

یکپارچه‌سازی سیستم‌ها در سیستم‌های فوق مقیاس وسیع با استفاده از رویکرد سرویس‌های غنی داده-محور

سعید شکرالهی، فریدون شمس و جواد اسماعیلی

قواعد طوری توصیف و اعمال می‌شوند که علاوه بر به کارگیری کارای منابع، هدف کلی از ایجاد سیستم نیز برآورده شود. سیستم‌های فوق مقیاس وسیع دارای مقیاس چشمگیری در ابعاد مختلف هستند. از جمله این ابعاد می‌توان به تعداد مؤلفه‌های توزیع‌شده که باید یکپارچه شوند، تعداد اتصالات بین مؤلفه‌ها، تعداد دامنه خط مشی‌ها، تعداد ساز و کارهای اعمال خط مشی‌ها و همچنین میزان داده‌ها اشاره کرد [۱] تا [۳]. نتیجه چنین مقیاسی سبب می‌شود که نیاز به یکپارچه‌سازی مقیاس‌پذیر، پویا و امن زیرسیستم‌های یک سیستم فوق مقیاس وسیع به عنوان یک نیاز اساسی در این سیستم‌ها مطرح شود.

روش‌های سنتی یکپارچه‌سازی شامل فرآیندهای زمان‌بری هستند که مخاطرات تکنیکی و مالی بسیار زیادی دارند. معماری سرویس‌گرا به عنوان راه حلی مناسب برای رسیدن به یکپارچه‌سازی و تعامل‌پذیری بهتر سیستم‌ها معرفی شده است. با پذیرفتن اصول سرویس‌گرایی، سیستم‌های فوق مقیاس وسیع به جای مجموعه‌ای از کاربردهای مجزا به اکوسیستمی از سرویس‌های مرتبط، یکپارچه و با قابلیت استفاده مجدد تبدیل خواهند شد. در سال‌های اخیر، معماری سرویس‌های غنی^۲ که نوعی از معماری سرویس‌گرا است به عنوان یک رویکرد مناسب جهت تسهیل یکپارچه‌سازی در سیستمی از سیستم‌ها معرفی شده است [۴] تا [۸]. معماری سرویس‌های غنی قابلیت یکپارچه‌سازی افقی و عمودی سرویس‌ها را فراهم می‌کند. یکپارچه‌سازی افقی، دغدغه‌های متداخل^۳ مربوط به یک مجموعه سرویس درون یک دامنه منطقی و فیزیکی را مد نظر قرار می‌دهد در حالی که یکپارچه‌سازی عمودی امکان تجزیه ساختاری سرویس‌ها جهت مقیاس‌پذیری آنها و محصورسازی دغدغه‌ها را فراهم می‌کند [۵] و [۶].

در معماری سرویس‌های غنی، یکپارچه‌سازی افقی از طریق اتصال مجموعه‌ای از سرویس‌های مرتبط به یک فراساختار ارتباطی که قابلیت رهگیری و مسیریابی تعاملات در میان سرویس‌ها را دارند به دست می‌آید. در سرویس‌های غنی و همچنین سایر چارچوب‌های پیشنهادشده برای یکپارچه‌سازی سرویس‌ها، اکثر فراساختارهای ارتباطی به کار گرفته شده پیام-محور^۴ بوده یا مبتنی بر گذرگاه سرویس سازمانی^۵ (ESB) هستند و از مدل نقش-مبنای^۶ برای تعیین ارتباطات امن بین سرویس‌ها استفاده می‌کنند [۴] تا [۱۶]. به همین جهت، مقیاس‌پذیری و پویایی سرویس‌های غنی و چارچوب‌های مذکور را در یکپارچه‌سازی سیستم‌ها در سیستم‌های فوق مقیاس وسیع کاهش می‌یابد. در این مقاله، جهت افزایش مقیاس‌پذیری و پویایی سرویس‌های غنی در سیستم‌های فوق

چکیده: سیستم فوق مقیاس وسیع، یک سیستم یکپارچه از سیستم‌هایی است که معمولاً دارای دغدغه‌های متداخل زیادی هستند. هنگامی که اندازه سیستمی از سیستم‌ها رشد کرده و تعامل‌پذیری بین زیرسیستم‌های آن افزایش پیدا می‌کند، یکپارچه‌سازی مقیاس‌پذیر و پویای زیرسیستم‌ها به یک چالش اساسی تبدیل خواهد شد. در این یکپارچه‌سازی، هر یک از زیرسیستم‌ها دامنه مخصوص به خود را داشته و هر دامنه ممکن است خط مشی‌های مستقلی داشته باشد. در سال‌های اخیر، استفاده از سرویس‌های غنی به عنوان یک رویکرد مناسب جهت تسهیل یکپارچه‌سازی سیستم‌ها مطرح شده است. در این مقاله، جهت بهبود پویایی، مقیاس‌پذیری و امنیت سرویس‌های غنی در سیستم‌های فوق مقیاس وسیع، رویکرد سرویس‌های غنی داده-محور پیشنهاد می‌شود. در رویکرد پیشنهادی، یک میان‌افزار دولایه و داده-محور جهت هم‌نوآوری سرویس‌های غنی ارائه و به کار گرفته شده است. لایه زیرین میان‌افزار شامل یک میان‌افزار سرویس توزیع داده است که مسئول ارتباطات پویا، مقیاس‌پذیر و بلادرنگ سرویس‌های غنی است. لایه بالایی میان‌افزار نیز جهت پیکربندی و بازپیکربندی پویا و امن سرویس‌های غنی به کار گرفته می‌شود. در بخش ارزیابی، کارایی رویکرد پیشنهادی با استفاده از روش شبیه‌سازی، مورد تحلیل و بررسی قرار گرفته است. نتایج حاصل از تحلیل و بررسی انجام‌شده نشان می‌دهد که رویکرد پیشنهادی در یکپارچه‌سازی سیستم‌ها در سیستم‌های فوق مقیاس وسیع در حد قابل توجهی مفید است.

کلیدواژه: یکپارچه‌سازی سیستم‌ها، سیستم‌های فوق مقیاس وسیع، سرویس‌های غنی، میان‌افزار سرویس توزیع داده.

۱- مقدمه

سیستمی از سیستم‌ها به مجموعه‌ای از سیستم‌های مستقل و مفید گفته می‌شود که در قالب یک سیستم بزرگ‌تر جمع شده و قابلیت‌های منحصر به فردی ارائه می‌کنند. با رشد استفاده و ایجاد سیستمی از سیستم‌ها، به تدریج سیستمی از سیستم‌های بزرگ و پیچیده‌ای تولید خواهد شد که سیستم‌های فوق مقیاس وسیع^۱ (ULSS) نامیده می‌شود [۱]. یک سیستم فوق مقیاس وسیع شامل مجموعه‌ای پویا از سازمان‌های مستقل و تعامل‌پذیر است که برای به دست آوردن منابع با یکدیگر رقابت کرده و گاه اهداف متناقضی دارند. در چنین سیستمی، خط مشی‌ها و

این مقاله در تاریخ ۳۱ تیر ماه ۱۳۹۳ دریافت و در تاریخ ۲۸ مهر ماه ۱۳۹۴ بازنگری شد.

سعید شکرالهی، دانشکده علوم و مهندسی کامپیوتر، دانشگاه شهید بهشتی، تهران، (email: sa_shokrollahi@yahoo.com).

فریدون شمس، دانشکده علوم و مهندسی کامپیوتر، دانشگاه شهید بهشتی، تهران، (email: f_shams@sbu.ac.ir).

جواد اسماعیلی، دانشکده علوم و مهندسی کامپیوتر، دانشگاه شهید بهشتی، تهران، (email: j-esmaeili@sbu.ac.ir).

2. Rich Services Architecture

3. Crosscutting Concerns

4. Message-Centric

5. Enterprise Service Bus

6. Role-Based Policies

1. Ultra Large Scale Systems

بالایی، لایه بازسازی محلی داده نام دارد و یکپارچه‌سازی کاربردها با میان‌افزار را تسهیل می‌کند. شکل ۱ موجودیت‌ها و مفاهیم اصلی میان‌افزار سرویس توزیع داده را نمایش می‌دهد. در این شکل، یک دامنه شامل دو شریک دامنه^۵ وجود دارد. شریک دامنه موجودیتی است که شراکت کاربردها در یک دامنه با آن نمایش داده می‌شود. میان‌افزار می‌تواند شامل دامنه‌های مختلفی باشد. هر دامنه یک محیط ارتباطی مجازی برای شرکای خود ایجاد می‌کند. هر شریک دامنه، ناشران^۶ و متقاضیان^۷ خود را مدیریت می‌کند. ناشر مانند محفظه‌ای برای گروه‌بندی نویسندگان داده^۸ و متقاضی مانند محفظه‌ای برای گروه‌بندی خوانندگان داده^۹ فراهم می‌کند. هر خواننده، داده‌های مربوط به خود را به صورت همگام یا غیر همگام می‌تواند دریافت کند. در میان‌افزار سرویس توزیع داده، هر عنوان^{۱۰} یک نگهدارنده داده است که اتصال نویسندگان و خوانندگان را برقرار می‌کند. هر عنوان یک اسم و یک نوع دارد. این میان‌افزار، توانایی تفسیر داده‌های مبادله‌شده مربوط به عناوین را داشته و می‌تواند آنها را نهان‌سازی یا فیلتر کند.

داده‌های یک عنوان با کلید اصلی مشابه، یک نمونه داده از آن عنوان محسوب می‌شوند. در میان‌افزار سرویس توزیع داده، ارتباطات از طریق عنوان‌ها و به صورت شفاف و ناشناس صورت می‌گیرد و نویسندگان و خوانندگان نگران این نیستند که عنوان‌ها را چه کسی و چگونه ایجاد کرده، نوشته یا می‌خواند. از مهم‌ترین ویژگی این میان‌افزار، پشتیبانی از خط مشی‌های کیفیت سرویس^{۱۱} است که برخی از جنبه‌های رفتار سرویس را کنترل می‌کند. این خط مشی‌ها در سطوح مختلف دانه‌بندی مانند ناشر، متقاضی، خواننده، نویسنده و عنوان قابل تنظیم هستند.

۲-۲ مدل OASIS

مدل OASIS یک مدل معماری کنترل دسترسی مبتنی بر نقش برای تعامل امن سرویس‌های مستقل در محیط‌های باز و توزیع‌شده است. در این معماری، سرویس‌ها در دامنه‌های مستقل توسعه پیدا کرده و با استفاده از توافق سطح سرویس امکان تعامل امن بین آنها فراهم می‌شود [۲۱] تا [۲۳]. در مدل OASIS نقش‌ها توسط سرویس‌ها تعیین می‌شوند و مدیریت متمرکز و سراسری بر روی اسم نقش‌ها و امتیازدهی‌ها وجود ندارد. هر سرویس، نقش‌های خود را نامگذاری کرده و قوانین خود را برای فعال‌سازی نقش و درخواست سرویس بیان می‌کند. یک عامل می‌تواند بر اساس قاعده-فعال‌سازی-نقش^{۱۲} که توسط سرویس تعیین شده است و ارائه گواهی‌نامه خود یک نقش را گرفته و به یک عامل اصلی^{۱۳} تبدیل شود که می‌تواند سرویس را درخواست نماید. در مدل OASIS نقش‌ها می‌توانند بر حسب نیاز پارامتر بگیرند. این پارامترها می‌توانند امتیازها را در زمان درخواست انتشار دهند.

در مدل OASIS یک قاعده-فعال‌سازی-نقش به صورت زیر است

$$r_1, r_2, \dots, a_1, a_2, \dots, e_1, e_2, \dots \mid \text{-role} \quad (1)$$

5. Domain Participant

6. Publishers

7. Subscribers

8. Data Writers

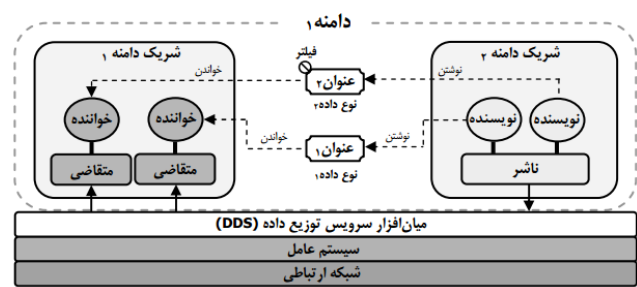
9. Data Readers

10. Topic

11. Quality of Service (QoS) Policies

12. Role-Activation-Rule

13. Principal



شکل ۱: مفاهیم اصلی در میان‌افزار سرویس توزیع داده.

مقیاس وسیع، رویکرد سرویس‌های غنی داده-محور^۱ (DCRS) پیشنهاد شده است. در این رویکرد، یک میان‌افزار دولایه و داده-محور به عنوان فراساختار ارتباطی سرویس‌های غنی پیشنهاد و به کار گرفته شده است. لایه زیرین این میان‌افزار که لایه توزیع (DL) نامیده می‌شود، مسئول ارتباطات پویا، مقیاس‌پذیر و بلادرنگ بین سرویس‌های غنی است. از معماری میان‌افزار سرویس توزیع داده^۲ (DDS) در لایه توزیع استفاده شده است. لایه بالایی میان‌افزار که لایه پیکربندی (CL) نامیده می‌شود، برای پیکربندی و بازپیکربندی پویا و امن سرویس‌های غنی استفاده می‌شود. لایه پیکربندی هر سرویس غنی داده-محور، اطلاعات پیکربندی و بازپیکربندی مربوطه را بر اساس نقش‌هایی که آن سرویس غنی داده-محور دارد دریافت می‌کند. در این لایه از مدل OASIS برای تعیین ارتباطات امن بین سرویس‌ها استفاده می‌شود.

ساختار ادامه مطالب این مقاله به شکل زیر است. در بخش دوم، پس‌زمینه‌ای از میان‌افزار سرویس توزیع داده و همچنین مدل OASIS ارائه می‌شود. در بخش سوم، برخی از کارهای مرتبط با یکپارچه‌سازی سیستم‌ها مورد بررسی قرار می‌گیرد. بخش چهارم به معرفی الگوی معماری سرویس‌های غنی داده-محور می‌پردازد. در بخش پنجم، کارایی سرویس‌های غنی داده-محور مورد ارزیابی قرار می‌گیرد و در بخش ششم، نتیجه‌گیری مقاله ارائه خواهد شد.

۲-۲ پس‌زمینه‌ای از سرویس توزیع داده و مدل OASIS

در این بخش به معرفی مختصری از میان‌افزار سرویس توزیع داده می‌پردازیم و همچنین پس‌زمینه‌ای را از مدل OASIS ارائه می‌کنیم.

۱-۲ میان‌افزار سرویس توزیع داده

میان‌افزار سرویس توزیع داده که توسط گروه مدیریت شیء توصیف شده است [۱۷]، امکان ارتباطات ناشر-متقاضی، داده-محور و بلادرنگ را برای سطح گسترده‌ای از محیط‌های محاسباتی فراهم می‌کند. این میان‌افزار از مدل ارتباطی یک به یک استفاده کرده و مستقل از موقعیت و سکو است. در سال‌های اخیر مقیاس‌پذیری و کارایی میان‌افزار سرویس توزیع داده در سطح شبکه جهانی و سیستم‌های فوق مقیاس وسیع افزایش پیدا کرده است [۱۸] تا [۲۱]. در این میان‌افزار، واسطه‌ها در دو سطح توصیف می‌شوند. سطح زیرین، لایه ناشر-متقاضی داده-محور نام دارد و کاربردها را قادر می‌سازد که بر روی سکوهای ناهمگن و از طریق یک فضای داده مشترک سراسری با یکدیگر تبادل داده نمایند. سطح

1. Data-Centric Rich Services

2. Distribution Layer

3. Data Distribution Service

4. Configuration Layer

کرده‌اند [۲۷] و [۲۸]. در یکی از کارهای اخیر نشان داده شده که پیاده‌سازی سرویس‌ها بر روی یک میان‌افزار داده-محور مانند میان‌افزار سرویس توزیع داده، می‌تواند فواید زیادی نسبت به میان‌افزارهای پیام-محور برای یکپارچه‌سازی سیستم‌ها در سیستم‌های توزیع‌شده و مقیاس وسیع داشته باشد [۲۹]. در میان‌افزارهای داده-محور به جای تمرکز بر روی مجموعه پیام‌های مربوط به سرویس‌ها بر روی داده‌های به اشتراک گذاشته شده توسط سرویس‌ها تمرکز می‌شود. میان‌افزارهای داده-محور از داده‌های مبادله‌شده بین سرویس‌ها آگاه هستند و این آگاهی منجر به توانایی‌هایی همچون پشتیبانی از خط مشی‌های کیفیت سرویس در این میان‌افزارها می‌شود. این توانایی‌ها سبب می‌شود که یکپارچه‌سازی سرویس‌ها با استفاده از میان‌افزارهای داده-محور از مقیاس‌پذیری، پویایی و دسترسی‌پذیری بیشتری نسبت به یکپارچه‌سازی سرویس‌ها با میان‌افزارهای پیام-محور برخوردار شود [۲۱]. در یکی از کارهای اخیر، خدادادی و همکاری یک چارچوب داده-محور را جهت یکپارچه‌سازی مقیاس‌پذیر کاربردها در محیط‌های رایانش ابری پیشنهاد داده‌اند [۳۰]. در اکثر فراساختارهای ارتباطی پیشنهادشده برای سرویس‌های غنی و همچنین سایر چارچوب‌های پیشنهادشده برای یکپارچه‌سازی سرویس‌ها، از مدل نقش-مبنا برای تعیین ارتباطات امن بین سرویس‌ها استفاده شده است [۴] تا [۱۶]. مدل نقش-مبنا برای تعامل امن سرویس‌های مستقل در محیط‌های باز و توزیع‌شده دارای محدودیت‌هایی است. در [۲۲] و [۲۳] به این محدودیت‌ها اشاره شده و جهت افزایش مقیاس‌پذیری و پویایی مدل نقش-مبنا و به کارگیری آن در محیط‌های مقیاس وسیع و توزیع‌شده مدل OASIS پیشنهاد شده است.

۴- سرویس غنی داده-محور

دغدغه‌هایی که در یک سیستم فوق مقیاس وسیع باید مد نظر قرار گیرد بسیار زیاد و متنوع بوده و سبب افزایش پیچیدگی در این سیستم‌ها شده است. مفهوم سرویس‌گرایی گزینه‌ای مناسب برای مقابله با این پیچیدگی‌ها است. ایده سرویس‌های غنی، انفصال^۶ دغدغه‌های موجود در سیستمی از سیستم‌ها را فراهم می‌کند. در این بخش، جهت افزایش مقیاس‌پذیری و پویایی سرویس‌های غنی در سیستم‌های فوق مقیاس وسیع، رویکرد سرویس‌های غنی داده-محور پیشنهاد می‌شود. الگوی معماری سرویس‌های غنی داده-محور در شکل ۲ نشان داده شده است. یک سرویس غنی داده-محور خود شامل سه نوع سرویس با نام‌های سرویس کاربردی^۷، سرویس متداخل^۸ و سرویس مدیریت^۹ است. سرویس‌های کاربردی، فراهم‌کننده عملکردهای کاربردی هستند و می‌توانند آغازگر ارتباط با دیگر سرویس‌های شرکت‌کننده در یک گردش کاری باشند. سرویس‌های متداخل، دغدغه‌های متداخل سرویس‌های کاربردی را مد نظر قرار می‌دهند و می‌توانند تعاملات تعریف‌شده بین سرویس‌های کاربردی را مانیتور کرده یا آنها را از طریق مسیریابی مجدد، فیلترکردن یا تصحیح مسیر تبادل پیام تغییر دهند. سرویس رویدادنگاری^{۱۰} و سرویس کنترل دسترسی^{۱۱} دو نمونه از سرویس‌های متداخل برای سرویس‌های کاربردی محسوب می‌شوند.

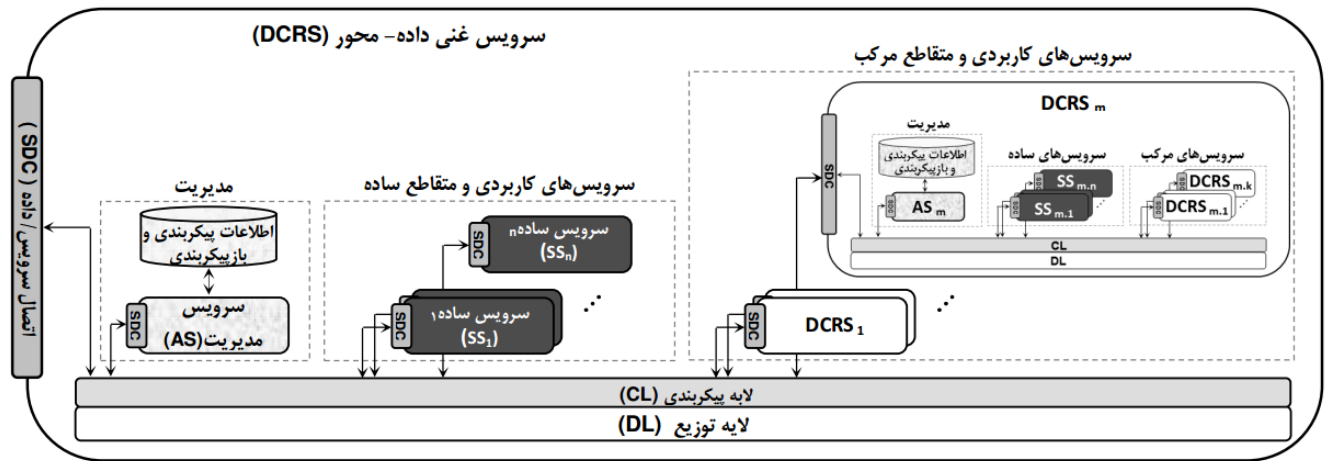
در این قاعده، r_x یک نقش پیش‌نیاز را نشان می‌دهد. یک عامل برای گرفتن نقش مربوط به یک سرویس باید یک یا چند گواهی‌نامه ارائه نماید که ثابت کند که در یک یا چند نقش خاص از آن سرویس یا سرویس‌های دیگر فعال است. در این معماری از گواهی-عضویت-نقش^۱ (RMC) برای این منظور استفاده می‌شود. a_y نشان‌دهنده یک گواهی‌نامه انتصاب است که یک انتصاب‌کننده آن را به منتصب می‌دهد تا منتصب بتواند یک یا چند نقش را بپذیرد. e_z یک محدودیت محیطی را نمایش می‌دهد. محدودیت محیطی می‌تواند وابسته یا غیر وابسته به کاربر باشد. عضویت در یک گروه خاص و محدودیت زمان نمونه‌هایی از این محدودیت‌ها می‌باشد.

۳- کارهای مرتبط

معماری سرویس‌گرا به عنوان راه حلی مناسب برای رسیدن به یکپارچه‌سازی و تعامل‌پذیری بهتر در سیستمی از سیستم‌ها معرفی شده است. نگاه در سرویس‌گرایی، یک نگاه بالا به پایین است یعنی به جای این که یک کسب و کار به صورت متمرکز و توسط یک واحد بزرگ ارائه شود، راه حلی تفکیک‌پذیر برای آن ارائه می‌شود. این راه حل به سرویس‌های تعامل‌پذیر و سست‌اتصال^۲ تجزیه شده و با هم‌نوایی^۳ و هم‌آرایی^۴ سرویس‌ها به اهداف اصلی خود می‌رسد [۷]. ترکیب سرویس‌ها سرویس‌ها به عنوان یک راه حل انعطاف‌پذیر جهت یکپارچه‌سازی در سیستمی از سیستم‌ها شناخته می‌شود. گردش کاری کسب و کار و وب معنایی دو رویکرد مناسب در ترکیب سرویس‌ها محسوب می‌شوند. در رویکرد گردش کاری کسب و کار، تعاملات بین سرویس‌های مختلف به صورت صریح و با زبان‌های گردش کاری توصیف می‌شود. توصیف تعاملات در این زبان‌ها یا مانند زبان BPEL بر اساس مدل هم‌نوایی بوده یا مانند زبان WS-CDL بر اساس مدل هم‌آرایی است. در رویکرد وب معنایی از هستان‌شناسی برای توصیف ویژگی‌های سرویس‌ها استفاده می‌شود. انجمن وب معنایی، زبان OWL-S را برای توصیف معنا و ترکیب سرویس‌ها پیشنهاد کرده است [۲۴]. در سال‌های اخیر، معماری و چارچوب‌های مختلفی جهت ترکیب سرویس‌ها در محیط‌های پویا و مقیاس وسیعی همچون سیستمی از سیستم‌ها و اینترنتی از اشیا پیشنهاد شده است [۴] تا [۱۶]. در روش‌های پیشنهادشده، معماری سرویس‌های غنی که از دو رویکرد گردش کاری کسب و کار و وب معنایی پشتیبانی می‌کند به عنوان رویکردی مناسب در یکپارچه‌سازی سیستمی از سیستم‌ها معرفی شده است [۴] تا [۸]. برخی از کارهای انجام‌شده در زمینه یکپارچه‌سازی از ایده سرویس‌های غنی جهت یکپارچه‌سازی سرویس‌های کاربردی با سرویس‌های امنیتی استفاده کرده و آن را روشی مقیاس‌پذیر معرفی کرده‌اند [۲۵] و [۲۶]. اکثر فراساختارهای ارتباطی پیشنهادشده برای سرویس‌های غنی و همچنین سایر چارچوب‌های پیشنهادشده برای یکپارچه‌سازی سرویس‌ها، یا مانند میان‌افزار AMQP پیام-محور بوده یا مانند چارچوب Mule مبتنی بر گذرگاه سرویس سازمانی هستند [۴] تا [۱۶]. در برخی از کارهای پیشنهادشده از الگوی ناشر-مقتضی^۵ جهت افزایش مقیاس‌پذیری و دسترسی‌پذیری رویکردهای پیام-محور استفاده

6. Decoupling
7. Application Service
8. Crosscutting Service
9. Administration Service
10. Logging

1. Role Membership Certificate
2. Loosely-Coupled
3. Orchestration
4. Choreography
5. Publish-Subscribe



شکل ۲: الگوی معماری سرویس‌های غنی داده-محور.

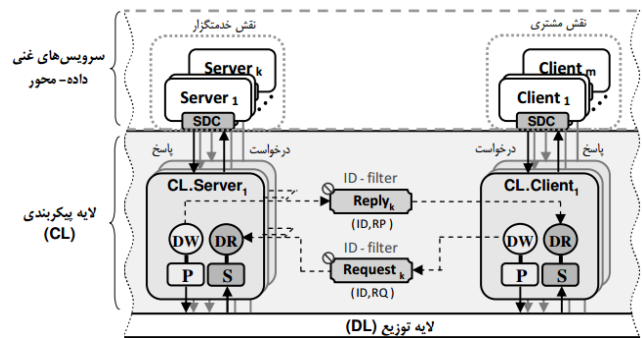
اتصال دهنده سرویس/داده در یک سرویس ساده یک واسط ساده است در حالی که این واسط در یک سرویس مرکب مانند یک سرویس دروازه^۱ مسئول مسیریابی پیام‌ها بین سطوح مختلف می‌باشد. این موضوع وابستگی بین سرویس‌ها و محل‌های مرتبط به آنها را در معماری منطقی از بین می‌برد.

جزئیات بیشتری از ساختار و رفتار سرویس‌های غنی داده-محور در سه زیربخش زیر مورد بررسی قرار می‌گیرد. زیربخش اول به چگونگی تعامل سرویس‌های غنی از طریق میان‌افزار پیشنهادی می‌پردازد، زیربخش دوم شامل فرآیندهای پیکربندی و بازپیکربندی سرویس‌های غنی داده-محور است و زیربخش سوم، خط‌مشی‌های کیفیت سرویس به کار گرفته شده در سرویس‌های غنی داده-محور را مورد بررسی قرار می‌دهد.

۴-۱ تعامل بین سرویس‌های غنی داده-محور

یک سرویس غنی داده-محور را می‌توان یک انتزاع از عملکردهای مختلف دانست که به صورت مجموعه‌ای از عملگرها بیان می‌شود. هر عملگر نمایانگر یک نوع تعامل درخواست-پاسخ است که سرویس مدیریت بین سرویس‌های مشتری و خدمتگذار آن عملگر تعریف می‌کند. شکل ۳ چگونگی تعامل درخواست-پاسخ بین سرویس‌های غنی داده-محور را که از طریق یک میان‌افزار داده-محور صورت می‌گیرد نشان می‌دهد. هر سرویس غنی داده-محور ممکن است چند نقش مشتری و خدمتگذار داشته باشد. سرویس‌های غنی موجود در سطوح مختلف سلسله مراتب می‌توانند از طریق میان‌افزار سطح خود و با کمک سرویس‌های دروازه مناسب با یکدیگر تعامل برقرار کنند. توزیع درخواست و پاسخ سرویس‌ها توسط لایه توزیع هر یک از میان‌افزارها صورت می‌گیرد. برای این منظور، سرویس‌های غنی داده-محور باید به صورت پویا و صحیح پیکربندی و بازپیکربندی شوند. این کار توسط لایه‌های پیکربندی و بر اساس اطلاعات پیکربندی و بازپیکربندی مربوطه صورت می‌گیرد. پیاده‌سازی تعامل درخواست-پاسخ بین سرویس‌های غنی با استفاده از میان‌افزار سرویس توزیع داده مزیت‌های زیر را به همراه خواهد داشت:

الف) شفافیت مکانی: سرویس‌های درخواست‌کننده و سرویس‌های پاسخ‌دهنده در هر جایی و به هر ترتیبی می‌توانند به میان‌افزار متصل شوند. این موضوع برای پیکربندی و بازپیکربندی پویای



شکل ۳: تعامل درخواست-پاسخ بین سرویس‌های غنی داده-محور.

سرویس‌های رویدادننگاری پیام‌های سرویس‌های کاربردی را مانیتور و ثبت کرده و سرویس‌های کنترل دسترسی می‌توانند سبب فیلترشدن یا مسیریابی مجدد آنها شوند. هر سرویس غنی داده-محور یک سرویس مدیریت دارد که اطلاعات پیکربندی و بازپیکربندی مربوط به آن سرویس غنی را تعریف می‌کند. این اطلاعات جهت پیکربندی و بازپیکربندی فراساختار ارتباطی به کار گرفته شده در سرویس غنی داده-محور مورد استفاده قرار می‌گیرد.

فراساختار ارتباطی به کار گرفته شده در سرویس غنی داده-محور یک میان‌افزار دولایه و داده-محور است. لایه بالایی این میان‌افزار که لایه پیکربندی نامیده می‌شود، برای پیکربندی و بازپیکربندی پویا و سازگار سرویس‌های غنی استفاده می‌شود. لایه زیرین میان‌افزار که لایه توزیع نامیده می‌شود مسئول ارتباطات پویا، مقیاس‌پذیر و بلادرنگ بین سرویس‌های غنی است.

سرویس‌های غنی داده-محور می‌توانند ساده یا مرکب باشند. یک سرویس ساده یک عملکرد ساده را ارائه می‌کند در حالی که سرویس‌های مرکب خود به سرویس‌های غنی دیگری تجزیه می‌شوند و می‌توانند عملکردهای مختلف و پیچیده‌ای را ارائه کنند. معماری سرویس غنی داده-محور، سرویس‌ها را در یک مدل ساختار سلسله‌مراتبی یکپارچه می‌کند. سرویس‌های غنی داده-محور از یکپارچه‌سازی افقی و عمودی پشتیبانی می‌کند. یکپارچگی افقی به وسیله اتصال مجموعه‌ای از سرویس‌های مرتبط به یک میان‌افزار دولایه و داده-محور به دست می‌آید. رهگیری و مسیریابی در این میان‌افزار بر اساس عنوان‌هایی صورت می‌گیرد که سرویس مدیریت تعریف می‌کند. یکپارچگی عمودی با استفاده از تجزیه هر سرویس به سرویس‌های دیگر به دست می‌آید که با استفاده از اتصال دهنده سرویس/داده به یکدیگر متصل می‌شوند.

دستکاری شود.

سرویس مدیریت با تعریف چندین نمونه از یک عنوان در یک سرویس غنی داده-محور می‌تواند وابستگی‌های مناسبی برای تعامل سرویس‌ها ایجاد کرده و از دریافت درخواست‌ها و پاسخ‌های نامربوط توسط آنها جلوگیری کند. فرض کنید که دو نمونه از سرویس خدمت‌گزار با نام‌های $Server_1$ و $Server_2$ وجود دارد که دارای اطلاعات بازیگر بندی متفاوتی هستند. در این حالت، سرویس مدیریت می‌تواند دو نمونه از عنوان Request با نام‌های $Request_1$ و $Request_2$ را تعریف کرده، به طوری که سرویس $Server_1$ فقط داده‌های مربوط به عنوان $Request_1$ و سرویس $Server_2$ فقط داده‌های مربوط به عنوان $Request_2$ را دریافت کنند. به این ترتیب درخواست‌های مربوط به سرویس $Server_1$ توسط سرویس $Server_2$ دریافت نخواهد شد.

۴-۱-۲ الگوریتم‌های ترکیب

لایه پیکربندی یک سرویس مشتری ممکن است برای یک درخواست چندین پاسخ دریافت کند. لایه پیکربندی باید پاسخ‌ها را ترکیب کرده و یک پاسخ نهایی به سرویس مشتری ارسال کند. برای این منظور، الگوریتم‌های ترکیب با اهداف زیر مورد استفاده قرار می‌گیرند:

الف) دسترس پذیری: برای به دست آوردن دسترس پذیری بالاتر پاسخ‌ها در سرویس‌های غنی داده-محور، درخواست یک سرویس غنی به چندین سرویس غنی افزونه فرستاده می‌شود. در این حالت، حتی با خرابی برخی از سرویس‌های افزونه احتمال دریافت یک پاسخ بسیار زیاد خواهد بود. در این مورد، الگوریتم ترکیب اولین پاسخ دریافت شده را به عنوان پاسخ نهایی برگردانده و سایر پاسخ‌ها را در نظر نمی‌گیرد.

ب) ترکیب: برخی مواقع پاسخ مورد نیاز برای یک سرویس غنی داده-محور شامل چند قسمت بوده و هر قسمت توسط یک سرویس خدمت‌گزار فراهم می‌شود. در این مورد، الگوریتم ترکیب تمام پاسخ‌ها را دریافت کرده و آنها را با یکدیگر ادغام می‌کند و به عنوان پاسخ نهایی برمی‌گرداند.

ج) سازگاری و جامعیت: زمانی که لایه پیکربندی یک سرویس مشتری، پاسخ‌هایی را از چندین سرویس غنی افزونه دریافت می‌کند، ممکن است مشکل سازگاری و مشکل عدم جامعیت در پاسخ‌ها رخ دهد. این مشکلات باید توسط الگوریتم ترکیب پاسخ‌ها بررسی و حل شود. برای این منظور حداقل n ($n \geq 2$) پاسخ، انتخاب شده و زمان مهر پیکربندی و همچنین مقادیر پاسخ‌ها با یکدیگر مقایسه می‌شود. اگر زمان مهر پاسخ‌ها با یکدیگر متفاوت باشد، به این معنی است که اطلاعات پیکربندی برخی از سرویس‌ها هنوز به روز رسانی نشده است و این یک مشکل سازگاری محسوب می‌شود. برای حل این مشکل، الگوریتم ترکیب باید پاسخ با زمان مهر جدیدتر را به عنوان پاسخ نهایی برگرداند. اگر زمان مهر پاسخ‌ها یکسان بوده و مقدار پاسخ‌ها متفاوت باشد به این معنی است که یک مشکل عدم جامعیت وجود دارد. الگوریتم ترکیب در این حالت، پیغام عدم جامعیت را برمی‌گرداند.

۴-۲ پیکربندی و بازیگر بندی سرویس‌های غنی

داده-محور

سرویس مدیریت در هر سرویس غنی داده-محور مسئول ایجاد و ارسال اطلاعات پیکربندی و بازیگر بندی به میان‌افزار ارتباطی سطح خود

سرویس‌های غنی داده-محور امری ضروری است. حالت میان‌افزار در هر جایی که سرویس‌ها به آن متصل می‌شوند در دسترس خواهد بود.

ب) مقیاس پذیری: نقش سرویس‌های درخواست کننده و نقش سرویس‌های پاسخ دهنده به صورت سست اتصال از هم قرار دارند. این موضوع سبب می‌شود که تعداد سرویس‌های هر نقش بتواند به صورت مستقل زیاد شده و مقیاس پذیری افزایش پیدا کند.

ج) دسترس پذیری: از آنجایی که سرویس‌ها به صورت سست اتصال و تنها از طریق نوشتن و خواندن عنوان‌ها با یکدیگر ارتباط دارند، پیاده‌سازی سرویس‌های افزونه آسان بوده و میزان دسترس پذیری به راحتی قابل افزایش است.

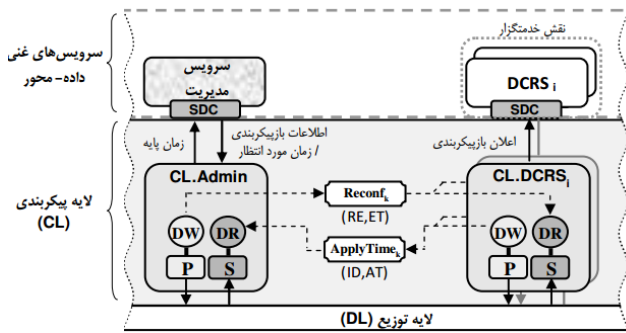
د) رویدادنگاری: با استفاده از میان‌افزار سرویس توزیع داده، رویدادنگاری بسیار راحت بوده و تنها از طریق تقاضا کردن و خواندن عناوین مربوطه توسط سرویس‌های رویدادنگاری صورت می‌گیرد.

ه) خط مشی‌های کیفیت سرویس: میان‌افزار سرویس توزیع داده از خط مشی‌های کیفیت سرویس متعددی پشتیبانی می‌کند. این خط مشی‌ها می‌توانند در موارد مختلفی از جمله کنترل زمان پاسخ سرویس‌ها مورد استفاده قرار گیرند.

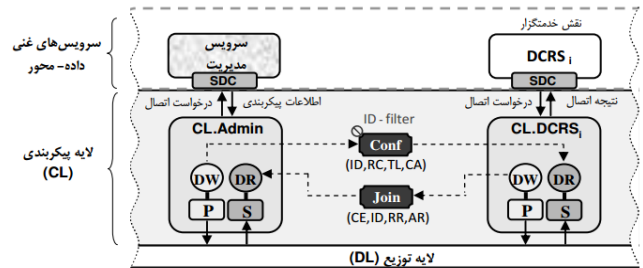
لیست عنوان‌ها و الگوریتم‌های ترکیب از مهم‌ترین عناصر اطلاعات پیکربندی و بازیگر بندی محسوب می‌شوند که در تعامل بین سرویس‌های غنی داده-محور مورد نیاز هستند. در ادامه به تشریح هر یک از این عناصر می‌پردازیم.

۴-۱-۱ لیست عنوان‌ها

در سرویس‌های غنی داده-محور، هر یک از درخواست‌ها یا پاسخ‌ها از طریق یک عنوان که دارای خصوصیت فیلتر بر روی شناسه سرویس است منتشر می‌شود. برای هر درخواست یک عنوان از نوع Request و برای هر پاسخ یک عنوان از نوع Reply در نظر گرفته می‌شود. تمام عنوان‌هایی که یک سرویس غنی داده-محور برای تعامل با سایر سرویس‌ها نیاز دارد به صورت لیستی از عنوان‌ها و در قالب اطلاعات پیکربندی و بازیگر بندی به آن سرویس غنی داده-محور ارسال می‌شود. لایه پیکربندی یک سرویس مشتری، یک درخواست را از طریق عنوان Request مربوطه منتشر کرده و منتظر دریافت پاسخ از طریق عنوان Reply متناظر با آن می‌شود. ویژگی فیلتر در عنوان Reply کمک خواهد کرد که لایه پیکربندی تنها پاسخ به درخواستی را دریافت کند که سرویس مشتری آن ارسال کرده است. لایه پیکربندی یک سرویس خدمت‌گزار، درخواست‌های سرویس‌ها را از طریق عنوان‌های Request دریافت کرده و پاسخ‌ها را از طریق عنوان‌های Reply متناظر با آنها منتشر می‌کند. ویژگی فیلتر در عنوان‌های Request کمک می‌کند که لایه پیکربندی یک خدمت‌گزار تنها درخواستی را دریافت کند که آن خدمت‌گزار قادر به پاسخگویی به آن درخواست است. لایه پیکربندی مربوط به هر سرویس مشتری یا سرویس خدمت‌گزار، قبل از ارسال هر درخواست یا پاسخ ممکن است تعدادی از سرویس‌های متداخل مانند کنترل دسترسی را فراخوانی کنند. با این کار ممکن است یک درخواست یا یک پاسخ فیلتر شود یا این که با انتخاب یک عنوان مناسب، مسیریابی مجدد صورت گیرد. استفاده از عنوان‌های Request و Reply در تعامل درخواست-پاسخ بین مشتری و خدمت‌گزار سبب می‌شود که حالت تعامل از منطق کاربردها منفصل شود. این حالت مستقل از موقعیت بوده و می‌تواند به صورت مستقل توسط سرویس‌ها کشف، ثبت، بازیابی و



شکل ۵: فرآیند بازپیکربندی برای دو سرویس غنی.



شکل ۴: فرآیند پیکربندی برای یک سرویس غنی.

جدول ۱: نقش‌های سرویس.

نوع سرویس	دنباله نقش‌های مورد نیاز
CR\	CR\۱), CR\۲), CR\۳), CR\۴)
SR\	SR\۱), SR\۲), SR\۳)

این درخواست، لایه پیکربندی اقدام به ایجاد یک نویسنده برای عنوان Join کرده و یک نمونه داده از این عنوان را که شامل چهار فیلد گواهی‌نامه (CE)، شناسه (ID)، نقش درخواستی (RR) و نقش‌های فعلی سرویس (AR) است می‌نویسد. بعد از نوشتن این نمونه داده، لایه پیکربندی اقدام به ایجاد یک خواننده برای عنوان Conf کرده و منتظر دریافت اطلاعات پیکربندی می‌شود. درخواست اتصال توسط لایه پیکربندی سرویس مدیریت با نام CL.Admin دریافت شده و به سرویس مدیریت تحویل داده می‌شود. سرویس مدیریت بر اساس خط‌مشی‌های OASIS و مشخصات درخواست دریافت‌شده، اطلاعات پیکربندی مناسب را تعیین و به لایه پیکربندی تحویل می‌دهد. سپس لایه پیکربندی یک نمونه داده از عنوان Conf را که شامل چهار فیلد شناسه (ID)، گواهی عضویت نقش (RC)، لیست عنوان‌ها (TL) و الگوریتم‌های ترکیب (CA) است می‌نویسد. فیلد شناسه در عنوان Conf از روی درخواست دریافت‌شده از عنوان Join مقداردهی می‌شود. از آنجایی که عنوان Conf بر روی شناسه سرویس فیلتر دارد، اطلاعات پیکربندی تنها به سرویسی که شناسه‌اش با مقدار فیلد شناسه در عنوان Conf یکسان باشد ارسال می‌شود. لایه پیکربندی سرویس DCRS_i با دریافت اطلاعات پیکربندی، نویسنده عنوان Join و خواننده عنوان Conf را پاک کرده و شروع به پیکربندی فراساختار ارتباطی مربوط به سرویس DCRS_i می‌کند. پس از تکمیل پیکربندی، نتیجه اتصال و پیکربندی به سرویس DCRS_i گزارش داده می‌شود.

۴-۲-۲ فرآیند بازپیکربندی

نقش‌های مربوط به یک سرویس غنی و همچنین اطلاعات پیکربندی مربوط به این نقش‌ها به صورت پویا توسط سرویس مدیریت تغییر می‌کنند و بنابراین سرویس‌های غنی نیز به بازپیکربندی پویا دارند. شکل ۵ فرآیند بازپیکربندی برای دو سرویس غنی داده-محور با نام‌های DCRS_i و DCRS_j که دارای یک نقش می‌باشند را نشان می‌دهد. اطلاعات بازپیکربندی و زمان مورد انتظار برای به کارگیری آنها توسط سرویس مدیریت تعریف شده و به لایه پیکربندی سرویس مدیریت ارسال می‌شود. زمان مورد انتظار، زمانی است که سرویس مدیریت انتظار دارد بازپیکربندی بعد از آن زمان به کار گرفته شود و بازپیکربندی قبل از آن زمان قابل قبول نخواهد بود. با دریافت اطلاعات بازپیکربندی، لایه پیکربندی یک داده از عنوان Reconf_k را که شامل دو فیلد اطلاعات بازپیکربندی (RE) و زمان مورد انتظار (ET) است می‌نویسد. داده نوشته‌شده عنوان Reconf_k توسط لایه پیکربندی سرویس‌های DCRS_i و DCRS_j خوانده شده و پس از زمان مورد انتظار به کار گرفته می‌شود. پس از به کارگیری بازپیکربندی توسط لایه‌های پیکربندی، یک پیغام به سرویس‌ها داده می‌شود. سپس هر یک از لایه‌های پیکربندی سرویس‌ها،

است. زمانی که یک سرویس به سرویس غنی داده-محور دیگر متصل می‌شود، لایه پیکربندی آن سرویس نیاز به دریافت اطلاعات پیکربندی اولیه از سرویس مدیریت موجود در آن سرویس غنی داده-محور خواهد داشت. از آنجایی که این اطلاعات بعد از فرآیند پیکربندی ممکن است تغییر کند، سرویس‌های غنی داده-محور به بازپیکربندی پویا و قابل اعتماد نیز نیازمند هستند. لایه پیکربندی هر سرویس غنی داده-محور، اطلاعات پیکربندی و بازپیکربندی مربوطه را بر اساس نقشی که آن سرویس غنی داده-محور دارد دریافت می‌کند. سرویس مدیریت برای تعریف نقش‌ها و انتساب آنها به سرویس‌های غنی داده-محور از خط‌مشی‌های مدل OASIS بهره می‌برد. تمام نقش‌های مربوط به یک سرویس غنی داده-محور در جدول نقش‌های سرویس نگهداری می‌شود. در این جدول، هر نوع سرویس یک نقش مفهومی دارد. نمونه‌ای از جدول نقش‌های سرویس در جدول ۱ نشان داده شده است.

در این مثال ساده، نقش مشتری ۱ (CR\۱) و نقش خدمتگزار ۱ (SR\۱) به ترتیب برای سرویس‌های مشتری ۱ و خدمتگزار ۱ در نظر گرفته شده است. بر اساس مدل OASIS، هر نقش مفهومی پارامتر جدیدی گرفته و نقش جدیدی را ایجاد می‌کند. برای مثال نقش‌های CR\۱(۱) تا CR\۱(۴) چهار نقش از نقش‌های مفهومی مربوط به سرویس مشتری ۱ هستند. جدول نقش‌های سرویس و قوانین-فعال‌سازی-نقش‌ها در مخزن اطلاعات پیکربندی و بازپیکربندی نگهداری می‌شود. قوانین-فعال‌سازی-نقش‌ها ساز و کاری را فراهم می‌کند که تنها سرویس‌های مجاز قادر به اتصال به یک سرویس غنی داده-محور بوده و اطلاعات پیکربندی و بازپیکربندی مناسب را دریافت می‌کنند. برای انتساب یک نقش به یک سرویس غنی داده-محور، یک گواهی-عضویت-نقش برای آن صادر می‌شود. یک سرویس باید تمام شرط‌ها و گواهی‌های تعریف‌شده در قوانین-فعال‌سازی-نقش مربوط به یک نقش را داشته باشد تا گواهی-عضویت-نقش مربوط به آن نقش را دریافت کند. یک سرویس با داشتن گواهی‌ها و شرط‌های لازم می‌تواند به چندین سرویس غنی داده-محور متصل شود. برای این منظور نیاز به یک مدیریت نامتمرکز اعتماد است. در [۳۱] جزئیات بیشتری در مورد به کارگیری از یک مدیریت نامتمرکز اعتماد مطرح شده است.

۴-۲-۱ فرآیند پیکربندی

شکل ۴ نمونه‌ای از فرآیند پیکربندی برای یک سرویس غنی داده-محور با نام DCRS_i را نشان می‌دهد. درخواست اتصال سرویس DCRS_i به لایه پیکربندی سرویس DCRS_j با نام CL.PDP_i ارسال می‌گردد. با

جدول ۲: خط مشی‌های کیفیت سرویس برای عنوان‌های به کار گرفته شده در سرویس غنی داده-محور.

خط مشی	مقدار خط مشی برای عنوان‌های		
	Conf و Join	Reply و Request	ApplyTime و Reconf
ماندگاری	پایا	ناپایا	پایا
تاریخچه	نگهداری همه داده‌ها	نگهداری آخرین داده	نگهداری همه داده‌ها
قابلیت اطمینان	قابل اطمینان	قابل اطمینان	قابل اطمینان
طول عمر	بی‌نهایت	LS	بی‌نهایت
تنظیم تأخیر	LB _p	LB _p	LB _p
اولویت انتقال	TP _p	TP _p	TP _p

ذخیره‌سازی شوند با خط مشی تاریخچه تعیین می‌شود. این خط مشی در عنوان‌هایی که شامل درخواست و پاسخ‌هایی با زمان پاسخ محدود هستند تعیین می‌کند که تنها آخرین داده منتشر شده باید نگهداری شود.

ج) خط مشی قابلیت اطمینان: این خط مشی، چگونگی قابلیت اطمینان در انتشار داده‌های عنوان‌ها را مشخص می‌کند. مقدار قابل اطمینان برای این خط مشی تعیین می‌کند که داده‌ها باید آن قدر ارسال شوند تا از گیرنده یک تأیید مبنی بر خوانده شدن داده‌ها دریافت شود.

د) خط مشی طول عمر: بازه زمانی که در آن یک داده معتبر است با این خط مشی تعیین می‌شود. این بازه برای عنوان‌های Request و Reply برابر با مقدار LS در نظر گرفته شده است. بنابراین درخواست‌ها و پاسخ‌های سرویس‌های غنی تنها در بازه زمانی LS معتبر خواهند بود.

ه) خط مشی تنظیم تأخیر: بیشترین زمان تأخیر برای داده‌های یک عنوان از زمانی که توسط سرویس ناشر نوشته می‌شوند تا زمانی که در اختیار سرویس متقاضی قرار می‌گیرند با این خط مشی تعیین می‌شود. فراساختار ارتباطی سه زمان تأخیر را برای عنوان‌ها تعریف می‌کند که نشان‌دهنده میزان فوری بودن^۸ داده‌های آنها می‌باشد. این سه زمان مطابق فرمول زیر تعریف می‌شوند

$$LB_p > LB_r > LB_s \geq 0 \quad (2)$$

و) خط مشی اولویت انتقال: این خط مشی به لایه توزیع این اجازه اجازه را می‌دهد که داده‌های عنوان‌های با اولویت مختلف را به صورت بهینه انتقال دهد. فراساختار ارتباطی سه زمان اولویت را برای عنوان‌ها تعریف می‌کند که نشان‌دهنده اهمیت^۹ انتقال داده‌های آنها می‌باشد. این سه اولویت مطابق فرمول زیر تعریف می‌شوند

$$TP_s > TP_r > TP_p \quad (3)$$

۵- ارزیابی کارایی سرویس‌های غنی داده-محور

در سرویس‌های غنی داده-محور پیشنهاد شده در این مقاله از میان‌افزار داده-محور و مدل OASIS به جای میان‌افزار پیام-محور و مدل نقش-

یک نمونه داده از عنوان ApplyTime_k را که شامل دو فیلد شناسه (ID) و زمان به کارگیری (AT) است می‌نویسند. با دریافت زمان‌های به کارگیری مربوط به یک بازپیکربندی، سرویس مدیریت کمترین آنها را به عنوان زمان پایه برای آن بازپیکربندی ثبت می‌کند. اگر سرویسی با پیکربندی قدیمی به درخواستی که زمان ارسال آن بزرگ‌تر از زمان پایه است پاسخ داده باشد یک مشکل ناسازگاری در سرویس غنی داده-محور به وجود می‌آید. همان طور که قبلاً اشاره شد الگوریتم‌های ترکیب می‌توانند برای رفع چنین ناسازگاری‌هایی به کار گرفته شوند.

در فرآیند بازپیکربندی ممکن است یک یا چند نقش از نقش‌های مربوط به یک سرویس غنی داده-محور حذف شود. در این صورت لایه پیکربندی بر اساس ساز و کاری که در مدل OASIS وجود دارد، گواهی-عضویت-نقش‌هایی که بر اساس نقش‌های حذف شده صادر گردیده است را حذف می‌کند.

۴-۳ خط مشی‌های کیفیت سرویس

خط مشی‌های کیفیت سرویس هر یک از عنوان‌های به کار گرفته شده در یک سرویس غنی داده-محور توسط سرویس مدیریت مربوطه تعریف شده و توسط لایه پیکربندی سرویس‌ها تنظیم می‌شود. این خط مشی‌ها در کنترل جنبه‌های مختلف توزیع درخواست و پاسخ سرویس‌ها به کار گرفته می‌شوند و پیچیدگی برقراری تعامل صحیح بین سرویس‌های غنی داده-محور را کاهش می‌دهند. خط مشی‌های مورد نیاز برای هر یک از عنوان‌ها و مقادیر مربوط به آنها در جدول ۲ نشان داده شده است. در ادامه به تشریح هر یک از این خط مشی‌ها می‌پردازیم:

الف) خط مشی ماندگاری:^۱ این خط مشی طول عمر داده‌های مربوط به هر عنوان را مشخص می‌کند. مقدار ناپایایی^۲ این خط مشی برای یک عنوان تعیین می‌کند که نیازی به نگهداری داده‌های منتشر شده آن عنوان برای تحویل به سرویس‌هایی که با تأخیر به یک سرویس غنی داده-محور متصل می‌شوند توسط لایه توزیع نیست. در مقابل، مقدار پایایی^۳ این خط مشی برای یک عنوان تعیین می‌کند که داده‌های منتشر شده آن عنوان توسط لایه توزیع نگهداری شده و به سرویس‌ها تحویل داده می‌شود حتی اگر سرویس‌ها خراب شده باشند و راه‌اندازی مجدد شوند.

ب) خط مشی تاریخچه:^۴ سرویس‌های غنی داده-محور روی گره‌های مختلفی که میان‌افزار روی آنها قرار دارد مستقر هستند. تعداد داده‌های منتشر شده هر عنوان که توسط هر یک از گره‌ها باید

5. Reliability Policy
6. Lifespan Policy
7. Latency-Budget Policy
8. Urgency
9. Transport-Priority Policy
10. Importance

1. Durability Policy
2. Volatile
3. Persistent
4. History Policy

جدول ۳: ارزیابی زمان پاسخ مربوط به درخواست از سرویس غنی داده- محوری که از RTI CONNEXT DDS استفاده می‌کند.

زمان پاسخ درخواست از سرویس غنی داده- محور	زمان اجرای الگوریتم‌های ترکیب در لایه پیکربندی اتصال‌دهنده	ماکسیمم زمان پاسخ درخواست‌های اتصال‌دهنده	زمان تأخیر پاسخ‌های سرویس‌های مربوطه به اتصال‌دهنده	زمان تأخیر درخواست‌های سرویس‌های مربوطه به اتصال‌دهنده	تعداد سرویس‌های مربوطه در سرویس غنی داده- محور
۴۵۳ μs	۲۰ μs	۲۵۰ μs	۹۱ μs	۹۱ μs	۱
۴۵۹ μs	۲۰ μs	۲۵۰ μs	۹۲ μs	۹۲ μs	۲۰۰
۴۶۱ μs	۲۱ μs	۲۵۰ μs	۹۲ μs	۹۲ μs	۴۰۰
۴۶۴ μs	۲۱ μs	۲۵۰ μs	۹۲ μs	۹۲ μs	۶۰۰
۴۶۶ μs	۲۲ μs	۲۵۰ μs	۹۲ μs	۹۲ μs	۸۰۰
۴۶۷ μs	۲۲ μs	۲۵۰ μs	۹۲ μs	۹۲ μs	۸۸۸

درخواست‌های اتصال‌دهنده توسط اتصال‌دهنده دریافت و ترکیب شده و پاسخ نهایی برگردانده می‌شود. زمان پاسخ هر یک از درخواست‌های اتصال‌دهنده که توسط سرویس‌های دیگر دریافت می‌شود بنا بر تعداد سطوح سلسله مراتبی که در سرویس غنی داده- محور قرار دارد به همین ترتیب و به صورت بازگشتی محاسبه می‌شود.

ب) عامل دوم، زمان‌های تأخیر عنوان‌های Request و Reply است که از طریق آنها درخواست‌های اتصال‌دهنده به سرویس‌های مربوطه ارسال شده و پاسخ‌های متناظر با آنها دریافت می‌شود. زمان تأخیر یک عنوان به مدت زمانی اطلاق می‌شود که طی آن نمونه داده‌های آن عنوان از ناشران آن عنوان به متقاضیان آن عنوان می‌رسند. این زمان به تعداد متقاضیان آن عنوان، حجم نمونه داده‌های آن عنوان و نرخ ارسال نمونه داده‌های آن عنوان بستگی دارد. باید توجه داشت که زمان پاسخ مربوط به درخواست‌های اتصال‌دهنده خود شامل زمان‌های تأخیر عنوان‌های Request و Reply که در سطوح زیرین سلسله مراتب قرار دارند هم می‌شود.

ج) عامل سوم، زمان اجرای الگوریتم‌های ترکیب و فعالیت‌های دیگری همچون مبادله پیغام‌ها است که در لایه پیکربندی سرویس غنی داده- محور صورت می‌گیرد. زمانی که سرویس غنی داده- محور تنها دارای یک سطح سلسله مراتب باشد زمان اجرا فقط به لایه پیکربندی اتصال‌دهنده آن سرویس غنی داده- محور مربوط می‌شود.

در حال حاضر، پیاده‌سازی‌های مختلفی از میان‌افزار سرویس توزیع داده موجود است که میان‌افزار RTI Connnext DDS یکی از آنها است که توسط RTI و متناسب با نیازهای سیستم‌های فوق مقیاس وسیع طراحی شده است [۳۳]. تحلیل‌های میان‌افزار RTIConnnext DDS نشان می‌دهد که تأثیر تعداد متقاضیان یک عنوان بر روی زمان تأخیر آن عنوان قابل قبول است. در این پیاده‌سازی با افزایش تعداد متقاضیان یک عنوان از ۱ به ۸۸۸ متقاضی، زمان تأخیر در آن عنوان حداکثر ۱۳٪ افزایش پیدا می‌کند. ما بخشی از تحلیل میان‌افزار ۴.۳ RTI Connnext DDS [۳۳] را برای شبیه‌سازی ۶ درخواست از یک سرویس غنی داده- محور با یک سطح سلسله مراتب استفاده کردیم. نتایج این شبیه‌سازی در جدول ۳ نشان داده شده است. در این شبیه‌سازی نرخ درخواست‌های اتصال‌دهنده برای هر درخواست ۱۰۰۰۰۰ درخواست در ثانیه و نرخ پاسخ‌های دریافت‌شده برای آن نیز ۱۰۰۰۰۰ پاسخ در ثانیه در نظر گرفته شده و حجم هر یک از درخواست‌ها یا پاسخ‌ها ۲۰۰ بایت لحاظ شده است. همچنین تعداد متقاضیان از ۱ تا ۸۸۸ متقاضی در نظر گرفته شده است. نرخ، حجم و تعداد متقاضیان در نظر گرفته شده برابر آن چیزی است که

مینا استفاده شده است. در سرویس غنی که از میان‌افزار پیام- محور استفاده می‌کند نیاز است که اتصال سرویس‌ها به صورت نقطه به نقطه برقرار شود. در این صورت اگر این سرویس غنی دارای n سرویس باشد به $n \times (n-1)$ اتصال نیاز خواهد داشت تا مطمئن شود تمام سرویس‌ها با یکدیگر ارتباط دارند. این موضوع زمانی که سرویس غنی شامل صدها سرویس می‌شود بسیار پیچیده خواهد بود. در سرویس‌های غنی داده- محور نیاز به اتصال مستقیم سرویس‌ها به یکدیگر نیست و هر سرویس تنها یک بار به میان‌افزار داده- محور متصل شده و عنوان‌هایی که باید بخواند و بنویسد برای آن سرویس تنظیم می‌شود. به این ترتیب پیچیدگی برقراری ارتباط بین سرویس‌ها و یکپارچه‌سازی آنها کاهش پیدا می‌کند. هر چند که مدل OASIS به کار گرفته شده در سرویس‌های غنی داده- محور مبتنی بر نقش هستند ولی نسبت به مدل نقش- مینا دارای چند برتری است که به مقیاس‌پذیری و پویایی یکپارچه‌سازی سرویس‌ها کمک می‌کند [۲۲] و [۲۳]. استفاده از مدل OASIS مزایای زیر را نسبت به مدل- نقش مینا در سرویس‌های غنی به همراه دارد:

الف) در هر سرویس غنی، نقش‌ها توسط سرویس مدیریت تعیین می‌شود و مدیریت متمرکز و سراسری بر روی اسم نقش‌ها و امتیازدهی‌ها وجود ندارد. هر سرویس مدیریت، نقش‌های خود را نامگذاری کرده و قوانین خود را برای فعال‌سازی هر نقش و درخواست سرویس مرتبط با آن نقش بیان می‌کند. این ویژگی خودمختاری و مقیاس‌پذیری سرویس‌های غنی را افزایش می‌دهد.

ب) نقش‌ها می‌توانند بر حسب نیاز پارامتر بگیرند. این ویژگی مقیاس‌پذیری و انعطاف‌پذیری سرویس‌های غنی را افزایش می‌دهد.

ج) مدل OASIS یک محیط فعال امنیتی را برای سرویس‌های غنی فراهم می‌کند. در چنین محیطی محدودیت‌های تعریف‌شده در طول فعال‌بودن نقش چک می‌شوند. در صورت نقض شدن یکی از این محدودیت‌ها، این موضوع با استفاده از سرویس توزیع داده به سرویس‌های مرتبط اعلان می‌شود و ممکن است منجر به غیر فعال شدن آن نقش شود. این ویژگی پویایی و امنیت سرویس‌های غنی را افزایش می‌دهد.

زمان پاسخ به یک درخواست در سرویس غنی داده- محور به مدت زمانی اطلاق می‌شود که آن درخواست توسط اتصال‌دهنده سرویس/ داده آن سرویس غنی دریافت شده تا زمانی که اتصال‌دهنده پاسخ نهایی مربوط به درخواست را برمی‌گرداند. زمان پاسخ به یک درخواست در سرویس غنی داده- محور به سه عامل زیر وابسته است:

الف) عامل اول، ماکسیمم زمان پاسخ به درخواست‌های اتصال‌دهنده است. این درخواست‌ها توسط اتصال‌دهنده و متناسب با درخواست دریافت‌شده به سرویس‌های مربوطه ارسال می‌شود. پاسخ به

- Foundations of Computer Software, Future Trends and Techniques for Development*, Springer Berlin Heidelberg, pp. 93-115, 2010.
- [5] B. Demchak, V. Ermagan, E. Farcas, T. Huang, I. H. Kruger, and M. Menarini, "A rich services approach to cocome," *The Common Component Modeling Example*, pp. 85-115, Springer Berlin Heidelberg, 2008.
- [6] M. Arrott, et al., "Rich services: the integration piece of the SOA puzzle," in *Proc. IEEE Int. Conf. on Web Services, ICWS*, pp. 176-183, Jul. 2007.
- [7] B. Demchak, et al., "Rich services: addressing challenges of ultra-large-scale software-intensive systems," in *Proc. of the 2nd Int Workshop on Ultra-Large-Scale Software-Intensive Systems*, pp. 29-32, May 2008.
- [8] M. Arrott, et al., "Integrating marine observatories into a system-of-systems: messaging in the US ocean observatories initiative," in *Proc. MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, 10 pp., Oct. 2009.
- [9] Y. Zhang, L. Duan, and J. L. Chen, "Event-driven SOA for IoT services," in *Proc. IEEE Int. Conf. on Proc. Services Computing, SCC'14*, pp. 629-636, Jun. 2014.
- [10] S. Hachem, A. Pathak, and V. Issarny, "Service-oriented middleware for the mobile internet of things: a scalable solution," in *Proc. IEEE Global Communications Conf., GLOBECOM'14*, 6 pp., Dec. 2014.
- [11] S. Rodriguez-Valenzuela, J. A. Holgado-Terriza, J. M. Gutierrez-Guerrero, and J. L. Muros-Cobos, "Distributed service-based approach for sensor data fusion in IoT environments," *Sensors*, vol. 14, no. 10, pp. 19200-19228, Oct. 2014.
- [12] N. Lee, H. Lee, W. Ryu, and K. Heo, "Web of object service architecture for device orchestration and composition," in *Proc. IEEE Int. Conf. on Information Science and Applications, ICISA'14*, 3 pp., May 2014.
- [13] L. Li, S. Li, and S. Zhao, "QoS-aware scheduling of services-oriented internet of things," *IEEE Trans. on Industrial Informatics*, vol. 10, no. 2, pp. 1497-1505, May 2014.
- [14] A. Giordano and G. Spezzano, "Service-oriented middleware for the cooperation of smart objects and web services," in *Internet of Things Based on Smart Objects*, Springer International Publishing, pp. 49-68, 2014.
- [15] F. Ning, D. Junhua, and G. Yan, "A Service Composition Environment Based on Enterprise Service Bus," in *Proc. of the 11th Int. Conf. on Ubiquitous Intelligence and Computing and 11th Int. Conf. on Autonomic and Trusted Computing and 14th Int. Conf. on Scalable Computing and Communications and Its Associated Workshops*, pp. 738-743, Dec. 2014.
- [16] P. Schauer and G. Debita, "Internet of things service systems architecture," in *New Trends in Intelligent Information and Database Systems*, pp. 239-248, Springer International Publishing, Mar. 2015.
- [17] OMG, Object Management Group. Data Distribution Service for Real-Time Systems Specification, version 1.2, 2007, www.omg.org/spec/DDS/1.2/
- [18] J. M. Lopez-Vega, J. Povedano-Molina, G. Pardo-Castellote, and J. M. Lopez-Soler, "A content-aware bridging service for publish/subscribe environments," *J. of Systems and Software*, vol. 86, no. 1, pp. 108-124, Jan. 2013.
- [19] A. Hakiri, P. Berthou, A. Gokhale, D. C. Schmidt, and G. Thierry, "Supporting SIP-based end-to-end data distribution service QoS in WANs," *J. of Systems and Software, Elsevier Science*, vol. 95, pp. 100-121, Sep. 2014.
- [20] S. Chae, S. Ahn, K. Kang, J. Kim, S. Lee, and W. Kim, "Fast discovery scheme using DHT-like overlay network for a large-scale DDS," in *Control and Automation, and Energy System Engineering*, Springer Berlin Heidelberg, pp. 128-137, 2011.
- [21] S. Shokrollahi, F. Shams, and J. Esmaeili, "Access control in ultra-large-scale systems using a data-centric middleware," *The ISC International J. of Information Security*, vol. 6, no. 1, pp. 3-22, Jul. 2014.
- [22] J. Bacon, D. M. Eyers, J. Singh, and P. R. Pietzuch, "Access control in publish/subscribe systems," in *Proc. of the 2nd Int. Conf. on Distributed Event-Based Systems*, pp. 23-34, Jul. 2008.
- [23] J. Bacon, K. Moody, and W. Yao, "A model of OASIS role-based access control and its support for active security," *ACM Trans. on Information and System Security*, vol. 5, no. 4, pp. 492-540, Nov. 2002.
- [24] S. Dustdar and W. Schreiner, "A survey on web services composition," *International J. of Web and Grid Services*, vol. 1, no. 1, pp. 1-30, Jan. 2005.
- [25] B. Demchak and I. Kruger, "Policy driven development: flexible policy insertion for large scale systems," in *Proc. IEEE Int. Symp. on Policies for Distributed Systems and Networks, POLICY'12*, pp. 17-24, Jul. 2012.

در تحلیل RTI آمده است. این برابری به ما این اجازه را می‌دهد که از زمان‌های تأخیری که RTI گزارش کرده است در شبیه‌سازی خود استفاده کنیم. جدول ۳ نشان می‌دهد که اگر سرویس‌های مربوطه در یک سرویس غنی داده-محور که درخواست‌های اتصال‌دهنده آن سرویس غنی را دریافت می‌کنند از ۱ به ۸۸۸ سرویس افزایش پیدا کند، زمان پاسخ آن سرویس غنی داده-محور حداکثر ۴٪ افزایش خواهد داشت. در میان‌افزار RTI Connex DDS یک عنوان جدید می‌تواند بدون آن که تأثیر منفی بر روی عملکرد سایر عنوان‌ها بگذارد به میان‌افزار اضافه شود و بنابراین ویژگی چندگانگی عنوان‌ها که در لایه توزیع میان‌افزار به کار گرفته شده است تأثیر منفی بر روی کارایی سرویس‌های غنی داده-محور نداشته و همان طور که قبلاً گفته شد می‌تواند کارایی آنها را بهبود ببخشد. در میان‌افزار RTI Connex DDS، فیلتر محتوی در سمت فرستنده صورت می‌گیرد. این موضوع سبب می‌شود که نمونه داده‌های یک عنوان به متقاضیان فیلترشده آن عنوان ارسال نشده و مقیاس‌پذیری سرویس غنی داده-محور افزایش پیدا کند.

۶- نتیجه‌گیری

در این مقاله، رویکرد سرویس‌های غنی داده-محور به عنوان رویکردی مناسب برای یکپارچه‌سازی سیستم‌ها در سیستم‌های فوق مقیاس وسیع پیشنهاد گردیده که در آن یک میان‌افزار دولایه و داده-محور جهت هم‌نوآوری سرویس‌های غنی ارائه شده است. در این میان‌افزار از سرویس توزیع داده برای ارتباط سرویس‌های غنی و از مدل OASIS جهت توصیف و انتساب نقش‌ها به سرویس‌های غنی استفاده گردیده است. به کارگیری میان‌افزار سرویس توزیع داده و مدل OASIS در توسعه سرویس‌های غنی سبب می‌شود که یکپارچه‌سازی سرویس‌ها از مقیاس‌پذیری، کشسانی، دسترس‌پذیری، پویایی و خودمختاری بالاتری برخوردار شود. در بخش ارزیابی، کارایی رویکرد پیشنهادی با استفاده از روش شبیه‌سازی مورد تحلیل و بررسی قرار گرفته است. نتایج حاصل از تحلیل و بررسی انجام‌شده نشان می‌دهد که کارایی رویکرد پیشنهادی در یکپارچه‌سازی سیستم‌های فوق مقیاس وسیع قابل قبول است. پیاده‌سازی سرویس‌های غنی داده-محور با استفاده از میان‌افزار سرویس توزیع داده OpenDDS از جمله کارهای آتی این تحقیق است.

۷- سپاس‌گزاری

از آنجایی که بخشی از این تحقیق در دوره فرصت مطالعاتی در آزمایشگاه امنیت و آزمایشگاه رایانش ابری در دانشگاه میلان صورت گرفته است، نویسندگان این مقاله از پروفیسور ساماراتی و پروفیسور پیوری بابت میزبانی و همکاری در انجام این تحقیق کمال تشکر را دارند.

مراجع

- [1] L. Northrop, et al., *Ultra-Large-Scale Systems: The Software Challenge of the Future*, Carnegie Mellon Software Engineering Institute, Ultra-Large-Scale Systems Study Report, 2006.
- [2] S. Herold, H. Klus, D. Niebuhr, and A. Rausch, "Engineering of it ecosystems: design of ultra-large-scale software-intensive systems," in *Proc. of the 2nd Int. Workshop on Ultra-Large-Scale Software-Intensive Systems, ACM*, pp. 49-52, May 2008.
- [3] K. Sullivan, W. Knaus, and R. Marks, "An ultra-large-scale systems approach to national-scale health information systems," in *Proc. of the FSE/SDP Workshop on Future of Software Engineering Research, ACM*, pp. 365-368, Nov. 2010.
- [4] C. Farcas, E. Farcas, and I. Kruger, "Requirements for service composition in ultra-large scale software-intensive systems,"

سعید شکرالهی تحصیلات خود را در مقطع کارشناسی کامپیوتر- نرم افزار در سال ۱۳۸۱ از دانشگاه اصفهان و در مقاطع کارشناسی ارشد و دکتری کامپیوتر- نرم افزار به ترتیب در سال های ۱۳۸۴ و ۱۳۹۳ از دانشگاه شهید بهشتی به پایان رسانده است. ایشان دوره فرصت مطالعاتی خود را در سال ۱۳۹۱ در آزمایشگاه امنیت دانشگاه میلان سپری کرده است. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: سیستم های فوق مقیاس وسیع، معماری نرم افزار، معماری سرویس گرا، معماری سازمانی، امنیت و کنترل دسترسی، اینترنت اشیاء و میان افزارهای مبتنی بر رویداد.

فریدون شمس تحصیلات خود را در مقطع کارشناسی کامپیوتر- نرم افزار در سال ۱۳۶۶ از دانشگاه شهید بهشتی، در مقطع کارشناسی ارشد کامپیوتر- نرم افزار در سال ۱۳۶۹ از دانشگاه صنعتی شریف و در مقطع دکتری کامپیوتر- نرم افزار در سال ۱۳۷۵ از دانشگاه منچستر انگلستان به پایان رسانده است. ایشان هم اکنون استاد دانشکده علوم و مهندسی کامپیوتر دانشگاه شهید بهشتی می باشد. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: سیستم های فوق مقیاس وسیع، معماری نرم افزار، معماری سرویس گرا، معماری سازمانی، اینترنت اشیاء و رایانش ابری.

جواد اسماعیلی تحصیلات خود را در مقطع کارشناسی علوم کامپیوتر در سال ۱۳۵۷ از مدرسه عالی برنامه ریزی و کاربرد کامپیوتر و در مقاطع کارشناسی ارشد و دکتری کامپیوتر- نرم افزار به ترتیب در سال های ۱۳۶۹ و ۱۳۷۴ از دانشگاه سالفورد انگلستان به پایان رسانده است. ایشان از سال ۱۳۷۶ تا ۱۳۸۵ ریاست دانشکده مهندسی برق و کامپیوتر دانشگاه شهید بهشتی را به عهده داشتند. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: پردازش های موازی و توزیع شده، تحلیل سیستم ها، طراحی و پیاده سازی زبان های برنامه سازی، بانکداری الکترونیک، امنیت و کنترل دسترسی.

- [26] B. Demchak, J. Kerr, F. Raab, K. Patrick, and I. H. Kruger, "PALMS: a modern coevolution of community and computing using policy driven development," in *Proc. IEEE 45th Hawaii Int. Conf. on System Science, HICSS'12*, pp. 2735-2744, Jan. 2012.
- [27] Q. Wei, M. Ripeanu, and K. Beznosov, "Authorization using the publish-subscribe model," in *Proc. IEEE Int. Symp. on Parallel and Distributed Processing with Applications, ISPA'08*, pp. 53-62, Dec. 2008.
- [28] T. Goovaerts, L. Desmet, and W. Joosen, "Scalable authorization middleware for service oriented architectures," in *Engineering Secure Software and Systems*, Springer Berlin Heidelberg, pp. 221-233, Feb. 2011.
- [29] R. Joshi, "Data-centric invocable services: a core design pattern for building scalable distributed real-time systems," in *Proc. of IEEE 15th Int. Symp. on Object/Component/Service-Oriented Real-Time Distributed Computing, IEEE Computer Society*, 7 pp., Apr. 2012.
- [30] F. Khodadadi, R. N. Calheiros, and R. Buyya, "A data-centric framework for development and deployment of internet of things applications in clouds," in *Proc. IEEE 10th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP'15*, 6 pp., Apr. 2015.
- [31] L. I. Pesonen, D. M. Eysers, and J. Bacon, "Access control in decentralised publish/subscribe systems," *J. of Networks*, vol. 2, no. 2, pp. 57-67, Apr. 2007.
- [32] RTI. *RTI Connex DDS*, 2014. Available: www.rti.com