Bringing Zero Trust with OpenEdge

Moving from Citrix to Azure.
The GUI for .NET frontend application IBIS at the COA organization had to comply to zero trust architecture.

flusso IT

Introduction



Richard Kelters

- Flusso employee since 2008
- 25+ years experience as a Progress OpenEdge developer
- PUG Netherlands board member from 2002 till 2018

What does Flusso do?

From complex IT to clarity and progress

OpenEdge

- Progress OpenEdge & ABL
- Application Maintenance & modernization
- Reliable, database-driven software

Other Software

- Java & Spring Boot
- Angular, TypeScript & Frontend
- OutSystems (LowCode)
- Mobile applications

Engineering, DevOps & Cloud

- CI/CD & delivery automation
- DevSecOps, security & SBOM
- Containerization & orchestration
- Identity & access management

Integration, Modernisation & AI

- System integration
- Hybrid and cloud-native architectures
- LLMs, Rag & MCP

flusso IT

Don’t miss

Thursday 13:45 in Brussels 1
Thomas Brom – Security officer and frontend developer

Advanced AI Security

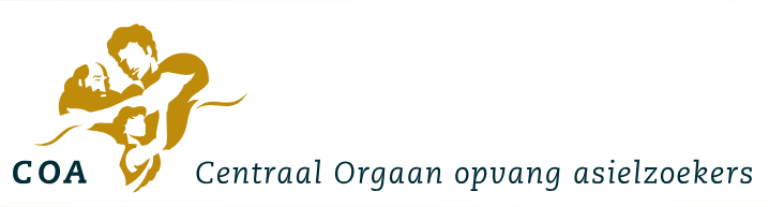
This session combines practical insight with concrete measures, helping you to adopt AI in a controlled and secure way within your development process.

flusso IT

Agenda

- COA organization
- IBIS application
- Architecture
- APSV challenge
- Azure Entra
- PASOE
- Proxy
- Demo
- Questions

flusso IT



COA

Central Organization for Assylum seekers

- Governmental organization
- Justice department’s responsibility
- Headquarters in The Hague
- 300 locations
- 75,000 residents
- 10,000 employees

flusso IT



IBIS

Integraal **B**ewoner **I**nformatie **S**ysteem
or
Integrated Resident Information System

- 2000 combining different applications into one ADM2 clientserver app
- 2008 start modernizing to GUI for .NET in version 10.2b using AppServer
- 2018 fully migrated
- 2021 migrated to pasoe version 11.7
- 2024 OpenEdge 12.2 WebClient
- 2025 OpenEdge 12.8

flusso IT



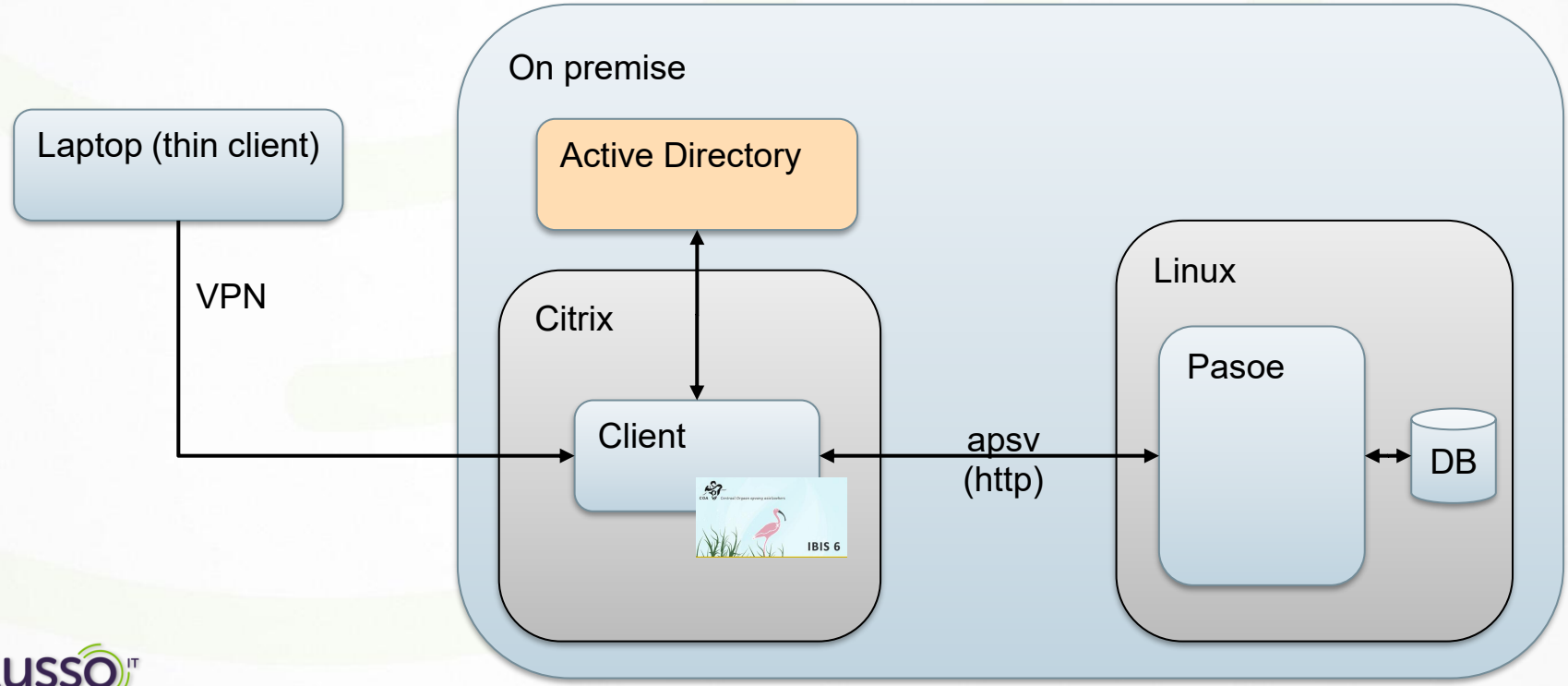
IBIS

Integraal **B**ewoner **I**nformatie **S**ysteem

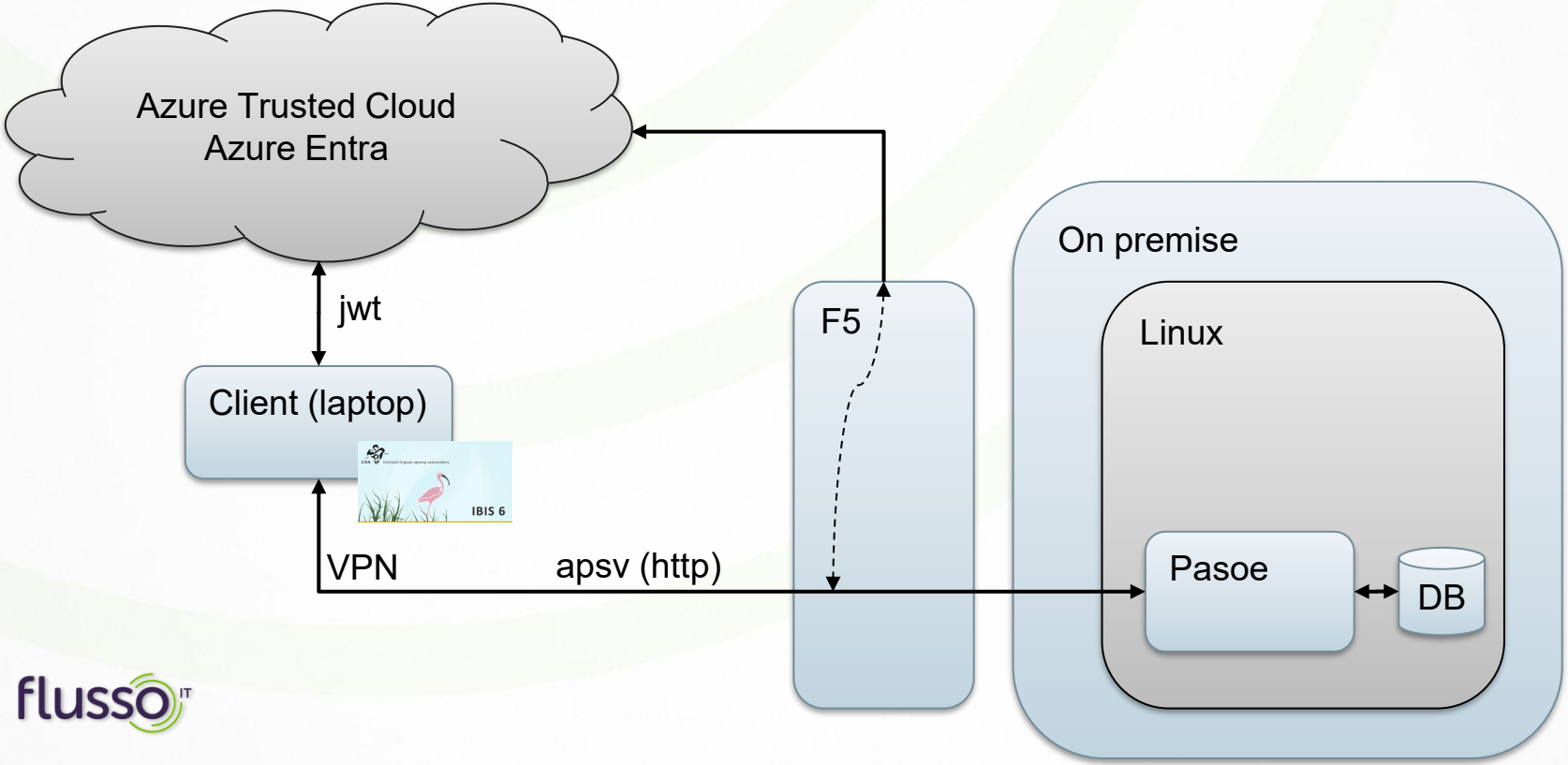
- 8000 active users
- 10 developers
- 4 databases
- IBIS database 150 GB
- 2 pasoe instances
- 3400 classes
- 1780 procedural programs
  - 950 database triggers
  - 90 Smart Data Objects
  - 200 service interfaces

flusso IT

Previous architecture



Zero Trust architecture



APSV protocol

Proprietary protocol from Progress for OpenEdge

- Classic Appserver (V8 – V11)
- Progress Appserver for OpenEdge (V11-?)
  - tomcat
  - uses http
  - network packages in the body

```
create server appserver.
appserver:connect("-AppService IBIS -S 20211 -H codex ").
appserver:connect("-URL https://codex:20211").
```

flusso IT

APSV challenge

COA requested us to add an authorization header in all HTTP requests to comply to the zero trust architecture.

But

We don’t have access to http requests or responses as an OpenEdge developer ....

COA options for APSV

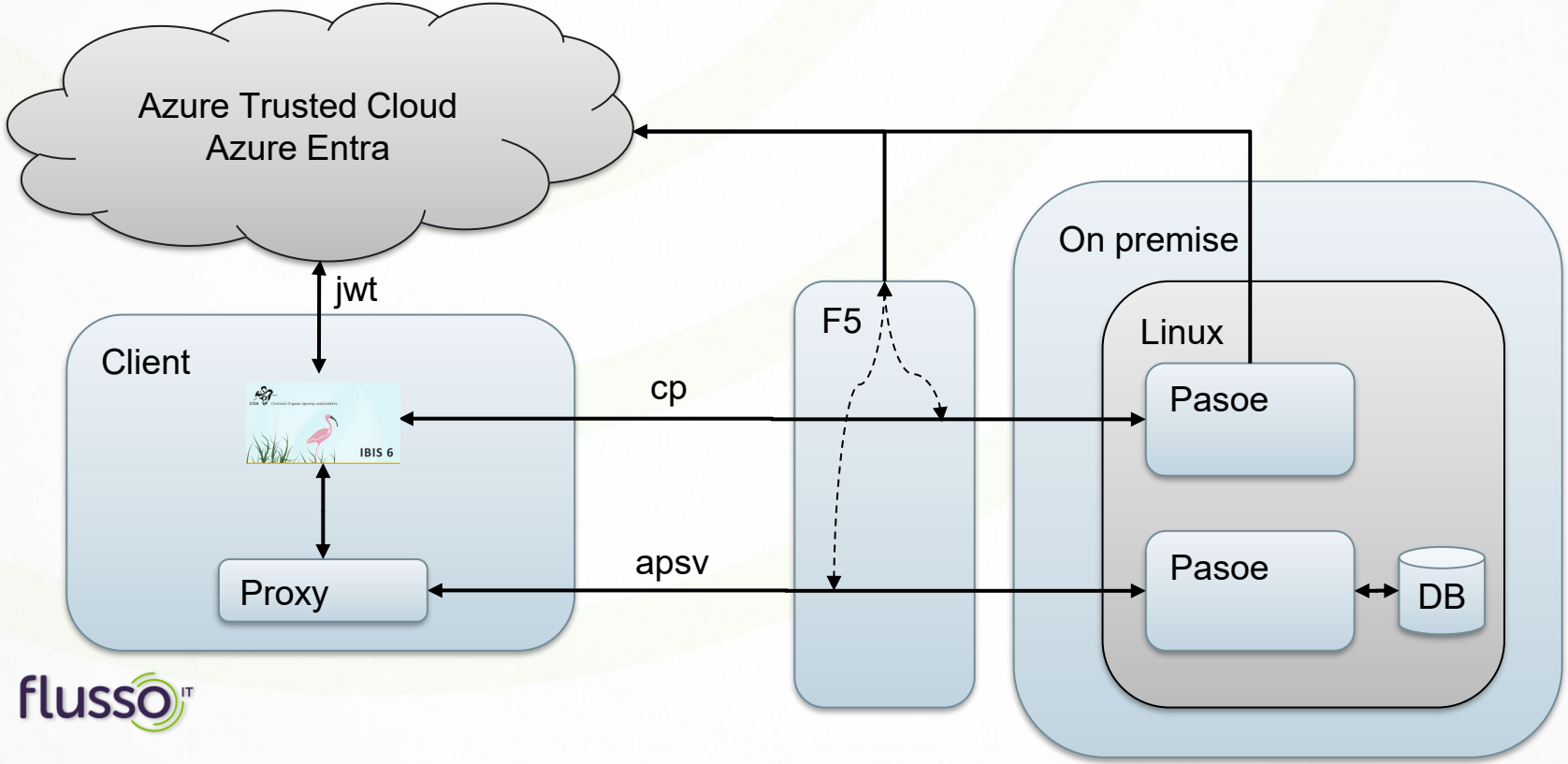
- Make an exception and allow unauthorized http requests for IBIS
- Wait for Progress Software to support http authorization for APSV
- Rewrite all APSV calls to http requests
- Make solution to add Authorization header to HTTP

Decision

- Build a client proxy to inject Authorization header
- With language Go
  - Strong in networking
  - Stand alone executable

flusso IT

Zero Trust architecture



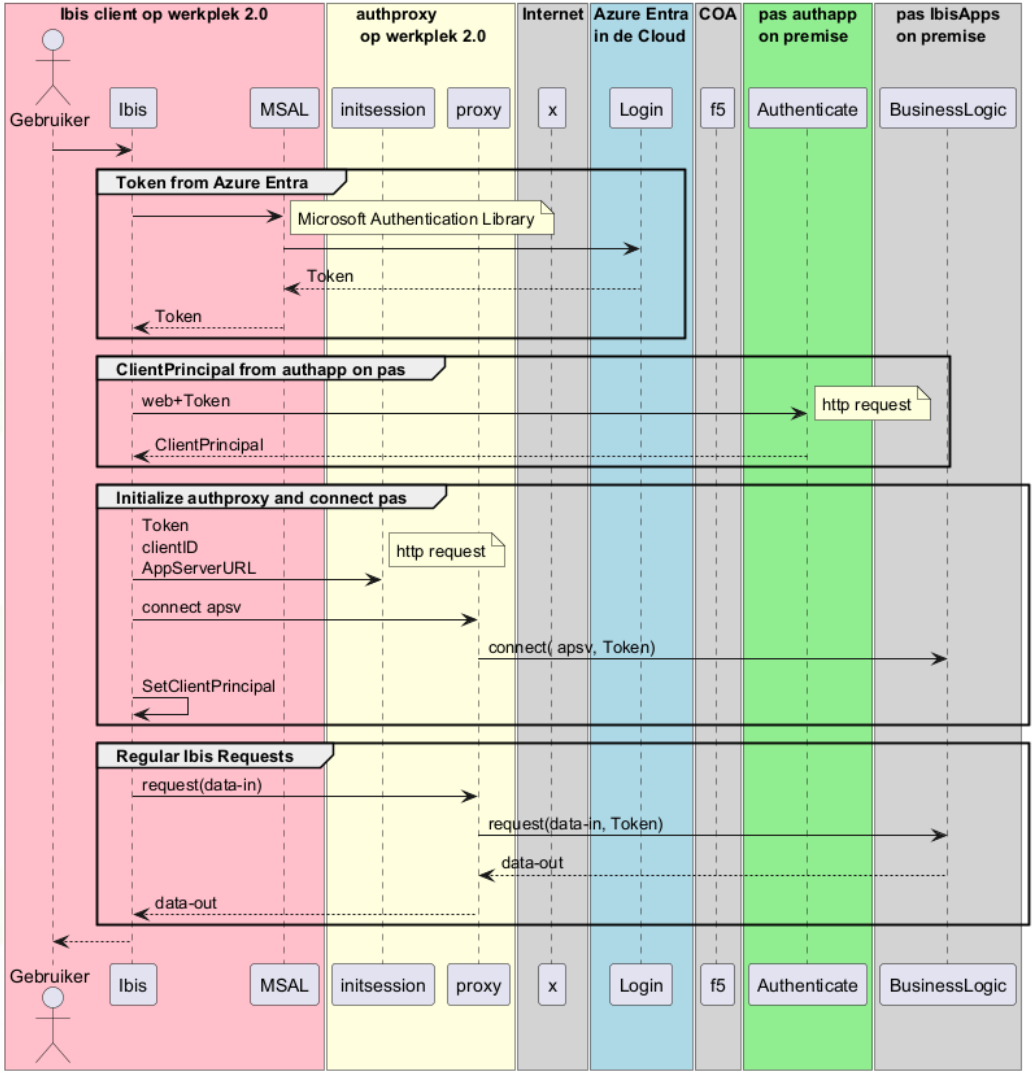
New login flow

- Azure Entra SSO
- Receive JWT

- Fetch ClientPrincipal

- Setup proxy
- Connect proxy
- SetClientPrincipal

- Ready



Challenges

- Acquire a JWT from Azure Entra using MSAL
- Setup pasoe with jwt validation
  return IBIS specific client-principal
- Build a proxy
  to add Authorization

flusso IT

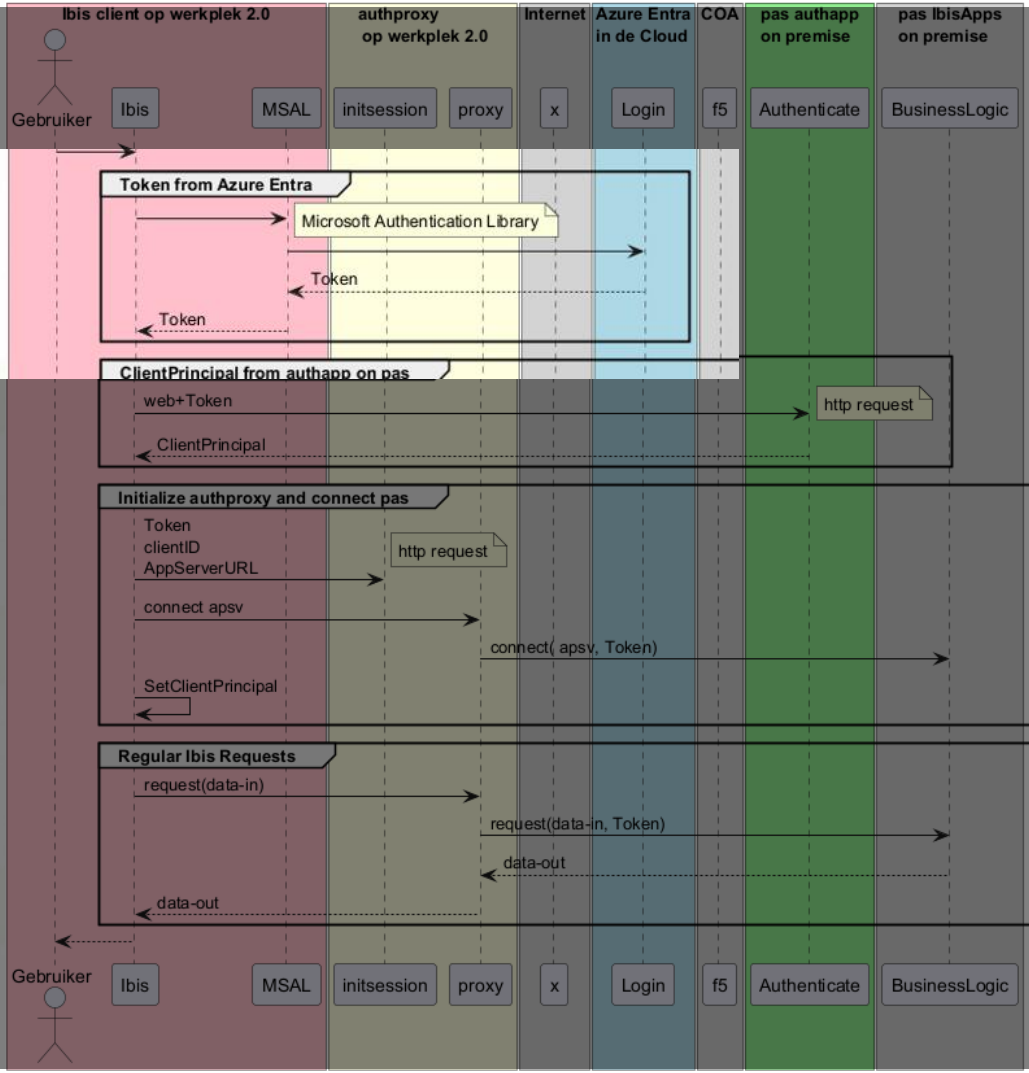
New login flow

- Azure Entra SSO
- Receive JWT

- Fetch ClientPrincipal

- Setup proxy
- Connect proxy
- SetClientPrincipal

- Ready



Azure Entra AppRegistration

- Single Sign On
- Desktop application
- Provide callback URL
  `https://localhost:12345`
- Supply userID from AD for backward compatibility
  `onpremisessamaccountname`

You get:
- TenantId
- ClientId

flusso IT

Aquire JWT from Entra using MSAL

```
ServicePointManager:SecurityProtocol = SecurityProtocolType:Tls12.

publicApp = PublicClientApplicationBuilder
           :Create(clientId)
           :WithAuthority(AzureCloudInstance:AzurePublic, tenantId, true)
           :WithRedirectUri(callBackUrl)
           :Build()
           .

authResult = publicApp:AcquireTokenInteractive("<clientId>/.default")
            :WithPrompt(Prompt:NoPrompt)
            :ExecuteAsync()
            :GetAwaiter()
            :GetResult()
            .

message "JWT from Azure Entra " authResult:AccessToken
view-as alert-box.
```

flusso IT

Refresh JWT with Entra

```
ServicePointManager:SecurityProtocol = SecurityProtocolType:Tls12.

publicApp = PublicClientApplicationBuilder
            :Create(clientId)
            :WithAuthority(AzureCloudInstance:AzurePublic, tenantId, true)
            :WithRedirectUri(callBackUrl)
            :Build()
            .

authResult = publicApp:AcquireTokenSilent("<clientId>/.default", authResult?:account)
             :WithForceRefresh(true)
             :ExecuteAsync()
             :GetAwaiter()
             :GetResult()
             .

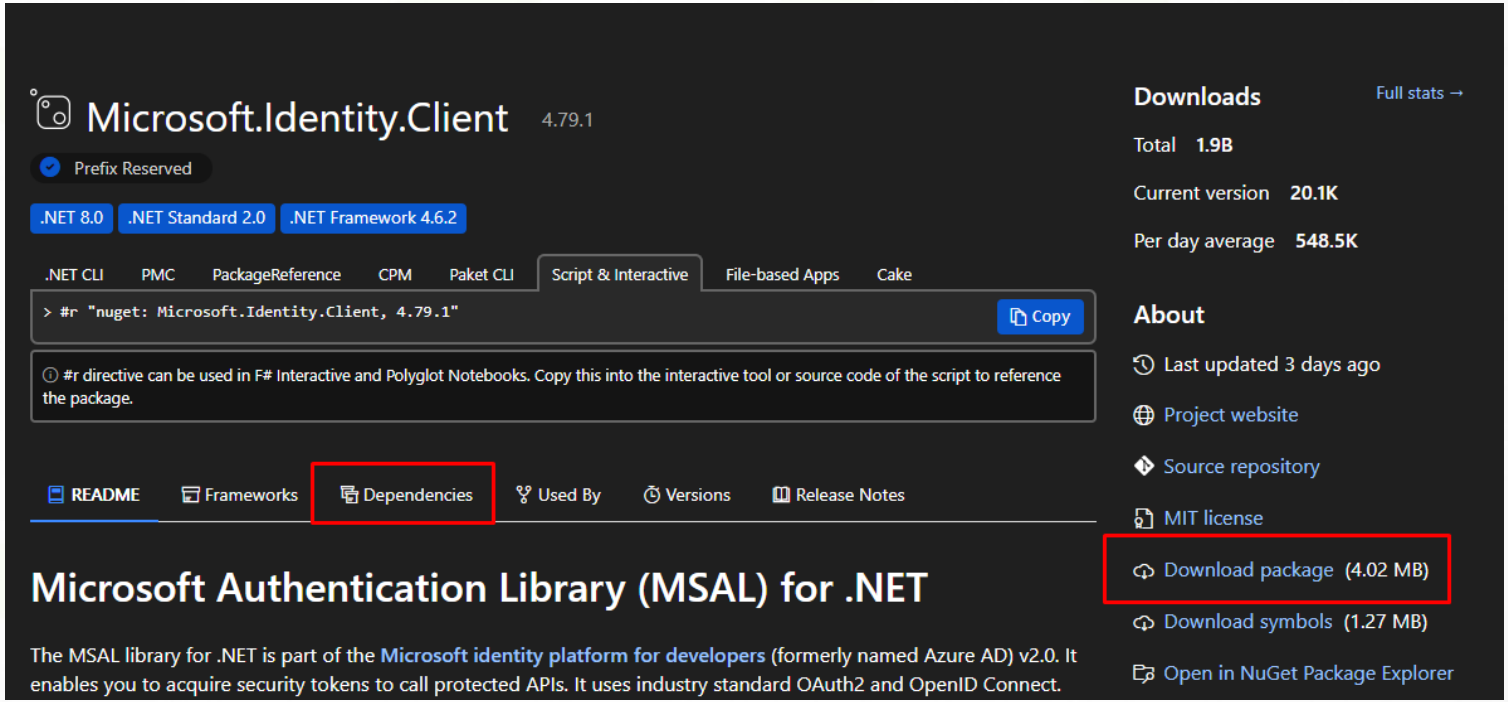
message "JWT from Azure Entra " authResult:AccessToken
view-as alert-box.
```

flusso IT

MSAL

Microsoft Authentication Library

https://www.nuget.org/packages/Microsoft.Identity.Client/



MSAL controls

Microsoft Identity Client
https://www.nuget.org/packages/Microsoft.Identity.Client/

Extract `lib\net8.0\Microsoft.Identity.Client.dll` to your assemblies folder

Microsoft IdentityModel Abstractions
https://www.nuget.org/packages/Microsoft.IdentityModel.Abstractions/

Extract `lib\net8.0\Microsoft.IdentityModel.Abstractions.dll` to your assemblies folder

MSAL controls

- Update assemblies.xml accordingly

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<references>
    <assembly name="System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"/>
    <assembly name="Microsoft.Identity.Client, Version=4.74.1.0, Culture=neutral, PublicKeyToken=0a613f4dd989e8ae"/>
    <assembly name="Microsoft.IdentityModel.Abstractions, Version=8.3.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364
</references>
```

flusso IT

MSAL dotnetcore

%DLC%\bin\Progress.clrbridge.netcore.runtimeconfig.json

https://community.progress.com/s/article/Developer-Studio-fails-to-start-due-to-a-misconfigured-runtimeconfig-json



```
{
  "runtimeOptions": {
    "tfm": "net8.0",
    "frameworks": [
      {
        "name": "Microsoft.NETCore.App",
        "version": "8.0.0"
      },
      {
        "name": "Microsoft.WindowsDesktop.App",
        "version": "8.0.0"
      }
    ],
    "configProperties": {
      "System.Reflection.Metadata.MetadataUpdater.IsSupported": false
    }
  }
}
```

flusso IT

MSAL

Update .config in `%DLC%\bin\`

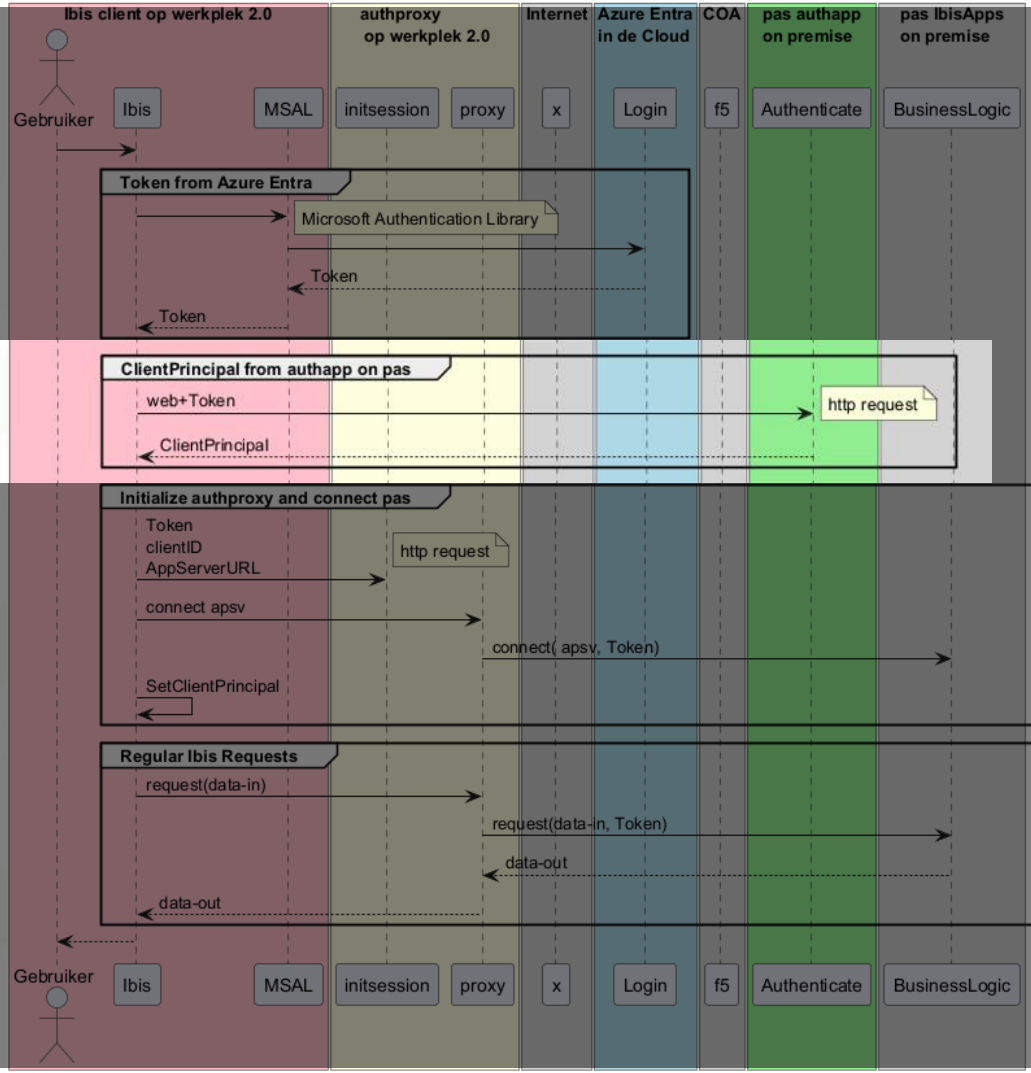
- `prowin.exe.config`
- `prowc.exe.config`
- `prowin32.exe.config (?)`

```
<configuration>
<startup useLegacyV2RuntimeActivationPolicy="true">
  <supportedRuntime version="v4.0"/>
</startup>
<runtime>
  <loadFromRemoteSources enabled="false" />
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1" >
    <!-- DO NOT REMOVE THIS probing ELEMENT -->
    <probing privatePath="Infragistics\winforms;telerik\winforms;dotn
    <!-- START REDIRECTIONS -->
    <!-- for MSAL -->
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.IdentityModel.Abstractions" pu
      <publisherPolicy apply="no"/>
      <bindingRedirect oldVersion="6.35.0.0" newVersion="8.3.1.0"/>
    </dependentAssembly>
    <!-- END REDIRECTIONS -->
  </assemblyBinding>
</runtime>
</configuration>
```

flusso IT

New login flow

- Azure Entra SSO
- Receive JWT
- Fetch ClientPrincipal
- Setup proxy
- Connect proxy
- SetClientPrincipal
- Ready



PASOE ablapp

- add a new ablapp on existing pasoe
- only configure web protocol
- have a single webhandler
- only accept GET
- configure security for jwt

PASOE security

update oeablSecurity.properties

`<pasoe>/webapps/<ablApp>/WEB-INF/oeablSecurity.properties`

```
client.login.model=oauth2

jwtToken.keystore.jwkurl=https://login.microsoftonline.com/<tenantId>/discovery/v2.0/keys
jwtToken.usernameField=onpremisessamaccountname
jwtToken.scopeNameField=roles

oauth2.resSvc.tokenServices=jwt
oauth2.resSvc.audience=<clientId>
```

flusso IT

PASOE debugging

configure logging logging.xml

`<pasoe>/webapps/<ablApp>/WEB-INF/logging.xml`

```
<!-- You may include logback <logger>....</logger> directives here in the event
       that logging can only be turned on for specific Tomcat web application contexts.
       For debugging security settings uncomment the following lines

  <logger name="org.springframework.security.oauth2" level="TRACE" />
  <logger name="org.springframework.security.jwt" level="TRACE" />
  <logger name="org.springframework" level="TRACE"/>
  <logger name="com.progress.appserv.services.security" level="TRACE"/>
-->
```

flusso IT

Client principal service

- OpenEdge converts jwt to client-pricipal
  jwt claims -> client principal properties
- Create a IBIS specific client-principal
- return client-principal in base64

flusso IT

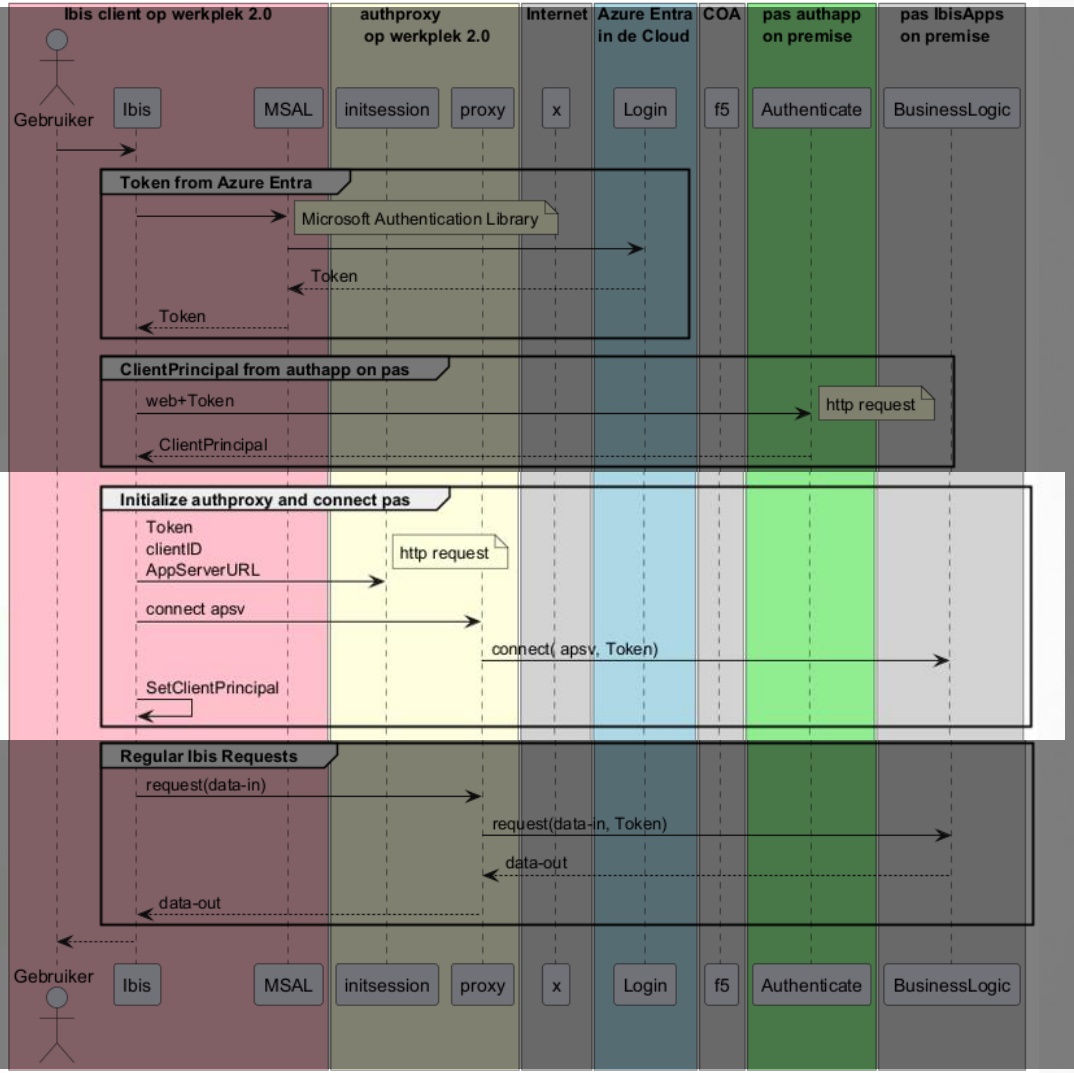
New login flow

- Azure Entra SSO
- Receive JWT

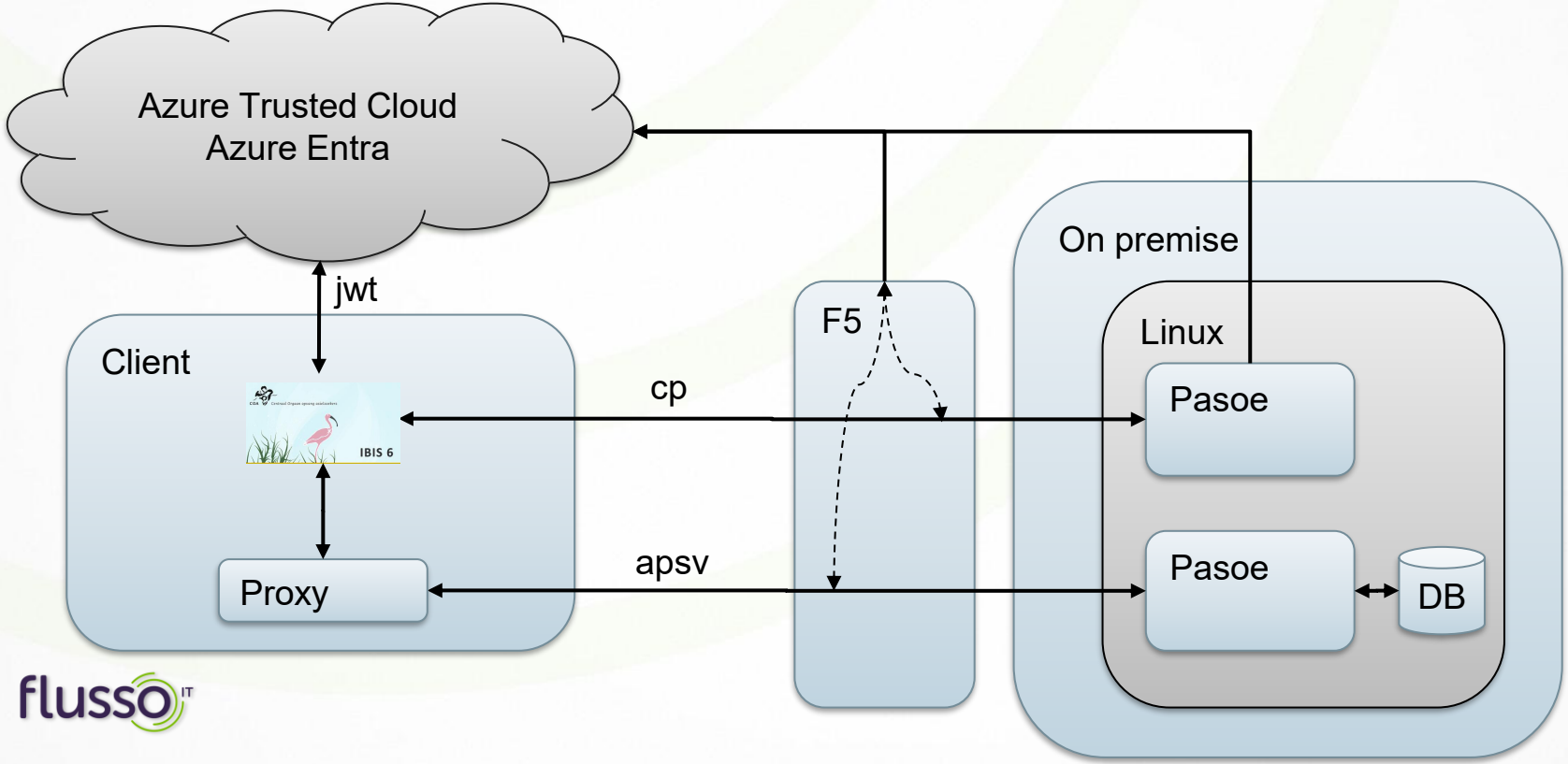
- Fetch ClientPrincipal

- Setup proxy
- Connect proxy
- SetClientPrincipal

- Ready



Zero Trust architecture



Authorization Proxy

Create a proxy between OpenEdge client and pasoe

- Runs on the client
- Started by the IBIS client
- Receives a JWT and URL (initsession)
- Connected by IBIS client as if it is an appserver
- Inserts Authorization header to each outgoing http request

Authorization Proxy Start

```
authproxy.exe --nohostverify --proxyport 18820 --loglevel trace
```

Command line arguments

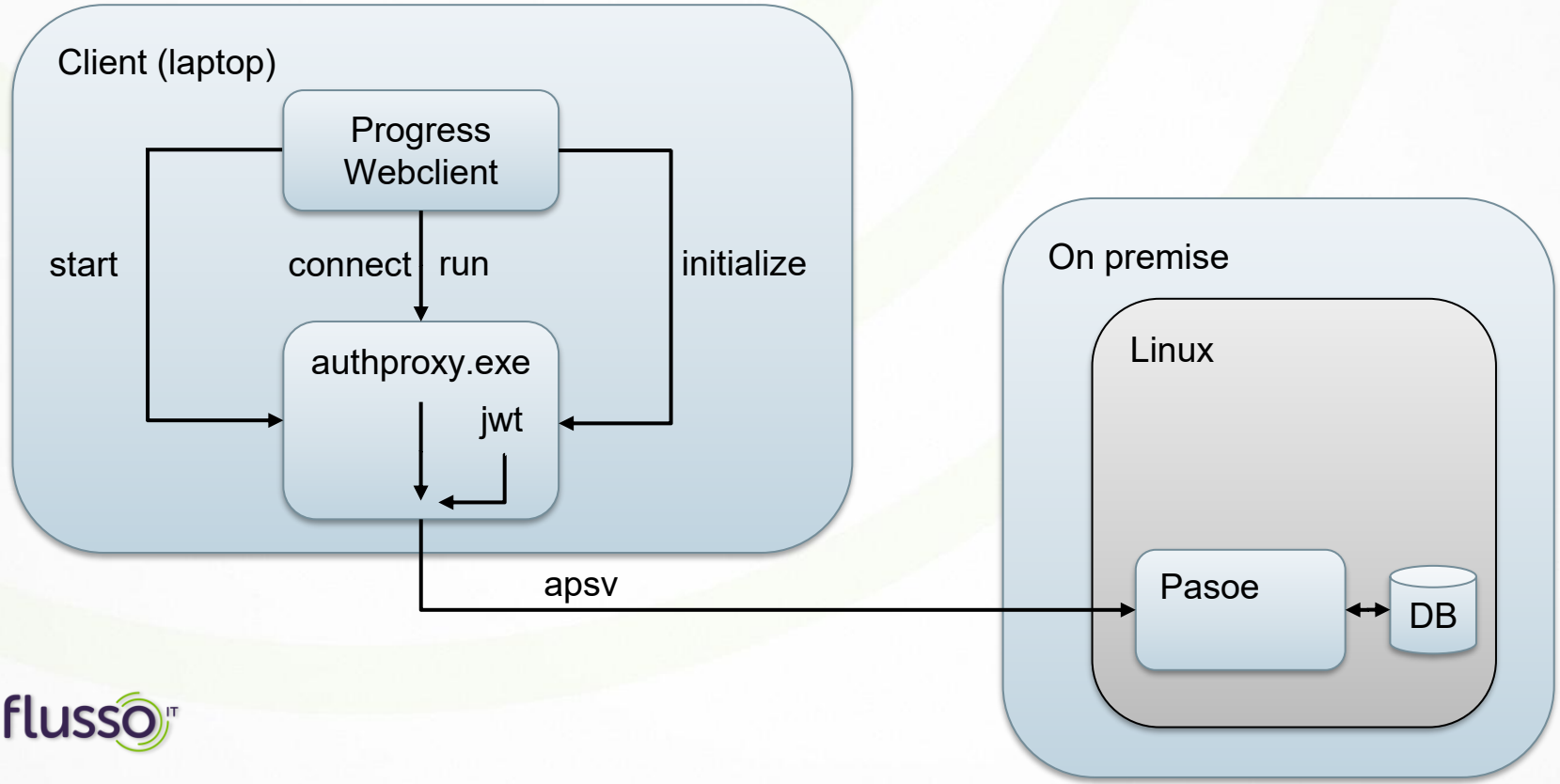
--proxyport (default=18810)
--nohostverify
--loglevel (default= info)
--authscheme (default= Bearer)
--version

flusso IT

Authorization Proxy Endpoints

POST	/quit	shutsdown the proxy
GET	/alive	returns 200 OK
GET	/stats/{sessionId}	statistics
POST	/stats/{sessionId}	reset statistics
POST	/loglevel	set (body: {“loglevel”:”debug”}
POST	/initsession	initialize for appserver connection
POST	/apsv	used by OpenEdge

Authorization Proxy



flusso IT

Authorization Proxy Initialize

OpenEdge pasoe connection string

```
-URL https://backend.server/pasoe/apsv -sessionModel Session-free
```

OpenEdge http request

```
POST http://localhost:18810/initsession
Content-Type: application/json

{ "sessionid": "12345",
  "token": "jwt_recieved_from_entra",
  "appserver": "https://backend.server/pasoe/apsv" }
```

OpenEdge proxy connection string

```
-URL http://localhost:18810/apsv?AppService=12345 -sessionModel Session-free
```

flusso IT

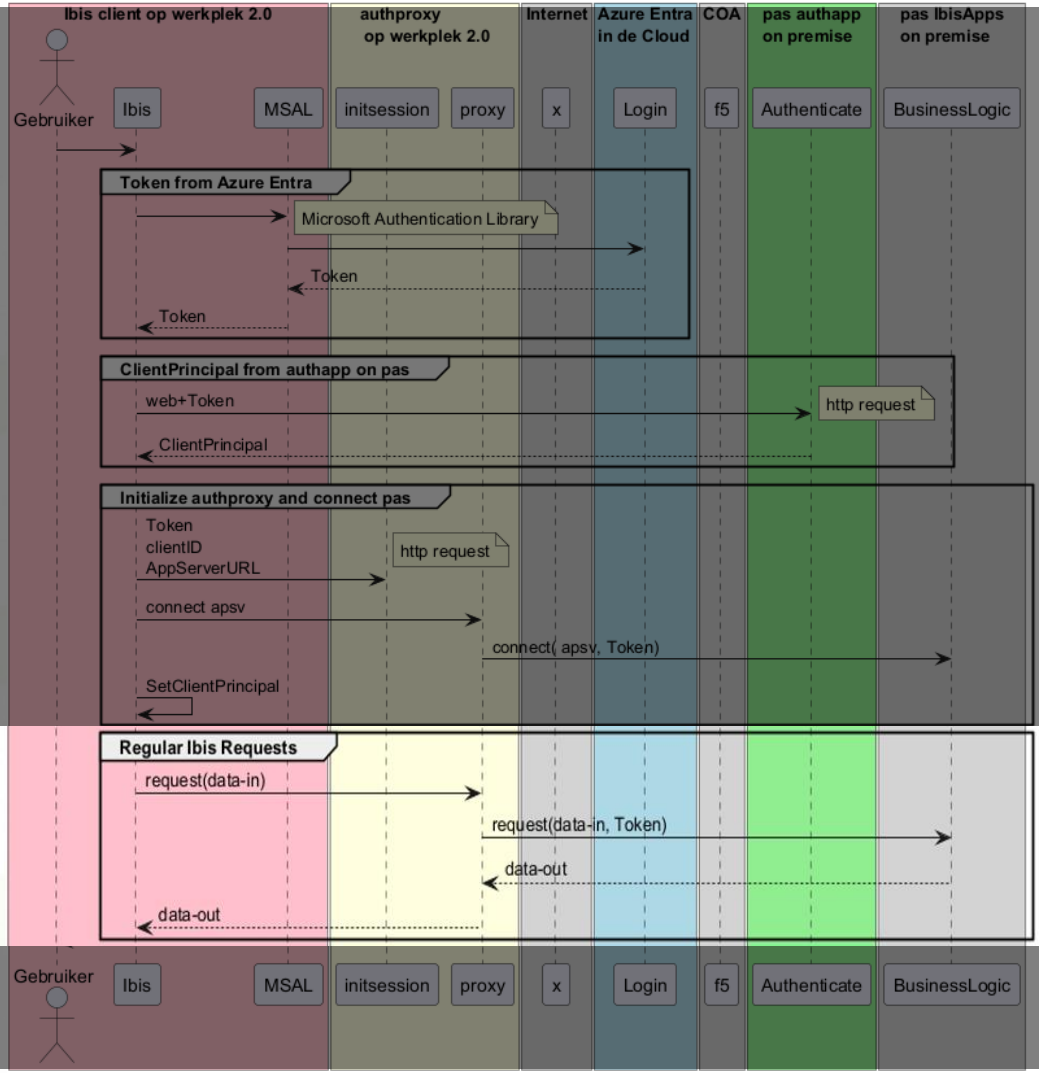
New login flow

- Azure Entra SSO
- Receive JWT

- Fetch ClientPrincipal

- Setup proxy
- Connect proxy
- SetClientPrincipal

- Ready



Demo

Conclusion

To comply to zero trust is possible

But it is not trivial

- DIY
- reverse engineer APSV

flusso IT

Questions

- ?

Don’t miss

Thursday 13:45 in Brussels 1
Thomas Brom – Security officer and frontend developer

Advanced AI Security

This session combines practical insight with concrete measures, helping you to adopt AI in a controlled and secure way within your development process.

flusso IT