مقاله پژوهشی

مجله انفورماتیک سلامت و زیست پزشکی
مرکز تحقیقات انفورماتیک پزشکی
دوره پنجم شماره چهارم. ص ۴۳۴-۴۲۳، ۱۳۹۷

پیشنهاد معماری مناسب سامانه‌های تصمیم‌یار پزشکی در حیطه مراقبت‌های پیچیده بیماری‌های مزمن: استفاده از میکروسرویس‌ها در مراقبت بیماران پیوند کلیه

پرویز رشیدی خزاعی[۱]، جمشید باقرزاده[۲]، زهرا نیازخانی[۳]، حبیب‌الله پیرنژاد[۳]*

• دریافت مقاله: ۹۷/۲/۲۲ • پذیرش مقاله: ۹۷/۶/۱۵

مقدمه: به کارگیری موفقیت‌آمیز سیستم‌های تصمیم‌یار پزشکی مبتنی بر دانش در حوزه مراقبت از بیماران پیوند کلیه می‌تواند در اتخاذ بهترین تصمیمات ممکن، کاهش هزینه و افزایش کیفیت مراقبت مؤثر واقع شود. در این بین، علاوه بر چالش‌های به کارگیری آن‌ها در حیطه مراقبت‌های بیماران مزمن، عدم وجود ساختار معماری مناسب چالشی است که به مرور زمان برجسته‌تر می‌گردد؛ لذا مطالعه حاضر با هدف پیشنهاد معماری مناسب برای این سیستم‌ها در حوزه پیوند کلیه انجام گردید.

روش: در این مطالعه موردی، از تجربه به دست آمده در طراحی سیستم مدیریت پیوند کلیه ارومیه استفاده شد و سپس با در نظر گرفتن زمینه‌های اصلی کاربرد این سیستم‌ها در حوزه پیوند کلیه و چالش‌های به کارگیری آن‌ها، ساختار معماری مناسبی برای آن‌ها با بهره‌گیری از آخرین پیشرفت‌ها درحوزه فناوری‌های توسعه نرم‌افزار در قالب میکروسرویس‌ها پیشنهاد شد.

نتایج: به ازای هر عملکرد مستقل این سیستم در حوزه پیوند کلیه، سه سرویس جداگانه جمع‌آوری داده‌ها، ساخت مدل بهینه و به‌کارگیری آن پیشنهاد شد. این سرویس‌ها با اجرا بر روی بسترهای فراهم شده می‌توانند به درخواست‌های مختلف کاربران پاسخ داده، دانش و اطلاعات مناسبی را تولید و در اختیار آنان قرار دهند.

نتیجه‌گیری: با بهره‌برداری از این ساختار، تیم‌های پزشکی و تحقیقاتی مختلف در سراسر کشور قادر خواهند بود فقط بر روی ساخت یک عملکرد خاص متمرکز شده و آن را به بهترین شکل ممکن انجام دهند. این ساختار پیشنهادی توسعه‌پذیر و مقیاس‌پذیر بوده و قادر است علیرغم افزایش حجم و نوع داده‌ها، افزایش کاربران و تعداد درخواست‌ها به‌راحتی پاسخگو باشد.

کلیدواژه‌ها: سیستم‌های تصمیم‌یار پزشکی، پیوند کلیه، میکروسرویس، معماری نرم‌افزار

• **ارجاع:** رشیدی خزاعی پرویز، باقرزاده جمشید، نیازخانی زهرا، پیرنژاد حبیب‌الله. پیشنهاد معماری مناسب سامانه‌های تصمیم‌یار پزشکی در حیطه مراقبت‌های پیچیده بیماری‌های مزمن: استفاده از میکروسرویس‌ها در مراقبت بیماران پیوند کلیه. مجله انفورماتیک سلامت و زیست پزشکی ۱۳۹۷؛ ۵(۴): ۴۳۴-۴۲۳.

۱. دانشجوی دکتری فناوری اطلاعات، کامپیوتر، مهندسی برق و کامپیوتر، دانشگاه ارومیه، ارومیه، ایران
۲. دکتری مهندسی نرم‌افزار، کامپیوتر، دانشیار، مهندسی برق و کامپیوتر، دانشگاه ارومیه، ارومیه، ایران
۳. دکتری تخصصی انفورماتیک پزشکی، دانشیار، فناوری اطلاعات سلامت، پیراپزشکی، علوم پزشکی ارومیه، ارومیه، ایران
* **نویسنده مسئول:** ارومیه، بلوار رسالت، کوی اورژانس، ستاد مرکزی دانشگاه علوم پزشکی ارومیه، مدیریت آمار و فناوری اطلاعات دانشگاه، کد پستی: ۵۷۱۴۷۸۳۷۳۴
• **شماره تماس:** ۰۴۴۳۲۲۴۰۶۵۸ • **Email:** pirnejad.h@umsu.ac.ir

[Downloaded from jhbmi.ir on 2024-04-20]

مقدمه

حوزه سلامت، امروزه درگیر مشکلات و مسائل بسیار زیادی است. از جمله این مشکلات هزینه‌های سرسام‌آور و افزایش مداوم آن‌ها، خطاهای پزشکی و کیفیت نامطلوب خدمات می‌باشند. این مشکلات در حوزه بیماری‌های مزمن و مراقبت‌های طولانی‌مدت مانند پیوند کلیه که بیمار بایستی مادام‌العمر تحت مراقبت‌های پزشکی پزشکان متخصص مختلفی قرار گیرد، حتی بیشتر نیز می‌باشد [۱]. در سال‌های اخیر، سیستم‌های انفورماتیک پزشکی مخصوصاً پرونده‌های الکترونیک سلامت و سیستم‌های تصمیم‌یار بالینی به‌عنوان یکی از راه‌های غلبه بر این چالش‌ها در دنیا مطرح بوده است. با این که این سیستم‌ها به دست‌اندرکاران این حوزه در کشورهای پیشرفته کمک شایانی نموده است؛ ولی پیچیدگی خدمات درمانی در کنار افزایش روزافزون تعداد، حجم و تنوع داده‌های پزشکی و اهمیت استفاده از آن‌ها برای کمک به پزشکان در اخذ بهترین تصمیم ممکن، بهره‌گیری از سیستم‌های تصمیم‌یار پزشکی را به منظور بهره‌برداری و ترکیب این اطلاعات با شواهد بالینی، نظرات متخصصین و نیز یافته‌های حاصل از اطلاعات سایر بیماران ضروری می‌نماید [۲]. متأسفانه این دسته از سیستم‌ها، هنوز در داخل کشور در ابتدای راه هستند.

دانش کاوی و سیستم‌های تصمیم‌یار مبتنی بر دانش در حوزه پزشکی (Knowledge Based Clinical Decisions Support System) KBCDSS یکی از راه‌های عملی و خوب مواجه با مشکلات و چالش‌های موجود در حیطه تصمیم‌گیری‌های پیچیده از جمله مراقبت‌های مزمن در حیطه پزشکی می‌باشد [۲]؛ اما استفاده مرتب از آن‌ها در مواجه با مشکلات پیچیده طبی با چالش‌های مختلفی روبه‌رو بوده است [۳]. اولین چالش به‌کارگیری مناسب KBCDSSها، عدم در دسترس بودن داده مناسب می‌باشد. داده‌ها و اطلاعات پزشکی یا به صورت کاغذی بوده و یا در بهترین حالت به صورت جداگانه در سیستم‌های بیمارستانی، مطب پزشکان، آزمایشگاه‌ها در مراکز مختلف کشور نگهداری می‌شوند. بخشی از این اطلاعات نیز در قالب سامانه‌هایی مانند سپاس در مرکز داده وزارت بهداشت و درمان جمع‌آوری می‌گردد. این اطلاعات بالقوه منابع بسیار باارزشی برای به دست آوردن اطلاعات و دانش نهفته می‌باشند. فراهم‌سازی امکان جمع‌آوری و یکپارچه‌سازی این دسته از اطلاعات، جهت استخراج دانش و اطلاعات نهفته و به‌کارگیری الگوریتم‌های داده‌کاوی، اولین قدم در به‌کارگیری این دسته از سامانه‌ها خواهد بود [۴]. چالش دیگر، تولید روزانه داده‌ها و اطلاعات متنوع در اندازه، اشکال و شکل‌های مختلف با عنایت به پیشرفت فناوری و دسترسی ارزان به ابزارهایی مانند سنسورهای مراقبت پزشکی و ... می‌باشند. ذخیره‌سازی، پردازش و به‌کارگیری همه این داده‌ها در یک بستر یکپارچه مستلزم به‌کارگیری روش‌های نوین ذخیره‌سازی داده‌های بزرگ و بهره‌گیری از پردازش ابری (Cloud Computing) است که فرصت‌های زیادی را جهت بهره‌برداری از آن‌ها فراهم می‌نماید [۵]؛ بنابراین به‌کارگیری سیستم‌های KBCDSS در داخل کشور، در دسترس نبودن بستر سخت‌افزاری مناسب (Central Processing Unit) CPU و (Graphics Processing Unit) GPU قوی و سریع) جهت پردازش حجم بسیاری زیادی از داده‌ها و به دست آوردن مدل‌های مناسب هست.

با فرض طراحی و به‌کارگیری KBCDSSها، چالش اصلی دیگر، عدم تطابق سریع آن‌ها با پیشرفت روزانه علم پزشکی است. این پیشرفت‌ها، باعث تغییر روش‌های تشخیصی و شیوه‌های درمانی شده و چه‌بسا باعث کنار گذاشتن شیوه‌ها و پروتکل‌های قدیمی درمان و جایگزینی آن‌ها با شیوه‌های نوین شوند [۶]. ازاین‌رو ساختار KBCDSSها بایستی به گونه‌ای باشند که راهکارهای جدید، دانش جدید و نظرات جدید متخصصین در سریع‌ترین زمان ممکن در سامانه‌ها قابل تعریف بوده و به‌روزرسانی گردند. چالش دیگر در ارتباط با این دسته از سیستم‌ها، ماهیت تحقیقاتی و تکامل تدریجی آن‌ها در طول زمان است. به این معنی که دانشمندان، محققین و متخصصین مختلفی در سراسر دنیا به مرور زمان و در طی سال‌های طولانی، روش‌ها و الگوریتم‌های مؤثرتر و یا تکمیل‌تری را برای حل یک مشکل ارائه می‌کنند تا جایگزین الگوریتم‌های کم اثر و یا ناقص قبلی شوند [۶]؛ بنابراین ساختار این برنامه‌ها بایستی به گونه‌ای باشند که تغییرات موردنیاز در موارد فوق به راحتی و در کمترین زمان ممکن قابل اعمال باشند. معماری کنونی سامانه‌های موجود به دلیل ساختار یکپارچه آن‌ها عملاً امکان پاسخگویی به چالش‌های مطرح شده را با مشکلات جدی مواجه می‌سازد. محققین مجبورند بر اساس همان فناوری قدیمی امکانات جدید را به برنامه‌های قبلی اضافه نموده و یا این‌که تمامی برنامه قدیمی را از نو بازنویسی نمایند که عملاً مقرون به صرفه نمی‌باشد. از این رو بایستی معماری سامانه‌ها به گونه‌ای باشد که هنگام به‌روزرسانی، قسمت‌های قبلی

[Downloaded from jhbmi.ir on 2024-04-20]

سامانه همچنان به کار خود ادامه داده و بخش‌های جدید که با فناوری جدید نوشته شده‌اند به صورت مستقل کار کنند. در این صورت یکپارچگی در سطح عملکرد بدون وابستگی به فناوری محقق می‌گردد.

ایجاد زیرساخت، فراهم‌سازی داده‌ها و طراحی ساختار مناسب برای سامانه‌های تصمیم‌یار مبتنی بر دانش می‌تواند از عوامل اصلی موفقیت و به کارگیری آن‌ها در سیستم سلامت باشند [۷]. پیشرفت فناوری اطلاعات منجر به ارائه ساختارهای نرم‌افزاری جدید مانند میکروسرویس‌ها شده است [۸]. ساختارهای مبتنی بر میکروسرویس‌ها می‌توانند در راستای غلبه بر چالش‌های فوق‌الذکر مثمر ثمر باشند. دلیل اصلی محبوبیت این ساختار، ایجاد Containerهای مناسب مانند Docker است که زیرساخت اصلی استقرار برنامه‌های مبتنی بر میکروسرویس‌ها را فراهم نموده است [۹]. در حوزه پزشکی بسترهای Bioboxes و BioDocker برای اشتراک‌گذاری داده‌های بیوانفورماتیکی طراحی شده است [۱۰]. میکروسرویس‌ها توسط شرکت‌های بزرگی مانند NetFlix، به صورت موفقیت‌آمیزی مورداستفاده قرار گرفته و بالغ بر ۶۰۰ سرویس مختلف را به مشتریان میلیونی ارائه می‌کند [۱۱]. در طراحی سامانه‌های سیستم اطلاعاتی اورژانس که شامل سرویس‌های مدیریت حادثه، انتقال و بستری در بیمارستان می‌باشد، میکروسرویس‌ها نقش بسزایی داشته‌اند [۱۲]. با گسترش اینترنت اشیاء و استفاده از سنسورهای پزشکی مانند سنسور دمای بدن، فشارخون و ضربان قلب و دریافت این اطلاعات توسط میکروسرویس‌ها، پزشکان می‌توانند با آگاهی از وضعیت بیمار اقدامات اورژانسی مناسبی را انجام دهند [۱۳]. این بستر امکان ایجاد محیط مشاوره تلفنی و اشترک تصاویر پزشکی بین رادیولوژیست‌های مختلف را فراهم نموده تا آن‌ها بتوانند به صورت مستقیم و بدون تأخیر زمانی با همدیگر ارتباط برقرار نموده و مشاوره‌های لازم را ارائه نمایند [۱۴]. به‌کارگیری این فناوری‌ها در حوزه انفورماتیک پزشکی دارای چشم‌انداز روشنی بوده و لازم است زمینه به‌کارگیری این فناوری‌ها در صنعت سلامت کشور و هوشمندسازی آن بیشتر موردتوجه قرار گیرد [۱۵،۱۶].

در صورت ایجاد سامانه یکپارچه KBCDSS برای کمک به پزشکان در حوزه مراقبت‌های طولانی‌مدت نظیر پیوند کلیه، درخواست‌های خیلی زیادی به این دسته از سامانه‌ها داده خواهد شد. افزایش درخواست‌ها، ممکن است منجر به افت کارایی سامانه و عدم پاسخگویی آن شود. از طرف دیگر، فرآیند آموزش و یادگیری مدل‌های هوش مصنوعی نیاز به بار پردازشی زیادی داشته و می‌تواند کل منابع سرور را درگیر نموده و امکان سرویس‌دهی به بقیه فعالیت‌ها را پایین آورد. ازاین‌رو ساختار پیشنهادی بایستی مقیاس‌پذیر بوده و با افزودن سرورها و امکانات جدید بتوان کارایی آن را بهبود داده و به تمامی درخواست‌ها در زمان قابل قبولی پاسخ داده شود. مطالعه حاضر بر آن است زمینه‌های مختلف عملکرد KBCDSSها در حوزه پیوند کلیه، چالش‌های اصلی ساخت و به‌کارگیری آن‌ها با استفاده از فناوری‌های قبلی و پاسخگویی به چالش‌های مطرح شده در سایه بهره‌گیری از فناوری‌ها و روش‌های کاربردی جدید مانند میکروسرویس‌ها را بررسی نموده و ساختار معماری مناسبی را برای طراحی KBCDSSها پیشنهاد نماید تا زمینه‌های به‌کارگیری و شکوفایی این دسته از سیستم‌ها را در صنعت سلامت کشور فراهم سازد.

روش

در این مطالعه موردی، از تجربه طراحی و ساخت سامانه مدیریت پیوند کلیه (Renal Transplant Management System) RTMS [۱۷] در مرکز پیوند کلیه بیمارستان امام خمینی (ره) ارومیه به عنوان یک مورداستفاده شد. ابتدا زمینه‌های به‌کارگیری و چالش‌های اصلی سامانه تصمیم‌یار مبتنی بر دانش در زمینه پیوند کلیه که اصطلاحاً RTMS_MicroCDSS نامیده شد، بررسی گردید. سپس، سبک‌های مختلف معماری نرم‌افزاری با تأکید بر معماری میکروسرویس‌ها در راستای طراحی ساختار پیشنهادی سامانه RTMS_MicroCDSS معرفی شد. برای این منظور علاوه بر بررسی متون، به شکل غیر سیستماتیک، از تجارب علمی و عملی نویسندگان این مقاله که تجارب زیادی در طراحی و ساخت نرم‌افزارهای مختلف در حیطه علوم پزشکی از جمله سیستم RTMS دارند نیز استفاده شد.

زمینه‌های اصلی کارکرد سامانه RTMS_MicroCDSS

KBCDSSها می‌توانند به عنوان یک ابزار کمکی در اختیار پزشکان قرار گرفته تا آن‌ها بتوانند بهترین تصمیمات ممکن را در سریع‌ترین زمان اتخاذ نمایند و خدمات درمانی با کیفیتی را برای بیماران فراهم نمایند [۱۸]. RTMS_MicroCDSS می‌توانند با بهره‌گیری از دانش و اطلاعات به دست آمده از مطالعات بین‌المللی و داده‌های بیماران بومی، در راستای تولید

[Downloaded from jhbmi.ir on 2024-04-20]

مدل‌های بومی به کار رفته و به افزایش کارایی سیستم سلامت کشور کمک نماید. به‌عنوان‌مثال وقتی بیمار مرگ مغزی جدید به‌منظور اهدای اعضای بدنش آماده می‌شود، RTMS_MicroCDSS می‌تواند از لیست بیماران موجود در لیست انتظار پیوند کلیه، با در نظر گرفتن خصوصیات دموگرافیک و بالینی آن‌ها و انجام مقایسه بین بیماران بهترین گزینه‌ها را پیدا نموده و به تیم پزشکی پیشنهاد نماید تا عمل پیوند کلیه بر روی یکی از آن‌ها انجام گیرد. با مطالعه و بررسی کارهای انجام شده در سطح بین‌المللی، زمینه‌های مختلفی از کارکرد KBCDSSها در جدول ۱ آورده شد [۱۹،۲۰].

جدول ۱: زمینه‌های اصلی کارکرد KBCDSSها

مرحله قبل از پیوند	مرحله پیوند (بستری)	مرحله پس از پیوند
– **تطبیق** بهترین گیرنده– اهداکننده – پیش‌بینی میزان بقای عضو پیوندی * – پیش‌بینی احتمال رد عضو پیوندی * – پیش‌بینی طول عمر بیمار بعد از پیوند * – پیش‌بینی میزان احتمال ابتلا به سایر بیماری‌های مزمن پس از پیوند مانند دیابت، بیماری‌های قلبی، عروقی و ... *	– تطبیق میزان دوز داروی سرکوب کننده ایمنی مصرفی * – پیش‌بینی میزان ابتلا به عفونت‌های حاد * – کنترل تداخلات دارویی * – کنترل Biomarkerها و اعلان هشدارهای مقتضی * کنترل تداخلات دارو–آزمایش *	– تطبیق هوشمندانه بهترین زمان مراجعه بعدی بر اساس نتایج آزمایش‌های قبلی و پروتکل درمانی – پیش‌بینی عوارض جانبی پس از پیوند مانند خطر ابتلا به سرطان و...

*– به این معنی است که این فعالیت در مرحله بعدی نیز کاربرد دارد.

با مطالعه لیست زمینه‌های اصلی کارکرد RTMS_MicroCDSS، مشاهده شد که هر فعالیت مستقل از دیگری می‌باشد؛ بنابراین لازم است برای انجام هر فعالیت تیم‌های تحقیقاتی و پزشکی مستقلی در کنار هم قرار گیرند تا ضمن انجام پژوهش‌های قوی، زمینه توسعه و ایجاد KBCDSS مناسب آن فعالیت را فراهم نمایند تا بعداً توسط تیم‌های توسعه نرم‌افزاری به سرویس مناسبی جهت استفاده پزشکان تبدیل گردند.

فرایند ساخت و به کارگیری سامانه‌های تصمیم‌یار

فرایند ساخت و به‌کارگیری سامانه‌های KBCDSS را به‌طورکلی به دو فاز می‌توان تقسیم نمود. الف) فاز ساخت و آموزش مدل و ب) فاز استفاده از مدل و دانش تولید شده [۲۱]. در فاز اول، این دسته از سامانه‌ها با دریافت تنظیمات و پیکربندی اولیه و نیز با به‌کارگیری الگوریتم‌ها، روش‌ها و تکنیک‌های مختلف هوش مصنوعی و داده‌کاوی سعی می‌کنند با پردازش داده‌ها، قواعد و قوانین پنهان در میان آن‌ها را یاد گرفته و الگوریتم‌ها و مدل‌های مناسبی را برای حل مشکل استخراج نمایند. برای این منظور معمولاً نیاز به توان پردازشی قوی و حافظه زیادی می‌باشد. به‌طوری‌که بعضی از پردازش‌ها ممکن است حتی چند روز طول بکشد. در طراحی و پیکربندی این دسته از سامانه‌ها، معمولاً یافته‌های جدید علمی، یافته‌های بالینی، نتایج حاصل از تحقیقات محققین، الگوریتم‌های جدید و ... مدنظر قرار می‌گیرند. با گذشت زمان، افزایش حجم داده‌ها و نیز پیشرفت‌های جدید علم، بر اساس صلاحدید مدیران و پزشکان بایستی فرآیند آموزش و ساخت مدل مجدداً انجام گیرد. در فاز دوم، با بهره‌گیری از مدل‌ها و الگوریتم‌های به دست آمده طی فاز آموزش، سامانه بایستی به درخواست‌های ارسالی از نرم‌افزارهای مطب پزشکان، (Hospital Information System) HIS، RTMS و یا برنامه‌های موبایل بیماران پاسخ دهد. به‌عنوان‌مثال، بیماران پیوندی می‌توانند با ورود اطلاعاتی مانند نتایج آزمایش‌های خود، درخواست بررسی وضعیت سلامتی کلیه خویش را داشته باشند و یا با ورود اطلاعات داروهای مصرفی، وضعیت تداخلات احتمالی دارویی را مشاهده و توصیه‌های لازم را دریافت نمایند.

کارکرد سامانه مدیریت پیوند کلیه (RTMS)

اصلی‌ترین مشکل به‌کارگیری سیستم‌های KBCDSS در سیستم سلامت کشور، عدم وجود داده مطمئن برای تصمیم‌گیری است. چه بسا داده‌های بسیار ارزشمندی از بیماران به شکل کاغذی هم وجود داشته باشند ولیکن به‌کارگیری این داده‌ها در راستای اهداف KBCDS عملاً ممکن نیست. در این بخش به‌صورت خلاصه قابلیت‌ها و نوع داده‌های جمع‌آوری‌شده توسط سامانه RTMS، به‌عنوان زیرساخت

[Downloaded from jhbmi.ir on 2024-04-20]

اصلی جمع‌آوری داده برای RTMS_MicroCDSS، معرفی شده است. این سامانه تمامی اطلاعات دموگرافیک و داده‌های بیماران از مرحله تصمیم‌گیری برای پیوند، بستری شدن در بیمارستان، انجام عمل پیوند تا مراقبت‌های بعد از آن را جمع‌آوری نموده و در بانک اطلاعاتی سامانه ذخیره می‌نماید و تمامی فرآیندهای درمان و مراقبت‌های بعد از آن را مدیریت می‌نماید. اطلاعات قبل از پیوند نیز در سامانه‌های مانند رجیستری بیماری مزمن کلیوی نگهداری می‌شود که در آینده می‌توان آن‌ها را نیز یکپارچه نمود [۲۲]. این اطلاعات به عنوان منبع بسیار باارزشی برای تصمیم‌گیری توسط پزشکان می‌باشند. اطلاعات و سوابق کاغذی مراجعات، نتایج آزمایش‌ها، داروهای تجویزی بیماران از حدود ۱۷ سال قبل تاکنون در سامانه فوق وارد شده است. این اطلاعات به صورت خلاصه در جدول ۲ نشان داده شد.

جدول ۲: اطلاعات و داده‌های سامانه مدیریت پیوند کلیه (RTMS)

قبل از عمل	حین بستری و عمل	بعد از عمل
اطلاعات دموگرافیک	اطلاعات و یافته‌های بالینی حین عمل و بعد از عمل	ثبت مراجعات بیمار بر اساس پروتکل
نتایج مشاوره‌های پزشکی	وضعیت عملکرد کلیه پیوندی	کنترل وضعیت ابتلا به سایر بیماری‌ها مانند دیابت، سرطان و ...
نتایج آزمایش‌ها	نتایج آزمایش‌ها	نتایج آزمایش‌ها
معاینات بالینی	معاینات بالینی	معاینات بالینی
داروهای مصرفی	داروهای مصرفی	داروهای مصرفی

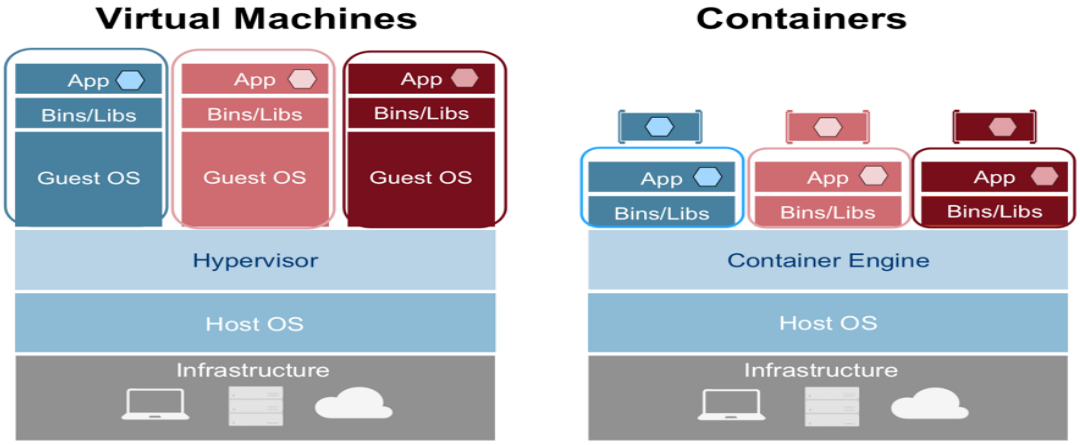
معرفی سبک‌های پرکاربرد معماری نرم‌افزار

تاکنون بر اساس پیشرفت فناوری و نیازهای کاربران، ساختارها و معماری‌های متفاوتی مانند شیءگرا (Object Oriented)، کامپونت‌گرا (Component Based)، سرویس‌گرا (Service Oriented) و ... ارائه شده‌اند که بسته به نیاز از هر کدام از آن‌ها برای طراحی ساختار نرم‌افزارهای یکپارچه سازمانی استفاده می‌شود [۲۳]. در نرم‌افزارهای یکپارچه (Monolithic)، معمولاً نرم‌افزار به صورت لایه‌ای طراحی و ساخته می‌شوند سپس تمامی لایه‌ها در کنار هم قرار گرفته تا سیستم را ساخته و خدمات موردنیاز را برای کاربران فراهم نماید. با بزرگ شدن نرم‌افزارها و به منظور افزایش قابلیت استفاده مجدد از اجزای مختلف برنامه‌های یکپارچه و کاهش هزینه‌ها، سبک معماری سرویس‌گرا (Service Oriented Architecture) SOA ارائه شده که بخش‌های مختلف برنامه‌ها به صورت سرویس‌های جداگانه‌ای نوشته شده که با استفاده از قوانین و استانداردهای مشخص با هم تبادل اطلاعات انجام می‌دهند. در واقع سرویس‌گرایی سبک مدرن‌تری از پیاده‌سازی برنامه‌های یکپارچه است [۲۳]. مثال عینی این کار، سرویس‌های پرداخت الکترونیک فراهم شده توسط بانک‌ها، برای شرکت‌های تجارت الکترونیک می‌باشند که بدون درگیرشدن در جزئیات، فقط از آن سرویس استفاده می‌کنند. پیشرفت فناوری، افزایش نیازها و خواسته‌های کاربران، افزایش بسیار سریع تعداد کاربران، کاهش زمان دسترسی به بازار، کاهش زمان توسعه نرم‌افزار، جوابگویی سریع به خواسته‌های جدید مشتریان در فضای رقابتی روز دنیا باعث شده است که شرکت‌های بزرگ نرم‌افزاری با بهره‌گیری از روش‌های جدید توسعه نرم‌افزار مانند (DevOps is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). DevOsp [۲۴] به فکر توسعه پلتفرم‌های جدیدی مانند Docker [۹] افتادند که بتوانند درخواستی‌های میلیونی کاربران را در کسری از ثانیه پاسخ دهند. این پلت فرم‌ها این امکان را برای گروه‌های توسعه نرم‌افزار فراهم نموده است که در طی فرایند ساخت نرم‌افزار، دیگر درگیر مشکلات فنی، تست، استقرار و به‌روزرسانی نباشند؛ زیرا این فعالیت‌ها به‌صورت خودکار در بستر ایجاد شده در دسترس می‌باشد و تیم پروژه تنها به توسعه قابلیت‌های موردنیاز مشتریان می‌اندیشند تا آن‌ها را در سریع‌ترین زمان ممکن در اختیار مشتریان قرار دهند. این امکانات باعث ایجاد سبک جدیدی در توسعه سامانه‌های نرم‌افزاری شده است که این سبک جدید را معماری میکروسرویس‌ها (Micro MSA(Services Architecture می‌نامند [۸].

در این سبک معماری، برنامه‌های بزرگ و توزیع‌شده نرم‌افزاری از ترکیبی از اجزای همکار به وجود می‌آیند و بر خلاف SOA، اجزا (سرویس‌ها) در سطح فرایند (Process) تعریف می‌شوند

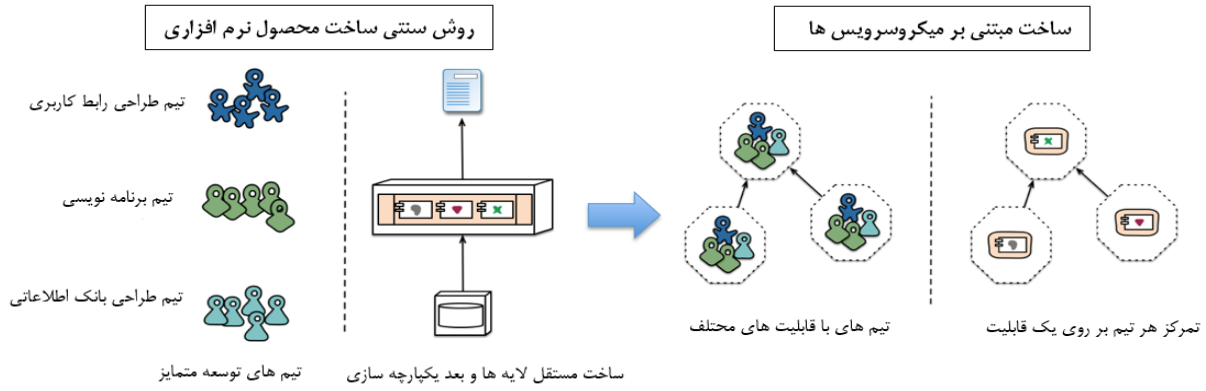
[Downloaded from jhbmi.ir on 2024-04-20]

و به اندازه کافی کوچک‌اند تا فقط یک خصوصیت و یا یک عملکرد از فرایند موردنظر را به درستی انجام دهند [۲۵]. بستر ایجاد شده در MSA ویژگی‌های متمایزی دارد که باعث محبوبیت روزافزون آن شده است [۸]. یکی از این ویژگی‌ها، فراهم‌سازی بستر مناسب جهت استفاده از Containerهایی مانند Docker توسط شرکت‌های بزرگ نرم‌افزاری می‌باشد. Container برخلاف ماشین مجازی، سبک وزن بوده و مستقیماً از امکانات سطح پایین سیستم عامل استفاده می‌کند و برای سرویس‌دهی نیازی به گذردهی دستورات از لایه‌های مختلف نیست به همین خاطر سرویس‌های سریع‌تری را در اختیار استفاده‌کنندگان قرار می‌دهد [۲۶]. شکل ۱ تفاوت ساختار ماشین مجازی با ساختارهای مبتنی بر Containerها را نشان می‌دهد.



شکل ۱: تفاوت ساختار ماشین مجازی با ساختارهای مبتنی بر Containerها [۲۶]

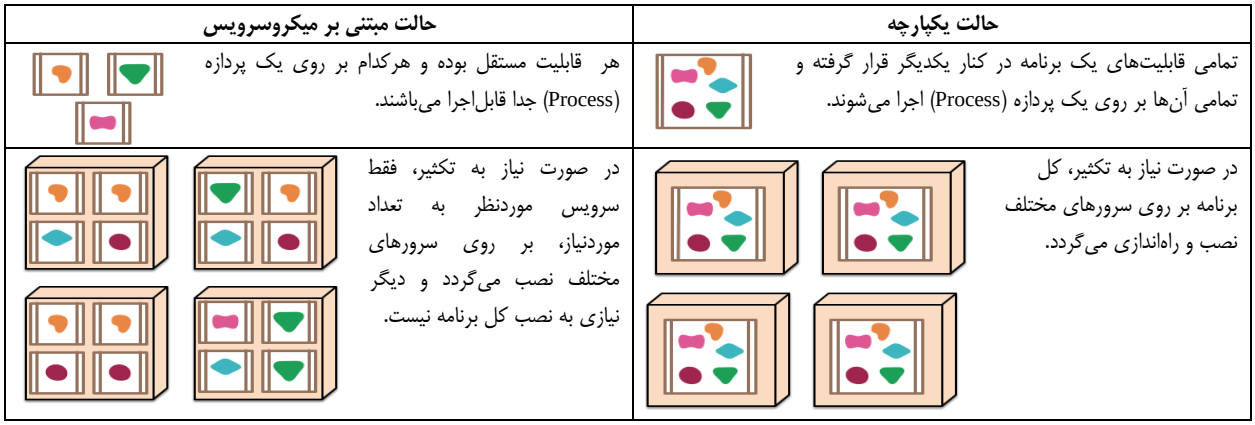
همان‌طور که در تصویر سمت راست شکل ۱ مشاهده می‌گردد در Containerها هسته و موتور آن مستقیماً با سیستم عامل ارتباط برقرار می‌کنند که خود این هسته نیز سبک‌وزن بوده و بر خلاف تکثیر در حالت ماشین مجازی، نیازی به نصب سیستم عامل جداگانه ندارد و همین ویژگی باعث افزایش سرعت اجرای سرویس‌ها می‌شود. به علاوه به دلیل استفاده کمتر از منابع سرور، امکان تکثیر و نصب نسخه‌های زیادی از یک سرویس را به راحتی فراهم می‌نماید. دلیل دیگر محبوبیت آن، تغییر شیوه توسعه برنامه‌های نرم‌افزاری مبتنی بر MSA در مقایسه با حالت سنتی می‌باشد [۲۷]. شکل ۲ تفاوت نحوه توسعه برنامه در روش‌های سنتی و روش مبتنی بر MSA را نشان می‌دهد.



شکل ۲: مقایسه فرایند تولید محصول نرم‌افزاری به روش سنتی با روش مبتنی بر میکروسرویس‌ها [۲۷]

[Downloaded from jhbmi.ir on 2024-04-20]

در روش تولید مبتنی بر میکروسرویس‌ها شیوه، تیم‌ها کوچک‌تر، مستقل‌تر و حرفه‌ای‌تر شده و از افرادی با جهت استفاده عملیاتی آن را در اختیار مشتری قرار می‌دهد. سپس تیم موردنظر به سراغ توسعه سرویس بعدی می‌رود درحالی‌که تیم‌های دیگر بر روی بقیه سرویس‌ها کار می‌کنند. این حالت باعث می‌شود که کیفیت ساخت اجزای محصول بالاتر رفته و به علاوه تحویل آن به طور پیوسته انجام می‌گیرد [۲۴]. در این روش نیازی نیست تیم‌های مختلف از یک نوع توان فنی و تبحر، مثلاً زبان برنامه‌نویسی، برخوردار باشند. این موضوع سبب افزایش انعطاف‌پذیری و سرعت بالای کار تخصص‌های مکمل تشکیل یافته و تمامی مراحل توسعه تا نصب و راه‌اندازی یک سرویس را تیم به تنهایی انجام داده و می‌گردد. با افزایش تعداد کاربران و برای پاسخگویی سریع، می‌توان با بهره‌گیری از بستر فراهم شده توسط زیرساخت Docker، نسخه‌های متعددی از یک سرویس را نصب و تحت یک پردازه جدا اجرا نموده تا به درخواست‌ها به سریع‌ترین شکل ممکن جواب داده شود. شکل ۳ تفاوت ساختاری یک برنامه یکپارچه با یک برنامه مبتنی بر میکروسرویس‌ها و نحوه مقیاس‌پذیری آن‌ها را توضیح می‌دهد [۲۷].

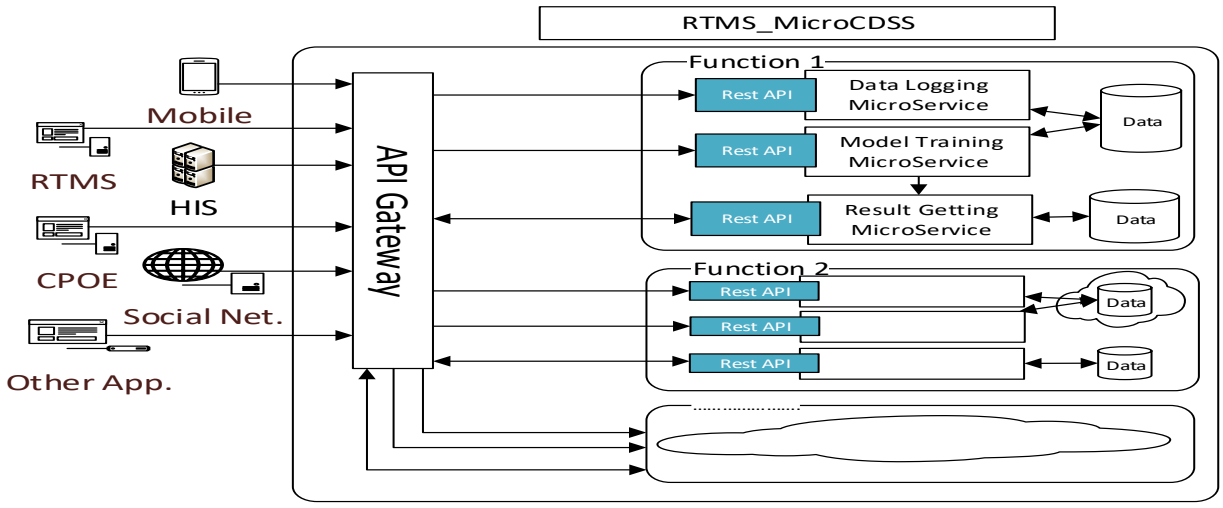


شکل ۳: تفاوت ساختاری برنامه یکپارچه با برنامه مبتنی بر میکروسرویس و نحوه مقیاس‌پذیری آن‌ها [۲۷]

نتایج

با بررسی کارکرد RTMS_MicroCDSS (جدول ۱) مشاهده شد تمامی فعالیت‌ها مستقل از همدیگر بوده و برای بهره‌برداری و پاسخگویی به خواسته‌ها و نیازهای کاربران می‌توانند به صورت تکی و مستقل اجرا شوند. این قابلیت کمک می‌کند که وقتی میزان درخواست‌ها زیاد شد، بتوان به راحتی آن را تکثیر و نسخه‌های مختلفی از آن را هم‌زمان اجرا نموده تا افت کارایی وجود نداشته باشد (شکل ۳). با بررسی نحوه ساخت و به‌کارگیری سیستم‌های داده‌کاوی و اینکه هر دو فاز قابلیت اجرای مستقل را دارند معماری پیشنهادی برای پیاده‌سازی RTMS_MicroCDSS بر اساس میکروسرویس‌ها به فرم شکل ۴ پیشنهاد شد.

[Downloaded from jhbmi.ir on 2024-04-20]



شکل ۴: ساختار معماری پیشنهادی سامانه RTMS_MicroCDSS بر اساس میکروسرویس‌ها

CDSS = Clinical Decision Support System, Rest API = RESTful Application Program Interface, RTMS = Renal Transplant Management System, HIS = Hospital information system, CPOE = Inpatient Computerized Provider Order Entry, NET = Network, App = Application.

در ساختار پیشنهادی برای هر قابلیت سامانه RTMS_MicroCDSS سه تا سرویس جداگانه در نظر گرفته شد:

۱) سرویس جمع‌آوری داده‌ها (Data Logging MicroService): وظیفه دریافت اطلاعات و ذخیره‌سازی آن در بانک اطلاعاتی را بر عهده دارد. مثلاً دریافت و ذخیره‌سازی نتایج آزمایش‌های بیمار از آزمایشگاه‌ها، HIS، RTMS و حتی برنامه موبایل بیماران.

۲) سرویس آموزش و ساخت مدل مناسب (Model Training MicroService): با استفاده از داده‌های جمع‌آوری شده، الگوها و مدل‌های مختلف هوش مصنوعی را آموزش داده و بهترین مدل را جهت استفاده ارائه می‌کند.

۳) سرویس استفاده از مدل و دریافت نتیجه (Result Getting MicroService): با آمدن بیمار و یا داده جدید با استفاده از مدل بهینه به دست آمده توسط سرویس ۲، دانش و اطلاعات مناسبی را تولید نموده و در اختیار گروه پزشکی قرار می‌دهد.

کاربران و افراد مختلفی، از پزشکان نفرولوژیست در مطب‌ها گرفته تا کاربران برنامه‌های موبایل، می‌توانند به عنوان کاربر سامانه از آن استفاده نمایند. برای دسترسی به تمامی سرویس‌ها لازم است کاربر و نرم‌افزار درخواست‌کننده مجوزهای لازم را داشته باشند. بررسی صحت مجوزها و سطح دسترسی کاربران، بایستی در تمامی سرویس‌ها پیاده‌سازی شوند که باعث افزایش حجم کار سرویس‌ها می‌شود. به جای انجام این کار در هر کدام از سرویس‌ها، لایه‌ای API Gateway اضافه شده است تا فعالیت‌هایی مانند کنترل سطح دسترسی، انتخاب سرویس مناسب بسته به نوع کاربر، نظارت بر اجرای سرویس‌ها و تقسیم بار بین نسخه‌های مختلف یک سرویس و ... را انجام دهد. این دروازه باعث می‌شود سرویس‌ها بیش از پیش ساده‌تر و سبک‌وزن‌تر شده و فقط عمل موردنظر را انجام دهند. درخواست‌ها از طریق پروتکل Rest به هر کدام از سرویس‌ها تحویل داده می‌شود. این پروتکل از ساده‌ترین و سبک‌وزن‌ترین شیوه‌های انتقال اطلاعات در بستر اینترنت بوده که باعث تسریع در انتقال پیام‌ها می‌شود. در سطح پایین‌تر، بانک اطلاعاتی برحسب نیاز می‌تواند از نوع رابطه‌ای (Structured Query Language) SQL، No-SQL و ... باشند و در صورت نیاز یک بانک اطلاعاتی می‌تواند به‌صورت اشتراکی در اختیار چند سرویس قرار گیرد. ازآنجایی‌که هر سرویس به‌صورت مستقل اجرا و ساخته خواهد شد تکنولوژی ساخت و زبان برنامه‌نویسی آن‌ها می‌تواند با هم متفاوت باشند. در این ساختار این امکان وجود دارد که بسته به نیاز فقط مخزن داده، سرویس و یا هر دو بر روی بستر cloud قرار گرفته تا سرویس‌های مناسب‌تری را فراهم نماید. با استفاده از امکانات فراهم شده توسط containerها تکثیر

[Downloaded from jhbmi.ir on 2024-04-20]

و مقیاس‌پذیری سرویس‌ها به راحتی انجام می‌گیرد و بسته به نیاز از هر سرویس به تعداد موردنیاز می‌توان ایجاد و اجرا نمود.

بحث و نتیجه‌گیری

در این مطالعه، کاربردی نمودن و استفاده از سامانه‌های تصمیم‌یار بالینی مبتنی بر دانش در حوزه پیوند کلیه، زمینه‌های اصلی کارکرد و چالش‌های به‌کارگیری آن بررسی شده و با بهره‌گیری از آخرین پیشرفت‌های تکنولوژی و بسترهای فراهم شده توسط شرکت‌های بزرگ فناوری اطلاعات، ساختار معماری مناسبی برای طراحی و ساخت سامانه RTMS_MicroCDSS مبتنی بر میکروسرویس‌ها ارائه گردید. این ساختار بر این اساس استوار است که هر سرویس به اندازه کافی کوچک شده تا فقط یک کار را انجام داده و آن را هم به‌درستی انجام دهد [۸]. با مشاهده لیست قابلیت‌های مورد انتظار از سامانه (جدول ۱)، مشاهده شد که می‌توان همگی سرویس‌ها را به صورت مستقل طراحی و اجرا نمود [۱۹]. برای توسعه هر کدام از قابلیت‌ها سه سرویس جداگانه جمع‌آوری داده‌ها، استخراج مدل بهینه و به‌کارگیری مدل تعریف شده است که پاسخگویی نیازهای محققان و کاربران مختلف می‌باشند. بهره‌گیری از این ساختار جدید می‌تواند به توسعه پلت‌فرم مناسب نرم‌افزارهای پزشکی و سامانه‌های پشتیبان تصمیم کمک نموده تا سرویس‌های بهتر و مناسب‌تری را با هزینه کمتری در اختیار داشته باشند [۱۸].

بر اساس نتایج پژوهش حاضر، به دلیل جدید بودن این فناوری فعلاً چنین ساختاری برای سامانه‌های تصمیم‌یار مشابه معرفی نشده است. این موضوع امکان مقایسه ساختار پیشنهادی این مطالعه با ساختارهای مشابه را غیر عملی می‌نماید؛ و لیکن برای روشن شدن موضوع این سامانه با سایر معماری‌های تولید نرم‌افزار از جمله ساختار لایه‌ای و یا سرویس‌گرایی مقایسه شد. لازم به ذکر است که هر کدام از سبک‌های معماری نرم‌افزار ویژگی‌های خاص خود را داشته و بسته به نوع برنامه‌ها بایستی از سبک معماری مناسبی برای آن‌ها استفاده نمود [۲۵]؛ که مقایسه آن با معماری‌های مشابه ذکر شد. امکان توسعه مستقل هر سرویس، اجرای هر سرویس بر روی پردازه (Process) جداگانه، استفاده از زبان برنامه‌نویسی، ابزارها و پلت‌فرم‌های متمایز برای توسعه هر سرویس، راحتی پاسخگویی به سایر نرم‌افزارها و کاربران نهایی، استقلال و جدا بودن تیم‌های توسعه، امکان کار هم‌زمان بر روی هر قابلیت در هر جایی از کشور توسط هر تیمی، تمرکز تیم بر روی ساخت سرویس بدون درگیر شدن در پیچیدگی‌های فناوری و راه‌اندازی آن به دلیل آماده بودن بستر، راحتی نصب نسخه‌های متعدد از یک سرویس برای پاسخگویی به درخواست‌های هم‌زمان کاربران مختلف از مزیت‌های این روش می‌باشد [۸]. مزیت بسیار مهمی دیگری که این ساختار برای تیم‌های توسعه فراهم می‌نماید قابلیت تغییرپذیری برنامه است به صورتی که در صورت نیاز به تغییر سرویس، بر خلاف ساختارهای یکپارچه، فقط همان سرویس اصلاح و به‌روزرسانی می‌گردد و بقیه سرویس‌ها به کارشان ادامه می‌دهند. حتی در صورت نیاز فناوری ساخت سرویس می‌تواند اصلاح شده و از فناوری جدیدتری استفاده شود و برخلاف ساختار یکپارچه، دیگر نیازی به اصلاح و بازنویسی کل پروژه نیست.

همچنین از دیدگاه کاربری پزشکی، استفاده از میکروسرویس‌ها مزیت‌های زیر را فراهم می‌نماید. سیستم‌های موجود در مطب تمامی پزشکان و نفرولوژیست‌ها با هر فناوری می‌تواند به‌راحتی به سرویس‌های سامانه RTMS_MicroCDSS وصل شده و از اطلاعات و پیشنهادات سامانه RTMS_MicroCDSS استفاده نمایند. از آنجایی که این ساختار قابلیت مقیاس‌پذیری را فراهم می‌نماید پزشکان پاسخ درخواست‌ها را خیلی سریع دریافت نموده و تأخیر ملموسی را مشاهده نخواهند کرد. به‌علاوه، می‌توان امکان به اشتراک‌گذاری داده‌های مختلف پرونده پزشکی بیمار، از تمامی مراکز کشور را در قالب یک بستر یکپارچه فراهم نمود [۶]. این ساختار از طریق به‌کارگیری سرورهای ابری، حجم بسیار زیاد داده‌های تولیدی توسط فناوری‌های جدید مانند سنسورهای پزشکی را مدیریت نموده و به کوچک شدن حجم برنامه‌ها و سیستم‌های پزشکی کمک می‌نماید و هزینه‌های به‌کارگیری این سیستم‌ها را برای دست‌اندرکاران حوزه سلامت و پزشکان کاهش می‌دهد [۵]. به علاوه برنامه‌های مبتنی بر موبایل و شبکه‌های اجتماعی بیماران می‌توانند از این سرویس‌ها برای کمک و بهبود شرایط زندگی آنان استفاده نماید.

علیرغم مزایای ذکر شده، این ساختار برای هر نوع برنامه‌ای مناسب نیست. به‌عنوان‌مثال با افزایش میزان پیچیدگی برنامه‌ها، تعداد سرویس‌ها هم زیاد می‌شوند و این موضوع باعث می‌شود مدیریت ارتباط آن‌ها با همدیگر و پیکربندی آن‌ها به‌راحتی مقدور نباشد. در حالاتی که نیاز به مدیریت تراکنش وجود داشته باشد و یا لازم باشد به‌طور هم‌زمان چندین رکورد در بانک‌های اطلاعاتی مختلفی به‌روزرسانی شوند و یا لازم باشد چهار خاصیت اصلی تراکنش‌های بانک

[Downloaded from jhbmi.ir on 2024-04-20]

اطلاعاتی رابطه‌ای (Atomicity, Consistency, Isolation, and Durability)ACID برقرار باشد، ساختارهای سرویس‌گرا چندان مناسب نمی‌باشد و نیاز به سربار بیشتری دارند [۲۵]. البته این مشکل در سامانه پیشنهادی در این پژوهش وجود ندارد؛ زیرا فقط اطلاعات به سامانه انتقال داده و ذخیره‌سازی می‌شود و ممکن است فقط نیاز به پاک‌سازی داده‌های دریافتی باشد که در آن صورت بایستی سرویس مناسبی هم برای آن طراحی نمود. بهره‌برداری از فناوری‌های مختلف در ساخت سرویس‌ها یکی از ویژگی‌های معماری مبتنی بر میکروسرویس‌ها می‌باشد. این موضوع باعث افزایش هزینه نگهداری برنامه‌ها در بلندمدت می‌شود. به علاوه پیدا کردن نیروهای متخصص که در تمامی مراحل مختلف توسعه نرم‌افزار حرفه‌ای باشند، کار راحتی نیست به همین خاطر ممکن است کار آن‌طوری که انتظار می‌رود پیش نرود. به‌طورکلی این ساختار برای سامانه‌های که سرویس و فرآیندهای آن‌ها وابستگی زیادی به هم دارند چندان مناسب نیست و بیشتر به کار برنامه‌هایی می‌خورند که بتوان اکثر سرویس‌های آن را به‌صورت مستقل طراحی نمود [۲۵]. مقاومت پزشکان و ارائه‌دهندگان خدمات بهداشتی –درمانی به‌عنوان چالش دیگری همچنان وجود خواهد داشت و نیاز به آموزش‌ها و زمان زیادی برای غلبه دارد [۲۸]. در نهایت از آنجایی که برای ارتباط بین سرویس‌ها در میکروسرویس‌ها از بستر شبکه استفاده می‌شود نسبت به برنامه‌های یکپارچه که بر روی حافظه سرور پردازش می‌شوند، کندتر هستند.

در این مطالعه ضمن بررسی زمینه‌های اصلی کارکرد سامانه RTMS_MicroCDSS و چالش‌های اصلی به‌کارگیری آن در صنعت سلامت کشور، ساختار مناسبی بر اساس میکروسرویس‌ها برای آن پیشنهاد شده است تا ضمن غلبه بر چالش‌های به‌کارگیری آن‌ها به توسعه مستقل قابلیت‌های مختلف RTMS_MicroCDSS توسط گروه‌های مستقل تحقیقاتی سراسر کشور کمک نماید. این ساختار باعث می‌شود هر گروه فقط بر روی یک قابلیت تمرکز داشته و آن را به‌درستی انجام دهد. برای هر قابلیت، سه سرویس ذخیره‌سازی داده‌ها، سرویس آموزش و انتخاب مدل بهینه و سرویس استفاده از مدل تعریف شده است که هر کدام از این سرویس‌ها بسته به نیاز می‌توانند به دفعات مختلف توسط کاربران مجاز فراخوانی و مورداستفاده قرار گیرند. بستر موجود این امکان را فراهم می‌کند که تیم توسعه نرم‌افزاری با دانش و فناوری‌های متفاوت، فقط به پیاده‌سازی قابلیت موردنظر پرداخته و درگیر مسائلی مانند نحوه نصب، پیکربندی، امنیت و نحوه تکثیر سرویس‌ها نباشند. این معماری برای سامانه RTMS_MicroCDSS ارائه شد که می‌تواند برای توسعه سیستم‌های پشتیبان تصمیم سایر حوزه‌ها نیز به کار رود. به‌علاوه می‌تواند به عنوان ساختاری برای توسعه سرویس‌های فراهم شده توسط وزارت بهداشت و آموزش پزشکی نیز در نظر گرفته شود.

تعارض منافع

این مقاله دارای تعارض منافع نمی‌باشد.

References

1. Kidney Disease: Improving Global Outcomes (KDIGO) Transplant Work Group. KDIGO clinical practice guideline for the care of kidney transplant recipients. Am J Transplant 2009;9 Suppl 3:S1-155.

2. Casimir P. Role of clinical decision support systems in improving clinical practice. MOJ Clinical & Medical Case Reports 2015;2(6):1-6.

3. Rajalakshmi K, Mohan SC, Babu SD. Decision support system in healthcare industry. International Journal of Computer Applications 2011;26(9):42-4.

4. Haynes RB, Wilczynski NL. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: methods of a decision-maker-researcher partnership systematic review. Implement Sci 2010;5:12.

5. Raghupathi W, Raghupathi V. Big data analytics in healthcare: promise and potential. Health Inf Sci Syst 2014; 2: 3.

6. Abdullah AL, Albeladi KS, AlCattan RF. Clinical decision support system in healthcare industry success and risk factors. International Journal of Computer Trends and Technolog 2014;11(4):188-92.

7. Schoff L. Comparing the challenges of implementing decision support systems in the healthcare and financial industries. Health Care and Informatics Review Online 2009;13(3):3-10.

8. Newman S. Building Microservices: Designing Fine-Grained Systems. 1th ed. USA: O'Reilly Media; 2015.

9. Docker. [cited 2018 Jun 1]. Available from http://www.docker.com.

[Downloaded from jhbmi.ir on 2024-04-20]

10. Bioboxes and Biodocker. [cited 2018 Jun 1] Available from (http://bioboxes.org) and (http://biodocker.org).
11. Mauro T. Adopting microservices at netflix: Lessons for architectural design. [cited 2018 Jun 1]. Available from: https://www.nginx.com/blog/microservices-at-netflix-architectural-best-practices/.
12. Martsenyuk VP, Andrushchak IY. On development of information system for emergency medicine based on microservices architecture. Medical Informatics and Engineering 2017; 30(1): 12-20.
13. Thiyagarajan S. Microservice architecture to monitor body temperature using internet of things. International Research Journal of Engineering and Technology 2018; 5(1): 506-9.
14. Andrikos C, Rassias G, Tsanakas P, Maglogiannis IG. An enhanced device-transparent real-time teleconsultation environment for radiologists. IEEE Journal of Biomedical and Health Informatics 2019; 23(1): 374 –86.
15. Williams CL, Sica JC, Killen RT, Balis UGJ. The growing need for microservices in bioinformatics. J Pathol Inform 2016; 7: 45.
16. Masuda Y, Yamamoto S, Shirasaka S. A Vision for Open Healthcare Platform 2030. 11th International KES Conference on Intelligent Interactive Multimedia: Systems and Services; 2018 Jun 20; Gold Coast, Australia: Springer Science and Business Media Deutschland GmbH; 2018. p. 175-85.
17. Niazkhani Z, Pirnejad H, Rashidi P, Makhdomi K, Taghizadeh Afshari A, Mohammadi Fallahi M, et al. Modeling and designing a computerized transplant information system for clinical management of kidney transplant patients in the Urmia Kidney Transplant Center, Iranian Journal of Kidney Diseases 2015; 9(S1): 34.
18. Oh S, Cha J, Ji M, Kang H, Kim S, Heo E, et al. Architecture design of healthcare software-as-a-service platform for cloud-based clinical decision support service. Healthc Inform Res 2015;21(2):102-10.
19. Khazaee PR, Pirnejad H, Bagherzadeh J, Niazkhani Z. Towards realizing benefits of information technology in organ transplant: a review. Studies in Health Technology and Informatics 2016; 226: 29-32.
20. Niazkhani Z, Pirnejad H, Rashidi Khazaee P. The impact of health information technology on organ transplant care: a systematic review. Int J Med Inform 2017;100:95-107.
21. Friedman E, Dunning T. Machine Learning Logistics: Model Management in the Real World. USA: O'Reilly Media; 2017.
22. Cheshme-Kabodi M, Pirnejad H, Makhdomi K, Nikibakhsh AA, Taghizadeh Afshari A, Niazkhani Z. Chronic Kidney Disease registry in Iran: aims, stakeholders, and minimum data set. 1st International Congress of Diseases and Health Outcomes Registry; 2017 Feb; Mashhad: Mashhad University of Medical Sciences; 2017. p. 30-1.
23. Chu SC. From component-based to service oriented software architecture for healthcare. Proceedings of 7th International Workshop on Enterprise networking and computing in Healthcare Industry; 2005 Jun 23-25; Busan, South Korea, South Korea: IEEE; 2005. p. 96-100.
24. Balalaie A, Heydarnoori A, Jamshidi P. Microservices architecture enables devops: migration to a cloud-native architecture. IEEE Software 2016; 33(3):42-52.
25. Richards M, Microservices VS. Service-oriented architecture. O'Reilly Media; 2015.
26. Seo KT, Hwang HS, Moon IY, Kwon OY, Kim BJ. Performance comparison analysis of linux container and virtual machine for building cloud. Advanced Science and Technology Letters 2014; 66:105-11.
27. Lewis J, Fowler M. Microservices [cited 2018 Jun 2]. Available from https://martinfowler.com/articles/microservices.html.
28. Anooj PK. Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules. Journal of King Saud University - Computer and Information Sciences 2012;24(1):27-40.

[Downloaded from jhbmi.ir on 2024-04-20]

Original Article

Journal of Health and Biomedical Informatics
Medical Informatics Research Center
2019; 5(4): 423-434

Proposing an Appropriate Architecture for Decision Support Systems in the Field of Complex Chronic Care: Micro-Services Based Software Architecture in Kidney Transplant Care

Rashidi Khazaee Parviz [1], Bagherzadeh Jamshid [2], Niazkhani Zahra[3], Prinejad Habibollah[3]*

• **Received:** 12 May, 2018 • **Accepted:** 6 Sep, 2018

Introduction: Development and successfully implementation of knowledge based clinical decision support system (KBCDSS) in kidney transplantation (KT) could support decision-making, reduce cost and improve quality of care. For practical use of these systems, however, many challenges have to be met. Besides to well-recognized challenges of design and implementation of information systems in healthcare domain, designing an appropriate architecture for KBCDSS that can address requirements of complex and multidisciplinary chronic care, is a new growing challenge.

Method: This case study was done based on the experience achieved in design of kidney transplantation management system in Urmia. Then, considering main applications and challenges of development and implementation of KBCDSS in Kidney transplantation and using the latest advancements in software technology field, appropriate software architecture based on micro-services has been proposed.

Results: For each function (process) of the KBCDSS, three different micro-services of data gathering, optimal model designing and its application were proposed. These services, through implementation on provided backgrounds, can respond to different requests of patients and provide them with adequate knowledge and information.

Conclusion: Use of this architecture makes it possible for different research teams with different skills and knowledge throughout the country to focus on designing a specific function and doing it in a best way. The proposed architecture can be extended and scaled. Moreover, in spite of increase in size and type of data, users and requests, it can respond well.

Keywords: Decision Support Systems, Clinical, Kidney Transplantation, Microservices, Software Architecture

• **Citation:** Rashidi Khazaee P, Bagherzadeh J, Niazkhani Z, Prinejad H. Proposing an Appropriate Architecture for Decision Support Systems in the Field of Complex Chronic Care: Micro-Services Based Software Architecture in Kidney Transplant Care. Journal of Health and Biomedical Informatics 2019; 5(4): 423-34.

1. Ph.D. Student in Information Technology, Electrical and Computer Engineering Dept., Urmia University, Urmia, Iran
2. Ph.D. in Software Engineering, Associate Professor, Electrical and Computer Engineering Dept., Urmia University, Urmia, Iran
3. Ph.D. in Medical Informatics, Associate Professor, Health Information Technology Dept., Urmia University of Medical Sciences, Urmia, Iran

***Correspondence:** Orjhans Street, Resalat Blvd, Urmia, Iran (Postal Code: 571478334)
• **Tel:** 04432240658 • **Email:** pirnejad.h@umsu.ac.ir

[Downloaded from jhbmi.ir on 2024-04-20]