الجمهورية العربيـة السـورية
المعهد العالي للعلوم التطبيقية والتكنولوجيا
العام الدراسي 2017/2016

منهجية لإدارة السماحيات في بنوك المعلومات الصغيرة

Methodology for Managing Permissions in Small Data Banks

دراسة أعدت لنيل درجة الماجستير في المعلوميات – نظم المعلومات ودعم القرار

إعداد

م. ياسر المفعلاني

إشراف

د.عصام غياض – د.محمد الجنيدي – د.مصطفى دقاق

دمشق في 26/7/2017

إهداء:

إلى شجرتي الزيتون العظيمتين، إلى من علماني الصبر والمثابرة، وغمراني دائماً بحبهما... أبي وأمي
إلى زوجتي الغالية الحنونة التي أستمد منها دائماً القوة والعزيمة لإنجاز أي عمل... وئام
إلى ملائكتي الصغار اللواتي أتنفس من ضحكاتهم... حامد، ماسة

نتقدم بالشكر إلى كل من

أتوجه بالشكر الجزيل للدكتور عصام غياض، الدكتور محمد الجنيدي، الدكتور مصطفى الدقاق، لكل ما قدموه من توجيهات ونصائح وأفكار وإشراف مميز لإتمام هذا العمل، وإخراجه بشكل مميز.

الملخّص

تحتاج أي شركة أو مؤسسة لنظام متين يكون قادر على إدارة بياناتها وحمايتها وتنظيم عمليات الوصول لهذه البيانات من قبل موظفي المؤسسة أو زبائنها. إن من أشهر نماذج التحكم في الوصول للموارد أو البيانات في وقتنا الحالي هما نموذج التحكم في الوصول المعتمد على الأدوار (Role base Access Control) ونموذج التحكم في الوصول المعتمد على السمات (Attribue Base Access Control).

لكلا النموذجين مجموعة من السيئات والمزايا التي يتمتع بها، إن اختيار نموذج لإدارة التحكم في الوصول للموارد في شركة ما يعتمد على بنية المؤسسة والسياسة المتبعة للوصول للموارد في هذه المؤسسة.

فإذاً لا يمكن الاعتماد على نموذج واحد وتطبيقه في جميع المؤسسات، ولكن بنفس الوقت تواجه أغلب المؤسسات مشكلة أن النموذج المختار لكي يناسب طبيعة هذه المؤسسة لا يخلو من بعض النقص، أو عدم القدرة على الالمام بجميع السيناريوهات أو السياسات المتبعة في هذه المؤسسة لتحقيق الوصول الآمن للموارد.

لذا سنقدم في هذه الرسالة نموذج مقترح للتحكم في الوصول يعتمد على تمديد نموذج التحكم في الوصول المعتمد على الأدوار بنموذج التحكم في الوصول المعتمد على السمات، بهدف الاستفادة من مزايا النموذجين معاً وحل مشكلة الوصول للموارد بشكل آمن في المؤسسات ذات الطبيعة الخاصة التي لا يستطيع فيها نموذج واحد إدارة عمليات الوصول.

هذه المؤسسات التي تملك العديد من البيانات والملفات المتنوعة بحاجة لبنك معلومات يضمن الحماية اللازمة لهذه البيانات، هذا البنك يفترض أن يكون بنك معلومات صغير لأنه موجه لمؤسسة ذات هرمية، وغير موجه لجميع العموم.

الكلمات المفتاحية: أنظمة التحكم في الوصول، نظام التحكم المعتمد على الأدوار، نظام التحكم المعتمد على السمات، **RBAC**، **ABAC**.

Abstract

Every company needs a reliable system which is able to manage its data and secure it, and allows its users the accessibility to the resources they are authorized to access. Role Based Access Control system (RBAC) and Attributes Based Access Control system (ABAC) are the most famous access control systems, each of them has some advantages and disadvantages.

Choosing the right model for a company depends on its hierarchy and the policy used in accessing its resources, a single model can’t be adapted in every company.

The main idea of this work is to develop a new model that depends on both RBAC and ABAC models so we can benefit from both systems advantages. We propose a hybrid access control model (AARBAC) which is a new access control model that uses the RBAC model and extends it with the ABAC model. The proposed solution is based on the roles and hierarchy applied on roles provided by RBAC model and extends it by adding attributes to the roles in order to benefit from the advantages of both models and solve the problem of accesing the resources safely in companies where a single model can’t work properly.

The companies that have a huge amount of data need a bank of information that provides the security to the data, this bank should be small because it exists in a hierarchical company and not directed to the public.

المحتويات

قائمة الأشكال

قائمة الجداول

الاختصارات

التحكم بالوصول التقديري	DAC: Discretionary Access Control
التحكم بالوصول المعتمد على الدور	RBAC: Role Based Access Control
التحكم بالوصول حسب الاستخدام	UCON: Usage access Control
التحكم بالوصول المعتمد على السمات	ABAC: Attribute Based Access Control

الفصل الأول

مقدمة

سنقوم في هذا الفصل بعرض مقدمة عن المشروع .بالإضافة إلى إشكالية وأسئلة البحث.

1.1– مقدمة

من المعروف أن من أكبر المشاكل التي تواجه المؤسسات الكبيرة هي عملية إدارة البيانات وعملية الوصول إليها من قبل الأشخاص المخولين بذلك. العديد من أنظمة التحكم في الوصول تم تطويرها مع مرور الوقت لحل هذه القضايا. يعد نظام Role Based Access Control ونظام Attribute Based Access Control من أشهر الأنظمة الحالية. تعتمد أغلب المؤسسات نظام RBAC، وقد أكد المعهد الوطني للمعايير والتكنولوجيا National Institute of Standards and Technology أن أغلب المؤسسات التي يزيد عدد موظفيها عن 500 تعتمد نظام RBAC للتحكم في الوصول للموارد[1].

يعتبر نموذج إدارة الصلاحيات المعتمد على الأدوار (RBAC) المهيمن على جميع النماذج في الوقت الحالي، وذلك من حيث التكلفة والبساطة، حيث يعتمد هذا النموذج على تعيين أدوار للمستخدمين حيث يسند كل مستخدم إلى دور أو أكثر، ومن جهة أخرى يسند مجموعة الصلاحيات أو السماحيات إلى كل دور.

بشكل عام عندما تسند السماحيات إلى الأدوار فإنها تبقى ثابتة خلال زمن طويل نسبياً وتبقى مهمة الإدارة هي عملية إعطاء وسحب الأدوار من المستخدمين. عندما يضاف مستخدم جديد نقوم بإضافته إلى دور معين وعندما ينتقل إلى عمل ما نقوم بسحب الأدوار منه وإعطائه أدوار جديدة تتناسب مع عمله الجديد.

تكمن المشكلة الأساسية في نظام RBAC أنه لم يضف أي سمة تخص المستخدم، أو الموارد كالأخذ بعين الاعتبار مكان وجوده عند أخذ قرار بامتلاكه الصلاحية أو عدم امتلاكه، بل اكتفى هذا النموذج بالقرار حسب الدور الذي يلعبه المستخدم.

وهذا سبب منطقي لأنه لو تم أخذ بعين الاعتبار هذه السمات سينتج عدد كبير من الأدوار والسمات، لذلك كان لابد من التفكير في اتباع نهج جديد قادر على معالجة سمات مختلفة للمستخدم، تكون قادرة من الحد من امكانية المستخدمين الذين لديهم أدوار شخصية متضاربة.

لهذا السبب كان لابد من إيجاد حل بديل يهتم بالسمات المختلفة للأشخاص أو الموارد، لذا كان التوجه نحو نظام المعتمد على السمات ABAC.

يعتمد هذا النظام على طريقة الوصول للقيام بعملية على غرض معين تمنح أو ترفض بحسب السمات المسندة لهذا الغرض، وشروط العمل، إضافة إلى مجموعة من السياسات المعرفة على السمات.
ونعني بالسمات هي خصائص الغرض أو المستخدم الذي يريد القيام بأي عملية على هذا الغرض، إضافة للشروط البيئية والمكانية.

[1] http://csrc.nist.gov/groups/SNS/rbac/

فالسمات هي معلومات عن هذا الغرض، بينما خصائص البيئة تكون مستقلة عن الأغراض أو المستخدمين، فمن الممكن أن تشمل الوقت الحالي، أحد أيام الأسبوع، مكان وجود المستخدم، أو مستوى التهديد الحالي الممكن حدوثه.

عندما يطلب المستخدم الأذن بالوصول لغرض معين، فإن سياسة هذا النظام تقوم بالبداية بتفحص قائمة السمات الخاصة بالغرض، والشروط البيئية، والفاعلين لاتخاذ القرار المناسب بالوصول أو لا.

فعلى سبيل المثال لو لدينا ملف فيديو وأراد أحد المستخدمين الوصول لهذا الملف، ومن سمات هذا الملف أنه لايمكن الوصول إليه إلا من قبل المستخدمين الذين أعمارهم فوق 18 سنة، إذاً سياسة الوصول في هذا النموذج تقوم في البداية بالنظر في حال كان المستخدم يحقق شروط السمات، فإذا كان المستخدم عمره فوق 18 يتم التحقق من الشروط البيئية الأخرى وطبيعة المستخدم. فمن الشروط البيئية الأخرى أن هذا الفيديو مخصص لبلدان معينة فقط على سبيل المثال، وطبيعة المستخدم. فمثلاً أن يكون مخصص للأشخاص ذوي شهادات معينة مثلاً، فعند تحقق أن الشخص عمره فوق 18 ويحمل شهادة الهندسة ومن فرنسا عندها يعطى إذن بالدخول، وإلا عدم تحقق أي شرط لا يسمح له بالدخول.

كمثال أخر على شروط الوقت والمكان يمكن للأطباء الدخول إلى سجلات المرضى في المشفى وفي ساعات الدوام فقط.

من الواضح أن نظام ABAC أكثر تعقيداً من نظام RBAC من حيث تحديد السياسات وتحليلها، ومن حيث مراجعة أو تغيير سماحيات المستخدمين. يتطلب نظام ABAC العديد من خطوات المعالجة وصولاً إلى اتخاذ القرار بناءً على السياسات.

إن اعتماد أحد النظامين في المؤسسات الكبيرة يتطلب مجموعة من التساؤلات من أهمها، هل يكفي نموذج واحد فقط لحل ومراقبة الوصول للموارد في المؤسسة؟

يعتبر NIST [9] المؤسس الرئيسي لنموذج RBAC من بداية التسعينات، فقد قدم أول وصف رسمي لنموذج RBAC في عام 1992، ومن ذلك الوقت لازلنا نرى الإضافات التي تقدم لهذا النموذج حتى وقتنا هذا، وهذا لا يعني عدم اكتمال النموذج وإنما ازياد متطلبات الحماية والمراقبة لدى المؤسسات، ولزيادة مخصصات التوفير التي يقدمها هذا النموذج من ناحية البرمجيات.

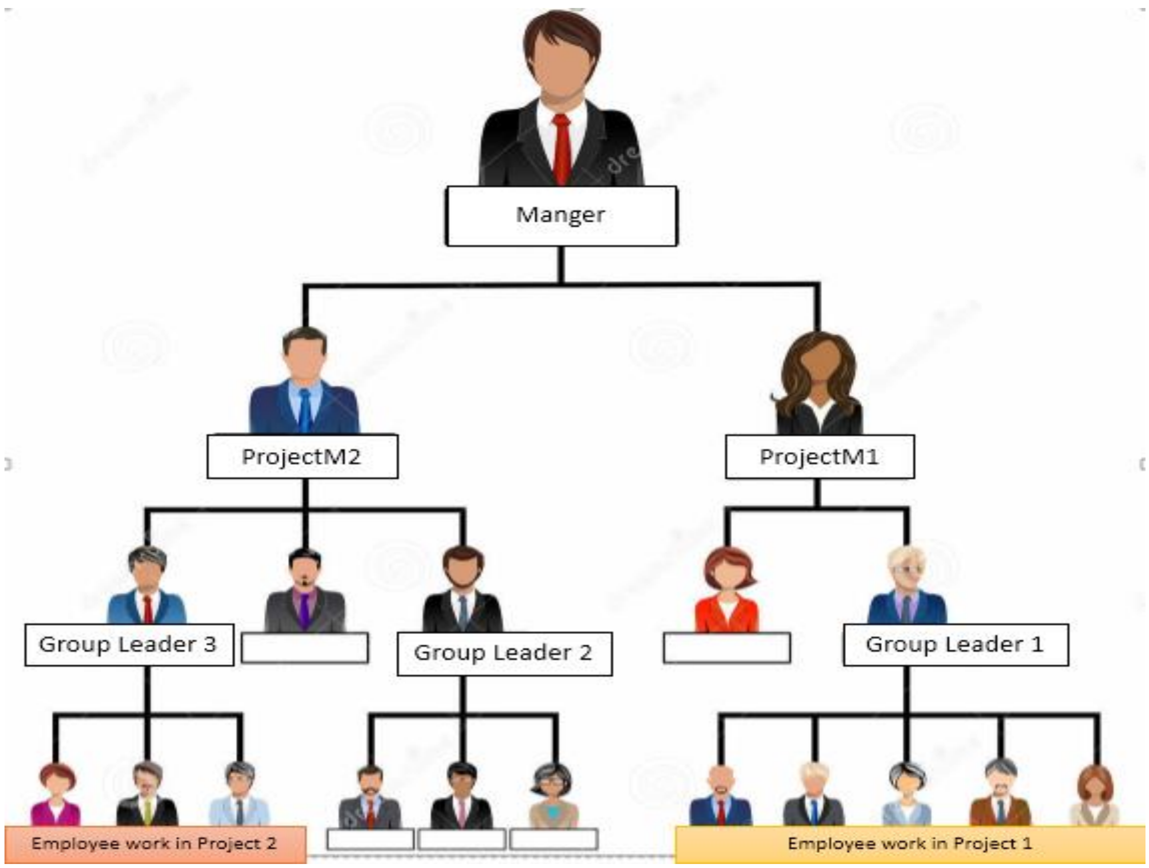
ويحتاج نظام ABAC للتكلفة والعمل الكبير على السياسات والسماحيات في حال وجود عدد كبير من الموارد أو المستخدمين.

في هذا العمل سنقدم نموذج مقترح يعتمد على الدمج بين المزايا التي يقدمها RBAC من حيث تسهيل الإدارة وتحليل السياسات واستعراض المستخدمين والسماحيات ومزايا ABAC من حيث الاعتماد على السمات والمرونة.

2.1– إشكالية البحث

تحتاج بعض المؤسسات لأنظمة تحكم في الوصول خاصة وفقاً لسياسة الوصول للموارد التي تتبعها المؤسسة أو وفقاً لهرمية المؤسسة الإدارية، فلا تستطيع أغلب نماذج إدارة صلاحيات التحكم بالوصول في عملية الوصول لموارد هذه المؤسسات من قبل موظفيها، بل تحتاج لبعض الإضافات أو الدمج مع نماذج أخرى لتحقيق السياسة الخاصة بالمؤسسة للوصول للموارد.

يمثل الشكل رقم 1 هرمية مؤسسة ما:



رسم توضيحي 1: هرمية مؤسسة ما

كما في الشكل 1، لدينا مدير عام لمؤسسة، ويتبع له مدراء مشاريع، وكل مدير مشروع يتبع له رؤساء فرق عمل، وكل رئيس فريق عمل لديه مجموعة من العاملين يعملون في مشاريع معينة.

السياسة العامة في الشركة تقوم على مايلي:

يقوم كل شخص في المؤسسة بتحميل البيانات والملفات بكافة أنواع وتخزينها ضمن بنك معلومات صغير، وفقاً للسماحية المسندة لدوره في هرمية المؤسسة، ما يتم القيام به من أعمال كتحميل ملفات أو التعديل عليها في مشروع معين من قبل

عناصر العمل المشاركين في المشروع، يحصل عليه من هم أعلى منهم في الهرمية، فمثلاً ما يحصل عليه عناصر فريق العمل في المشروع 1 من صلاحيات (كتحميل الملفات أو التعديل أو ... الخ) يحصل عليه مباشرةً كل من رئيس فريق العمل 1، ومدير المشاريع 1، و المدير العام للشركة.

وبنفس الوقت أي عملية يقوم بها عناصر فريق العمل بالمشروع كإضافة الملفات مثلاً، يجب أن يكون من أعلى منهم بالهرمية قادراً على رؤيتها.

فما هو نظام الصلاحيات المناسب لإدارة صلاحيات الوصول للملفات في هذه المؤسسة لتحقيق السياسة المتبعة في المؤسسة؟

في الحقيقة إن كل من نظام RBAC ونظام ABAC يعمل لتحقيق السياسة المتبعة في المؤسسة، لكن إذا أخذنا بعين الاعتبار احتمال توسع الشركة وازدياد عدد الموظفين والمشاريع، فإن تطبيق نظام ABAC سيشكل عبئاً على المؤسسة، فمن حيث تحديد سمات الموظفين والمشاريع والملفات التي بداخلها.

لذلك من الأفضل تطبيق نظام RBAC وخاصة أنه يدعم موضوع الهرمية، بحيث يرث الأدوار الأعلى في الهرمية من الأدوار الأدنى.

إذاً أين تكمن المشكلة التي تتم معالجتها؟

تكمن المشكلة أنه من الممكن أن يشارك عناصر تابعين لرئيس مجموعة 2 في مشروع لرئيس مجموعة 1وهم غير تابعين له في الهرمية.

فإذا ما تم تطبيق نظام RBAC وإضافة عناصر من المجموعة 2 للعمل في مشروع تابع للمجموعة 1، عندها لا يمكن لرئيس المجموعة 1 وما فوقه في الهرمية أن يرث صلاحيات عناصر المجموعة 2 أو أن يرى مجموعة العمليات التي يقومون بها في المشروع 1.

فما يقدمه نظام RBAC هو أن يسمح لرئيس المجموعة 2 ومن فوقه بالهرمية برؤية ووراثة صلاحيات أبناءه فقط.

وهذا يتناقض مع سياسة المؤسسة التي تنص أيضاً على قدرة رئيس المجموعة التابع لها المشروع ومن فوقه بالهرمية على رؤية كافة العمليات التي تتم في المشروع سواءً كانت من العناصر التابعين له أو لا.

من الواضح أن تطبيق نظام RBAC في مثل هذه المؤسسات لا يكفي لوحده، لذا سنقترح نظام جديد يقوم على المحافظة على نظام RBAC وتمديده بنظام ABAC لحل مشكلة المؤسسة.

3.1– أسئلة البحث

1.3.1– ما هي المتطلبات الواجب توافرها في النموذج للوصول للحل المراد؟

يجب على النموذج أن يكون قادر على حماية البيانات من قبل الأشخاص الغير المخول لهم بالوصول إليها، مع الأخذ بعين الاعتبار التداخلات والتشعبات في هرمية المؤسسة.

2.3.1– هل تكفي نماذج أنظمة الوصول الحالية لحل مشاكل الهرميات الخاصة؟

لا تكفي أنظمة الوصول الحالية لحل مشاكل الهرميات الخاصة، فالنموذج المعتمد على الأدوار لا يعالج الهرميات التي لا تعرف ترتيباً جزئياً موحداً على كامل الهرمية بين عناصرها، بحيث يسمح بوراثة السماحيات بين عناصر ترتبط وفق هذا الترتيب.

حيث نناقش في هذا البحث تداخل بعض الهرميات التي تؤدي إلى نشوء التضاربات في الهرميات التي لا تعرف ترتيباً جزئياً بين عناصرها، فالمقصود من الترتيب الجزئي هو إمكانية تحديد وجود علاقة وراثة من عدمها بين أي عنصرين بناءً على علاقة ترتيب معرفة رياضياً بشكل مسبق ومحدد.

بينما تطبيق نموذج المعتمد على السمات سيحل المشكلة في حال عدد المستخدمين والموارد قليل، أما في حال العكس سيكون عبء كبير على الشركة من حيث الإدارة وتعريف العديد من السمات للأشخاص والموارد.

3.3.1– اذا لم يتوافر نموذج يقودنا إلى الحل فهل يمكن دمج عدة نماذج لتحقيق الهدف؟

يدعم النموذج المعتمد على الأدوار الهرميات والوراثة من خلال تعريف الأدوار، بحيث ترث الأدوار السماحيات من الأدوار التي تحتها في الهرمية، ويدعم النموذج المعتمد على السمات عملية تعريف السياسات المختلفة المتعلقة بسمات الفاعلين والأغراض والبيئة.

من الممكن الاستفادة من خواص كل نموذج، بدمج النموذجين معاً بهدف حل مشكلة البحث المعرفة مسبقاً.

4.1– مخطط البحث

تم تنظيم البحث كما يلي:

- **الفصل الثاني**: يحتوي هذا الفصل دراسة مرجعية تتضمن عرضاً للإطار النظري الخاص بكل من أنظمة إدارة صلاحيات الوصول ومساوئ ومحاسن كل نظام.

تم في هذا الفصل توضيح بدايات ظهور أنظمة التحكم في الوصول، وعرض معظم أنظمة التحكم في الوصول منذ بداية ظهورها حتى وقتنا الحالي، كما تم استعراض مجموعة من النماذج التي تم فيها دمج نموذج معتمد على الأدوار ونموذج معتمد على السمات وهدف هذه النماذج هو حل الحالات ذات الطبيعة الخاصة.

- **الفصل الثالث**: يحتوي هذا الفصل على شرح مفصل عن المنهجية المقترحة للإجابة على أسئلة البحث، والنموذج الجديد المقترح لتحقيق عملية الوصول للموارد في المؤسسات ذات الهرمية الهجينة، والميزات الجديدة التي تحققت بفضل الشكل الجديد للنموذج.
- **الفصل الرابع**: يستعرض هذا الفصل عدداً من الاختبارات لتطبيق الحل المقترح، ومناقشة هذه النتائج.
- **الفصل الخامس**: الخاتمة والآفاق المستقبلية.

الفصل الثاني

دراسة مرجعية

سنقوم في هذا الفصل بعرض دراسة مرجعية تتضمن عرضاً للإطار النظري الخاص بكل من أنظمة إدارة صلاحيات الوصول ومناقشة وتحليل مساوئ ومحاسن كل نظام والمقارنة بينها.

1.2– مقدمة

إحدى أهم القضايا الأمنية هي عملية الوصول إلى البيانات من أصحاب الصلاحيات فقط. يفترض أن تكون الأنظمة مزودة بنموذج تحكم بالوصول يحقق سياسة أمنية معينة توافق احتياجات مزود الخدمة وتضمن منع عمليات الوصول الغير مصرح بها وتمنع المستخدمين من الوصول إلى أغراض لا يملكون صلاحية الوصول إليها. يجب أن يضمن نموذج التحكم بالوصول متابعة المستخدمين وتحقيقهم لشروط الصلاحية والشروط التي تفرضها البيئة. نعرض فيما يلي أهم النماذج المستخدمة حالياً بعد فقرة التعاريف الأساسية.

الهدف الرئيسي من نماذج التحكم بالوصول، هو تأمين نظام حماية وتحقق من الصلاحيات وذلك من أجل حماية المعلومات من أي خروقات أمنية بغض النظر عن عمليات المستخدم.

2.2– تعاريف أساسية

1.2.2– العناصر الأساسية

- الفاعل **Subject:** وهو العنصر الذي **ي**ُمثل المستخدمين أو البرامج التي تمارس نشاط معين مثل التخاطب مع الفاعلين الآخرين أو العناصر الأخرى (على سبيل المثال: المستخدمين، المعالجات، البرامج).
- الأغراض **Objects:** وهي العناصر التي يتم تطبيق النشاطات عليها (الملفات، الأجهزة، البرامج).
- الحقوق **Rights:** وهي السماحية **أو** مجموعة من السماحيات التي يملكها الفاعل لتطبيق نشاط على الأغراض مثل (القراءة والكتابة والإنشاء والتنفيذ create، read، write، execute).

2.2.2– بنوك المعلومات

1.2.2.2– مقدمة

بنك المعلومات هو مجموعة بيانات وملفات ضخمة من المعلومات بكافة أشكالها وصورها، مخزنة ومحتفظ بها بشكل يسهل التعامل معها والبحث عنها بواسطة الحاسب، ويتألف من مباني ضخمة مكونة من العديد من الخوادم التي تعمل على مدار 24/24 دون انقطاع لكل من الأنترنت والكهرباء عنها، بحيث يكون هذا المبنى مجهز بأحدث التقنيات، ويعتبر بنية تحتية منهجية مؤلفة من العديد من تقنيات المعلومات والاتصالات، وأليات الحماية والأمن التي تتيح للمستخدم استخدام نظم المعلومات والخدمات التي يقدمها.

2.2.2.2– أنواع بنوك المعلومات[2]

- بنوك معلومات صغيرة: تتكون على الأكثر من 10 مخدمات ووحدة تخزين 10TB، مثل بنوك المعلومات المستخدمة لتخزين ملفات وبيانات المؤسسات.
- بنوك المعلومات المتوسطة: تتكون من أكثر من 10 مخدمات ووحدة تخزين 100TB، مثل بنوك المعلومات المستخدمة لحجز نطاقات الويب Web Hosting.
- بنوك معلومات كبيرة: تتكون من أكثر من 100 مخدم ووحدة تخزين أكثر من 100TB، تستخدم مثل هذه البنوك في الهيئات والمؤسسات الحكومية والشركات التي تتعامل مع حجم كبير من المعلومات وكم كبير من العمليات عليها.

وتقسم بنوك المعلومات حسب عملها إلى:

- بنوك معلومات للمعالجة Processing.
- بنوك معلومات للتخزين storage.
- بنوك معلومات كمزود خدمات التطبيقات Application Service Providing.

3.2.2.2– أهمية بنوك المعلومات

يمكن الاستفادة من بنوك المعلومات في المجالات التالية:

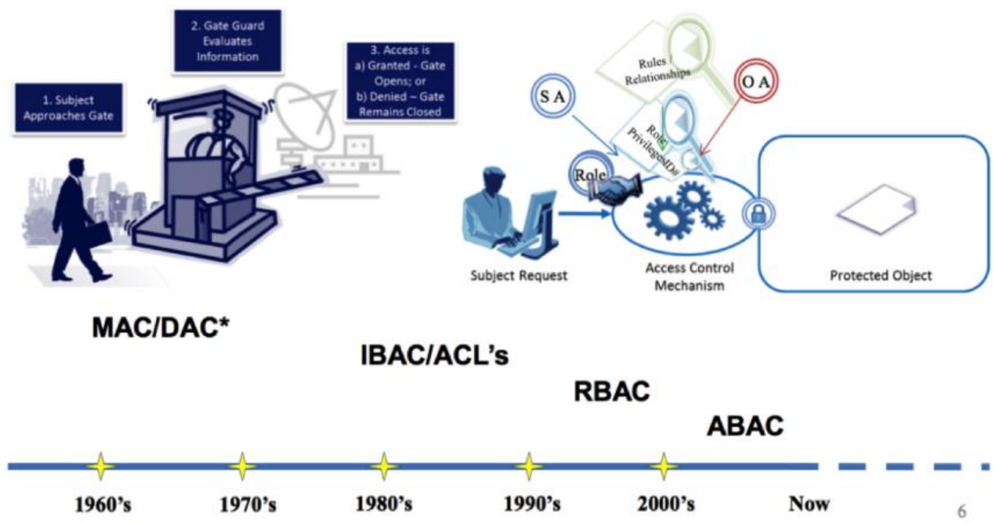
- إدارة ومعالجة البيانات بسرعة ودقة، خاصة في حالة استخدام بيانات مثل الصور وملفات الفيديو والخرائط.
- تكون إمكانية التحكم في البيانات قوية وأمنة ضد الهجمات والمخاطر.
- مجهزة بنظام إدارة الأزمات، للعمل ضمن ظروف خاصة.
- امكانية تشفير البيانات وتناقلها بشكل أمن.
- إمكانية استضافة مجالات الويب والخدمات والشبكات الافتراضية.

3.2– نماذج التحكم بالوصول العامة

بدأت فكرة إدارة صلاحيات الوصول للبيانات بالتطور منذ عام 1960، وذلك من أجل حماية البيانات والمعلومات من الاختراق أو التزوير، حيث يمثل الشكل رقم 2 مراحل تطور نماذج إدارة الصلاحيات[1].

---

[2] https://en.wikipedia.org/wiki/Data_center



رسم توضيحي 2: مراحل تطور نماذج التحكم بالوصول

1.3.2– نموذج التحكم الأمنية الإلزامية Mandatory Access Control

تعتبر هذه الطريقة من أكثر الطرق صرامة، ففي هذه الطريقة لا يمكن للمستخدم منح سماحية الوصول إلى البيانات لمستخدم آخر. بل على العكس يوجد سياسة شاملة ومعرفة مسبقاً، تنتج عنها السماحيات وفق انتماء العناصر إلى مستويات أمنية محددة تطبق عليها سياسة ضبط الوصول الشاملة، و لايوجد هنالك مرونة في تحديد السماحيات على مستوى البيانات.

يقوم هذا النموذج بعملية ضبط للسياسات الأمنية [2] ويرتبط هذا النموذج بنموذج Bell-LaPudula 1973. وهو يقوم على إسناد علامات أمنية لكل من الأغراض والفاعلين Subjects, Objects مرتبة بشكل تسلسلي حسب السرية Top Secret، Secret، Confidential،Unclassified. يستخدم خاصيتين هما خاصية السرية البسيطة، وخاصية السرية المعقدة، ويستخدمها من أجل ضبط الإجراءات الأمنية العسكرية، حيث تستخدم الأولى من أجل منع أي إجرائية من الوصول إلى معلومات مصنفة بشكل أعلى. والثانية بمنع الإجرائيات من الكتابة على أغراض ذات تصنيف أقل.

يتم استخدام الخاصيتين السابقتين من أجل تحديد سياسات للوصول إلى الأغراض حسب قرارات الحماية التي يتخذها مالك البيانات. باستثناء الأغراض العسكرية فإن هذا النموذج لا يعد من النماذج واسعة الإنتشار وذلك لصعوبة توصيف معظم سياسات الوصول.

إذا في هذا النموذج يوجد سياسة شاملة ومعرفة مسبقاً لتحديد صلاحية الوصول، لا يمكن للمستخدم منح صلاحية و إنما يوجد سياسة اجبارية مفروضة على الجميع، ويتم تصنيف المستخدمين والبيانات في مستويات أمن، حيث يمثل الجدول رقم 2 طريقة تمثيل نماذج التحكم الأمنية الإلزامية.

جدول 1: طريقة تمثيل نماذج التحكم الأمنية الإلزامية

Subject Security Level	Object Label		
	Top Secret	Secret	Confidential
Top Secret	Allow	Allow	Allow
Secret	Deny	Allow	Allow
Confidential	Deny	Deny	Allow

2.3.2– نموذج التحكم المبني على القرارات الشخصية Discretionary Access Control

تعتبر هذه الطريقة من أقل الطرق صرامة، ففي هذه الطريقة يمكن للمستخدم منح صلاحية الوصول إلى البيانات لمستخدم آخر. وتحتوي هذه الطريقة على شيئ من الخطورة، حيث يمكن إعطاء مستخدم صلاحيات لا يجب أن يملكها.

بعكس النموذج السابق [2] يسمح هذا النموذج للفاعلين بتقدير السماحيات المراد منحها للفاعلين الآخرين على الأغراض التي يملكونها. يتم ذلك من خلال استخدام مصفوفة الوصول Access Matrix والتي تكون السطور فيها عبارة عن الفاعلين، بينما الأعمدة هي عبارة عن الأغراض، نقاط التقاطع بينها هي عبارة عن السماحيات التي يملكها الفاعل من أجل الغرض، يمثل الجدول رقم 3 نموذج التحكم بالوصول المبني على القرارت الشخصية.

جدول 2: نموذج التحكم بالوصول المبني على القرارات الشخصية

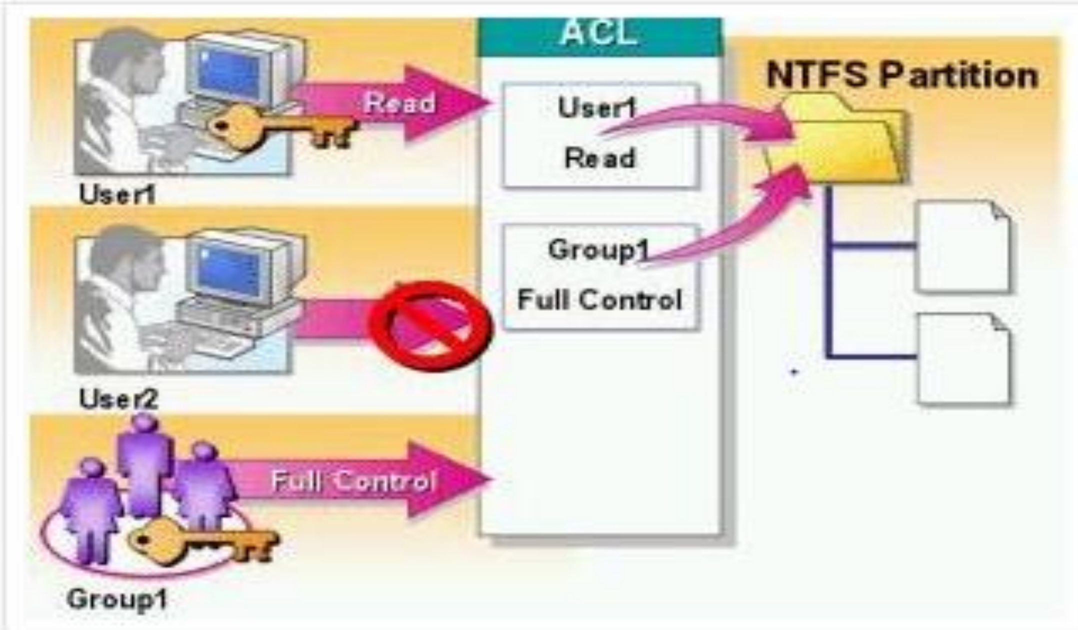
Subject	Resource	Authorization
Alice	Report	Allow
Alice	Finance	Deny
Alice	Customer	Allow
Bob	Report	Allow
Bob	Finance	Deny
Bob	Customer	Deny

ولكن رغم ذلك فإن حجم هذه المصفوفة من مرتبة $O(n^2)$، في حالة المصفوفة ممتلئة فإنه لا مشكلة بالحجم ولكن في الحقيقة إن معظم الفاعلين ليس لهم سماحية على معظم الأغراض وبالتالي فإن البحث في المصفوفة السابقة مُكلف ويأخُذ الكثير من الوقت، لذلك تم الاستعاضة عنها بسلسلة من سماحيات الوصول خاصة بكل غرض على شكل ملف.

3.3.2– سلاسل التحكم بالوصول Access Control Lists

وهي عبارة عن جدول مرتبط بكل غرض يعبر في أسطره على الفاعلين مع السماحية المقابلة له من أجل هذا الغرض فقط. هذه السلاسل هي المعتمدة بشكل رسمي في Unix. يمكن اعتبار كل عمود من مصفوفة الوصول السابقة سلسلة تحكم بالوصول مستقلة. يتطلب ما سبق من نظام التشغيل قبل السماح للمستخدم بالوصول إلى غرض، البحث ضمن السلسة الخاصة به (الغرض) للتحقق من السماحيات.

رغم ما سبق فإن زيادة عدد الأغراض والفاعلين يمكن أن يشكل مشاكل على صعيد الأداء، يمثل الشكل رقم 3 نموذج سلاسل التحكم في الوصول.



رسم توضيحي 3: نموذج سلاسل التحكم في الوصول

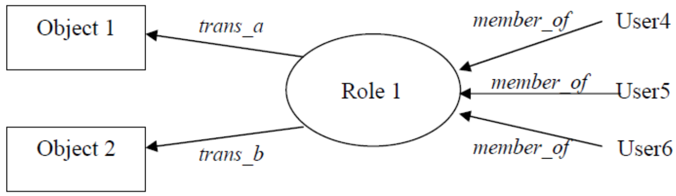
4.3.2– نظام التحكم بالوصول المعتمد على الدور Role Based Access Control

1.4.3.2– مقدمة

الدور هو عبارة عن مجموعة من السماحيات، غالباً ما تتخذ قرارات التحكم بالوصول حسب الأدوار التي تسند للمستخدمين حسب موقعهم في المؤسسة. وهي تضمن تفصيلات العمل والمسؤوليات والمؤهلات. على سبيل المثال في المشفى يمكن تقسيم الأدوار بين الممرضين والأطباء والإداريين ونفس الأمر بالنسبة إلى البنوك أو أي مؤسسة أخرى.

يمكن توصيف الدور كإجرائيات processes على أغراض تُسند إلى مستخدم أو عدة مستخدمين، كمثال يسند إلى الطبيب سماحية "إجراء زيارة له من قبل المريض ووصف الدواء اللازم له" (إجرائية process)، بينما دور الصيدلاني هو "عملية بيع الدواء للمريض" وليس من سماحياته "وصف دواء" له. مدير النظام هو الذي يقوم بعملية إدارة العلاقات بين الأدوار وحذفها وإسنادها.

يربط بالدور إجرائية أو عدة إجرائيات processes ، كما يُربط بالدور مستخدم أو عدة مستخدمين، الشكل(4) يوضح العلاقة بين الأدوار والأغراض والمستخدمين.



رسم توضيحي 4: العلاقة بين الأدوار والأغراض والمستخدمين في نموذج **RBAC**

ترتبط الإجرائيات بأغراض ويستطيع المستخدم الوصول إلى الأغراض من خلال هذه الإجرائيات. يمكن تجميع الأدوار المختلفة من أجل إعطاء سماحيات مشتركة لها مع صلاحيات مميزة أخرى لكل منها.

2.4.3.2- توصيف RBAC

من أجل الفاعل (المستخدم) Subject، الدور النشط هو الدور الذي يستخدمه المستخدم حالياً[3]:

AR(s:subject) = {the active role for subject s}

يمكن أن يُسند أكثر من دور لمستخدم معين:

RA(s:subject) = {authorized roles for subjects}

كل دور يسمح بإجراء معين أو عدة إجرائيات:

TA(r:role) = {transactions authorized for role r}

يستطيع المستخدم أن يُفعِل إجرائية ما ونرمز لها بالتابع exec(s,t) والذي يأخذ قيمة true إذا استطاع الفاعل تنفيذ الإجرائية.

$$exec(s{:}subject, t{:}transaction) = true\ iff\ s\ can\ execute\ transaction\ t$$

3.4.3.2- القواعد الرئيسية في RBAC

1- إسناد الأدوار[3]

يمكن للمستخدم أن يقوم بتفعيل إجرائية إذا وفقط إذا تم إسناد أدوار إليه.

$$\forall s{:}subject, t{:}transaction, (exec(s,t) \Rightarrow AR(s) \neq \emptyset)$$

عملية تسجيل الدخول لا تعتبر إجرائية process أما غيرها من نشاطات المستخدمين داخل النظام تعتبر إجرائيات تحتاج إلى سماحيات من خلال الأدوار.

2- ترخيص الدور[3]

الدور النشط الحالي للمستخدم يجب أن يكون مسند له كما يلي:

$$\forall s: subject, \left(AR(s) \subseteq RA(s)\right)$$

تضمن هذه القاعدة مع القاعدة الأولى، عدم تمكن المستخدمين من الحصول على أدوار غير مرخص لهم بها.

3- ترخيص الإجرائيات[3]

يستطيع المستخدم أن يُفعِل إجرائية ما إذا كان بالدور النشط حالياً للمستخدم.

$$\forall s: subject, t: transaction, (exec(s,t) \Rightarrow t \in TA(AR(s))$$

نظام التحكم بالوصول RBAC لا يقوم بالتحقق من سماحية المستخدم بالوصول إلى عناصر البيانات، حيث يصل إليها المستخدم من خلال إجرائية process تضمن له الوصول للأغراض التي تحوي البيانات بحصوله على أدوار معينة.

4- يُمكن إعادة تعريف الإجراء process لتشمل دراسة سماحية المستخدم بالوصول إلى الأغراض كما يلي[3]:

$$\forall s: subject, t: tran, o: object, \left(exec(s,t) \Rightarrow access(AR(s), s, t, x)\right)$$

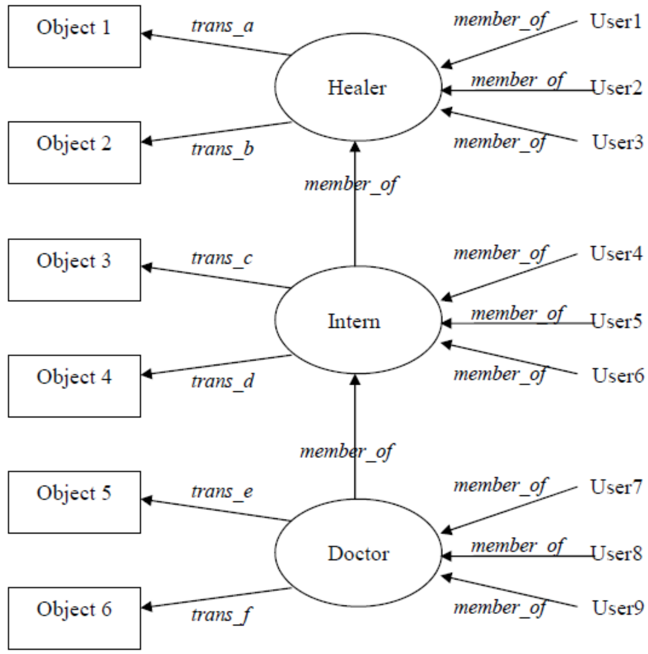
يشير الـ access(AR(s),s,t,x) إلى سماحية وصول المستخدم Subject إلى الغرض o من خلال المعاملة t في النمط x والذي يأخذ إحدى القيم (read,write,append).

كمثال على استخدام القاعدة الرابعة يمكن للطبيب في المشفى أن يُزوَد بسماحيات قراءة وكتابة على ملف وصفات معين، بينما الصيدلي لا يملك سوى سماحية قراءة على هذا الملف.

4.4.3.2- الإدارة الأمنية المركزية بالاعتماد على RBAC

تعد RBAC مرنة حيث أنها تأخذ هرمية المؤسسة كبنية وسياسة أمنية، وأحد أهم مناقب هذا النموذج هو الإدارة المرنة فيها. بشكل عام عندما تُسند الإجرائيات process إلى الأدوار فإنها تبقى ثابتة خلال زمن طويل نسبياً وتبقى مهمة الإدارة هي عملية إعطاء وسحب الأدوار من المستخدمين. عندما يُضاف مستخدم جديد، نقوم بإضافته إلى دور معين. وعندما ينتقل إلى عمل ما، نقوم بسحب الأدوار منه وإعطائه أدوار جديدة تتناسب مع عمله الجديد.

يمكن أن يرث الدور صلاحيات دور آخر وأن يُضاف إليه صلاحيات أخرى. الشكل رقم 5 مثال عن العلاقات متعددة الأدوار[4].



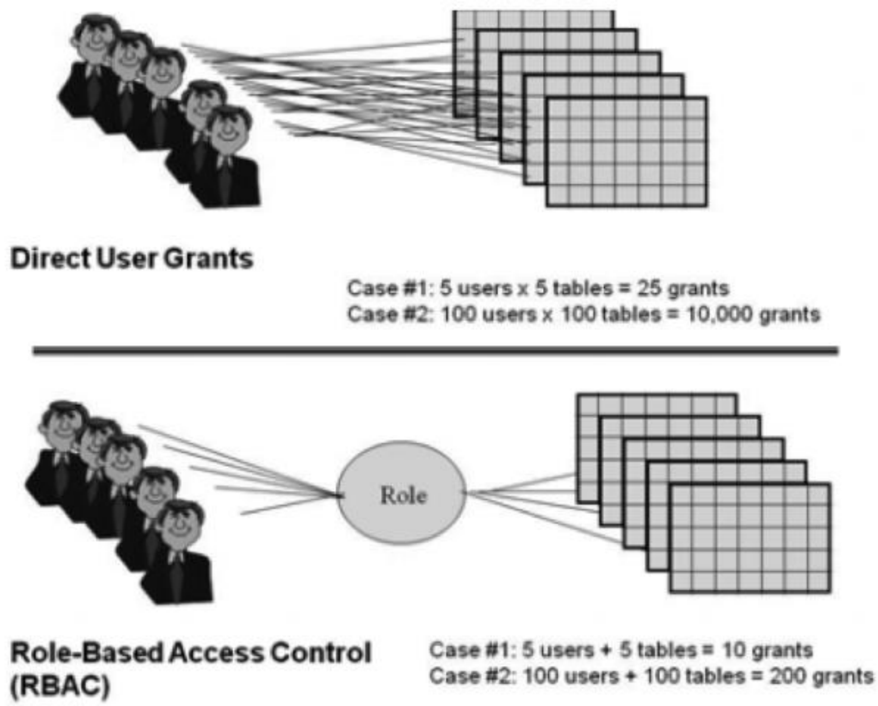
رسم توضيحي 5: العلاقة المتعددة بين الأدوار

يمكن للطبيب الحصول على الأدوار الخاصة بكل من المتدرب والمعالج. بنفس الوقت لا يستطيع المعالج أو المتدرب الحصول على صلاحيات للإجرائيات الخاصة بالطبيب.

5.4.3.2– مراحل تطور نموذج إدارة الصلاحيات المعتمد على الأدوار

يُعتبر نموذج إدارة الصلاحيات المعتمد على الأدوار المهيمن على جميع النماذج في الوقت الحالي، وذلك من حيث التكلفة والبساطة، حيث يعتمد هذا النموذج على تعيين أدوار للمستخدمين حيث يُسنَد كل مستخدم إلى دور أو أكثر، ومن جهة أخرى يُسنِد مجموعة الصلاحيات أو السماحيات إلى كل دور.

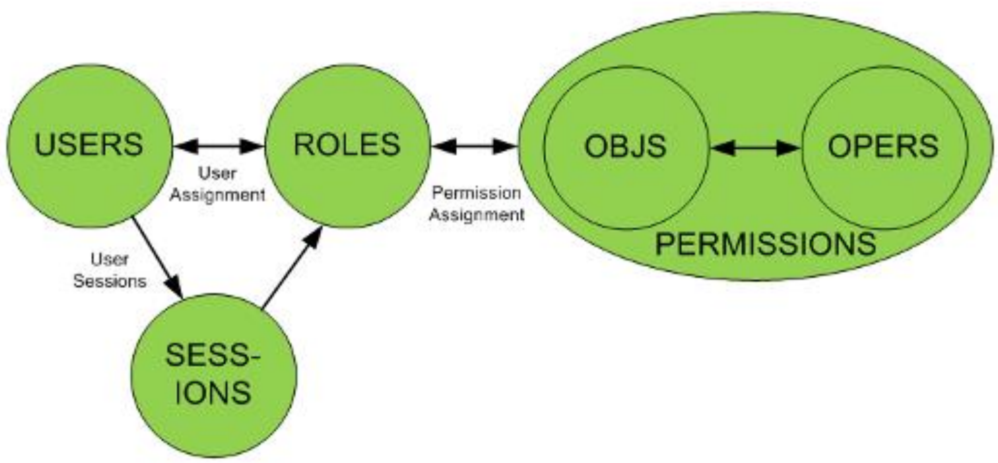
يوضح الشكل رقم 6: الفائدة الرئيسية لهذا النموذج فبدلاً من تعيين مجموعة من الأُذونات الضخمة والمسندة مباشرة إلى المستخدمين، يتم تعيين الأُذونات بحسب الأدوار وبالتالي أدوار مستقرة لن تتأثر بتغيير المستخدمين، إضافة إلى أن التسلسل الهرمي يجعل الأدوار الأعلى في الهرمية ترث الصلاحيات من الأدوار الأدنى منها، وهذا يُعطي النموذج البساطة والفعالية من حيث التكلفة[5].



رسم توضيحي 6: مثال على أهمية **RBAC**

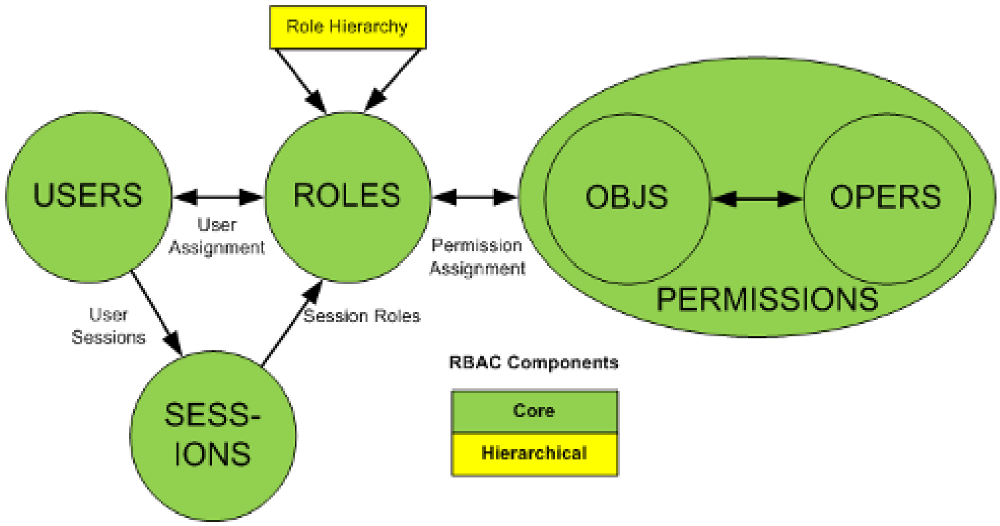
وقد مر نموذج إدارة الصلاحيات المعتمد على الأدوار بعدة مراحل هي [7] [6]:

- Flat RBAC: وهو النموذج البدائي والأساسي لمفهوم إدارة الصلاحيات المعتمدة على الأدوار، حيث يتم إسناد المستخدمين للأدوار ويتم إسناد أُذونات أو صلاحيات إلى الأدوار والموضح بالشكل رقم 7.



رسم توضيحي 7: **Flat RBAC**

- Hierarchical RBAC: هذا النموذج يُبرهن كفاءة إدارة الصلاحيات المعتمدة على الأدوار من خلال خفض التكاليف الإدارية. فالأدوار التي في المستوى الأعلى (الآباء) ترث السماحيات والأُذونات من الأدوار الدنيا (الأبناء) دون الحاجة إلى تعيين هذه الأُذونات، ويوضح الشكل رقم 8 هذا النموذج.

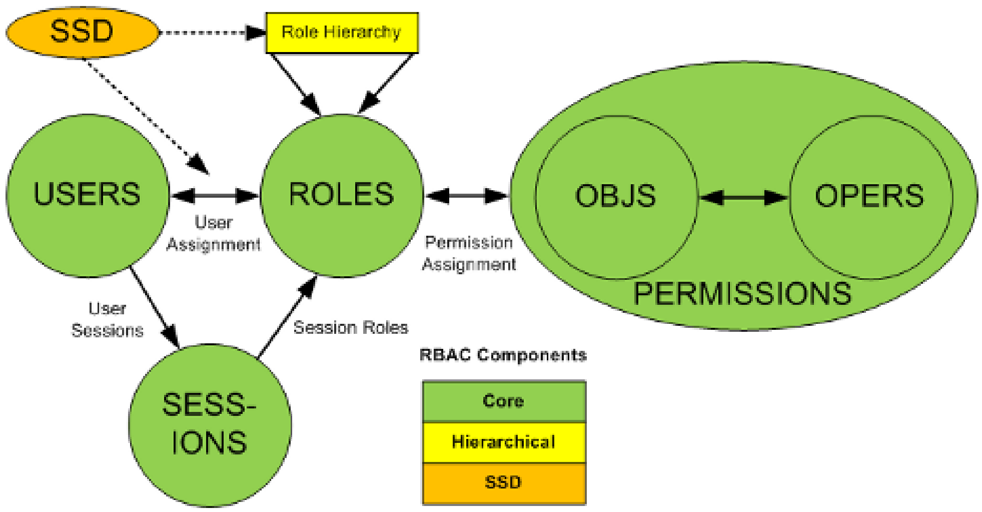


رسم توضيحي 8: **Hierarchical RBAC**

- Constrained RBAC:

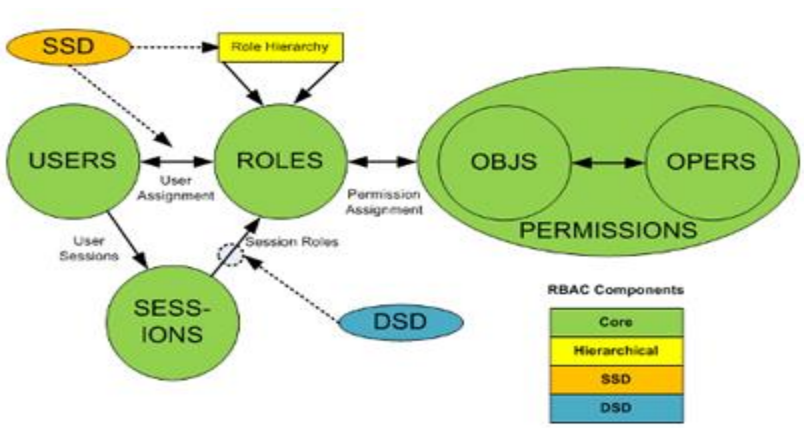
في هذا النموذج يُضاف شرط للنماذج السابقة لفرض الفصل بين الواجبات.

الفصل بين الواجبات يمنع التلاعب والأخطاء الناتجة من إعطاء مستخدم مستوى مقبول من السلطة أو السماحيات لمهام عمله، فمثلاً دور معين مسؤول عن تحديد النفقات لمنتج معين، والدور الآخر مسؤول عن الموافقة على هذه النفقات. ففي هذه الحالة يجب عدم إسناد نفس الشخص لهذين الدورين معاً، وهذا ما يُدعى بالفصل الثابت بين الواجبات static separation of duties كما هو موضح بالشكل رقم 9:



رسم توضيحي 9: **static separation of duties**

بينما هنالك فصل آخر بين الواجبات هو الفصل الديناميكي بين الواجبات dynamic separation of duties، وهو الذي يسمح بإسناد أدوار متضاربة لنفس الشخص لكن ليس في نفس الوقت، كما في الشكل رقم 10:



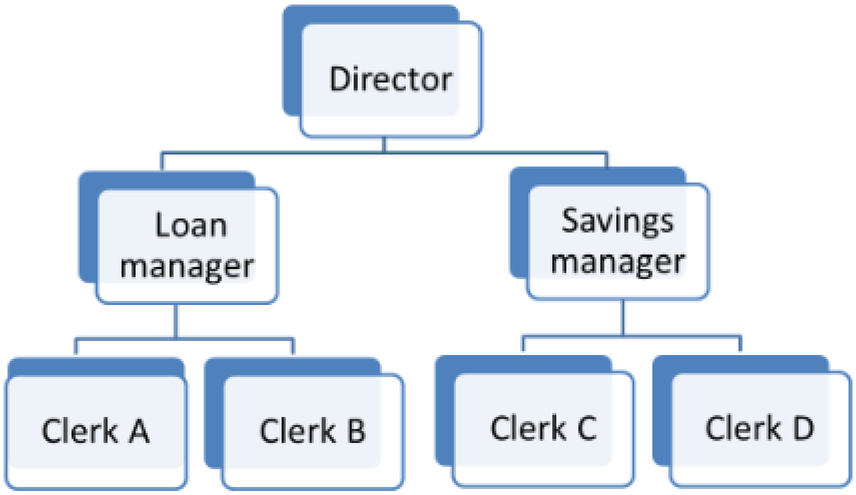
رسم توضيحي 10: **dynamic separation of duties**

6.4.3.2– تضارب المصالح Conflicts of interests

تَنتُج تضارب المصالح عند إسناد سماحيات واسعة، كمثال توكيل مهام استشارية لخبير لصالح شركتين متنافستين، عندها تتدخل الأمور الشخصية في العمل، حيث يكون المستخدم مسند له دور معين في العمل وينفذ قرارات تضر بسياسة هذا العمل ولكن تفيده بشكل شخصي، كأن يكون مثلاً مستخدم مسند له إدارة الحسابات في بنك وهو عميل أو زبون في هذا البنك.

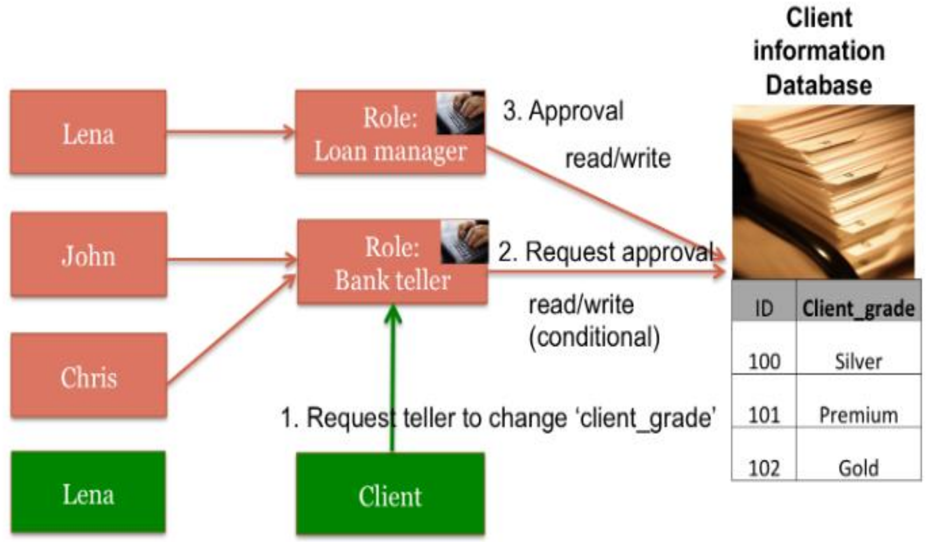
7.4.3.2– معالجة تضارب المصالح في نموذج إدارة الصلاحيات المعتمدة على الأدوار

لنفترض أن لدينا هرمية معينة في مؤسسة كما في الشكل رقم 11[8]:



رسم توضيحي 11: مثال عن هرمية مؤسسة

ولدينا في هذا البنك المستخدم Lena مُسند لها الدور loan manger وهي في نفس الوقت زبون لدى البنك. تريد Lena الرفع من مستوى درجتها كزبونة في البنك للحصول على قرض معين، فتكون الإجرائية بحاجة إلى أن تطلب Lena من أمين صندوق البنك أن يرفع لها درجتها كزبون، فيقوم أمين صندوق البنك برفع المقترح لمدير القروض والذي أساساً مسند هذا الدور إلى Lena. في النهاية مدير القروض Lena والذي هو نفسه الزبون سيوافق على مقترح أمين الصندوق.



رسم توضيحي 12: إجرائية طلب قرض في البنك

لذلك يقدم نموذج إدارة الصلاحيات حل لهذه المشاكل من خلال الفصل بين الواجبات، يدعم إدارة الصلاحيات مبدأ الفصل بين الواجبات من خلال مبدأ أقل الامتيازات least privilege، هذا المبدأ يُجنب مشكلة الفردية في قدرة بعض الأدوار على القيام أو أداء إجراءات يمكن أن تكون ضارة.

يوجد ثلاثة أنواع من مبدأ الفصل بين الواجبات هي:

- Static separation of duties[8]:

إذا كان المستخدم مسند له دور معين، عندها يحظر أن يمتلك دور آخر.

- Dynamic separation of duties[8]:

يمكن للمستخدم أن يمتلك أكثر من دور لكن لايتم تفعيل هذه الأدوار في نفس الوقت.

- History-based separation of duties[9]:

لا يمكن لشخص يمتلك دور معين الوصول إلى غرض أو مصدر عدد معين من المرات.

8.4.3.2– مدى فعالية معالجة نموذج إدارة الصلاحيات المعتمدة على الدور لتضارب المصالح

بحسب الدراسة السابقة لنموذج إدارة الصلاحيات المعتمدة على الأدوار، فهي لم تضف أي سمة تخص المستخدم كالأخذ بعين الاعتبار مكان وجوده عند أخذ قرار بامتلاكه الصلاحية أو عدم امتلاكه، بل اكتفى هذا النموذج بالقرار حسب الدور الذي يلعبه المستخدم.

وهذا سبب منطقي لأنه لو تم أخذ بعين الاعتبار هذه السمات سينتج عدد كبير من الأدوار والسمات، لذلك كان لابد من التفكير في اتباع نهج جديد قادر على معالجة سمات مختلفة للمستخدم، تكون قادرة على الحد من إمكانية المستخدمين الذين لديهم أدوار شخصية متضاربة.

5.3.2– نموذج التحكم المعتمد على المهام Task-Based Access Control Model

هذا النموذج يحل مشكلة الأمن والسرية في التطبيقات والمؤسسات التجارية، الفكرة الأساسية في هذا النموذج هي المهمة، حيث أن سماحية الوصول ليست ثابتة وإنما يمكن أن تتغير تبعاً لسياق المهمة المراد تنفيذها[10].

من المعروف أن المهمة لها وقت محدود، لذا في هذا النموذج لدى المستخدمون وقت محدد أيضاً في استخدام السماحيات المسندة وفقاً للمهمة التي يؤدوها، عند تشغيل مهمة ما يسند للمستخدم السماحيات التي يحتاجها فقط، وعندما تنتهي المهمة أو لا تكون قيد العمل لا يسند للمستخدم أي سماحية.

من الملاحظ أن هذا النموذج يخدم المؤسسات التي تعتمد على إسناد المهام للعاملين لديها لإنجاز أعمال معينة، ومن المعروف أن أي شركة قد لا تخلو من نظام الأدوار، فمن هنا كان لابد للدمج بين نموذجي RBAC و TBAC، وهذا الدمج مهم في الحياة التطبيقية، فلو أفترضنا أن شخص ما قمنا بتغيير الدور المسند له، فبالتالي السماحيات المسندة لهذا الشخص قد تغيرت أيضاً وهذا هو مبدأ الـ RBAC، ولكن في نفس الوقت نريد من هذا الشخص تنفيذ مهمة معينة ومحددة ولكنه لا يملك السماحية لهذه المهمة بالدور الجديد، لذا في هذه الحالة لابد من إدخال مفهوم الـ TBAC.

إذاً تفتقر RBAC لتنفيذ تدفق الأعمال أو تسلسلها (workflow) بينما هذا محقق في TBAC، فهي تحل مشكلة التتالي والعلاقات بين المهام الجزئية أثناء تنفيذ المهمة.

لذا تم الدمج بين هذين المفهومين بمفهوم TRBAC.

6.3.2– نموذج التحكم بالمهام المعتمد على الأدوار Task-Role-Based Access Control

الفكرة الأساسية في هذا النموذج هو دمج مفهومي الدور والمهمة معاً، حيث يتم تعيين مهمة لدور معين ومن ثم يتم إسناد السماحيات إلى المهمة وليس إلى الدور[10].

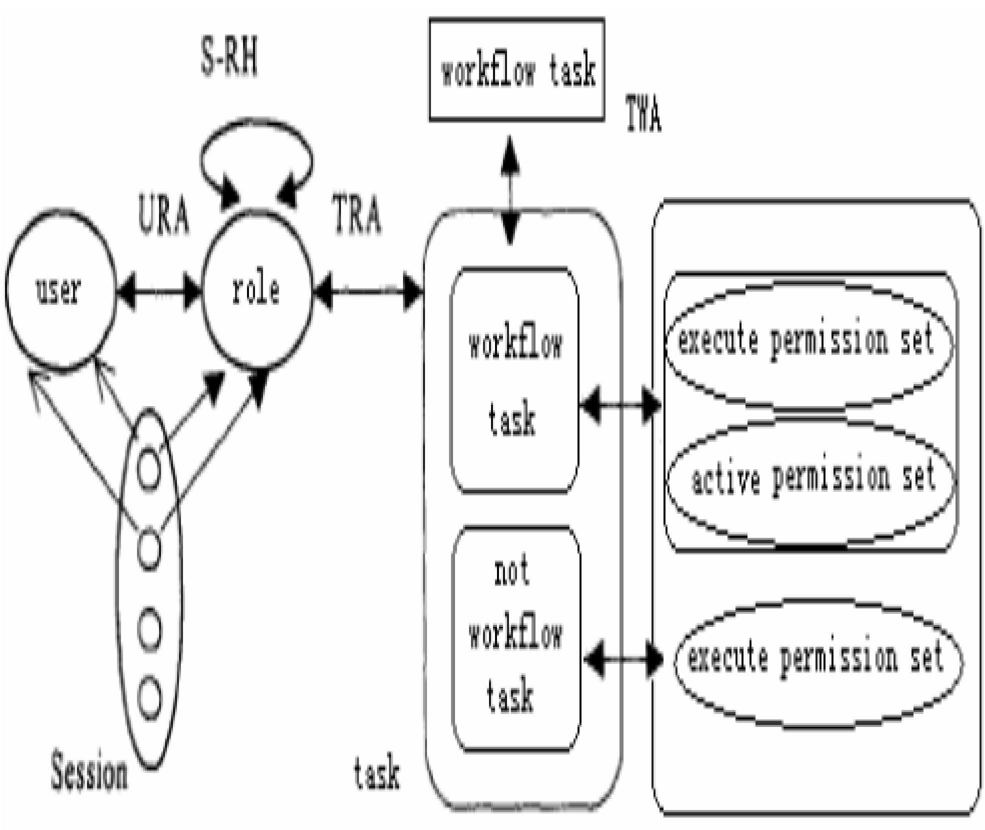
مفاهيم أساسية:

- **المهمة task**: تدفق من الأعمال، يمكن أن ترتبط بعدد من المستخدمين، وممكن أن تشمل مهام جزئية.
- **Task instance**: تعني تشغيل المهمة.

- **Authorization Step**: تعني خطوة معالجة الأُذونات، أي إجرائية المعالجة للغرض في تدفق الأعمال.
- **Dependence الاعتماد**: يعني الإرتباط بين المهام، أي مدى إعتماد تشغيل مهمة على انتهاء مهمة أخرى.

بشكل مختصر: يتم معالجة تدفق الأعمال من خلال مجموعة من المهام المترابطة والمعتمدة على بعضها البعض، وكل مهمة مسند لها عدد من الأدوار، فعندما يكون هنالك ارتباط بين المهمات، عندها الدور المسند للمهمة يقوم بالحصول على الإذن أو السماحية للبدء بتنفيذ المهمة.

يمثل الشكل رقم 13 نموذج Task-Role-Based Access Control



رسم توضيحي 13: **Task-Role-Based Access Control**

من خلال الشكل السابق نلاحظ أن المستخدم مسند للدور، والدور مسند للمهام.

يوجد مجموعة من المهام عبارة عن تدفق من الأعمال، أي مرتبطة مع بعضها البعض، فيتم العمل كما يلي: يبدأ العمل بالمهام بالتسلسل من المهمة الفرعية الأولى ويتم إسناد السماحيات للأدوار المشاركة في هذه المهمة الفرعية، وعند احتياج المهمة

الفرعية الأولى لمهمة فرعية تانية يتم إسناد السماحيات للأدوار المشاركة في المهمة الفرعية الثانية وهكذا حتى يتم الانتهاء من تنفيذ تسلسل العمل أي كامل المهمة.

بينما في المهام الغير مرتبطة بمهام أخرى، بمجرد ما بدأ تنفيذ المهمة تُسند السماحيات و الأُذونات للأدوار المشاركة في المهمة.

7.3.2– نموذج إدارة الصلاحيات المعتمد على السمات Attribute-Based Acess Control

1.7.3.2– مقدمة

كما أوضحنا سابقاً أنه لابد من إضافة سمات مختلفة للمستخدم للحد من تضارب المصالح، وبما أن نموذج إدارة الصلاحيات لا يعتمد إلا سمة واحدة للمستخدم وهي الدور التنظيمي، لذلك تم العمل على نموذج إدارة الصلاحيات المعتمد على السمات ABAC.

حسب إحصائية للعالم Gartner's [11] فإن ABAC سيحل محل RBAC بنسبة 70% في عام 2020.

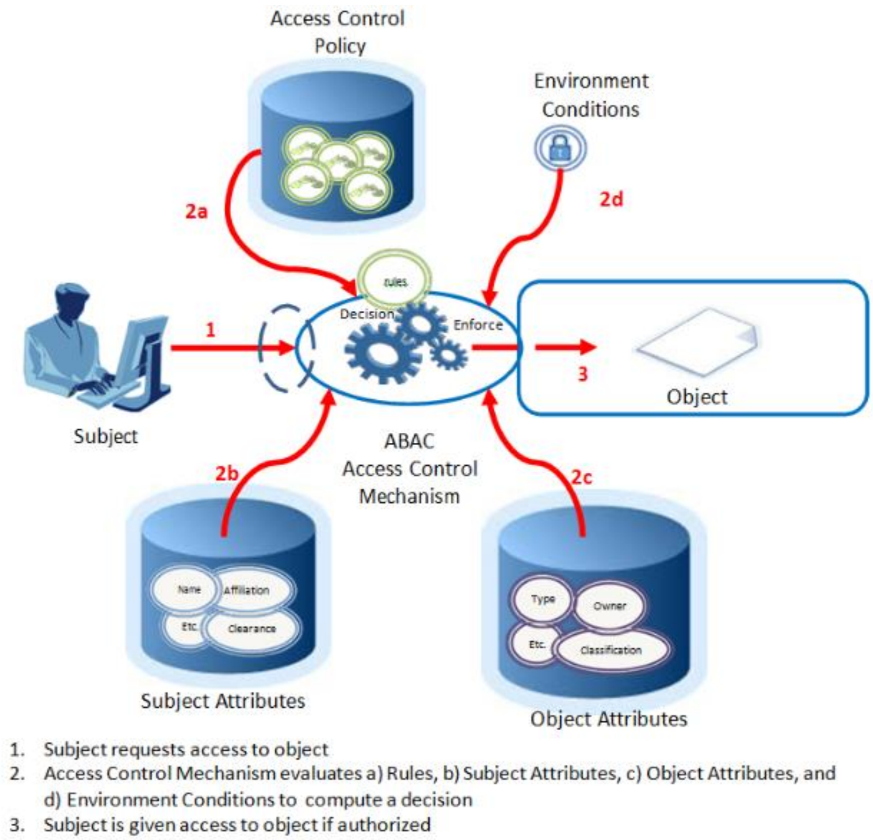
إن طريقة الوصول للقيام بعملية على غرض معين تُمنح أو تُرفض بحسب السمات المسندة لهذا الغرض، وشروط العمل، إضافة إلى مجموعة من السياسات المعرفة على السمات.

ونعني بالسمات أي خصائص الغرض أو المستخدم الذي يريد القيام بأي عملية على هذا الغرض، إضافة للشروط البيئية والمكانية.

فالسمات هي معلومات عن هذا الغرض، بينما خصائص البيئة تكون مستقلة عن الأغراض أو المستخدمين، فمن الممكن أن تشمل(الوقت الحالي، أحد أيام الأسبوع، مكان وجود المستخدم، أو مستوى التهديد الحالي الممكن حدوثه).

2.7.3.2– نموذج ABAC

يُبين الشكل رقم 14 إجرائية عمل النموذج المعتمد على السمات [12].



رسم توضيحي 14: نموذج **ABAC**

عندما يطلب المستخدم الإذن بالوصول لغرض معين، فإن سياسة هذا النموذج تقوم بالبداية بتفحص سياسة الوصول المعرفة (فمثلاً هل المستخدم من المستخدمين المعروفين لدى النظام)، ومن ثم سمات الشخص (هل هو صاحب الغرض أو لا)، ومن ثم سمات الغرض نفسه، وأخيراً الشروط البيئية كالزمان والوقت (فمثلاً هل يريد الوصول للملف في أوقات الدوام)، وبناءً على تحقق ما سبق يتم القرار بالوصول للملف أو عدم الوصول.

فعلى سبيل المثال [13] لنفترض لدينا ملف فيديو وأراد أحد المستخدمين الوصول لهذا الملف، ومن سمات هذا الملف أنه لايمكن الوصول إليه إلا من قبل المستخدمين الذين أعمارهم فوق 18 سنة، إذاً سياسة الوصول في هذا النموذج تقوم في البداية بالنظر في حال كان المستخدم يحقق شروط السمات، فإذا كان المستخدم عمره فوق 18 يتم التحقق من الشروط البيئية الأخرى وطبيعة المستخدم، فمن الشروط البيئية الأخرى مثلاً أن هذا الفيديو مخصص للعرض في بلدان معينة، ومن شروط طبيعة المستخدم مثلاً أن يكون مخصص لأشخاص ذو شهادات معينة، فعند التحقق من أن الشخص عمره فوق 18 سنة ويحمل شهادة الهندسة ومن فرنسا عندها يُعطى إذن الدخول، وإلا فإن عدم تحقق أي شرط لا يسمح له بالدخول.

كمثال آخر على شروط الوقت والمكان: يُمكن للأطباء الدخول إلى سجلات المرضى في المشفى وفي ساعات الدوام فقط.

3.7.3.2– توصيف نموذج ABAC[3]

تميزت شركة Axiomatic بمبادرتها لتنجيز ABAC حيث استخدمت لغة XACML، وهي اللغة المعبرة لسياسات التحكم ويمكن اعتبارها كلغة طبيعية.

على سبيل المثال لو فرضنا لدينا الجملة التالية:

"A user wants to do something with an information asset in a given context"

تشمل هذه الجملة أربع عناصر نحوية هي:

الفاعل، الحدث، المورد أو المصدر، والبيئة التي وقع فيها الطلب، يمكن التعبير أو وصف كل من هذه العناصر كصف سمات كما يلي:

- الفاعل:subject

هو الشخص الطالب للوصول للمعلومات، عادة ما تكون سمات الفاعل إما الأدوار التي يلعبها في المؤسسة، عضو في مجموعة، انتماء الشخص لقسم معين في المؤسسة، الكفاءة التي يملكها، رقم ذاتي ... الخ.

- الحدث:actions

هو الإجراء الذي يريد المستخدم القيام به.

سمات الأحداث تكون عادة عبارة عن طلبات القراءة والكتابة، إلا في حالة السيناريوهات المعقدة تكون هذه السمات معقدة.

فمثلاً عند الوصول لحساب في البنك يكون السمات معرفة بالشكل التالي:

"action type=transfer", and "amount=$500".

- المورد أو المصدر resource: يعرف معلومات الغرض أو الأصول المحددة بالحدث.

ففي مثال البنك الموارد يمكن أن تكون بالشكل التالي:

"debit account=<your account number>"

- البيئة environment: تعرف السياق الذي تم طلب الوصول إليه، عادة تتعلق سمات البيئة بالوقت الحالي أو المكان الذي تم فيه طلب الوصول.

---

[3] https://www.axiomatics.com

4.7.3.2– سياسة التحكم بالوصول في نموذج ABAC

ليكن لدينا المثال التالي:

"all users belonging to the sales department should have read access to proposals sent to customers within the sales region to which the user belongs."

تعني الجملة السابقة أنه جميع المستخدمين الذين ينتمون إلى قسم المبيعات ينبغي أن يملكو حق الوصول بقراءة الاقتراحات المرسلة إلى الزبائن الموجودين في المنطقة التي ينتمي إليها المستخدم.

حيث توصف لغة XACML العبارة السابقة كما يلي:

Subject's "department"="sales"

Subject's "sales region"="NW"

Action="read"

Resource "type"="proposal document"

Resource "region"="NW "

وأي شخص يحقق السمات السابقة يمكنه الوصول لقراءة الاقتراحات؟

8.3.2– نموذج التحكم بالوصول حسب الاستخدام Usage Control (UCON) Access Control

1.8.3.2– مقدمة

من خلال هذا النموذج يُمكن التحكم بوصول المستخدمين من خلال طلباتهم على الأغراض واستخدامهم لها، حيث يتم فرض صلاحيات على الوصول. من خلال النموذج السابق يُمكن تأمين مراقبة نشاط المستخدمين وهو الجزء الذي ينقص باقي نماذج الوصول السابقة. نموذج UCON نموذج مفاهيمي حتى تاريخه ولا يوجد تطبيق عملي حالي، كما أنه لا يوجد إدارة للنموذج حالياً.

يعتمد القرار بالسماح للمستخدم بالوصول للمصدر قبل وأثناء وصول المستخدم للمورد أو المصدر.

حيث أن القرار لايعتمد فقط على التراخيص authorizations للوصول إلى المصدر أو عدم الوصول، وإنما على الالتزامات والشروط obligations and conditions.

الالتزامات obligations تتطلب من المستخدم حدث معين للحصول على الصلاحية أو لا، كالضغط على موافق في اتفاقية الترخيص license agreement.

الشروط conditions : ملء فورم معين أو الإجابة على سؤال معين. ومن الشروط أيضاً أن الوصول يتم في أوقات محددة (أوقات الدوام)، أو في بيئات محددة.

2.8.3.2– اختلاف UCON عن RBAC

1– الالتزامات والشروط.

2– منح الصلاحية قبل الوصول وبعده وأثناءه.

3– *Context*-Aware البيئة والفترة

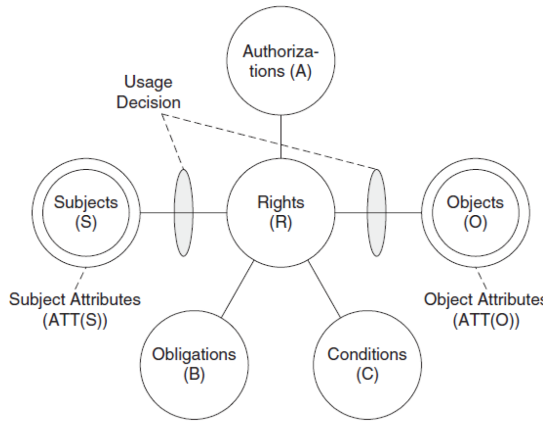
3.8.3.2– مكونات نموذج UCON ABC Model Component

يتكون النموذج من ثمان مركبات أساسية هي [16] [15]:

الفاعلين subject – واصفات الفاعل attributes of Subject – الأغراض – واصفات الأغراض attributes of Object – الحقوق rights – الصلاحيات Authorizations – الإلزاميات obligations والشروط Conditions.

الفاعلين والأغراض هي مفاهيم قد سبق شرحها في نماذج التحكم بالوصول RBAC وهي مستخدمة حالياً في هذا النموذج. الحقوق Rights يُفّعَل الوصول من قبل الفاعل Subject إلى الغرض Object بنمط خاص مثل القراءة أو الكتابة. مفهوم الحقوق rights هو تقريباً مشابه للموجود في نماذج الوصول السابقة.

يمثل الشكل رقم 15 مكونات نموذج UCON.



رسم توضيحي 15: مكونات نموذج **UCON**

يُظهِر الشكل السابق أن اتخاذ قرار الاستخدام Usage Decision يتم بناءاً على مُعاملات الفاعل ومًعاملات الأغراض والصلاحيات والشروط والإلزاميات في وقت طلب الاستخدام.

- الفاعل وواصفاته [16] **[**15]

وهو كيان (مستخدم، برنامج، خدمة ... الخ) يرتبط بواصفات ويُمارس على الأغراض نشاطات معينة (قراءة أو كتابة بصلاحية معينة. من أجل التبسيط في UCON، يمكن تخيل الفاعل كمستخدم. يعرف الفاعل بواصفات وهي تمثل قدراته وخصائصه التي تدخل في عملية اتخاذ القرار بالوصول. يمكن أن يكون للفاعل معرف وحيد. يمكن استخدام اسم مستخدم user name خلال التصريح. أمثلة عن الواصفات: مُعرّف، أسماء، مجموعة، أدوار، سماحيات أمنية ...الخ.

المجموعة: هي عدة مستخدمين لهم صلاحيات مشتركة، الدور هو سماحية معينة أو عدة سماحيات تسند إلى مستخدم أو مجموعة من المستخدمين، يمكن أن يكون للمجموعات أو الأدوار بنية هرمية.

كمثال في تطبيقات التجارة الإلكترونية، الكتاب الإلكتروني هو أحد الأغراض يمكن أن يُشترط كأحد الواصفات أن يكون سعره 10$ في حالة القراءة فقط أو 15$ في حال النسخ.

تكون بعض الواصفات غير قابلة للتغيير immutable إلا من قبل مدير يملك الصلاحية لذلك، بينما يمكن لبعض الواصفات أن تكون قابلة للتغير mutable حسب نشاط المستخدم على الأغراض.

في UCON يمكن أن يكون الفاعل كما يلي:

- مستهلك consumer subject (CS):

ويمارس المستخدم فيها حقه في استخدام الغرض (مثال: قارئ كتاب إلكتروني، مستمع إلى ملف صوتي mp3، الخ).

- مزود Provider Subject (PS):

وهو المستخدم الذي يؤمن غرض معين ويملك حقوق عليه rights (مثال: كاتب إلكتروني، موزع كتب إلكترونية، الخ).

- الأغراض ومواصفاتها **O**bject and Object Attributes [15] [16]

الأغراض هي كينونات يصل إليها أو يستخدمها الفاعلين الذين يملكون حقوق Rights معينة. للأغراض واصفات مستقلة خاصة بالأغراض أو مرتبطة بالحقوق Rights والتي يتم استخدامها في عملية اتخاذ قرار الوصول Access (أو الاستخدام).

الواصفات المرتبطة بالأغراض هي القيم، سماحيات الأدوار Role Permissions، علامات أمنية Security Labels، الصفوف Classes، الخ. صفوف الغرض Object Classes تستخدم لتصنيف الأغراض من أجل تطبيق الصلاحيات Authorization على صف معين أو أكثر حسب متطلبات النظام.

الأغراض في UCON هي إما حساسة أو غير حساسة أمنياً، وفي الأولى تحوي الأغراض مُعرّفات خاصة يمكن أن ينتج عنها مشاكل أمنية في حال استخدامها بشكل خاطئ. يمكن أن تكون الأغراض مشتقة أو أصلية، المشتقة هي ناتجة عن الوصول أو ممارسة أحد الصلاحيات على الغرض الأصيل، مثال تشغيل ملف موسيقى mp3 يقوم بإنشاء ملف تسجيل Log، هذا الأخير هو الغرض المشتق ويعامل معاملة الغرض العادي وملف ال mp3 هو الملف الأصلي.

- الحقوق **R**ights [15] [16]

الحقوق هي الامتيازات التي تسمح للفاعل Subject بالوصول إلى الغرض، ويتألف من مجموعة من التوابع تُمكن الفاعل من ذلك.

السماحيات، الإلزاميات، الشروط هي معاملات القرار والتي تدخل في توابع اتخاذ قرار الاستخدام (أو الوصول) والتي بناءً عليها يصل الفاعل إلى الغرض بطلب معين مثل read أو write.

- الصلاحيات **A**uthorization [15] [16]

من خلالها يتم تقدير فيما إذا الفاعل (الطالب request) يسمح له بتنفيذ الحق المطلوب Right على الغرض من خلال توابع تقديرية. من خلال تقييم واصفات الفاعل وواصفات الأغراض والحق المطلوب Requested Right والمقارنة مع قواعد سماحيات معينة.

السماحيات يفترض أن تكون موجودة قبل تفعيل الحق requested right(preA) وخلال الاستخدام (onA).

preA وهي التوابع التي تنفذ قبل اتخاذ قرار الاستخدام، onA وهي التوابع التي يتم تنفيذها خلال عملية الاستخدام وذلك بشكل مستمر أو بشكل متقطع في فترات زمنية محددة.

بعض الصلاحيات تتطلب تعديل واصفات الأغراض Objects أو الفاعلين Subjects قبل وخلال وبعد الاستخدام، مثال على ذلك بعض الخدمات المأجورة تتطلب تعديل قيمة الحساب الائتماني لضمان عدم الاستخدام الزائد عن القيمة المرصودة لذلك.

- الإلزاميات **o**bligation's **[**16**]** [15]

وهي توابع تقوم بالتحقق من متطلبات يجب على المستخدم ملؤها قبل وخلال الاستخدام. الإلزاميات يمكن أن تكون قبل الاستخدام (preB) أو خلال الاستخدام (onB). preB وهي التوابع التي تتأكد من ملئ المستخدم لبعض المعلومات المطلوبة ويعيد قيمة true أو false قبل اتخاذ قرار الوصول واستخدام الغرض، كمثال يمكن أن يُطلب من المستخدم الموافقة على تسجيل نشاطاته قبل الاستخدام أو الموافقة على بعض الاتفاقات. onB وهي توابع تؤمن استمرار استيفاء المستخدم لبعض الشروط خلال الاستخدام بشكل مستمر أو متقطع خلال فترات زمنية معينة، مثال قد يفرض على المستخدم مشاهدة إعلان معين خلال الاستخدام.

يمكن أن تستخدم توابع الإلزاميات واصفات كل من الفاعل والغرض لتحديد بعض الشروط الإجبارية، ويمكن تعديل هذه الواصفات والتي يمكن أن تؤثر على قرارات الاستخدام المستقبلية.

الجدير بالملاحظة: أنه تُستخدم الواصفات من أجل تحديد الشروط الإجبارية، في حالة pre نقوم بتحديد الموافقة أو عدمها على طلب الوصول أما في حالة on فهو يحدد الاستمرار أو وقف الاستخدام.

- الشروط **C**onditions (C) **[**16**]** [15]

وهي التوابع التي تقوم بتقدير البيئة الموجودة والنظام ومقارنتها مع واصفات الفاعل أو الغرض ويعيد قيمة true في حال التوافق أو false في حال عدم التوافق. وعلى عكس ما سبق لا تقوم هذه التوابع بتعديل قيم واصفات الفاعل أو الغرض.

بعض الشروط الواجب مراعاتها هي الوقت المحلي، المكان، الحالة الأمنية لنظام التشغيل (normal، high alert ، under attack…).

تختلف الشروط عن الإلزاميات والصلاحيات حيث أنه في الشروط يتم التركيز على شروط البيئة والنظام ولا ترتبط بشكل مباشر بواصفات الفاعل أو الغرض على عكس الصلاحيات والإلزاميات التي تقوم بتقدير قيم الواصفات للفاعل الطالب أو الغرض المطلوب وتعديلها في بعض الحالات.

4.2– نماذج التحكم بالوصول الخاصة:

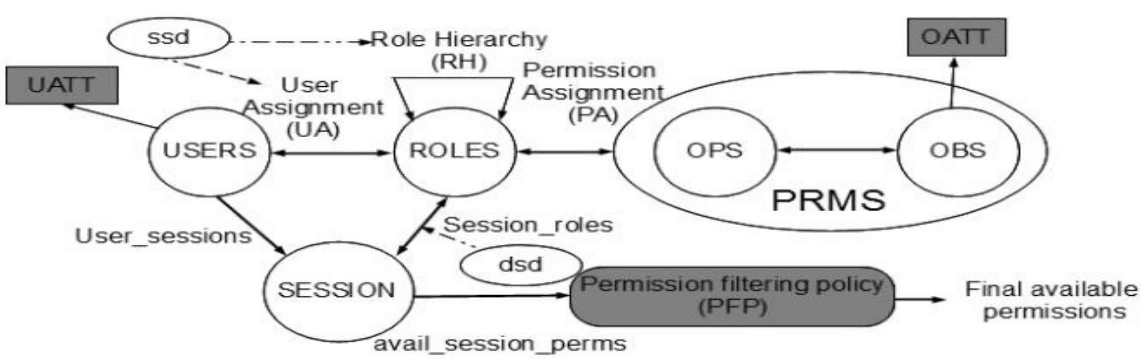
يعتبر Sandhu من الأوائل في تطوير نظام RBAC، فقد اقترح [17] نموذج RBAC96 يقارن من خلاله بين أربعة نماذج لنظام RBAC هي.RBAC0, RBAC1, RBAC2, RBAC3 :

وقد اعتبرت هذه النماذج هي أساس نموذج RBAC والتي أسست للتركيز على مفهوم الدور، ومن وقتها تم التطوير على هذه النماذج بحيث تحل مشاكل معينة.

مع ذلك مازالت نماذج RBAC غير قادرة على حل جميع أنظمة التحكم في الوصول، مما استدعى تدخل NIST لاطلاق مبادرة إضافة السمات لنموذج RBAC [18].

فقد اقترح Kuhn وأخرون [19] أن الدمج بين ميزات RBAC و ABAC ستعطي ميزات إضافية وقوة في أنظمة التحكم في الوصول، فقد قامو بالمقارنة بين ميزات ومساوئ النموذجين واقتراح نموذج جديد يقوم على إضافة سمات للمستخدم، وهو فكرة جيدة لحل مشاكل معينة ولكنها لا تحل جميع المشاكل، ففي حالة المشكلة التي نواجها أو مشكلة أي مؤسسة تعاني من عدد كبير من المستخدمين والموارد سيؤدي إضافة عدد كبير من السمات والسياسات، إضافة إلى الأدوار المعرفة في المؤسسة، مما يؤدي لتفاقم المشكلة وليس حلها.

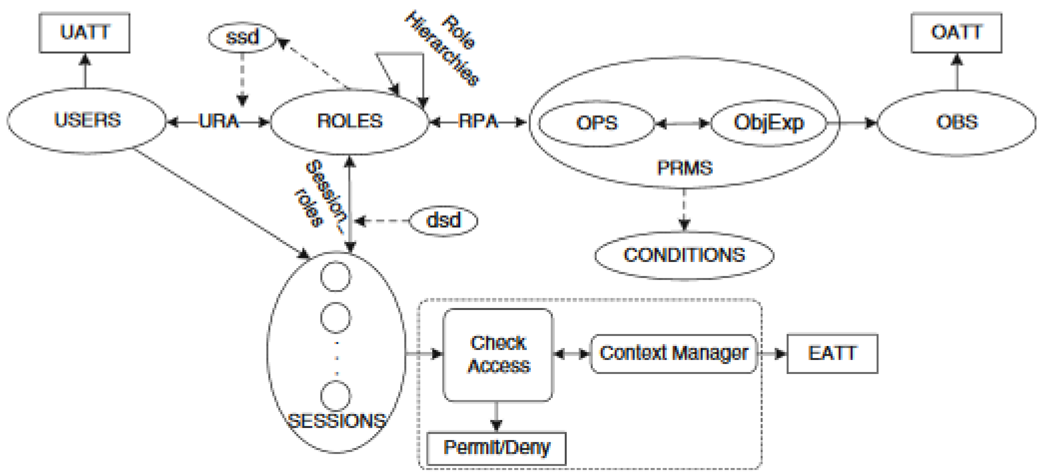
قدم Jin وأخرون [20] نموذج أخر RABAC تم فيه الدمج بين نموذج RBAC ونموذج ABAC، من خلال توصيف سمات للأشخاص والأغراض، أي تم تدارك بعض النقص في نموذج Kuhn ورفاقه من خلال توصيف سمة للإغراض، وبذلك يكون هذا النموذج قد حد من مشكلة role-explosion، يمثل الشكل رقم 16 نموذج RABAC، حيث نلاحظ من الشكل أنه تم إضافة سمات للفاعلين والأغراض، وتتم فلترة السماحيات من خلال session وبهذا يضمن النموذج عدم حصول الفاعل على سماحيات متضاربة.



رسم توضيحي 16: نموذج **RABAC**

قدم Qasim و أخرون [21] نموذج أخر AERBAC تم فيه الدمج بين RBAC و ABAC من خلال إضافة سمة جديدة خاصة بالبيئة (الزمان والمكان)، إضافة إلى سمة الأشخاص والأغراض، سمح هذا النموذج من خلال السمات الثلاث الأخذ بعين الاعتبار سياق الطلب (Request Context) وهي إضافة تميزه عن جميع النماذج السابقة، وقد قدم أيضاً مجموعة من الأليات أو السياسات (Mechanisms) لتقييم الطلبات ليتم استخدامها في تطبيقات مختلفة.

يمثل الشكل رقم 17 نموذج AERBAC:



رسم توضيحي 17: نموذج **AERBAC**

من الشكل 17 نلاحظ أن الباحثين قدموا إضافة جديدة على النماذج السابقة وهي سمات البيئة، الزمانية والمكانية، كما قدموا خاصية الشروط على السماحيات، حيث تم ربط كل سماحية مع شرط أو أكثر، حيث يجب أن تكون هذه الشروط محققة لكي يسمح للفاعل بتنفيذ هذه السماحية.

من الممكن أن تحوي هذه الشروط على جميع السمات، بما فيها سمات الفاعلين، الأغراض، وسمات البيئة والمكان.

ففي بعض الحالات لابد من مقارنة سمات الفاعلين والأغراض، كمثال في أحد المصارف، يسمح لمدير فرع بالوصول إلى الحسابات التي تنتمي إلى فرعه الخاص. فالنموذج AERBAC يسمح لأداء مثل هذه المقارنات باستخدام الشروط، إضافة أخرى لهذا النموذج من خلال Context Manager والذي يلعب دوراً مهماً في التحديث الألي للسمات، منعاً من إرتكاب الأخطاء في قرار الوصول للأغراض.

هدفت المساهمات الأخيرة للاستغناء عن نظام RBAC واستخدام ABAC كحل بديل[22]، فقد حظي التحكم في الوصول المعتمد على السمات ABAC باهتمام كبير في السنوات الأخيرة، على الرغم من أن هذا النموذج وجد من حوالي عقدين من الزمن، العديد من نماذج ABAC قد تم تعريفها، مع وجود اختلافات بين هذه النماذج، فقد كانت البدايات

بتعريف نموذج ABAC الأساسي ومن ثم تعريف نماذج تعتمد على نموذج ABAC الأساسي مع ميزات إضافية، كتعريف سمات لمجموعة من الأشخاص أو الأغراض، وتعريف هرمية لهذه السمات ...الخ.

فالعلاقة بين المجموعات والسمات عزز من سياسة التحكم في الوصول، وسهل من عملية إدارة هذه السياسة.

ومع ذلك إلى هذا الوقت الحالي تطبيق مثل هذه النماذج في العالم الحقيقي يكاد يكون معدوماً.

فقد قدم Palak [23] دراسة نظرية عن المساوئ والميزات التي ستنتج عن الاستغناء عن نموذج RBAC بنموذج ABAC، فقد وضح أن القيام بالاستغناء حل مناسب يمكن تطبيقه في حال المؤسسات التي لا تعتمد الهرمية في قرار الوصول، فهذا الحل يلغي مشكلة role-explosion، لكنه يسبب عبء على المؤسسة في حال عدد المستخدمين الكبير والموارد الكبيرة.

قامت أغلب الأبحاث الأخيرة على تحسين نموذج ABAC من خلال الحد من العمليات الكثيرة لتعريف السمات والسياسة المتبعة وتخفيف العبء الإداري على المؤسسات، فقد قام P. Mell [24] باتباع سياسة جديدة في تنجيز نموذج ABAC من خلال إسناد السمات للمجموعات بدلاً من الأشخاص أو الأغراض.

يقوم النموذج على تعريف مجموعة من المستخدمين يتشاركون بسمات مشتركة، وتعريف مجموعة من الأغراض يتشاركون بسمات مشتركة، بهذه العملية يكون النموذج قد خفف من العبء الإداري للمؤسسة، لكن هذا غير محقق دائماً كونه ليس من الممكن تواجد مجموعة من المستخدمين أو الأغراض تتشارك بنفس السمات، إلا في الحالات الخاصة.

قدم Servos وأخرون [25]، نموذج HGABAC عرفوا من خلاله السمات في مجموعات ووضع هرمية لهذه السمات، ما ميز هذا النموذج أنه سهل المهام الإدارية من خلال جمع السمات في مجموعات وإسنادها المجموعات للمستخدمين بدل من إسناد كل سمة لوحدها، إضافة إلى تعريف الهرمية للسمات فعند إسناد سمة ما لمستخدم فإنه يرث جميع السمات التي تحتها في الهرمية، وهذا وفر أيضاً من عمليات الإسناد.

تكمن المشكلة في هذا النموذج أن السمات يجب أن تكون معرفة مسبقاً، وذلك لتعريف المجموعات والهرميات، إضافة إلى حصول تضارب في حال وجود سمة ضمن هرمية معينة ولا نريد إسناد السمات التي تحتها.

5.2– مناقشة الدراسة المرجعية

من الدراسة السابقة نلاحظ العدد الكبير من نماذج التحكم في الوصول، في الحقيقة إذا أردنا التحدث عن جميع أنظمة التحكم في الوصول الحالية فالعدد أكبر بكثير، ولكن أقتصرت الدراسة المرجعية على هذه النماذج المختارة لأنها تسعى لحل مشاكل قريبة من مشكلة البحث المطروحة.

فهدف الدراسة المرجعية هو التعرف على أنظمة التحكم في الوصول، واختيار المناسب منها والتعديل عليه لكي يحل مشكلة البحث المطروحة.

لذا سنناقش في هذه الفقرة مدى قدرة النماذج المدروسة على حل المشكلة أو التعديل عليها، حيث يوضح الجدول رقم 4 المقارنة بين هذه النماذج.

جدول 3: مقارنة بين نماذج التحكم في الوصول

اسم النموذج	ما يحققه النموذج
نموذج المعتمد على الأدوار	يمنح كل رئيس مجموعة الوصول لملفات الأشخاص التابعين له إدارياً ولا يحقق المطلوب.
نموذج المعتمد على السمات	يحقق المطلوب لكن في حال عدد كبير من الملفات ومشاركة عدد كبير من الأشخاص من مجموعات مختلفة يؤدي إلى صعوبة في الإدارة وعدد كبير من السياسات والقيود وخاصة في حال انتقال شخص من مكان لأخر.
نموذج المهام المعتمد على الأدوار	يعتمد على فكرة منح السماحيات عند تفعيل المهام، وتلغى السماحيات عند انتهاء المهمة، فهو لا يحقق المطلوب في حل مشكلة البحث.
نموذج التحكم حسب الاستخدام	مازال هذا النموذج قيد التطوير، فلا يوجد له نموذج معترف عليه عالمياً، إضافة لعدم وجود نموذج إداري له.
نموذج RABAC	لا يحقق الهدف المراد لأن السمة تسند للشخص وليس للدور فلا يرث من أعلى منه بالهرمية السماحيات المعرفة بهذه السمات.
نموذج AERBAC	لا يحقق الهدف المراد لأن السمة تسند للأشخاص وللزمان وليس أيضاً للأدوار.
نموذج HGABAC	عرف هذا النموذج الهرمية من خلال دمج السمات في مجموعات، لكنه لم يأخذ بعين الاعتبار الأدوار والهرمية بين الأدوار، وبالتالي صعوبة في الإدارة وتعريف عدد كبير من السياسات والقيود وخاصة في حال انتقال شخص أو تغيير في وضعه الوظيفي في المؤسسة.

نلاحظ من الجدول السابق أن نموذج ABAC يحقق المطلوب، لكن لم نأخذ بعين الاعتبار في حال الهرميات الكبيرة وعملية الوراثة بين مستويات الهرمية العليا من المستويات الدنيا، إضافة إلى حالة وجود عدد كبير من الموارد، فهذا سيشكل عبء

على المؤسسة في إدارة التحكم في الوصول للموارد، وخاصة في حال انتقال الشخص من مستوى إلى مستوى أخر. وفي نفس السياق تعتمد كل من النماذج RABAC، AERBAC، HGABAC على نموذج ABAC الأساسي التي من الممكن أن تحل المشكلة ولكنها ستسبب عبء كبير على المؤسسة، وهي موجهة لحالات خاصة لا تشمل الحالة التي نقوم بدراستها، حيث سنبين في الفصل الأخير من الأطروحة مدى العبء وعدد العمليات التي ممكن أن تحتاجها هذه النماذج لحل هذه المشكلة.

الفصل الثالث

النموذج المقترح

سنقوم في هذا الفصل بعرض النموذج المقترح لتحقيق عملية الوصول للموارد في المؤسسات ذات الهرمية الهجينة، ونستعرض في هذا الفصل الأدوات والطريقة المتبعة لتنجيز الحل.

1.3– مقدمة

من خلال النماذج التي تم استعراضها في الفصل الثاني، يتبين مدى أهمية وجود نظام تحكم في الوصول للبيانات، يسمح للمستخدمين بالوصول للبيانات المخولة لهم الوصول إليها.

من خلال مناقشة الدراسة المرجعية واعتماداً على مشكلة البحث نلاحظ أن اعتماد نموذج ABAC أو النماذج التي اعتمدت على ABAC مع إضافة إضافة تحسينات على هذا النموذج، يزيد من تعقيد حل المشكلة، لأن مشكلة البحث تخص المؤسسات ذات الهرميات الشجرية، فاعتماد نموذج ABAC سيزيد من عدد العمليات والسياسات المعرفة في المؤسسة، إضافة إلى العبء الإداري الناتج عن إجراء تغيير بسيط في هرمية الأشخاص في المؤسسة والذي سيؤدي إلى تغيير أغلب السياسات التي تم تعريفها.

لذا كان علينا الاتجاه نحو تطبيق نموذج RBAC وإضافة تحسينات على هذا النموذج بحيث نكون نموذج جديد قادر على حل مشكلة البحث بأقل التكاليف والعمليات.

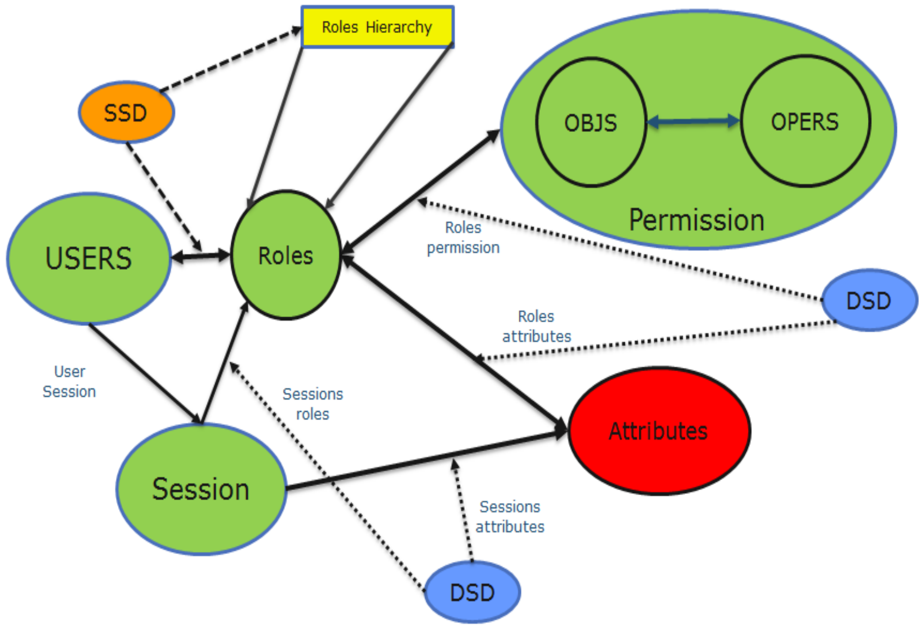
المشكلة الأساسية في بحثنا هي السماح لأشخاص أو أدوار ذات خاصية معينة بالوصول لأغراض يملكون صلاحية الوصول إليها(بحسب سياسة المؤسسة) ولا يمكن لنموذج RBAC أن يمنحهم هذه السماحية.

لذا سنعرف في هذا الفصل نموذج AARBAC: Attribute-Augmented Role Based Access Control والذي يعتمد على الدمج بين المزايا التي يقدمها RBAC من حيث تسهيل الإدارة وتحليل السياسات واستعراض المستخدمين والسماحيات ومزايا ABAC من حيث الاعتماد على السمات والمرونة.

2.3– النموذج المقترح AARBAC

1.2.3– الإطار المفاهيمي للنموذج المقترح

يمثل الشكل رقم 18 نموذج AARBAC



رسم توضيحي 18: نموذج **AARBAC**

في النموذج المقترح نحافظ على جميع مكونات النموذج RBAC، المستخدمين والأدوار والأغراض .. إلخ.

وقمنا بإسناد سمة للدور، ونقصد بالسمة هنا أي صفة ما للدور، كمثال يملك الدور المسمى كرئيس فريق عمل لمجموعة في شركة سمة رئيس فريق عمل لمشروع ما، فقد جاءت فكرة إسناد السمة للأدوار من إشكالية البحث فنحن في إشكالية بحثنا نسعى إلى السماح لأدوار تملك خاصية أو سمة معينة بالوصول للأغراض التي تملك صلاحية وصول إليها ولا يقدمها نموذج RBAC، حيث إسناد السمة للدور بدلاً من إسنادها للاشخاص (بعكس النماذج السابقة) يسمح للأدوار الأعلى في الهرمية من الوصول أيضاً للاغراض السابقة.

2.2.3– مثال يشرح الحل المقترح

لنفترض لدينا الهرمية المعرفة بالشكل التوضيحي رقم 1

يتبع الموظفون الذين يعملون على إنجاز المشروع رقم 1 إدارياً لرئيس المجموعة 1، والموظفون الذين يعملون على إنجاز المشروع رقم 2 يتبعون إدارياً لرئيس المجموعة 3.

يشارك بعض العناصر الذين يتبعون لرئيس المجموعة 3 في إنجاز المشروع 1، وجميع هذه العناصر تقوم بإنجاز مهام معينة وحفظها في مجلد خاص بالمشروع رقم 1.

تقوم سياسة المؤسسة على قدرة رئيس فريق العمل أو مدير المشروع (أي سمة ما) على الوصول لجميع ملفات المشروع، المنجزة من قبل جميع عناصر المشروع.

يمنح نموذج RBAC رئيس فريق العمل صلاحية الوصول لملفات المشروع المنجزة من قبل الأشخاص التابعين له إدارياً فقط وفقاً للدور المسند له، لذا قمنا بإسناد سمات للأدوار كسمة رئيس فريق عمل.

أي أنه في هذا المثال فإن رئيس فريق العمل (رئيس المجموعة 1) لديه صلاحية الوصول إلى جميع ملفات المخزنة من قبل الموظفين الذين يعملون في المشروع 1 سواء كانوا يتبعون له إدارياً أو مشاركين في إنجاز المشروع.

وبنفس الوقت تقوم سياسة المؤسسة على أن الأدوار الأعلى في الهرمية ترث ما يحصل عليه من هم أدنى بالهرمية، ففي مثالنا فإن project1, manager سيملكون صلاحية الوصول لجميع ملفات المشروع.

ما يقدمه نموذج RBAC هو قدرة رئيس المجموعة 1 ومن هم أعلى منه بالهرمية على الوصول لملفات المشروع 1 الخاصة بالموظفين التابعين إدارياً فقط لرئيس المجموعة 1 ولا يمكنهم الوصول للملفات الخاصة بالموظفين التابعين إدارياً لرئيس المجموعة 3 كونهم ليسوا أعلى مستوى بالهرمية.

نلاحظ هنا أن النموذج الذي نقترحه، يسند السمة للدور وليس للشخص بعكس النماذج السابقة والمعرفة بالدراسة المرجعية، ففي حالة هذا المثال لو أسندت سمة رئيس فريق العمل للمشروع 1 إلى الشخص الذي يملك الدور رئيس المجموعة 1، عندها سوف يحصل هو فقط على هذه السمة ولا يرثه من هم أعلى منه بالهرمية وهذا يناقض السياسة التي تقوم عليها المؤسسة والمعرفة في الفصل الأول، إضافة إلى أنه في حال تغير الشخص الذي أسندت له السمة لسبب ما، فإنه يجب تعريف سياسة جديدة يتم من خلالها إسناد هذه السمة لشخص آخر، أما في حالة النموذج الذي نقترحه فإنه في حال تغير الشخص أو انتقاله فلن يؤثر أي شيء فقط نسند الدور إلى الشخص الجديد، فيملك مباشرة هذه السمة من خلال الدور أي توفير عدد من العمليات التي يجب أن تتم في حال تبدل الأشخاص أو انتقالهم لمستويات أعلى في الهرمية.

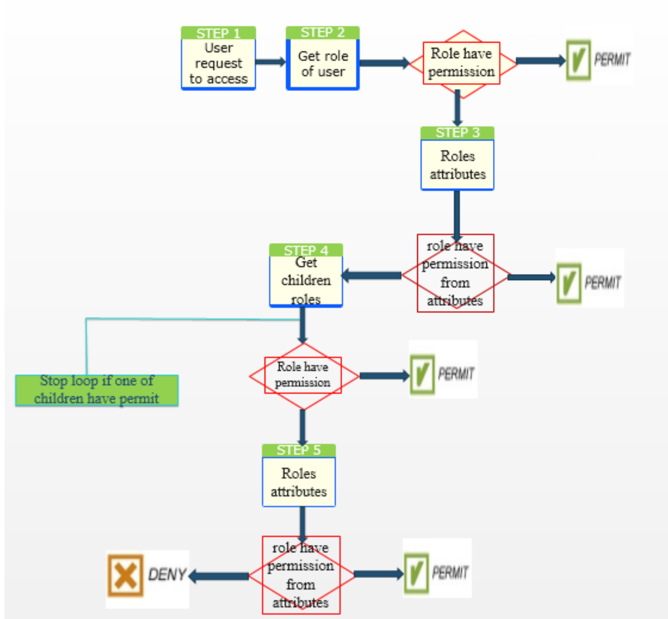
3.2.3– المخطط الصندوقي للنموذج والخوارزمية المتبعة:

يمثل الشكل رقم 19 المخطط الصندوقي للنموذج، حيث تتمثل الخطوة الأولى بطلب الوصول لغرض معين نرمز له ب o للقيام بعملية معينة على هذا الغرض(حذف، تعديل...إلخ) نرمز لها ب op، في البداية نتأكد إذا كان المستخدم الحالي u يملك دور معين في النظام، فإذا لم يملك أي دور فلا يمكنه الوصول لأي غرض أو ملف في النظام، أما إذا امتلك دور ما نتأكد إذا كان هذا الدور يملك سماحية بالقيام بالعملية op على الغرض o فإذا كان ذلك محققاً يسمح له بالقيام بالعملية.

أما إذا لم يملك الدور هذه السماحية نتأكد فيما إذا كان هذا الدور يملك سمة ما تمنحه سماحية الوصول، فإذا امتلك هذه السمة عندها نمنحه الوصول، إما إذا لم يملك ذلك، نبحث عن أدوار من هم أدنى منه في الهرمية.

من أجل كل دور من الأدوار التي تقع تحت الدور الأساسي في الهرمية تقوم بما يلي:

نتأكد إذا كان هذا الدور يملك السماحية للقيام بالعملية op على الغرض o فإذا امتلك ذلك تمنح صلاحية الوصول للدور الأساسي وإذا لم يملك ذلك نتأكد فيما إذا كان الدور يملك سمة ما تمنحه سماحية الوصول، فإذا امتلك هذا الدور هذه السمة يمنح الدور الأساسي صلاحية الوصول وإذا لم يمتلك نكرر العملية على باقي الأدوار فإذا لم يحقق أي دور الصلاحية عندها لا يمنح الدور الأساسي سماحية الوصول.وبمجرد الحصول على دور واحد من الأدوار الأدنى في الهرمية يحقق المطلوب يمنح الدور الاساسي سماحية الوصول وتتوقف العملية عند باقي الأدوار.



رسم توضيحي 19: نموذج المخطط الصندوقي للنموذج

4.2.3– التوصيف الرياضي لنموذج AARBAC

في الحقيقة في نموذج AARBAC قمنا كما ذكرنا سابقاً بالمحافظة على نموذج RBAC وتمديده بإضافة السمات للأدوار فقط، لذا توصيف نموذج AARBAC سيكون مشابهاً لتوصيف نموذج RBAC مع بعض الإضافات.

من أجل الفاعل (المستخدم) Subject، الدور النشط هو الدور الذي يستخدمه المستخدم حالياً[3]:

AR(s:subject) = {the active role for subject s}

يمكن أن يُسند أكثر من دور لمستخدم معين:

RA(s:subject) = {authorized roles for subject s}

من أجل كل دور، السمة المفعلة للدور هي السمة التي يستخدمها الدور حالياً:

AAt(r : role) = {the active attribute for role r}

يمكن أن يسند أكثر من سمة لدور معين:

AtR(r:role) = {authorized attributes for role r}

كل دور يسمح بإجراء معين أو عدة إجرائيات:

TA(r:role) = {transactions authorized for role r}

يستطيع المستخدم أن يُفعِل إجرائية ما ونرمز لها بالتابع exec(s,t) والذي يأخذ قيمة true إذا استطاع الفاعل تنفيذ الإجرائية.

$$exec(s{:}\, subject, t{:}\, transaction) = true\ iff\ s\ can\ execute\ transaction\ t$$

حيث يستطيع المستخدم تنفيذ إجرائية ما بعد التأكد من القواعد التالية:

5- إسناد الأدوار

يمكن للمستخدم أن يقوم بتفعيل إجرائية إذا وفقط إذا تم إسناد أدوار إليه .

$$\forall s{:}\, subject, t{:}\, transaction, (exec(s,t) \Rightarrow AR(s) \neq \emptyset) \quad \ldots\ldots\ldots\ldots\ldots (1)$$

عملية تسجيل الدخول لا تعتبر إجرائية transaction، أما غيرها من نشاطات المستخدمين داخل النظام تعتبر إجرائيات تحتاج إلى سماحيات من خلال الأدوار.

6- ترخيص الدور

الدور النشط الحالي للمستخدم يجب أن يكون مسند له كما يلي:

$$\forall s{:}\, subject, \left(AR(s) \subseteq RA(s)\right) \quad \ldots\ldots\ldots (2)$$

تضمن هذه القاعدة مع القاعدة الأولى، عدم تمكن المستخدمين من الحصول على أدوار غير مرخص لهم بها.

7- إسناد السمات للأدوار

يمكن للدور أن يقوم بتفعيل إجرائية حصل عليها من سمة إذا وفقط إذا تم إسناد سمات إليه .

$$\forall r{:}\, role, t{:}\, transaction, (exec(r,t) \Rightarrow AAt(r) \neq \emptyset) \quad \ldots\ldots\ldots (3)$$

8- ترخيص السمات

السمة النشطة للدور يجب أن تكون مسندة له كما يلي:

$$\forall r{:}\, role, \left(AAt(s) \subseteq AtR(s)\right) \quad \ldots\ldots (4)$$

تضمن القاعدة الثالثة مع القاعدة الرابعة، عدم تمكن الأدوار من الحصول على سمات غير مرخص لهم بها.

من القواعد السابقة يمكن تعريف ترخيص الإجرائية التي يقوم بها المستخدم للسماح له بالوصول للأغراض أو عدم السماح له بذلك.

$$\forall s{:}\,subject, \exists \mathrm{r{:}\,role}, \left(AR(s) \subseteq RA(s)\right),$$
$$t{:}\,tran, o{:}\,object,$$
$$\left(exec(s,t) \Rightarrow access(AR(s),s,t,x)\right) \cup \left(exec(s,t) \Rightarrow access(\mathrm{AAt}(r),r,t,x)\right)$$

يشير الـ access(AR(s),s,t,x) إلى سماحية وصول المستخدم s إلى الغرض o من خلال المعاملة t (والتي يحصل عليها من خلال الدور النشط) في النمط x والذي يأخذ إحدى القيم (read,write,append).

يشير الـ $access(\mathrm{AAt}(r),r,t,x)$ إلى سماحية وصول الدور (عملياً هو المستخدم) إلى الغرض o من خلال المعاملة t (والتي يحصل عليها من خلال السمة النشطة) في النمط x والذي يأخذ إحدى القيم (read,write,append).

3.3– التنجيز

1.3.3– مقدمة

تم تصميم النموذج لتطبيقه في بيئة العمل لحل مشكلة حقيقية نواجهها في مجال عملنا، لذا تم تنجيز النموذج كنموذج برمجي يدل على إمكانية تحقق هذا النموذج في بيئة عمل حقيقية، وتحويله لنموذج رياضي عبارة عن علاقات ومجموعات، لذلك تم إثبات صحة النموذج المقترح من خلال تطبيق عملي، عبارة عن موقع الكتروني يتم من خلاله تعريف هرمية معينة، ومن ثم تعريف الأدوار والمستخدمين والسماحيات وسمات الأدوار.

تم اعتماد بيئة yii framework لما تقدمه من تسهيلات برمجية ومكتبات تساعد على سرعة وسهولة في تطوير المواقع، حيث سنقدم شرحاً مختصراً في البداية نعرف من خلالها أهمية هذه البيئة.

2.3.3– تعريف

إن Yii هي اختصار لعبارة "yes it is" وهي بيئة عمل مفتوح المصدر تستخدم لغة البرمجة PHP لتطوير تطبيقات ويب ديناميكية ذات كفاءة عالية وبنفس الوقت بأقل مجهود برمجي ووقت ممكن عن طريق استخدام المكتبات والنماذج الجاهزة المضمنة ضمن إطار العمل.

تتضمن تلك المكتبات العديد من الإجراءات الجاهزة للتعامل مع كافة أجزاء المشروع مثل عمليات الاتصال والتعامل مع قواعد البيانات، وتحويلات الروابط، ومكتبات خاصة بتطبيقات جافا سكربت وغيرها.

مما يضمن لنا إنجاز العديد من المهام المعقدة في وقت قصير وبدون جهد.

3.3.3– مزايا العمل باستخدام Yii

1- توفير الوقت للمبرمج أثناء قيامه بتطوير أي تطبيق.

2- سهولة ومرونة في كتابة الكود أثناء البرمجة نظراً لتوفر المكتبات والإجراءات الجاهزة.

3- تنظيم ملفات ومجلدات، وأكواد المواقع بشكل رائع مما يسهل علينا متابعة المشروع عند إعادة العمل فيه بعد وقت.

4- تعتمد على أساليب البرمجة غرضية التوجه Object-oriented programming - OOP في العمل.

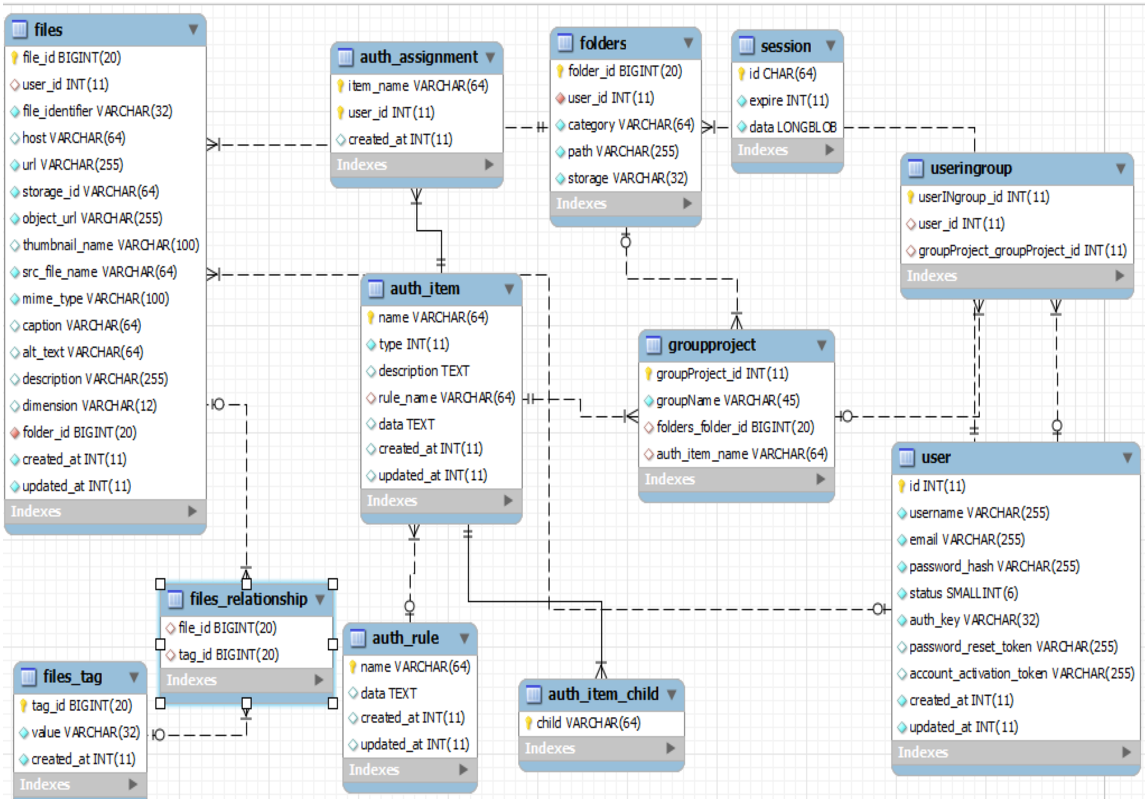
4.3.3– كيفية عمل yii

إن Yii مبني على مفهوم ال MVC (Models-views-Controllers) ولكل جزء وظيفة.

- Models: هي الطبقة المسؤولة عن التعامل مع قواعد البيانات.
- Views: هي الطبقة التي تمثل واجهة التطبيق وهو الجزء المرئي من التطبيق والذي يظهر للمستخدم ويتضمن في العادة أكواد التصميم.
- Controllers: تمثل هذه الطبقة حلقة الوصل التي تربط ما بين طبقة البيانات models وطبقة الواجهة Views.

5.3.3– قواعد المعطيات:

يمثل الشكل رقم 20 مخطط قواعد المعطيات للنموذج المقترح.



رسم توضيحي 20: مخطط قواعد المعطيات للنموذج المقترح

في هذا النموذج هنالك أربع جداول رئيسية تمثل نموذج RBAC.

الجدول الأول هو auth-item يخزن معلومات الأدوار والأحداث المطبقة على الأغراض، يحوي هذا الجدول الحقول التالية:

Name: هو اسم الغرض والذي يشمل حدث معين مثلاً إنشاء ملف أو تعديله ويأخذ الشكل التالي files/Create، أو يكون اسم دور معين مثلاً رئيس قسم أو مدير مالي.

Type: ويعرف هذا الحقل بعدد حيث في حالتنا يأخذ هذا الحقل القيمة 1 إذاكان دور والقيمة 2 إذا كان حدث معين وهدف هذا الحقل هو التمييز بين النوعين السابقين.

Description: يمثل وصف للدور أو الحدث.

Rule-name: يمثل اسم القاعدة المطبقة على هذ الحدث أو الدور.

Create-at: تمثل تاريخ إنشاء دور أو الحدث.

Update-at: تمثل تاريخ تعديل الدور أو الحدث.

أما الجدول الثاني فهو جدول user وهو جدول المستخدمين، يحول معلومات عن المستخدمين من الاسم، البريد الالكتروني وكلمة السر.

الجدول الثالث هو auth-assignment: يربط هذا الجدول المستخدمين مع أدوارهم، ويحوي الحقول التالية:

Item-name: يمثل اسم الدور.

User-id: يمثل رقم المستخدم المسند له هذا الدور.

Created-at: تاريخ إنشاء الإسناد بين المستخدم والدور.

الجدول الرابع: auth-item-child:

يمثل هذا الجدول العلاقة بين الآباء والأبناء ويحوي:

Parent: يمثل الأب ويمكن أن يحوي دور أو سماحية أو حدث معين.

Child: يمثل الابن ويمكن أن يحوي دور أو سماحية أو حدث معين أيضاً.

فمثلاً يمكن أن يكون parent هو عبارة عن مدير عام (ويمثل دور) و child عبارة عن مدير مالي (دور)، أي أن المدير العام هو أب للمدير المالي، بمعنى آخر المدير العام في مستوى أعلى من الهرمية من المدير المالي.

مثال آخر:

ممكن أن يكون parent عبارة عن دور كمدير مالي و child يكون عبارة عن حدث معين كحذف ملف، أي أن الدور(مدير مالي) يملك سماحية حذف ملف.

الجدول الخامس: هو auth-rule

يمثل هذا الجدول القواعد المطبقة في النظام، ويحوي هذا الجدول :

Name: هو اسم القاعدة

Data: معلومات القاعدة

Created-at: تاريخ الإنشاء

Update-at: تاريخ التعديل

يمكن تعريف قاعدة ما تنص على أن شخص يقوم بتحميل ملف ما فهو قادر على تعديل محتوى هذا الملف.

يعبر الجدولين files, folders جداول تخزين الأغراض، حيث أفترضنا أنها ملفات مضمنة ضمن مجلدات، حيث يوضح الشكل السابق بنية هذين الجدولين.

يعبر الجدول UserInGroup الأشخاص المشاركين في إنجاز مشروع.

يعبر الجدول groupProject عن اسمية مجموعة العمل لمشروع، ورئيس فريق العمل فيها ومرتبط هذا الجدول مع جدول UserInGroup لتحدد العناصر المشاركة بإنجاز هذا المشروع.

6.3.3– العلاقة بين الجداول

يرتبط الجدول auth-item مع كل من الجداول التالية:

- الجدول auth-item-child لتوضيح العلاقة بين الأدوار في الهرمية، فمن خلال هذا الربط نستدل من أجل كل دور ما هي الأدوار التي تقع تحته أو فوقه في الهرمية.
- الجدول auth-assignment لمعرفة كم دور مسند للمستخدم.
- الجدول auth-rule لمعرفة القواعد المطبقة على دور ما.
- الجدول groupProject لمعرفة السمات المرتبطة بكل دور، ولم يتم ربطه مع جدول المستخدمين لتحقيق الهدف المطلوب.

يرتبط الجدول groupProject مع الجدول userInGroup لمعرفة العناصر المشاركة في إنجاز مشروع ما، ويرتبط مع الجدول folders لتخزين ملفات كل مشروع في مجلد منفصل.

يرتبط الجدول files مع الجدول users لمعرفة من قام بتحميل الملف أو تعديله أو حذفه.

7.3.3– خاتمة

تمت عملية تنجيز النموذج المقترح، بتنجيز نظام RBAC بحيث يتم تعريف مجموعة من المستخدمين ومجموعة من الأدوار ضمن هرمية معينة، وتعريف مجموعة من السماحيات.

بعد عملية التعريف السابقة يتم اسناد المستخدمين للأدوار، وإسناد السماحيات للأدوار.

تم تعريف سمة واحدة في الوقت الحالي للقيام باختبار النموذج المقترح، وهي سمة رئيس فريق العمل.

حيث ستم توضيح هذه العمليات من خلال الواجهات المبينة في فصل الاختبارات والنتائج العملية.

الفصل الرابع

الاختبارات والنتائج العملية

سنقوم في هذا الفصل بعرض مجموعة من الاختبارات لإثبات إمكانية تحقيق النموذج المقترح في بيئة بنك معلومات مؤسساتي فعلي، إضافة إلى مناقشة النتائج.

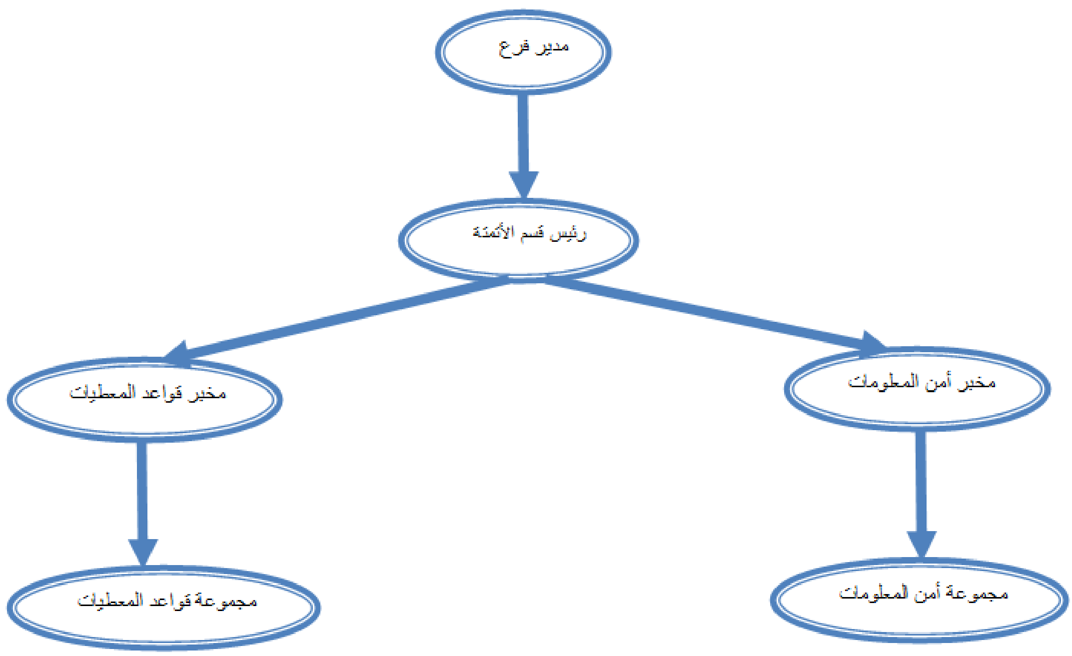
1.4- نظرة عامة

سنقدم في هذا الفصل سيناريو عملي لاختبار النموذج المقترح، والميزة التي قدمها عن النماذج المبينة في الدراسة المرجعية.

سيتم الاختبار على مؤسسة افتراضية ذات هرمية معينة، تحوي مجموعة من الموارد سنبين من خلالها مدى قدرة المستخدمين عن طريق الأدوار التي يمتلكوها بالوصول لهذه الموارد.

2.4- هرمية المؤسسة وسيناريو الاختبار

يبين الشكل التالي الهرمية التي يتم بناء سيناريو الاختبار عليها.

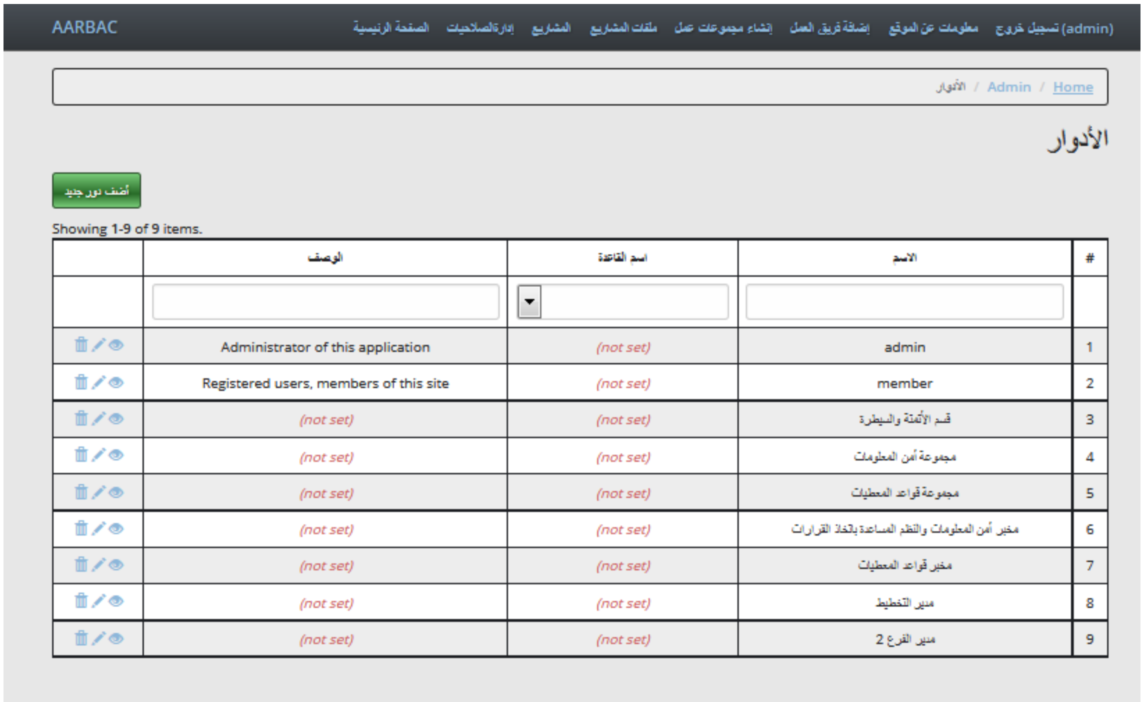


رسم توضيحي 21: هرمية سيناريو الاختبار

سنتفرض في هذه المؤسسة مجموعة من العاملين المعرفين بالشكل التالي:

رسم توضيحي 22: تعريف المستخدمين في النظام

ويبين الشكل التالي الأدوار المعرفة في المؤسسة:



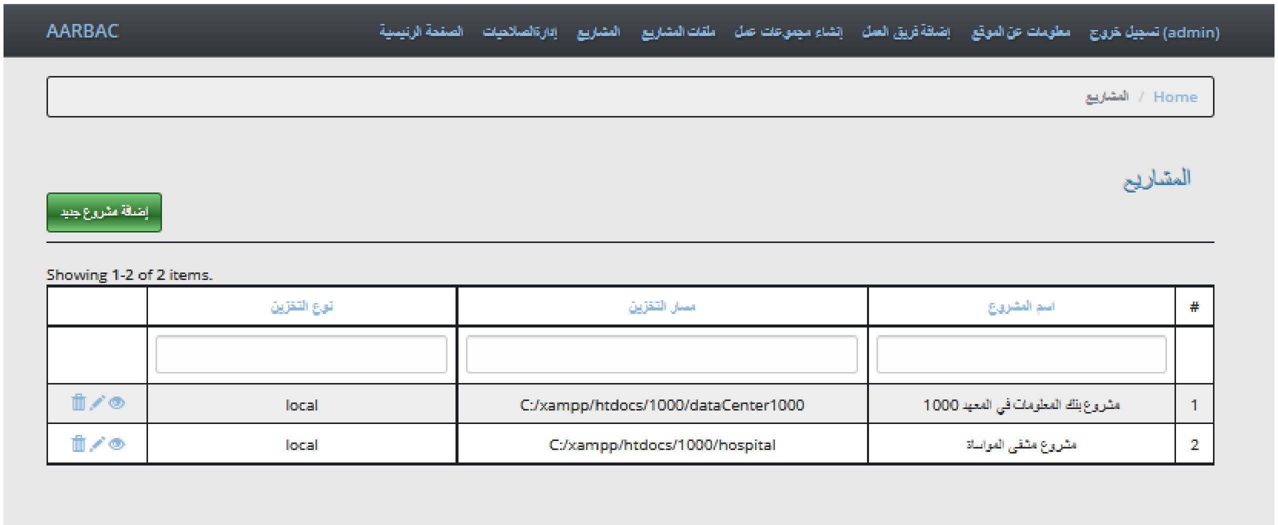
رسم توضيحي 23: أدوار العاملين ضمن النظام

يبين الجدول التالي كل مستخدم والدور المسند له والدور الأعلى منه بالهرمية:

جدول 4: المستخدمين والأدوار المستدة لهم

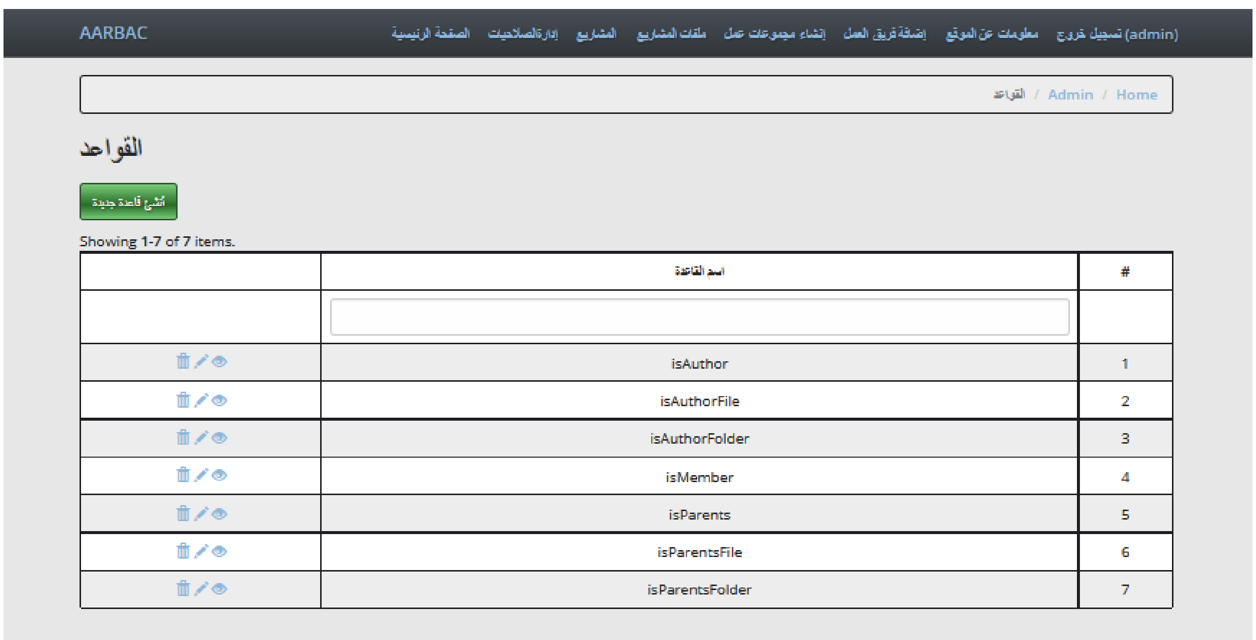
الدور	الدور الأعلى في الهرمية	المستخدم
مدير التخطيط	هو مدير النظام يقوم بتعريف الموارد	الدكتور توفيق كاملة
مدير الفرع 2	مدير المعهد	الدكتور جمال
قسم الأتمتة والسيطرة	مدير الفرع 2	الدكتور عصام
مخبر أمن المعلومات	قسم الأتمتة والسيطرة	الدكتور محمد
مخبر قواعد المعطيات	قسم الأتمتة والسيطرة	المهندس مدين
مجموعة أمن المعلومات	مخبر أمن المعلومات	المهندسين: ياسر، فادي، عمار، دانا
مجموعة قواعد المعطيات	مخبر قواعد المعطيات	المهندسين: نور، هيثم

نفترض لدى المؤسسة المشروعين المعرفين بالشكل التالي:



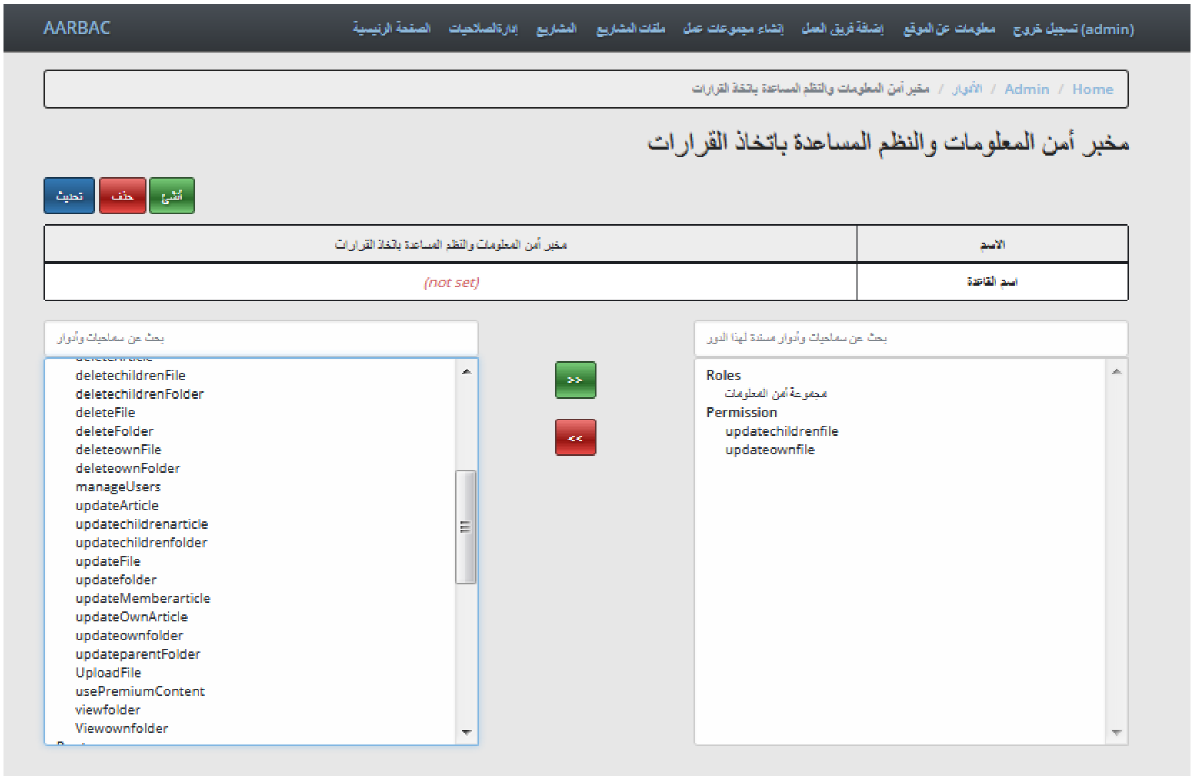
رسم توضيحي 24: المشاريع المعرفة ضمن المؤسسة

ونعرف مجموعة من القواعد المعرفة بالشكل التالي:



رسم توضيحي 25: تعريف القواعد

ومجموعة من السماحيات المسندة للأدوار:



رسم توضيحي 26: تعريف السماحيات

حيث نلاحظ من هذا الشكل أن الدور "مخبر أمن المعلومات والنظم المساعدة باتخاذ القرار" يملك الدور "مجموعة أمن المعلومات" أي أنه أدنى منه بالهرمية، كما يملك سماحية "updateChildreenFile" وسماحية "updateOwnFile" والتي تعني أن الشخص الذي يملك هذا الدور له سماحية تعديل ملفاته وملفات الأشخاص الذين يملكون الدور "مجموعة أمن المعلومات".

ملاحظة: السماحية "updateOwnFile" هي نفسها السماحية "updateFile" والتي هي عبارة عن حدث (action) مطبق عليها القاعدة "IsAuthorFile" والتي تنص على أن الشخص الذي يحمل الملف هو مالك له.

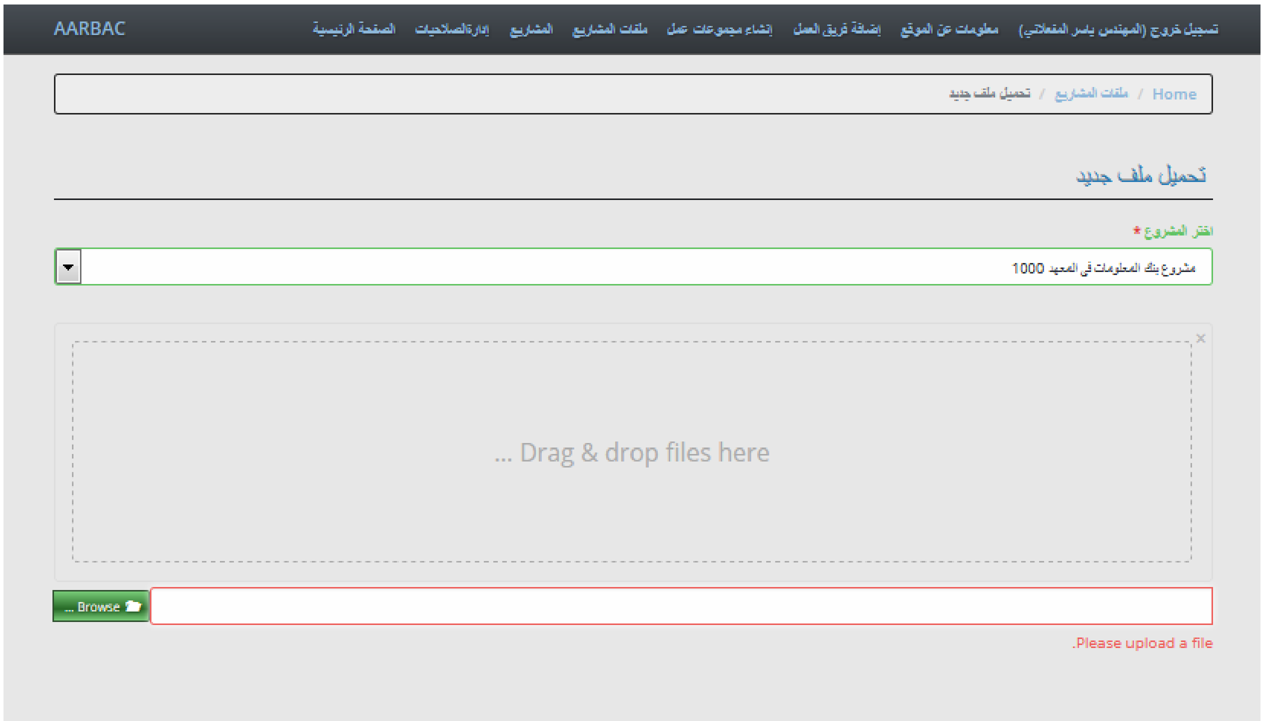
يعمل "مخبر أمن المعلومات والنظم المساعدة باتخاذ القرار" على إنجاز المشروع الأول المعرف مسبقاً وهو "مشروع بنك المعلومات في المعهد 1000"، ويعمل على ذلك المهندسين الذين يملكون دور مجموعة أمن المعلومات والمعرفين مسبقاً أيضاً.

يشارك في إنجاز هذا المشروع أحد الأشخاص الذين يملكون الدور "مجموعة قواعد المعطيات"، يعرف صاحب الدور "مخبر أمن المعلومات والنظم المساعدة باتخاذ القرار" كرئيس فريق عمل لهذا المشروع.

نذكر بالسياسة المتبعة في المؤسسة عن طريق السيناريو الحالي:

يملك الدور الذي يلعب رئيس فريق العمل لمشروع ما القدرة على الوصول لجميع ملفات هذا المشروع، ويملك هذه السماحية جميع الأدوار التي أعلى منه في الهرمية.

يسمح النظام المنجز للأشخاص بتحميل الملفات وتخزينها بمجلدات المشاريع المشاركين بها وفق السماحيات المسندة لهم كما في الشكل التالي:



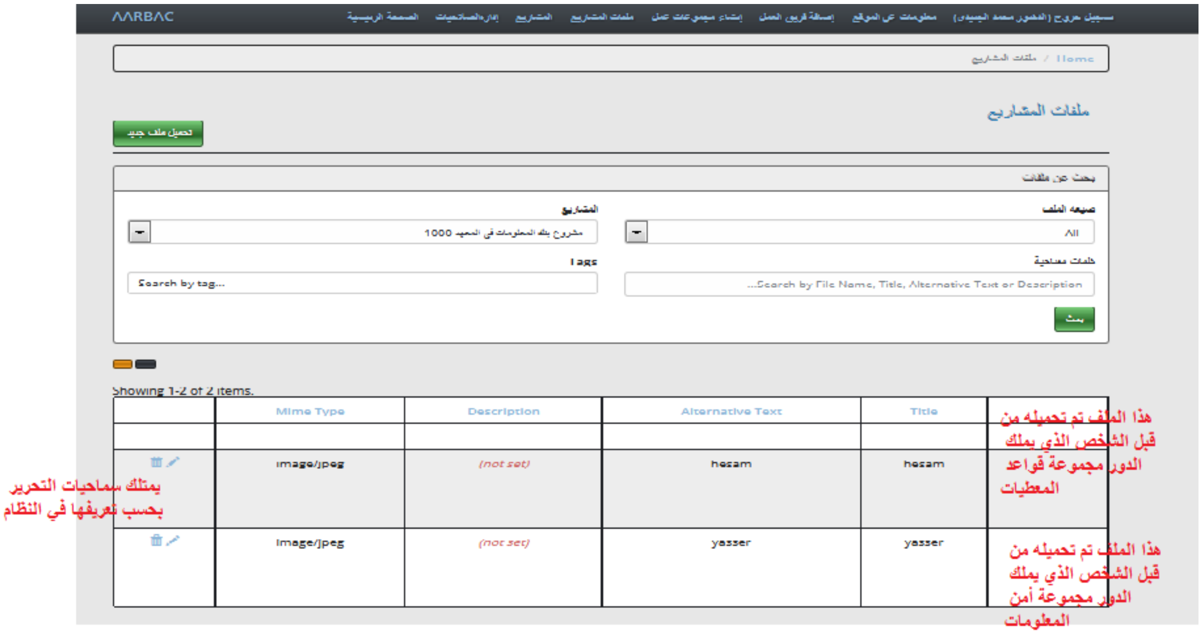
رسم توضيحي 27: واجهة تحميل ملف جديد

لنفترض أن أحد الأشخاص الذين يملك الدور "مجموعة أمن المعلومات" قام بتحميل ملف في "مشروع بنك المعلومات في المعهد 1000"، وأحد الأشخاص الذي يملك الدور "مجموعة قواعد المعطيات" قام بتحميل ملف في نفس المشروع، للعلم كلا الشخصين مشاركين بتنجيز المشروع وإلا ليس لهما السماحية للقيام بذلك ولن يظهر لهما المشروع أصلاً (وهو نوع من السرية لكي لا يعرف الأشخاص إلا المشاريع المشاركة بها).

سنناقش ما سيحدث بعد القيام بهذه العمليات:

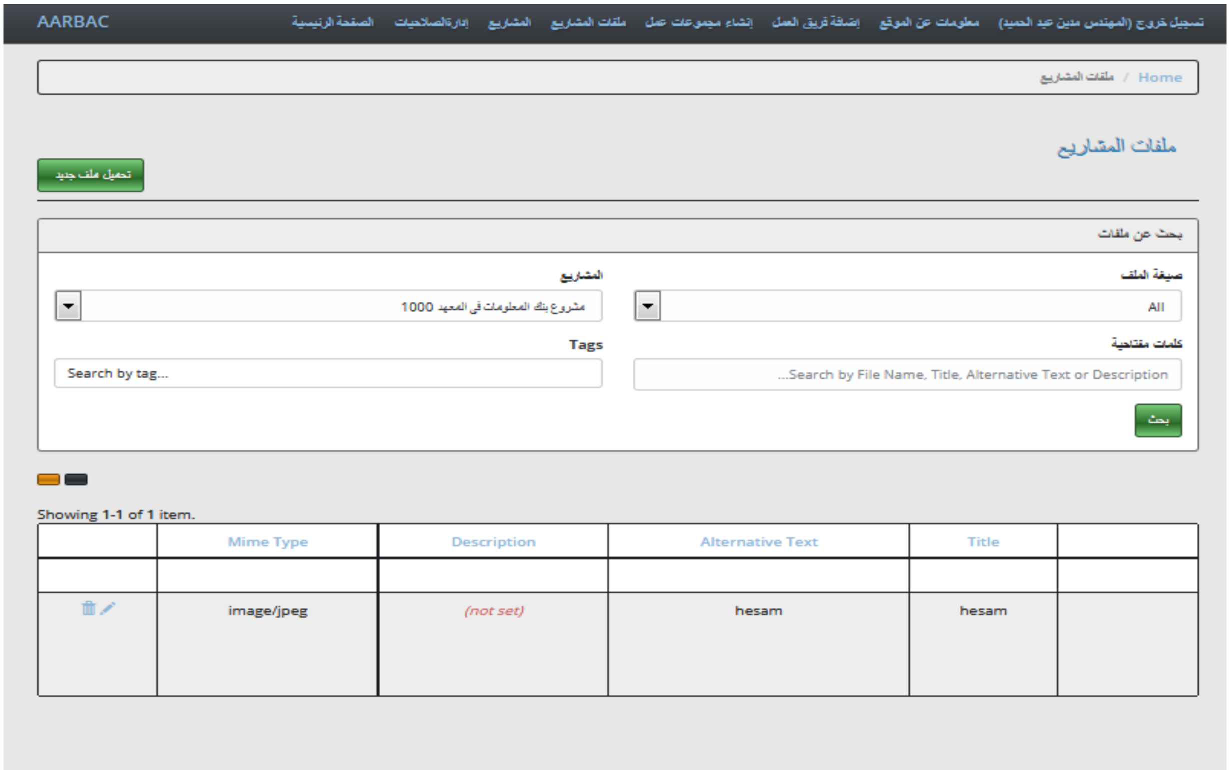
بحسب سياسة المؤسسة عند دخول الشخص الذي يملك سمة رئيس فريق العمل يجب أن يشاهد جميع الملفات الموجودة في المشروع، ويملك هذه السماحية من هم أعلى منه بالهرمية، ويملك الشخص صاحب الدور "مخبر قواعد المعطيات" والأدوار الأعلى منه بالهرمية السماحية لرؤية ملفات الأشخاص الذين هم أدنى بالهرمية، وليس جميع ملفات المشروع.

حيث نلاحظ في الشكل التالي إمكانية الوصول لكلا الملفين من قبل الدور الذي يملك سمة رئيس فريق العمل.



رسم توضيحي 28: عرض كافة الملفات للدور صاحب سمة رئيس فريق عمل

بينما نلاحظ من الشكل التالي قدرة الشخص الذي يملك الدور "مخبر قواعد المعطيات" على رؤية ملفات الأشخاص الذين تحته في الهرمية.



رسم توضيحي 29: عرض ملفات المشروع للأدوار المشاركة بالمشروع

بالمثل الأدوار الأعلى في الهرمية تمتلك نفس السماحية للدورين السابقين، حيث سيتم عرض جميع إمكانيات النظام وأهمية النموذج المقترح خلال العرض العملي، بينما في الفقرة التالية سيتم مناقشة النتائج ومقارنة النموذج المقترح مع نماذج أخرى.

3.4– النتائج

من خلال فقرة مناقشة الدراسة المرجعية بين الجدول رقم 4 أن نموذج ABAC يحقق المطلوب، لكن لم نأخذ بعين الاعتبار في حال الهرميات الكبيرة وعملية الوراثة بين مستويات الهرمية العليا من المستويات الدنيا، إضافة إلى حالة وجود عدد كبير من الموارد، فهذا سيشكل عبء على المؤسسة في إدارة التحكم في الوصول للموارد، وخاصة في حال انتقال الشخص من مستوى إلى مستوى أخر، لذا سنقارن النموذج المقترح مع نموذج ABAC.

1.3.4– مقارنة بين نموذج ABAC ونموذج AARBAC:

لنفرض لدينا هرمية مشابهة للهرمية المعرفة بالشكل رقم 1، في هذه الهرمية u مستخدم، r دور، o غرض (ملف)، s سماحية.

في نموذج ABAC نحتاج لتعريف u سياسة من أجل كل مستخدم نعرف من خلالها من هم أعلى أو أدنى منه بالهرمية، ومن أجل كل غرض نحتاج أن نعرف o سياسة لتحديد الأشخاص المسموح لهم بالوصول لهذا الغرض، ومن أجل كل مستخدم u نحتاج لتعريف s سماحية.

إذا في نموذج ABAC نحتاج عدد من العمليات:

عدد العمليات = $u(1+s) + o = u+o+u*s$

بينما في نموذج AARBAC نحتاج لتعريف r دور، وسياسة واحدة لكل الملفات من خلال السمة التي تمت إضافتها للدور، ومن أجل كل دور r نحتاج لتعريف s سماحية.

إذا في نموذج AARBAC نحتاج عدد من العمليات:

عدد العمليات = $r(1+s) + 1 = r+1+r*s$

نلاحظ من المعادلتين السابقتين، أن نموذج AARBAC يحقق المطلوب ويوفر عمليات O بعدد الأغراض (الملفات)، علاوة على ذلك فمن المعروف أن عدد الأدوار هو أقل بكثير من عدد المستخدمين، وبالتالي أيضاً تخفيض عدد أكبر من العمليات.

وتكمن القوة في نموذج AARBAC في حال إجراء أي تعديل على الهرمية، يتم القيام بعملية واحدة، بينما في نموذج ABAC يحتاج لعدة عمليات، على سبيل المثال انتقال شخص إلى مستوى أعلى بالهرمية، يحتاج نموذج AARBAC لاسناد دور جديد لهذا الشخص وإلغاء الدور المسند إليه، بينما يحتاج نموذج ABAC إلى تعديل جميع السياسات المرتبطة بهذا الشخص، وإنشاء سياسات جديدة.

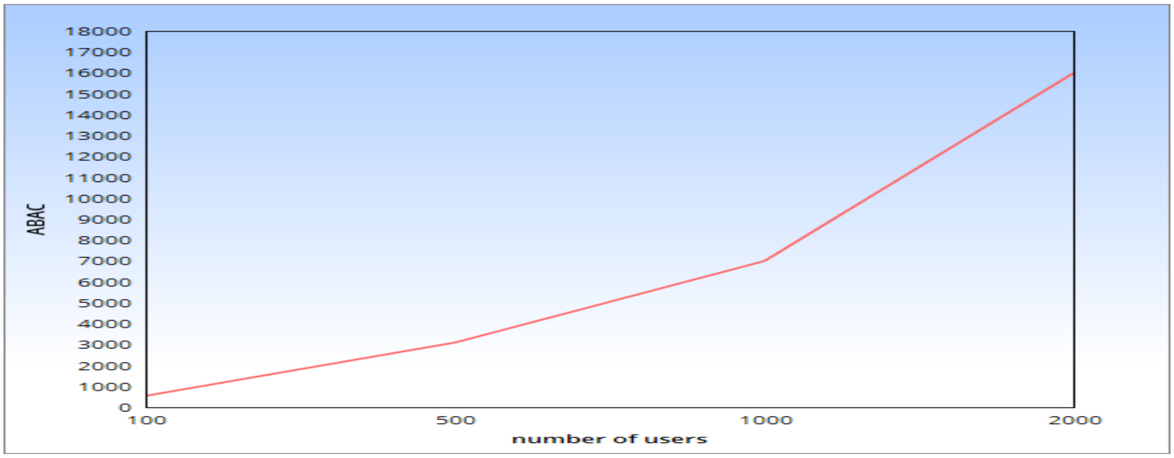
2.3.4– دراسة حالة

ليكن لدينا الجدول التالي يمثل معلومات عن المستخدمين والأدوار والسماحيات لأربعة مؤسسات مختلفة، مع عدد العمليات التي يحتاجها كل من نموذجي ABAC ونموذج AARBAC لإتمام الهدف المراد.

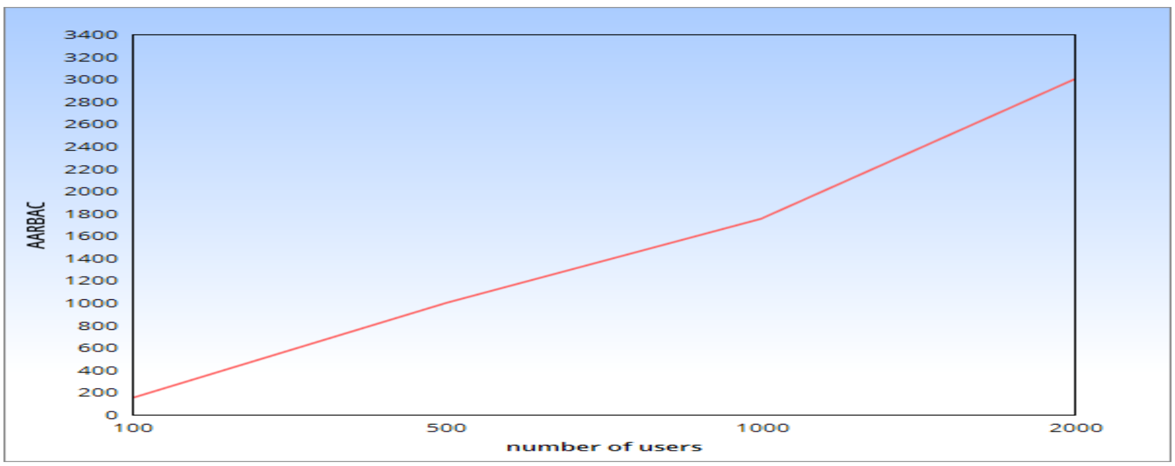
جدول 5: عدد العمليات التي يحتاجها كل من نموذجي **ABAC** ونموذج **AARBAC** لإتمام الهدف المراد

number of user	number of roles	number of objects	number of permision	abac	AARBAC
100	30	50	4	550	151
500	200	600	4	3100	1001
1000	350	2000	4	7000	1751
2000	600	6000	4	16000	3001

يمثل كل من الشكلين التالين عدد العمليات التي يحتاجها كل من نموذج ABAC والنموذج المقترح AARBAC، حيث نلاحظ أن نموذج AARBAC يحتاج عدد عمليات أقل بكثير من نموذج ABAC مقارنة بنفس البيانات.



رسم توضيحي 30: عدد العمليات التي يحتاجها نموذج **ABAC**



رسم توضيحي 31: عدد العمليات التي يحتاجها نموذج **AARBAC**

كما ذكرنا سابقاً ليس عدد العمليات الأقل المطلوبة لتنفيذ الحل هي السبب في أفضلية النموذج المقترح على النماذج السابقة، فلو أفترضنا على سبيل المثال أن الشخص الذي يملك دور "مخبر أمن المعلومات والنظم المساعدة في اتخاذ القرار" قد ترك العمل لدى المؤسسة، فما الذي سيحصل وكم عملية أو تعديل يحتاج كل نموذج لتشكيل هرمية جديدة وإسناد الدور السابق لشخص جديد.

ففي حالة نموذج ABAC نحتاج لتغيير عدد كبير من السياسات والتي تزداد طرداً مع عدد الأشخاص في المؤسسة وارتباطهم مع الشخص الذي ترك العمل، وتزداد مع عدد الموارد التي له سماحية الوصول إليها.

بينما في حالة AARBAC فنحن لسنا بحاجة إلا لعملية واحدة وهي إسناد الدور "مخبر أمن المعلومات والنظم المساعدة في اتخاذ القرار" للشخص، عندها يمتلك هذا الشخص جميع سماحيات الشخص السابق، مضافاً إليها السماحيات المسندة له

عن طريق السمات، ومن هنا أيضاً تبرز مدى الفعالية في إسناد السمات للأدوار وليس للأشخاص، فالأدوار ثابتة في المؤسسة بينما الأشخاص متغيرون.

الفصل الخامس

الخاتمة والآفاق المستقبلية

الخاتمة والآفاق المستقبلية

قمنا في هذه الرسالة بدراسة شاملة لأغلب أنظمة التحكم في الوصول، وتحليل هذه الأنظمة لتبيان فائدتها أو عدم فائدتها في حل المشكلة المطروحة، ومن ثم اقترحنا نموذج جديد للتحكم في الوصول، من خلال توسيع تمديد نموذج المعتمد على الدور RBAC بنموذج المعتمد على السمات ABAC والاستفادة من مزايا كلا النموذجين.

تم تمديد نموذج المعتمد على الدور بإضافة سمات للأدوار، على عكس النماذج السابقة التي اعتمدت على إضافة السمات للمستخدمين، حيث حافظنا على التسلسل الهرمي وفي الوقت نفسه قدمنا حلاً للمشكلة المطروحة.

من النتائج لاحظنا أن النظام المقترح AARBAC لديها أداء أفضل من نموذج ABAC، وعدد عمليات أقل بكثير بالمقارنة بعدد العمليات في ABAC، وذلك في حال عدد المستخدمين هو نفسه في كلا النموذجين، وعملية تطبيق السيناريو تتم عل بيئة مؤسساتية تخزن ملفاتها في بنك معلومات يصنف صغيراً، وتمتلك هرميات متداخلة تنشئ بسببها تضارب في الصلاحيات.

هنالك العديد من الاتجاهات والمحاور التي يمكن العمل عليها مستقبلاً منها:

1- إضافة هرمية للسمات المرتبطة بالأدوار، بحيث إذا كان دور ما يملك سمة ما، فإنه يرث مباشرة جميع السمات التي تتوضع تحت هذه السمة في الهرمية، ولا داعي لإسنادها للدور مرة أخرى.

2- إضافة سمات للأغراض، فعلى سبيل المثال من المحتمل أن يكون المشروع سري أو غير سري وبناءً على هذه السمة يمكن السماح لأدوار معينة بالوصول لملفات هذا المشروع.

المراجع

[1] NIST (National Institute of standards and Technology) (2013) ABAC workshop presentation. Retrieved from http://csrc.nist.gov/projects/abacjuly2013_workshop/presentations.html/.Accessed June 2014.

[2] Ryan Ausanka Crues, Methods for Access Control Advances and Limitations, 2013.

[3] David F. Ferraiolo and D. Richard Kuhn, Role-Based Access Controls, 1992.

[4] Abhishek Majumder, Suyel Namasudra and Samir Nath, Taxonomy and Classification of Access Control Models for Cloud Environments, Springer-Verlag London 2014.

[5] Aaron Elliott and Scott Knight. Role explosion: Acknowledging the problem. In Software Engineering Research and Practice, pages 349–355, 2010.

[6] David Ferraiolo, Janet Cugini, and D Richard Kuhn. Role-based access control (RBAC): Features and motivations. InProceedings of 11th Annual Computer Security Application Conference, pages 241–48, 1995.

[7] Ravi S Sandhu, Edward J Coynek, Hal L Feinsteink, and Charles E Youmank. Rolebased access control models yz.IEEE computer, 29(2):38–47, 1996.

[8] Ravi S. Sandhu, David F. Ferraiolo, and D. Richard Kuhn. The NIST model for rolebased access control: towards a unified standard. InACM Workshop on Role-Based Access Control, pages 47–63, 2000.

[9] Vincent C Hu, David Ferraiolo, and D Richard Kuhn.Assessment of access control systems. US Department of Commerce, National Institute of Standards and Technology, 2006.

[10] Cui-xiao ZHANG, Ying-xin HU, Guo-bing ZHANG, Task-Role Based Dual System Access Control Model, IJCSNS International Journal ofComputer Science and Network Security, VOL.6 No.7B, July 2006.

[11] Gartner. Predicts 2014: Identity and access management (2013). Retrieved from ttp://www.gartner.com/. Accessed April 2014.

[12] Vincent C Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to Attribute Based Access Control (ABAC) definition and considerations. NIST (National Institute of standards and Technology) Special Publication, 800:162,2014.

[13] Elisa Bertino. Data protection from insider threats. Synthesis Lectures on Data Management, 4(4):1–91, 2012.

[14] JAEHONG PARK, RAVI SANDHU, The UCON ABC Usage Control Model, 2004.

[15] Chen Danwei, Huang Xiuli, and Ren Xunyi, Access Control of Cloud Service Based on UCON, 2009.

[16] JAEHONG PARK, George Mason University, and RAVI SANDHU, the UCON ABC Usage Control Model, 2004 ACM 1084-4309/2004/0400-0001.

[17] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role Based Access Control Models," Computer (Long. Beach. Calif)., vol. 29, no. 2, pp. 3847, 1996.

[18] Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding Attributes to Role-Based Access Control. IEEE Computer 43, 79-81 (2010).

[19] Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding Attributes to Role-Based Access Control. IEEE Computer 43, 79-81 (2010).

[20] Jin, X., Sandhu, R., Krishnan, R.: RABAC: role-centric attribute-based access control. In: Kotenko, I., Skormin, V. (eds.) MMM-ACNS 2012. LNCS, vol. 7531, pp. 84–96. Springer, Heidelberg (2012).

[21] Qasim Mahmood Rajpoot1, Christian Damsgaard Jensen, and Ram Krishnan: Attributes Enhanced Role-Based Access Control Model: Springer International Publishing Switzerland 2015 S. Fischer-H¨ubner et al. (Eds.): TrustBus 2015, LNCS 9264, pp. 3–17, 2015.

[22] Smriti Bhatt, Farhan Patwa, Ravi Sandhu, ABAC with Group Attributes and Attribute Hierarchies Utilizing the Policy Machine, ABAC'17, March 24 2017, Scottsdale, AZ, USA.

[23] Palak J Manjrawala, Prof. Gurucharan Sahani M.E Student, Dept. of C.E., SVIT, Vasad, India Asst. Professor, Dept. of C.E., SVIT, Vasad, India: An Automatic Mechanism for Converting Existing RBAC System's Roles into ABAC System Policies Using Machine Learning: International Journal of Innovative Research in Computer and Communication Engineering.

[24] P. Mell, J. M. Shook, and S. Gavrila. Restricting insider access through efficient implementation of multi-policy access control systems. In Proceedings of the International Workshop on Managing Insider Security Threats, pages 13{22. ACM, 2016.

[25] D. Servos and S. L. Osborn, \HGABAC: Towards a formal model of hierarchical attribute-based access control," in International Symposium on Foundations and Practice of Security. Springer, 2014, pp. 187-204.

النشر

تم نشر ورقة علمية بعنوان "Adding attributes to Roles in Role Bassed Access Control Model" في المؤتمر الدولي للحساسات والشبكات والتقنيات الحديثة (International conference on sensors, networks, smart and emerging technologies).

فيما يلي قبول المقالة في المؤتمر، مع دعوتي لحضور المؤتمر لتقديم المقالة.

وقد تم حضوري للمؤتمر وتقديم المقالة والاستفادة من الملاحظات والنصائح التي وجهتي لي أثناء التقديم.

SENSET2017
International Conference on
SENSORS, NETWORKS, SMART and
EMERGING TECHNOLOGIES
12-14 September 2017
Beirut - Lebanon

Invitation to attend and present at SENSET2017

Dear Dr. Yasser Almofaalani,

On behalf of the Conference SENSET2017 (Sensors, networks, Smart and Emerging Technologies) I am pleased to inform you that the technical paper entitled:

Adding attributes to Roles in Role Based Access control model

has been accepted by the SENSET2017 Technical Committee after review for technical merits. As part of the publication requirements, it is a **mandatory requirement** that you register and attend the Conference to present the paper and discuss your work. Please make the necessary travel arrangements and visa applications as early as possible to be able to present your paper and lead the subsequent technical discussions.

The Conference will be held on September 12-14, 2017 at The Movenpick hotel in Beirut. Thank you for submitting high quality paper to our conference, and we look forward to hearing your presentation.

The goal of SENSET2017 is to have good technical work written and presented for discussion. Scientists attend the Conference and purchase the Conference proceedings in order to learn about Sensors, Networks, Smart and Emerging Technologies.

Best Regards,
Sawsan SADEK
Conference Chair

SADEK

الملخّص

تحتاج أي شركة أو مؤسسة لنظام متين يكون قادر على إدارة بياناتها وحمايتها وتنظيم عمليات الوصول لهذه البيانات من قبل موظفي المؤسسة أو زبائنها. إن من أشهر نماذج التحكم في الوصول للموارد أو البيانات في وقتنا الحالي هما نموذج التحكم في الوصول المعتمد على الأدوار (Role base Access Control) ونموذج التحكم في الوصول المعتمد على السمات (Attribue Base Access Control).

لكلا النموذجين مجموعة من السيئات والمزايا التي يتمتع بها، إن اختيار نموذج لإدارة التحكم في الوصول للموارد في شركة ما يعتمد على بنية المؤسسة والسياسة المتبعة للوصول للموارد في هذه المؤسسة.

فإذاً لا يمكن الاعتماد على نموذج واحد وتطبيقه في جميع المؤسسات، ولكن بنفس الوقت تواجه أغلب المؤسسات مشكلة أن النموذج المختار لكي يناسب طبيعة هذه المؤسسة لا يخلو من بعض النقص، أو عدم القدرة على الالمام بجميع السيناريوهات أو السياسات المتبعة في هذه المؤسسة لتحقيق الوصول الآمن للموارد.

لذا سنقدم في هذه الرسالة نموذج مقترح للتحكم في الوصول يعتمد على تمديد نموذج التحكم في الوصول المعتمد على الأدوار بنموذج التحكم في الوصول المعتمد على السمات، بهدف الاستفادة من مزايا النموذجين معاً وحل مشكلة الوصول للموارد بشكل آمن في المؤسسات ذات الطبيعة الخاصة التي لا يستطيع فيها نموذج واحد إدارة عمليات الوصول.

هذه المؤسسات التي تملك العديد من البيانات والملفات المتنوعة بحاجة لبنك معلومات يضمن الحماية اللازمة لهذه البيانات، هذا البنك يفترض أن يكون بنك معلومات صغير لأنه موجه لمؤسسة ذات هرمية، وغير موجه لجميع العموم.