# مدل کنترل دسترسی مبتنی بر مستاجر معماری های چند اجاره ای و اجاره دست دوم در نرم افزار بعنوان سرویس


**چکیده**

نرم افزار بعنوان سرویس (SaaS) معماری چند اجاره ای (MTA) را معرفی می کند معماری اجاره در جاره (STA) یک بسط از MTA است که به مستاجر اجازه می دهد خدمات را به توسعه دهندگان مستاجر دست دوم برای شخصی سازی کاربردهایشان در زیر ساخت SaaS ارائه کنند. در یک سیستم SaaS مستاجرها می توانند مستاجر دست دوم ایجاد کرده و منابع خود را در اختیار مستاجر دست دوم قرار دهند. جداسازی روابط شراکت بین مستاجرهای والدـ فرزند، مستاجر خواهر ـ برادر یا دو مستاجر غیر مرتبط پیچیده‌تر از روابط بین مستاجرها در MTA است. خصوصی نگه داشتن داده یا مولفه های خدمات و در همان زمان به اشتراک گذاشتن آنها و حمایت از شخصی سازی کاربردها به مستاجران مهم است برای حل این مشکل این مقاله یک تعریف رسمی از یک مدل کنترل دسترسی مبتی بر مستاجر برای کنترل دسترسی مبتنی بر نقش مدیریتی (ARBAC) برای MTA و STA در SaaS های با منشاء خدماتی (به نام TMS-ARBAC) فراهم می کند. نواحی خود مختار (AA ) و درخت AA برای توصیف خود مختاری مستاجران شامل روابط جداسازی و اشتراک گذاری آنها ارائه شده است . عملیات مجاز سازی روی AA و استراتژی‌های مختلف تسهیم منابع تعریف شده اند تا یک طرح کنترل درسترس در مدلهای STA دو پیاده سازی شود. مدل TMS-ARBAC برای طراحی یک پلت فرم علم الکترونیک جغرافیا بکار رفته است.

**لغات کلیدی:** نرم افزار بهعنوان خدمات (SaaS) معماری چند مستاجری (MTA) معماری مستاجر دست دوم (STA) مدل کنترل دسترسی مبتنی به نقش (RBAC) مدل کنترل مبتنی به مستاجر.

## 1. مقدمه

محاسبات ابری دارای سه مولفه اصلی است: زیر ساخت بعنوان خدمات (IaaS) پلتفرم بعنوان خدمات (PaaS) و فرم افزار بعنوان خدمات (SaaS) معماری‌های چند مستاجری (MTA) اغلب در SaaS بکار می رود که در آن چندین مستاجر می توانند از پایگاه کد یکسان ذخیره شده در SaaS برای توسعه کاربردها استفاده کنند. یک برنامه مستاجر ممکن است تحت توسعه باشد در حالیکه SaaS برنامه مستاجر دیگری را بطرو همزمان اجرامی کند.

یک مستاجر می تواند یک برنامه یا یک نهاد سازمانی باشد یک برنامه مستاجر می تواند توسط چند کاربر نهایی سازمان به کار رود. امروزه سازمانهای بسیاری دارای زیر سازمان هستند، مثلاً یک تعاونی می تواند چند کمیته زیر مجموعه داشته باشد و این شرکت های تابعه در حالی که متفاوت از هم هستند دارای ملزومات مشترکی می باشند بعنوان مثال بانک ولزفارگو یک مستاجر سطح سرمایه گذاری Sales forece. Com است. که بیش از 2000 شعبه بانک حدود 270 هزار کارمند داشته و به 3600 مشتری در هر شعبه خدمات می دهد. یک شعبه ممکن است در آمریکا کار کند، در حالیکه دیگری در آسیا است و این دو طوری تنظیم شده اند که با قوانین علی سازگاری داشته باشند اما هر دو کارهای تجاری مهم یکسانی دارند. در این مورد شرکت ممکن است یک مستاجر باشد در حالیکه این دو شرکت تابعه مستاجر دست دوم هستند. این STA است تعمیمی از MTA بوه و در STA یک برنامه مستاجر می تواند شخصی سازی شود تا برنامه‌های مستاجر دست دوم را تشکیل دهد و مستاجران دست دوم می-توانند داده و نرم افزار را با مستاجران دست دوم خودشان یا مستاجران والد خود به اشتراک بگذارند. به لحاظ تکنیکی یک مستاجر دست دوم می‌تواند مستاجران دست دوم خودش را داشته باشد، امّآ مدیریت این مستاجران دست دوم ممکن است لحاظ شوند علاوه بر این یک MTA ممکن است. بعنوان یک مورد فرعی برای STA لحاظ شود که هیچ مستاجری، مستاجر دست دوم ندارد.

همانگونه که قبلاً بطور رسمی به عنوان STA مدلسازی شده برخی تامین کنندگان خدمات ابری مانند salesforce ، Netsuit و openStack از گروه کاربران و زیر مجموعه های برنامه روی MTA برای تمرین چند مستاجری سلسله مراتبی و ملزومات مختلف برنامه های آنها استفاده می کند. در MTA هر شرکت تابعه از هر مستاجر بعنوان یک مستاجر مستقل ساخته می شود که در این حالت به اشتراک گذاشتن منابع شخصی سازی شده یا شخصی سازی مجدد بین مستاجران مرتبط سخت است یا می تواند بعنوان کاربر این مستاجر باشد که روابط سلسله مراتبی جهان واقعی جداسازی منابع و روابط اشتراک گذاری با تخصیص نقش و محدودیت های پیچیده اعمال می شود.

STA یک معماری با انعطاف پذیری و بسط پذیری بیشتر برای برنامه های چند مستاجری سلسله مراتبی است. در STA چندین مستاجر همزیستی دارند. هر مستاجر باید خود مختار بوده و بتواند دسترسی به منابع خود شامل داده و مولفه های خدمات خصوصی را برای مستاجران خود فراهم کند. هر مستاجر دست دوم ممکن است که فقط منابع مستاجر خود را به ارث نبرد و حتی برنامه های خودش را شخصی سازی کند. و دسترسی به منابع آنرا به دیگران آزاد یا ممنوع کند. مستاجران خواهر ـ برادر نیز ممکن است مولفه خدمات خود را با یکدیگر به اشتراک بگذارند.

در حال حاضر بسیاری از مدل های کنترل دسترسی موجود برای MTA مبتنی بر RBAC یا ARBAC هستند این مدلها می تواند به سه دسته تقسیم شوند: 1) با استفاده ازطرح و استراتژی این در ابرهای دیتا محور بدون لحاظ کردن اشتراک گذاری مولفه خدمات (2) با اضافه کردن انواع مختلف سلسله مراتب به محدودیت ها یا جداسازی دانه مدیریت برای جداسازی مستاجر چند گانه و (3) اضافه کردن فدرال های مستاجر ـ صادر کننده برای اشتراک گذاری مستاجر متقابل مولفه های برون سپاری

علاوه بر این استراتژی های اسمن فراهم شده با MTA کنترل دسترسی STA باید مسائل زیر را حل کند:

1) اشتراک گذاری حریم شخص: مستاجران و مستاجران دست دوم آنها ممکن است در داده و مولفه های خدمات خصوصی شریک باشند. یک مستاجر می تواند منابع خودش شامل داده و مولفه های مشخص شده را در اختیار مستاجران دست دوم خود قرار دهد و همچنین می تواند دسترسی مستاجران دست بالا و مدیران سیستم به آنها را

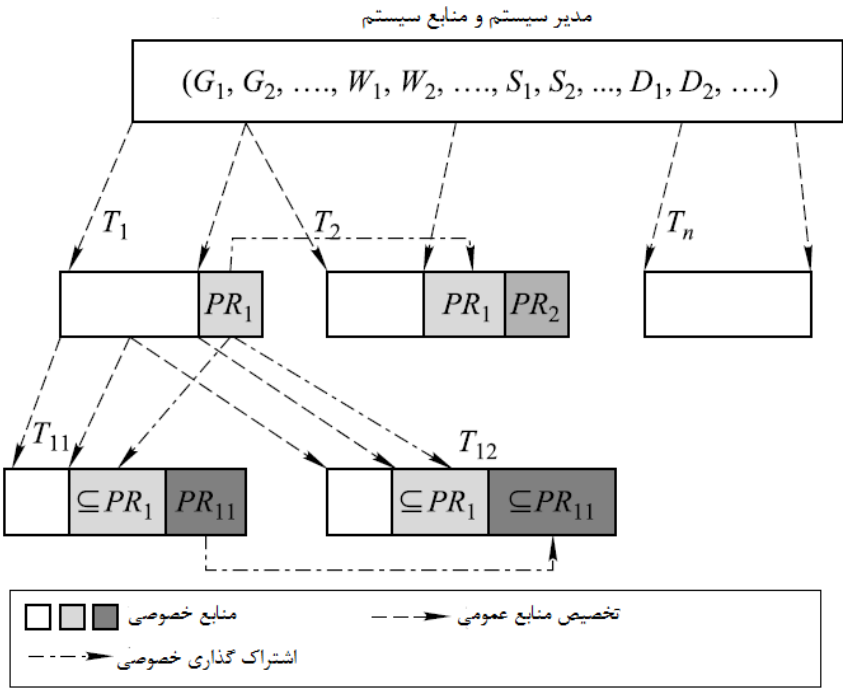
قطع کند. بعنوان مثال، شکل 1) نشام می دهد که منابع خصوصی مستاجر T1 ($PR_1$) بطور جزئی با مستاجر دست دوم T1 شریک شده است. یک مستاجر می هراسد مولفه های خصوصی خود را با خواهر ـ برادر خود شریک شود بعنوان مثال $PR_1$ توسط مستاجر برادر ـ خواهر T2 به اشتراک گذاشته شده است و منابع خصوصی T11 ، PR11 توسط مستاجر دست دوم برادر (هم رده) اش T12 به اشتراک گذاشته شده است.

2) مستاجران خود مختار: با توجه به اعطا کردن مزایا به مستاجران دست دوم ، مستاجران همانند عوامل خود مختار عمل می کنند حتی اگر عملیات آنها توسط زیر ساخت SaaS محدود شده باشد. یک مستاجر منابع خود را مدیریت کرده و می تواند مستاجران دست دوم خود را ایجاد کرد و اجازه دسترسی به منابع خودش را به آنها اعطا کند یک مدیر سیستم می تواند مستاجران را ایجاد کرد و منابع را به آنها اجاره دهد اما نمی تواند در امور داخلی مستاجران دخالت کند علاوه بر این بعلت جداسازی حریم خصوصی به ارث بردن مزایای نقش در دانه سیستم وجود ندارد. اینکه چه مزیت دسترسی می تواند یا نمی تواند بر ارث برسد یا اینکه چه منابعی یا نمی تواند بطور سطح متقابل کنترل شود متفاوت از سیستم های MTA سنتی است.اینها باید دوباره تعریف شوند.

3) روابط اشتراک گذاری بین مستاجران: اشتراک گذاری ممکن است بین هر مستاجر برادر (مانند T1 و T2 یا T11 و T12 در شکل 1) یا مستاجران والد ـ فرزند (مانندT1 به T11 در شکل 1) باشد اشتراک گذاری منابع شامل همه انواع منابع از قبیل مولفه برنامه ها و داده در SaaS خدمات محور است مولفه های منابع خصوصی می تواند برای سایر مستاجران برای دسترسی با اجازه مالک خود اعطا شود. جهات اشتراک گذاری ممکن است از یک به مستاجر والد به مستاجران دست دوم یا از یک مستاجر دست دوم به مستاجر والد یا از یک مستاجر به برادر خودش باشد.

4) مولفه های به اشتراک گذاشته شده: مولفه هایی مانند GUI ها، گردش کارها، خدمات و مولفه های دیتا می تواند به اشتراک گذارد شود اما مولفه های مختلف با ویژگی های دسترسی مختف باید متمایز شوند.

و یک مولفه ممکن است یک مولفه ترکیب، با مولفه های فرعی ازتامین کننده های مختلفی یا انتقالی از مستاجران مختلف باشد. مدیر سیستم ممکن است تنها مالک منابع نباشد یک مستاجر می تواند یک تامین کننده مولفه و یک اجاره دهنده برنامه باشد کنترل دسترسی مولفه های مرکب مشترک پیچیده است.




شکل 1روابط مدیر، مستاجر و مستاجر دست دوم.

برای حل این مشکلات ضمن گرانی برای ویژگی های اساسی مکانیزم های اجازه دهی ابری شود یک مدل اجازه دهی و مجاز سازی انعطاف پذیر، غیر متمرکز و قابل مقیاس دهی برای مستاجران با سطوح مختلف برای به اشتراک گذاری منابع شخص سازی برنامه های خود و حفظ رازهایشان باید پشتیبانی شود.

نکات مهم این مقاله به شرح ذیل است.

1) یک مدل کنترل دسترسی مبتنی بر مستاجر (به نام TMS-ARBAC) بر اساس ARBAC برای STA ارائه شده است همانگونه که STA شامل MTA است برای MTA نیز لحاظ می شود.

2) استراتژیهای مختلف کنترل دسترسی اشتراک منابع با توجه به مدل سازی دو سطحی TSTA) STA) تعریف شد و این ها می تواند برای ایجاد مسئولیت و مجوز برای هر مستاجر باشد.

3) طرح کنترل دسترسی برای یک داده علم الکترونیکی جغرافیا و پلتفرم اشتراک ابزارها براساس زیر ساخت Easy SaaS توصیف شده است تا مدل TMS-ARBAC را نشان دهد.

این مقاله به شرح زیر سازمان دهی شده است در بخش 2 زیر ساخت های MTA و مدلهای کنترل دسترسی مرتبط مرور می شود در بخش 3، یک تعریف رسمی از مدل TMS-ARBAC فراهم شد و استراتژی های اشتراک منابع برای STA توصیف شده است. در بخش 4 تحلیل ایمنی TMS-ARBAC ارائه شد. و در بخش 5 TMS-ARBAC به طرح کنترل دسترسی برای اشتراک گذاری برنامه های علم الکترونیکی جغرافیا اعمال شده است و در بخش 6 از مقاله نتیجه گیری شده است.

## 2.کارهای مرتبط

در این بخش MTA و STA در SaaSو مکانیزم های اسمنی مرتبط با آنها مرور می شود.

### 2.1 MTA و STA در SaaS

در حال حاضر 5 روش برای پیاده سازی MTA وجود دارد : یکپارچه سازی با پایگاه داده، روش میانی، SaaS خدمات محور، روش PaaS محور و روشی هدف محور، اکثر این روشها روی شکل سازی مستاجران و اجرای برنامه آنها روی یک پلتقدم ابری متمرکز هستند.

اما کدبز و همکارانش اشاره کردند که چند مستاجری فاقد یک تعریف واضح است. به نظر می رسد که بسیاری از تکنولوژی های MTA موجود از قبیل مرکز داده، مجازی سازی و روشهای اشتراکی میانی یا MTA ، PaaS محور. MTA واقعی نیستند اما روشهای استفاده چند برنامه ای یا چند لحظه ای فاقد کارایی و اشتراک منابع هستند.

در MTA اکثر سیستم های موجود مستاجران را بعنوان نهادهای واحد که کاملاً از سایر مستاجران جدا شده‌اند لحاظ می کنند. اما SaaS اشتراک گذاری را ترویج می کند در جهان واقعی بسیاری از تعاونی ها، سازمانهای سطوح

یکسان یا هم نوع دارای ملزومات برنامه ای یا پروسرهای تجاری مشابه هستند.رابطه بین در مستاجر باید بررسی شود. اما فقط تعداد اندکی از کارهای مرتبط چاپ شده است.

از نقطه نظر موازنه بار چهار نوع وابستگی (بدون وابستگی، وابسته به سرور، وابسته به کلاستر و وابستگی میان کلاستر) برای دسته بندی کاربران مختلف یک مستاجر برای گره های پردازش برای اشتراک منابع و کارایی موثر داده شده است.

من هات و همکارانش (17) یک مدل سلسله برای را برای نمایش منطقی درخت مستاجر و نگاشت به ذخیره فیزیکی معرفی که دراد اما انکار وی کارایی و تماس پذیری ذخیره دسترسی داده احراز هویت تمرکز کرده اند.

کسیتون رابط برنامه ریزی برنامه های هویت (APL) چند مستاجری سلسله مراتبی را برای Open Stack با استفاده از حوزه ها و پروژه برای ساخت سلسله مراتب گروه های کاربری و زیر مجموعه های منابع فراهم کرده است.

با در نظر گرفتن روابط سلسله مراتبی بین مستاجران در زندگی واقعی یک الگوی چند مستاجری سلسله مراتبی معرفی شده است .(6) اما هیچ یک از آنها اشتراک منابع شخصی شده یا جداسازی میان مستاجران را بررسی نمی کنند.

یک پلتوم. MTA SaaS ، Sales force است که یک نمونه برنامه و یک طرح پایگاه داده را برای حمایت از چند مستاجری (شامل مستاجران سطح سرمایه گذاری) با استناد از یک معماری نرم افزار درایو شده از متاداده، APL های مبتنی بر استاندارد و تولید کننده برنامه زمان اجرای برای فعال سازی برنامه های CRM چند مستاجری را اجرا می کند.

از نقطه نظر استفاده مجدد از منابع مشترک و مشخص سازی آسان، STA برای مدلسازی همه انواع چند مستاجری سلسله مراتبی با اشتراک منابع بین مستاجران سطوح مختلف ارائه شده است. این مستاجران اجازه می دهد که خدمات را برای توسعه دهندگان مستاجر دست دوم برای شخصی سازی کاربردهای آنها مجاز می سازد مدلهای مختلف STA با مدلهای مختلف شخص سازی تعریف شده اند.

## 2.2 کنترل دسترسی بر MTA

این یک موضوع مهم در SaaS است وقتی که همه مستاجران در منابع محاسباتی یکسان شریک هستند. MTA نیازمند اشتراک منابع، اشتراک کارایی، جداسازی حریم خصوصی داده و برنامه بطور همزمان است. امنیت سیستم بطور گسترده در لایه های مختلف مانند انتقال شبکه، مدیریت سیستم و ذخیره دیتا مطالعه شده است این مقاله روی احراز هویت و مجازسازی MTA و STA تمرکز کرده است.

برای اعمال RBAC برای شناسایی فاعل ها مفعول و اجازه لازم است در یک سیستم SaaS یک فاعل می تواند (1) یک مستاجر (2) یک مستاجر دست دوم (3) یک کاربر برنامه های مستاجر یا مدل های مستاجر دست دوم باشد یک مفعول یم تواند برنامه یا مولفه داده درون SaaS باشد اجازه بین فاعل و مفعول ها درون سیستم SaaS باشد. روش های بعدی برای اعمال RBAC به SaaS بکار رفته اند .

*طرح پایگاه داده و مدل RBAC

زیر ساخت های سنتی SaaS IT های مبتنی بر PaaS داده محور مانند IBM یا مایکرو سافت را ایجاد کرده اند. چندین مستاجر بوسیله تعریف طرح پایگاه داده و مدل RBAC با اشتراک گذاری یک مرکز داده جداسازی شده است.

پایش و همکارانش یک مدل کنترل دسترسی چند مستاجر را بر اساس جدول های بواالاستیک (EET) ارائه کرده اند کابر والد ـ فرزند و جدول گرده به ترتیب برای توصیف گروه کاربر و گرره مستاجر بکار رفته اند. نقش ها به آنها برای جداسازی یا اشتراک گذاری داده تخصیص داده شده است.

در این مدل ها استراتژی های امنیت روی داده و نه روی اشتراک گذاری خدمات تمرکز می کنند.

لی و همکارانش، RBAC را به سیستم های SaaS اعمال کرده و سه مسئله را تحریف کردند. تضاد اسم نقش، مدیریت سطوح متقابل و ایزومویزم کنترل دسترسی مستاجر شامل روابط نا همگن نقش ها و محدودیت های ناممکن تخصیص اجازه آنها یک مدل S-RBAC را ارائه کردند که در آن کنترل دسترسی به دو بخش تقسیم شده

است: سطح مستاجر و سطح سیستم.با درنظر گرفتن سلسله مراتب نقش و محدودیت های مرتبط آنها S-RBAC را به H-RBAC با نمایندگی نقش و محدودیت ها زمانی بسط داده اند.

یک مدل کنترل دسترسی مبتنی بر مستاجر T-ARBAC با اضافه کردن مستاجر به مدل ARBAC و جدا کردن وظایف مدیران سیستم و مدیران مستاجر ارائه شده است. منابع سیستم به دو بخش زیر منابع تقسیم شده است . به هر مستاجر یک استخر تخصیص داده می شود که به شدت مستاجران مختلف را جدا می کند.

در مرجع(6) براساس RBAC کاربران به واحد ، واحد مجزا، واحد مرکب، . کاربر برای نمایش سلسله مراتب کاربران بسط یافته اند.

با هدف گذاری نواقص ARBAC97 ارائه شده در سازمانهای بزرگ با چندین شرکت تابعه خود مختار N-RBAC از ساختار فضای اسم سلسله مراتبی برای چیدمان کاربران و نقش های مناسب تر برای مدیریت نقش موزع شده خود مختار بکار رفته است.

برای ارضای نیازهای مدیریت غیر متمرکز چند سلسله مراتبی رد برنامه هیا بزرگ، یک مدل مدیریتی سلسله مراتبی نقش محور MHARBAC رو به جلو قرار داده شده است که از درخت نقش برای حمایت رمز احراز هویت بالا به پایین استفاده می کند. رابطه ارثی در سلسله مراتب نقش حذف شه است  محدودیت های جدید اضافه شده است دامنه مدیریتی به دامنه کاربرد دامنه نقش و دامنه اجازه تقسیم شده است.

• مدل شبیه RBAC برای شرکت چند مستاجری

بسط های مدل RBAC برای احراز هویت همکاری در ابرها ارایه شده است. بجز آنهایی که دارای مجوز متمرکز هستند که برای ابر مناسیب نیستند. سه راه برای ساختن احراز هویت به شرح زیر می‌باشد: (1) نمایندگی در RBAC براساس تصمیمات کاربر مجزا، (2) هویت فدرالی شده و خدمات احراز هویت و (3) مدیریت اعتماد در مکانیزم های کنترل دسترسی.

برای همکاری در منابع برون سپاری یک خانواده مدل MT-RBAC برای فراهم کردن احراز هویت ریزدانه ای در محیط برای همکاری با ساخت روابط اعتماد بین مستاجران ارایه شده است. اما این نیاز دارد که هر مستاجر بعنوان

متولی به یک صادر کننده واحد بعنوان امانت گذار تعلق داشته باشد که مسول ایجاد رابطه اعتماد و اضافه کردن کاربران متولی به نقش های امانت گذاران است. اما در SaaS یک منبع مشترک ممکن است از چندین مولفه از تامین کنندگان مختلف تشکیل شده باشد که این فرض را ارضا نمی کند. کنترل دسترسی امانت گذاری روی کاربران متولی خودمختاری متولی را نقض می کند.

در اکوسیستم های SaaS فدراسیون خدمات ایجاد شده است تا یک مشتری بتواند از خدماتی که در فدراسیون روی سازمانها های چندگانه معین کرده است استفاده کند. سه عامل اصلی (یک زنگ روی خدمات، تامین حساب های کاربری و مکانیزم های امن) برای کنترل دسترسی در فدراسیون لازمند. IBM یک چهارچوب مدیریت هویت آزاد را برای اکوسیستم SaaS ارایه کرده است.

SalesForce از یک تعیین کننده سازمان منحصر به فرد برای هر مستاجر استفاده می کند. احراز هویت فرد و رمز نگاری داده بدین منظور بکار رفته اند که تضمین کنند که دسترسی به دنیا توسط مالک خودش دارای امنیت است طرح های اشتراک گذاری سه لایه ای برای نشان دادن مجموعه های مختلف داده به مجموعه های مختلفی از کاربران تعریف شده اند. آنها این سطح شیء سطح میدان و سطح رکورد هستند جدول NameDenorm برای ارائه روابط والد ـ فرزند بکار رفته است اشتراک در سراسر سازمان، سلسله مراتب نقش و سلسله مراتب قلمرو می تواند برای تعریف دسترسی اشتراکی به رکوردهای داده بکار می رود پروفایل‌ها با یک تابع شغل کاربر تعریف می شوند. احراز هویت اختصاصی یک رنگ مجزا و احراز هویت فدرالی برای کشف به وصل شدن کاربران به Sales Force بردن یک شرکت جهت استفاده از خدمات مرکب فراهم شده است.

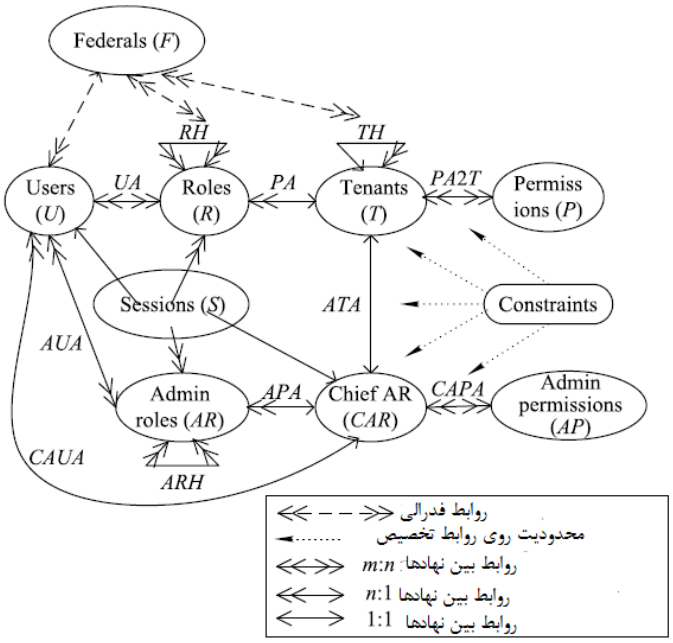
کار روی روشهای ایمنی ترکیب خدمات و مسئله جریان اطلاعات در خدمات مرکب از دیدگاه خدمات محور انجام شده است (19) اما بسیاری از آنها روی تنظیم و اجرای خدمات ترکیب تمرکز کرده اند. هیچ کاری مسائل ایمن در سیستمهای STA SaaS انجام نشده است.

## 3. مدل کنترل دسترسی مبتنی بر مستاجر برای STA , MTA در SaaS های سرویس محور

در پلتفرم های SaaS، MTA یا STA یک مستاجر یک نهاد خود مختار است که می‌تواند تصمیم گیری های بیماری از قبیل حفظ حریم خصوصی خود، اشتراک گذاری منابع خود با مستاجران دست دوم خود، مجاز ساختن آنها به شخصی سازی بیشتر آنچه که فراهم می کند و عمل بعنوان مدیر SaaS در قلمرو خود را انجام دهد در این بخش یک مدل کنترل دسترسی مبتنی بر مستاجر به نام TMS-ARBAC روی ARBAC برای جداسازی مستاجران سطوح مختلف و اشتراک گذاری در SaaS ، STA سرویس محور معرفی شود.

### 3.1 کلیات TMS-ARBAC

همانگونه که در شکل 2 نشان داده شده است مدل TMS-ARBAC سه مولفه نهاد جدید را اضافه می کند مستاجران نقش های مدیران ارشد و فدرال ما روی ARBAC- سلسله مراتب مستاجر اضافه شده است تا روابط سلسله مراتب بین مستاجران و مستاجران دست دوم را نشان دهد. اما هر مولفه در TMS-ARBAC مشابه – ARBAC نیست.




شکل 2. مدلTMS-ARBAC

مستاجر (T): یک مستاجر نهادی است که سیاست های خود و یک سازنده برنامه را توسعه می دهد وی مالک منابع اجاره شده از سیستم بوده و برنامه های خودش را شخصی سازی می کند می تواند مستاجران دست دوم (ST) و کاربران را ایجاد کرده و به آنها مجوز دهد که با منابع که مالکشان هستند در کنترل دارند چه کاری انجام دهند یک اسم معضد به خود مستاجر برای تعیین ناحیه خودمختاری خودش (شامل واحدها، منابع و مجوزها) از سایر مستاجران بکار می رود مدیر سیستم بر داده های خصوصی مستاجر درسترسی نداشتند و نمی تواند تغییری در دامنه مجوز مستاجر فراتی از توافق سطح سرویس (SLA) ایجاد کند. هر ST باید یک مستاجر والد داشته باشد. اما هر مستاجر والد ممکن است چندین ST داشته باشد یک ST نتایج را از مستاجر والد خود گرفته و بعد می داند شخصی سازی خودش را انجام داد. و کاربران خودش را ایجاد کند در STA های چند سطحی یک ST های چند سطحی یک STمی تواند ST های خودش را ایجاد کند.

**نقش ها(R)**: **R های تعریف** مشابه تعریف ارائه شده در را دارد. ARBAC اما برخلاف . RBAC سنتی یک نقش به دامنه مستاجر باریک شده است که برای بسیاری از مستاجران بطور عادی نامرئی است تخصیص نقش از مدیران مستاجر می آیند از مدیران SaaS

**کاربران(U)** مستاجران و مستاجران دست دوم می توانند کاربران خودشان را ایجاد کنند فرق بین مستاجر و کاتر این است که مستاجر ها سازمانها هستند اما کابرها کاربران انتهایی هستند مدیران مستاجر با مستاجر دست دوم کاربران انتهایی خود را نقش ها مجاز می سازند. یک کاربر ممکن است چندین نقش داشته باشد وظایف یک کاربر توسط نقش هایی که به او اختصاص یافته است تشخیص داده می شود که می تواند یک مدیر، مدیر مستاجر با یک کاربر عادی برنامه درمنطقه کاری باشد.

**فدرال ها(F)** یک فدرال برای اشتراک منابع بین مستاجران ایجاد شده است که شامل آنهایی که بین مستاجران والد ـ فرزند است نمی شود این یک سازمان موسسه واقعی نیست اما یک نهاد مجازی است تخصیص مستاجران ، نقش ها و کاربران در یک فدرال در بخش 3.3 و 6.3 بطور مفصل تر بررسی خواهد شد.

**مجوزها (P)** P در اینجا همان تعریف مشابه ARBAC را دارد اما در SaaS سرویس محور SaaS منابع و مجوزهای دسترسی آنها باید بیشتر تقسیم شو که در بخش 3.5 بیشتر بررسی خواهد شد.

**جلسات (S)** S همان تعریف مشابه RBAC را طرح یک جلسه، نقش تخصیص یافته به کاربر می تواند فعال شود. توجه داشته باشید که در MT و SaaS ST به یک کاربر و نقش های فعال یک جلسه می تواند از سیستم، مستاجر/ مستاجر دست دوم یا یکی فدرال باشد برای ملزومات حریم شخصی مستاجر محدودیت های شدید باید برای اجتناب از در دیون اطلاعات بین مستاجر ها بکار رود. نقش های مدیریتی ارشد (CAR) نقش های مدیریتی (AR) و مجوزهای مدیریتی (AP) به کار مدیریتی مستاجران نقش ها، کاربران و تخصیص مجوز و مدیریت ناحیه خود مختاری مربوط می شود که در بخش 4.3 بیشتر بررسی خواهد شد.

در جدول 1 همه اختصارات و توصیف آنها که بر مدل و استراتژی های بکار رفته است ارائه شده اند همه روابط در شکل 2 روی پیکان ها برچسب زده شده اند بطوری که در بخش 3.3 معرفی خواهند شد.

جدول 1 مفاهیم بکار رفته در مدلها و استراتژی ها

| نماد | مفهوم |
|---|---|
| AA | ناحیه خودمختاری |
| AP | اجازه مدیر |
| APA | اجازه برای اختصاص AR |
| ARH | سلسله مراتب نقش های مدیر |
| AR | نقش مدیر |
| ATA | تخصیص مستاجر به CAR |
| AUA | تخصیص به کاربر AR |
| CAR | نقش مدیر ارشد |
| CAP | مجوز مدیر ارشد |
| CAPA | تخصیص به کاربر CAR |
| CSO | افسر ارشد ایمنی |
| CAUA | تخصیص به کاربر CAR |
| IP | اجازه به ارث رسیده |

| | |
|---|---|
| فدرال | F |
| تخصیص نقش بیرونی به کاربر | FUA |
| روابط فدرال | FR |
| نقش بیرونی | O-R |
| مجوز | P |
| RRتخصیص مجوز به | PA |
| AAتخصیص مجوز به | PA2T |
| مجوز خصوصی | PP |
| نقش ها | R |
| سلسله مراتب نقش | RH |
| نقش عادی | RR |
| جلسه | S |
| مستاجر دست دوم | ST |
| مستاجر | T |
| سلسه مراتب مستاجر | TH |
| روابط درخت | TR |
| کاربر | U |
| به کاربرRRتخصیص | UA |

### 3.2 مدیران سیستم

مستاجران و مستاجران دست دوم آنها، نقشها و کاربران همه در پلتونیم SaaS زندگی می کنند هر نهاد دارای حوزه کاری خودش است بر خلاف هویت آن در RBAC سنتی مدیر سیستم دیگر قوی ترین خود نیست وی مستاجران را ایجاد می کند به مستاجران مجوزهای بناب را می دهد تا به منابع بلتوزم SaaS طبق SLA دسترسی داشته باشند همچنین مجوزهای مستاجران را لغو کرده و آنها را با SLA پاک می کند. به محض اینکه یک مستاجر حذف شد همه مستاجران دست دوم آنها، نقش ها و حریم شخصی اش نیز حذف می‌شود.

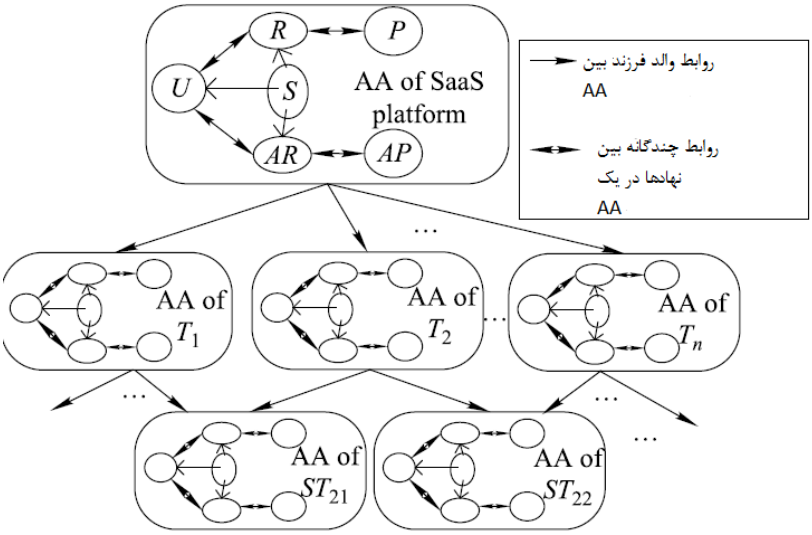
در MTA و STA مستاجران نهادهای خودمختار هستند مستاجران دست دوم نقش ها و کاربران باید در یک دامنه مستاجر تعریف شوند. برای جداسازی مستاجر نقش ها نمی تواند به کاربران سایر مستاجر ها اعطا شود به محض

اینکه یک مستاجر ایجاد شد، اعطای مجوز و احراز حویت در دامنه مستاجر نمی تواند توسط مدیر سیستم یا سایر مستاجران مختل شود.

برای ارضای شرایط فوق یک ناحیه خود مختاری (AA) برای توصیف ناحیه کنترل ایمنی یک مستاجر ارایه می شود یک AA یک دامنه نام گذاری شده برای محدود کردن محدوده مدیریت ایمنی RBAC مستاجر است هر AA یک اسم منحصر به فرد دارد. هر نهاد ( مانند نقش کاربرد یا مجوز ) یک مستاجر در AA مستاجر منحصر به فرد است که یک اسم منحصر به فرد و یک اسم AA منحصر به فرد به عنوان پیشوند دارد. دستری به سیاست های کنترل برای یک مستاجر فقط می تواند به نهاد های AA خودش تخصیص داده شود.

به وضوح هر AA دارای یک فضای اسم مستقل است لذا نهاد ها در AA های مختلف می تواند اسم مشابه داشته باشد که تضمین می کند مستاجر شی های درونی خود را آزادانه بدون لحاظ کردن تضاد اسمی با سایر مستاجران در محیط MTA و STA نام گذاری کند.

چون هر مستاجر ممکن است مستاجران دست دوم خودش را داشته باشد یک AA ممکن است AA زیر مجموعه خود را نیز داشته باشد همه AA ها در محیط MTA و STA یک درخت AA را تشکیل می دهند که سلسله مراتب مستاجران را مشابه شکل 3 نشان می دهد اما از نظر سلسله مراتب مستاجران و ترتیب ایجاد AA ها به ارث رسیدن مجوز از گره فرزند به گره والدش در درخت AA ممنوع است. اشیا در یک AA نمی تواند از هر AA دیگر قابل مشاهده باشد حتی اگر در AA دارای رابطه والد فرزندی مستقیم بینشان باشد. مگر اینکه یک مجوز واضح اشتراک گذاری وجود داشته باشد که در بخش 3.6 بررسی خواهد شد.




شکل 3. ساختار درخت AA مدل TMS-ARBAC

### 3.3 مدل TMS-ARBAC

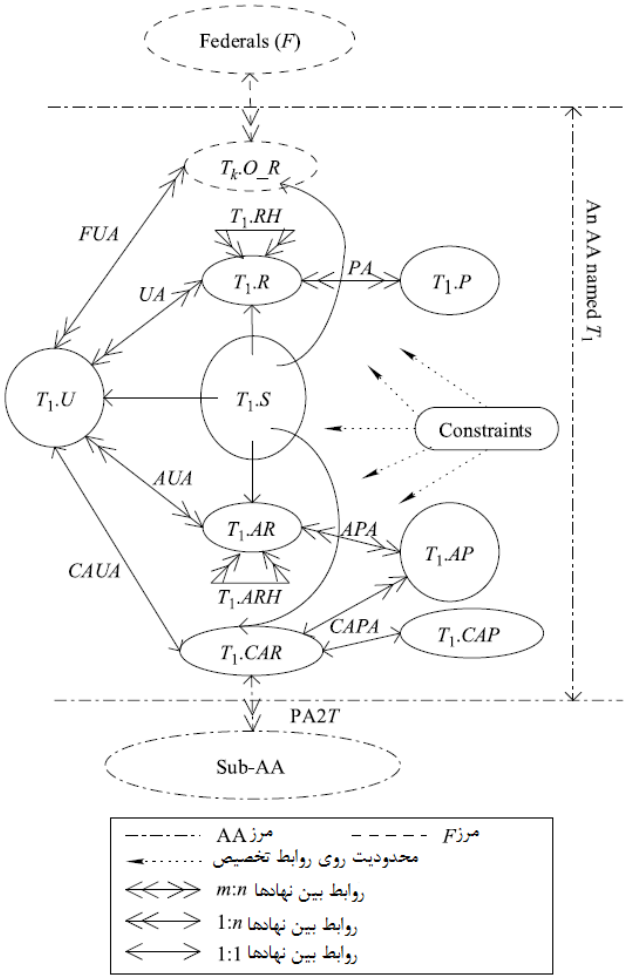
مولفه های نهاد و روابط آنها در یک AA در شکل 4 نشان داده شده است و مدل TMS-ARBAC از یک مجموعه از AA ها تشکیل شده است.

تعریف رسمی مدل TMS-ARBAC به شرح زیر است:

تعریف 1: یک AA دارای مولفه های زیر است:

* uیک مجموعه از کاربرهای ایجاد شده در AA جاری
* R، AR،CAR : مجموعه های گسسته از نقش های عادی، مدیریتی و مدیریت ارشد ایجاد شده در AA جاری.
*O-R : یک مجموعه از نقش های بیرونی به اشتراک گذاشته شده با سایر AA ها با AA جاری.
* P،AP،CAP: مجموعه گسسته ای از مجوز های عادی ، مدیریتی و مدیریت ارشد در AA جاری .
*S : یک مجموعه از جلسات
* $PA \subseteq P \times R$ مجوز های چند به چند برای رابطه تخصیص نقش عادی
* $PA2T \subseteq P \times A$ مجوز چند به چند برای رابطه تخصص مجوز AA

* $APA \subseteq AP \times AR$ مجوز چند به چند برای رابطه تخصیص نقش مدیریتی
* $CAPA \subseteq (AP \cup CAP) \times CAR$: مجوز چند به یک برای رابطه تخصیص نقش مدیریت ارشد
* $UA \subseteq U \times R$: کاربر چند به چند برای رابطه تخصیص نقش های عادی
* $FUA \subseteq U \times O\_R$: کاربر چند به چند برای تخصیص O-R روابط ارثی یا انتقالی در بین نقش های بیرونی مجاز نیست.
* $AUA \subseteq U \times AR$ کاربر چند به چند برای رابطه تخصیص نقش مدیریتی
* $CAUA \subseteq U \times CAR$ کاربر چند به یک برای رابطه تخصیص نقش مدیریت ارشد. در یک AA مستاجر فقط یک افسر ارشد امنیت (CSO) وجود دارد.
* $RH \subseteq R \times R$: سلسله مراتب نقش نسبتا منظم
* $ARH \subseteq AR \times AR$ سلسله مراتب نقش مدیریتی نسبتا منظم . (هر دو سلسله مراتب به صورت ≥ در مفهوم نشاندن نوشته می شود)
* کاربر $S \rightarrow U$ هر جلسه Si را به کاربر واحد user (si) نگاشت می کند ( که برای طول عمر جلسه ثابت است)




شکل 4. مولفه ها و روابط در یک ساختار درخت AA

* نقش ها : $S \rightarrow 2^{R \cup O\_R \cup AR \cup CAR}$ هر جلسه Si را به یک مجموعه از نقش ها نگاشت می کند

$\text{roles}(s_i) \subseteq \{r | (\exists r' \geqslant r)[(user(s_i), r') \in UA \cup AUA \cup FUA \cup CAUA]\}$ ( که می تواند با زمان تغییر کند) و جلسه Si دارای مجوز های

$$\bigcup_{r \in roles(s_i)} \{p | (r \in (R \cup AR) \wedge (\exists r'' \leq r)[(p, r'') \in PA \cup APA]) \vee$$
$$(r \in CAR \cup (p, r) \subset CAPA) \vee (r \subset O\_R \wedge (p, r) \subset area(r).PA)\}$$

است که area(r)، AA را نشان می دهد که در آن نقش R ایجاد شده است.

*یک مجموعه از محدودیت ها وجود دارد که تصریح می کند کدام مقادیر مولفه های مختلف فوق الذکر مجاز یا غیر مجاز هستند.

* تعریف R، U ، AR و S در یک AA مشابه مدل ARBAC97 است لذا در این مقاله به جزییات آن ها نمی پردازیم.

تعریف 2 مدل TMS-ARBAC به صورت بک چندتایی <AA, TR,FR> بیان می شود.

- $AA = \{a_1, a_2, \ldots, a_n\}$ که $a_i$ یک AA و n شمارشگر AA است.
- اگر $TR^{n \leqslant 1}$ یک مجموعه تهی است. در غیراینصورت TR یک مجموعه از روابط بیان شده به صورت <ai,aj,P> است که اگر $a_j$ یک AA فرعی از $a_j$ باشد، P مجموعه مجوزی است که $a_j$ از $a_i$، AA والد خود به ارث برده است. در غیراینصورت $a_j$ یک AA والد از $a_j$و P مجموعه مجوزی است که $a_i$ به $a_j$ والد خود برای اشتراک منابع اعطا می کند. هر AA باید یک و فقط یک AA والد داشته باشد بجز AA ریشه.
- FR مجموعه ای از روابط فدرالی است. یک فدرال بصورت یک 5 تایی تعریف می شود. $f = \langle F_{id}, C_a, F_m, F_o, F_c \rangle$، که روابط اشتراک منابع بین مستاجران غیر والد –فرزندان را نشان می دهد.

Fid، ID منحصر به فرد فدرال f است.

Ca رییس f انتخاب شده توسط همه اعضای فدرال است که یک AA است.

$F_m = \{a_1, a_2, \ldots, a_k\}, (k \geqslant 1)$ یک زیرمجموعه از AA است که AA های گردهم آمده در f را نشان می دهد.

$F_o = \{\langle a_i, a_j, a_i.r \rangle\} \mid a_j \in F_m \wedge a_k \in F_m \wedge i \neq k \wedge a_i.r \in a_i.R\}$ به این معناست که AAai یک نقش ai.r را با AAak دیگر در f شریک است.

Fc گروهی از محدودیت های روی f (مانند محدودیت زمانی روی f، محدودیت ویژه بین فدرال ها و غیره) و محدودیت های تخصیص از f به کاربرانش است.

براساس تعریف 2 ممکن است دو نوع رابطه بین AA وجود داشته باشد:

رابطه والد فرزندی،و فدرالی. روابط والد-فرزندی همواره وجود دارد لذاهر ناحیه خودمختار در AA ها از یک درخت تشکیل شده است.

رابطه فدرالی به چند AA اجازه می دهد یک فدرال را متحد کند. و نقش ها را بین یکدیگر بدون فدرال به اشتراک گذارد. O_R یک AAai مجموعه ای از نقش های داده شده به AA های دیگر در فدرال است که $a_i$ به آن تعلق دارد. Fid به O_R اضافه شده است تا نقش های بیرونی تخصیص یافته از همان AA فدرال های مختلف را ممتمایز سازد.

$$a_i.O\_R = \bigcup_{f \in FR}\{(a_j.r, f.F_{id}) \mid (\exists \langle a_i, a_j, a_j.r\rangle \in f.F_o \mid i \neq j)\}.$$

### 3.4 نقش مدیر ارشد و مجوز مدیر ارشد

در مدل ARBAC97 نقش های مدیریتی فقط اجازه کنترل نقش های عادی را دارند و مدیریت همه نقش های مدیریتی و مجوز ها تحت کنترل یک CSO است که بالاترین اختیار را در سیستم دارد اما این مجوز دیگر در محیط STA برقرار نیست.

وقتی مکانیزم RBAC به یک AA اعمال میشود. ایجاد و تخصیص نقش های عادی و مدیریتی توسط CSO مستآجر انجام میشود نه CSO پلتفرم SaaS علاوه بر این چون مستآجران ممکن است مستاجران دست دوم داشته باشند و مستآجران دست دوم نیز خود مختار هستند مستآجر والد باید به یک مستاجر دست دوم اجازه دهد که CSO خودش را برای کار مدیریتی در AA مستاجر دست دوم تعریف کند. در محیط STA چند سطحی این پروسه به طور وارون ادامه می یابد.

بنابراین در محیط STA هر مستاجر یک CSO دارد. یک CSO مرکزی کلی وجود ندارد اما CSO های زیادی در کل سیستم توزیع شده اند. کار مدیریتی شامل ایجاد کاربران RR ها AR ها, تخصیص مجوز و AAS زیرین یا فدرالهای جدید است که به صورت دینامیک در چرخه زندگی سیستم SaaS ادامه می یابد

در مدل TMS_RBACما هنوز از مکانیزم RBAC برای تخصیص اختیار CSO ها استفاده میکنیم ,در هر AA به جز AR اصلی تعریف شده در ARBAC97 یک نقش مدیریتی جدید CAR با بالاترین اختیار هر یک AA

تعریف می شود . همه ی کار مدیریتی ایمنی CSO ما میتواند از طریق CAR انجام شود. هر AA یک CAR دارد که مسول ایجاد نگهداری سیاست های RBAC درAA است

اختلاف بین AR,CAR این است اولی می تواند برای مدیریت نقش های عادی و دومی بکار رود. این تفاوت از یک مجموعه خاص از مجوزها به نام CAP مشابه شکل 6 حاصل شده است.

وقتی یک AA ایجاد میشود CAR این AA ایجاد شده وکل CAP را بطور خودکار تولید میکند. به همین ترتیب یک کاربر پیشفرض CSO ایجاد شده و CAR بطور اتوماتیک به آن تخصیص میابد. بعبارت دیگر ایجاد و تخصیص CSO,CAR مشمول عمل خودکار ایجاد یک AA است. CSO میتواند مسول این AA شود.

هیچ کاربری نمیتواند کاربر پیشفرض CSO یا نقش CAR را پاک یا اصلاح کند چون آنها توسط سیستم ایجاد شده اند اما حذف یک AAai به حذف CAR,CSO و همه ی نهاد ها و منابع مربوط به ai منجر میشود با در نظر گرفتن اینکه مستاجران والد باید در کنترل باشند.با توجه به این که یک مستاجر دست دوم بتواند مستاجران نو آموز استخدام کند حین ایجاد یک AA اعضای CAR والد مجاز است تعیین کند که آیا CAR فرزند اجازه ایجاد یا حذف یک AA زیرین را دارد یا نه؟

محدودیت 1 فقط و فقط در یک AA دارد.

2) مجوز مدیر ارشد (CAP)

درTMS_ARBAC مجوز های مدیر ارشد به اعضای CARc اجازه میدهند که برق کارهای مدیریت را انجام دهند. این کار ها به شرح زیر هستند :

- AddUser(c,u,a) که یک کاربر در AAa ایجاد میکند اثر عملیات به صورت $a.U \leftarrow a.U \cup \{u\}$ است .
- DeletUser(c,u,a) که کاربر U را از AA حذف میکند که $u \in a.U$ اثر این عملیات به صورت $a.U \leftarrow a.U - \{u\}$ است.
- AddAdminRole(c,ar,Rs,a) که یک نقش مدیریتی ar را با مجموعه نقش های ارشد فوری Rs در AAa ایجاد می کند اثر این دستور $a.AR \leftarrow a.AR \cup \{ar\}; a.ARH \leftarrow a.ARH \cup (\{ar\} \times R_s)$ است.

- DeleteAdminRole(c,ar,a) که نقش مدیریتی ar را از AAa حذف می‌کند که $ar \in a.AR$. همه چندتایی های مربوطه در روابط APA، AUA و ARH نیز حذف شده اند. اثر دستور به صورت زیر است.

$$a.AR \leftarrow a.AR - \{ar\}$$

- CreatwAutonomousArea(c,as,a,Pi,b)که یک AAas زیرین را با والد فوری a ایجاد می کند. مجموحه مجوزها که as از a به ارث می برد Pi است که $P_i \subseteq a.P$. و b یک عدد بولی است که نشان می دهد آیا as می تواند AA زیرین خود را تولید کند. اثر اعمال این دستور به صورت زیر است.

$$AA \leftarrow AA \cup \{a_s\};$$
$$TR \leftarrow TR \cup \{\langle a, a_s, P_i \rangle\};$$
$$a_s.P \leftarrow P_i;$$
$$a_s.U \leftarrow \{\text{CSO}\};$$
$$a_s.CAR \leftarrow \{CAR\};$$
$$a_s.CAUA \leftarrow \{(\text{CSO}, CAR)\};$$
$$a_s.CAPA \leftarrow a_s.CAP \times \{\text{CSO}\};$$

اگر b درست است، asCAP شامل همه دستورات در این بخش است بجز CreatAutonomousArea، DeleteAutonomous Area و ModifyAutonomousArea.

CAP توسط سیستم طی دستور CreatAutonomousArea به CAR تخصیص داده می شود. آنها فقط می-توانند به CAR تخصیص یابند و نمی تواند به سایر AR ها ارایه شود. علاوه براین چون CAP به CAR تخصیص یاقپفته است لغو آن توسط هرکاربر دیگر مجاز نیست.

- DeleteAutonomousArea(c,as) که یک AAas زیرین را حذف می کند. اثر این دستور به شرح زیر است.

$$AA \leftarrow AA - \downarrow a_s;$$

$TR \leftarrow TR - \{\langle a_i, a_{ij}, P \rangle \mid a_i \in \downarrow a_s\}; \downarrow a_s$ مجموعه ای از AAها است که به زیر درختی تعلق دارند که ریشه اش as است.

- ModifyAutonomousArea(c,a,as,Pi,Pi') که a مجموعه مجوزهای اشتراک گذاری Aaas از P'i به Pi اصلاح می کند. اثر این دستور به صورت زیر است.

$$TR \leftarrow (TR - \{\langle a, a_s, P'_i\rangle\}) \cup \{\langle a, a_s, P_i\rangle\};$$
$$a_s.P \leftarrow (a_s.P - P'_i) \cup P_i;$$
$$a_s.PA \leftarrow a_s.PA - \{(p, r) \mid p \in (P'_i - P_i)\}.$$

- CreateFederal(c,ai,cc,f,b) که یک فدرال f جدید را ایجاد می کند. B یک شرط بولی است که نشان می دهد آیا Ai عضوی از f است. اثر این دستور به صورت زیر است:

$$f.C_a \leftarrow a_i;$$
if b=TRUE, $f.F_m \leftarrow \{a_i\}$; else $f.F_m \leftarrow \varnothing$;
$$f.F_o \leftarrow \varnothing;$$
$$f.F_c \leftarrow \{cc\};$$
$$FR \leftarrow FR \cup \{f\}.$$

- Joinfederal(c,ai,f) که عضو جدید ai را به فدرال f اضافه میکند. اثر این دستور به صورت زیر است:

$$f.F_m \leftarrow f.F_m \cup \{a_i\}.$$

- ShareOuterRole(c,ai,aj,Ro,cc,f) که یک مجموعه از نقش های Ro در AAai را با AAaj دیگر با مجموعه ای از محدودیت های cc که $R_o \subseteq a_i.R$ به اشتراک می گذارد. و هر دوی ai و Aj اعضای فدرال f هستند. اثر این دستور به شرح زیر است:

$$f.F_o \leftarrow f.F_o \cup \{\langle a_i, a_j, a_i.r\rangle \mid a_i.r \in R_o\};$$
$$a_j.O\_R \leftarrow a_j.O\_R \cup \{(a_i.r, f) \mid a_i.r \in R_o\}$$
$$f.F_c \leftarrow f.F_c \cup \{cc\}.$$

نکته: اینکه کدام نقش می تواند O_R باشد توسط ai.CSO تعیین می‌شود. و تخصیص نقش های بیرونی به کاربران ai (FUA) توسط aj.CSO پیاده سازی می‌شود.

- RevokeOuterRole(c,ai,aj,Ro,cc,f) که یک مجموعه از نقشهای Ro را از AAaj دیگر در فدرال f لغو می کند که $R_o \subseteq a_i.R.$ اثر این دستور به صورت زیر است:

$$a_j.O\_R \leftarrow a_j.O\_R - \{(a_i.r, f) \mid a_i.r \in R_o\};$$
$$f.F_c \leftarrow f.F_c - \{cc \mid cc.r \subseteq a_i.r \mid a_i.r \in R_o\};$$
$$f.F_o \leftarrow f.F_o - \{\langle a_i, a_j, a_i.r\rangle \mid a_i.r \in R_o\}.$$

نکته: وقتی یک نقش بیرونی ai.r از aj حذف می شود FUA مربوطه در aj توسط aj.CSO لغو می شود.

- QuitFederal(c,ai,f) یک عضو aj را از فدرال f حذف می کند. اثر این دستور بصورت زیر است:

$$\text{RevokeOuterRole}(c, a_i, a_j, R_o, f), \text{where } a_j \in f.F_m \wedge i \neq j \wedge R_o \subseteq a_i.R\};$$
$$f.F_o \leftarrow f.F_o - \{\langle a_x, a_y, a_x.r\rangle \mid a_x = a_i \vee a_y = a_i\};$$
$$f.F_m \leftarrow f.F_m - \{a_i\}.$$

- Dropfederal(c,ai,f) که فدرال f ایجاد شده با ai را حذف می کند. فقط خالق فدرال f می تواند این دستور را اجرا کند. اثر این دستور به صورت زیر است:

$$\text{QuitFederal}(c, a_i, f), \text{where } a_i = f.C_a;$$
$$FR \leftarrow FR - \{f\}.$$

محدودیت 2 برای اجتناب از به ارث رسیدن ضمنی مزایا در سلسه مراتب نقش ترتیب جزیی روی O_R ها در هر دوی aj و ai مربوط به $\langle a_i, a_j, a_i.r\rangle \in F_o (i \neq j)$ مجاز نیست.

محدودیت 3 یک مجموعه از نقش های بیرونی (ai.r) می تواند به یک AAaj بعنوان یک O-R فقط توسط مالک آنها ai تخصیص یابد. هیچ تضمینی وجود ندارد که AA بتواند یک O_R را به دیگران تخصیص دهد.

محدودیت 4 اگر یک AAa به دو فدرال fi و fj به پیوندد $a \in f_i.F_m \wedge a \in f_j.F_m$ هر نقش بیرونی a.o_ri از f در fj نامرئی و نامعتبر است.

محدودیت 5 برای اجتناب از نشتی حریم خصوصی بین فدرال های مختلف، محدودیت های ویژه باید به تخصیص کاربر فدرال اضافه شود یعنی اینکه هیچ کاربری نمی تواند بطور همزمان در یک جلسه متعلق به فدرال های مختلف باشد.

### 3.5 مجوزها (P)

در یک AA روی یک پلتفرم SaaS مجوزها در ARBAC97 یکسان نیستند. تفاوتهای آنها به شرح زیر است:

1- دامنه مجوزهای تخصیص یافته به نقش به AA خودش محدود شده است.

مجوزها به منابع مرتبط شده اند. در محیط MTA و STA هر مستاجر فقط مالک و اجاره کننده بخشی از منابع در پلتفرم SaaS است لذا محدوده مجوزها برای نقش ها باید محدود شده باشد. در مدل ما مجوزهای مد نظر اعطا از AR به RR به یک زیر مجموعه از مجموعه مجوزهای AA محدود است.

2- مجوزها از مجوزهای به ارث رسیده یا خصوصی تشکیل شده است.

چون شخصی سازی و داده های سری نیز منابع هستند و به سیستم SaaS تلق ندارند مجموعه مجوزهای هر AA از دو بخش تشکیل شده است: مجوزهای به ارث رسیده از هر والد یا به اشتراک گذاشته شده از AAهای فرزند ( که مجوزهای به ارث رسده IP نامیده می شوند.) و مجوزهای منابع خصوصی تحت تملک AA جاری ( که مجوزهای خصوصی PP نامده می شود).

- برای یک AA $a_i$: $a_i.P = a_i.IP \cup a_i.PP$;
- $a_i.IP \subseteq parent(a_i).P \cup sub(a_i).P$ چون aj.IP از AA والد ai به ارث رسیده یا با AAهای زیرین به اشتراک گذاشته شده است.
- اگر یک زیر مجموعه از aiPP به AA زیرین یا AAaj والد اعطا شود یک مجوز نسبی aj خواهد بود. برای سری نگه داشتن آن نباید به سایر مستاجرهای aj اعطا شود.

محدودیت 6 یک مجموعه از مجوزهای خصوصی (ai.PP) می تواند به یک AAaj فقط با ai مالکشان تخصیص یابد. aj حق انتقال ai.PP به دیگران را ندارد.

یک مستاجر ممکن است تعیین کند تا یک مجوز را از مستاجر دست دوک خود لغو کند و بعد همه تخصیص های مربوط به مجوز به نقش باید لغو شود. علاوه براین اگر مستاجر دست دوم مجوز خود را به مستاجر دست دوم خود اعطا کند پروسه پیچیده تری برای تضمین یک لغو اضافی لازم است.

1- منابع مختلف دارای مجوزهای متفاوتی هستند.

مطابق EasySaaS منابع سیستم شامل GUI ها، گردش کارها، مولفه های خدمات و داده ها ( که با G، W، S و D نشان داده شده اند.). از نقطه نظر کنترل دسترسی منابع ترجیح می دهیم این منابع به دو بخش تقسیم شود: AppComponent(G, W,S) و داده، برای آها تفاوتهای روشنی در ویژگی های عملیات و داده دارد.

برای برنامه های STA و طرح دسترسی داده، سه سوال باید در نظر گرفته شود:

- مستاجران دست دوم مستاجرها کدام مجوزها را می توانند داشته باشند.
- مستاجران کدام مجوزها را می توانند به مستاجران دست دوم دهند؟
- مستاجران دست دوم کدام مجوزها را می توانند به مستاجران بدهند؟

مجوزهایی که یک AA می تواند داشته باشد و به سایرین اعطا کند در جدول 2 ارایه شده است.

توجه داشته باشید که کاربران عادی یک AA خالقان برنامه نیستند و نمی توانند شخصی سازی برنامه ها را انجام دهند. مجوز برنامه ها به کاربران فقط استفاده در دوره اجرای برنامه است در حالیکه مجوز مدیران AA اشتراک گذاری و شخصی سازی در دوره ساخت برنامه است. " ارتقایافته" یک مجوز خاص از یک اجاره کننده مولفه Ti به سیستم SaaS است که توصیف می کند که آیا Ti اجازه ارتقای خودکار مولفه را صادر میکند.

جدول 2 مجورهایی که میتواند به AAتخصیص داده شود

| Permission property | AppComponents' permission set | DataSpace/Data's permisstion set |
|---|---|---|
| IP for public | Own/grant/revoke: $\{Us/S, TUs/TS, Upd\}$ | Own/grant/revoke: $\{R, I, U, D, TR, TI, TU, TD\}$ |
| $T_i.PP \rightarrow T_{ij}.IP$ for ST | Own: $\{Us/S, Upd\}$ with no re-grant | Own: $\{R, I, U, D\}$ with no re-grant |
| PP for privacy owner | Own: $\{Us, U, D, Upd\}$; Grant/Revoke: $\varnothing/\{Us/S, Upd\}$ | Own: $\{R, I, U, D, TR, TI, TU, TD\}$; Grant/revoke: $\varnothing/\{R, I, U, D\}$ |

وقتی مجوزهای خصوصی یک مستاجر به یکی از مستاجرانش داده شده است، حریم خصوصی این مجوزها و اعطای این IP ها مجاز نیست. از نقطه نظر منابع داده، داده باید بصورت "داده" و " فضای داده " تفصل شود. اشتراک داده می‌تواند بعنوان اشتراک گذاری یک فضای داده با داده اشتراکی درونی دیده شود. اما مستاجران دست دوم که یک فضای داده را از یک مستاجر اجاره می کند می تواند منجر به یک مشکل کنترل دسترسی جدید شود.

وقتی یک مستاجر جدید Ti به سیستم SaaS وصل می شود، معمولا سیستم یک AAai را با فضای داده DSi به Ti می دهد. وقتی یک مستاجر دست دوم Tij از Ti ایجاد می شود معمولا یک فضای داده DSij با ai.aj به AAaij داده می شود. Dsij از فضای داده والد خودش پارتیشن بندی شده است، که شرط $a_{ij}.DS_{ij} \subseteq a_i.DS_i$ را ارضا می کند. به محض اینکه DSij به aij داده می شود ai نمی تواند بدون مجوز aij به آن دسترسی داشته باشد. اما از نقطه نظر مدیر AA، DSij هنوز به ai تعلق دارد. DSij به یک ناحیه سیاه برای ai تبدیل می شود چون ai مالک DSij است اما نمی تواند آنرا کنترل کند. با اضافه کردن رمزنگاری داده به DSij با آخرین مالک ai آن می تواند به حل مشکل ناحیه سیاه کمک کند. وقتی aij حذف شده است، ai مسولیت DSij را دوباره بدست می آورد. به همین ترتیب رمزنگاری aij روی DSij نیز حذف می شود. از دیدگاه سیستم اجاره فضای داده یک مورد فضای چند بخشی داده است.

از نقطه نظر AppComponant به محض اینکه یک مولفه ایجاد شده و در پلتفرم SaaS منتشر شد فقط تامین کننده آن می تواند پلتفرم را ارتقا یا به روزرسانی کند. اینکه مستاجرانی که آنرا به اشتراک گذاشته اند شخصی سازی را انجام بدهند یا ندهند تغییری در مولفه اصلی وجود نخواهد داشت. اما بدون توجه به اینکه یک مستاجر بتواند یا نتواند شخصی سازی را انجام دهد نیازمند مجوز است. برای قابلیت استفاده مجدد برای مولفه ها، مولفه های برنامه ها باید روی پلتفرم SaaS بصورت گردش کار و مولفه های خدمات تقسیم شود. به محض اینکه یکی از مولفه های زیرین مولفه مرکب ارتقا یافت، این مولفه مرکب کلی باید برای هر اجاره کننده اش ارتقا یابد. از نقطه نظر سیستم اجاره مولفه های برنامه یک مورد ترکیب چندگانه مولفه است.

برای حل مشکلات کنترل دسترسی در دومورد فوق دو جدول متادیتا باید اضافه شود تا سابقه اجاره فضای داده تجزیه شده یا مولفه های برنامه مرکب ثبت و ضبط شود. آنها عبارتند از:

• RCompositeInfoTable(r_id,r_parent,r_owner) روباط تجزیه و ترکیب مولفه های منابع و داده ها را ثبت می کند.

• RRenterTable(r_id, grantterAA, permissions, timeLimit)، مجوزهای منابعی را که AA اعطا کننده با محدودیت زمانی به AA اعطا شونده می دهد ثب می کند.

### 3.6 استراتژی های اشتراک گذاری بین مستاجران

همانگونه که در بلا اشاره شده در مدل TMS-ARBAC دو مستاجر (شامل مستاجر و مستاجران دست دومش) کاملا از یکدیگر بوسیله AA هایشان جدا شده اند. اما در STA SaaS روابط اشتراک گذاری بین دو مستاجر می تواند به صورت زیر دسته بندی شود: روابط مستاجر=مستاجر دست دوم (T-ST)، روابط مستاجر-مستاجر (T-T) و روابط مستاجر دست دوم-مستاجر دست دوم (ST-ST). استراتژی اشتراک گذاری برای هر موقعیت در ادامه داده شده است.

1) استراتژی اشتراک گذاری T-ST

استراتژی های اشتراک گذاری T-ST اشتراک گذاری سلسله مراتبی بین مستاجر و مستاجر دست دومش است. 5 زیر مدل از دو مدل سطح STA (TSTA) برای توصیف کنترل اشتراک گذاری بین مستاجر و مستاجر دست دوم تعذیف شده اند که عبارتند از: مدل سرور-مشتری (SC-TSTA)، مدل نرم افزار-داده (SD-TSTA)، مدل مستر-اسلیو (MS-TSTA) و مدل شریک-STA شرکا (PP-TSTA). با استفاده از این زیر مدلها ویژگی های اشتارک گذاری منابع بین یک مستاجر و مستاجر دست دومش در جدول سه ارایه شده است.

همانگونه که در جدول 3 نشان داده شده است برای اشتراک گذاری مولفه های برنامه ای، مجوزهای شخصی سازی و انتشار ارتقا در هر مدل STA لازمند. هر دوی آنها باید بصورت ویژگیهای اشتراک گذاری AppComponant

پیش فرض در MTA SaaS و STA تنظیم شده باشند. مد اشتراک گذاری AppComponant به پیاده سازی مولفه خدمات برای هر اجاره کننده مربوط است. از نقطه نظر کنترل دسترسی بیشتر به این مربوط است که چگونه اطلاعات احراز هویت بین زیر مولفه های مختلف یا لحظات مختلف عبور می کند.

جدول 3. ویژگی های اشتراک گذاری بین مستاجران والد-فرزند در مدل های TSTA

| ویژگی های اشتراک گذاری | SC-TSTA | SD-TSTA | MS-TSTA | SM-TST | PP-TST |
|---|---|---|---|---|---|
| اشتراک گذاری AppComp | $T \rightarrow ST$ | $T \rightarrow ST$ | $T \rightarrow ST$ | $T \rightarrow ST$ | $T \leftrightarrow ST$ |
| مد اشتراک گذاری | یک لحظه | یک لحظه | چند لحظه | multiple instances | multiple instances |
| شخصی سازی | $ST$ | $ST$ | $ST$ | $ST$ | $T$ and $ST$ |
| اشتراک گذاری داده | نه | $T \rightarrow ST$ | $T \rightarrow ST$ | $T \rightarrow ST$ | $T \leftrightarrow ST$ |
| ارتقای انتشار یافته | بله | بله | بله | Yes | Yes |

در مدل TMS-ARBAC، TR<ai,aj,P> در درخت AA برای توصیف روابط اشتراک گذاری سلسله مراتبی منابع والد-فرزند بکار رفته است. مجموعه مجوزهای P به ما می گوید که چه چیزهایی بین مستاجران والد- فرزند به اشتراک گذاشته شود. اشتراک گذاری از والد به AA فرزند اشتراک گذاری ارثی است که با دو عملیات پیاده سازی می شود:CreateAutonomousArea و ModifyAutonomousArea. اشتراک گذاری از فرزند به AA والد اشتراک گذاری حریم خصوصی فرزند است که با ModifyAutonomousArea پیاده سازی می شود.

عملیات تخصیص مجوز خودکار بین AAai و AAaij زیرین به صورت زیر توصیف می شود:

- $a_{ij}.P \leftarrow a_{ij}.P \cup \{P_i\}$, که $P_i \subseteq \{(r\_id_{ij}, ds), R, I, U, D, TR, TI, TU, TD\}$ که r_idij، id منبع است و aij مجاز نیست AAزیرین خودش را داشته باشد پس $P_i \subseteq \{(r\_id_{ij}, ds), R, I, U, D\}$.

در RCompositeInfoTable با چندگانه $(r\_id_{ij}, r\_id_i, r\_id_i.Owner)$ وارد کنید که r_idi، id منبع فضای داده ds والد است.

در RRenterTable با چندگانه $(r\_id_{ij}, a_i, a_{ij}, P_i, timeLimit)$ وارد کنید، aij به آخرین اجاره کننده جاری r_idij تبدیل می شود.

تذکر: اگر چه مجوز ai روی r_idi همچنان وجود دارد اما ds با رمزنگاری aij قفل شده است.

- $\text{DataSharePAssignFromTtoST}(a_i, a_{ij}, t\_name, key\_range[m,n], P_i)$ که به aij اجازه می دهد به t_name جدول ai با key_range[m,n] با مجموعه مجوز Pi دسترسی داشته باشد.

$$ds \leftarrow dataS\,pace(t\_name, key\_range[m,n]);$$

که $a_{ij}.P \leftarrow a_{ij}.P \cup \{P_i\}$ اگر $P_i \subseteq \{ds, R, I, U, D, TR, TI, TU, TD\}$

Data $(t\_name, range[m,n])$ داده خصوصی ai باشد پس $P_i \subseteq \{ds, R, I, U, D\}$.

- $\text{DataSharePAssignFromSTtoT}(a_i, a_{ij}, t\_name, key\_range\,[m,n], P_i)$ که به ai اجازه می دهد به t_name جدول aij درون key_range[m.n] با مجوز Pi دسترسی داشته باشد.

$$ds \leftarrow dataS\,pace(t\_name, key\_range[m,n])$$

$a_i.P \leftarrow a_i.P \cup \{P_i\}$ که $\{ds, R, I, U, D\}$ $P_i \subseteq$.

Data$(t_name, range[m,n])$ داده خصوصی aij است.

- $\text{AppCompSharePAssignFromTtoST}(a_i, a_{ij}, acp, P_i)$ که به aij اجازه می دهد مولفه های برنامه ai، cap را با مجموعه مجوز Pi اجاره کند.

$a_{ij}.P \leftarrow a_{ij}.P \cup \{P_i\}$, که $P_i \subseteq \{acp, Us, S, TUs, TS\}$. اگر acp مولفه برنامه خصوصی ai باشد، پس

$P_i \subseteq \{acp, Us, S\}$.

چندتایی $\text{tuple}(acp\_id, a_i, a_{ij}, P_i, timeLimit)$ را در RRenterTable وارد کرده و id، acp_id منبع acp است. هر اجاره کننده مولفه هر برنامه اینجا ثبت شده است که به ارتقا و ابطال مولفه کمک میکند.

- $\text{AppCompSharePAssignFromSTtoT}(a_i, a_{ij}, acp, P_i)$ که به aij اجازه می دهد مولفه برنامه خود acp را با ai با مجموعه مجوز Pi به اشتراک گذارد.

$a_i.P \leftarrow a_i.P \cup \{P_i\}$ که $P_i \subseteq \{acp, Us, S\}$. Acp مولفه برنامه خصوصی aij است.

چندتایی $\text{tuple}(acp\_id, a_i, a_{ij}, P_i, timeLimit)$ را در RRentTable وارد کرده و acp_id، id منبع acp است.

2) استراتژی اشتراک گذاری T-T

در MTA و STA مستاجران از والدین مختلف می توانند یک فدرال را برای اشتراک گذاری منابع با یکدیگر متحد کنند. که به این معناست که کاربران مستاجر Ti می تواند جداسازی بین مستاجران را برای دسترسی به منابع مستاجر دیگر بسط دهد. فرض میکنیم که یک فدرال نیاز ندارد که یک مولفه ویژه را برای همه اعضایش ایجاد کند. یا فدرال یک مستاجر جدید خواهد بود. لذا روابط اشتراک گذاری فدرال را در حد اشتراک گذاری نقش باریک می کنیم. در TMS_ARBAC اشتراک گذاری T-T با فدرال ها و O_R با عملیات CreateFederal، JoinFederal، و ShareOuterRole پشتیبانی می شود. اینکه فدرال یک مورد اشتراک گذاری داده یا اشتراک گذاری AppComponant است توسط تعریف O_R مشخص می شود.

رییس فدرال که یک AA انتخاب شده با اعضای فدرال است مسول مدیریت اعضای فدرال و O_Rها بدون ایجاد اخلال در هر AA است. AA اعطا کننده تصمیم می گیرد که کداک نقش می تواند O_R باشد. و کدام کاربر به O_R تخصیص می یابد توسط AA اعطا کننده با محدودیت های Fc مشخص می شود. اینکه کدام O_R می تواند به کدام AA یا همه اعضای فدرال اختصاص یابد توسط محدودیت های Fc فدرال مشخص می شود.

3) استراتژی ST-ST

ST-ST به روابط درون مستاجر مربوط می شود. در زندگی واقعی شاخه هایی از یک کارخانه یا سازمان وجود دارد که نیازهای کاربردی یکسانی دارند اما روی نواحی مختلف کار میکنند.

اگر منابع اشتراک گذاری شده بین دو مستاجر دست دوم مولفه های برنامه شخصی سازی خودشان یا داده جدا سازی خودشان باشد یک حالت خاص اشتراک گذاری T-T است. استراتژی اشتراک گذاری ST-ST یک اشتراک گذاری فدرال است. تنها فرق بین فدرال ST-ST و فدرال T-T این است که در فدرال ST-ST، AA والد آنها رییس فدرال است، اما در فدرال T-T رییس یک عضوی از فدرال نیز است.

در غیراینصورت دو مستاجر دست دوم باید منابع را بطور مستقیم از والدین خود به ارث بده باشند. این می تواند به سادگی با استراتژی اشتراک گذاری T-ST با اشتراک گذاری ارثی همان فضای داده یا همان مولفه های برنامه والدینشان حل شود.

فرق بین اشتراک گذاری فدرال (FR در درخت AA) و اشتراک گذاری والد-فرزند ( مانند TR در درخت AA) به این است که دومی دانه دانه بودن مجوز را می گیرد، لذا مدیران AA های دریافت کننده می توانند مجوزها را در نقش های مختلف بصورت انعطاف پذیر ترکیب کنند، تا ملزومات کنترل دسترسی سازمانها ارضا شود تا کاربران محلی فقط به منابع بیرونی از طریق نقش های تعریف شده با محدودیت های شدید برای دستیابی به امنیت قابل اتکاتر دسترسی داشته باشند.

## 4. ارزیابی TMS-ARBAC

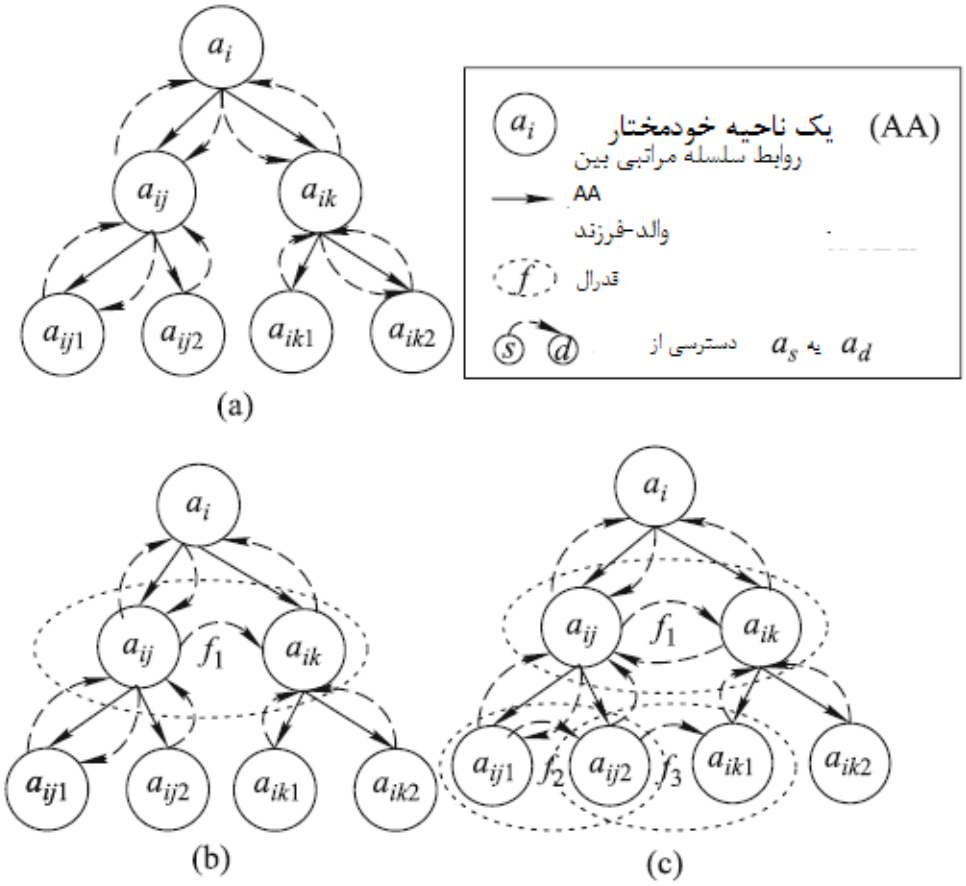
### 4.1 تحلیل ایمنی TMS-ARBAC

با در نظر گرفتن یک AA به عنوان یک سازمان خودمختار توزیع شده در ابر امنیت اشتراک گذاری در بین AA ها یک مسئله ارتباط بین عملکردی امن در یک محیط STA SaaS است.

از نقطه نظر کنترل دسترسی ویژگی های ایمنی با یک لیست کنترل دسترسی (ACL) بیان شده اند. جنبه ایمنی ارتباط بین عملکردی با حقوق دسترسی روی AA ها ارایه شده است. دو نوع ارتباط بین عملکردی بین AA ها وجود دارد، که TR و FR در TMS-ARBAC هستند. یک مسئله ارتباط بین عملکردی امن AA می تواند به صورت زیر تعریف شود: با داشتند لیست های کنترل دسترسی برای هر AA در MTA و STA سیستم SaaS، یک FR یا TR ارتباط بین عملکردی که یک مجموعه از ورودی های کنترل دسترسی است که برای هر ورودی موضوع و هدف به AA های مختلف تعلق دارند. مشکل کلی تصمیم گیری در این مورد است که که آیا دسترسی وجود دارد که در یک AA غیر مجاز باشد و در اثر ارتباط بین عملکردی مجاز شود، که وارد نقض ایمنی را بسط یا حذف می کند درحالیکه یک سطح معقول از ارتباط بین عملکردی را حفظ میکند.

طبق مدل TMS-ARBAC هر AA یک $ACI = \{\cup IP, \cup PP, \cup O\_R\}$ دارد که یک مجموعه از IP های ارایه شده با AA والد یا AA زیرین، IP آن و O_R داده شده با فدرال های وصل شده به آنرا نشان می دهد. $(a_i, a_j, p/o\_r)$ را بصورت دسترسی از AAai به منابع aj از طریق برخی مدهای اشتراک گذاری تعریف می کنیم. دو نوع از مشکلات ارتباط بین عملکردی امنیتی باید بررسی شود: ارتباط بین عملکردی اشتراک گذاری سلسله مراتبی و ارتباط بین عملکردی اشتراک گذاری فدرال.

با FR و TR ارتباط بین عملکردی دسته ای در یک درخت AA در شکل 5 نشان داده شده است.




شکل 5 . ارتباط بین عملکردی در درخت AA، (a) دسترسی سلسله مراتبی در یک درخت AA، (b) دسترسی اشتراک گذاری سلسله مراتبی و فدرالی. (c) دو فدرال با اشتراک گذاری دسترسی با یک AA.

• تحلیل ایمنی ارتباط بین عملکردی اشتراک گذاری سلسله مراتبی در بین AA ها

فرض کنید یک AAai یک $ACL_i = \{\cup IP_i, \cup PP_i, \cup O\_R_i\}$ دارد در حالیکه $\cup O\_R_i = \emptyset$.

**مورد 1** اگر $\cup IP_i \neq \emptyset, PP_i = \emptyset$ همه منابعی که در دسترس ai است از AA والد خود به ارث رسیده است یا توسط AA زیرین خود به اشتراک گذاشته شده است. دسته $\cup IP_i$ با انواع منابعش به شرح زیر است.

مورد 1-1 مجوزهای دسترسی انحصاری فضای داده از AA والد به فرزند ( که مورد اختصاص فضای داده است). مالک این فضای داده ( که با dsi نشان داده شده است) از AA والد به Aij تغییر می کند. لذا آن به یک منبع انحصاری ai تبدیل می شود. اگر ai یک مستاجر دست دوم Aij داشته باشد و مقداری از فضای داده آزاد ai را به aij اختصاص دهد، $a_i.DS = ds_i - ds_{ij}$ خواهد بود که هنوز یک منبع انحصاری ai است.

مورد 1-2 مجوزهای اشتراک داده بین AA های والد فرزند. این دسترسی ai به داده AA والد یا زیر AA یک دسترسی مجاز است. اگر چنین داده ای داده سری AA اعطا کننده باشد، AA گیرنده نمی تواند آنها را دوباره به AAهای دیگر اعطا کند.

مورد 1-3 مجوزهای اشتراک AppComponant بین AA های والد-فرزند. اگر AppComponant یک مولفه عمومی بدون اطلاعات شخصی یا سری باشد، هر AA می تواند از آن با مجوزها استفاده کند. درغیراینصورت فقط یک نمونه مجزا از آن به ai داده می شود. فقط مالک مولفه می تواند مولفه های اصلی را تغییر دهد.هیچ AA نمی تواند حریم شخصی مولفه های دیگر را بگیرد. اگر چنین مولفه ای یک مولفه شخصی شده از AA اعطا کننده باشد، AA گیرنده نمی تواند آنرا به دیگران اعطا کند.

لذا دسترسی ai به به منابع اشتراکی بدون توجه به اینکه برای AA والد یا زیرینش خصوصی باشد یا نباشد، انحصاری یا دارای مجوز است. برای AA والد ai منابعی که به مستاجران دست دوم خود می دهد با مجوزهای آن به اشتراک گذاشته شده اند یا بطور انحصاری به ai اختصاص داده شد است. هیچگونه نقض ایمنی در ai و AA والدش وجود ندارد.

مورد 2 اگر $\cup IP_i \neq \emptyset, \cup PP_i \neq \emptyset$، ai دارای مولفه های برنامه داده خصوصی با شخصی شده خودش است. اینجا مدیر AA داخلی را کنار بزار. ai منابع خصوصی را تک قطبی کرده یا آنها را با AA والد یا زیرین خود با مجوزها به اشتراک می گذارد. ( همانگونه که در مورد 1-2 یا 1-3 ارایه شده است.) محدودیت 6 برای جلوگیری از نشت حریم شخصی به بیرون ai و مستاجران نسبی دارای مجوز آن بکار می رود.

مورد 3 با درنظر گرفتن یک درخت AA با ارتفاع 3 مشابه شکل 5a ارتباط بین عملکردی بین AA والد و فرزند همگی با استفاده از اعطای مجوزها های یک AA به بقیه پیاده سازی شده است. برخلاف نقش ها مجوزها هیچ گونه رابطه ترتیب جزیی یا سلسله مراتبی ندارند. هیچگونه مسیر دسترسی اضافی نمی تواند برای دسرسی متقابل AA های والد- فرزند ایجاد شود.

به همین ترتیب منابع به اشتراک گذاشته شده (غیر از منابع خصوصی یا سری) می تواند بصورت شرطی در سراسر درخت AA منتقل شود. بعنوان مثال مطابق شکل 5a aij1 می تواند aij را مجوز دسترسی داده p به اشتراک گذارد که با $(a_{ij}, a_{ij1}, p)$ نشان داده می شود. فرض شد که چنین داده ای سری نیست اما با ارزش است $(a_i, a_{ij}, p')(p' \subseteq p)$ و aik می تواند بخشی از این داده را از $a_i(a_{ik}, a_i, p'')(p'' \subseteq p')$ به ارث ببرد. این نشان می دهد که منابع به اشتراک گذاشته شده در AA درخت می تواند انعطاف پذیر باشد که شرایط برنامه STA SaaS را ارضا می کند.

نوعی نقض ایمنی در این ارتباط بین عملکردی اشتراک گذاری سلسله مراتبی مشاهده شده است. با استفاده از نمونه فوق اگر آنچه که به اشتراک گذاشته می شود همان داده ذخیره شده در همان فضای داده باشد، ai و aik می توانند به داده aij1 بدون اجازه آن دسترسی داشته باشند. اما این داده های اشتراکی سری نیستند یا مجوز انتقال با محدودیت 6 ممنوع خواهد بود. بهمین ترتیب یک AppComponant می تواند بین سطوح به اشتراک گذاشته شود بدون داشتن مجوز مستقیم. اما اینمولفه فقط می تواند عمومی باشد بدون نشتی اطلاعات خصوصی. حتی می توان از کنترل دسترسی اجباری یا مدیریت حوزه اعتماد برای اجتناب از این نوع نقض ایمنی استفاده کرد.

- تحلیل ایمنی ارتباط بین عملکردی اشتراک گذاری فدرال بین AA ها

فرض کید AAai یک $ACL_i = \{\cup IP_i, \cup PP_i, \cup O\_R_i\}$ دارد درحالیکه $\cup O\_R_i \neq \emptyset$ که بدان معناست که ai می تواند به چندین فدرال بپیوندد تا منابع سایر AA ها را به اشتراک گذارد.

مورد 4 فرض کنید فقط یک فدرال در درخت AA وجود داشته باشد مانند شکل 5b، یک فدرال f دو AA به نام های aij و aik با دسترسی های $(a_{ij}, a_{ik}, O\_R)$ دارد. اگر $(a_{ij}, a_i, p_1)$ و $(a_i, a_{ik}, p_2)$ وجود داشته باشد(همان مورد مشابه نقض ایمنی فوق)، فرض کنید که O_R و p1 هر دو به یک کاربر u در aij اختصاص یافته اند. پس u دو راه برای دسترسی به aik دارد. آیا O_R و p1 هر دو می توانند به یک منبع aik اختصاص یابند؟

نه. چون در این حالت aij و aik خواهر و برادر هستند، آنچه که بین آنها و فدرال به اشتراک گذاشته می شود باید سری باشد همانگونه که در بخش 3.6 ، استراتژِی اشتراک گذاری ST-ST تعریف شده است. اما سایر لیست های کنترل دسترسی یک روش انتقال اختیار است. منابع اشتراکی نباید سری یا خصوصی باشند. بنابراین هیچگونه نقض ایمنی ارتباط بین عملکردی در این مورد وجود نخواهد داشت.

مورد 5 اگر یک دسترسی جدید $(a_{ik}, a_{ij}, O\_R')$ را در f1 اضافه کنی، آیا یک نقض ایمنی در f1 وجود خواهد داشت؟

نه. طبق محدودیت 2، ترتیب جزیی روی O_R ها در aij و aik قطع خواهد بود. هیچ حلقه دسترسی نمی تواند بین aij و aik ایجاد شود. و کاربری که O_R به آن اختصاص یافته است فقط توسط پذیرنده AA با محدودیت های تخصیص O_R در f1 تصمیم گیری می شود، تا از اختصاص O_R به هر کاربر غیر قابل اعتماد اجتناب شود.

**مورد 6** فرض کنید دو فدرال با یک AA مشابه در در یک درخت AA مطابق شکل 5c وجود داشته باشد فدرال f2 با اعضای فدرال aij1 و aij2، f3 با aij2 و aik1. اگر دسترسی های $(a_{ij1}, a_{ij2}, O\_R_1)$ و $(a_{ij2}, a_{ik1}, O\_R_2)$ بطور مجزا در f2 و f3 وجود داشته باشند، آیا ممکن است یک دسترسی از aij1 به aik1 ایجاد شود؟

بله. اگرچه با محدودیت های 3 و 4، aij2 نمی تواند آنها را به فضای داده خود اختصاص داده و بنویسد، اما اگر aij2 بتواند داده aik1 را بخواند و آنها را در فضای داده خود بنویسد پس می تواند آنها را با O_R به aij1 انتقال دهد. از محدودیت 5 برای اجتناب از انتقال مستقیم از aik1 به aij2 استفاده می شود. اما این دسترسی غیرقانونی ممکن است بصورت ضمنی وجود داشته باشد. می توان از جدایی وظایف، مانند سیاست امنیتی دیوار چین و مدیریت اعتماد برای مقابله با این مشکلات نقض استفاده کرد.

### 4.2 مشخشات TMS-ARBAC

مشخصات نوعی TMS-ARBAC برای وفق دادن تقاضای جداسازی مستاجران و اشتراک گذاری منابع در STA SaaS است.

1) جداسازی و خودمختاری هر مستاجر

• هر مستاجر دارای ناحیه کاری AA مربوط به خودش است. منابع، نقش ها و کاربران همه دراین ناحیه کاری محدود هستند. لذا یک AA یک دیوار جدا سازی را از سایر مستاجران می سازد.

• مدیران سطح سیستم و سطح مستاجر، به روشنی جدا شده اند. مدیران سیستم بیشتر مراقب تخصیص و مدیریت منابع سیستم هستند ما مدیران مستاجر بیشتر مراقب نحوه استفاده از منابع خودشان هستند. سیستم SaaS نیز یک AA است. هر AA دارای CAR و CSO خودش است. توابع شغلی مختلف می تواند به روشنی در AA های مختلف تعریف و اجرا شوند.

• هر مستاجر سیاست های دسترسی خودش را با ملزومات برنامه در AA خودش ایجاد می کند. مدیران سیستمی نمی توانند در مدیریت داخلی مستاجران دخالت کنند. مستاجر نیز نمی تواند در امور داخی مستاجرا دست دوم خود دخالت کند. فقط مجوز های تخصیص و لغو منابع از سلسله مراتب AA تبعیت می کند.

2) اشتراک منابع برای مستاجران

• اشتراک گذاری فدرالی منابع به اشتراک گذاشته شده بین مستاجران غیرمرتبط یا مستاجران برادر خواهر در یک فدرال به اشتراک گذاشته می شوند. مالکیت این منابع تغییر نمی کند. ایمنی توسط اعتماد فدرال و محدودیت های O_R تضمین می شود.

• اشتراک گذاری ارثی منابع سیستم از مشتاجران والد به فرزند به ارث می رسد.مالکیت این منابع ممکن است از مستاجر والد به مستاجر دست دوم تغییر کند، مانند مورد تخصیص دوبار فضای داده. به محض اینکه مالکیت تغییر کرد، مالک اولی اختیار خود روی منابع را تا لغو مستاجران دست دوم از دست می دهد. اگر تغییر نکند پذیرنده حق آپدیت کردن منابع اصلی را ندارد.

• اشتراک موارد خصوصی بین مستاجران منابع خصوصی شامل داده شخصی سازی و سری بطور خصوصی بین مستاجران والد-فرزند یا مستاجران خواهر-برادر به اشتراک گذاشته می شود که یک نوع استفاده مجدد منابع خصوصی است. مالکیت خصوصی قابل تغییر نیست. مزایای دسترسی به حریم خصوصی فقط می تواند توسط مالک آن منتشر می شود. پخش پذیرنده ها به حریم خصوصی سایرین ممنوعه است که از پخش شدن سری نگه داشته می شود.

مقایسه وظایف مدل بین TMS_ARBAC و چهار مدل کنترل دسترسی نوعی MTA (H-RBAC، T-ARBAC کنترل دسترسی روی EET و Saleforce) در جدول 4 نشان داده شده است.

جدول 4. مقایسه مدلهای کنترل دسترسی MTA و STA

| Model | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| جدایی وظایف بین مدیران و مستاجران سیستم | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| خود مدیریت مستاجر | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ایجاد مستاجر توسط مستاجر والد خود | ✓ | × | × | ✓ | ✓ | ✓ |
| خود مدیریت مستاجر دست دوم | ✓ | × | × | × | ✓ | / |
| نقش های محدود شده در دامنه مستاجر | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| جدایی منابع مستاجر | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| اشتراک منابع فدرال بین مستاجران | ✓ | / | / | ✓ | / | ✓ |
| منابع به ارث رسیده بین مستاجران والد-فرزند | ✓ | × | × | ✓ | / | ✓ |
| مستاجران دست دوم شخصی سازی مجدد را انجام می دهند | ✓ | / | / | × | / | ✓ |
| احترام حریم شخصی بین مستاجران | ✓ | × | × | × | × | / |

از مقایسه فوق روشهای ایمنی TMS_ARBAC و Saleforce بهترین روش برای ایمنی ملزومات کنترل دسترسی STA هستند. اما Saleforce فقط روی فیلد CRM تمرکز می کند، و MTA و STA را به روش ویژه خودش روی Force.com رسیدگی می کند. TMS-ARBAC هر مستاجر دست دوم را بعنوان یک فرد خودمختار می بینید و به روابط اشتراک گذاری مختلف با ساختار درخت کوپل شده شل رسیدگی میکند که یک مدل کلی انعطاف پذیر و غیرمتمرکز است که می تواند به آسانی به محیط ابر اعمال شود.

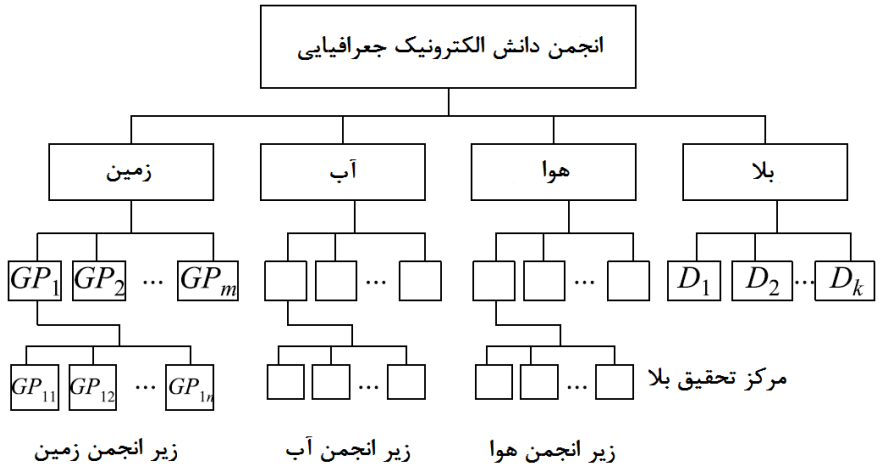
این مدل می تواند روی محیط های سیستم های دیگر علاوه بر ابر نیز بکار رود فقط اگر محیط چندین کاربر، منابع متعدد، ( مانند خدمات، مولفه برنامه ها و داده) داشته باشد و شرط کنترل دسترسی داشته باشد تا به کاربر یا گروهی از کاربران بطور انحصاری مالک برخی منابع بوده، برنامه های خود را شخصی سازی کنند یا بطور انعطاف پذیر تعدادی از منابع واحد یا بسته منابع را با دیگران شریک شوند.

## 5. یک مطالعه موردی

این مقاله مدل TMS_ARBAC را برای کنترل دسترسی یک داده جعرافیایی  دانش الکترونیکی و پلتفرم ابر اشتراک گذاری ابزارهای مبتنی بر EasySaaS را بکار می برد.

1) انجمن جعرافیایی دانش الکترونیکی

انجمن داده دانش الکترونیکی را از نواحی جعرافیایی مختلف از قبیل داده زمین شناسی، آب و هوا، داده هیدرولوژیک و داده بلایای جعرافیایی را نگهداری می کند. سازماندهی چنین انجمن دانش الکترونیکی در شکل 6 نشان داده شده است.




شکل 6. سازمان انجمن دانش الکترونیک جعرافیایی

هر زیر انجمن یک سازمان مستقل یا نهاد است که ممکن است شاخه های سلسله مراتبی مختلفی داشته باشد، که به صورت یک ستاد ملی، و چندین دپارتمان ایالتی و شهری تقسیم شده است. برخلاف زیر انجمن های فوق، مرکز تحقیقات بلایای طبیعی توسط وظایف شغلی مانند دپارتمان جمع آوری داده، دپارتمان تحلیل داده و دپارتمان پیشبینی بلاتقسیم شده است.

هر زیر انجمن دارای سرویسهای داده متناظر خود است که شامل سرویس های داده عادی (مانند آپلود، دانلود و پاک کردن داده) و خدمات ویژه ( مانند الگوریتم های برش داده و الگوریتم های مختلف استخراج داده) می باشد.

2) جداسازی منابع و ملزومات اشتراک گذاری

• ملزومات جداسازی

I-1 فیلدهای مختلف تحقیق یا زیر انجمن ها نهادها یا سازمانهای مستقل هستند: مرکز تحقیقات بلایای طبیعی نیز یک سازمان مستقل است.

I-2 در یک زیر انجمن مانند انجمن زمین هر شاخه سطح ایالتی یک ماهیت خودمختار است که مالک چندین زیر شاخه سطح شهری با توابع شغلی توزیع شده در ناحیه مختلف است.

I-3 هر زیر انجمن، شاخه یا زیر شاخه دارای فضای داده خودش است.

• ملزومات اشتراک گذاری

S-1 اشتراک گذاری سراسری مولفه های مشترک. مانند آپلود، دانلود داده که در کل انجمن به اشتراک گذاشته شده اند.

S-2 اشتراک گذاری رو به پایین مولفع عای برنامه. در یک زیر انجمن، ستاد ملی ممکن است مولفه های برنامه خود را با شاخه های استانی اش به اشتراک گذارد شاخه استانی نیز ممکن است مولفه های خود را با شاخه های شهری خود به اشتراک گذارد. شخصی سازی خدمات در سطح ملی و سطح استانی مجاز است.

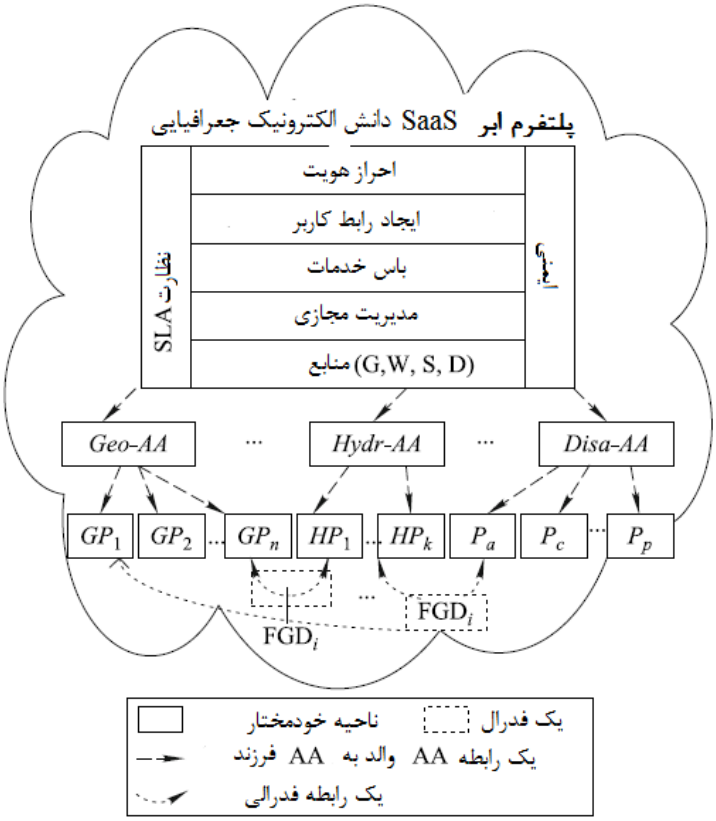
S-3 اشتراک گذاری خواهر برادری. دو شاخه خواهر –برادر (سطح استان یا سطح شهر) ممکن است روند تجاری مشابهی داشته باشند و مولفه های سرویس مشابهی را با شخصی سازی خودشان به اشتراک گذاشته باشند.

S-4 اشتراک گذاری داده رو به بالا. هر شاخه داده خودش را به دپارتمان بالاتر تسلیم می کند بعنوان مثال یک دپارتمان سطح شهری داده خودش را به دپارتمان سطح استانی که به آن تعلق دارد تسلیم می کند و دپارتمان سطح استانی نیز داده خودش را به دپارتمان سطح ملی تسلیم می کند.

S-5 اشتراک گذاری داده فدرال. مرکز بلایای طبیعی می تواند داده های علمی را از میدان های مختلف تحقیقاتی برای تحلیل و پیش بینی بلایای طبیعی جمع آوری می کند. و نتایج تحقیقات آن می تواند در میدان های مختلف منتشر شود.

از دید ملزومات اشتراک گذاری S-2و S-4 سیستم SaaS دانش الکترونیکی جعرافیایی یک مورد STA مستر اسلیو است. 3) استراتژی های کنترل دسترسی برای ابر SaaS دانش الکترونیکی جعرافیایی

طبق ملزومات برنامه های فوق، شکل 7 زیرساخت ابر SaaS دانش الکترونیک جعرافیایی را نشان می دهد. استراتژی های کنترل دسترسی به صورت زیر داده شده اند.




شکل 7. زیرساخت ابر SaaS دانش الکترونیک جعرافیایی

• استراتژی جداسازی: با استفاده از AA برای جداسازی سازمان/نهادها؛ در یک AA با استفاده از تخصیص نقش ها برای محدود کردن منطقه کاری کاربر

برای I-1 هر زیر انجمن بعنوان یک مستاجر در پلتفرم SaaS ساخته شده است. AA متناظر به هر مستاجر داده شده است: *Geo-AA, Hydr-AA, Weath-AA,* . مرکز بلایای طبیعی نیز یک مستاجر است *Disa-AA*. نقش ها به مدیران ستاد ملی یک AA اختصاص داده می شود تا نواحی کاری آنها را تقسیم کند.

برای I-2 هر شاخه سطح استان در یک زیر انجمن بعنوان یک مستاجر دست دوم ساخته می شود. برای مثال دپارتمانهای مختلف سطح استان مستاجران دست دوم Geo-AA هستند. AA انها با GP1، GP2 غیره نشان داده می شود.

برای I-2 و I-3 هر زیر شاخه از یک شاخه سطح استانی نیز می تواند بصورت یک مستاجر دست دوم تعریف شود. اما چون این شاخه های شهری ممکن است منطق کاربری یکسانی داشته باشند فقط فضای داده آنها باید جداسازی شود. این یک مورد MTA سنتی است. ما از تخصیص نقشها و کاربران برای اشتراک مولفه سرویس و جداسازی داده برای هر زیر-زیر شاخه استفاده می شود.

Disa-AA دارای شاخه های مختلف با توابع شغلی مختلف است، اما هر شاخه ممکن است از همان داده استفاده کند. چون تعداد کمی از اشتراک گذاری مولفه های برنامه ای بین شاخه ها وجود دارد، این شاخه ها بصورت مستاجر دست دوم تعریف نشده اند، اما از اختصاص نقش ها و کاربران برای کنترل دسترسی آنها به مولفه های سرویس استفاده می کنند.

• استراتژیهای اشتراک گذاری

برای S-1 یک علامت منبع عمومی برای اشتراک گذاری سراسری ارایه شده است. برای هر مستاجر برای دسترسی آزاد به مولفه های رایج یک علامت خاص "PUBLIC" به ویژگی دسترسی منابع داده می شود.

برای S-2 استراتژی اشتراک گذاری مولفه برنامه T به TS اعمال شده است.مولفه های برنامه ای مناسب از یک AA به AA زیرینش به اشتراک گذاشته شده است. AppComp- SharePAssignFromTtoST $(a_i, a_{ij}, acp, P_i)$.

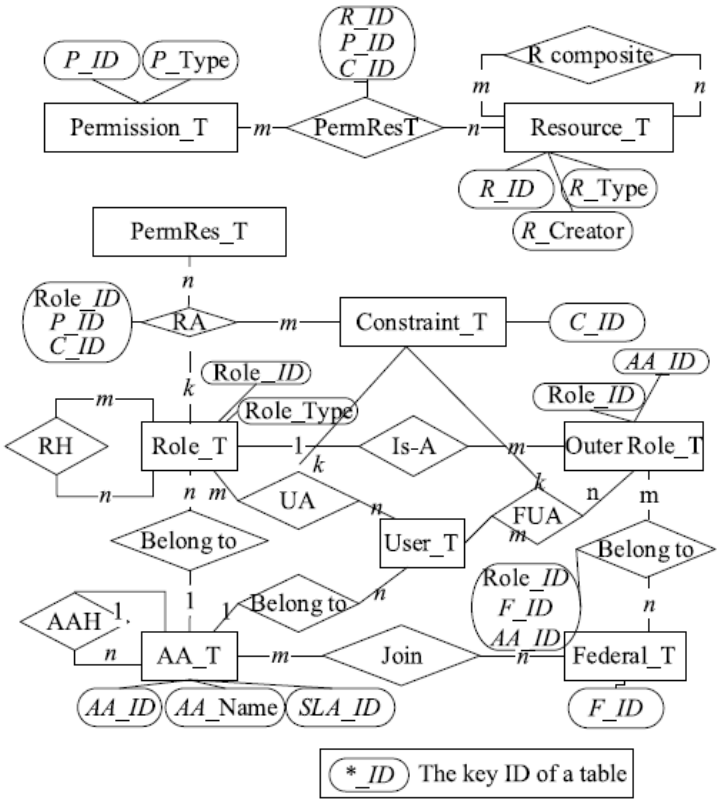
برای S-3 استراتژی اشتراک گذاری فدرال ST-T برای اشتراک گذاری بین مستاجران خواهر برادر بکار رفته است. مولفه های برنامه ای شخصی سازی شده بین مستاجران دست دوم خواهر برادر به اشترک گذاشته شده اند. ShareOuterRole $(a_i.CSO, a_{ij}, a_{ik}, a_{ij}.r, cc, f)$ که $a_{ij}.r$ یک مجموعه از نقش ها برای دسترسی به مولفه های شخصی سازی شده است. اشتراک گذاری مولفه های شاخه شهری می تواند با تخصیص نقش ها محقق شود.

برای S-4 استراتژِی اشتراک داده ST-T بکار رفته است. مولفه های داده از AA زیرین به AA والد خود به اشتراک گذاشته شده اند. $\mathrm{DataSharePAssignFromSTtoT}(a_i, a_{ij}, t\_name, key\_range[m,n], P_i)$.

برای S-5 استراتژی اشتراک گذاری T-T برای اشتراک گذاری بین مستاجران غیر مرتبط استفاده شده است. فدرالها بین Disa-AA و سایر AA های زیر انجمن ایجاد شده اند. O_R ها از AA زیر انجمن به Disa-AA برای اشتراک گذاری، تحلیل و پیش بینی بلایای طبیعی بکار رفته اند.

$\mathrm{ShareOuterRole}(c, a_i, a_j, a_i.r, cc, f)$ که ai.r یک مجموعه از O_R ها برای Disa-AA جهت دسترسی به داد آنها است. و Disa-AA یک O_R را برای AA زیر انجمن دیگر برای دسترسی به نتایج آنها ایجاد می کند.

1) طرح زیر سیستم کنترل دسترسی در پلتفرم ابری SaaS دانش الکترونیک جعرافیایی.




شکل 8. طرح دیاگرام E-R برای کنترل دسترسی در STA SaaS.

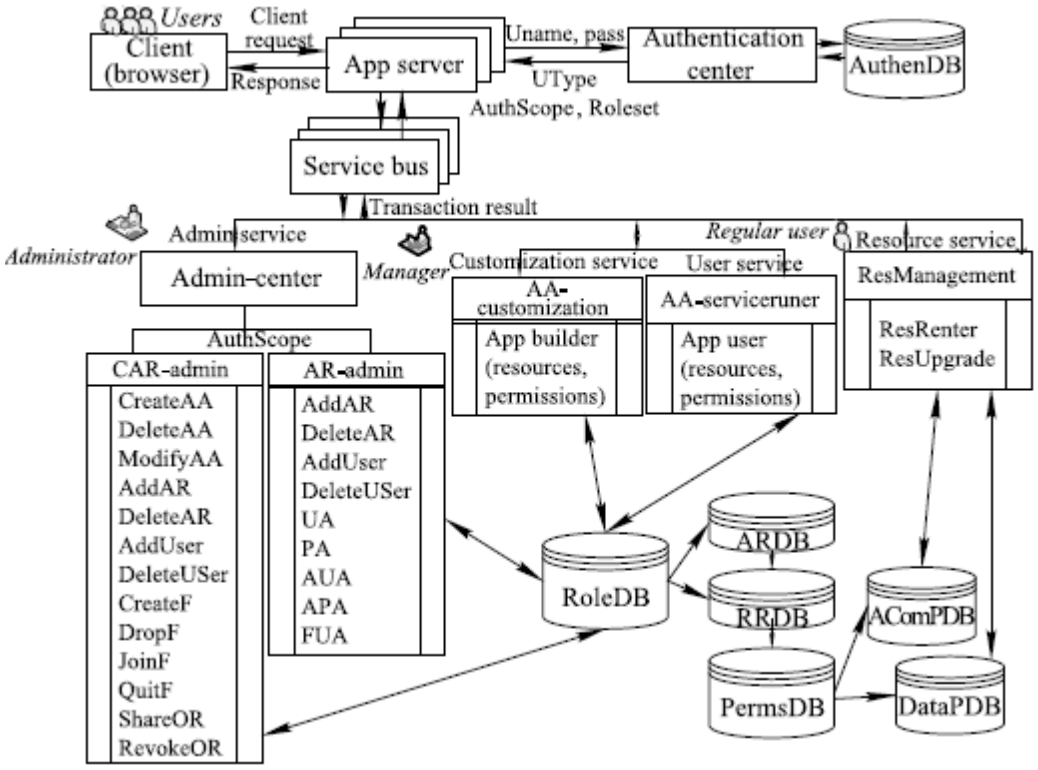
• طراحی دیتابیس کنترل دسترسی

طرح دیتابیس متادیتا برای کنترل دسترسی عمومی در STA SaaS در شکل 8 نشان داده شده است.

• طراحی ماژول کنترل دسترسی

دو نوع تفیض اختیار وجود دارد: تفیض اختیار سیستم و تفیض اختیار کاربر در پلتفرم های MTA و STA SaaS. تفیض اختیار سیستم AA و نهادها را در یک AA ایجاد کرده و دامنه اختیار را برای هر نهاد دارای اختیار مانند کاربران، نقش ها، مجوزها و منابع تعریف میکند. تفیض اختیار کاربر تخصیص های کاربر اعطا شده با AR ها در یک AA است. در یک AA یک مدیر مستاجر متفاوت از کاربر است. اولی می تواند مولفه ها را از AA های دیگر گرفته و شخصی سازی را انجام دهد اما آخری فقط می تواند از مولفه ها استفاده کند. لذا اجازه آنها برای مولفه های برنامه متفاوت است.

همانطور که در شکل 9 نشان داده شده است، وقتی کاربر وارد سیستم می شود، نام کاربری و رمز آن به مرکز احراز هویت فرستاده می شود.یک تگ با نوع دامنه کاربر و اطلاعات مجموعه نقش ها برای پیکربندی منابع مربوطه در سیستم برگشت داده می شود.




شکل 9. گردش کار کنترل دسترسی در SaaS دانش الکترونیک جعرافیایی.

سه نوع کاربر در سیستم وجود دارد: مدیران ایمنی، مدیران برنامه ای و کاربران برنامه. هر کاربر دارای ناحیه کاری مربوط به خود است. مجموعه نقشها به ما می گویند که کاربر می تواند به چه چیزهایی در ناحیه کاری اش دسترسی داشته باشد.

مدیران ایمنی و شخصی سازی برنامه به پلتفرم SaaS بعنوان مدیر و سرویس های شخصی سازی اضافه می شوند که روی باس سرویس با سرویس های یرنامه ای عادی و سرویس های منابع خالی می شوند.

طیق نوع کاربر یک کاربر می تواند به سرویس های مربوطه تحویل داده شود. دامنه اختیار تعیین میکند در چه محلی از AA کاربر کار کند.

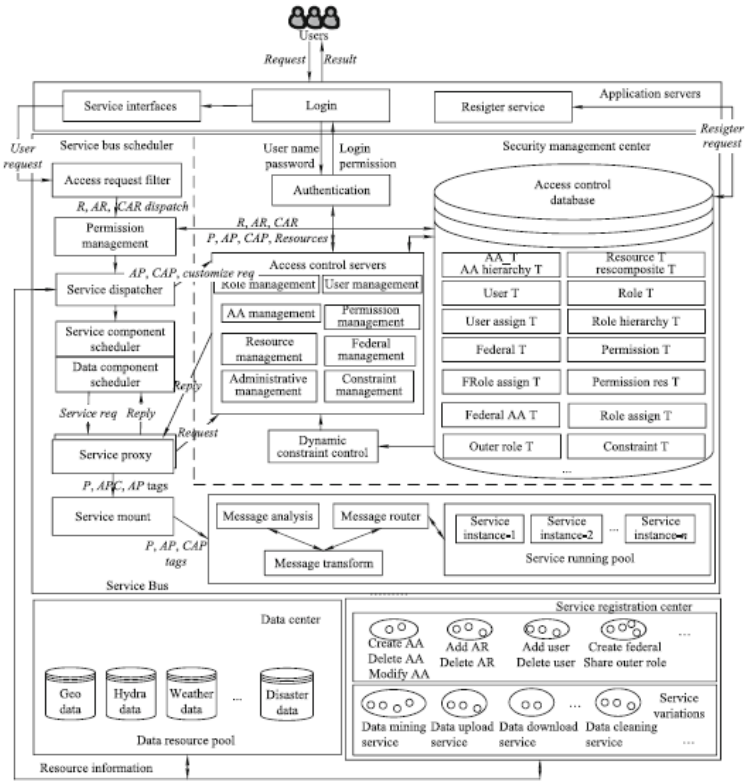
ماژول کنترل دسترسی پلتفرم SaaS دانش الکترونیک جعرافیایی در شکل 10 نشان داده شده است.

یک مرکز مدیریت ایمنی برای مدیریت کنترل دستری کل پلتفرم شامل چند سرور احراز هویت توزیع شده، سرورهای کنترل دسترسی، سرورهای کنترل محدودیت دینامیک و مرکز دیتابیس کنترل دسترسی ساخته شده است.

برنامه ریز باس سرویس سرویس های متناظر و سایر منابع را با نقش ها و مجوزهای کاربر ایجاد می کند. برای مقیاس پذیری سیستم وظایف می تواند با نقشهای مختلف ( مانند RR، AR و CAR) مجوزهای مختلف ( مانند P، AP و CAP) و دامنه های مختلف AA تقسیم شود. اجرای موازی برای تضمین کارایی و راندمان سیستم فراهم شده است.

## 6. نتیجه گیری و کار آتی

این مقاله به تحلیل ملزومات کنترل دسترسی پلتفرم های MTA و SaaS STA با تاکید بر جداسازی و اشتراک گذاری بین مستاجران پرداخته و به یک تعریف رسمی از یک مدل TMS-ARBAC می رسد. AA ها برای جداسازی هر مستاجر یا مستاجر دست دوم و محدود کردن نقش هاو کاربران در نواحی کار در AA بکار رفته است. هر مستاجر خود مدیر بوده و می تواند شخصی سازی را بسادگی با تغییر ملزومات برنامه انجام دهد. هر درخت AA یک ساختار با کوپل شل و بسیار بسط پذیر است که سلسله مراتب مستاجر و روابط مختلف اشتراک گذاری را توصیف می کند. برای کنترل دسترسی روی SaaS سرویس محور، مشکل ناحیه سیاه در پیکربندی AA زیرین، تعاریف دسته بندی منابع و مجوز و ویژگی های اشتراک گذاری مختلف در مدهای STA مختلف لحاظ شده اند. عملیات تفیض اختیار عمومی رای یک AA و استراتژی های مختلف اشتراک گذاری منابع روی یک پلتفرم ابری SaaS دانش الکترونیک جعرافیایی برای ارایه مدل داده و پیاده سازی شده اند.




شکل 10. طراحی ماژول کنترل دسترسی.

در حال حاضر روی استراتژی های مختلف کنترل دسترسی برای مدلهای مختلف STA کار می کنیم. پژوهش آتی روی مشخصات مقیاس پذیری و توزیع سیستم MTA SaaS و STA تمرکز می کند تا راندمان تفیض اختیار و احراز هویت را بهبود بخشد.

## References


1. Tsai W T, Zhong P. Multi-tenancy and sub-tenancy architecture in Software-as-a-Service (SaaS). In: Proceedings of the 8th IEEE International Symposium on Service Oriented System Engineering. 2014, 128–139
2. Sandhu R S, Coyne E J, Feinstein H, Youman C. Role-based access control models. IEEE Computer, 1996, 29(2): 38–47
3. Sandhu R, Bhamidipati V, Munawer Q. The ARBAC97 model for role-based administration of roles. ACM Transactions on Information and System Security, 1999, 2(1): 105–135
4. Yaish H, Goyal M. Multi-tenant database access control. In: Proceedings of International Conference on Computational Science and Engineering. 2013, 870–877
5. Zhong H, Wang W, Yan G, Lei Y. A role-based hierarchical administrative model. In: Proceedings of International Conference on Computational Intelligence and Software Engineering. 2009, 1–4
6. Bien N H, Thu T D. Hierarchical multi-tenant pattern. In: Proceedings of International Conference on Computing, Management and Telecommunications. 2014, 157–164
7. Li D, Liu C, Wei Q, Liu Z, Liu B. RBAC-based access control for SaaS systems. In: Proceedings of the 2nd International Conference on on Cloud Computing and Service Science. 2012, 426–431
8. Li D, Liu C, Liu B. H-RBAC: a hierarchical access control model for SaaS systems. International Journal of Modern Education and Computer Science, 2011, 3(5): 47–53
9. Cao J, Li P, Zhu Q, Qian P. A tenant-based access control model T-Arbac. Computer Science and Application, 2013, 3: 173–179
10. Xia L, Jing J. An administrative model for role-based access control using hierarchical namespace. Journal of Computer Research and Development, 2007, 44(12): 2020–2027
11. Tang B, Sandhu R, Li Q. Multi-tenancy authorization models for collaborative cloud services. In: Proceedings of International Conference on Collaboration Technologies and Systems. 2013, 132–138
12. Tang B, Li Q, Sandhu R. A multi-tenant RBAC model for collaborative cloud services. In: Proceedings of the 11th Annual International Conference on Privacy, Security and Trust. 2013, 229–238
13. Wang B, Huang H, Liu X, Xu J. Open identity management framework for SaaS ecosystem. In: Proceedings of IEEE International Conference on e-Business Engineering. 2009, 512–517
14. Tsai W T, Huang Y, Shao Q H. EasySaaS: a SaaS development framework. In: Proceedings of IEEE International Conference on Service-Oriented Computing and Applications. 2011, 1–4
15. Masood R, Shibli M A, Ghazi Y, Kanwal A, Ali A. Cloud authorization: exploring techniques and approach towards effective access control framework. Frontiers of Computer Science, 2015, 9(2): 297–321
16. Krebs R, Momm C, Kounev S. Architectural concerns in multi-tenant SaaS applications. In: Proceedings of the 2nd International Conference
17. Maenhaut P J, Moens H, Decat M, Bogaerts J, Lagaisse B, Joosen W, Ongenae V, De Truck F. Characterizing the performance of tenant data management in multi-tenant cloud authorization systems. In: Proceedings of IEEE/IFIP Network Operations and Management Symposium. 2014, 1–8
18. Weissman C D, Bobrowski S. The design of the Force.com multitenant Internet application development platform. In: Proceedings of ACM SIGMOD International Conference on Management of Data. 2009, 889–896
19. Wei S, Yen I L, Thuraisingham B, Bertinod E. Security-aware service composition with fine-grained information flow control. IEEE Transactions on Service Computing, 2013, 6(3): 330–343
20. Gong L, Qian X L. The complexity and composability of security interoperation. In: Proceedings of IEEE Symposium on Research in Security and Privacy. 1994, 190–200
21. Gong L, Qian X L. Cumputational issues in secure interoperation. IEEE Transactions on Software Engineering, 1996, 22(1): 43–52
22. Shafiq B, Joshi J B D, Bertino E, Ghafoor A. Secure interoperation in a multi-domain environment employing RBAC policies. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1557–1577
23. Lampson B W. Protection. ACM Operating Systems Review, 1974, 8(1): 18–24