

# احراز هویت به عنوان سرویس مبتنی بر رویداد راندگی برای یکپارچه سازی کاربردهای سازمانی

m.hashemi@eng.basu.ac.ir

معصومه هاشمی\*، دانشجوی کارشناسی ارشد مهندسی فناوری اطلاعات، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی

سینا، همدان

استاد راهنما: دکتر مهدی سخائی نیا، استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا

پاییز ۱۳۹۷

## چکیده:

تغییر نیازهای تجاری و فن آوری های جدید، افراد را وادار می کند تا رویکرد خود را از سیستم های کوچک و ایزوله به سوی یک سیستم بزرگ و یکپارچه تغییر دهند. یکپارچه سازی سیستم های ایزوله بخشی از معماری سرویس گرا (SOA) و معماری مبتنی بر رویداد (EDA) است. کاربرانی که قصد استفاده از سیستمهای امنیتی مانند احراز هویت را دارند، از هم مجزا و بالطبع ناهمگن هستند. عمل یکپارچه سازی باید در سیستم عاملهای مختلف و سخت-افزارهای مختلف به درستی و با سرعتی قابل قبول انجام گیرد. رویکرد SOA برای ادغام برنامه های موجود از راه حل های مختلفی استفاده میکند، هرچند رویکرد SOA در موارد امنیتی مانند احراز هویت نارسایی هایی دارد. منشا اصلی این نارسایی از ماهیت ارتباط همگام در SOA ناشی میشود. EDA یک طراحی معماری است که سرویس های مربوط کامپوننت های نرم-افزاری مستقل از طریق event notifications باهم ارتباط برقرار می کنند. هر کدام از معماری های گفته شده دارای ویژگی های مثبت و منفی می باشند، و معماری های سرویس گرا دارای مشکلاتی همچون، ارتباطات همگام و مشکلات ناشی از آن می باشد، از این رو، ما را بر آن داشت که به سمت معماری مبتنی بر رویداد برویم، که یک معماری با ویژگی ارتباطات غیر همگام با استفاده از رویداد است. در این مسیر قصد پژوهش در مورد فرایند احراز هویت به عنوان یک سرویس مبتنی بر رویداد گرای، را داریم.

واژه های کلیدی: احراز هویت، معماری سرویس گرا، معماری مبتنی بر رویداد.

## ۱. مقدمه:

با توجه به فرآیند احراز هویت که نیاز به استفاده از چند سرویس مختلف وجود دارد و نیازی نیست که کاربر منتظر پاسخ تک تک آنها بماند، زیرا مراحل باید پنهان از دید کاربر انجام شوند و فقط نتیجه به صورت موفقیت یا شکست به دست کاربر برسد، ارتباط همگام بین اجزا و سرویس ایده ی چندان مناسبی نیست. در ارتباطات ناهمزمان، برنامه فرستنده برای درخواستش را به یک برنامه گیرنده میفرستد و به وظیفه ی خود ادامه می دهد. به عبارت دیگر، درخواست فرستنده بستگی به برنامه گیرنده برای تکمیل پردازش آن ندارد. اگر برنامه های چندگانه به این شکل یکپارچه شده باشند، برنامه ارسال کننده می تواند پردازش خود را به اتمام برساند حتی اگر پردازش نهایی به پایان نرسیده باشد.

EDA از معماری Publish-subscribe برای برقراری ارتباط بین سرویس ها و کاربران نهایی استفاده می کند. این معماری سه جز اصلی دارد، این سه جزء عبارتند از: انتشاردهنده، رویداد، کارگزار broker رویداد و مشترک رویداد event subscriber یک انتشاردهنده، رویدادها را تشخیص می دهد و یک پیام به کارگزار رویداد ارسال می کند. کارگزار رویداد تمام رویدادهای ایجاد شده را جمع آوری می کند و آنها را به مشترکین علاقه مند می فرستد. در نهایت، مشترکین رویداد notification های رویداد را دریافت می کنند و به آن پاسخ می دهند. مشترکین رویداد میتوانند این رویداد را به سایر سرویسها هدایت کنند.

وقتی احراز هویت را با معماری SOA نجام میدهیم، این سرویس فقط از طریق فرمان کابر فراخوانده می شود و اگر مشکلی در ارتباط با سرویس ایجاد شود، حتی فرآیند قادر به آغاز شدن نیست. اما در معماری EDA هرگونه تغییر حالت در سیستم باعث ایجاد رویداد میشود، و عکس العمل مربوطه و فراخوانی سرویس مورد نظر انجام میشود.

در معماری SOA، از طریق مفهومی به نام ایزوله سازی داده ها از تکرار داده ها در مراحل مختلف احراز هویت، جلوگیری میشود. اما وابستگی بین داده ها ایجاد میشود، و استقلال سرویسها از هم ممکن نیست. اما در معماری EDA تکرار داده ها برای جلوگیری از وابستگی ها مطرح می شود و فرآیند هماهنگ سازی این داده های تکراری توسط رویدادها انجام میشود.

## ۲. مفاهیم پایه:

معماری سرویس گرا SOA یک روش طراحی برای سیستمهای نرم افزاری است که، اجزای نرم افزاری مستقل، خدمات را به کاربران نهایی و یا سایر اجزای نرم افزاری ارائه می دهند. یک سرویس را می توان به عنوان یک واحد از عملکرد تعریف کرد که مستقل است و می تواند به صورت پویا مورد استفاده قرار گیرد.

معماری مبتنی بر رویداد، عبارت معماری مبتنی بر رویداد اشاره به برنامه های کاربردی و خدماتی دارد که توانایی واکنش به تغییرات در شرایط را دارند.

## ۳. کارهای مرتبط:

Xiaorong و همکاران در برای حل مشکلات امنیتی احراز هویت در دامنه های متعدد و به هم پیوسته، مدلی را در SOA مبتنی بر Bus Service Enterprise پیشنهاد داده است. از CDAM-ESB به عنوان شخص ثالث در ارتباطات استفاده کرده است. در نهایت، مدیریت سیاست، و اعتبارسنجی مرکز ESB که هسته CDAM-ESB بود، به طور کامل توضیح داده شد تا توضیح دهد که CDAM-ESB می تواند با نزدیک شدن به SOA پیچیدگی های احراز هویت در دامنه های متعدد و به هم پیوسته را کمتر کند.

Jie Xu و همکاران در با در نظر گرفتن و پیاده سازی یک پروتکل احراز هویت برای حوزه های مختلف برای تعاملات پویا بین سرویس ها بر اساس SOA سعی در رفع مشکلات موجود در ارتباطات داینامیک هنگام احراز هویت داشتند.

Martin Wolf و همکاران در سوال مهمی را مطرح میکنند که آیا احراز هویت متحد شده می تواند در محیط های ناهمگن حاصل شود که شرکا از استانداردهای متفاوت استفاده می کنند؟ پس از ارزیابی استانداردهای و شناسایی شباهت ها، یک مدل بتا پیشنهاد کردند که احراز هویت متحد شده را در سطح انتزاعی توصیف می کند. آنها مدل را بر اساس پروتکل های استاندارد اعتبارسنجی و اجرا می کنند. هدف ما تأیید اعتبار فدرال در استانداردهای مختلف است.

## ۴. تحلیل عملکرد و ارزیابی:

همانطور که گفته شد، هر کدام از معماری های گفته شده دارای ویژگی های مثبت و منفی می باشند، و گفته شد معماری های سرویس گرا دارای مشکلاتی همچون، ارتباطات همگام و مشکلات ناشی از آن می باشد، از این رو، ما را بر آن داشت که به سمت معماری مبتنی بر رویداد برویم، که یک معماری با ویژگی ارتباطات غیر همگام با استفاده از رویداد است. و در نتیجه پیشنهاد استفاده از معماری EDA و استفاده از ESB برای پیاده سازی رخدادها، در فرآیند احراز هویت، باعث فائق آمدن بر مشکلات معماری SOA خواهد شد. و باعث می شود بتوانیم احراز هویت را به یک عنوان یک فرآیند جدا تعریف و پیاده سازی کنیم و سایر اپلیکشن ها وظیفه احراز هویت را به این سیستم بسپارند.

## ۵. نتیجه گیری:

با توجه به مسائل مطرح شده همچنان نیاز به بررسی معماری های سرویس گرا و مبتنی بر رویداد حس می شود. و با انجام پیاده سازی برای یک مساله با دامنه ای کوچک می توان طرز کار این معماری ها را به صورت عملی مشاهده کرد.

سیس نوبت به بررسی دقیق و جامع موضوعات هماهنگی سازمانی و احراز هویت می رسد. با بررسی دو مورد گفته شده، می توان مراحل احراز هویت را به صورت یکپارچه در قالب معماری رویداد محور پیاده سازی کرد.

## ۶. منابع:

Maréchaux, J.-L., Combining service-oriented architecture and event-driven architecture using an enterprise service bus. IBM Developer Works, 2006: p. 1269-1275.

McGovern, J., et al., Event-driven architecture. Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations, 2006: p. 317-355.

Michelson, B.M., Event-driven architecture overview. Patricia Seybold Group, 2006. 2.

Chandy, K.M., Event driven architecture, in Encyclopedia of Database Systems. 2009, Springer. p. 1040-1044.

Bertino, E. and L. Martino, Object-oriented database management systems: concepts and issues. Computer, 1991. 24(4): p. 33-47.

Bukhsh, Z.A., M. van Sinderen, and P.M. Singh. SOA and EDA: A comparative study: Similarities, differences and conceptual guidelines on their usage. in e-Business and Telecommunications (ICETE), 2015 12th International Joint Conference on. 2015. IEEE.

Pilawski, B., Event-driven architecture and soa-allies or enemies? TVORBA SOFTWARE, 2008. 2008: p. 140-147.