auth0

関係性で制御する新しいアクセス管理を用いたRAGにおける適切なフィルタリング方法とは

池原大然
プリンシパルデベロッパーアドボケイト
Okta Japan株式会社

セーフハーバー

このプレゼンテーションは、1995年私募証券訴訟改革法(Private Securities Litigation Reform Act of 1995)の「セーフハーバー」条項が規定する「将来の見通しに関する記述」を含んでおり、これには当社の財務見通し、事業戦略および計画、市場動向、市場規模、機会および位置づけに関する記述が含まれますがこれらに限定されません。これらの将来の見通しに関する記述は、現在の予想、見積、予測および見通しに基づいています。「期待する」、「予想する」、「はず」、「信じる」、「希望する」、「目標とする」、「見積もる」、「目標」、「推定する」、「可能性」、「予測する」、「可能性がある」、「するつもりである」、「かもしれない」、「あり得る」、「意図する」、「行う予定」、およびこれらの用語のバリエーションや類似の表現は、将来の見通しに関する記述を識別することを目的としていますが、すべての将来の見通しに関する記述にこれらの識別語が含まれているわけではありません。将来の見通しに関する記述は、当社のコントロールを超えたリスクや不確実性の影響を受けます。例えば、マクロ経済の状況は過去にも、また将来的にも当社ソリューションに対する需要を減少させる可能性があること、当社および当社の第三者サービス・プロバイダーは過去にも、また将来的にもサイバーセキュリティ・インシデントに遭遇する可能性があること、収益の成長や収益性を管理または維持できない可能性があること、当社の財源が、市場で効果的に競争するには不十分である可能性があること、新規顧客の獲得や既存顧客の維持や追加販売ができない可能性があること、ソリューションの推進または強化のための戦略的パートナーシップを維持できない可能性があること、既存のマーケティングおよびセールス組織を拡大（go-to-market組織のさらなる専門化を含む。）する上で、課題に直面する可能性があること、顧客数の伸びは最近減速しており、今後も減速する可能性があること、サービス停止を含む当社の技術に関連した中断やパフォーマンスの問題が生じる可能性があること、および当社および当社の第三者サービス・プロバイダーは、当社が従うべき様々なプライバシーおよびセキュリティ規定を完全に遵守できなかった、または遵守できなかったとみなされたことがあり、同様の事故が将来発生する可能性があることなどが挙げられます。当社の業績に影響を与える可能性のある要因に関する詳細は、当社の最新のおよびその他の米国証券取引委員会への提出書類に記載されています。このプレゼンテーションに含まれる将来の見通しに関する記述は、このプレゼンテーションの日時点での当社の見解を示すものであり、当社はこれらの将来の見通しに関する記述を更新する義務を負いません。

このプレゼンテーションで言及されている、現時点で一般に提供されていない、またはまだ取得されていないソリューション、特性、機能、認証、認可、または証明は、予定通りに、または全く提供もしくは取得されない可能性があります。当社は、かかる事項を提供する義務を負わず、お客様は購入の判断を下す上でこれらに依拠すべきではありません。

👋 自己紹介

池原 大然
デベロッパーアドボケイト

auth0

X (Twitter): @Neri78
Bluesky: neri78.bsky.social
LinkedIn: daizenikehara
GitHub: http://github.com/neri78

生成AIのビジネス活用とリスク

auth0

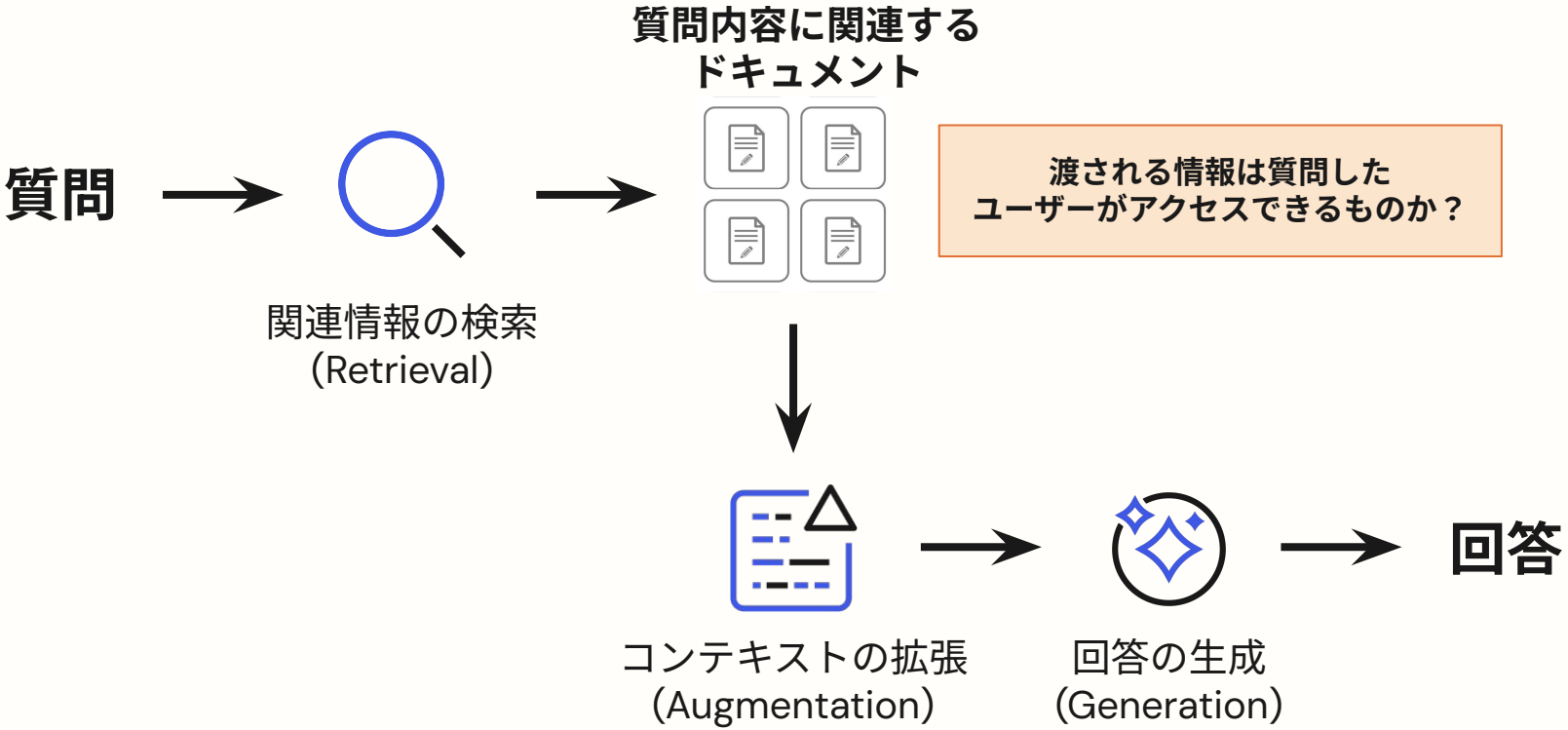
ビジネスに生成AI・LLMを利用するための外部連携

公開データのみから学習した
LLMだけではビジネスには不十分

- ツール実装
- API呼び出し
- RAG
- MCP

さまざまなものを組み合わせる

RAGを用いた回答の作成と課題



OWASP 2025 Top 10 Risk & Mitigations for LLMs and Gen AI Apps

1
プロンプト インジェクション
Prompt Injection

2
機微情報の 漏洩
Sensitive Information Disclosure

3
サプライ チェーン
Supply Chain

4
データやモデル の汚染
Data and Model Poisoning

5
不適切な 出力処理
Improper Output Handling

6
過剰な 代理行為
Excessive Agency

7
システム プロンプト の漏洩
System Prompt Leakage

8
ベクトルと 埋め込みの弱点
Vector and Embedding Weaknesses

9
誤情報
Misinformation

10
無制限な 消費
Unbounded Consumption

https://genai.owasp.org/llm-top-10/

OWASP Top 10 for Agentic Applications for 2026

1
エージェントゴールハイジャック
Agent Goal Hijack

2
ツールの誤用と悪用
Tool Misuse & Exploitation

3
アイデンティティと特権の濫用
Identity & Privilege Abuse

4
エージェント型サプライチェーンの脆弱性
Agentic Supply Chain Vulnerabilities

5
予期しないコード実行（RCE）
Unexpected Code Execution (RCE)

6
メモリとコンテキストのポイズニング
Memory & Context Poisoning

7
安全でないエージェント間通信
Insecure Inter-Agent Communication

8
連鎖的な障害
Cascading Failures

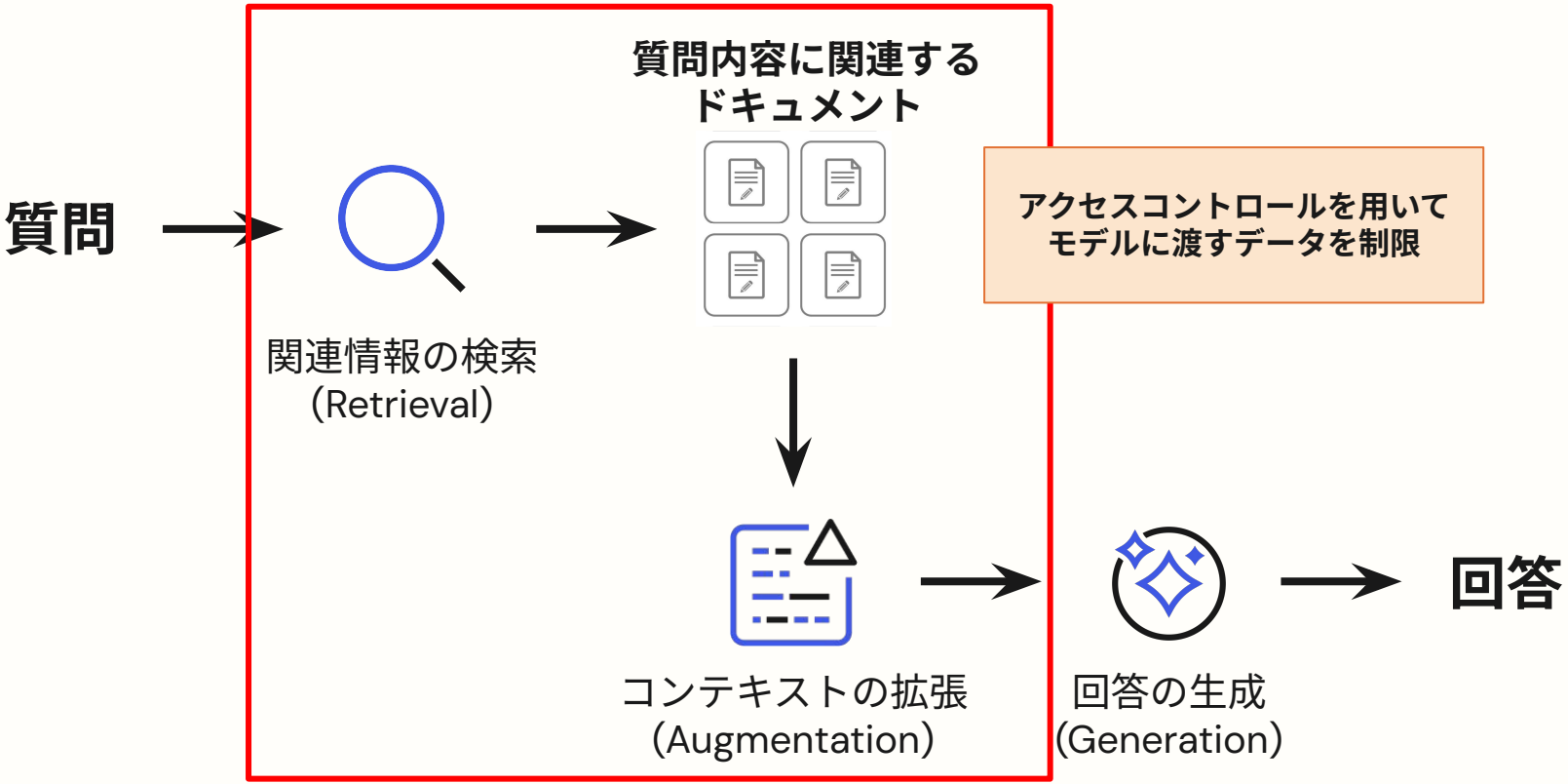
9
人間とエージェントの信頼の悪用
Human-Agent Trust Exploitation

10
ローグエージェント
Rogue Agents

https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

RAGを利用する場合に データをどう保護するか？

適切なフィルタリングが必要

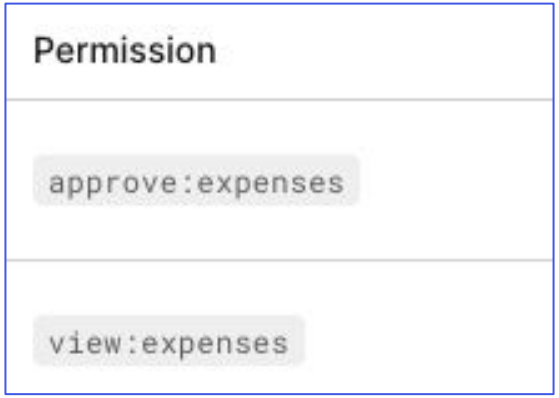


アクセスコントロールの種類

auth0

ロールを基にしたアクセス制御 (Role-Based Access Control, RBAC)

- **グループ/ロール**に割り当てられたユーザー
- **アプリケーション/API**レベルでの**大まか**な権限を管理することに適している




```
app.post('/expense/approve',
  claimIncludes('role', 'approver'),
  (req, res) => { ... });

app.post('/expense/approve',
  claimIncludes('permissions', 'approve:expenses'),
      (req, res) => { ... });
```

属性を利用した細かなアクセス制御 (Attribute Based Access Control, ABAC)

- [属性] マネージャーは **自分のチームの経費のみ**承認
- 時間や場所、言語など複雑な条件を制御できるが **アプリケーションコードが複雑化する**

```
async function approveExpenseABAC(user, expenseId, db) {
  try {
    if (!user.hasPermission("approve:expenses")) {
      throw new Error("認可エラー: ユーザーには 'approve:expenses' 権限がありません。");
    }

    const expense = await db.fetch(
      `SELECT e.*, u.manager_id AS submitter_manager_id
       FROM expenses e
       JOIN users u ON u.user_id = e.submitter_id
       WHERE e.id = ?`,
      [expenseId]
    );

    if (user.id !== expense.submitter_manager_id) {
      throw new Error("認可エラー: ユーザーは経費申請者のマネージャーではありません。");
    }

    // すべてのチェックを通過した場合、承認処理を実行 ...
    console.log(`ユーザー ${user.id} が経費 ${expenseId} の承認に成功しました。`);
  }
}
```

ポリシーエンジンを用いたアクセス制御 (Policy-based Access Control, PBAC)

ポリシーエンジンを実装

- 経費の提出者のマネージャーかつ、承認の権限を持つか

アプリケーションからこの**エンジンを呼び出し**

```
allow(user: User, "approve", expense: Expense) if
    user.user_id == expense.submitter_manager_id and
    user.has_permission("approve.expenses")
```

```
async function approveExpensePBAC(user, expenseId, db, policy) {
  try {
    const expense = await db.fetch(
      `SELECT e.*, u.manager_id AS submitter_manager_id
       FROM expenses e
       JOIN users u ON u.user_id = e.submitter_id
       WHERE e.id = ?`,
      [expenseId]
    );
    // ポリシーエンジンに認可の判断を委ねる
    await policy.authorize(user, "approve", expense);

    // policy.authorize がエラーをスローしなければ、認可されたと判断

    }
}
```

それぞれの特徴

名前	粒度	認可 ロジック	認可 データ
Role-based access control (RBAC)	粗い	集約	集約 （プリンシパル）
Attribute-based access control (ABAC)	細かい	アプリケーション コード	アプリケーション データベース
Policy-based access control (PBAC)	細かい	集約	アプリケーション データベース

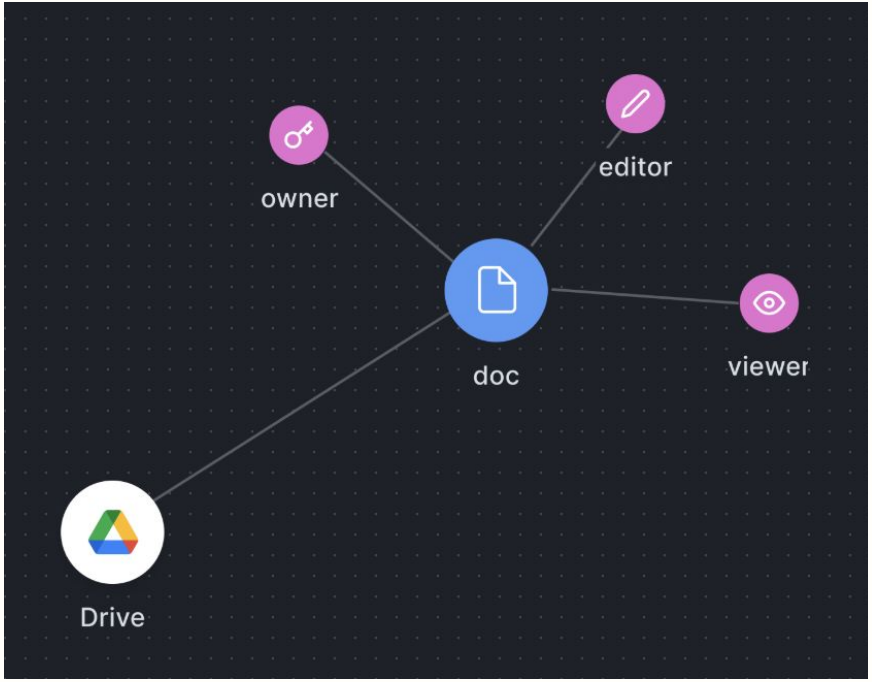
関係性で制御する
アクセスコントロール

auth0

関係性を用いたアクセス制御 (Relationship-based Access Control, ReBAC)

- リソースへのアクセス許可が「関係性」によって決定されるアクセス制御
- **「ユーザーは、アクセスを可能にするほど、指定されたオブジェクトやアクションと十分な関係があるか？」**
  - 直接関係がある
  - 関係性のグラフでつながっている

- 参考: Googleの様々な製品における認可エンジン**「Zanzibar」**
  - **認可の判断となる関係性のデータは認可システムに保存**



Zanzibar: Google's Consistent, Global Authorization System - https://research.google/pubs/zanzibar-googles-consistent-global-authorization-system/

Presented by auth0

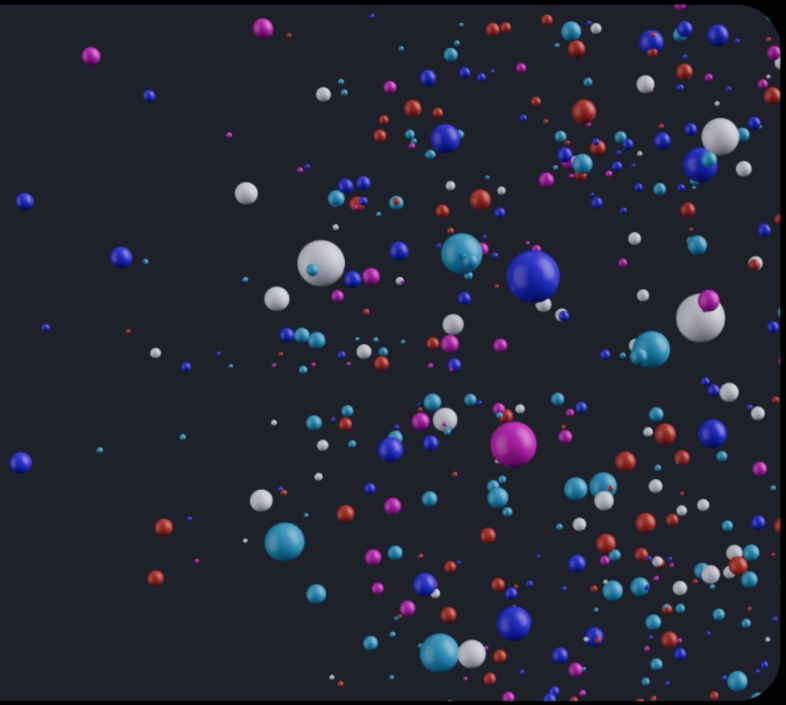
How it works Resources Community

zanzibar

A Globally Distributed Authorization System

Zanzibar handles authorization for YouTube, Drive, Google Cloud and all of Google's other products

See how it works ↓ Read the paper →

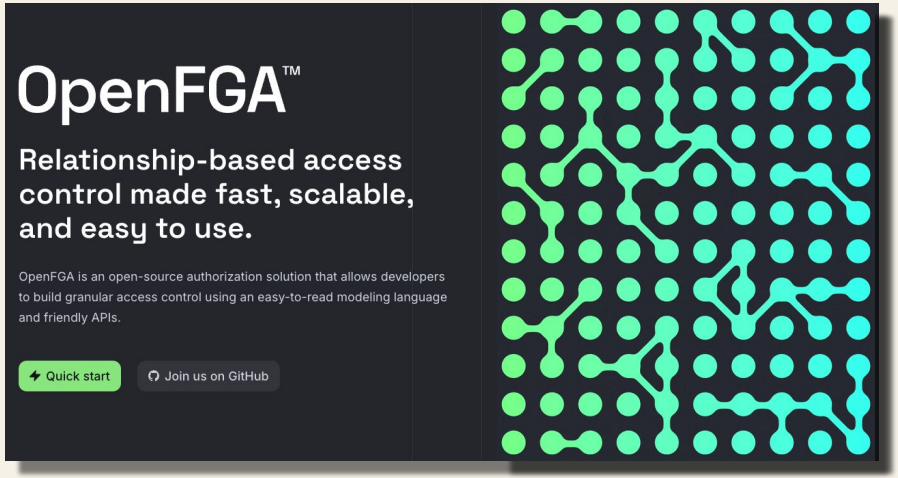


https://zanzibar.academy/

ReBACの実装例

OpenFGA

OpenFGA



openfga.dev

- Oktaによって開発・オープンソース化されたCloud Native Computing Foundation (CNCF) のインキュベーションプロジェクト
- Googleのサービス全体で利用されるきめ細かいアクセス制御 Fine-Grained Authorization (FGA) を実現する「Zanzibar」論文を基に開発
- PostgreSQLやMySQL、SQLiteなど自身で選択したデータベースと組み合わせて利用できる

認可モデル

```
model
  schema 1.1

type user

type doccument
  relations
    define owner: [user]
    define viewer: [user, user:*]
```

関係性タプル

```
user        user:*
relation    viewer
object      document:public-doc
```

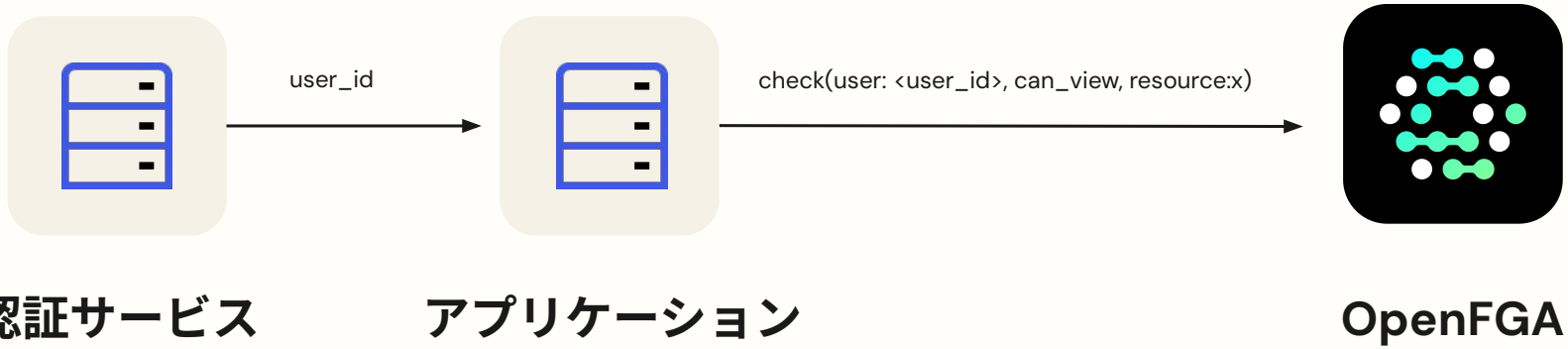
```
user        user:neri78
relation    viewer
object      document:private-doc
```

認可チェック（例）

```
// チェック
const { allowed } = await fgaClient.check({
    user: 'user:neri78',
    relation: 'viewer',
    object: 'document:private-doc',
 });
// allowedにtrue/falseの値が格納される
// 複数のリソースを一度にチェックする batchCheckメソッドも提供
```

Demo

認証とは独立した認可サービスとして利用できる

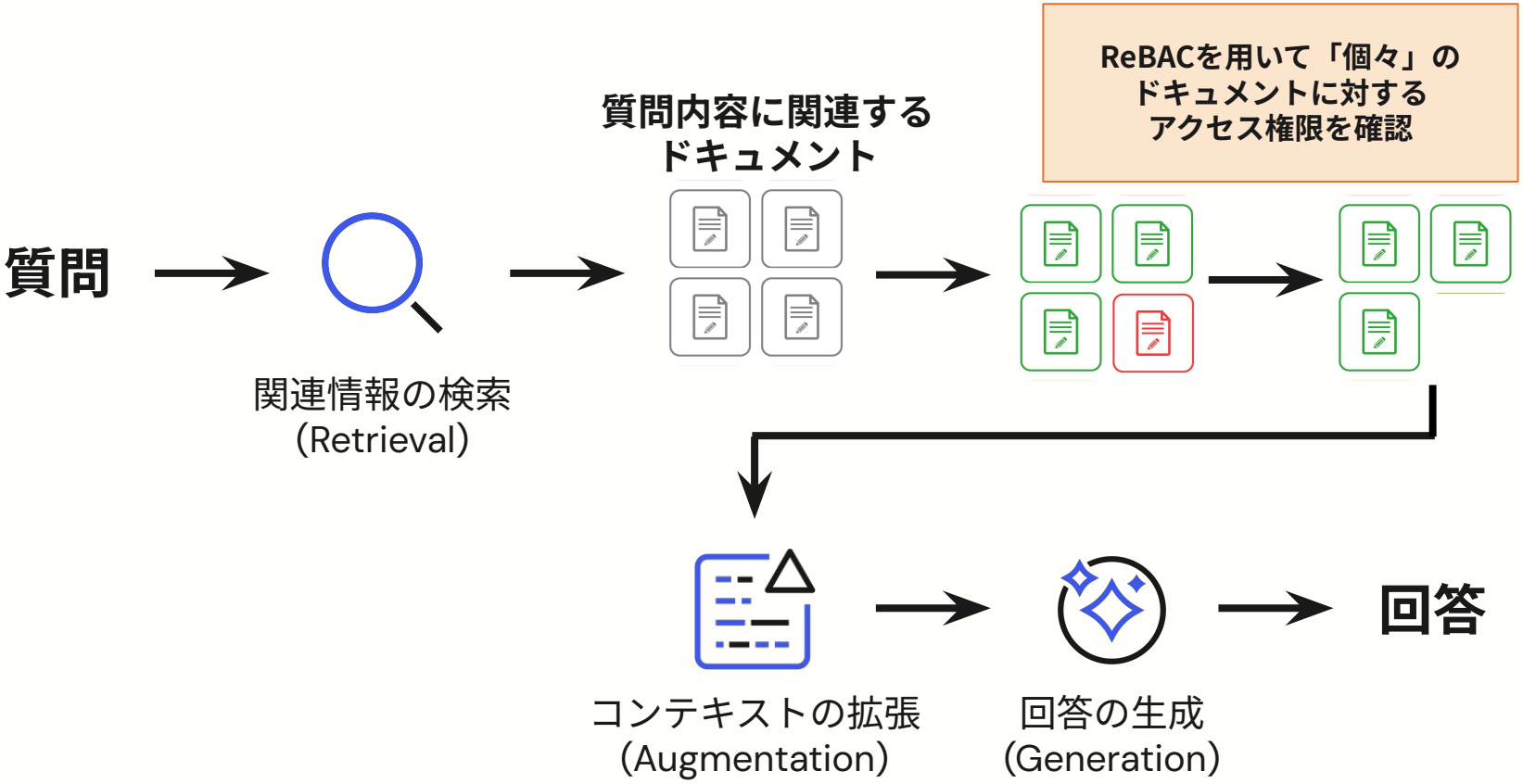


これまでのアクセスコントロール方法との比較

名前	粒度	認可 ロジック	認可 データ
Role-based access control (RBAC)	粗い	集約	集約 （プリンシパル）
Attribute-based access control (ABAC)	細かい	アプリケーション コード	アプリケーション データベース
Policy-based access control (PBAC)	細かい	集約	アプリケーション データベース
Relationship-based access control (ReBAC - Zanzibar based)	**細かい**	**集約**	**集約**

ReBACを利用した RAGのアクセス制御

auth0



Demo

```
    // ベクターストアからすべての結果を取得
    const results = await this.vectorStore.similaritySearch(query, 20);
...
    // 1. FGAからすべての認可されたドキュメント IDを取得
    const response = await this.fgaClient.listObjects({
      user: this.userId,
      relation: 'viewer',
      type: 'document'
    });

    const authorizedDocumentIds = response.objects.map(obj => {
      // オブジェクト IDからドキュメント名を抽出 (例: "document:core_engine" -> "core_engine")
      const parts = obj.split(':');
      return parts.length > 1 ? parts[1] : obj;
    }).filter(id => id && id !== 'undefined');
```

```
    // 2. 認可されたドキュメントに基づいて結果をフィルタリング
    const authorizedDocs = results.filter(doc => {
      const docId = doc.metadata?.doc_id;
      if (!docId) return false;

      // doc_idからドキュメント名を抽出 (例: "document:core_engine" -> "core_engine")
      const documentName = docId.replace('document:', '');

      // このドキュメントが認可リストに含まれているか確認
      return authorizedDocumentIds.includes(documentName);
    });

    return authorizedDocs;
```

まとめ

生成AIのビジネス活用とリスク

RAGを利用する場合に必要なデータ保護

OpenFGA(ReBAC)を用いたRAGの
細やか・かつ中央化された認可処理

再掲: アクセスコントロールの種類と特徴

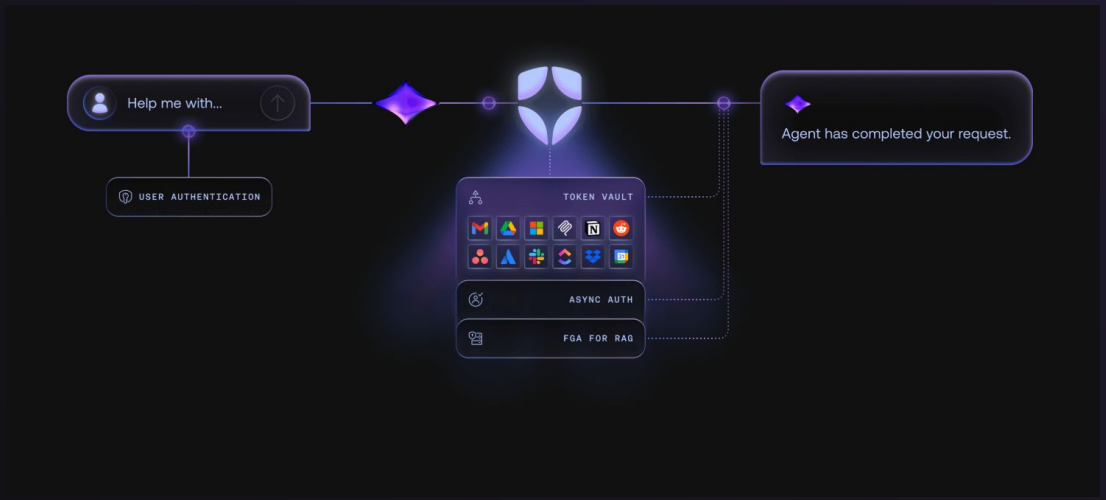
名前	粒度	認可 ロジック	認可 データ
Role-based access control (RBAC)	粗い	集約	集約 （プリンシパル）
Attribute-based access control (ABAC)	細かい	アプリケーション コード	アプリケーション データベース
Policy-based access control (PBAC)	細かい	集約	アプリケーション データベース
Relationship-based access control (ReBAC - Zanzibar based)	細かい	集約	集約

オープンソースとホステッドサービス

openfga.dev

fga.dev

Auth0 for AI Agents – AIアプリケーション向け認証、認可



a0.to/ai-event


auth0
okta — The World's Identity Company™