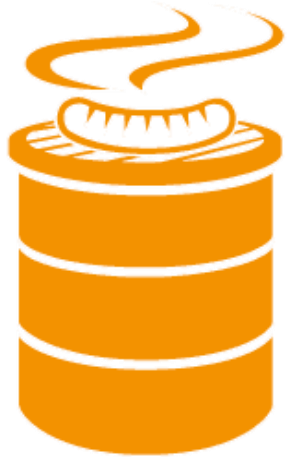


DataGrillen
Datenbanken
Bratwurst
Bier

#DataGrillen

Constantin „Kostja“ Klein - @KostjaKlein
May 17, 2024

Talking API with your data: REST, GraphQL and more

Modern Apps


2:36 PM
Mail
Calendar
Photos
Camera
Maps
Clock
Weather
Wallet
Notes
Reminders
Stocks
Videos
Books
iTunes Store
App Store
Home
Health
Settings
Touch ID
Pricing
Phone
Safari
Messages
Music

Introduction

Modern applications combine a diverse set of new technologies to supply users relevant information directly to their devices at just the right moment and in context.

Scenario : Next gen Public Transportation App

The story of a modern app idea

The idea is simple

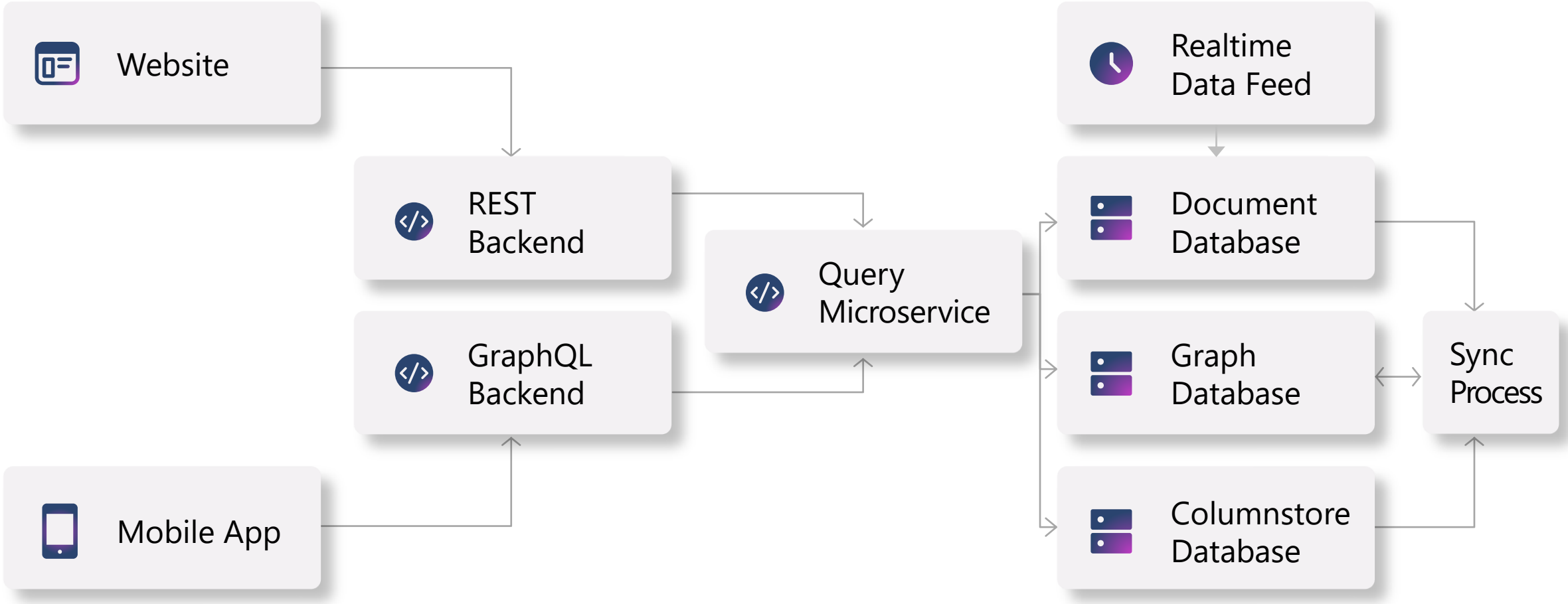
- Let's disrupt the public transportation market and combine different data sources and AI into an intelligent easy-to-use app – mobile and web – which helps people to get from A to B with the quickest option and never again miss a bus or train.

The solution requirements and constraints are complex

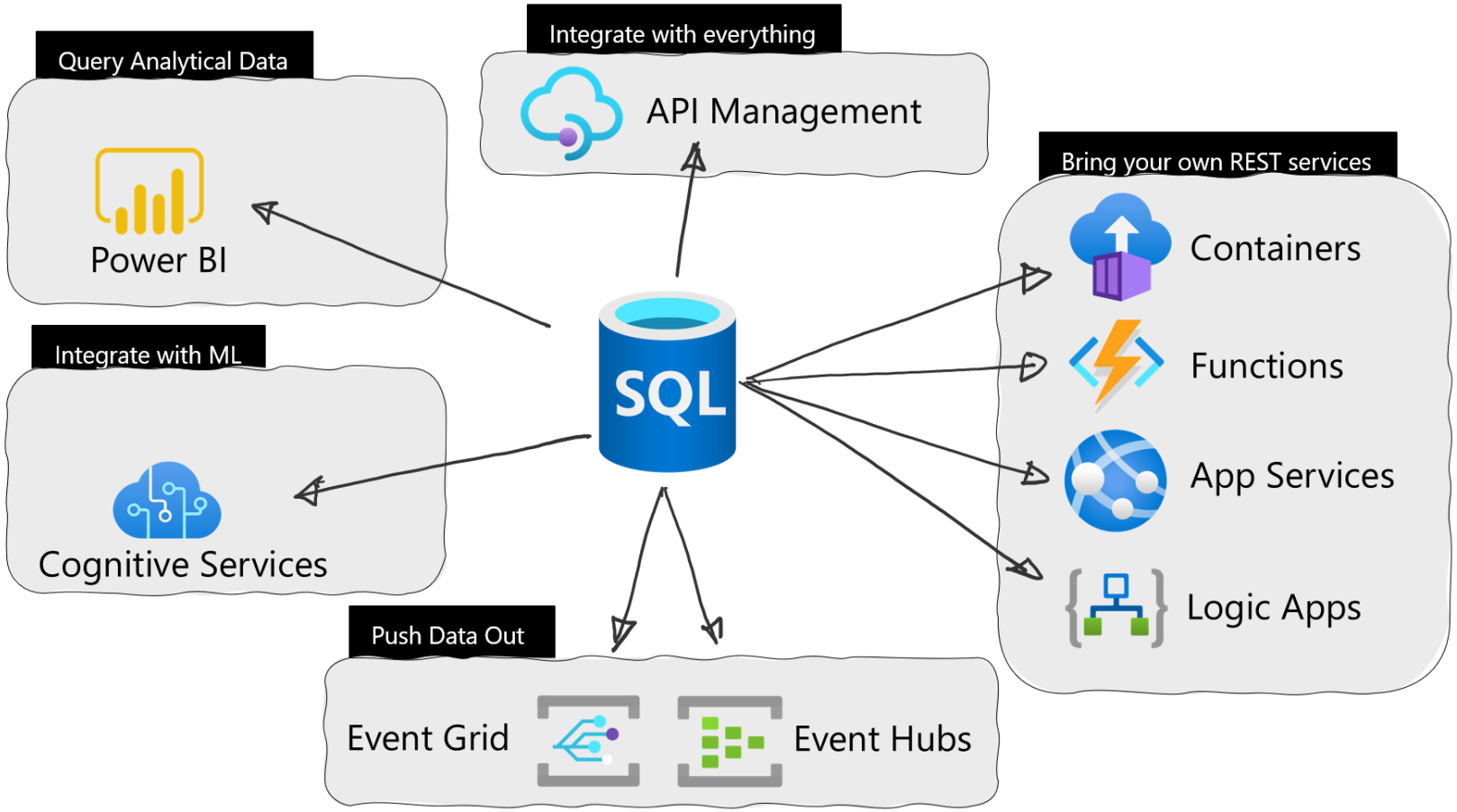
- Real time position data is available in well-known structured format
- Public transportation data is available as JSON
- Data will be queried for single information and aggregated to get historical trends
- The shortest path between two points needs to be identified
- A very agile approach is needed as the solution needs to be on the market in 3 months
- The entire solution should be simple and not be using too many components
- Security is a must, as sensitive data, e.g. user position is being processed


Solution Architecture

Full architecture overview



Calling (Azure) REST endpoints from within your DB



Source

sp_invoke_external_rest_endpoint

… or how do I include external endpoints into my own database?

```
EXEC @returnValue = sp_invoke_external_rest_endpoint
  [ @url = ] N'url'
  [ , [ @payload = ] N'request_payload' ]
  [ , [ @headers = ] N'http_headers_as_json_array' ]
  [ , [ @method = ] 'GET' | 'POST' | 'PUT' | 'PATCH' | 'DELETE' | 'HEAD' ]
  [ , [ @timeout = ] seconds ]
  [ , [ @credential = ] credential ]
  [ , @response OUTPUT ]
```

sp_invoke_external_rest_endpoint

Azure Function using an HTTP trigger binding without authentication

```
DECLARE @ret INT, @response NVARCHAR(MAX);

EXEC @ret = sp_invoke_external_rest_endpoint
  @url = N'https://<APP_NAME>.azurewebsites.net/api/<FUNCTION_NAME>?key1=value1',
  @headers = N'{"header1":"value_a", "header2":"value2", "header1":"value_b"}',
  @payload = N'{"some":{"data":"here"}}',
  @response = @response OUTPUT;

SELECT @ret AS ReturnCode, @response AS Response;
```

Allowed endpoints

If you need/want to go Azure external endpoints then API Management will be your vehicle.

Azure Service	Domain
Azure Functions	*.azurewebsites.net
Azure Apps Service	*.azurewebsites.net
Azure App Service Environment	*.appserviceenvironment.net
Azure Static Web Apps	*.azurestaticapps.net
Azure Logic Apps	*.logic.azure.com
Azure Event Hubs	*.servicebus.windows.net
Azure Event Grid	*.eventgrid.azure.net
Azure Cognitive Services	*.cognitiveservices.azure.com
Azure OpenAI	*.openai.azure.com
PowerApps / Dataverse	*.api.crm.dynamics.com
Microsoft Dynamics	*.dynamics.com
Azure Container Instances	*.azurecontainer.io
Azure Container Apps	*.azurecontainerapps.io
Power BI	api.powerbi.com
Microsoft Graph	graph.microsoft.com
Analysis Services	*.asazure.windows.net
IoT Central	*.azureiotcentral.com
API Management	*.azure-api.net
Azure Blob Storage	*.blob.core.windows.net
Azure Files	*.file.core.windows.net
Azure Queue Storage	*.queue.core.windows.net
Azure Table Storage	*.table.core.windows.net
Azure Communication Services	*.communications.azure.com
Bing Search	api.bing.microsoft.com
Azure Key Vault	*.vault.azure.net
Azure AI Search	*.search.windows.net

Outbound Firewall Rules control mechanism can be used to further restrict outbound access to external endpoints.


Data is great,
but my
developers
need an API!

API, REST, GraphQL

API

- An API (Application Programming Interface) is a set of functions and protocols that enables communication between different software applications or systems.

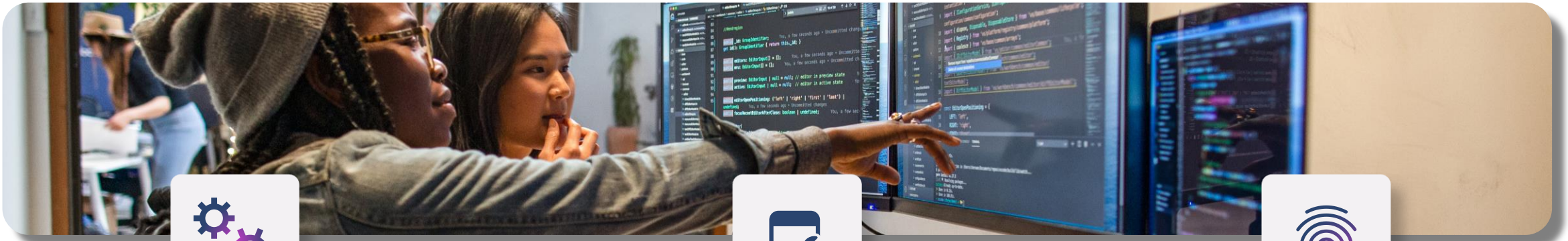
REST

- REST (Representational State Transfer) is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of a distributed, Internet-scale hypermedia system, such as the Web, should behave.

GraphQL

- GraphQL is an open-source data query and manipulation language for APIs and a query runtime engine. GraphQL enables declarative data fetching where a client can specify exactly what data it needs from an API.

Data is great, but my developers need an API!



There's a lot of work needed to create a scalable backend to support CRUD operations with all best practices and features

- Pagination
- Filtering
- Field Selection
- Sorting

Some services will use REST, but some will prefer to use GraphQL

Authentication and authorization must be put in place to avoid showing private data to the wrong user

Data API Builder – the answer?

An Open Source tool to build an API over your data

- https://aka.ms/dab

Exposes database operations

- Creates REST and/or GraphQL endpoints

Available Cross-Platform

- Windows, Linux, macOS

Multiple SQL Databases

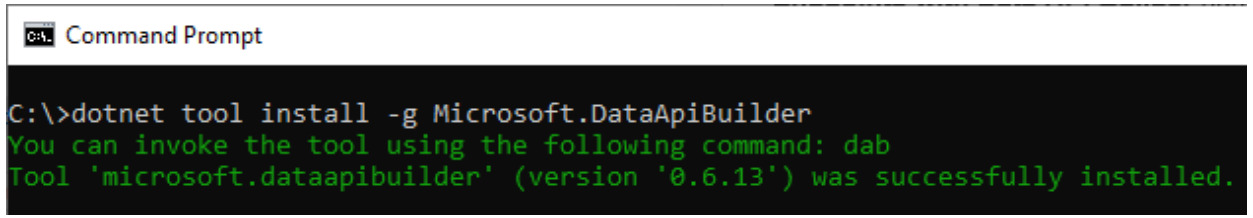
- Azure SQL DB, SQL Server, MySQL and PostgreSQL

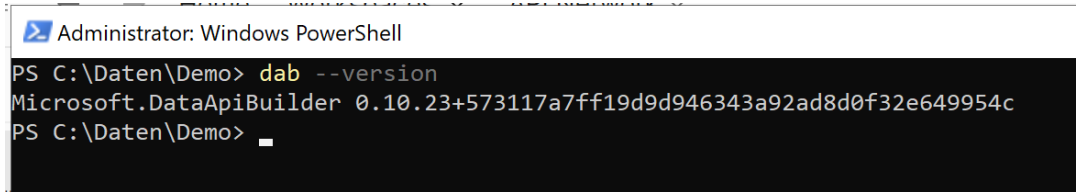
NoSQL Databases

- JSON documents in Azure Cosmos DB

Data API Builder – how to get started?

- Install .NET 6
  - Check installed versions with: dotnet --list-sdks
  - If needed install from: https://dotnet.microsoft.com/en-us/download/dotnet/6.0
- Install the DAB CLI
  - dotnet tool install --global Microsoft.DataApiBuilder
  - If installed, check version with: dab --version
  - To update an existing installation use:
    dotnet tool update --global Microsoft.DataApiBuilder





Data API Builder – using the DAB CLI

- Initialize the configuration file
  - dab init [options]
  - e.g. dab init --database-type "mssql" --connection-string "Server=localhost;Database=PlaygroundDB;User ID=PlaygroundUser;Password=<Password>;TrustServerCertificate=true" --host-mode "Development"
- Add database entities
  - dab add [entity-name] [options]
  - e.g. dab add Author --source dbo.authors --permissions "anonymous:*"
- Start the runtime engine
  - dab start


Demo

DAB Configuration File

dab-config.json

C: > Daten > Demo > dab-config.json > ...

```jsonc
{
  "$schema": "https://github.com/Azure/data-api-builder/releases/download/v0.10.23/dab.draft.schema.json",
  "data-source": {
    "database-type": "mssql",
    "connection-string": "Server=tcp:kkdemosqldbserver.database.windows.net,1433;Initial Catalog=Demo;Persist Security Info=False;User ID=sql
    "options": {
      "set-session-context": false
    }
  },
  "runtime": {
    "rest": {
      "enabled": true,
      "path": "/api",
      "request-body-strict": true
    },
    "graphql": {
      "enabled": true,
      "path": "/graphql",
      "allow-introspection": true
    },
    "host": {
      "cors": {
        "origins": [],
        "allow-credentials": false
      },
      "authentication": {
        "provider": "StaticWebApps"
      },
      "mode": "development"
    }
  },
  "entities": {
    "Author": {
      "source": {
        "object": "dbo.authors",
        "type": "table"
      },
      "graphql": {
        "enabled": true,
        "type": {
          "singular": "Author",
          "plural": "Authors"
        }
```

DAB Configuration File – Data Source

- mssql
  - For Azure SQL DB, Azure SQL MI, SQL Server
- postgresql
  - For PostgreSQL
- mysql
  - For MySQL
- cosmosdb_nosql
  - For Cosmos DB NoSQL API (JSON documents)
- cosmosdb_postgresql
  - For Cosmos DB PostgreSQL API

DAB Configuration File – Runtime global settings

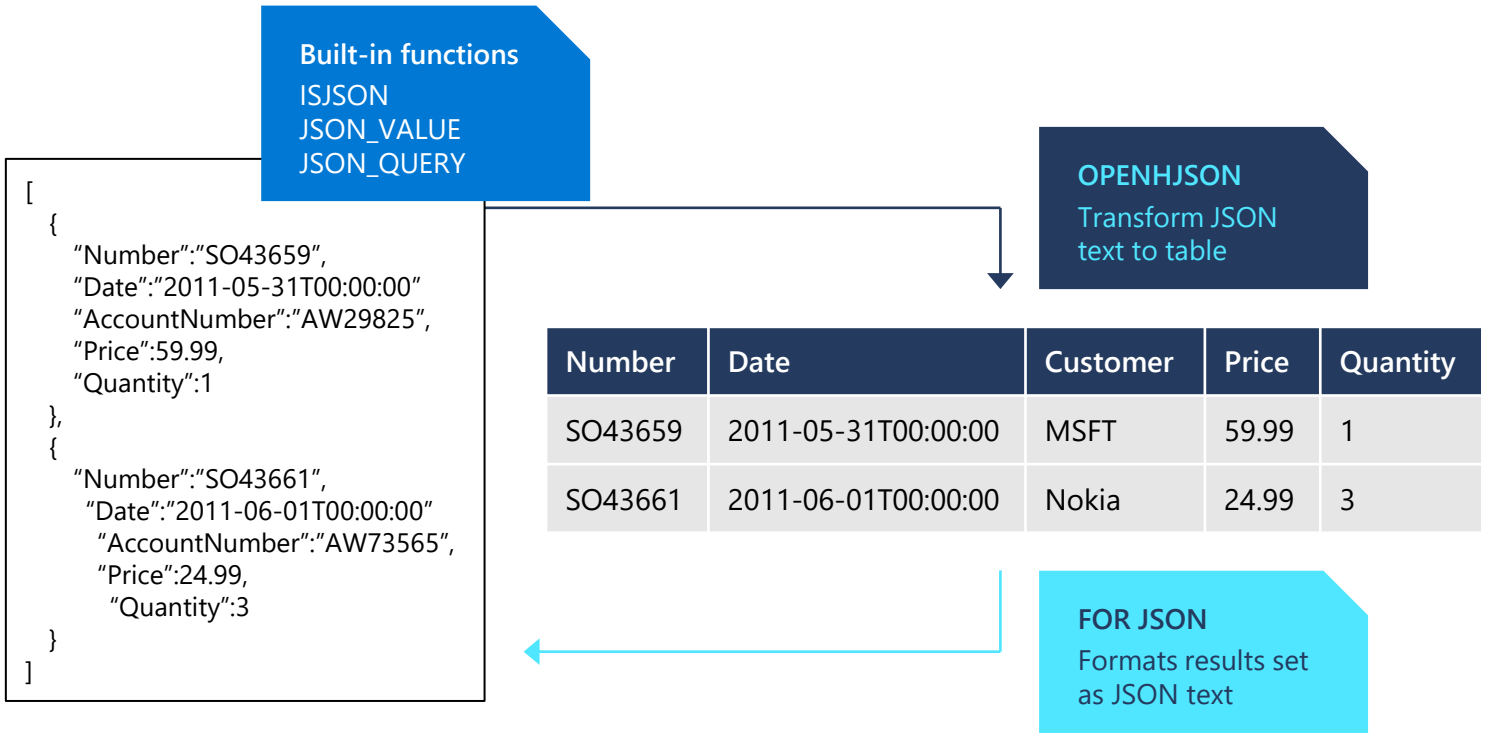
- rest
  - Path for REST endpoints
- graphql
  - Path for GraphQL endpoints
- host
  - Mode (prod/dev), CORS, authentication

DAB Configuration File – Entities

- <entity-name>
  - Entity name exposed by the data API
- source
  - Physical database object name (table, view, stored procedure)
- rest
  - Expose entity via REST (true/false)
- graphql
  - Expose entity via GraphQL (true/false)
- permissions
  - Array of allowed role/actions for the entity

Azure SQL Database supports JSON

Storage, manipulation and creation of JSON documents supported



Process JSON data with native JSON capabilities

Azure SQL allows developers to use JSON for both exchanging data with the database and storing JSON documents.

SQL

```
DECLARE @json NVARCHAR(MAX) = '[{"user":{"id": 1, "name": "Oliver"}},{"user":{"id": 2, "name":
"Kostja"}}]'
SELECT * FROM
    OPENJSON(@json) WITH (
        [Id] INT '$.user.id',
        [Name] NVARCHAR(100) '$.user.name'
    )
```

Microsoft SQL support for JSON

JSON can be stored (as string) and manipulated since SQL Server 2016

- FOR JSON operator
- OPENJSON operator
- ISJSON, JSON_VALUE, JSON_QUERY, JSON_MODIFY built-ins

In SQL Server 2022 (and Azure SQL) more improvements to JSON support were shipped

- Improved ISJSON built-in
- New JSON_PATH_EXISTS, JSON_OBJECT and JSON_ARRAY built-ins

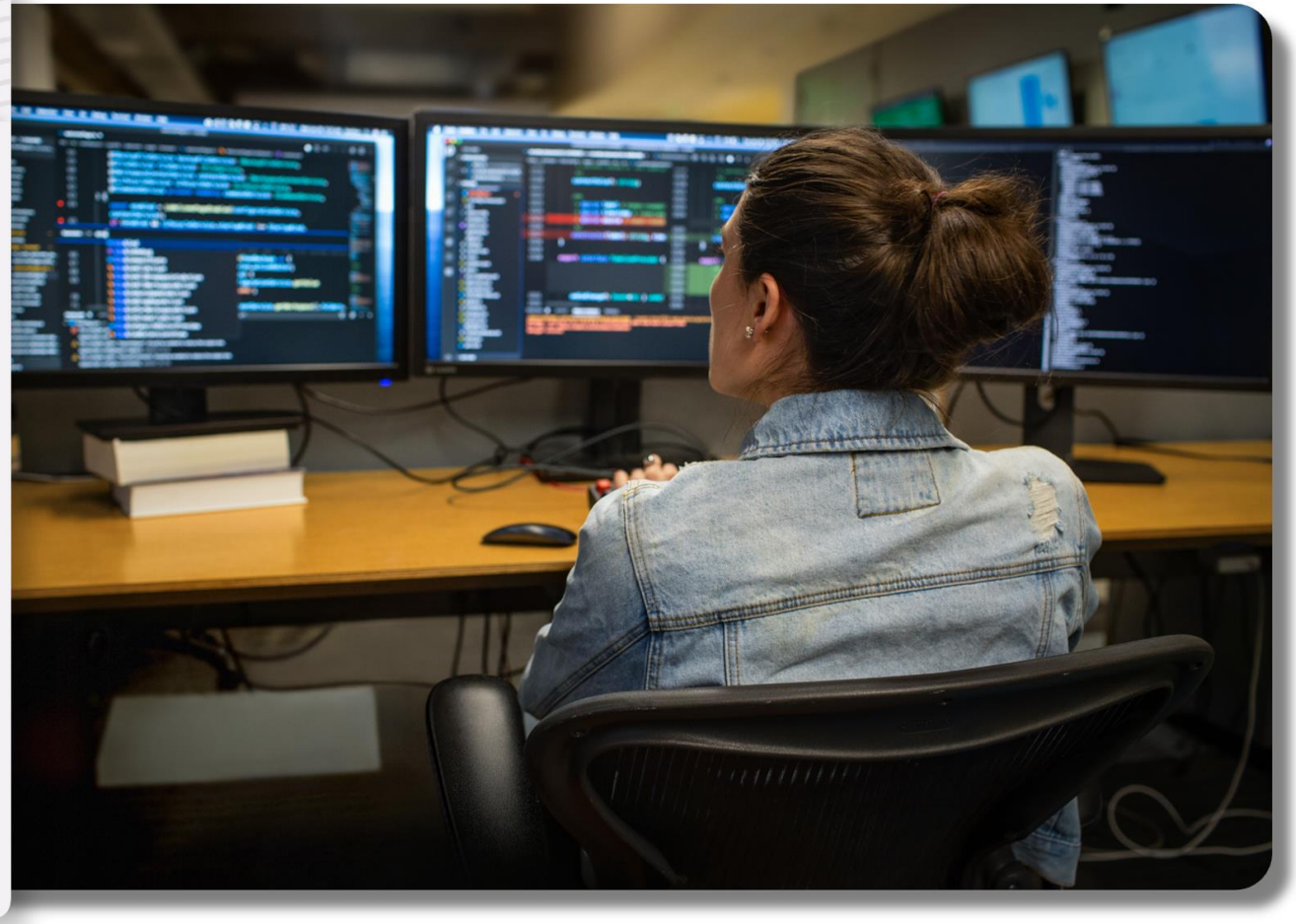
Azure SQL DB support for JSON will get even better …

Announcing

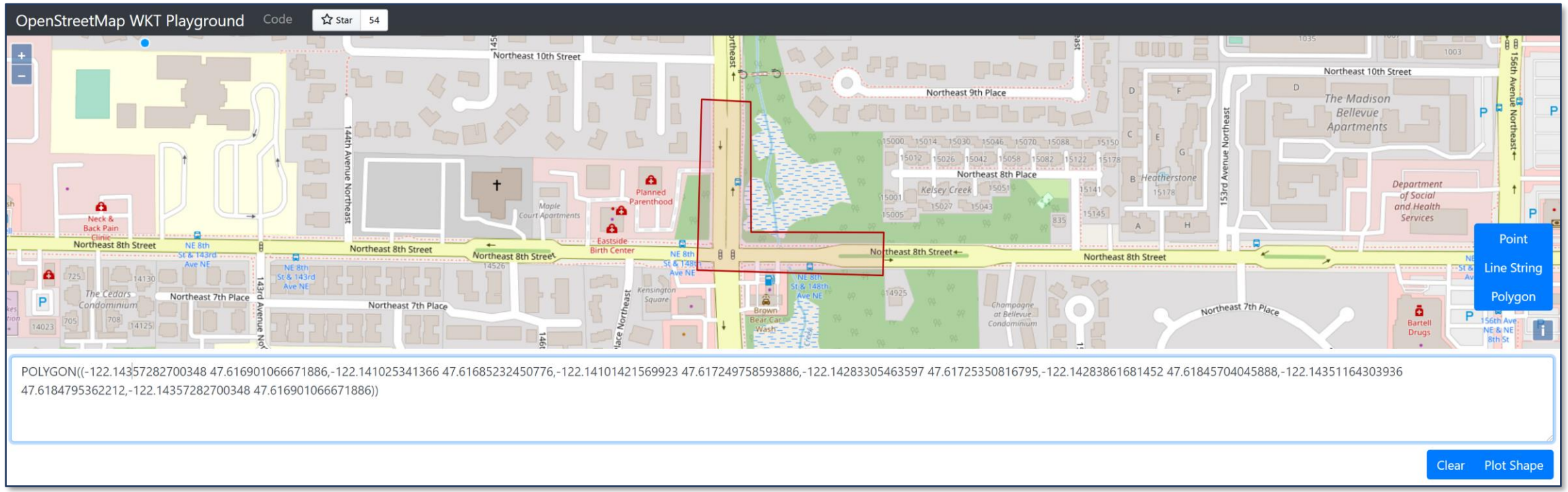
Build 2023 announcement: JSON data type is available for private preview

On Azure SQL DB only at the moment

aka.ms/json-preview



Azure SQL Database supports spatial



Analyze geospatial data with the native geospatial engine

Perform geospatial queries without installing external libraries or moving data, which could be expensive for significant volumes of data.

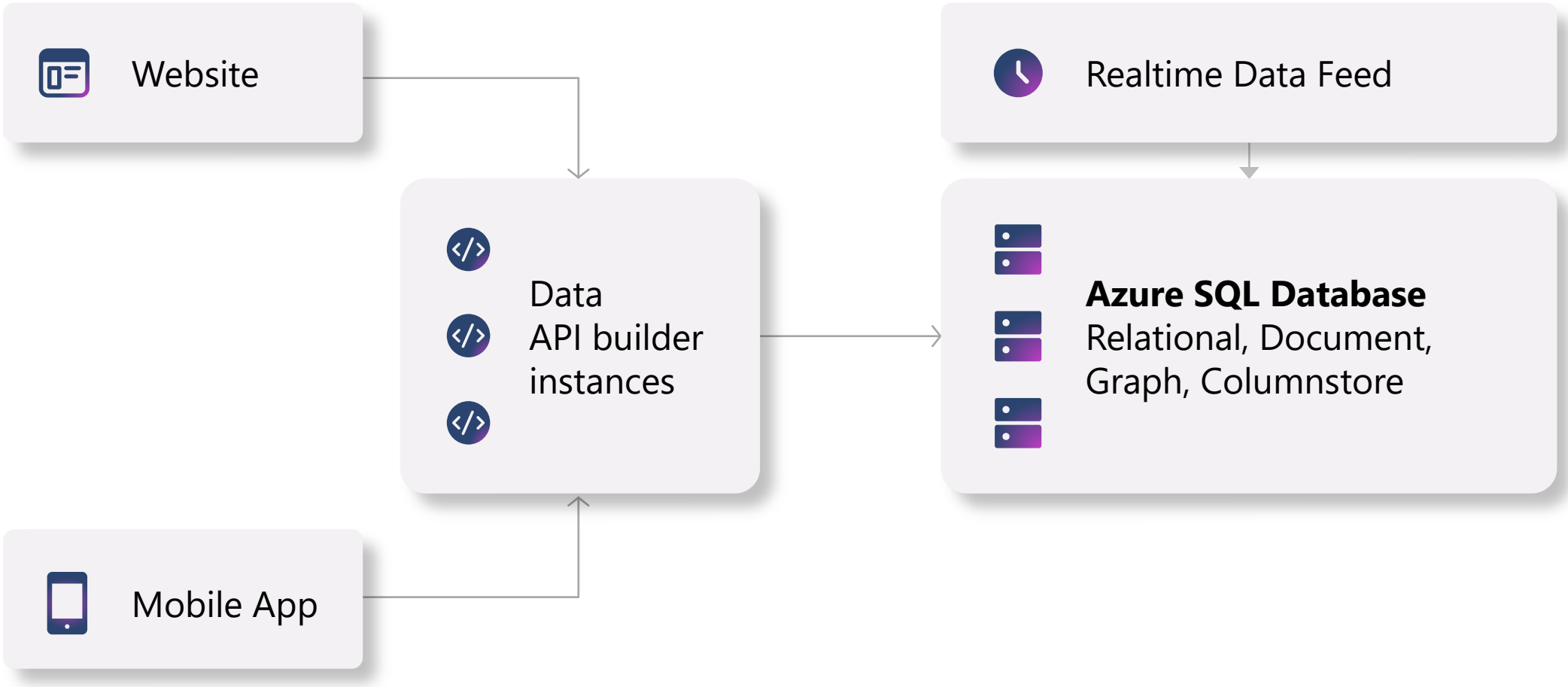
SQL

```
-- Microsoft Building 44
declare @bldg44 as geography = geography::STGeomFromText('POLYGON((-122.13403149305233
47.64136608877112,-122.13398769073248 47.64058891712273,-122.13319924946629 47.64011342667547,-
122.13285369830483 47.640106868176076,-122.13254221532335 47.640834858789844,-122.13257628383073
47.6410086568205,-122.13334039023833 47.64144150986729,-122.13403149305233 47.64136608877112))', 4326)

-- A point you want to check
declare @p as geography = geography::STPointFromText('POINT(-122.13315058040392 47.64101193601368)', 4326)

-- Is the point within the perimeter?
select @p.STWithin(@bldg44)
```

Full architecture overview with Azure SQL & DAB



SQL Konferenz 2024
powered by PASS Deutschland e.V.

HOME | REGISTRATION | AGENDA | CALL FOR SPEAKERS | SPEAKER | PARTNERS | LOCATION | CODE OF CONDUCT

SQL KONFERENZ 2024
September 30 – Oktober 2nd
Congress Park Hanau

Registration is open!

Release Candidate (until 15.06.2024)

This is going to be damontastic!

SQL Konferenz 2024 is the event par excellence for data(base) professionals, SQL (Server) enthusiasts and data architects. Experts and interested parties meet here to expand their knowledge and exchange information about the latest developments in the SQL Server community. There are exciting talks, workshops, and discussion panels on topics such as database design, performance optimisation, business intelligence, security, cloud integration and even artificial intelligence.This unique conference is also a great way to meet new people, network, and chat with other industry experts. Along the way, you can also improve your skills and strengthen the SQL Server community in Germany. What are you waiting for? Damon and his datamonster fellows are eager to meeting you in Hanau in September.

PreCons ahead!

This year's PreCon programme has it all! The illustrious round of experts holding PreCon workshops couldn't be more fabulous. The PreCon workshops are full-day deep dives into different topics. There is something for every SQL enthusiast and data architect. You have only booked MainCon? An upgrade to FullCon is easily possible and definitely a good investment. Check the PreCon Sessions out!

Registration is open!
Datamonsters will be back!

Learn, gain insights, and connect with other experts on SQL , Azure, Queries, AI, and much more…

Register now →

BECOME A PARTNER OR SPONSOR!

Are you interested in our partner packages and sponsoring options? There are plenty beautiful ways to support the SQL Konferenz 2024.

Download Partner Package Information

partner@sqlkonferenz.de →

Key takeaways

- Modern applications are enabled through APIs and JSON
- SQL Server and Azure SQL DB can be used to consume or create REST APIs
- They also have native capabilities for JSON and geospatial data requirements
- Have a great day today at #DataGrillen

DataGrillen
Datenbanken
Bratwurst
Bier

Evaluations PLEASE!

QUESTIONS???

Speakerinfo: Constantin „Kostja“ Klein

Constantin arbeitet bei der Syntax Systems GmbH & Co. KG. Dort beschäftigt er sich mit der Entwicklung von Lösungen auf Basis der Microsoft Produkte und Technologie Plattformen. Sein besonderes Interesse gilt dabei der Gewinnung, Verarbeitung und Nutzung von Daten und Informationen. Seit 2010 wurde er jährlich von Microsoft zum Most Valuable Professional (MVP) für die Microsoft Data Platform und seit 2020 auch zum Regional Director (RD) ernannt. Er engagiert sich zusätzlich im Vorstand des Just Community e.V. und des PASS Deutschland e.V.

Sein Blog finden Sie unter

http://kostjaklein.wordpress.com