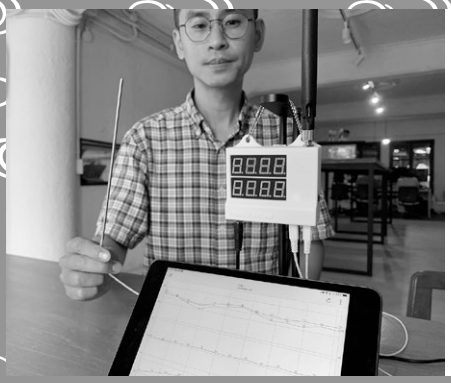


職人さんの手仕事を応援する

クラウド×酒蔵挑戦記

羽角 均

第5回 IoT温度管理クラウドのセキュリティ対策

本連載は島根県の小さな酒蔵，旭日酒造を舞台に，自作しているIoTシステムやシステムのさらなる成長の様子をリアルタイムに近いかたちで紹介します．

今回の話題は筆者が管理するデータベースに，別々の顧客のデータを読み書きする際に起こる「セキュリティ問題」です．コストの制約からA酒造のデータとB酒造のデータが，同じテーブルに保存されます．

クラウド・データベースへのアクセス管理が必要な背景

● 案件ごとにデータベースを作るのは大変

ウェブ・サービスにおけるマルチテナンシ(Multitenancy)アーキテクチャとは，複数の顧客が同一のデータベースを共用することを指します．マルチテナントとも呼ばれるこの方式では，A酒造のセンサ・データと，B酒造のセンサ・データが同じテーブルに保存されます．理由はコストを下げるためです．顧客ごとにデータベースを1つずつ作っていたら管理が大変です．

● サーバレスの落とし穴…サーバのハックは難しいけれどスマホのハックは可能

連載第3回(2020年1月号)では，リレーショナル・データベース管理システム(RDBMS)とNoSQLデータベースの違いを説明しました．RDBMSは関係モデルに基づくクエリ(問い合わせ)を発行できるため，A酒造に所属するデータはA酒造にしか見せないという制限がデータベースのレベルでコントロールできます．

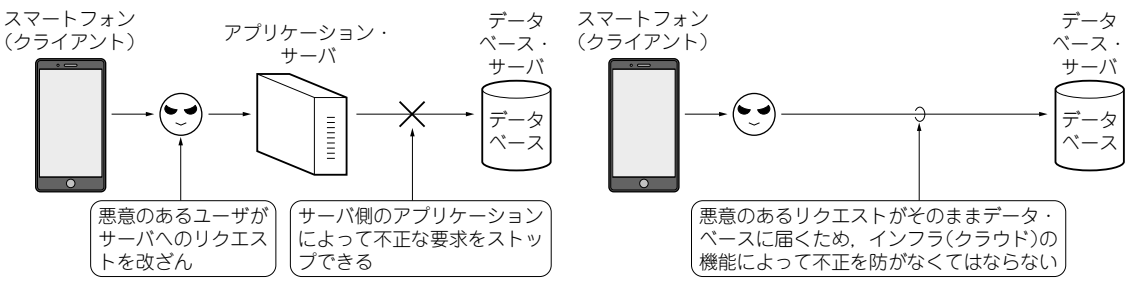
一方，NoSQLデータベースは関係モデルに基づいたクエリを発行できません．A酒造の蔵人がB酒造のセンサ・データを見てしまうことを防ぐような機能は，一般的にNoSQLデータベースの側にはありません．

ここでの問題は，関係モデルによるクエリかどうかではなく，「クエリを発行するのは誰なのか」です．サーバの動作をクラックするのは一般的に難しいのに対し，クライアント側(ここではスマートフォン・アプリ)の動作ならば，能力と悪意に満ちたユーザであれば変更できてしまいます．理由はモバイル端末がユーザの管理下にあるからです．これはサーバレスの落とし穴だと言えるかもしれません(**図1**)．

クラウド・データベースへのアクセス管理対策

● クラウド・サービスの機能を活用する

もちろん解決手段もあります．使用しているDynamoDBが提供しているたくさんの機能の1つに，「認可されたCognito[注1] IDに基づいたDynamoDBテーブルへの行レベルのアクセス制限」があります．



(a) サーバがクエリを発行する従来手法　(b) スマートフォン・アプリがクエリを発行するサーバレス

図1　課題…モバイル端末がクエリを発行するサーバレス・ユーザ管理はクラウドの機能で不正を防ぐ必要がある

第1回　職人さんを応援するIoT温度管理システム(2019年11月号)
第2回　注目「サーバレス」IoT温度管理システムの基本クラウド構成(2019年12月号)
第3回　IoT温度管理システムのデータベース(2020年1月号)