Service-to-service authentication in a microservice deployment

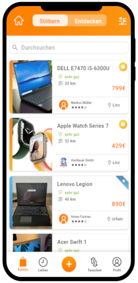
UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA

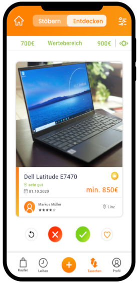
FH Hagenberg, WS 2021/2022
Benjamin Ellmer

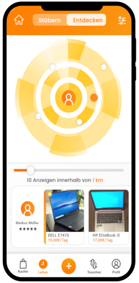
Bachelor Thesis Presentation

Swapindo

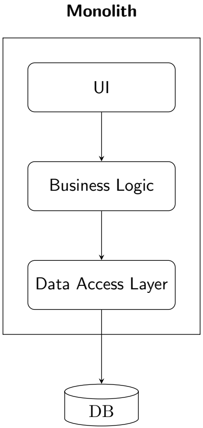
- Flea market app developed in android (so far)
- Easy and playful swapping
- Simple and safe lending and renting
- Fast and secure buying and selling
- All-in-one-platform as mobile application
- Backend based on the microservice architecture

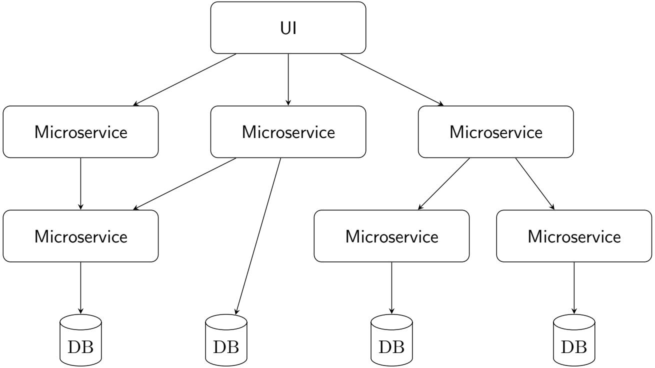






Monolithic Architecture vs. Microservice Architecture

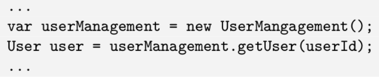




Microservice Architecture

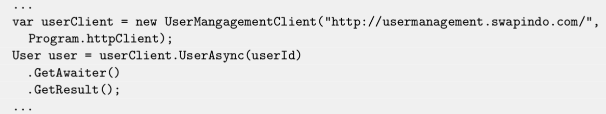
- All approaches that could centralize business logic are avoided
- Declaration of service boundaries is very hard
- Implementation details are hidden, to avoid coupling between services
- Failures can be isolated
- Language level calls become remote calls

Language-Level Call:



```
...
var userManagement = new UserMangagement();
User user = userManagement.getUser(userId);
...
```

Remote Call:



```
...
var userClient = new UserMangagementClient("http://usermanagement.swapindo.com/",
  Program.httpClient);
User user = userClient.UserAsync(userId)
   .GetAwaiter()
   .GetResult();
...
```

Remote Calls and Authentication

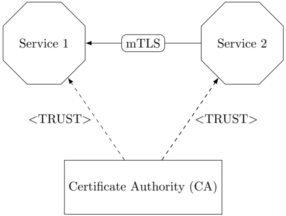
Term Description

Confidentiality guarantees that only the sender and the receiver are allowed to read and understand the communication between them.
Integrity guarantees the trustworthiness and correctness of data, by providing the ability to detect unauthorized modifications.
Authentication is the process of identifying the communication partner to protect a system from spoofing.

- Remote calls have to provide confidentiality, integrity and authentication
- Confidentiality and integrity are assured using TLS
- TLS also authenticates the server to the client, but not the client to the server
- Most popular service-to-service authentication mechanisms are mutual TLS and self-signed JWTs
- Trust the network is a deprecated approach for service-to-service authentication

Mutual TLS

- Each service owns a certificate, signed by a Certificate Authority
- The certificates are exchanged during the TLS-handshake
- When the signature of the certificate is valid, the service is trusted
- Efficient and straightforward
- Based on the TLS protocol, therefore hard to adapt for other purposes like sharing the user-context



Service 1

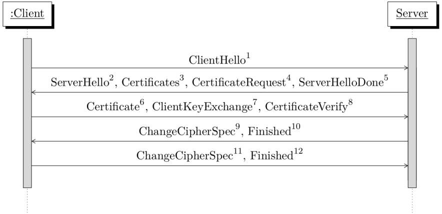
mTLS

Service 2

<TRUST>

<TRUST>

Certificate Authority (CA)



:Client

Server

ClientHello[1]

ServerHello[2], Certificates[3], CertificateRequest[4], ServerHelloDone[5]

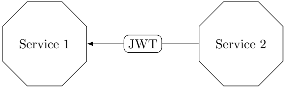
Certificate[6], ClientKeyExchange[7], CertificateVerify[8]

ChangeCipherSpec[9], Finished[10]

ChangeCipherSpec[11], Finished[12]

Self-signed JWT

- Each service owns a key pair
- A JWT signed with the private key has to be embedded within the Authorization header
- Additional parameters can be embedded within the JWT
- User-context is usually shared using a nested JWT
- Nonrepudiation can be achieved by storing the received JWTs and requests



```
eyJhbGciOiJSUzI1NiIsImtpZCI6IkYwMURFNTBCNTU5MzVBQ0VEOUNDNzdCRjR
FMjY5NkVDOTE4MUI5NjkiLCJ0eXAiOiJKV1QifQ. eyJuYmYiOjE2NDU1MjU5MjM
sImV4cCI6MTY0NTUyNjIyMywiaXNzIjoic2VydmljZTEuc3dhcGluZG8uY29tIiwiYXV
kIjoic2VydmljZTIuc3dhcGluZG8uY29tIn0.
```

```
{
        "alg": "RS256",
        "kid": "F01DE50B55935ACED9CC77BF4E2696EC9181B969",
        "typ": "JWT"
}
{
        "nbf": 1645525923,
        "exp": 1645526223,
        "iss": "service1.swapindo.com",
        "aud": "service2.swapindo.com"
}
```

Comparison Result

mTLS:

- Each service owns a key pair

Self-signed JWT:

- Each service owns a key pair

Best authentication mechanism for your project

- Is the microservice architecture fitted for your project?
- Is nonrepudiation an requirement ? → **self-signed JWT**
- Does your application tend to share the user-context? → **self-signed JWT**
- Does your authentication require custom parameters ? → **self-signed JWT**
- None of the previous reasons ? → **mTLS**

Presenting a regular slide

This is a regular slide

Block title

And a basic block for emphasizing important information, using **`\fhblock{Title}{Text}`** . Below is an example of the **`\fhdualpage`** command, which lets you define the pane proportions and content for each pane.

This text will be on the left side of the dual-page, and takes up 40 % of the pagewidth. Lorem ipsum dolor sit amet, consectetur adipisici elit, ...

This text will be on the right side of the dual-page, and takes up 60 % of the page width. Lorem ipsum dolor sit amet, consectetur adipisici elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua.

Presenting a list-slide with \fhslist

- Many slides actually contain only multiple bullets, rather than text in paragraphs
- In order to keep it simple, theres the `\fhslist` command, which begins a new slide and opens an itemize environment for your convenience.
- After starting the slide with the command above, you can immediately continue to provide `\item` bullets.
  - For sub-bullets, there's no need for begin/end blocks either. Simply use `\fhlist` command and drop your `\item` s.
  - By the way, monospaced and grey-backgrounded text can be done with `\cmd{code-like text}`

Centered text via `\ctr{}`

\fhtopicslide{content} renders a new slide, e.g. for a new topic, with large bold font and vertically centered text.

Images, essential slides, and other macros

- There are a lot more easy-to-use macros
- Just look at the first 40-something lines of the template tex source file
- There are commands for URL references, including images, quad-pages, aligned math equations, fragile slides (e.g. for lstlistings), etc.
- Easy image inclusion with `\fhimg[width]{path}`

- See the burning-hot icon on the top left? You can mark slides as important using the `\essential` command.
- Theres also a `\moot` command, if you want to tag supplementary slides. Of course, the images can be easily changed - just change the filepath in the source, or overwrite the image itself in the template folder.

That’s it - have fun

- Just look at the provided `example.tex` file, how all these commands are used in-source – it’s really easy
- For a more complete list of available commands and macros, look at the template source-code
- If you find errors, have suggestions or even patches for new commands and macros you think others can benefit from, don’t hesitate to tell me: `erik.sonnleitner@fh-hagenberg.at`