

Sales-Cat - Salesforce Data Synchronization Platform

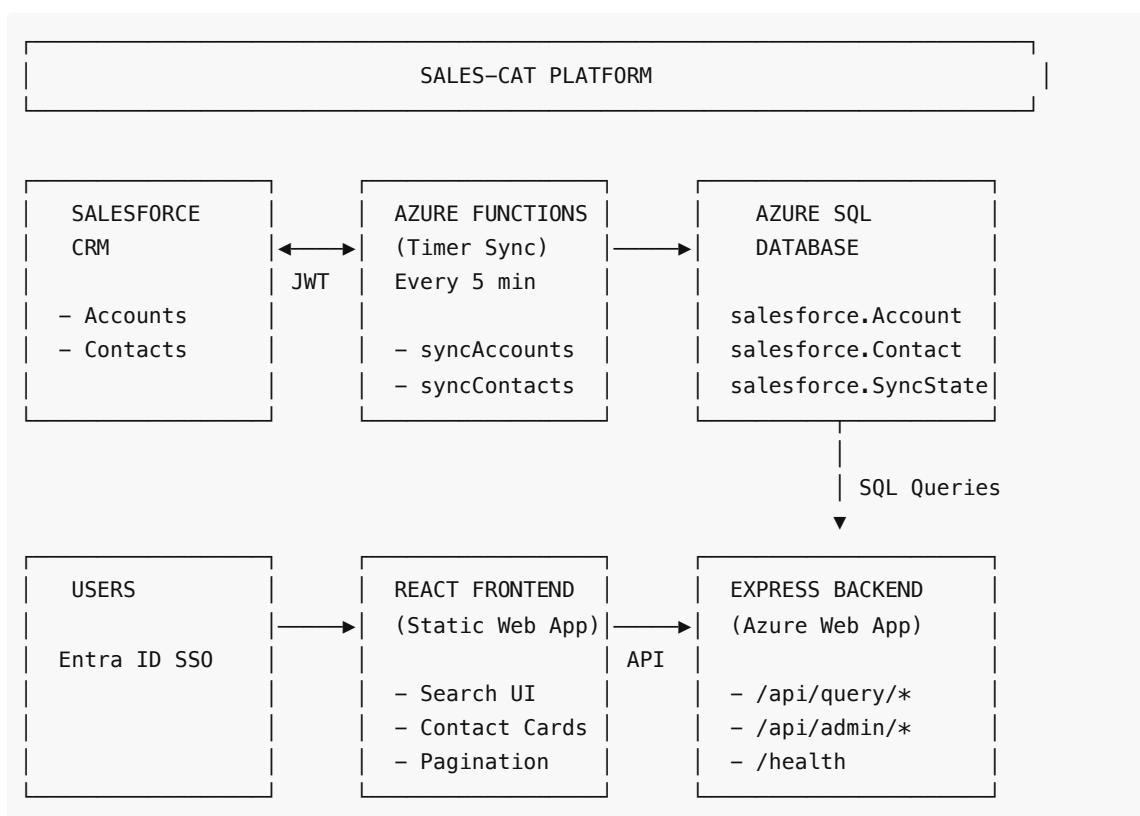
Executive Summary

Sales-Cat is a **full-stack Salesforce data synchronization and search platform** that bridges Salesforce CRM data with Azure infrastructure. It continuously syncs Account and Contact records from Salesforce to Azure SQL, providing a fast, searchable web interface for end users. The platform uses Microsoft Entra ID for authentication and features a modern React frontend with an Express.js backend API.

Business Value

Benefit	Description
Fast Search	Query contacts instantly from Azure SQL instead of slow Salesforce API calls
Always Current	Automatic sync every 5 minutes keeps data fresh
Enterprise Security	Microsoft Entra ID SSO with role-based access control
Modern UX	Clean, responsive React interface with Catalyst branding
Scalable Architecture	Azure-native services handle growth automatically

Architecture Overview



Components

1. Azure Functions (Data Sync)

Timer-triggered functions that sync Salesforce data to Azure SQL every 5 minutes.

2. Express Backend API

RESTful API providing authenticated access to synchronized data.

3. React Frontend

Modern single-page application for searching and viewing contacts.

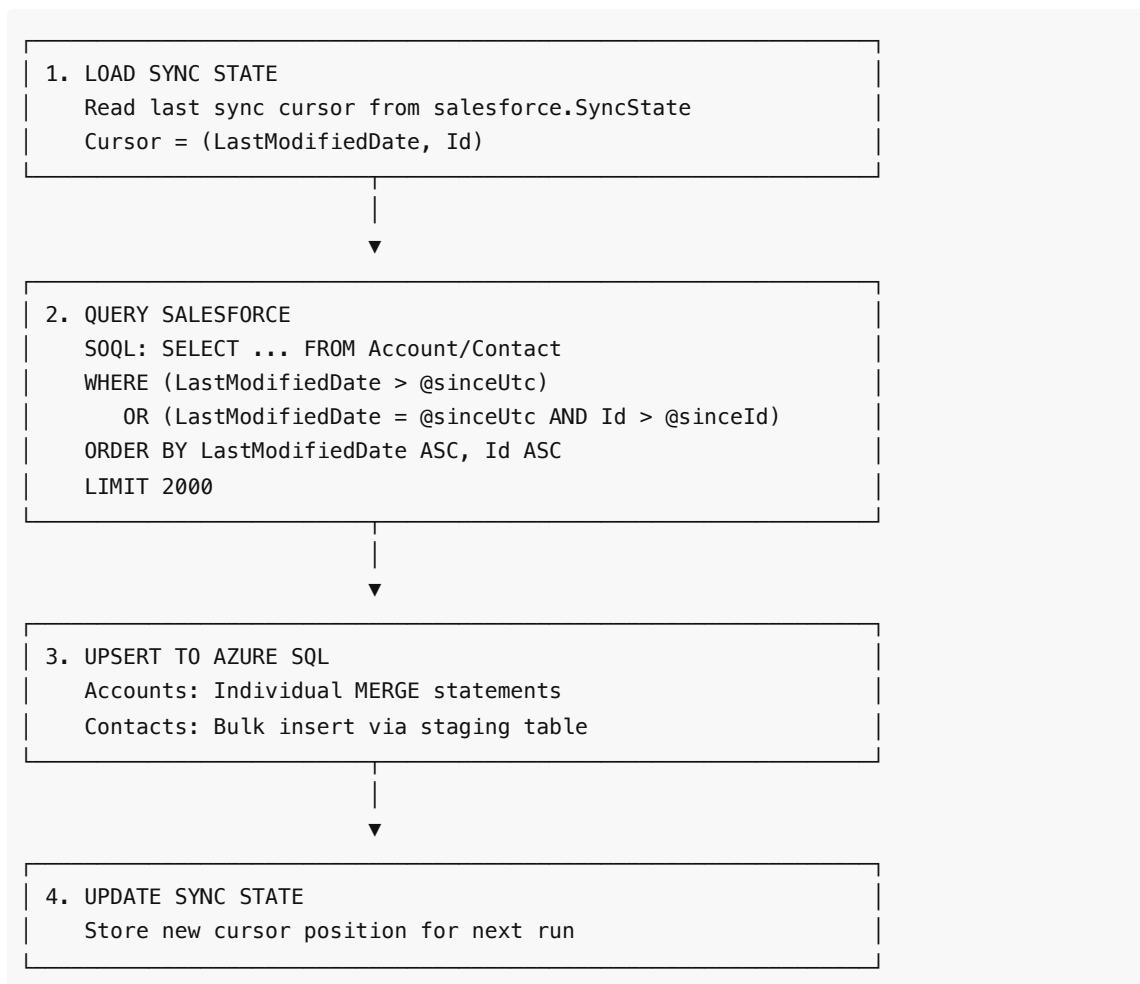
4. Azure SQL Database

Central data repository storing synced Salesforce records.

Data Synchronization

Sync Strategy: Incremental Cursor-Based

The sync process uses a stable cursor pattern to efficiently fetch only changed records:



Sync Schedule

- **Frequency:** Every 5 minutes (`0 */5 * * *`)
 - **Functions:** `syncAccounts` , `syncContacts`
 - **Batch Size:** Up to 2,000 records per query
 - **Recovery:** Falls back to Jan 1, 2000 if state is reset
-

Database Schema

salesforce.Account

```
CREATE TABLE salesforce.Account (
    Id          NVARCHAR(18) PRIMARY KEY,   -- Salesforce ID
    Name        NVARCHAR(255) NOT NULL,
    Type        NVARCHAR(100),
    Industry    NVARCHAR(100),
    OwnerId     NVARCHAR(18),
    LastModifiedDate DATETIME2 NOT NULL
)
```

salesforce.Contact

```
CREATE TABLE salesforce.Contact (
    Id          NVARCHAR(18) PRIMARY KEY,   -- Salesforce ID
    Name        NVARCHAR(255),
    Email       NVARCHAR(255),
    Phone       NVARCHAR(50),
    MailingStreet NVARCHAR(255),
    MailingCity  NVARCHAR(100),
    MailingState NVARCHAR(100),
    MailingPostalCode NVARCHAR(20),
    MailingCountry NVARCHAR(100),
    LastModifiedDate DATETIME2 NOT NULL
)
```

salesforce.SyncState

```
CREATE TABLE salesforce.SyncState (
    KeyName     NVARCHAR(100) PRIMARY KEY,
    Value       NVARCHAR(4000),
    LastUpdated DATETIME2
)
```

API Endpoints

Query Endpoints (Authenticated)

Method	Endpoint	Description
GET	/api/query/me	Current user info
GET	/api/query/accounts	Paginated account list
GET	/api/query/contacts	Contact search with pagination

Admin Endpoints (SalesforceAdmin Role Required)

Method	Endpoint	Description
GET	/api/admin/	Admin access check
GET	/api/admin/db-health	Database connectivity test

Public Endpoints

Method	Endpoint	Description
GET	/health	Health check (no auth)

Query Parameters

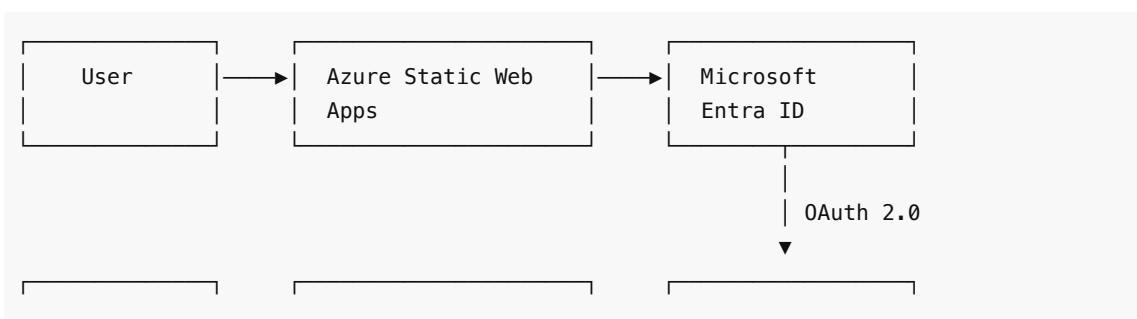
Contacts Search:

```
GET /api/query/contacts?search=john&limit=25&offset=0&sortBy=Name&sortDir=ASC
```

Parameter	Type	Default	Description
search	string	-	Search term (name, email, phone)
limit	number	25	Results per page (max 100)
offset	number	0	Pagination offset
sortBy	string	Name	Sort field
sortDir	string	ASC	Sort direction

Authentication & Authorization

User Authentication Flow





Salesforce Authentication (JWT Bearer)

Azure Functions → Salesforce

1. Create JWT with:
 - iss: Client ID
 - sub: Salesforce username
 - aud: https://login.salesforce.com
 - exp: Current time + 3 minutes
2. Sign JWT with RS256 private key
3. POST to /services/oauth2/token


```
grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer
assertion=[signed JWT]
```
4. Receive access_token (cached for 10 minutes)

Role-Based Access Control

Role	Access
Authenticated	Query contacts and accounts
SalesforceAdmin	Admin endpoints, database health

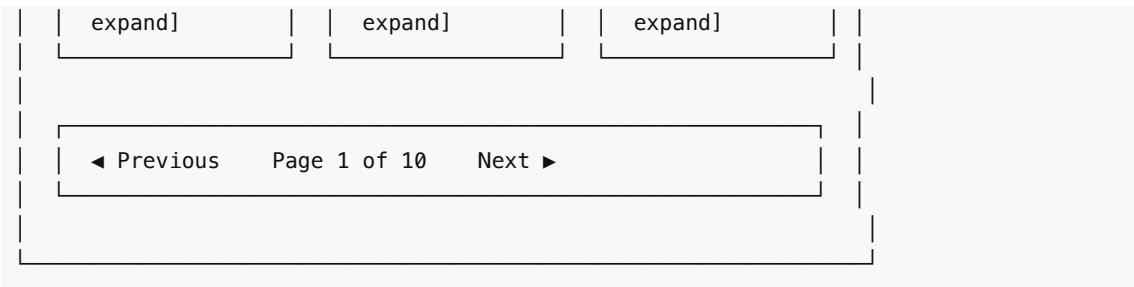
Frontend Features

Contact Search Interface

[Catalyst Logo]
Sales-Cat

Search

John Smith john@email.com (555) 123-4567	Jane Doe jane@email.com (555) 234-5678	Bob Wilson bob@email.com (555) 345-6789
[Click to ...]	[Click to ...]	[Click to ...]



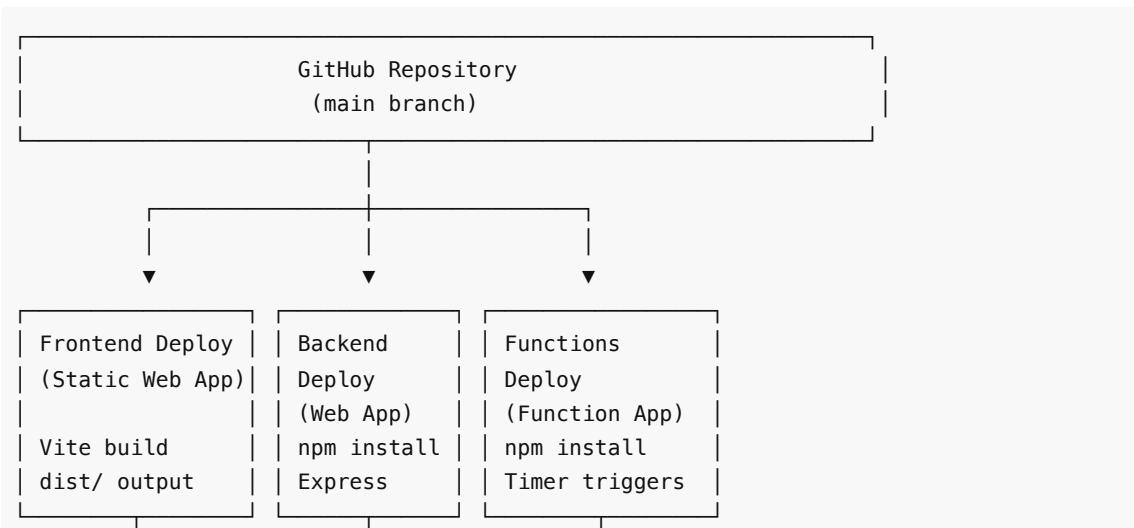
Features

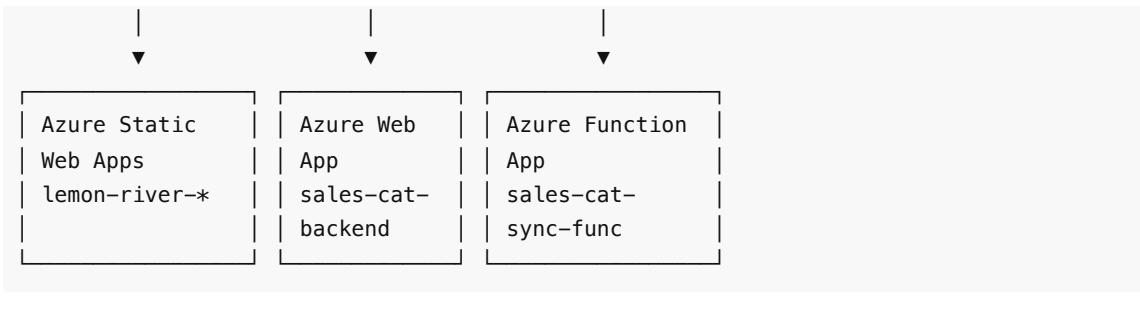
- **Search:** Full-text search by name, email, or phone
- **Pagination:** 25 contacts per page with navigation
- **Expandable Cards:** Click to reveal full address details
- **Sorting:** By name or last modified date
- **Responsive:** Mobile-optimized grid layout

Technical Stack

Layer	Technology
Frontend	React 19, Vite 7, CSS
Backend	Node.js, Express.js
Sync Functions	Azure Functions (Node.js)
Database	Azure SQL Server
Authentication	Microsoft Entra ID
Hosting	Azure Static Web Apps, Azure Web App
CI/CD	Github Actions

Deployment Architecture





Project Structure

```

sales-cat/
├── sales-cat-frontend/          # React + Vite SPA
│   ├── src/
│   │   ├── App.jsx             # Main component
│   │   ├── App.css             # Styling
│   │   └── main.jsx            # Entry point
│   ├── index.html
│   ├── vite.config.js
│   └── staticwebapp.config.json # Azure SWA config

├── sales-cat-backend/           # Express API
│   ├── src/
│   │   ├── app.js              # Express setup
│   │   ├── server.js            # Server entry
│   │   ├── auth/                # Auth middleware
│   │   ├── routes/               # API routes
│   │   ├── services/             # DB & SF services
│   │   └── config/               # Configuration
│   └── package.json

└── sales-cat-function/          # Azure Functions
    ├── functions/
    │   ├── syncAccounts/         # Account sync
    │   └── syncContacts/         # Contact sync
    ├── src/
    │   ├── services/              # SF & SQL clients
    │   └── sync/                  # Sync logic
    └── local.settings.json

└── .github/workflows/
    ├── azure-static-web-apps-*.yml # Frontend CI/CD
    ├── main_sales-cat-backend.yml  # Backend CI/CD
    └── main_sales-cat-sync-func.yml # Functions CI/CD

```

Environment Configuration

Backend (Express)

```
PORT=3000
NODE_ENV=production
AZURE_SQL_SERVER=salescat-salesforce-sqlserver.database.windows.net
AZURE_SQL_DATABASE=salescat-salesforce-db
AZURE_SQL_USER=catadmin
AZURE_SQL_PASSWORD=<password>
ALLOW_DEV_AUTH=false
```

Functions (Sync)

```
AZURE_SQL_SERVER=salescat-salesforce-sqlserver.database.windows.net
AZURE_SQL_DATABASE=salescat-salesforce-db
AZURE_SQL_USER=catadmin
AZURE_SQL_PASSWORD=<password>
SALESFORCE_LOGIN_URL=https://login.salesforce.com
SALESFORCE_CLIENT_ID=<connected app client id>
SALESFORCE_USERNAME=dan.payne@catalystsolutions.com
SALESFORCE_PRIVATE_KEY=<RSA private key>
SALESFORCE_API_VERSION=60.0
```

Frontend (Vite)

```
VITE_API_URL=https://sales-cat-backend.azurewebsites.net
```

Security Features

Feature	Implementation
SQL Injection Prevention	Parameterized queries throughout
Authentication	Microsoft Entra ID with token validation
Role-Based Access	Admin endpoints require SalesforceAdmin role
Encrypted Connections	TLS for SQL, HTTPS for all APIs
JWT Signing	RS256 with private key for Salesforce auth
CORS	Restricted to known origins
Credential Storage	Environment variables, not in code

Performance Optimizations

Optimization	Benefit
Incremental Sync	Only fetches changed records
Batch Staging	Bulk inserts for contacts
Token Caching	10-minute cache for Salesforce tokens

Pagination	Limits query results to manageable sizes
Query Limits	Max 100 records per API request
Index-Friendly Queries	Uses LastModifiedDate + Id cursor

Summary

Sales-Cat provides a robust solution for organizations needing fast, searchable access to Salesforce data within their Azure infrastructure. The incremental sync strategy ensures data freshness while minimizing API calls, and the modern React frontend delivers a responsive user experience. Enterprise-grade security through Microsoft Entra ID and role-based access control makes it suitable for production use.

Document generated: January 2026