Microsoft

Azure Serverless Conf

Serverless web applications with GraphQL, Cosmos DB and Azure Functions ⚡

Microsoft

Speaker

Jan-Henrik Damaschke

CTO / Senior Cloud Architect @ Visorian

Microsoft MVP for Azure

@JanDamaschke

/in/jan-henrik-damaschke

itinsights.org

Azure Serverless

Conf

Microsoft

Why serverless?

- **Scalability** - Scale-as-you-go by time, user count, requests, etc.
- **Performance** - From edge workers over caching to server side rendering, everything is possible
- **Pay-as-you-go** - Pricing scales with demand
- **Maintenance** - Just run your code and let Azure take care of the rest
- **CI/CD** - Deploy your code easier and react to changes rapidly
- **Developer experience** - Separation and decoupling of services, choice of language
- **Security** - Serverless has the potential to reduce your attack surface and introduce modern security standards

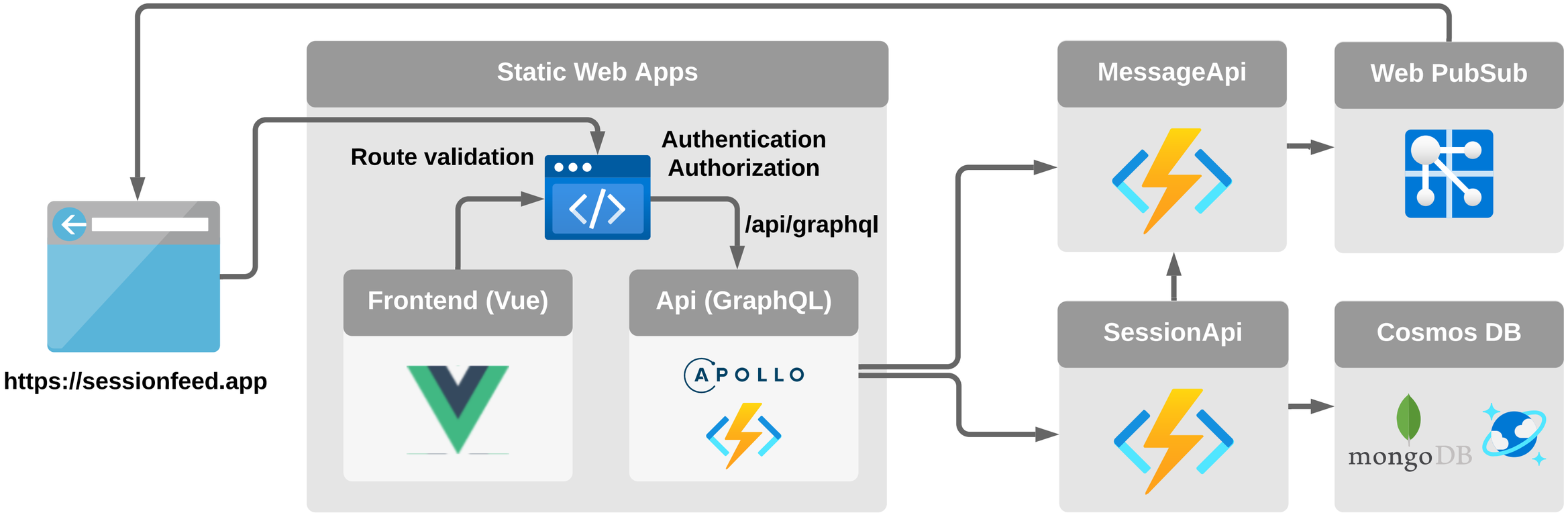
Microsoft

Static Web Apps

- Seamless security model with a reverse-proxy with customizable routes
- Integrated CDN
- First-party GitHub integration
- Free SSL certificates and custom domains(yes, on domain apex)
- Built-in Authentication provider like AAD, GitHub, Twitter, etc.
- Pull requests previews
- Free basic plan for static content including Azure Functions
- Bring your own functions and custom OpenID Connect provider in standard plan
- 100 GB bandwidth limitation per subscription per month

Microsoft

Sessionfeed.app



Azure Serverless
Conf

Microsoft

Architecture overview

Component	Azure Service
Frontend	Azure Static Web Apps Vue.js
Backend	Azure Static Web Apps (Azure Functions)
Database	CosmosDB (Mongo API)
Microservices	Azure Functions
Publish/Subscribe	Azure Web PubSub

Microsoft

Why GraphQL

- Query language for APIs
- Shared definitions of resources
- Strongly connected entities coming from multiple APIs Central API Gateway
- Works great in serverless
- Authorization and Authentication support
- Rich ecosystem

Limitations with serverless

- No subscription support
- GraphQL API only usable with bring your own app in Static Web Apps
- Performance (kind of)
- Authentication/Authorization can be difficult

Microsoft

Cosmos DB

 Demo Time 

Azure Serverless
Conf

Microsoft

Session API

sessionApi/getComment (simplified)

```
const { id } = req.query;
const question: Question = await makeMongoQuery(
  sessionfeedbDbName,
  questionsCollectionName,
  "findOne",
  [{ _id: id }],
  0,
  mongoConfiguration,
  cacheConfiguration
);
context.res = {
  body: question : undefined,
};
```

Microsoft

GraphQL Datasource

graphql/datasources/SessionAPI (simplified)

```
export class SessionAPI extends RESTDataSource {
this.baseURL = sessionApiBaseUrl;

willSendRequest(request: RequestOptions) {
  request.headers.set("x-functions-key", sessionApiFunctionKey);
}

async getQuestion(id: string): Promise<Question> {
  const result = await this.get("/getQuestion", {
    id,
  });
  return this.convertQuestionFields(JSON.parse(result));
}
 ...
```

Microsoft

GraphQL typing

```
type Question {
  id: ID!
  comments: [Comment]!
  commentCount: Float!
  created: DateTime!
  likeCount: Float!
  liked: Boolean!
  likedBy: [String!]!
  modified: DateTime!
  talk: Talk!
  ...
}
```

```
export class Question {
  @Field((type) => ID)
  id: string;
  @Field({ nullable: "items" })
  comments: Comment[];
  @Field()
  created: Date;
  @Authorized(["ADMIN"])
  @Field()
  talkId: string;
  @Authorized(["ADMIN"])
  @Field()
  modified: Date;
}
```

```
@Resolver((of) => Question)
export class QuestionResolver {
  ...
  @FieldResolver()
  likeCount(@Root() question: Question): number {
    return question.likedBy ?
      question.likedBy.length : 0;
  }
  ...
}
```

Azure Serverless

Conf

Microsoft

Authentication / Authorization

```
{
  "routes": [
    {
      "route": "/talks/*",
      "allowedRoles": [
        "authenticated",
        "admin",
      ]
    },
    {
      "route": "/api/*",
      "allowedRoles": [
        "authenticated"
      ]
    },
    {
      "route": "/login/github",
      "rewrite": "/.auth/login/github"
    },
    {
      "route": "/logout",
      "redirect": "/.auth/logout",
      "statusCode": 301
    }
  ],
  "navigationFallback": {
    "rewrite": "index.html",
    "exclude": [
      "/@vite/*",
      "/assets/*",
      "/.vite/*"
    ]
  },
  "responseOverrides": {
    "401": {
      "redirect": "/login",
      "statusCode": 302
    }
  ...
```

Azure Serverless Conf

Microsoft

Access user information

Frontend

```
const {
  data: swaUser,
  isFetching: swaUserLoading,
  error: swaUserError,
  execute: refetchSwaUser,
  onFetchResponse,
} = useFetch("/.auth/me").get();
```

```
async function getUserInfo() {
  const response = await fetch("/.auth/me");
  const payload = await response.json();
  const { clientPrincipal } = payload;
  return clientPrincipal;
}
```

Backend

```
const context = async ({ request }: any) => {
  try {
    const header = request.headers["x-ms-client-principal"];
    const encoded = Buffer.from(header, "base64");
    const decoded = encoded.toString("ascii");
    const swaUser: SwaUser = JSON.parse(decoded);
    return { swaUser: swaUser };
  } catch (error) {
    throw new AuthenticationError(
      "This API can only be accessed through Azure Static Web Apps"
    );
  }
};
```

Microsoft

What else?

- Durable Function -> Next talk
- No config management
- No secrets management
- No premium plans
- No caching
- Advanced architecture

Microsoft

Takeaways

- Know your workloads -> Scalability targets
- Know your culture -> Dev(Sec)Ops
- Know your code -> Testing, testing testing
- Know your risks -> Think security from the beginning on
- Know your cloud platform -> Mix FaaS and PaaS services
- Know your environment -> Think about local development and testing
- Know your budget -> Invest time to plan for scalable service and configure alerts (e.g. CosmosDB)

Microsoft

Questions?

Ask your questions at https://sessionfeed.app

If you want to create great development presentations like this one, use https://sli.dev by Anthony Fu @antfu7