Progress® MarkLogic®

書類

このユニットのキーコンセプト

- 対応ドキュメント
  - フォーマット
  - 参考
  - 組織
- 推奨される文書の構成
- バルクロード/インジェスト。
- MVCCとデータの修正。
- 一括更新。

専有情報 | ©2024 MarkLogic Corporation は Progress Software Corporation の完全子会社です。無断複製、転載を禁じます。

コンセプトドキュメント

- MarkLogicは、さまざまなタイプのドキュメントコンテンツをサポートしています：
  - 構造化：XMLDocument, JSONDocument
  - 非構造化テキスト文書
  - グラフ：XMLまたはJSONで表現されたRDFトリプル。
  - バイナリバイナリドキュメント
- ドキュメントは "そのまま "読み込むことができる
  - ロード前に定義されたスキーマや構造は必要ない。
  - インデックスとテンプレートは、ロード後に構造とデータ型を強制するために使用することができる。

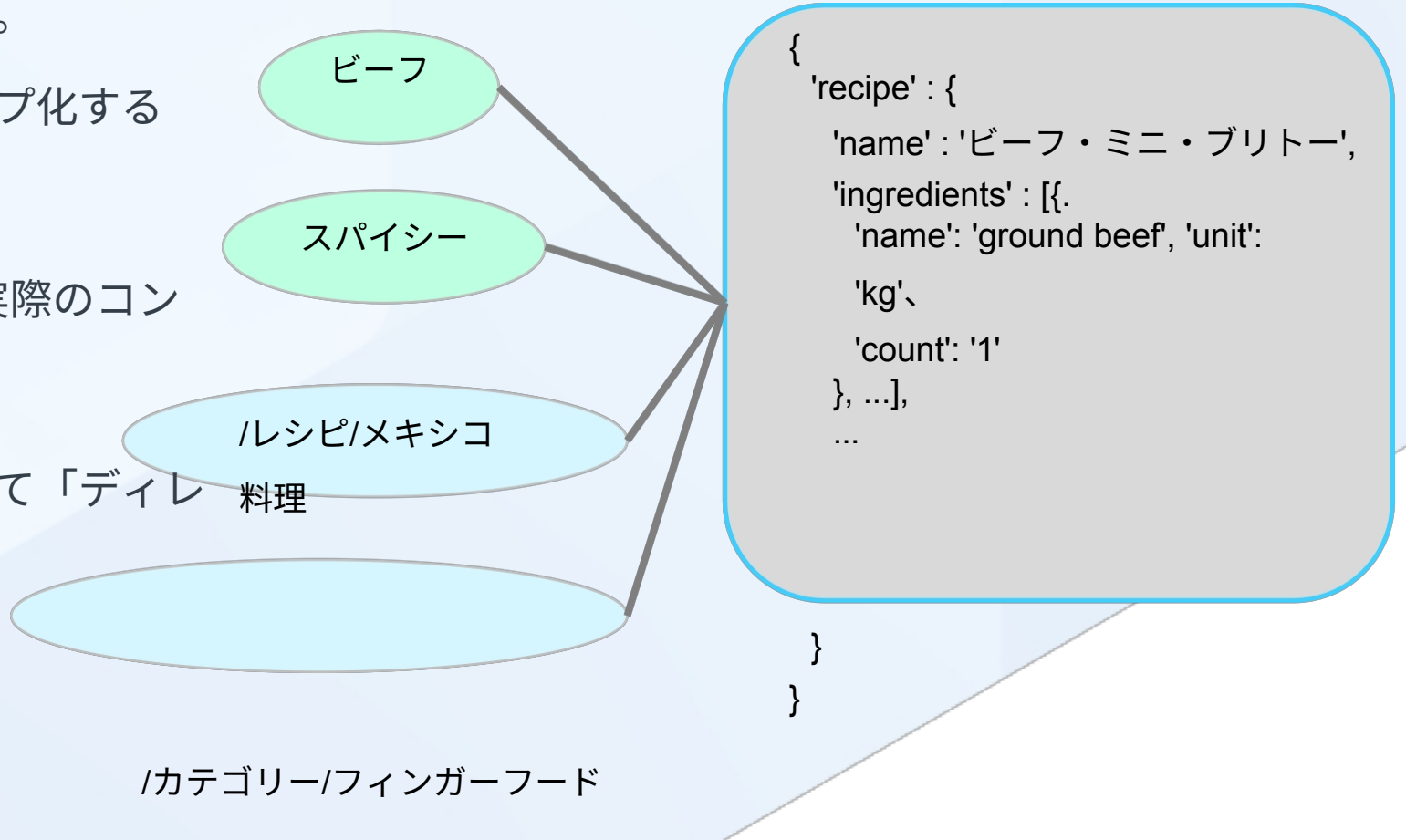
概念統一資源識別子（URI）

- 文書は読み込まれるときにURIが割り当てられる。
- URI:
  - MarkLogic に格納されているドキュメントを一意に識別する**文字列で**、

    `/directory/filename**.json**` や `/employee/142**.json**' などがあります。
  - ` **Row_ID**` のように、RDBMS の主キーと同様の機能を果たします。
- ディレクトリ」は文書を整理するために使われる:
  - 一つの文書は一つの属さない。
  - フォレスト内に実際のディレクトリ構造は作成されない。

コンセプトコレクション

- 文書に「タグ付け」するための文字列。
- 検索や変更操作のために文書をグループ化するために使用される。
- Collectionの値は、メタデータなど、実際のコンテンツの一部ではない。
- コレクションは、テキストの一部として「ディレクトリ」を使用することもできる。
- ドキュメントには
  を望んで
  いる。



コンセプトエンベロープ・パターン

- 情報の区分けを可能にする。
- エンベロープはセクションで構成されている:
  - ヘッダー - 追加のメタデータをキャプチャします
  - Triples - リレーションシップを格納できる

    埋め込みトリプルを格納する。
  - インスタンス - データサービスで使用さ

    れる「エンティティ」格納。

```
{
  エンベロープ」 :{
    'headers' : {
      'update-history' : [{.
        'date' : '2022-01-23'、
        'ユーザー名': 'dan'
      }]
    },
    'triples' : [],
    'instance' : {
      'EntityName' : { 'id' :
        '1278821'、
        'タイプ' : 'レコード'、
        ...
      }
    }
}
}
```

©2024 MarkLogic Corporation は Progress Software Corporation の完全子会社で、無断複製、転載を禁じます。

コンセプトトリプル

- トリプルは3つの部分で構成される：
  - テーマ
  - 述語
  - 対象
- ドキュメントの一部として「埋め込む」ことも、MarkLogicによって「管理する」こともできます。

エンベデッド・トリプル

```
{
  エンベロープ」 :{
 'triples' : [{] で
              す。
 トリプル」  :{
        'subject' : 'http://my.ns/id-1234', 'predicate'
        : 'http://my.ns#property', 'object' : 'value'
      }
    }...
```

マネージド・トリプル

```
<sem:triples   xmlns:sem="http://marklogic.com/semantics">
   <sem:トリプル
      <sem:subject>http://my.ns/id-1234</sem:subject>。
      <sem:predicate>http://my.ns#property</sem:predicate>
      <sem:object     datatype="http://www.w3.org/2001/XMLSchema#string"     xml:lang="en">値</sem:object>。
   </sem:triple>
</sem:triples>
```


©2024 MarkLogic Corporation は Progress Software Corporation の完全子会社です。無断複製・転載を禁じます。

リレーショナル・レコードをドキュメントに変換

- RDBMSのテーブルに格納されているデータは、一般的に正規化されている。
  - 関連する複数の複数のレコードを持つ。
- 関連レコードは、MarkLogicにロードされ、個々のドキュメントに変換されるときに、多くの場合、非正規化されます。

オーダーテーブル

オーダーID	注文日
10072	2017-01-14

注文項目テーブル

アイテムID	オーダーID	製品	数量
23122312	**10072**	スピードプロ・アルティメッ	1

URI→ /orders/10072.json

```
{
   「orderNum": "10072",
  "orderDate": "2017-01-14", "items": [{
    "項目": {
      "製品": 「スピードプロ・アルティ
      メット"、"価格": 999,
      "quantity": 1
    }
  }, {
    "項目": {
      "product": 「Mens Racer Helmet",
      "price": 95,
      "quantity": 1
    }
```

		ト	
23857292	**10072**	メンズ・レーサー・ヘルメット	1

専有情報 | ©2024 MarkLogic Corporation | Progress Software Corporation の完全子会社 | 無断複製・転載を禁じます。

ツールMarkLogic Content Pump - MLCP

- Javaベースのツール：
  - ディレクトリとそのサブディレクトリにある複数の文書を読み込む。
  - 区切りテキストファイルの各行を文書に変換。
  - RDFトリプルを同等のXML変換。
  - カスタムURIの生成とコンテンツ変換。
  - スレッドのスロットルとバッチサイズ。
- あるクラスタのMarkLogicデータベースから、別のクラスタのMarkLogicデータベースへドキュメントを抽出できます（DevからTest/UATなど）。

MLCPの実行

- コマンドラインまたはカスタムml-gradleタスクとして：
  - コマンドライン

```
$home/cent/mlcp/bin/mlcp.sh import -mode local -host localhost -port 8100 -username admin -password admin -
output_collections "star-wars,characters" -input_file_path /home/cent/Desktop/fundamentals/data/star- wars/characters/ -
output_uri_replace ".*characters,'/characters'"
```

  - カスタムml-gradleタスク - build.gradle定義します。

その他のデータ・ロード・ツール

- MLCPだけでなく、他のツールでもMarkLogicデータベースにデータをロードできます:
  - Javaデータ移動SDK
  - NodeJS MarkLogicクライアント
  - Apache NiFiプロセッサ
  - MuleSoftコネクタ
  - ペガ・コネクター
  - Hadoopコネクタ
  - Kafka Connectorなど。



Progress MarkLogic

研究室

負荷データ

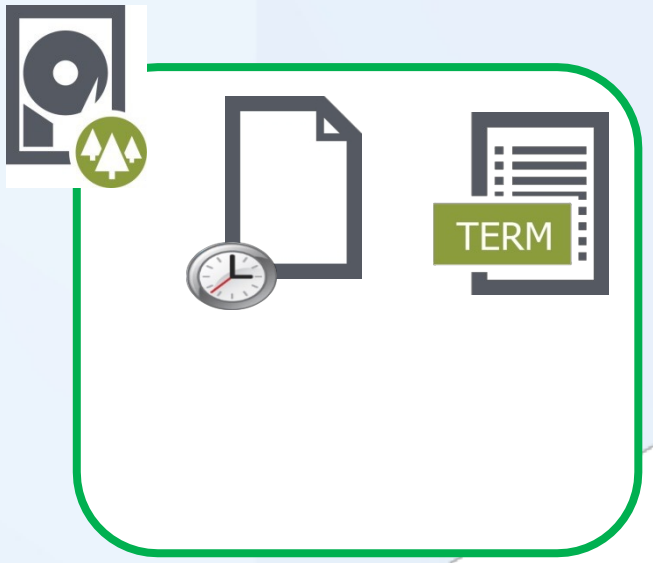
MLCP - ml-Gradleカスタムタスクの作成

Progress MarkLogic

文書管理

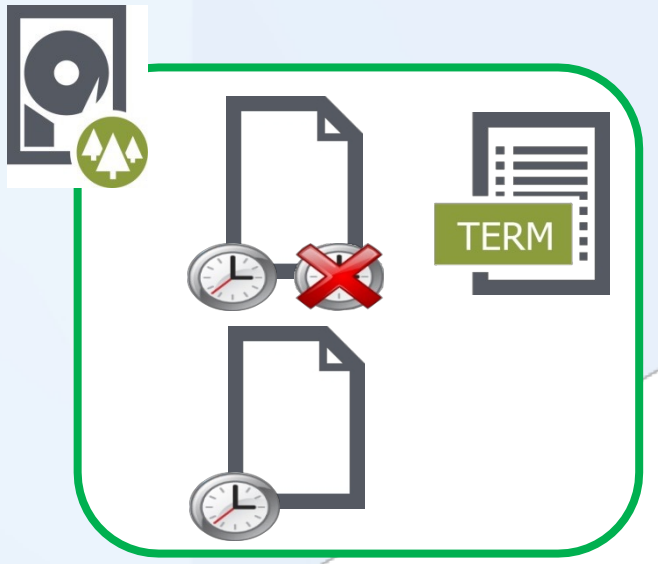
マルチバージョン同時実行制御 - MVCC

- ドキュメントが最初にロードされると、MarkLogic はドキュメントの「開始タイムスタンプ」を設定します。
- 並行性についての詳細は、ドキュメントをご覧になるか、データ・サービス・コースを受講してください。



更新情報

- 同じURIを持つドキュメントを削除または変更するために `xdmp` 関数を使用すると、削除されたフラグメントが生成される。
- 以下のアクションは更新としてカウントされます:
  - 同じURIで新しいコンテンツをロードする。
  - ドキュメントの権限を変更する。
  - コレクションの追加と削除
  - プロパティ/エレメントの置換/追加/削除。
  - ドキュメントのメタデータ（品質、プロパティ、メタデータ）の変更。
- 更新のたびに、その文書の新しいバージョンが作成される。

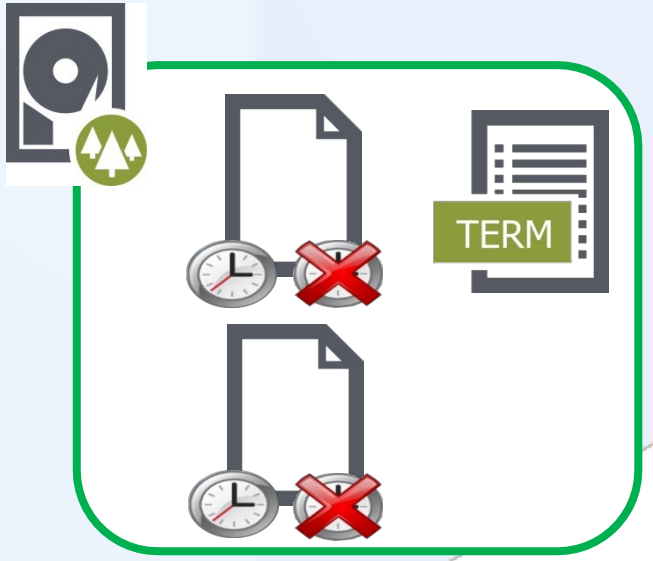


- 終了タイムスタンプ」が設定されている。

専有情報 | ©2024 MarkLogic Corporation は Progress Software Corporation の完全子会社です。無断複製、転載を禁じます。

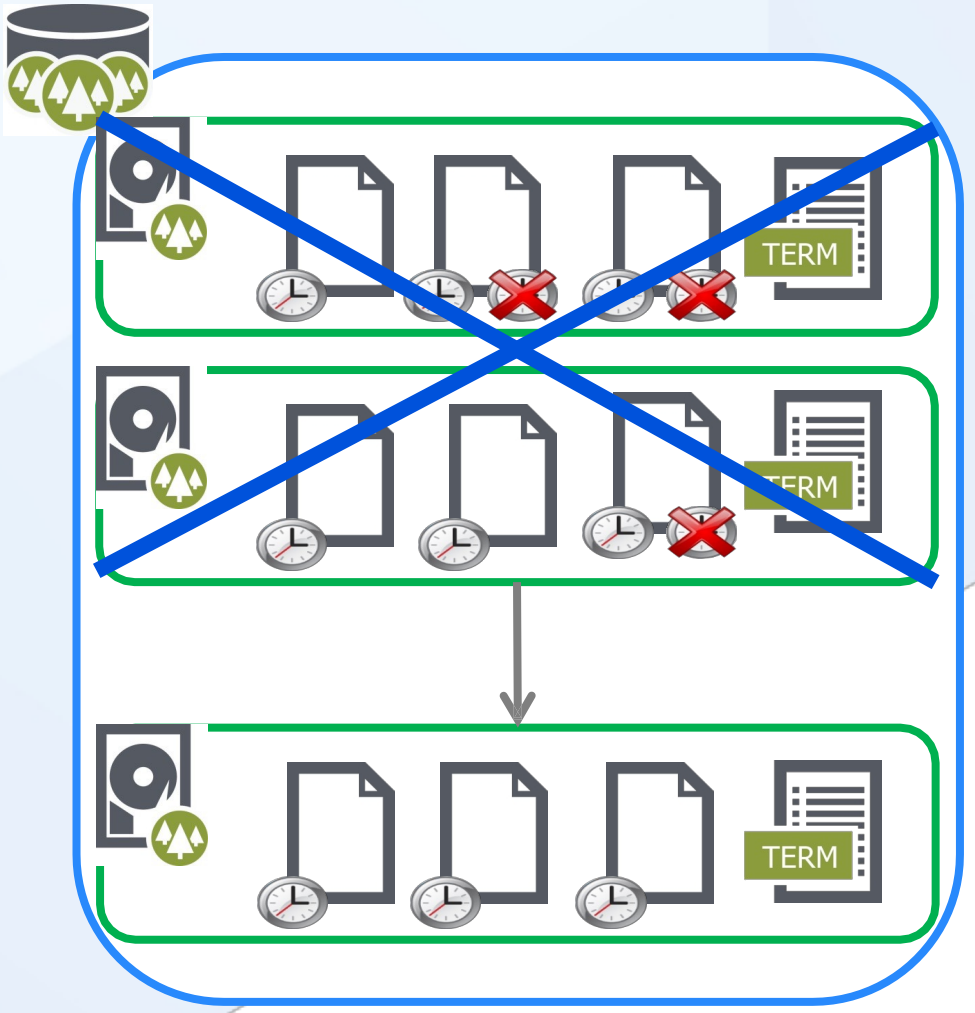
削除

- 文書を削除すると、文書の「終了タイムスタンプ」が設定されます。
- 終了タイムスタンプ」を持つ文書は「削除された断片」と呼ばれる。
- 物理的な削除はすぐには行われない。
  - 削除されたフラグメントは時間の経過とともに蓄積されますがクエリー速度に大きな影響はありません。
- MarkLogic は、` **Merge**` プロセスを実行していない限り、ポイントインタイムクエリをサポートします。



マージ

- MarkLogicは、「マージ」時に削除された
  フラグメントを削除します。
- これによって資源を回収することができる:
  - ディスク
  - RAM



文書作成ライブラリ

- XMLデータ管理プラットフォーム（xdmp）：
  - **xdmp.documentLoad** - ファイルシステムからデータベースにドキュメントをロードするために使用される関数。
  - **xdmp.documentInsert** - コードブロック内で作成されたドキュメントなど、インメモリドキュメントをデータベースに書き込むために使用される関数。
- ユニ・ライブラリー、バイ・タイム・ライブラリー:
  - temporal` - 保護されたコレクション内の一連のバージョン管理されたドキュメント。
  - **temporal.documentLoad**および**temporal.documentInsert** - 一時間または二時間のドキュメントを実装する際に使用される同様の関数。
- REST API

- コンテンツをロードするためにHTTP REST APIを呼び出すことができるクライアント側アプリケーションを許可する。

専有情報 | ©2024 MarkLogic Corporation、Progress Software Corporation の完全子会社。無断複製、転載を禁じます。

マルチ・バージョンと`時間`関数

- Temporalは、さまざまな時間次元にわたってトランザクションのスナップショットを維持する必要がある場合に使用される。
  - 例金融・保険業界では、厳格な規制やコンプライアンス要件を遵守するために、契約、ポリシー、イベントの変更を追跡するためにビットタイムデータを使用しています。
- Temporal Functions を使用すると、MarkLogic は新しいシステム時間（およびオプションの有効時間）でドキュメントのバージョンを生成します。
  - これらのバージョンはマージの際に保持される。
  - 時間に関連する制約の詳細については、Temporal Developer's Guideを参照してください。

文書を更新する内蔵メカニズム

- xdmp` と `temporal` ライブラリである：
  - nodeReplace`や他の同様の関数は、メタデータに影響を与えることなく、ドキュメントの内容や構造だけを更新することができます。
  - documentAddCollections`、`documentAddPermissions` などは、実際のコンテンツに影響を与えることなく、ドキュメントのメタデータを変更するために使用することができる。
- REST API：
  - PUTとPOSTは、コンテンツとメタデータ（品質、パーミッション、コレクションなど）の両方を更新できる。
  - PATCHは、文書の内容やメタデータだけを個別に更新することができる。
    - nodeReplace、documentAddCollections、documentAddPermissionsなどの

関数に相当する。

ツールコンテンツ一括再処理 - CORB

- Javaベースのツール:
  - 2つのモジュール/ファイルを使用:
    - 処理する文書を定義するカスタムURIセレクタモジュール。
    - URIセレクタモジュールによって識別されたドキュメントを処理するためのカスタム変換モジュール。
  - バッチ前後のタスクフック。
  - スレッドのスロットルとバッチサイズ
- コマンドラインまたはカスタムGradleタスクで起動します。

ツールMarkLogicデータハブ

- 一元化された運用データ・ハブを構築するための共通タスクと構成を統合するフレームワーク。
- などのベストプラクティスを採用:
  - 封筒パターンの使用
  - 団体サービスの利用
- 一般的なキュレーション・タスクに対応する、さまざまな構築済みステップを提供します:
  - 既存のコンテンツをターゲットマッピングする。
  - スマートマスタリング - 重複文書のマッチングとマージ。

専有情報 | ©2024 MarkLogic Corporation は Progress Software Corporation の完全子会社です。無断複製、転載を禁じます。

Progress MarkLogic

研究室データへのアクセスと更新

Progress MarkLogic

セキュリティの基本

特典

- データを作成または変更するアクション、あるいは特殊な方法でデータにアクセスするアクションには、対応する権限が必要です。
- 権限をユーザーに直接割り当てることはできない。

xdmp.documentInsert

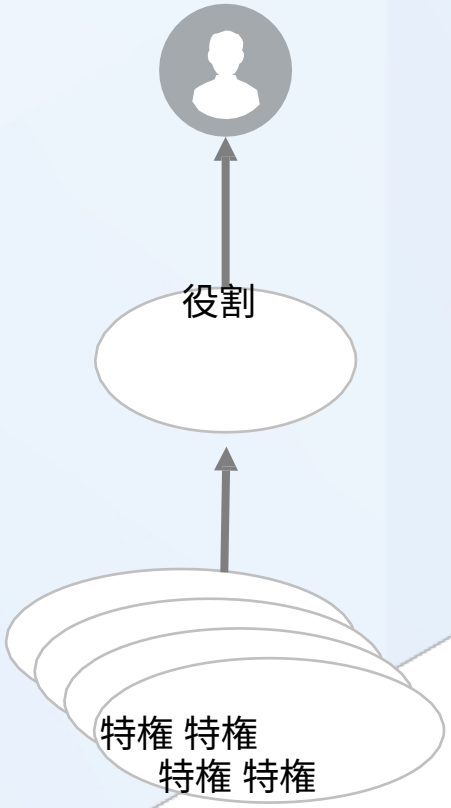
XQuery xdmp:document-insert

Required Privileges

If a new document is inserted, the `unprotected-uri` privilege (only if the URI is not protected), the `any-uri` privilege, or an appropriate URI privilege is also needed. If adding an unprotected collection to a document, the `unprotected-collections` privilege is needed; if adding a protected collection, the user must have either permissions to update the collection or the `any-collection` privilege.

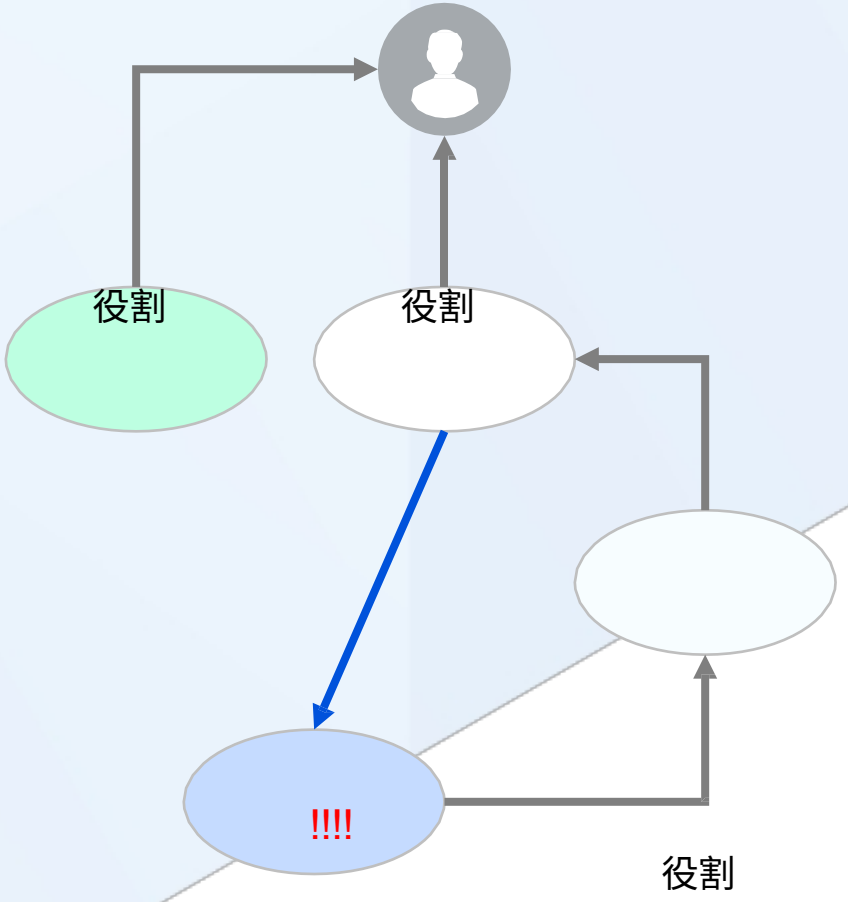
役割

- **ロールは**、MarkLogicデータベースのドキュメントへのアクセスを制御する上で中心的な役割を果たします。
- 各ロールは、ユーザーがタスクを実行するための機能へのアクセスを許可する**権限の**集まりである。
- ユーザーにはロールが割り当てられる。



役割

- ユーザーは複数のロールを割り当てることができ、ロールは複数のユーザーに割り当てることができる。
- ロールは他のロールを継承して、他のロールが許可されているすべてのアクションを実行できるようにすることができる。
    - 循環的な役割継承に注意。

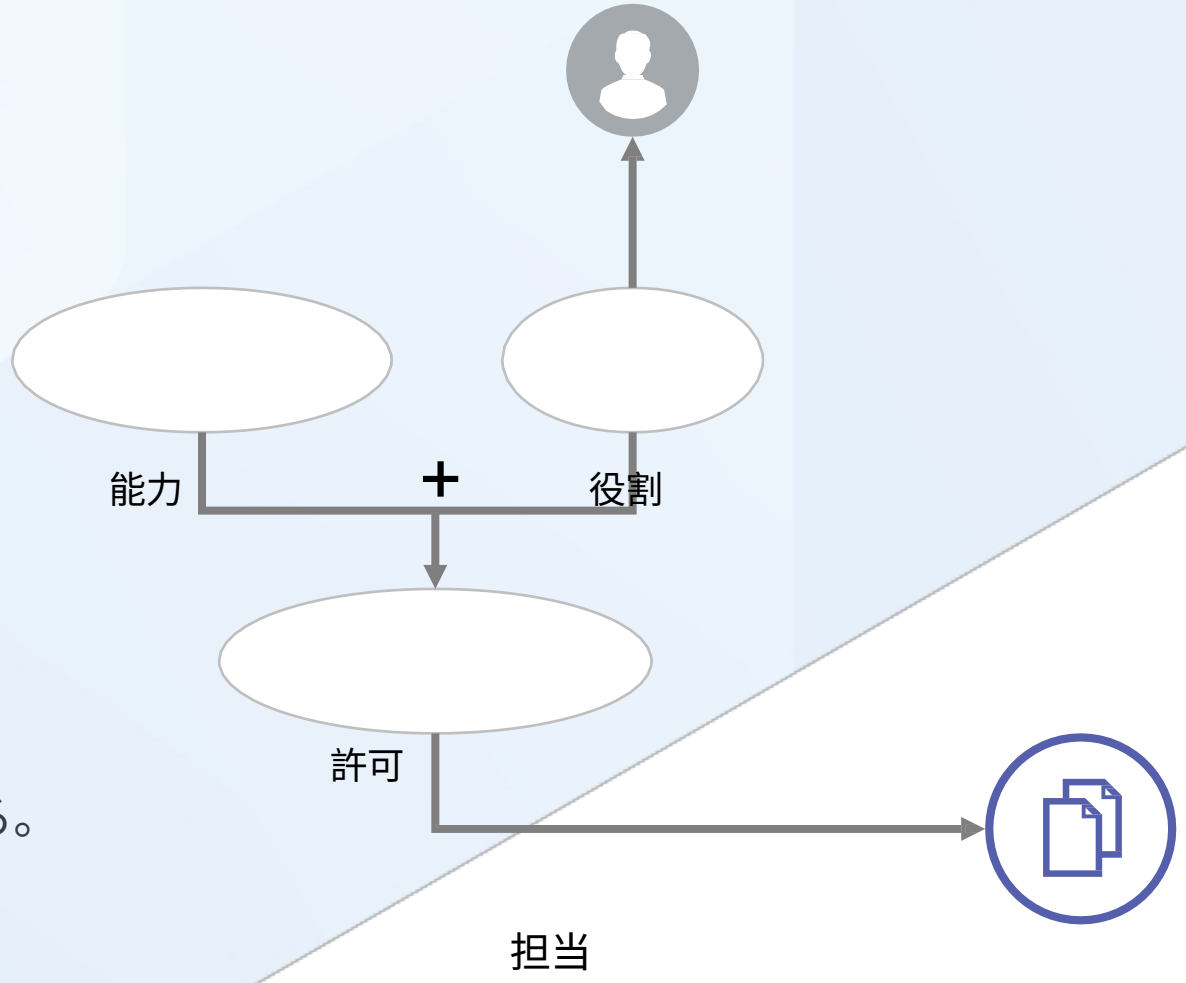


役割

役割ベースのアクセス制御（RBAC）

- 文書パーミッション= ロール+ ケイパビリティ
- 能力は以下の通り：
    - 挿入
    - 読む
    - 更新
    - ノードアップデート
    - 実行（モジュール固有）
- 文書には複数の権限を割り当てることができる。
- パーミッションは、挿入時に割り当てることができる。

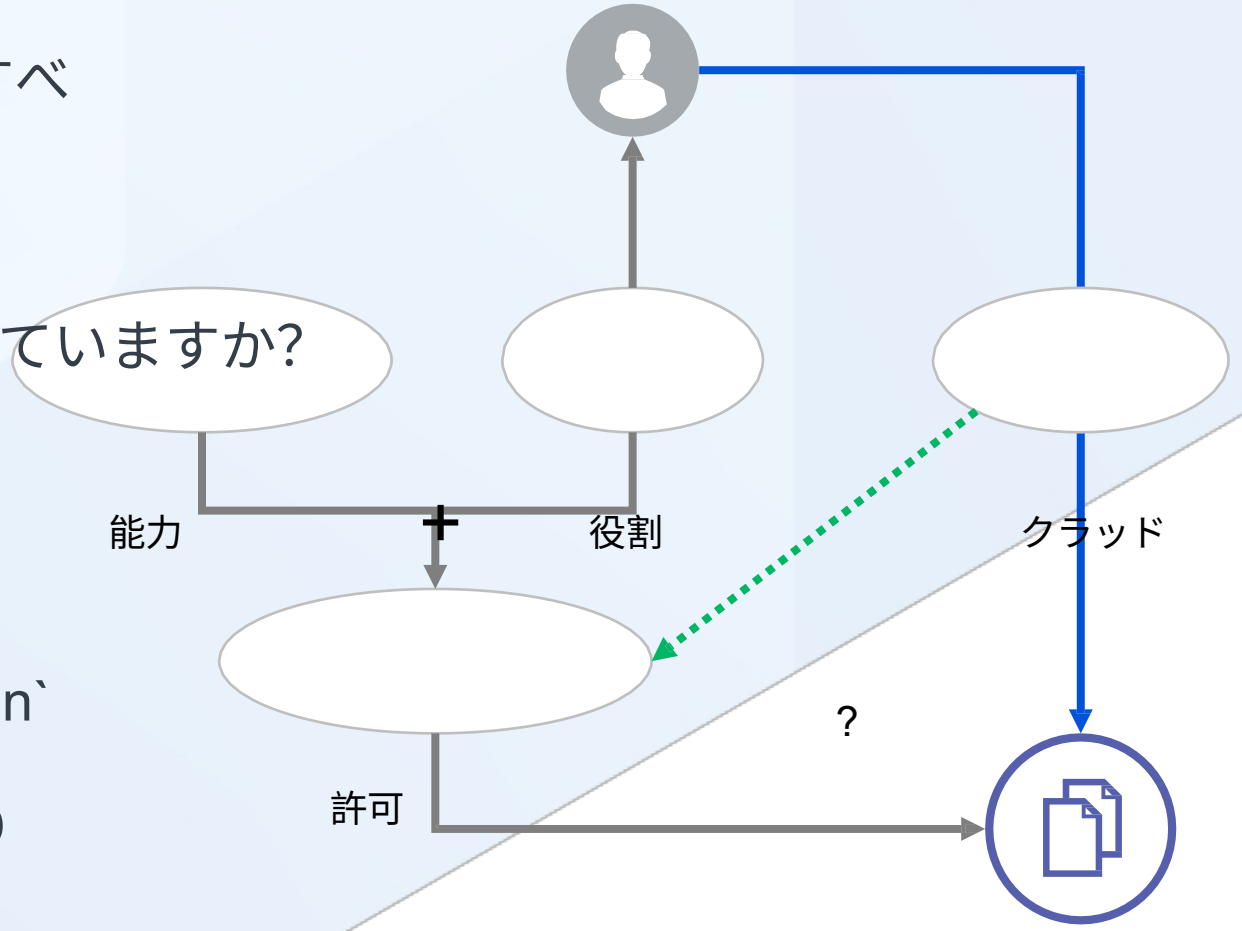
  (推奨)またはそれ以降 - アップデートとして



役割ベースのアクセス制御（RBAC）

- ユーザの各アクションは、ユーザの割り当てられたロールを、ドキュメントの設定されたすべてのパーミッションと照合します:
  - そのユーザーにはロールが割り当てられていますか?
  - その役割は適切か?

    能力?
- パーミッションのないドキュメントは `admin` レベルのユーザーだけが読んだり変更したりすることができます。



Progress MarkLogic

研究室

CORB2 - ml-Gradleカスタムタスクを使用する WebDav を使用してファイルをアップロードする(オプション)

総括

- MarkLogic Serverは、構造化コンテンツ、非構造化コンテンツ、バイナリコンテンツをサポートしています。
- グラフは、ある形状／構造を持つ構造化された文書である。
- ドキュメントは、ディレクトリやコレクションを使って整理することができる。
- 封筒の柄を利用して、書類を区分けする。
- MarkLogic Serverは、挿入/更新のたびに新しいバージョンのドキュメントを作成します。
- MLCPのようなツールを使って、効率的に一括インジェストを行う。
- CoRB2のようなツールを使って、効率的な一括更新を行う。