

ORACLE
DATABASE **12^c**

Oracle Database 12c

Real Application Security

Oracleホワイト・ペーパー | 2014年9月



目次

はじめに	2
Oracle Real Application Securityの概要	3
アプリケーション開発のセキュリティ要件	4
現在のセキュアなアプリケーションの開発	4
Oracle Real Application Securityモデル	6
アプリケーション・ユーザー	8
アプリケーション・セッション	8
アプリケーション権限	10
アプリケーション・ロール	10
データ・レلم	11
アクセス制御リスト	12
データ・セキュリティ・ポリシー	12
Oracle RAS認可サービス	13
Oracle RASセキュリティ・ポリシーのライフ・サイクル	13
Oracle Real Application Securityの使用例	14
Oracle RASを使用したデータ・セキュリティ	15
アプリケーション開発プラットフォームとの統合	17
Java EEのネイティブ統合	17
Oracle APEXとのネイティブ統合	18
Oracle RASの拡張機能	19
Oracle Real Application Security (Oracle RAS) の利点	20
その他	21

はじめに

過去20年間にわたり、一般的なアーキテクチャとして、従来のクライアント・サーバー・モデルは3層モデルに移行してきました。この移行中に、セキュリティ制御が実際のデータ自体から遠く離れたため、アプリケーションのエンド・ツー・エンドのセキュリティが低下し、開発プロセスが複雑になりました。また、このような変化によって、プライバシーやコンプライアンスの規制の実施が複雑になり、データ侵害やその他の不正アクセスの増加につながるセキュリティ・ギャップが生まれるようになりました。3層モデルには、次のような多くのセキュリティ上の課題があります。

- » セキュリティ・モデルと認可ロジックがアプリケーション固有であるため、複数のアプリケーション間でのセキュリティ実施モデルが一貫性のない不完全なものになります。
- » 操作がデータベース内でシングル・ユーザーによって実行されるため、ID伝播とエンドユーザー・アクティビティの監査機能が低下します。開発者が監査ログを作成する必要があるため、アプリケーション固有の監査が断片化されます。
- » アクセス制御ポリシーがアプリケーション・ロジック内に組み込まれており、アプリケーションごとに固有のポリシー・インフラストラクチャがあります。メンテナンス・コストが増加し、ポリシーの拡張が複雑になります。
- » (中間層のJava、組込みのPL/SQLロジック、データベースの直接接続などの) データ・エントリ・アクセス・ポイントごとに個別のアクセス制御ポリシーが必要です。未確認の接続で無制限にアクセスできる場合があります。
- » アプリケーション層とデータベース層の間で、共通のポリシー・モデルと実施インフラストラクチャがないため、ポリシー管理者がセキュリティ・モデルを完全に把握できません。アプリケーションとデータベースの両方で、ポリシーが手続き型ロジックとして組み込まれます。

このため、現在のアプリケーション・セキュリティ・ポリシー（およびそのカスタマイズによる実施）では、アプリケーションが不安定で断片化された、脆弱なものになる可能性があります。データ・セキュリティがデータから離れたものになると、すべての関係者の課題が増えます。

- » アプリケーション・アーキテクト：データベース内のアプリケーション・エンドユーザーに関する知識がないまま、アクセス制御実施の大部分をアプリケーション開発者が担当することになります。データベース内では、アプリケーション・データへの完全アクセス権を持つ特権データベース・ユーザーとしてアプリケーション・コードが実行されます。このためアプリケーション開発者は、不正アクセスやデータ漏えいの防止に細心の注意を払う必要があります。
- » アプリケーションのセキュリティ管理者：アプリケーションごとに独自のセキュリティ・ポリシー構造と実施メカニズムが構築されます。このためセキュリティ・チームにとっては、アプリケーション・セキュリティ・ポリシーと、それが企業全体のデータ・アクセス・ポリシーに与える影響を検証するのが困難です。アプリケーション・ミドルウェアとアプリケーション・ファイアウォールがデータベースへのすべてのデータ・アクセス・パスをカバーしているとは限りません。このため、アプリケーションで実施されるセキュリティでは、機密データが多くの攻撃（アプリケーションの迂回攻撃や、攻撃者がデータベースに直接接続する場合のインサイダー攻撃など）に対して脆弱なままである可能性があります。
- » セキュリティ監査の担当者：データベースでエンドユーザーのIDが認識されておらず、ネイティブでエンドユーザーのアクティビティを監視できないため、アカウントビリティが低下します。

Oracle Real Application Securityの概要

Oracle Real Application Security (Oracle RAS) では、次世代のアプリケーション・アクセス制御フレームワークがデータベース内に導入されているため、3層と2層のアプリケーションで、そのセキュリティ要件を宣言的に定義、プロビジョニング、実施できます。Oracle RASではポリシー・ベースの認可モデルが導入されており、データベース内でアプリケーションレベルのユーザー、権限、ロールが認識されるため、ビジネス・オブジェクトを表すレコードの静的コレクションと動的コレクションの両方に対するアクセスを制御できます。Oracle RASでは、データベースに対するアプリケーション・ユーザー・セッションのセキュアな伝播が組み込みでサポートされているため、アプリケーション・ユーザー、およびそのロールとセキュリティ・コンテキストに関して、データについてのセキュリティ・ポリシーを直接表すことができます。また、Oracle RASは認可決定サービスとして機能し、アプリケーションの中間層でのセキュリティ実施をサポートすることもできます。

Oracle RASには、次のようなセキュリティ設計の原則が実装されています。

- » データに関するアクセス制御は、データベースの特権ユーザーやスキーマ所有者ではなく、エンドユーザーの権限に基づいて決定されます。アクセス制御の実施は**最小権限の原則**に従っており、エンドユーザーが実行できる操作は、その業務に必要な権限によって制限されます。
- » アクセス制御ポリシーが、データのできるだけ近くで実施されます。このため、アプリケーション・コードや2層または3層のアプリケーション経由のアクセス・パスに関係なく、同一のセキュリティ・ポリシーが実施されます。
- » アクセス制御ポリシーが宣言的で、アプリケーションの手続き型ロジックから分離されています。このため、アプリケーションの開発とセキュリティ・ポリシーの管理が簡単です。宣言的アクセス制御ポリシーによって、最小限のアプリケーション・コード変更で（または変更なしで）、既存のアプリケーションで新しいセキュリティ要件を定義、指定、変更、実施できます。
- » データ・アクセス制御ポリシーは、汎用的なパターン（主キー/外部キー、マスター・ディテール、組織ツリー、マルチテナントのストライプ化、パラメータ化可能な一般的なデータ・アクセス・パターンなど）をサポートします。

このため、アプリケーション内でのOracle RASのセキュリティは従来のソリューションより強力です。データベース内のエンドユーザー認可コンテキストに基づき、エン트리・ポイント（直接的なOracle Application Express (Oracle APEX) やミドルウェア・アプリケーション・サーバーなど）に関係なく実施されるためです。データ・アクセス制御の実施は、さまざまなデータ・アクセス・パターンに基づいて最適化されます。データベースで、すべての認可ポリシーとユーザーのセッション・コンテキストの実行が認識されているためです。Oracle RASでは、認可コードの書込みと組み込み、およびその安全性の確保が不要であるため、宣言的ポリシーを使用して簡単にアプリケーションを開発できます。また、中間層とデータベースのセキュリティが同一であるため、ポリシー管理が簡単です。

Oracle RASでは、アプリケーションのデータと操作に宣言的アクセス制御ポリシーを使用して、データの近くでセキュリティを実施し、3層と2層の両方のアプリケーションでエンド・ツー・エンドのセキュリティを実現します。

アプリケーション開発のセキュリティ要件

アプリケーションでは通常、最初にその情報モデルを定義してから、ユーザーとそのグループ・メンバーシップの役割、およびシステム、環境、企業ポリシーによるその他の制約などの要素に基づいて、操作とデータのセキュリティを実施します。一般的な人事管理（HR）アプリケーションの運用上のセキュリティ要件を以下に示します。

- » すべての従業員が電話番号、勤務先住所、電子メール、役職、プロジェクトの割当て、組織ツリーを見ることができる。ただし、従業員は自分の携帯電話番号、自宅住所、銀行口座情報を更新できる。また従業員は、自分の給与と公的ID（社会保障番号（SSNなど））を見ることができる。
- » マネージャーは自分の部下に対してのみ、“給与査定”期間中だけ、その給与を見て更新できる。ただし、部下の役職や担当プロジェクト名はいつでも更新できる。また、役職やプロジェクト名の更新権限を管理者に委任できる。
- » HRマネージャーは、担当グループ内のすべての従業員のすべてのデータを見ることができる。ただし、EUの規制などの地理的な制約がない場合に限る。
- » 法務部門の特定の人は、特定の従業員のすべてのデータにアクセスできる。
- » 給与報告アプリケーション（2層ツール）は、すべての従業員のSSN、銀行情報、自宅住所にアクセスできるが、付与されたストック・オプションや従業員の業務査定を読み込むことはできない。
- » 内部監査チームは、機密データへのアクセスまたは更新を行った人の監査記録を見ることができる。

以上のような運用要件の場合、HRアプリケーションのセキュリティ要件は次のようになります。

- » ユーザーのID、グループ、役割（マネージャー、法務、人事管理など）に基づき、アプリケーション・データにアクセスする。
- » 機密情報の列のデータ（SSNや給与）にアクセスするには、認可を受ける必要がある。
- » マネージャーのアクセス権は、個々の権限レベルで他のユーザーに委任される。
- » 組織および行政上の要件を実施するため、時間、地理的条件、データに基づいて追加制約を実施する。
- » 2層の管理およびレポート・ツールに対しては、アクセスを認可されたデータだけに制限する。
- » エンドユーザーと管理者のアクティビティを監査し、違反を追跡する。

以上の例はHRアプリケーション用ですが、市販のパッケージ・アプリケーションや社内用のカスタム・アプリケーションにも、同様の要件が適用されます。

現在のセキュアなアプリケーションの開発

アプリケーションを保護するには、さまざまな考慮事項（認証、認可、脆弱性スキャン、ファイアウォール保護、セキュアな構成など）への対応が必要ですが、このホワイト・ペーパーでは、認可について重点的に説明します。Oracle RASについて説明する前に、データ中心のアプリケーションの保護方法の現状と、そのようなソリューションの制限事項について簡単に説明します。

データベースを利用するアプリケーションは、2つのグループに分類できます。一般的な3層アプリケーションは中間層で実行され、基本的にアプリケーション・フレームワークから認可サービスを実装または使用し、特権スキーマ・ユーザーとしてデータベースに接続します。一方、メタデータの管理やパッチのインストールに使用される2層アプリケーションは、データへの完全アクセス権や、データベース・ストアド・プロ

シー ज्या内でコーディングされた実施方法によって、信頼できるアプリケーションとして実行されます。要件を追加して管理者によるアクセスを制御すると、図1のように、セキュリティの全体像がすぐに複雑になります。南京錠は、アクセス制御が実施されるポイントを示しています。パッチ適用とダイレクト・ユーザー・アクセスは、DBのセキュリティ・メカニズムを使用して実施されることになっています。データを保護するには、3層アプリケーションのセキュリティの問題に対応するだけでは不十分であり、データ・アクセス・パスに関係なく、関連するすべての特権アプリケーションでも同様の対応を行う必要があります。

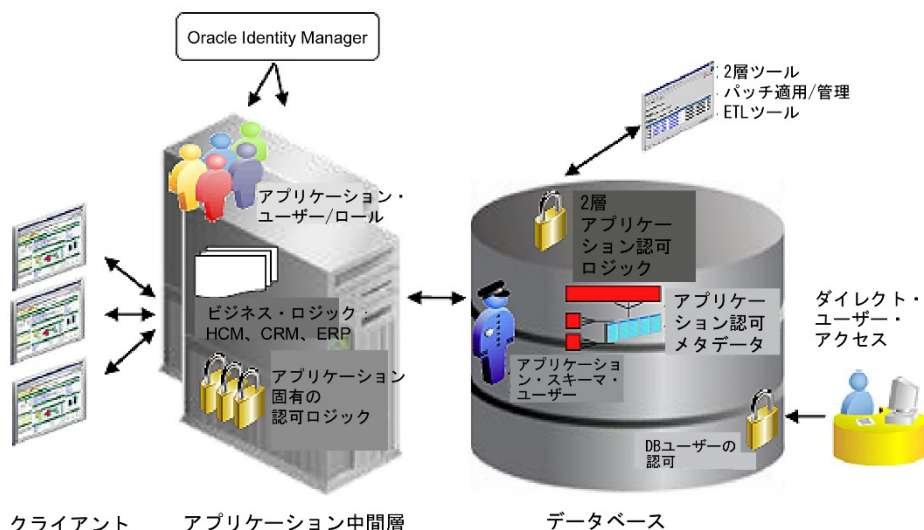


図1：多層アプリケーションの一般的なセキュリティ実施

アプリケーション・サーバーで実行される一般的な3層アプリケーションは、図1のように、特権スキーマ・ユーザーとしてデータベースに接続してから、アプリケーション・ユーザーと権限に基づいて適切なアクセス権を実施します。データベースのユーザーとロールの既存のデータベース・オブジェクト権限（選択、挿入、更新、削除）は、数千ものアプリケーション・ユーザーがいる3層アプリケーションでは、限定的にしか使用できません。アプリケーション・アーキテクト・モデルでは、一連の関連する表でのセキュリティ要件が複雑になります。このような表では、適用可能な行と列のリストがアプリケーション・オブジェクトを表し、サブジェクトがアプリケーションのユーザーとロールであり、操作がアプリケーション固有です。このようなアプリケーション構造（オブジェクト、ロール、操作）は通常、アプリケーションによって直接データベースに保存され、管理されます。その後、アプリケーション開発者がこれらの構造を使用してカスタム・セキュリティ・ポリシーを記述および実装します。アクセス制御ポリシーは、アプリケーション・コード内に分散しており、一部は中間層のプログラム・コードに、一部はデータベースのストアード・プロシージャにあります。また、アプリケーション開発者は、認可メタデータと実施モジュールの保護も行います。

現在、認可の実装に使用されている一般的な手法には、次のようなものがあります。

» 接続共有によるOne Big Application User

多層アプリケーションでは、ユーザー単位でデータベース接続を維持するにはコストがかかるため、アプリケーションで、すべてのアプリケーション・ロジックのデータベース接続を共有します。アクセス制御とアカウントビリティの懸念事項を減らすため、一部のアプリケーションではアプリケーション変数を使用してエンドユーザーのセキュリティ・コンテキストをデータベース・セッションに保存していますが、このためにはセキュリティ・コンテキストを頻繁に切り替える必要があり、パフォーマンスとスケーラビリティに影響します。

アプリケーション・ユーザーのIDとセキュリティ・コンテキストはデータベース内でセキュアに使用できるわけではないので、アプリケーションで監査ロジックを実装して維持する必要があります。アプリケーションでは、追加の列や表を使用して監査データを記録できますが、この監査はデータベースの監査サービスとは統合されていません。

» クエリー・リライト

アプリケーション・ロジックやOracle Virtual Private Database (Oracle VPD) 経由で使用できるクエリー・リライト・メカニズムとセキュア・アプリケーション・コンテキストを使用すると、問合せで行レベルのアクセス制御ルールを適用して、機密情報を保護できます。ただし、クエリー・リライト・メカニズムの場合、ミドルウェア・プログラム・コードかサーバー側のストアド・プロシージャで、セキュリティ・ポリシーを手動で実装する必要があります。アプリケーションによっては、クエリー・リライト・ルールがとても複雑になり、拡張や理解が非常に困難になる可能性があります。Oracle VPDを使用した場合、データのフィルタリングのためのWHERE句を追加するコールアウト関数のフックを使用できるのですが、アプリケーションでは引き続き、アプリケーション・ユーザー/ロール、セッション・ステート、アクセスをリクエストするアプリケーションレベルの操作、およびフィルタの作成に必要なアプリケーションレベルのプリミティブに関する情報を提供するインフラストラクチャを構築する必要があります。共通のインフラストラクチャがないと、アプリケーションごとにそのメタデータを管理する必要があるため、それを把握して再利用することが難しくなります。

従来のデータベースには、セキュアなアプリケーションの開発に必要な次の機能がありません。

- » データベースでは、アプリケーションのエンドユーザーとそのアプリケーションレベルの認可やロールが認識されません。これらは、Oracle Identity Management (Oracle IDM) ストアド・プロビジョニングされる場合があります。データベースで認識されているアプリケーション・ユーザーがないと、データベースがアプリケーション・セッションをサポートできません。
- » アプリケーションレベルのアクセス制御要件は、データベースの表やビューには表示されませんが、通常は、レコードや細かいビジネス・オブジェクト（注文書、従業員レコード、従業員の個人識別情報 (PII) 属性など）の属性に表示されます。ターゲット・ビジネス・オブジェクトのデータは、さまざまなデータベース・スキーマの多くの表やビューにわたる場合があります。アプリケーションの統合展開やクラウド展開では、セキュリティ・ポリシーがテナント固有のデータにあります。このデータではテナント識別子がセキュリティ・ポリシーのターゲット属性になります。
- » 最後に、アプリケーション・データ・セキュリティ・ポリシーは、このようなビジネス・オブジェクトのアプリケーションレベルの操作（“従業員の休暇申請の許可”や“PIIデータの表示”など）に基づいて指定されます。従来のデータベースには、データベース表の行と列に対してこのようなポリシーを指定するための、アプリケーション操作に基づく認可プリミティブがありませんでした。

Oracle Real Application Securityモデル

図2の多層アプリケーションは図1と同じですが、Oracle RASセキュリティ・フレームワークを使用して開発、展開されています。Oracle Database 12cのReal Application Securityのアクセス制御モデルは機能豊富で宣言的であり、データベース内でアプリケーション固有の認可プリミティブ（ユーザー、ロール、権限）をネイティブでサポートしています。データ・セキュリティについては、Oracle RAS対応のデータベースでデータ・アクセスを自動的に決定できるため、関連する行と列のみがユーザーに戻されます。また、アプリケーションでは、Oracle RASを使用してアプリケーションの操作に関連するアクセス制御ポリシーを見つけ、これを中間層で実施することもできます。Oracle RASの認可結果は中間層の問合せ結果セットにキャッシュできるため、データベースのラウンドトリップを回避できます。Oracle RASは、セキュリティとパフォーマンスの両方の問題に対応できるように設計されています。

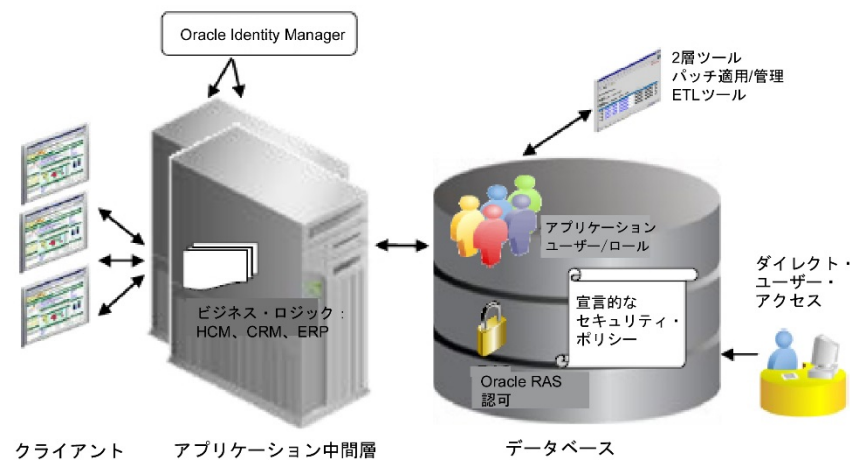


図2：Oracle RASを使用した多層アプリケーションでのセキュリティの実施

Oracle RASモデルでは、Oracleデータベース内で次のコンポーネントが導入されています。

- » アプリケーション・ユーザー：アプリケーション層とデータベース層で統合されているアプリケーションのエンドユーザーIDです。これらのユーザーはスキーマレスであり、自分のデータベース・オブジェクトやデータベース・リソースを所有していません。
- » アプリケーション権限：アプリケーションレベルの操作の実行を制御する、名前付き権限です。これらの操作は、データベース表の行や列、またはアプリケーション・アーチファクト（ワークフローのタスクやボタンを表すUIアーチファクトや、Webアプリケーションのページなど）にあります。
- » アプリケーション・ロール：アプリケーション権限や、その他のアプリケーション・ロールまたはデータベース・ロールのグループです。ユーザー・プロビジョニング中は、これらのロールがアプリケーション・ユーザーに割り当てられます。
- » データ・レルム：表または関連するアプリケーション表のグループ内の、行の論理セットのコレクションです。データ・レルムは、データ・セキュリティ・ポリシーを指定するための主要な構造です。データ・レルムはアプリケーションレベルのリソースやビジネス・オブジェクトを表し、SQL条件を使用して定義されます。
- » セッションの名前空間の属性：データ・レルムを定義するためにSQL条件で使用できる、属性値ペアのコレクションです。各コレクションは、アプリケーションの名前空間で、関連するアクセス制御ポリシーを使用して管理されます。
- » アプリケーション・セッション：データベース内でアプリケーション・ユーザーのセキュリティ・コンテキスト（ロールや名前空間の属性）に対応する、アプリケーション・ユーザーのセッションです。このようなエンドユーザー・セッションはアプリケーション層経由で作成され、データベースでネイティブにサポートされます。
- » アクセス制御リスト（ACL）：ユーザーやロールに対する権限付与の名前付きリストです。Oracle RAS ACLでは、権限付与にさまざまな制約（所定の権限の剥奪など）を付けることができます。
- » データ・セキュリティ・ポリシー：データ・レルムをACLと関連付けることで、データ・レルムを保護します。

認可サービス：現在のセッションで、権限がユーザーに付与されているかどうかを確認します。
図3は、ACLを使用して保護対象の特定の行セットやデータ・レلم（部下のレコードなど）に権限を付与する方法を示しています。

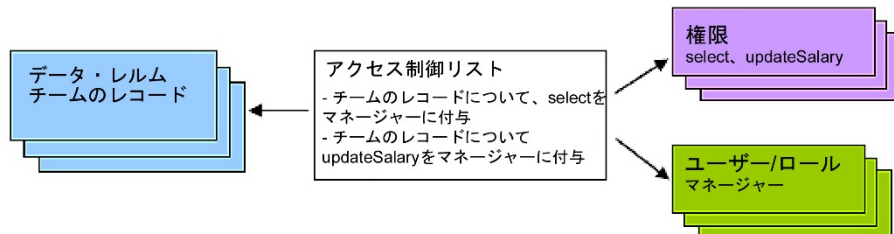


図3：Oracle RASデータ・セキュリティ・ポリシーのコンポーネント

アプリケーション・ユーザー

Oracle RASでは、アプリケーション・エンドユーザーを示すために、データベース内のアプリケーション・ユーザーの概念が導入されています。これらのユーザーはスキーマレスであり、自分のデータベース・スキーマやデータベース・オブジェクトを所有できません。ただし、アプリケーション・ユーザーはデータベース・ユーザーと同様に、オブジェクトの名前解決のためのデフォルト・スキーマを持つことができます。データベース権限は、アプリケーション・ロール経由でアプリケーション・ユーザーに付与できます。

アプリケーション・ユーザーは、IDストアやデータベースでプロビジョニングできます。3層アプリケーションにアクセスするユーザーは、通常は中間層で認証され、そのIDコンテキストがデータベースにセキュアに伝播されます。レポート、パッチ適用、メンテナンス、またはその他のバッチ・プログラム用の2層アプリケーションの場合、アプリケーション・ユーザーもデータベースに直接接続して認証を受けることができます。

アプリケーション・セッション

Oracle RASアプリケーション・セッションには、データベース内のエンドユーザーとそのセキュリティ・コンテキストが、セキュアかつ効率的に表示されます。Oracle RASアプリケーション・セッションは軽量です。このセッションにはユーザーのセキュリティ関連のステートしか保持されず、従来のデータベース・セッションとの多対1の対応付けが含まれるためです。Oracle RASアプリケーション・セッションは、多数のエンドユーザー間の重量データベース・セッションでは多重化されます。このため、数十万のエンドユーザーが含まれる3層アプリケーションに適しています。Oracle RASでは、Oracle RASアプリケーション・セッションとデータベース・セッションの結合と分離の概念が導入されています。このため、関連するすべてのデータベース操作にアプリケーション・ユーザーのセキュリティ・コンセプトのみが使用されます（図4を参照）。

認証後の最初のログイン時に、ユーザーIDと関連するロールがカプセル化された、ユーザーのOracle RASアプリケーション・セッションが作成されます。このセッションのライフタイム中は、追加のIDと認可コンテキストをセッションに関連付けることができます。たとえば、ユーザーの組織、操作を開始するアプリケーション、リクエストが出されるネットワークIPアドレスなどです。3層アプリケーション（図5を参照）では、ユーザー・リクエストの処理中に、データベース接続がプールから取得され、Oracle RASユーザー・アプリケーション・セッションがデータベース接続に結合されます。

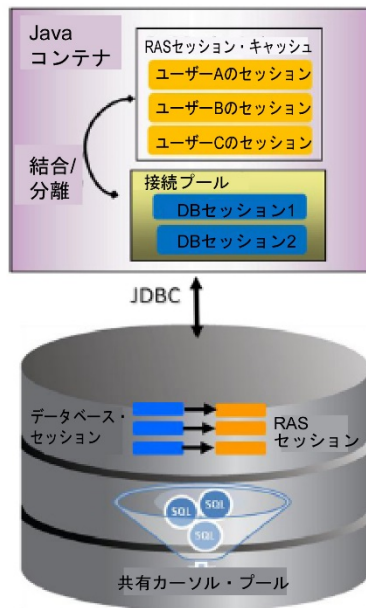


図4：Oracle RASアプリケーション・セッションとDBセッションの結合/分離

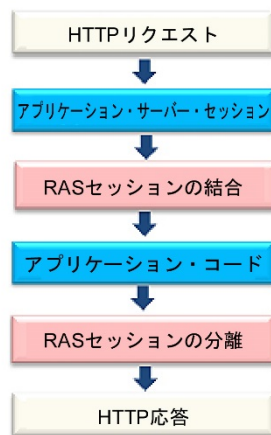



図5：WebアプリケーションでのOracle RASセッションの結合/分離

Oracle RASには、“Servletフィルタ”などのJavaアプリケーション・サーバー・コンポーネントがあります。このため、認証済みのユーザーIDに基づいて、Oracle RASセッションが透過的に作成されます。Java Webコンテナの接続取得コールバック中に、アプリケーション・フレームワーク・コードとOracle RASセッションが結合されます。Oracle RAS Javaコンポーネントによって、実行中のユーザーのOracle RASセッションのみが結合されます。Oracle RASセッションは透過的に作成され、アプリケーション・コードから直接アクセスすることはできません。結合後に、アプリケーション・セッション中のセキュリティ・コンテキストのみを使用して、すべてのデータベース操作が認可されます。Oracle RASセッションでは、データベースへのデータ・チャネルとして、おもに低い権限のデータベース接続が使用されます。また、接続プールに関連付けら



れた基盤データベース・セッションからの権限が使用されることはありません。Oracle RASでは、データベースに対するすべての相互作用が、アプリケーション・ユーザー・セッションに関連付けられた権限に制限されるため、基本的に**最小権限の原則**が実施されます。

Oracle RASでは、次のメカニズムを使用して、アプリケーション・セッションの高いパフォーマンスが維持されます。

- » 軽量なセッション・ステート：アプリケーション・セッションは、コンパイル済みSQL（サーバー側のカーソル）などのデータベース・リソースに関連付けられません（図4を参照）。Oracle RASでは、ユーザー認可関連のステートとセッション・ステートのみが小さいメモリ・フットプリントに保存されるため、多くのアプリケーション・セッションをサポートできます。Oracle RASアプリケーション・セッションによって、コンパイル済みSQLやカーソルが所有されるわけではないので、データベース・セッション間でカーソルを共有できるようになり、カーソルの使用率が上がります。
- » 中間層のキャッシング：アプリケーション・セッションやユーザーの認可ステートは、Java中間層にキャッシュされるため、セッションの結合/分離や認可確認機能のパフォーマンスが向上します。
- » セッション・コンテキストの効率的な送信：Oracle RASセッション関連のデータは、データベースへのアプリケーション・データ・トラフィックと一緒に送信されます。このため、データベースへのラウンドトリップ全体を削減できます。アプリケーション自体でデータベース内にアプリケーション・セッションの概念を構築してデータ・セキュリティを実施する場合は、このようなラウンドトリップが必要になります。
- » 最適なSQL計画：Oracle RASを使用すると、アプリケーション開発者のSQLチューニング作業の負担が軽減され、アプリケーション・セキュリティ・ポリシーを効率的に実施できます。Oracle RASでは、さまざまな内部手法を利用して、保護されたオブジェクトにアクセスする最適なSQL文を生成します。

アプリケーション権限

Oracle RASアプリケーション・セキュリティ・ポリシーによって、アプリケーション固有のビジネス・オブジェクトやエンティティに関するアプリケーションレベルの操作が制御されます。たとえば、人事管理アプリケーションでは、*RequestLeave*や*ApproveLeave*の権限を定義して、従業員の休暇レコードに対するアプリケーションレベルの操作や対応するSQLアクション（選択、挿入、更新）の実行を制御できます。Oracle RASでは、データベースの表やビューの行でアプリケーション権限が認可されているかどうかを確認するためのSQL演算子を使用できます。

同様に、アプリケーション権限を定義して、セルレベルの細かいオブジェクト権限（*ViewSSN*など）を表すことができます。ViewSSNは、ユーザーの従業員レコードを実行するための、SSN列のSELECTを表します。最後に、アプリケーション権限をデータベース以外のオブジェクト（タスク・フローやページ・フローのUI要素など）に関連付けることができます。アプリケーション権限がデータベース・オブジェクトに対するアクションを表さない場合は、Oracle RASのSQL演算子とJava APIを使用して、アプリケーション権限の認可を確認します。このように、Oracle RASアプリケーション開発者は、権限を定義してすべてのアプリケーションレベルの操作に関連付けられたアクションを制御することもできます。

アプリケーション・ロール

アプリケーション・セキュリティ・ポリシーは通常、アプリケーション・ロールを使用して指定します。このとき、アプリケーションの実行中に参加するユーザーはわかっていません。たとえば、*Manager*ロールへの参加者は、休暇のリクエストを承認できます。Oracle RASでは、アプリケーション・ロールを定義して、ACL中のこれらのロールにアプリケーション権限を付与できます。アプリケーション・ロールは、他のアプリケーション・ロールやデータベース・ロールに付与できます。ユーザー・プロビジョニング中は、アプリケーション・ロールがアプリケーション・ユーザーに割り当てられます。

Oracle RASは、次のようなさまざまなロールベースのアクセス制御ポリシーをサポートしています。

- » **職務の分離**：企業は、ユーザーが休暇リクエストの開始と承認の両方を実行できないようにするなどのルールを実施できます。
- » **時間ベースの制約**：従業員は、11月中だけ株式購入計画に登録できます。
- » **委任の制約**：マネージャー・ロールは、*AdministrativeAssistant*ロールを持つユーザーに対して、期間限定で特定の業務のみを委任できます。
- » **ステートベースの制約**：Oracle RASは、アプリケーション・ユーザーに対し、そのランタイム・ステートに基づいて、ロール経由で権限を制約するメカニズムをサポートしています。たとえば、企業のファイアウォール内から接続するアプリケーション・ユーザーに対して、“inside the firewall”というロールを有効にできます。これらのユーザーには、ファイアウォール外から接続するユーザーより多くの権限が付与されます。
- » **コードベースの権限の昇格**：Oracle RASは、ロール経由での、アプリケーション・コードへの権限の対応付けをサポートしています。データベース内で重要なPL/SQLプロシージャが実行される場合や、重要なJavaプログラムが中間層で実行される場合は、ロールをいつでも有効にできます。つまり、重要な操作を実行する場合には権限の昇格が可能です。たとえば、payrollユーザーはpayrollの実行を開始できます。この場合、プログラムは税額を計算するため従業員の給与データにアクセスする必要がありますが、ユーザー自体にはこのようなセキュリティ上の機密データにアクセスする権限はありません。

データ・レلم

データ・レلمでは、保護可能なビジネス・オブジェクトが、アプリケーションの表やビューに、データ行の論理コレクションとして表示されます。たとえば、ビジネス・オブジェクトが組織、部門、地理的な場所に属していたり、その他の対応付けによって関連したりしているデータセットである場合があります。データ・レلمの概念は、アクセス制御要件がビジネス・オブジェクトやビジネス・エンティティを表すデータセットに関連付けられているような、一般的なビジネス・シナリオに適用されます。このコレクションやデータセットは、コレクションの各行が条件を満たす、SQL条件を使用して指定されます。たとえば、データ・レلمは、すべての従業員のレコードであったり、特定のマネージャーにレポートする従業員のレコードであったり、1人の従業員自身のレコードであったりします。図6は、Employee Detailビューの、このようなデータ・レلمを示しています。Nancyがマネージャーだとすると、すべてのレコード、Nancyにレポートする従業員のレコード、Nancy自身の従業員レコードという3つのコレクションがあります。

Name	ID	SSN	Salary	Manager	Phone Number
Steven	SKING	100-51-4567	24000	-	515.123.4567
Neena	NKCOCHHAR	101-51-4568	17000	Steven	515.123.4568
Nancy	NGREENBE	108-51-4569	12008	Neena	515.124.4569
John	JCHEN	110-51-4269	8200	Nancy	515.124.4269
Luis	LPOPP	113-51-4567	6900	Nancy	515.124.1111

すべての従業員レコード

Nancyのレコード

Nancyにレポートする従業員

図6：Employee Detailレコードのデータ・レلمのサンプル

通常、データ・レلمはSQL条件で示され、複数の表の結合が必要な場合があります。オラクルは、さまざまなOracleアプリケーションのセキュリティ・ポリシー・モデリングの経験に基づき、もっとも汎用的なデータ・アクセス・パターンを特定し、次のような数種類のデータ・レلمを提供して、ポリシーを簡単に指定できるようにしています。

- » **セッション属性ベースのレلم**：保護対象の行は、セッション属性やアプリケーション固有のコンテキスト（セッション中のユーザー名やアプリケーションのテナントIDなど）に基づいて選択できます。HRアプリケーションの場合、従業員は自分の連絡先情報を変更できます。同様に、部長は自分の組織内の従業員の給与を表示できます。このような場合、データ・レلمは実行するユーザーのIDコンテキストに依存する動的な行セットを表します。Oracle RASには、Oracle RASセッション・コンテキストとアプリケーション定義のセッション・コンテキストにアクセスするためのSQL演算子があります。
- » **リレーショナル・レلم**：データ・レلمの行は、他の表との結合条件に基づく場合もあります。たとえばデータ・レلمが、問合せを発行するユーザーに直接的/間接的にレポートするすべての従業員を表す場合があります。この場合、Employee表の行は、管理階層に基づいて選択されます。
- » **マスター・ディテール・レلم**：マスター・ディテールは、レコードとその明細項目を表す一般的なデータ・モデリング・パターンです。たとえば、“Employee Leave”と“Leave Detail”の表で、Detail表の休暇リクエストとその明細項目のアクセス制御ポリシーを同一にすることができます。同様に、EmployeeレコードとそのJob History明細項目の行を、1つの論理レコードとして保護できます。
- » **パラメータ化レلم**：パラメータ化データ・レلمは、SQL条件のパラメータ化された条件に基づく、さまざまな行セットを表します。たとえば、東部の営業マネージャーは東部の顧客レコードに、西部の営業マネージャーは西部の販売レコードにアクセスできます。この場合、地域がパラメータ化されて、地域固有のレコードへのアクセス権が、地域固有の異なるマネージャーに付与されます。
- » **レلمの例外**：データ・レلمをモデル化した場合、既存ポリシーの例外が必要となるレコードが存在する場合があります。たとえば、契約社員による、特定の従業員レコードへの一時的なアクセスが必要な場合があります。

アクセス制御リスト

Oracle RASのACLは権限付与やアクセス制御エントリ（ACE）のコレクションであり、ACEによってユーザーやロールへの権限の付与/拒否が行われます。Oracle RASでは、ACLに権限付与がまとめられているため、認可の管理が簡単です。Oracle RASのACLでは、高度なセキュリティ・ポリシー機能を使用できます。たとえば権限の剥奪、複数のオブジェクトやデータ・レلمに対する同じ権限の付与、権限付与リストに対するさまざまな制約（特権操作の実行に複数のロールを必要とするなど）です。

データ・セキュリティ・ポリシー

データ・セキュリティ・ポリシーによって、各データ・レلمとACLが関連付けられます。したがって、データ・セキュリティ・ポリシーは基本的に、データ・レلمとその関連付けられたACLのコレクションです。またOracle RASデータ・セキュリティ・ポリシーは、追加の権限確認に基づいて特定の列の値をマスキングする、列レベルの認可もサポートしています。Oracle RASデータ・セキュリティでは、行レベルと列レベルの認可が行われるため、セルレベルの保護が可能です。

列レベルの認可では、セキュリティ上重要な列がOracle RASアプリケーション権限に関連付けられます。列が権限と関連付けられると、権限が付与されている場合にのみ、行の列値にアクセスできます。図7では、まずセキュリティが重要なSSN列とSALARY列が、ViewSSNとViewSalaryの権限に関連付けられます。次に、図6のように、ACLがデータ・レلمに関連付けられます。“Nancyのレポートのデータ・レلم”の行では、マネージャーのNancyが自分の部下の給与を表示できます。Nancy自身の従業員レコードのデータ・レلمでは、Nancyが自分のSSNと給与を表示できます。最後に、権限に関連付けられていない残りの列（SSNと給与以外）は、“すべての従業員”のデータ・レلمに関連付けられたACLに基づいて、すべての従業員が表示できます。Oracle RASにはSQL問合せ結果セットに列インジケータがあり、セルへのアクセスが認可されていない場合はその旨が表示されます。アプリケーションでこのインジケータを使用して、列値をマスキングしたり、固定値を適切な形式で表示したりすることができます。

Name	ID	SSN	Salary	Manager	Phone Num
Steven	SKING			-	515.123.4567
Neena	NIKCHHAR			Steven	515.123.4568
Nancy	NGREENBE	108-51-4569	12008	Neena	515.124.4569
John	JCHEN		8200	Nancy	515.124.4269
Luis	LPOPP		6900	Nancy	515.124.1111

ViewSSN ViewSalary

ACL: 従業員は自分のSSN、給与を表示できる
 ACL: マネージャーは給与を表示できる
 ACL: すべての従業員は、残りの列のセルを表示できる

図7: ACLを使用したデータ・レلمと列の認可

Oracle RAS認可サービス

Oracle RASデータ・セキュリティがデータベースの表やビューに適用されると、オブジェクトに対するすべてのアクセス・パスのすべてのSQL文で、アクセス制御ポリシーが実施されます。このとき、アクセス経路が2層クライアント・サーバーでも3層アプリケーションでも関係ありません。Oracle RASでは、データに関するアプリケーション操作固有の認可をサポートするだけでなく、次の2つの追加機能のためのSQL演算子を使用できます。

- SQL演算子ORA_CHECK_ACLを使用すると、アプリケーション権限のコンテキストでSQL文が実行されます。たとえばユーザーは、自分がApproveLeave権限を持っている表の休暇申請レコードを更新できます。
- SQL演算子ORA_GET_ACLIDSを使用すると、データ・セキュリティで保護されている表やビューの各行について、ACLへの問合せが実行されます。アプリケーション開発者は、これらのOracle RAS APIを使用して、行で特定の権限が認可されているかどうかを特定できます。たとえば、これらのAPIを使用して、認可された従業員が自分の電話番号を編集できるようにするためのボタンを表示できます。

またアプリケーションで、一連の機能レベルのセキュリティ要件を指定、実施することが必要な場合もあります。たとえば、Webアプリケーションのタスク・フローやページ・フローに必要なセキュリティ要件です。例には、簡単な操作確認が含まれます。たとえば、メニュー項目を呼び出す権限やページ・ナビゲーション用の矢印のクリックです。Oracle RASの認可APIを使用して、データベース外部のこのような保護されたリソースに対するアクセスを決定できます。Oracle RASがない場合、このような機能を利用するには、アクセス制御ポリシーとその実施のカスタム・コーディングが必要となります。

Oracle RASセキュリティ・ポリシーのライフ・サイクル

Oracle RAS認可ポリシーは、アプリケーション開発フェーズ中に定義され、アプリケーションと一緒に展開され、アプリケーションのライフ・サイクルの間管理されます。図8は、Oracle RASセキュリティ・ポリシーの管理のために実行されるタスクを示しています。

アプリケーションの設計時に、権限が必要なすべての操作をアーキテクトが特定します。アプリケーション表の設計とセキュリティの要件に基づいて、データ・セキュリティによる保護が必要な表とビューが特定され、列の保護を含むデータ・レلمが定義されます。次に、アーキテクトが一連のアプリケーション・ロールを作成し、データ・セキュリティ・ポリシーや機能セキュリティで使用されるACL中のロールにアプリケーション権限を割り当てます。

Oracle RASでは、接続プールがアプリケーションによってデータベースの接続用に使用されている場合、権限を持たないユーザーを使用するだけで済みます。Oracle RASの中間層のJavaコンポーネントでは、ユーザーがアプリケーションにログインすると、そのユーザーに対応するOracle RASセッションが作成されて、中間層にキャッシュされます。Oracle RASセッションを使用するには、アプリケーション・コードで、接続

の取得/リリース・コールバックで、Oracle RASの結合/分離セッション・サービスAPIを呼び出す必要があります。Oracle RASセッション・サービスAPIを使用すると、エンドユーザーに対応する正しいOracle RASセッションのみをデータベース接続に結合することができます。また、アプリケーションでOracle RASを認可エンジンとして使用して、中間層のアクセス権を評価し、メニュー項目とコンテンツを表示することもできます。

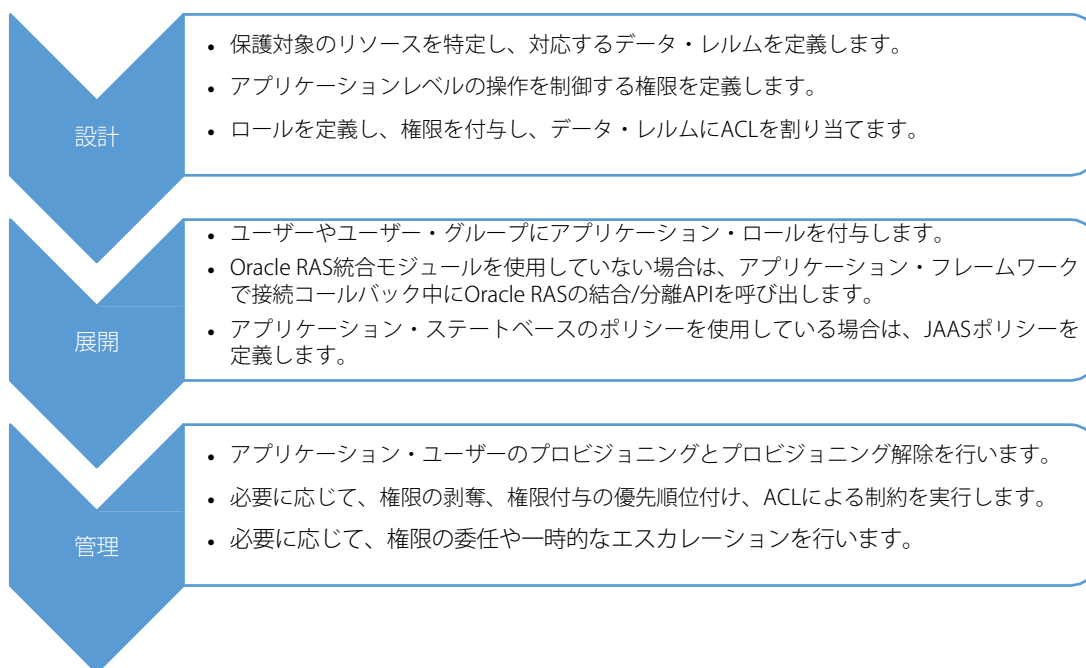


図8：Oracle RASポリシー管理の段階

アプリケーションで表示されるアプリケーション・ロールはほとんどのユースケースに対応しており、通常は企業のユーザーやユーザー・グループにマッピングして、ロール割当てを簡素化できます。場合によっては、管理者が新しいロールを特定して、追加のデータ・セキュリティや機能セキュリティに使用します。また、管理者がACLをカスタマイズして、特定の時間ベースの制約に対応したり、権限の剥奪を使用して例外をサポートしたりすることもできます。

Oracle Real Application Securityの使用例

この例では、このホワイト・ペーパーで最初に説明したHRアプリケーションの一部の要件に基づいて、Oracle RASのセキュリティ・フレームワークについて説明します。このサンプル・アプリケーションでは、各従業員に関する情報は、次の定義のHRMスキーマに基づき、EMPLOYEES表とMANAGERS表に保存されています。

```
EMPLOYEES (EMPLOYEE_ID, NAME, SSN, SALARY, PHONE_NO)
```

```
MANAGERS (MANAGER_ID, EMPLOYEE_ID)
```

図9は、セキュリティ・ポリシーのない従業員レコードのサンプルを示しています。

Name	Manager	Phone Number	SSN	Salary
Steven King	-	515.123.4567	100-51-4567	24000
Neena Kochhar	Steven King	515.123.4568	101-51-4568	17000
Nancy Greenberg	Neena Kochhar	515.124.4569	108-51-4569	12008
John Chen	Nancy Greenberg	515.124.4269	110-51-4269	8200
Luis Popp	Nancy Greenberg	515.124.1111	113-51-4567	6900

図9：HRアプリケーションの従業員レコードのサンプル

この例で実施するアクセス制御ポリシーは、次のとおりです。

1. 従業員は、全員の名前、マネージャー、電話番号を表示できる。
2. 従業員は、自分の社会保障番号（SSN）と給与のみを表示できる。従業員は、自分の電話番号を更新できる。
3. HR担当者は、全従業員のSSNを表示できる。
4. マネージャーは、自分の直接的/間接的な部下の給与を表示できる。

Oracle RASを使用したデータ・セキュリティ

Oracle RASには、Oracle RASポリシーを管理するためのPL/SQL管理APIとデータ・ディクショナリ・ビューがあります。これらの機能に基づいて、Oracle Application Expressを使用してポリシーを管理するOracle RAS管理ツールを開発しました。このホワイト・ペーパーでは、以後このツールのスナップショットを使用して、サンプルHRアプリケーションのデータ・セキュリティ・ポリシーを説明します。

上記の4つのポリシー要件では、従業員、マネージャー、HR担当者という3種類のユーザーと、“全従業員のレコード”、“従業員個人のレコード”、および“マネージャーにレポートする従業員のレコード”という3セットの従業員レコードがあります。これらのレコードのうち、SALARYとSSNの列のセキュリティが重要です。

これらの情報に基づき、まずEMPLOYEE、MANAGER、HRREPという3つのアプリケーション・ロールを定義します。次に、VIEW_SALARYとVIEW_SSNという2つの権限を定義します。これらの権限は、データ・セキュリティ・ポリシーの列認可の一部として、対応するSALARY列とSSN列に関連付けられます（図10を参照）。続いて、次の3つのデータ・レلمを定義します。

1. ALL_RECORDS：SQL条件“1=1”を使用して表示されます。この条件は、表のすべての行に適用されます。
2. MY_RECORD：実行ユーザーのレコードは、Oracle RASセッション・コンテキスト確認演算子を使用して、次の条件に基づいて特定されます。次のように、演算子によってセッションのログオン・ユーザーIDが戻されます。
EMPLOYEE_ID= XS_SYS_CONTEXT('XS\$SESSION', 'USERNAME')
3. MY_REPORTS：マネージャーにレポートするすべての従業員は、Manager表の階層的な問合せCONNECT BYに基づいて検索されます。

図10の右側に示すとおり、これらのデータ・レلمの適切な権限は、ACL経由でロールに付与されます。すべての従業員レコードで、EmployeeロールにSELECT権限が、HRREPロールにVIEW_SSN権限が付与されます。SSNとSALARYの列は権限に関連付けられているため、SELECT権限によって付与されるのは、Employee表の残りの列へのアクセス権だけです。同様に、従業員個人のレコードでは、EmployeeロールにVIEW_SSN、VIEW_SALARY、UPDATEの権限が付与されます。これで、従業員は自分のレコードのセキュリティが重要な

列の表示と、電話番号の更新を行うことができます。最後に、Managerロールに対して、自分の部下のSALARY値に関するVIEW_SALARY権限が付与されます。

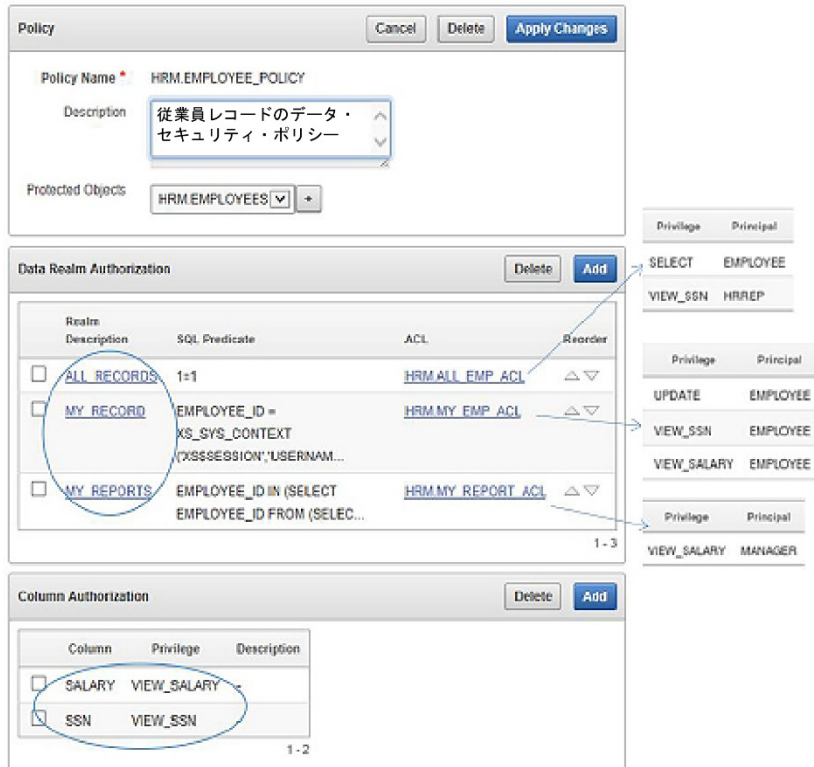


図10：Employees表のデータ・セキュリティ・ポリシー

図11は、図10のデータ・レルムで付与された権限を、ロール中心で表示したものです。これらの権限付与は、サンプルHRアプリケーションの4つの認可要件を表しています。

ロール	説明	権限	オブジェクト	データ・レルム
<u>EMPLOYEE</u>	一般従業員	SELECT	HRM.EMPLOYEES	ALL_RECORDS
<u>EMPLOYEE</u>	一般従業員	UPDATE	HRM.EMPLOYEES	MY_RECORD
<u>EMPLOYEE</u>	一般従業員	VIEW_SALARY	HRM.EMPLOYEES	MY_RECORD
<u>EMPLOYEE</u>	一般従業員	VIEW_SSN	HRM.EMPLOYEES	MY_RECORD
<u>HRREP</u>	HR担当者	VIEW_SSN	HRM.EMPLOYEES	ALL_RECORDS
<u>MANAGER</u>	マネージャー	VIEW_SALARY	HRM.EMPLOYEES	MY_REPORTS

図11：データ・レルムでのロールと権限の付与

このアプリケーションのプロビジョニング中に、EmployeeとManagerのロールをNancyに割り当てます。図12は、Nancyが表示できるセルの値を示しています。このレポートでは、Nancyが自分のレコードのすべての列、自分の部下の給与（図11のJohnとLuis）、およびその他のすべての従業員の公開情報を表示できます。

ただし、他の従業員のSSNや、部下ではない従業員の給与は表示できません。Oracle RAS列のマスキング演算子を使用して、認可されていないSSNセル値は“111-11-1111”で、認可されていないSALARYセル値は“xxxxxx”でマスキングされます。

Name	Manager	Phone Number	SSN	Salary
Steven King	-	515.123.4567	111-11-1111	xxxxxx
Neena Kochhar	Steven King	515.123.4568	111-11-1111	xxxxxx
Nancy Greenberg	Neena Kochhar	515.124.4569	108-51-4569	12008
John Chen	Nancy Greenberg	515.124.4269	111-11-1111	8200
Luis Popp	Nancy Greenberg	515.124.1111	111-11-1111	6900

図12：Oracle RASデータ・セキュリティによる、Employee表でのNancyの問合せ結果

アプリケーション開発プラットフォームとの統合

Oracle RAS導入のための開発作業を減らすため、Oracle RASランタイム・セッションは、Fusion Middleware 12.1.3に付属しているOracle Application Express (Oracle APEX 5.0) およびOracle Platform Security Services (OPSS) と統合されています。以下で、このすぐに使用可能な2種類のOracle RASセッション統合モジュールについて説明します。同じ方法が、Oracle RASのJava APIとPL/SQL APIを使用した、Oracle RASセッション統合のアプリケーション・プラットフォーム固有の開発でも使用できます。Oracle RASでは、外部のIDストアが定義したエンドユーザーが認識されるためです。

Java EEのネイティブ統合

Oracle RASは、OPSSがそのアプリケーション・セキュリティ・プロバイダとしてサポートしているJava Enterprise Edition (Java EE) Webアプリケーションと統合されています。この統合では、Oracle RASがOPSSと一緒にJava EE Webアプリケーションで展開されます。OPSSでは、Webユーザーのコンテナ認証コンテキストに基づき、ユーザーのアプリケーション・ロールと認可属性を使用して、アプリケーション・セキュリティ・コンテキストを計算します。このコンテキストがOracle RASによって拡張、使用され、すべての認可が決定されます。図13は、OPSSによるOracle RASのランタイム展開を示しています。

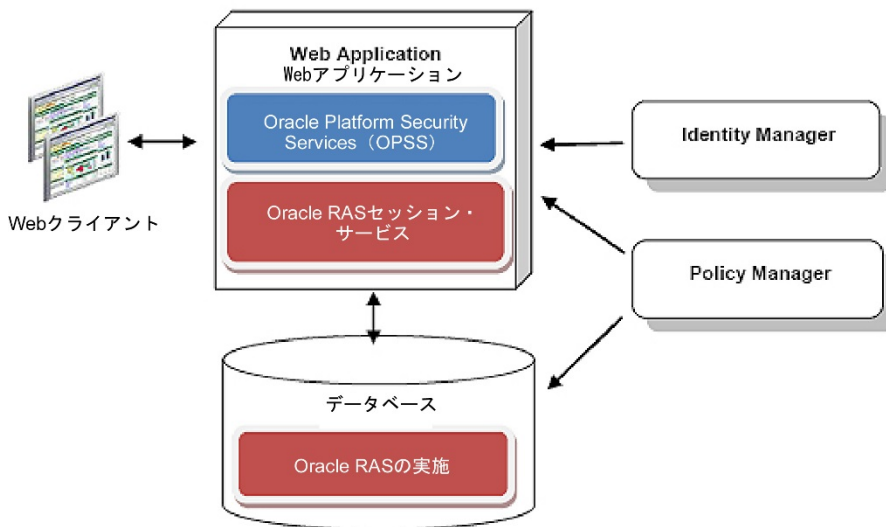


図13：Oracle RASとJava EEを使用したWebアプリケーションの展開

Oracle APEXとのネイティブ統合

Oracle Application Expressは、宣言的なデータベース中心の2層Webアプリケーションを迅速に構築するための、Webブラウザベースの開発ツールです。Oracle APEXでは、データ・アクセス制御機能が直接提供されるわけではないので、Oracle APEXの開発者が、アクセス制御アーチファクト（アプリケーションの権限、ロール、データに関する細かいアクセス制御、アプリケーション定義のビジネス・オブジェクトなど）を実装することになります。Oracle RASとOracle APEX 5.0の統合により、Oracle APEXアプリケーションのより高度なアクセス制御機能を実現しています。このため、Oracle RASセッションが、Oracle APEXアプリケーション開発フレームワーク内で、透過的かつネイティブに作成、管理されます（図14を参照）。このフレームワーク内では、データベース中のOracle APEXアプリケーション・コードが、Oracle RASセッション・コンテキスト内で実行されます。Oracle APEXアプリケーションの開発者は、データへのアクセス制御を実施するために、Oracle RASポリシー実施用のアプリケーション・ランタイム・コードを記述する必要はありません。Oracle RASアクセス制御ポリシーに基づいてUI項目をレンダリングするには、Oracle APEXの宣言的認可ルールで、Oracle RAS権限確認演算子を使用できます。

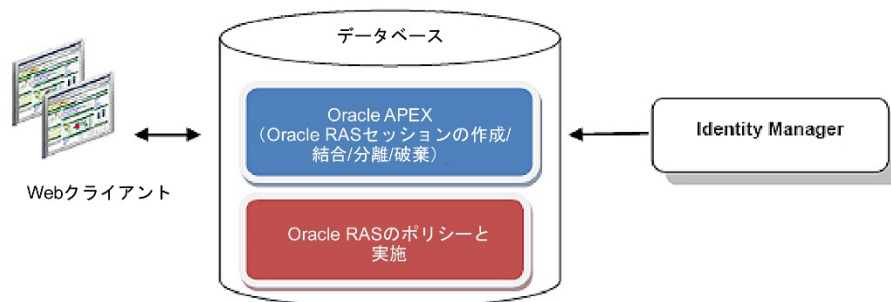


図14：Oracle RASを使用したOracle APEXアプリケーション

Oracle RASの拡張機能

Oracle RASを使用すると、次のようなさまざまなアプリケーションのアクセス制御要件を実施できます。

アプリケーション要件	必要なセキュリティ機能	Oracle RASのサポート
部長がアシスタントに給与管理を委任したい。	特定のタスクの委任のサポート	アプリケーション・ロールが、委任と時間ベースの制限をサポートしています。
契約社員が、特定の従業員レコードへの一時的なアクセスを必要としている。	既存ポリシーの例外	Oracle RASは、表の各行の行単位のACLをサポートしています。行単位のACLは既存ポリシーより優先されるため、より高レベルなポリシーを追加します。
プロジェクトの最後に、プロジェクト所有者に付与される権限を呼び出す必要がある。	ユーザー固有の時間ベースの権限の呼出し	Oracle RASは、ユーザーに対する時間ベースの制約をサポートしています。ユーザーに対して、時間ベースのロールと権限の付与を行うことができます。
従業員が所定の登録期間のみ、株式購入計画に登録できる。	ロール固有の時間ベースの認可	Oracle RASは、ロールに対する時間ベースの制約をサポートしています。すべての従業員に <i>stock purchase plan</i> というロールを付与し、所定の期間のみ有効にすることができます。
マネージャーは昇進の申請も承認もできない。	職務の分離	Oracle RASロールとレムル条件を組み合わせ、さまざまなロールベースの制約を表すことができます。
寄贈者権限を持つ部長だけが、企業献金を承認できる。	ロールの組合せによる制約	Oracle RASのACLは、操作の実行には同一の権限を複数のロールに付与する必要があるという条件を実施する追加のACLによって、制約をかけることができます。
機密性の高い給与データは、企業のファイアウォール内からしか表示できない	ステートベースの認可	ステートベースのロールを使用して、追加のアクセス制御を実施できます。
Compensation Portalには、マネージャーしかアクセスできない。	アプリケーション定義の権限による操作の保護	Oracle RASは、このようなACLを使用したアプリケーション固有の操作に関するアクセス制御をサポートしています。
自宅住所を機密情報として扱うための新しい要件が必要である。	行と列両方の、拡張可能なアクセス制御フレームワーク	Oracle RASは、ACLおよびその他のアクセス制御ポリシー・コンポーネントの新しい権限と変更の宣言をサポートしています。
企業ファイアウォールの外にいるユーザーや、強力な認証が適用されていないユーザーに対して、HRアプリケーションとERPアプリケーションで共通のポリシーを実施する必要がある。	共通の認可ポリシー	Oracle RASは、展開されているすべてのアプリケーションで権限を付与または拒否できる、システム全体のACLをサポートしています。
エンドユーザーのアクティビティの監査。	アプリケーション・ユーザーのアクティビティの監査	アプリケーション・セッションがデータベース内でネイティブに統合されているため、データベース監査ログでエンドユーザーの詳細を監査できます。

Oracle Real Application Security (Oracle RAS) の利点

Oracle Database 12cには、Real Application Securityによる、アプリケーション用の次世代の認可アーキテクチャが搭載されています。

- » **エンドユーザーIDのセキュアな伝播**：アプリケーション・セッションで、エンドユーザーIDと関連属性をデータベースにセキュアに送信できるため、データベースでエンドユーザーのアクセス制御や監査の情報を使用できます。
- » **同一のデータ・セキュリティ**：Oracle RASセキュリティ・モデルでは、アクセス・パスに関係なく、ビジネス・オブジェクトに対するアクセス制御ポリシーの仕様と実施が共通です。このため、アクセス制御ロジックが組み込まれた特定のコード・パス経由でオブジェクトへのアクセスが行われた場合にのみ機能する、カスタムビルド・アプローチの制限事項を克服することができます。
- » **宣言的な細かいアクセス制御**：Oracle RASポリシー・コンポーネントでは、アプリケーションのアクセス制御要件が、アプリケーション・ユーザー、アプリケーション・ロール、アプリケーション権限に関する宣言的ポリシーの形式でカプセル化されます。Oracle RASモデルでは、列のセキュリティによって、認可が列レベルまで拡張されるため、SSNなどの機密データを保護できます。Oracle RASはマスター・ディテール、パラメータ化、委任、および例外ベースの宣言的ポリシーをサポートしているため、アプリケーションの実用的な展開要件に対応できます。
- » **パフォーマンスの低下なしのセキュリティ**：現在のほとんどのシステムでは、セキュリティがアプリケーション内にコーディングされているか、外部化されていてもパフォーマンスに影響する複数のラウンドトリップが必要です。Oracle RASはこれらとは違って、データベース内にネイティブで実装された、パフォーマンスの低下なしのセキュリティ・ソリューションです。

このホワイト・ペーパーでは3層アプリケーションについて重点的に説明しましたが、Oracle RASは、データにアクセスするすべてのアプリケーション（スタンドアロンのクライアント・サーバー・アプリケーションを含む）を保護します。アプリケーションで、データベース内の独自のアクセス制御ポリシー・インフラストラクチャを開発する必要はありません。アクセス制御ポリシーの管理はプログラム・コードから切り離され、柔軟性と拡張性が向上しています。

最後に、Oracle RASでは、データベース内でアプリケーション固有の権限、ユーザー、ロールを定義および使用できるため、データベースのアプリケーション固有のアクセス制御モデルが統合されています。Oracle RASでは、データベース内に長期間必要なアプリケーション認可機能があり、データに対するアクセス制御ポリシーについて共通の管理モデルを使用できます。図15は、アプリケーションの保護に関心があるさまざまなユーザーがOracle RASを使用する利点を示しています。

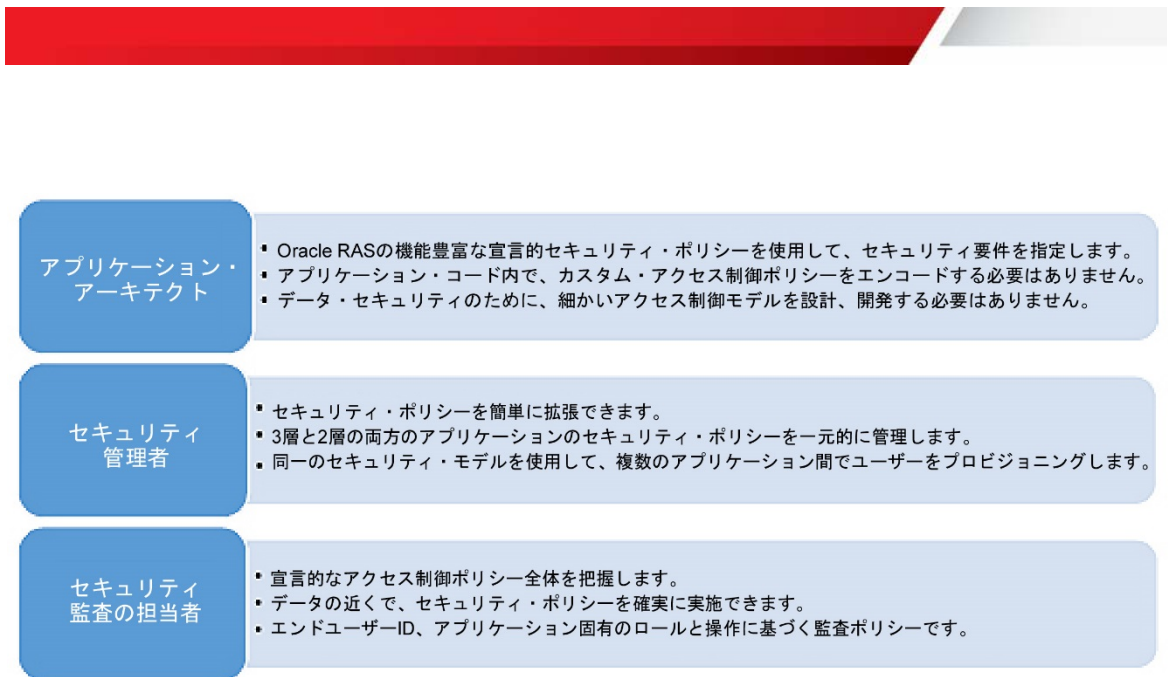


図15：各種ユーザーにとってのOracle RASセキュリティの利点

Oracle RASでは、パッケージ・アプリケーションの構築におけるオラクルの経験が生かされており、セキュリティ、スケーラビリティ、パフォーマンス、管理性を考慮しながら、各種モジュールの認可のさまざまなユースケースをサポートできるように設計されています。

その他

本書では、Oracle Real Application Securityが開発された理由と概念について説明してきました。Oracle RAS 開発フレームワークには、ポリシー管理用のPL/SQL APIと、3層アプリケーションと2層アプリケーションでOracle RASセキュリティ・ポリシーを実施するためのJava APIとPL/SQL APIが含まれています。詳しくは、『Oracle Real Application Security Architecture』のホワイト・ペーパーと開発者ガイドを参照してください。



Oracle Corporation, World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065, USA

著者：Michael Ramchand、Peter Wilson、Martien Ouwens

海外からのお問い合わせ窓口

電話：+1.650.506.7000

ファクシミリ：+1.650.506.7200

Hardware and Software, Engineered to Work Together

CONNECT WITH US



blogs.oracle.com/soa



facebook.com/oraclesoa



twitter.com/oraclesoa



oracle.com/soa

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0914



Oracle is committed to developing practices and products that help protect the environment