

DYNAMICS CON

POWERED BY
DUG





Handling files with Azure Blob Storage the right way

by Simon Fischer

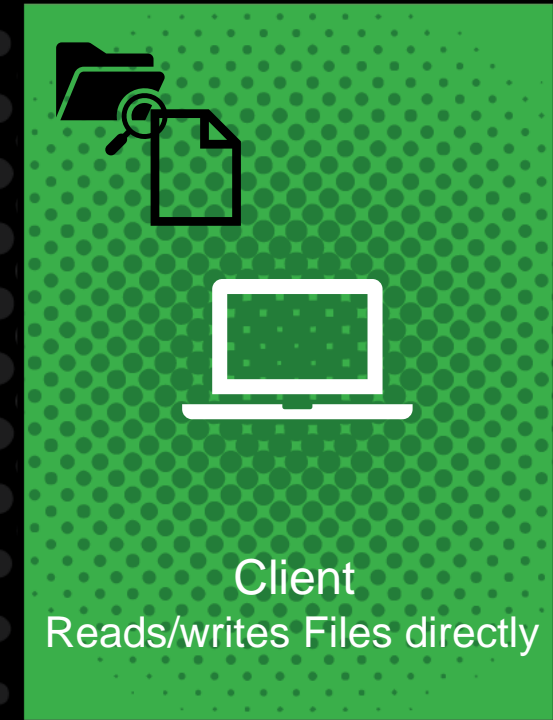
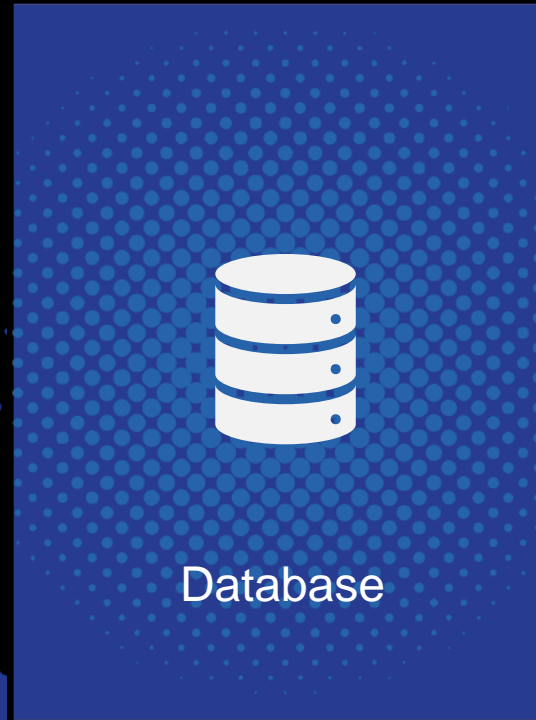
What to expect?

- **Theory (approx. 10 minutes)**
 - General overview of File handling and its evolution
 - Limitations and challenges in SaaS-scenarios
 - Short introduction into Azure Storage Accounts
- **Practice (Sample Scenario, Demos; approx. 20 minutes)**
 - Possible solution for File handling in SaaS-scenarios
 - Sample Scenario, Demos and introduction into new app



File Handling – an evolution

File Handling – the early days



File Handling – the early days

- In 2-Tier architecture file access was rather simple
- Straight-forward commands via client
- File needed to be accessible from the user client
- In this simple form usable until NAV 2009

```
ReadFile()  
FileVar.WRITEMODE(FALSE);  
FileVar.OPEN('C:\My\Local\file.txt');  
FileVar.READ(Content);  
// ...  
// Do something with "Content"  
// ...  
FileVar.CLOSE;
```

(simplified examples)

```
SaveFile(Content : Text[100])  
FullFilename := 'C:\My\Local\file.txt';  
FileVar.WRITEMODE(TRUE);  
IF NOT FILE.EXISTS(FullFilename) THEN  
    FileVar.CREATE(FullFilename)  
ELSE  
    FileVar.OPEN(FullFilename);  
FileVar.WRITE(Content);  
FileVar.CLOSE;
```

File Handling – the later days



File Handling – the later days

- In 3-Tier architecture things became a bit more complex
- Dependent on which tier accessed the file, it needed to be available either on the server or the user client
- File might need to be uploaded / downloaded before / after processing
- From NAV 2009 onwards

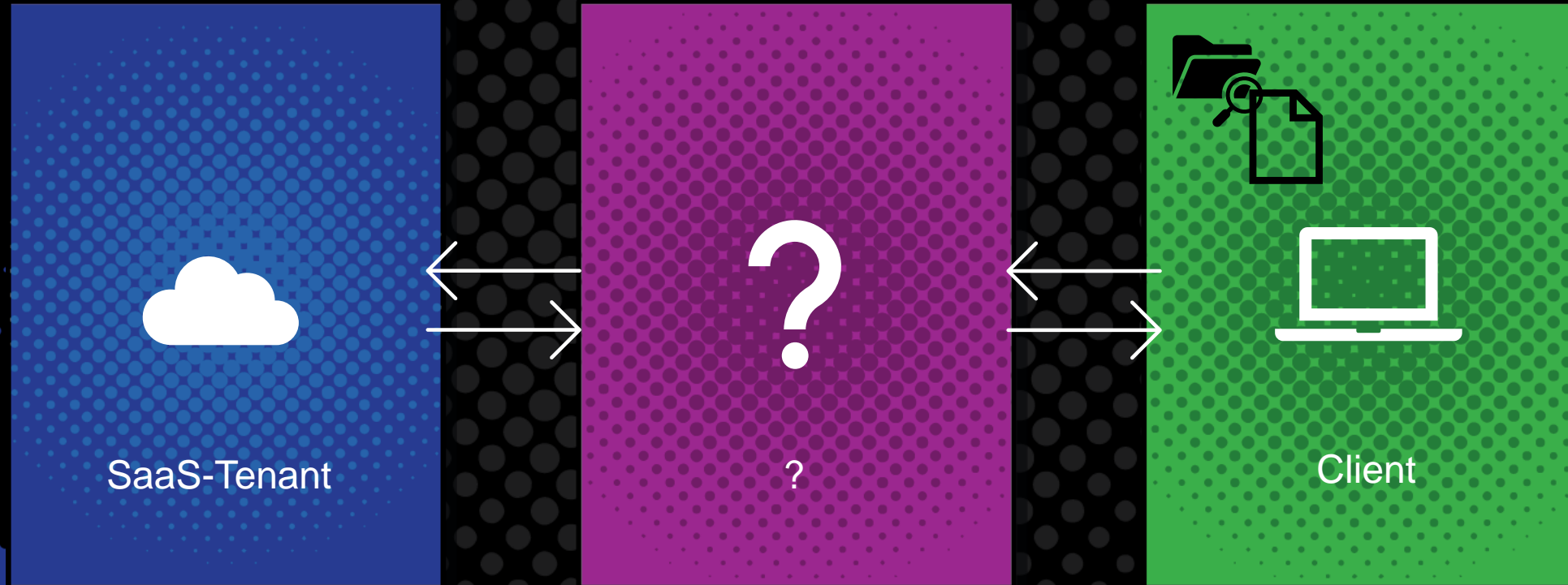
```
ReadLocalFile(FullFilename : Text)
IF NOT FileMgt.ServerFileExists(FullFilename) THEN
    FullFilename := FileMgt.UploadFileSilent(FullFilename);

FileVar.WRITEMODE(FALSE);
FileVar.OPEN(FullFilename);
FileVar.READ(Content);
// ...
// Do something with "Content"
// ...
FileVar.CLOSE;
```

(simplified examples)

```
SaveLocalFile(FullFilename : Text;Content : Text[100])
IF ISSERVICETIER THEN
    TempFilename := FileMgt.ServerTempFileName('.txt')
ELSE
    TempFilename := FullFilename;
FileVar.WRITEMODE(TRUE);
FileVar.CREATE(TempFilename);
FileVar.WRITE(Content);
FileVar.CLOSE;
IF ISSERVICETIER THEN
    FileMgt.DownloadToFile(TempFilename,FullFilename);
```


File Handling – in the cloud



File Handling – in the cloud

- **There is no (accessible) file system in the cloud**
- **Depending on function, users can upload or download files via browser**
- **Some questions arise for automatic/background processes:**
 - How should automatic/background processes handle files?
 - How to make files periodically available to external programs?
 - How to import files from external programs?
- **Different approaches are possible**

File Handling – in the cloud

- In some scenarios it might make sense to publish a custom API/web service to receive data/make data available
- Not all (legacy) systems can work with such APIs
- Some approaches that were possible on-prem are not possible on SaaS anymore (e.g., using 3rd party libraries for access to SFTP servers)



Introducing...



Azure Storage Accounts

Azure Storage Accounts

- **Azure Storage Accounts can be used for different kinds of data objects**
 - Blobs
 - Files
 - Queues
 - Tables
- **Storage accounts...**
 - ... are easily setup and accessible from anywhere
 - ... provide a comprehensive API
 - ... have no upfront cost and low prices for small capacities (e.g., 50 GB can be less than 2 USD/month)



... back to file handling

File Handling – in the cloud



File Handling – in the cloud

- One approach could be to only interact with Storage Accounts as the single (file) source from Business Central
- Other systems can provide or receive files via Storage Accounts
- Systems not capable of this can be integrated in other ways, e.g., with Azure Functions



Sample Scenario

Sample Scenario

- **Scenario: Integration of an SFTP-based file exchange**
- **You need to:**
 - Periodically upload data to a 3rd party SFTP server
 - Periodically download data from a 3rd party SFTP server
- **Problem:**
 - A direct upload or download is not possible, because Business Central does not provide functionality to connect to a SFTP server

Sample Scenario





The (not so) secret sauce

The (not so) secret sauce

- Demo uses the (open source) app “Azure Blob Storage API”
- The handling of the Storage Account is done with only a couple of lines of code
- Provides (almost) complete Blob service API functionality

```
local procedure ReadRemoteFile()
begin
    AuthKey := '89qw2EwvsNH...el0p590fzlsy7S8u0AakKMQ2f2/6+rw==';
    Request.InitializeRequest('dynamicscon2021demo001', // Account
                             'source-container',        // Container
                             'DemoData-001.txt');       // Blob
    Request.InitializeAuthorization(AuthType::SharedKey, AuthKey);

    API.DownloadBlobAsStream(Request, Content);
    // ...
    // Do something with "Content"
    // ...
end;
```

(simplified examples)

```
local procedure WriteRemoteFile(Content: InStream)
begin
    AuthKey := '89qw2EwvsNH...el0p590fzlsy7S8u0AakKMQ2f2/6+rw==';
    Request.InitializeRequest('dynamicscon2021demo001', // Account
                             'source-container',        // Container
                             'DemoData-001.txt');       // Blob
    Request.InitializeAuthorization(AuthType::SharedKey, AuthKey);

    API.UploadBlobIntoContainerStream(Request, 'DemoData-002.txt', Content);
end;
```

The (not so) secret sauce

- API offers much more possibilities than simple uploading/downloading
- Blob handling alone is covered with 46 API-operations
- 45 of these operation are (currently) covered with this App
- available on GitHub:

<https://github.com/cosmoconsult/D365BC-Blob-Storage-API>



The (not so) secret sauce



Automatic Export from Payment Journal to
Storage Account

The (not so) secret sauce



Automatic Bank Account Reconciliation importing
Bank Statements from Storage Account

The (not so) secret sauce



Lock a Storage Container for other users
while processing

Resources

- Documentation: <https://docs.microsoft.com/en-us/rest/api/storageservices/blob-service-rest-api>
- GitHub: <https://github.com/cosmoconsult/D365BC-Blob-Storage-API> (Contributions are welcome 😊)
- Demos, etc.: <https://dev.azure.com/DynamicsCon21-AzBlobStorage>

Resources

If you prefer links as QR code...

Documentation:



GitHub:



Demos, etc.:



Stay in touch

Want to connect and discuss about this and other topics?
Join me on:



Twitter



Linkedin



Xing



My Blog



Thank you for joining!