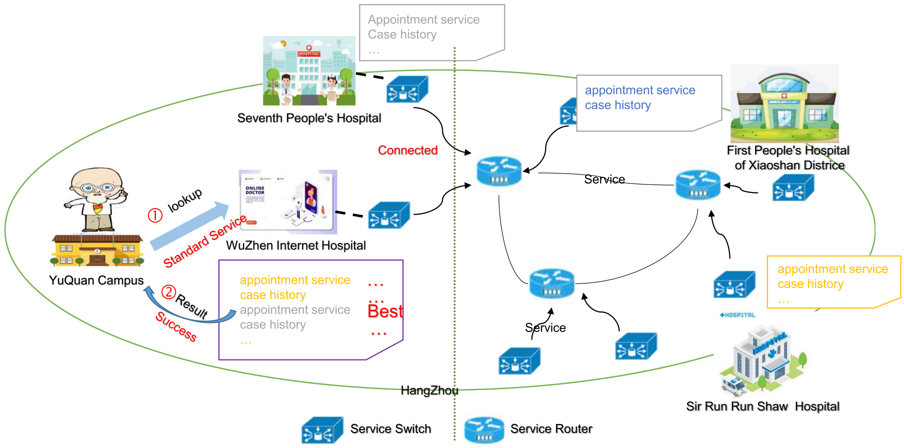


Figure 6. Medical Example with Service Network. Medical services are integrated by service network approach. What John need to do is just to look up for the standardized services and invoke them, which is quite convenient.

With JTangYdrail applied as shown in Figure 6, to make an appointment service, John simply needs to look up the standardized services and invoke them. All the hospitals are connected by service infrastructures and different appointment services are published in the service network. The mechanism of standardized service will route the request to the best actual service and return the result. It is quite convenient for users to use these services after being processed by this system. JTangYdrail can provide the transparency between the complex crossover service environment and the convenient user service access.

In industry, JTangYdrail is also applied to practice and gets some good results. In the medical field, JTangYdrail is applied to dozens of hospitals, including more than 30 large-scale hospitals and two regional health institutions, covering ten provinces in China. For example, for The Children's Hospital, Zhejiang University School of Medicine, JTangYdrail has been running successfully for about a year, and the web services published were accessed hundreds of millions of times.

RELATED WORK

Services are a key technology for implementing SOA solutions on the Internet. Most of the services are in the isolate status, namely "the service island." This has dramatically increased the need to build a fundamental infrastructure for the efficient deployment and operation of the exponentially growing plethora of crossover services. Some research focusing on these areas has been carried out.

Service mesh is the dedicated infrastructure layer for handling service-to-service communication. As part of the microservices ecosystem, service mesh technologies hold the promise of addressing communication-related concerns(e.g., interopeability).[8,9] Service mesh mainly focuses on the running time operations of services within enterprises. By contrast, the service network not only provides the mechanism of service discovery for enterprises to publish services, but also adopts several technologies to manage services and provide a high performance.

In recent years, research on crossover service has gradually emerged. Wu *et al.*[3] proposed the concept of crossover service. Xu *et al.*[10] presented the concept of big service, which refers to the complex service ecosystem that combines lots of domain services caused by big data. Xue *et al.*[11] presented the service bridge to evaluating the trans-boundary impact of the Internet. There is currently a lack of research on how to manage crossover service.

In the area of service extraction, Ferrara *et al.*[12] have a comprehensive overview of the literature in the field of web data extraction. Many methods tend to define a type of special rules, such as the Web Wrapper Factory (W4F),[13] which uses its declarative language to extract the data from Web pages. As for service routing, a novel method was presented for solving the node-constrained service chain routing problem in a software-defined network.[6] The main idea is to transform the network representation to a layered graph that considers processing steps. In Distributed Service Networks, the main concern of the users is mostly focused on the response time of a service and the mathematical model is proposed to minimize the expected response time.[7] As for service matching, a structured and systematic procedure to evaluate service matching during early design phases can be particularly useful to reduce the risk of neglecting important tasks and to avoid wasting time.[14]

CONCLUSION

In this article, a novel service approach named Service Network (ServNet), which is composed of service switches and service routers, is proposed to bridge the gap between service consumers and service providers in crossover service ecosystems. JTangYdrail is the concrete implementation of the ServNet approach that combines hardware with software to take care of the distributed environment and *heterogeneous* service. In addition, several key techniques or tools are designed to achieve high performance and use convenience: the service wrapper tool to convert web data into services, the standardized service mechanism to provide a transparent logic view for heterogeneous services and the service routing algorithm to locate and discover the services efficiently.

However, there are still works to be studied in the future. A good service registry should support developers at design time, but how and to what extent is not exactly clear. With the running of ServNet, the changes of services in ServNet will affect the performance. Service process based on ServNet will be a valuable work in the near future.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400603, in part by the National Natural Science Foundation of China under Grant 61825205 and Grant 61772459, and in part by the National Science and Technology Major Project of China under Grant 50-D36B02-9002-16/19.

REFERENCES

1. IBM. API economy. [Online]. Available: https://developer.ibm.com/ apiconnect/ 2018/01/04/api-api-economy/
2. S.-C. Oh, D. Lee, and S. R. T. Kumara, “Effective web service composition in diverse and large-scale service networks,” *IEEE Trans. Services Comput.*, vol. 42, no. 10, pp. 15–32, Jan./Mar. 2008.
3. Z. Wu, J. Yin, S. Deng, J. Wu, Y. Li, and L. Chen, “Modern service industry and crossover services: Development and trends in China,” *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 664–671, Sep./Oct. 2016.
4. S. Guo, C. Xu, S. Chen, X. Xue, Z. Feng, and S. Chen, “Crossover service fusion approach based on microservice architecture,” in *Proc. IEEE Int. Conf. Web Services*, 2019, pp. 237–241.
5. J. Yin *et al.*, “Crossover service: Deep convergence for pattern, ecosystem, environment, quality and value,” in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 1250–1257.
6. A. Dwaraki and T. Wolf, “Adaptive service-chain routing for virtual network functions in software-defined networks,” in *Proc. ACM Workshop Hot Topics Middleboxes Netw. Function Virtualization*, 2016, pp. 32–37.
7. S. Gupta and A. P. Singh, “Mathematical routing model based on nodes constraints in a distributed service network,” *Aryabhatta J. Math. Informat.*, vol. 10, pp. 361–366, 2018.
8. N. C. Mendoncça, D. Garlan, B. R Schmerl, and J. Cámara, “Generality vs. reusability in architecture-based self-adaptation: The case for self-adaptive microservices,” in *Proc. 12th Eur. Conf. Softw. Archit. (Companion)*, 2018, pp. 18–1.
9. H.-L. Truong, L. Gao, and M. Hammerer, “Service architectures and dynamic solutions for interoperability of IoT, network functions and cloud resources,” in *Proc. 12th Eur. Conf. Softw. Archit. (Companion)*, 2018, pp. 2–1.
10. X. Xu, Q. Z. Sheng, L.-J. Zhang, Y. Fan, and S. Dustdar, “From big data to big service,” *Computer*, no. 48, pp. 80–83, 2015.

11. X. Xue, G. Gao, S. Wang, and Z. Feng, “Service bridge: Transboundary impact evaluation method of internet,” *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 3, pp. 758–772, Sep. 2018.
12. E. Ferrara, P. D. Meo, G. Fiumara, and R. Baumgartner, “Web data extraction, applications and techniques: A survey,” *Knowl.-Based Syst.*, vol. 70, pp. 301–323, 2014.
13. A. Sahuguet and F. Azavant, “Building light-weight wrappers for legacy web data-sources using W4F,” in *Proc. 25th Int. Conf. Very Large Data Bases*, 1999, vol. 99, pp. 738–741.
14. F. Barravecchia, F. Franceschini, and L. Mastrogiacomo, “A service network perspective to evaluate service matching in early design,” *J. Service Theory Practice*, vol. 28, no. 3, pp. 356–383, 2018.

Bangpeng Zheng is working toward the Ph.D. degree with Zhejiang University, Hangzhou, China. His current research interests include crossover service, service computing, and cloud storage systems. Contact him at zhengbangpeng@zju.edu.cn.

Jianwei Yin is a Full Professor with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. His research interests include Service Computing, Cloud Computing, and Data Science. He is the corresponding author. Contact him at zjuyjw@cs.zju.edu.cn.

Shuiguang Deng is a Full Professor with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. His research interests include Edge Computing, Service Computing, Mobile Computing, and Business Process Management. Contact him at dengsg@cs.zju.edu.cn.

Zhaohui Wu is a Full Professor with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. His research interests include service computing, artificial intelligence. Contact him at wzh@cs.zju.edu.cn.

Schahram Dustdar is a Full Professor of computer science and the Head with the Distributed Systems Research Division, TU Wien, Vienna, Austria. Contact him at dustdar@dsg.tuwien.ac.at.