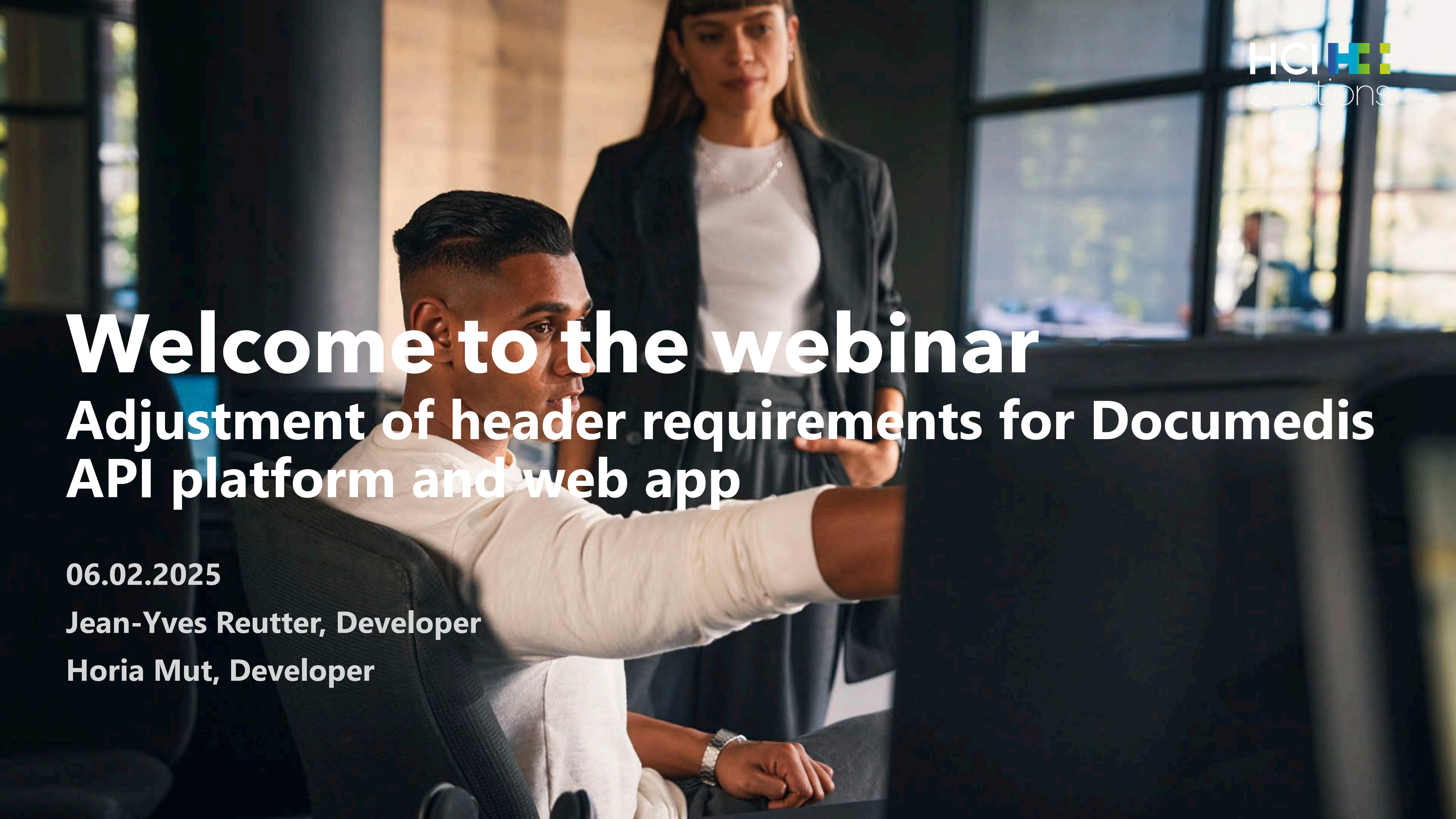


Welcome to the webinar

Adjustment of header requirements for Documedis API platform and web app

06.02.2025

Jean-Yves Reutter, Developer

Horia Mut, Developer

Agenda

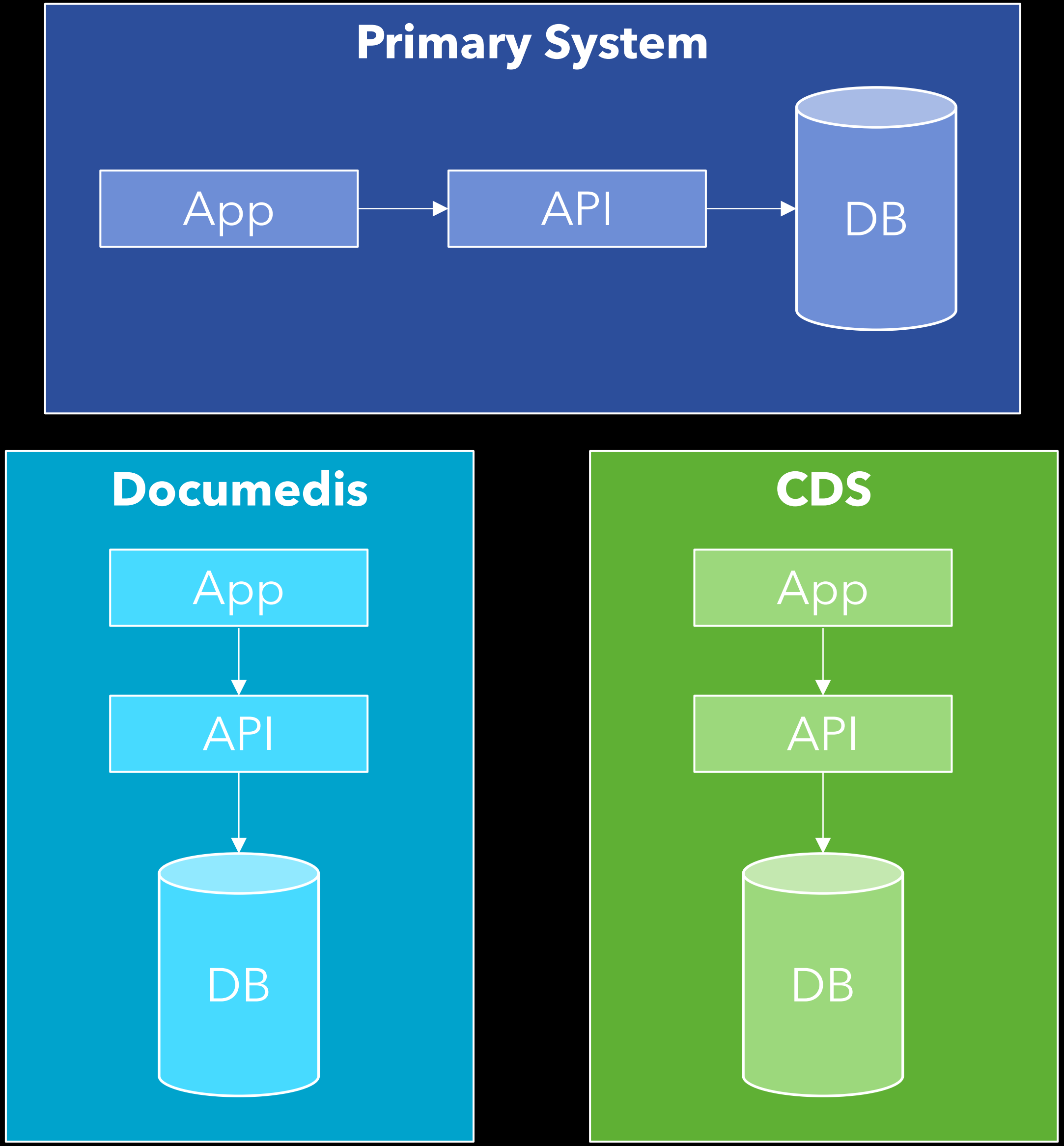
1. Various integrations of Documedis & CDS
   - –Light VS deep integration
   - –Resources
2. Documedis authentication
   - –Difference between our environments
   - –HTTP Headers
3. Some examples
   - –with Swagger
   - –with Curl
4. Questions
5. Miscellaneous

The session is being recorded

Various integrations of Documedis & CDS

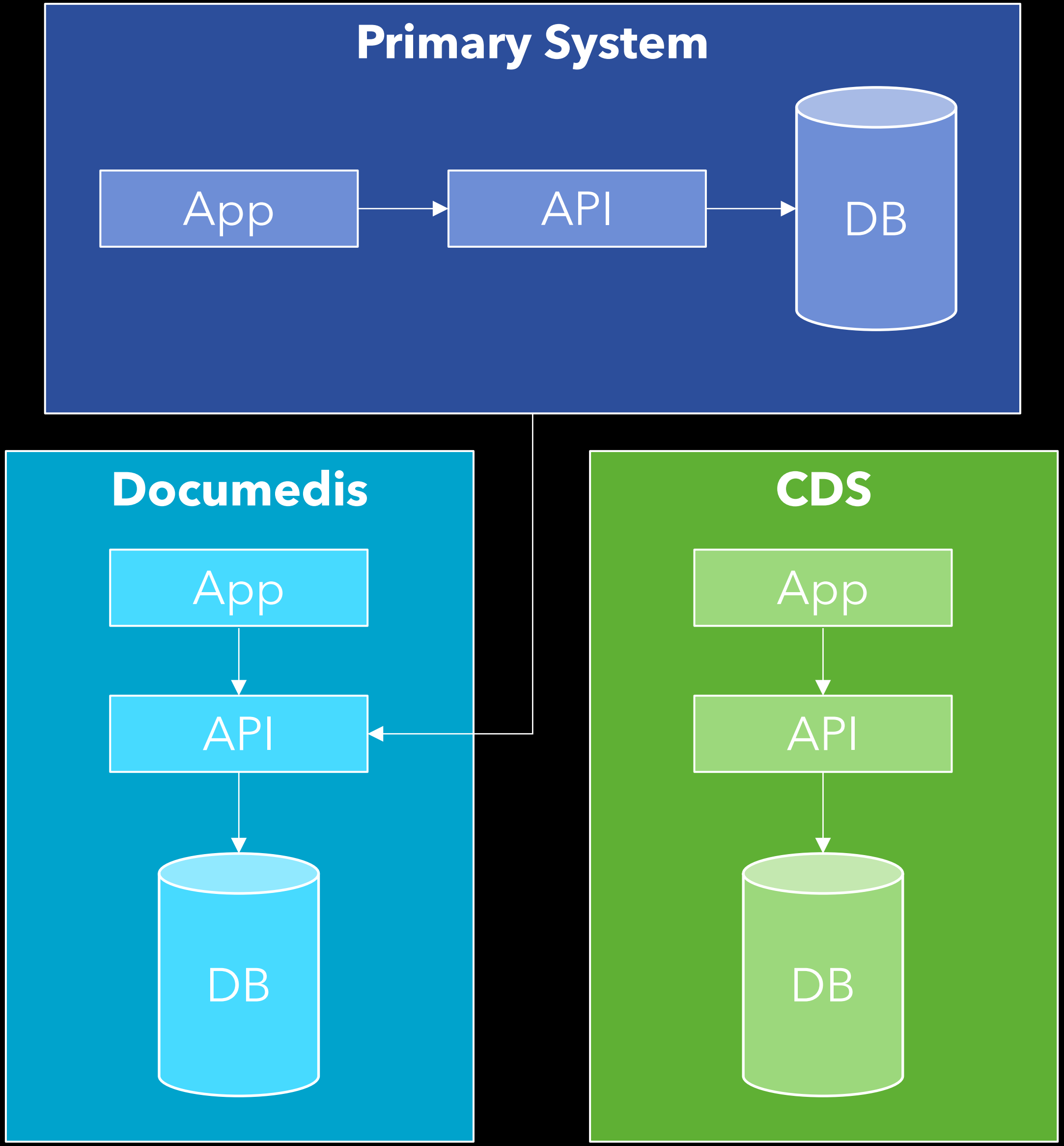
Jean-Yves Reutter, Developer

Architecture



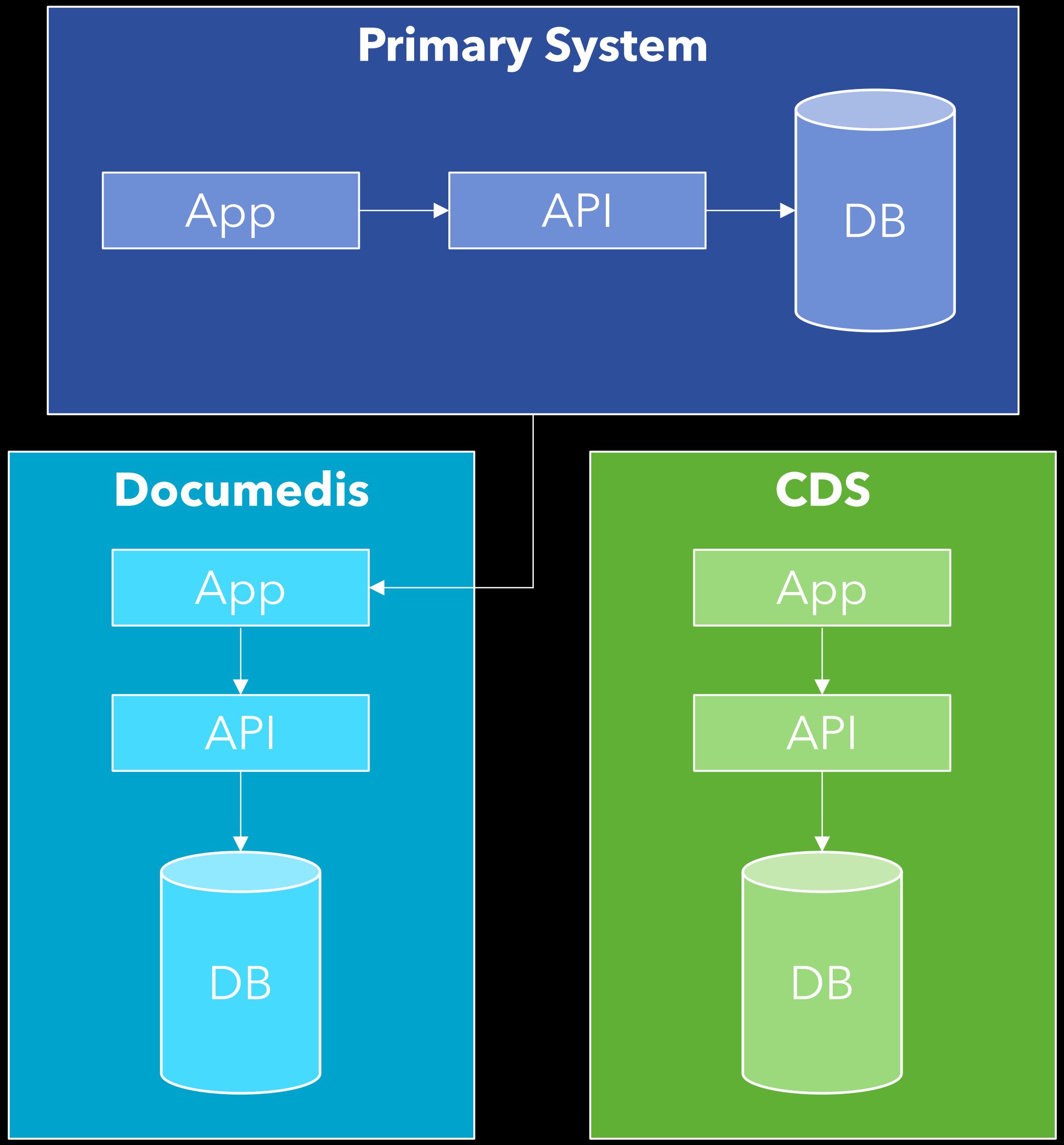
Integration with Documedis API

Deep Integration



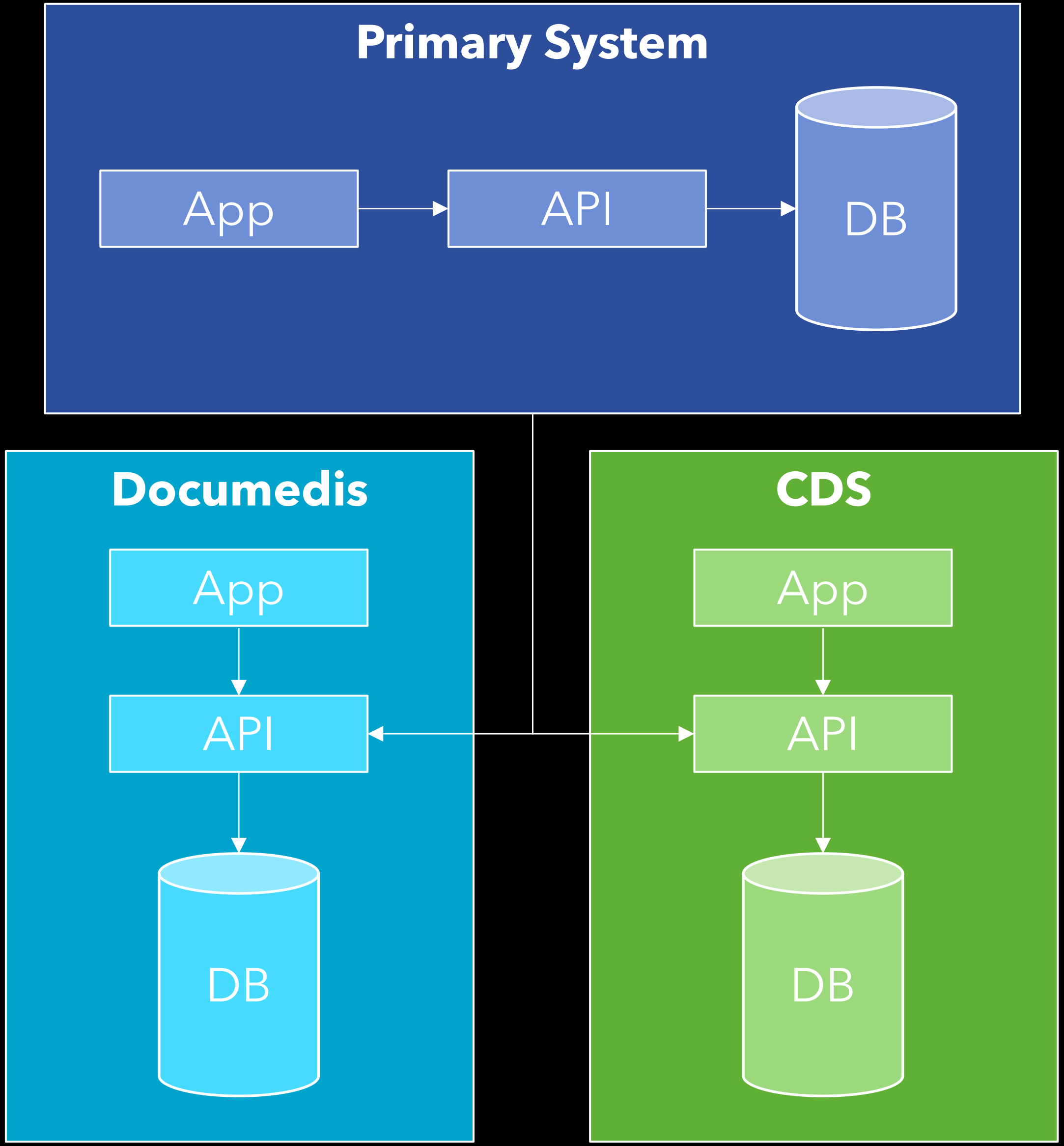
Integration with Documedis App

Light Integration

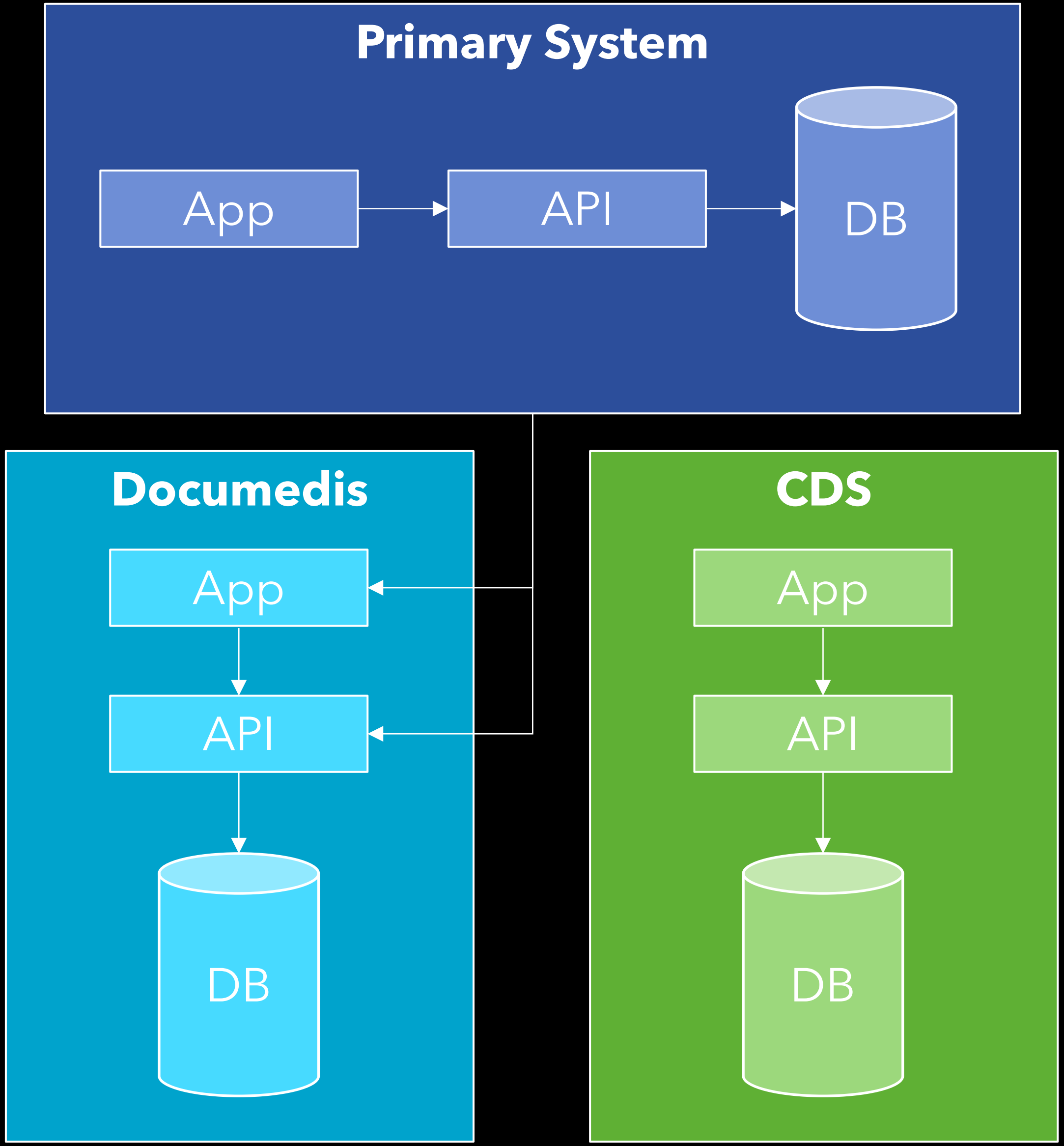


Integration with Documedis API and CDS API

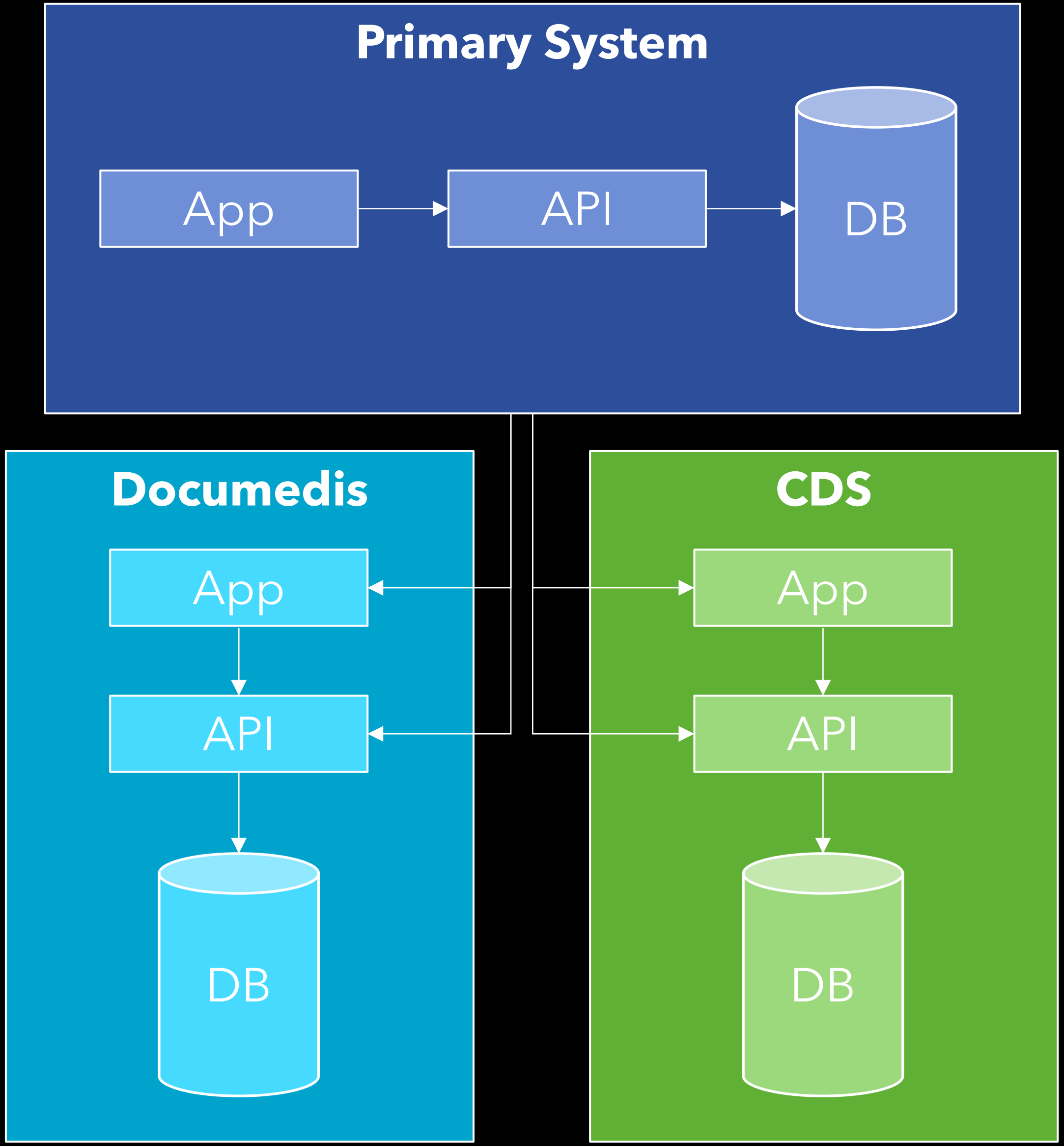
Deep Integration



Mixed Integration

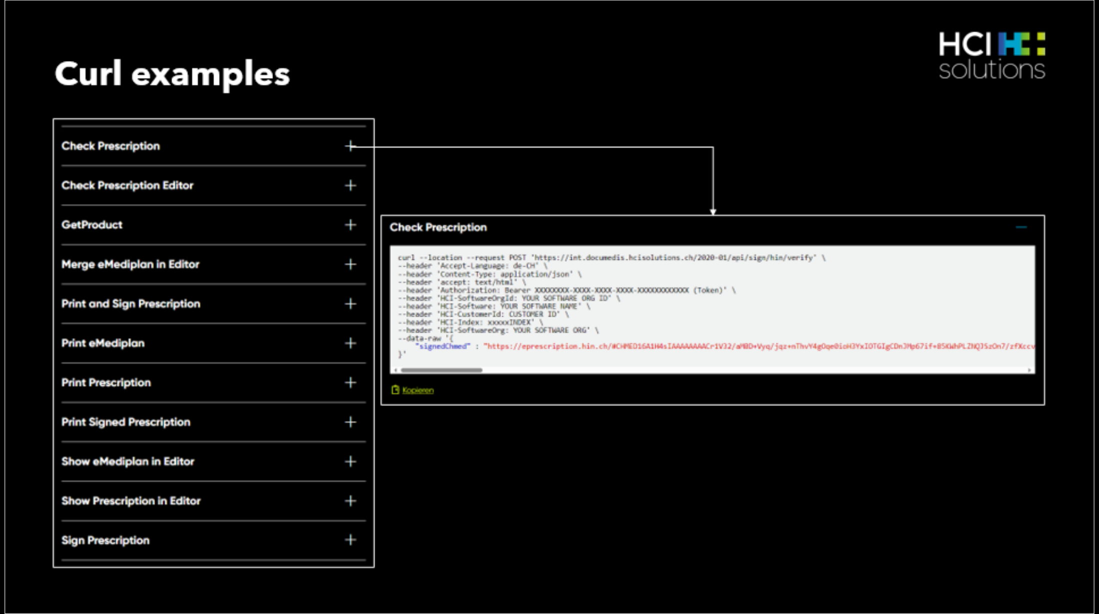


Mixed Integration with CDS



HCI Solutions’ Support pages

Manuals and examples
https://www.hcisolutions.ch/de/support/dokumentationen/manuals.php

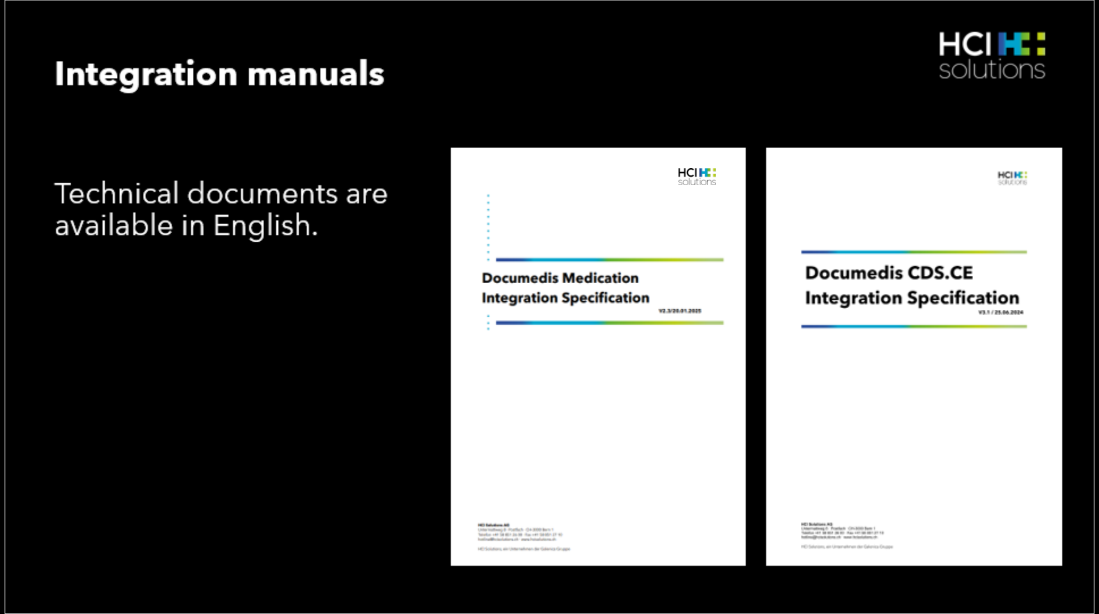


Curl examples

- Check Prescription +
- Check Prescription Editor +
- GetProduct +
- Merge eMediplan in Editor +
- Print and Sign Prescription +
- Print eMediplan +
- Print Prescription +
- Print Signed Prescription +
- Show eMediplan in Editor +
- Show Prescription in Editor +
- Sign Prescription +

Check Prescription

```
curl --location --request POST 'https://int.documedis.hcisolutions.ch/2020-01/api/sign/hin/verify' \
--header 'Accept-Language: de-CH' \
--header 'Content-Type: application/json' \
--header 'accept: text/html' \
--header 'Authorization: Bearer XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (Token)' \
--header 'HCI-SoftwareOrgId: YOUR SOFTWARE ORG ID' \
--header 'HCI-Software: YOUR SOFTWARE NAME' \
--header 'HCI-CustomerId: CUSTOMER ID' \
--header 'HCI-Index: xxxxxINDEX' \
--header 'HCI-SoftwareOrg: YOUR SOFTWARE ORG' \
--data-raw '{
    "signedChmed" : "https://eprescription.hin.ch/#CHMED16A1H4sIAAAAAAAACr1V32/aMBD+Vyq/jqz+nThvY4gOqe0icH3YxIOTGIgCDnJMp67if+85KWhPLZNQJ5zOn7/zfXcc
}'
```



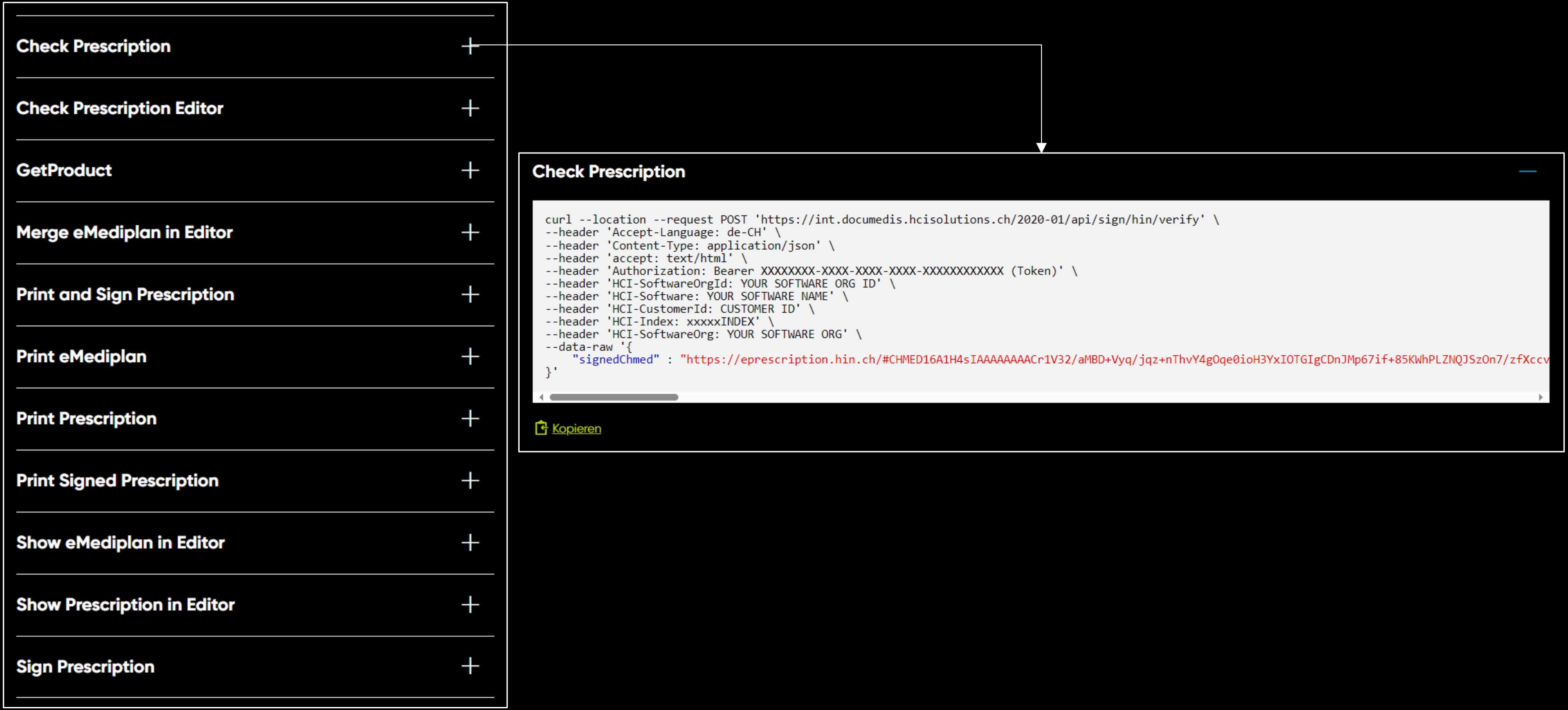
Integration manuals

Technical documents are available in English.

Documedis Medication Integration Specification
V2.3/20.01.2025

Documedis CDS.CE Integration Specification
V3.1 / 25.06.2024

Curl examples



Check Prescription +

Check Prescription Editor +

GetProduct +

Merge eMediplan in Editor +

Print and Sign Prescription +

Print eMediplan +

Print Prescription +

Print Signed Prescription +

Show eMediplan in Editor +

Show Prescription in Editor +

Sign Prescription +

Check Prescription —

```
curl --location --request POST 'https://int.documedis.hcisolutions.ch/2020-01/api/sign/hin/verify' \
--header 'Accept-Language: de-CH' \
--header 'Content-Type: application/json' \
--header 'accept: text/html' \
--header 'Authorization: Bearer XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX (Token)' \
--header 'HCI-SoftwareOrgId: YOUR SOFTWARE ORG ID' \
--header 'HCI-Software: YOUR SOFTWARE NAME' \
--header 'HCI-CustomerId: CUSTOMER ID' \
--header 'HCI-Index: xxxxxINDEX' \
--header 'HCI-SoftwareOrg: YOUR SOFTWARE ORG' \
--data-raw '{
    "signedChmed" : "https://eprescription.hin.ch/#CHMED16A1H4sIAAAAAAAACr1V32/aMBD+Vyq/jqz+nThvY4gOqe0ioH3YxIOTGIgCDnJMp67if+85KWhPLZNQJSzOn7/zfXccv
}'
```

Integration manuals

Technical documents are available in English.

Documedis Medication Integration Specification

V2.3/20.01.2025

HCI Solutions AG
Untermattweg 8 · Postfach · CH-3000 Bern 1
Telefon +41 58 851 26 00 · Fax +41 58 851 27 10
hotline@hcisolutions.ch · www.hcisolutions.ch

HCI Solutions, ein Unternehmen der Galenica Gruppe

Documedis CDS.CE Integration Specification

V3.1 / 25.06.2024

HCI Solutions AG
Untermattweg 8 · Postfach · CH-3000 Bern 1
Telefon +41 58 851 26 00 · Fax +41 58 851 27 10
hotline@hcisolutions.ch · www.hcisolutions.ch

HCI Solutions, ein Unternehmen der Galenica Gruppe

Documedis Authentication

Horia Mut, Developer

What you need to know

- The **Int**egration testing environment is protected against un-authenticated calls.
- Our **Prod**uction environment will follow suit.
- You will need to send us the correct HTTP Request with the **correct HTTP headers** on production to get in.
- You have time to adapt.

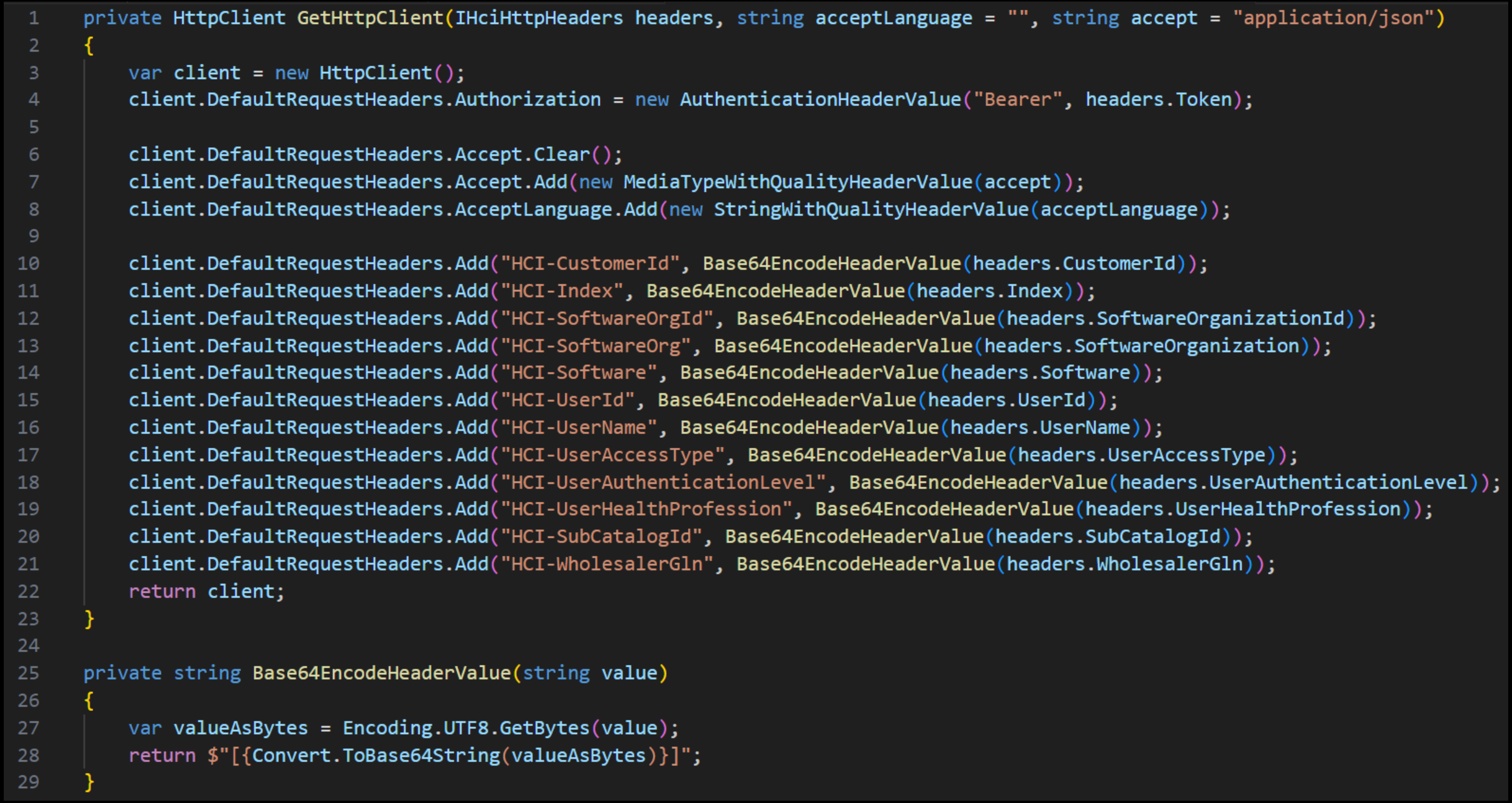
General information

- Communicating with Documedis requires the caller to define dedicated Headers of the HTTP Post/Get request.
- The headers are described in detail in the Integration Specification Documedis Medication manual mentioned previously. (Chapter 4.2 Integrating the Documedis medication software into your clinical information system).
- The specification is in English.
- Let’s go over the most important ones you must provide!

The important HTTP Headers

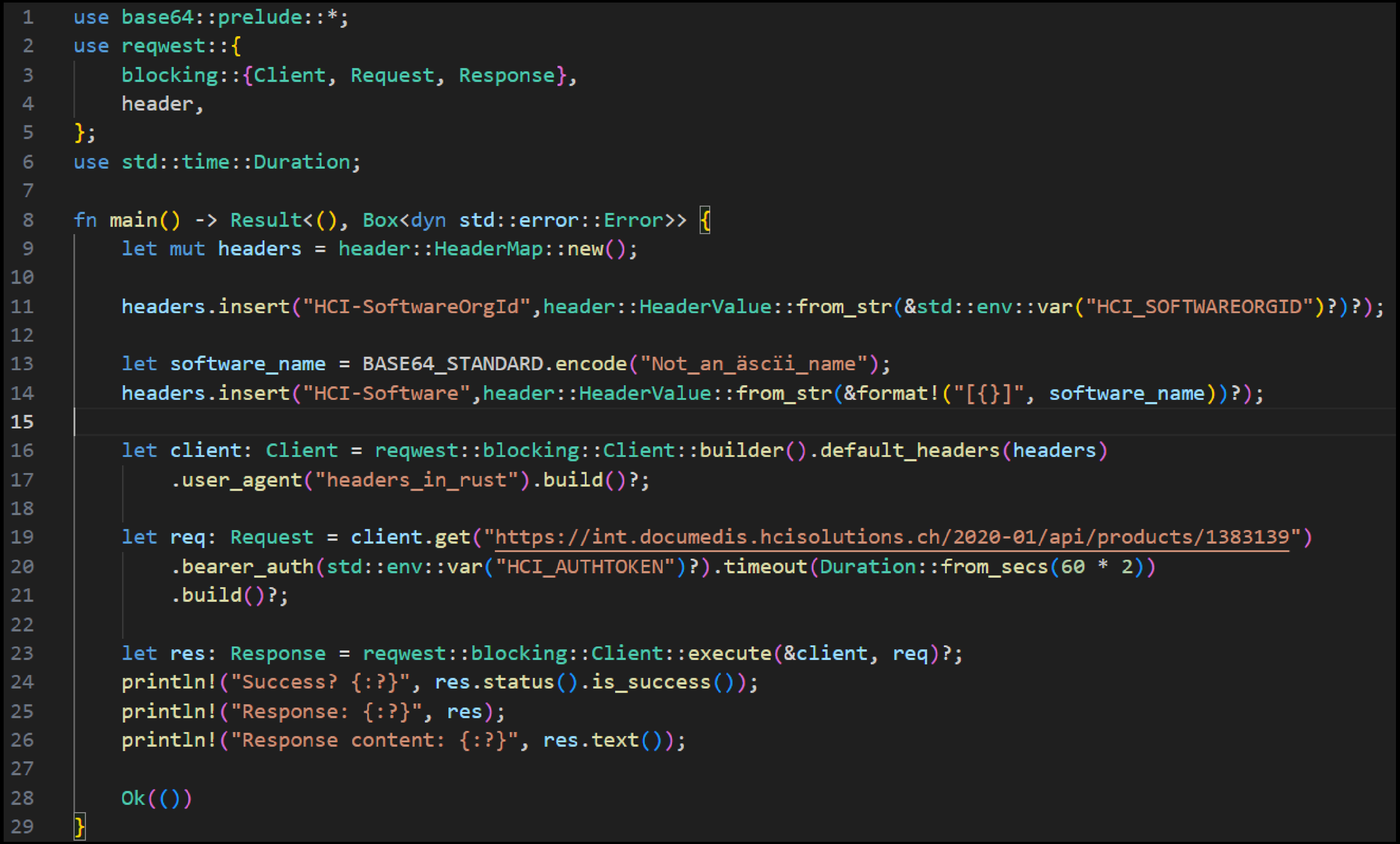
Header field	What is it	Example
HCI-CustomerId	GLN of the healthcare organization	7601234567890
HCI-SoftwareOrgId	PartnerId of the software manufacturer	1234
HCI-SoftwareOrg	Name of the software manufacturer	YourCompany
HCI-Software	Name of the application	NameOfYourApplication
HCI-Index	The Index you have a contract for	MedIndex
Authorization	The bearer token	Bearer 12345678-1234-1234-1234-123456789012

Providing the headers
Example 1 - .net



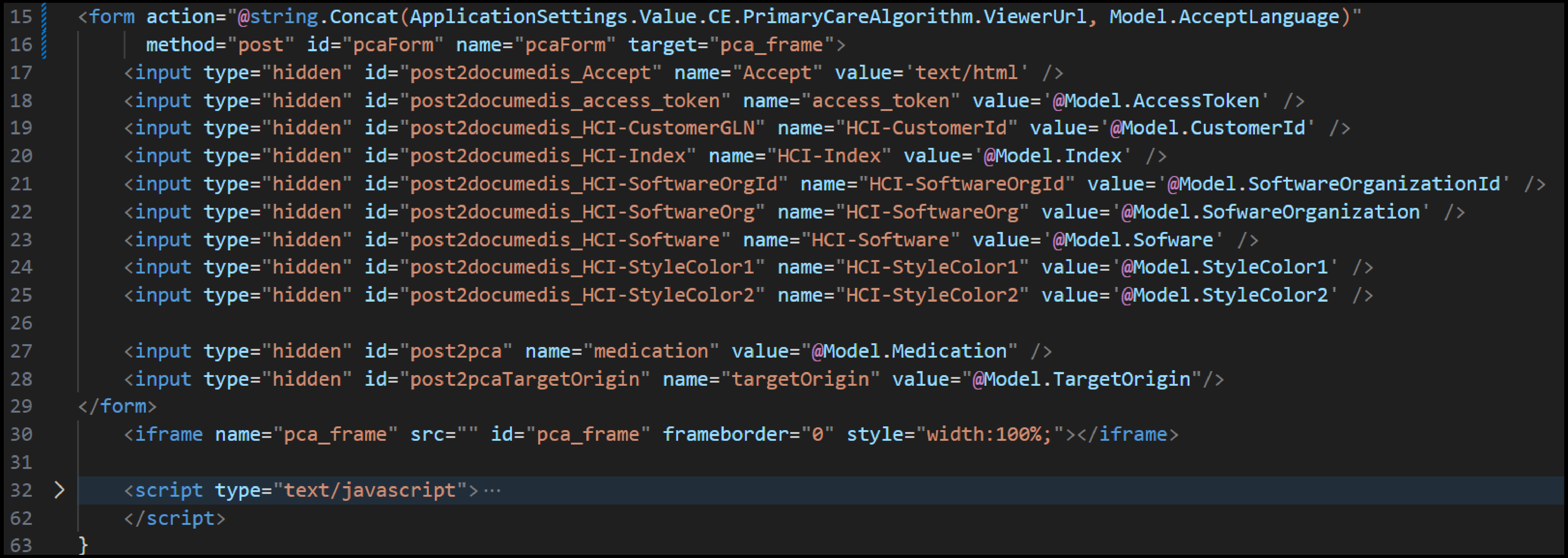
```
private HttpClient GetHttpClient(IHciHttpHeaders headers, string acceptLanguage = "", string accept = "application/json")
{
    var client = new HttpClient();
    client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", headers.Token);

    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue(accept));
    client.DefaultRequestHeaders.AcceptLanguage.Add(new StringWithQualityHeaderValue(acceptLanguage));

    client.DefaultRequestHeaders.Add("HCI-CustomerId", Base64EncodeHeaderValue(headers.CustomerId));
    client.DefaultRequestHeaders.Add("HCI-Index", Base64EncodeHeaderValue(headers.Index));
    client.DefaultRequestHeaders.Add("HCI-SoftwareOrgId", Base64EncodeHeaderValue(headers.SoftwareOrganizationId));
    client.DefaultRequestHeaders.Add("HCI-SoftwareOrg", Base64EncodeHeaderValue(headers.SoftwareOrganization));
    client.DefaultRequestHeaders.Add("HCI-Software", Base64EncodeHeaderValue(headers.Software));
    client.DefaultRequestHeaders.Add("HCI-UserId", Base64EncodeHeaderValue(headers.UserId));
    client.DefaultRequestHeaders.Add("HCI-UserName", Base64EncodeHeaderValue(headers.UserName));
    client.DefaultRequestHeaders.Add("HCI-UserAccessType", Base64EncodeHeaderValue(headers.UserAccessType));
    client.DefaultRequestHeaders.Add("HCI-UserAuthenticationLevel", Base64EncodeHeaderValue(headers.UserAuthenticationLevel));
    client.DefaultRequestHeaders.Add("HCI-UserHealthProfession", Base64EncodeHeaderValue(headers.UserHealthProfession));
    client.DefaultRequestHeaders.Add("HCI-SubCatalogId", Base64EncodeHeaderValue(headers.SubCatalogId));
    client.DefaultRequestHeaders.Add("HCI-WholesalerGln", Base64EncodeHeaderValue(headers.WholesalerGln));
    return client;
}

private string Base64EncodeHeaderValue(string value)
{
    var valueAsBytes = Encoding.UTF8.GetBytes(value);
    return $"[{Convert.ToBase64String(valueAsBytes)}]";
}
```

Providing the headers
Example 2. Rust



```rust
use base64::prelude::*;
use reqwest::{
    blocking::{Client, Request, Response},
    header,
};
use std::time::Duration;

fn main() -> Result<(), Box<dyn std::error::Error>> {
    let mut headers = header::HeaderMap::new();

    headers.insert("HCI-SoftwareOrgId",header::HeaderValue::from_str(&std::env::var("HCI_SOFTWAREORGID")?)?);

    let software_name = BASE64_STANDARD.encode("Not_an_äscïi_name");
    headers.insert("HCI-Software",header::HeaderValue::from_str(&format!("[{}]", software_name))?);

    let client: Client = reqwest::blocking::Client::builder().default_headers(headers)
        .user_agent("headers_in_rust").build()?;

    let req: Request = client.get("https://int.documedis.hcisolutions.ch/2020-01/api/products/1383139")
        .bearer_auth(std::env::var("HCI_AUTHTOKEN")?).timeout(Duration::from_secs(60 * 2))
        .build()?;

    let res: Response = reqwest::blocking::Client::execute(&client, req)?;
    println!("Success? {:?}", res.status().is_success());
    println!("Response: {:?}", res);
    println!("Response content: {:?}", res.text());

    Ok(())
}
```

Providing the headers
Example 3. HTML form



```
<form action="@string.Concat(ApplicationSettings.Value.CE.PrimaryCareAlgorithm.ViewerUrl, Model.AcceptLanguage)"
      method="post" id="pcaForm" name="pcaForm" target="pca_frame">
    <input type="hidden" id="post2documedis_Accept" name="Accept" value='text/html' />
    <input type="hidden" id="post2documedis_access_token" name="access_token" value='@Model.AccessToken' />
    <input type="hidden" id="post2documedis_HCI-CustomerGLN" name="HCI-CustomerId" value='@Model.CustomerId' />
    <input type="hidden" id="post2documedis_HCI-Index" name="HCI-Index" value='@Model.Index' />
    <input type="hidden" id="post2documedis_HCI-SoftwareOrgId" name="HCI-SoftwareOrgId" value='@Model.SoftwareOrganizationId' />
    <input type="hidden" id="post2documedis_HCI-SoftwareOrg" name="HCI-SoftwareOrg" value='@Model.SofwareOrganization' />
    <input type="hidden" id="post2documedis_HCI-Software" name="HCI-Software" value='@Model.Sofware' />
    <input type="hidden" id="post2documedis_HCI-StyleColor1" name="HCI-StyleColor1" value='@Model.StyleColor1' />
    <input type="hidden" id="post2documedis_HCI-StyleColor2" name="HCI-StyleColor2" value='@Model.StyleColor2' />

    <input type="hidden" id="post2pca" name="medication" value="@Model.Medication" />
    <input type="hidden" id="post2pcaTargetOrigin" name="targetOrigin" value="@Model.TargetOrigin"/>
</form>
    <iframe name="pca_frame" src="" id="pca_frame" frameborder="0" style="width:100%;"></iframe>

    <script type="text/javascript">…
    </script>
}
```

Adaptation phases

- Both environments validate headers mentioned
- Access depends on the current phase and environment
- An information message is present in response header :
  `Header-Validation-Result`
- An error message is displayed along with a 401 HTTP Status code
- Information & Error messages are described in specification

	Phase 1	Phase 2
Header validation performed	Yes	Yes
Can I get in	Yes (Yes/No on Integration)	No on both environments
Validation result	Information Message in `Header-Validation-Result`	Error Message

Adaptation phases

- We log all requests that are being sent to Documedis
- We log your
  - Token,
  - SoftwareOrgId,
  - SoftwareOrg,
  - Software
  - Index,
  - CustomerId
- So we can contact customers that have not adapted before the deadline

Wrap up

- Our **Prod**uction environment is going to require correct HCI HTTP Headers
- You can already test the HTTP Headers on **Int**egration.
- In the future if you can not access **Int**, you can not access **Prod**!
- You have time to adapt.

Postman & DID
Demo Time

Questions, ideas or suggestions?

Thanks for your attention!