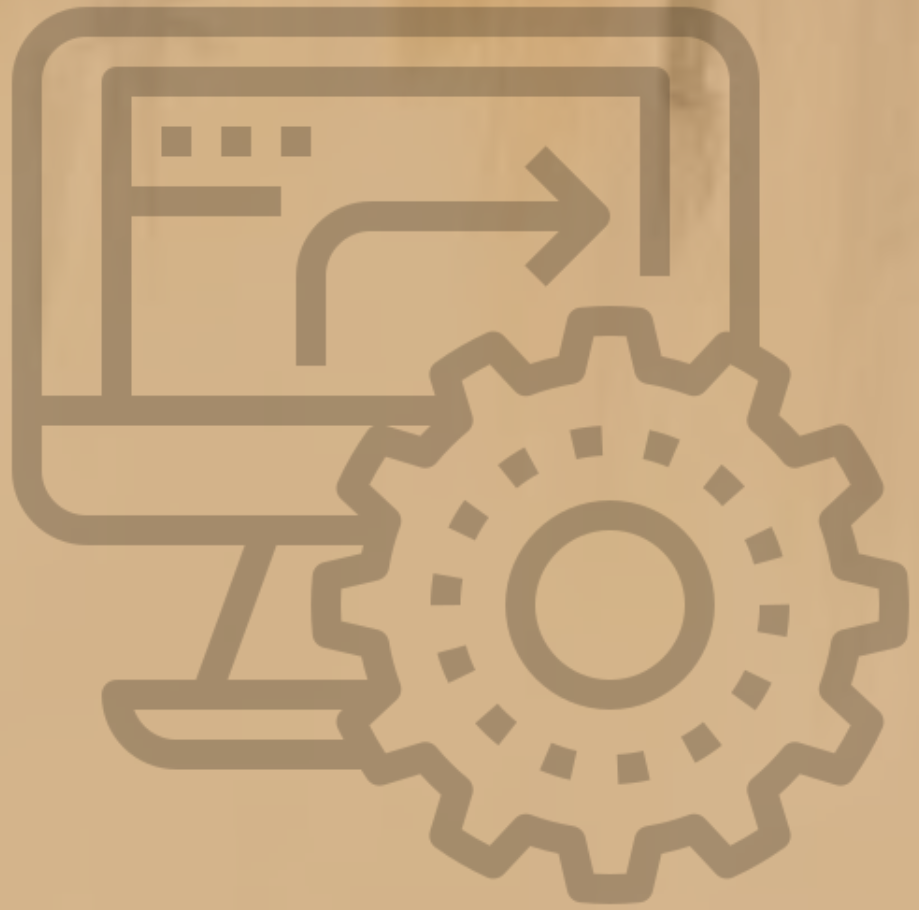
The Florida Lottery Department- Data Integration Research Team

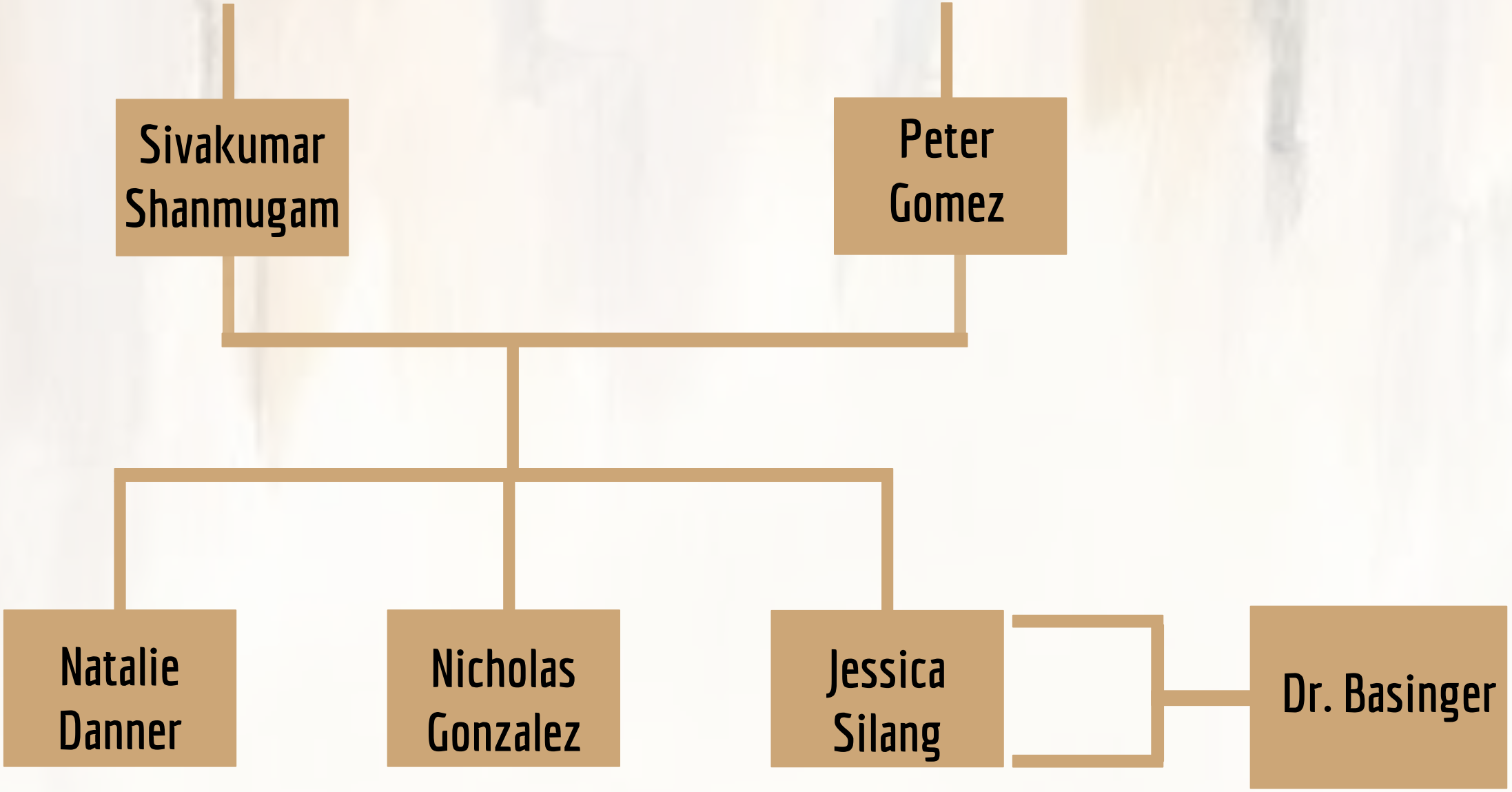
23 April 2021
EIN4335 Final Presentation

Nicholas Gonzalez
Jessica Silang
Natalie Danner



The Florida Lottery Department

Florida Department of Management Services

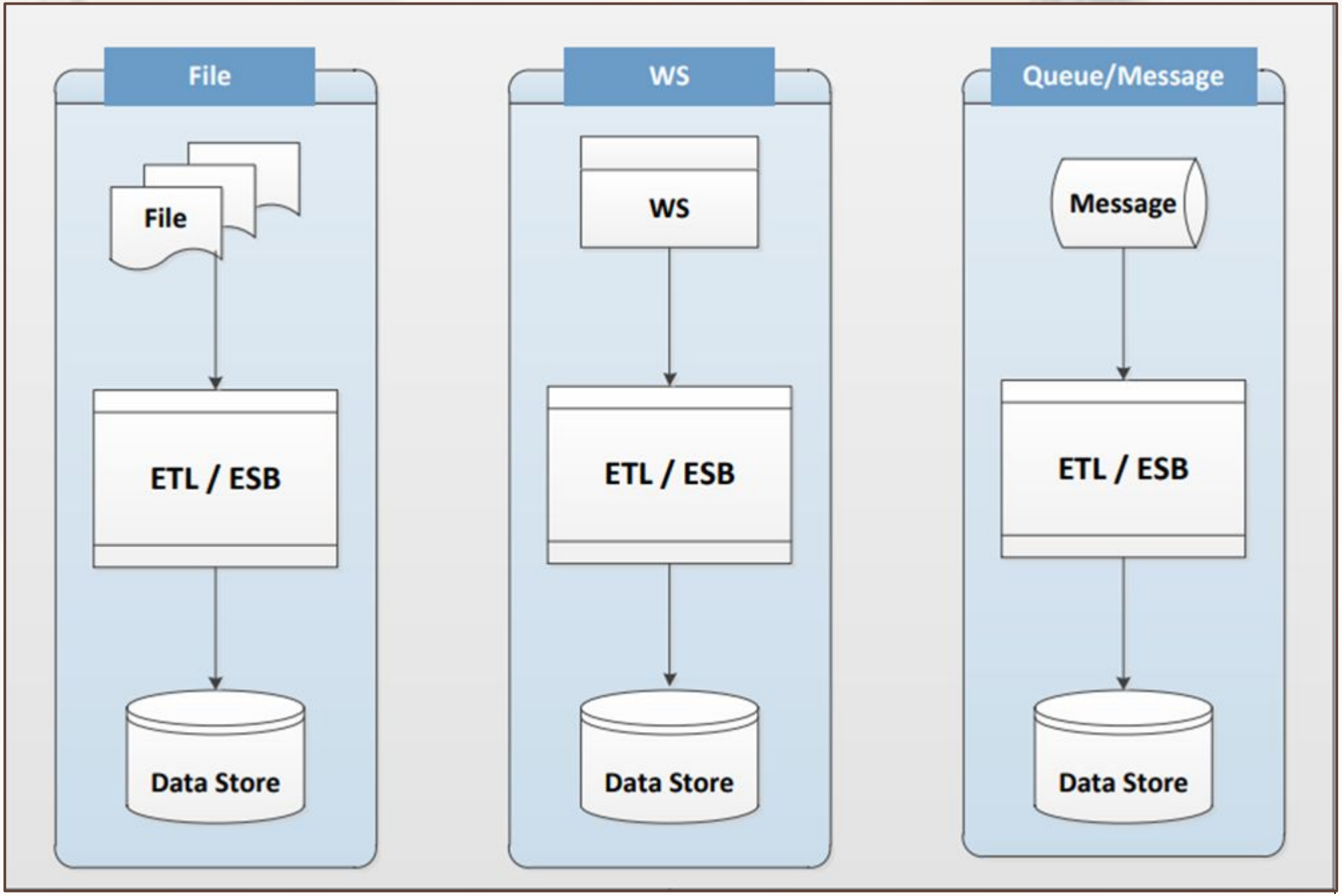


THE PROBLEM

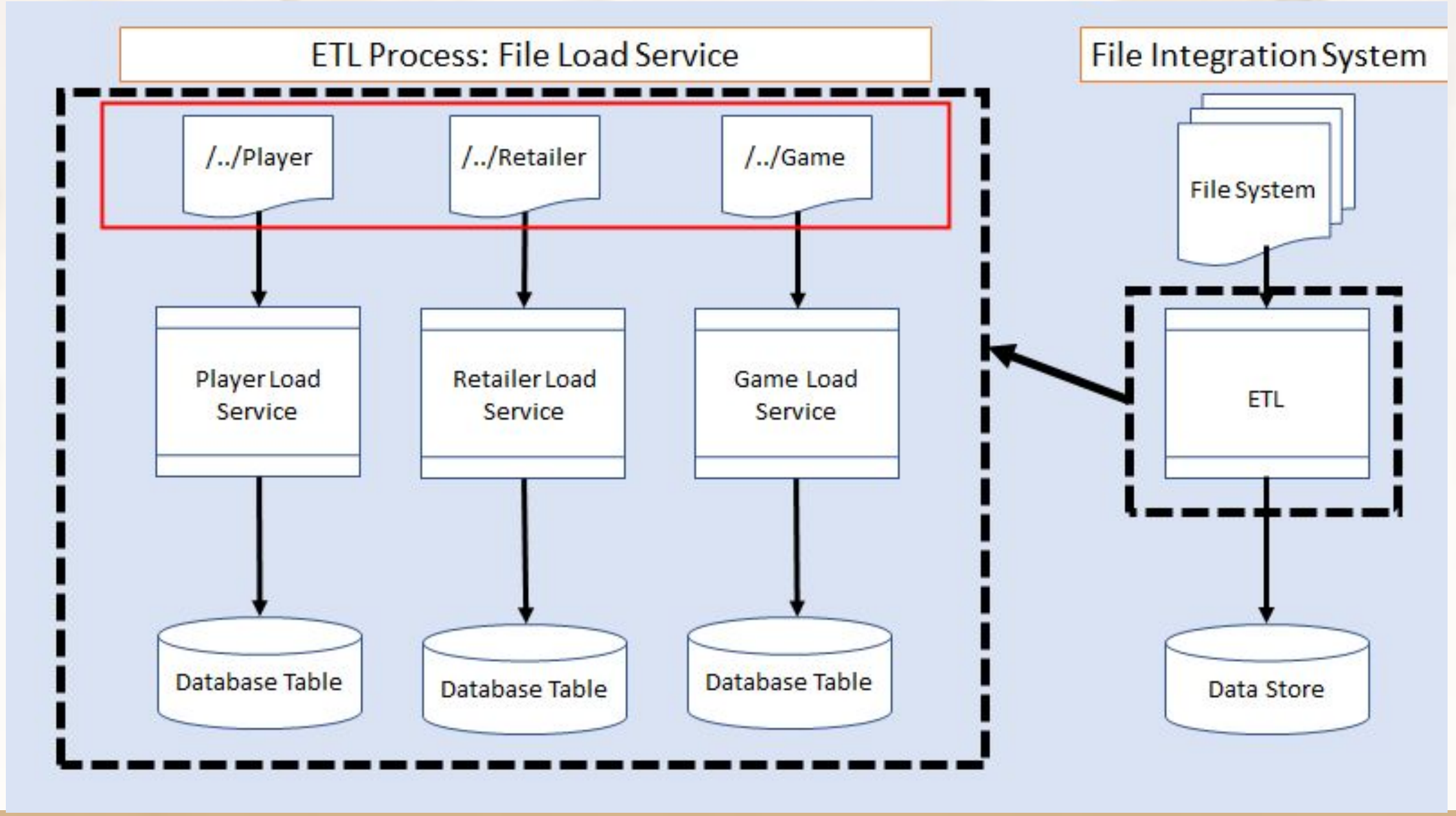
The Florida Lottery Department currently has 200+ Extract, Transform, Load (ETL) processes and has limited experience with Azure

Need to improve their current file storage ETL process for their game, retailer, and customer data.
Legacy Integration⟶ Modern Integration

Scope:
Integration
Solutions



Scope: File ETL Process



OBJECTIVES

Research and Implement Microsoft Azure Applications (Logic Apps & Data Factory) to automate the Florida Lottery Department's file ETL processes by providing two prototypes & decision matrices by April 2021.

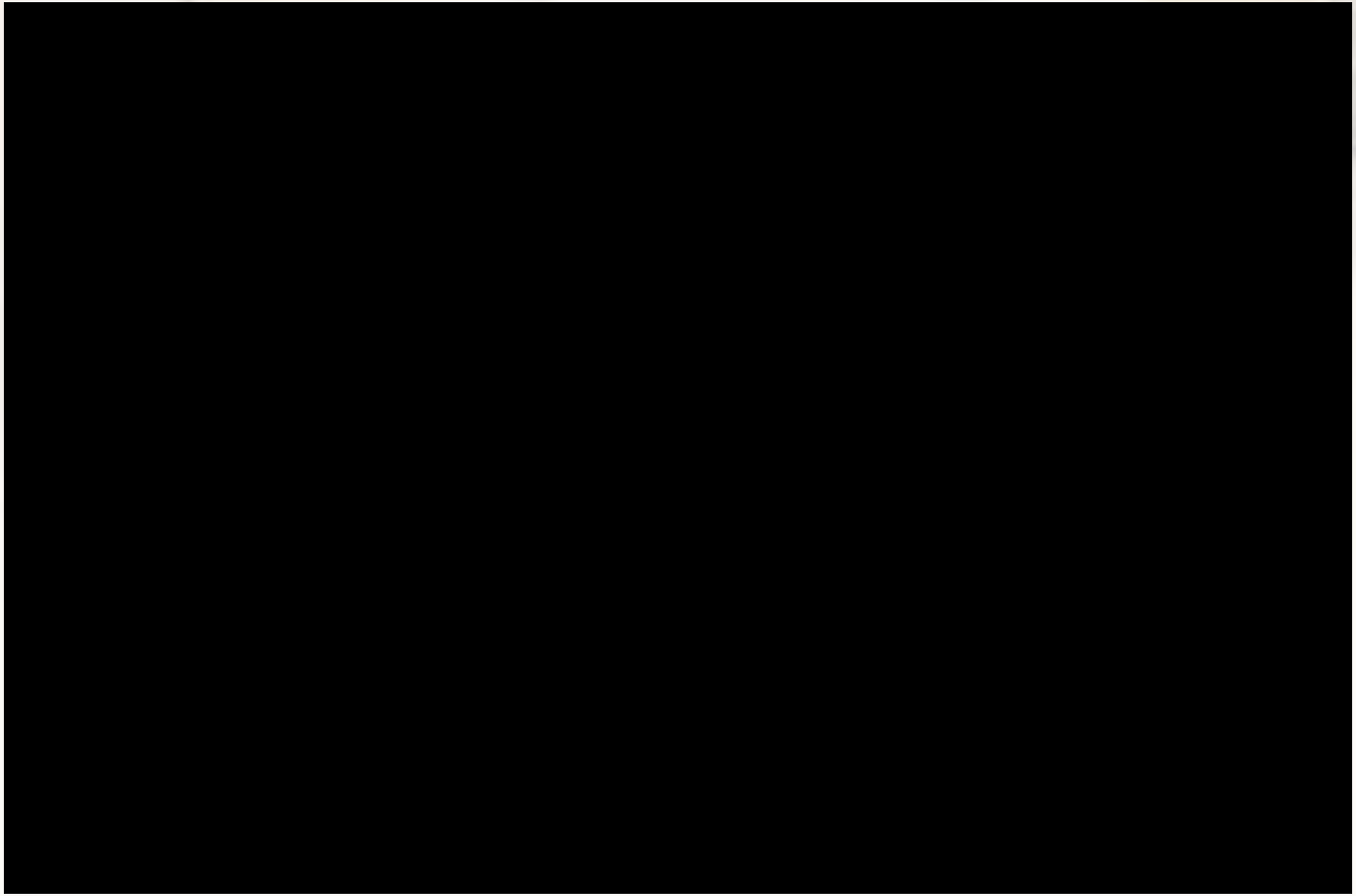
Access Local Directory

Retrieve Data Files

Transform File Contents via Azure

Store in SQL Database

AZURE LOGIC APPS OVERVIEW



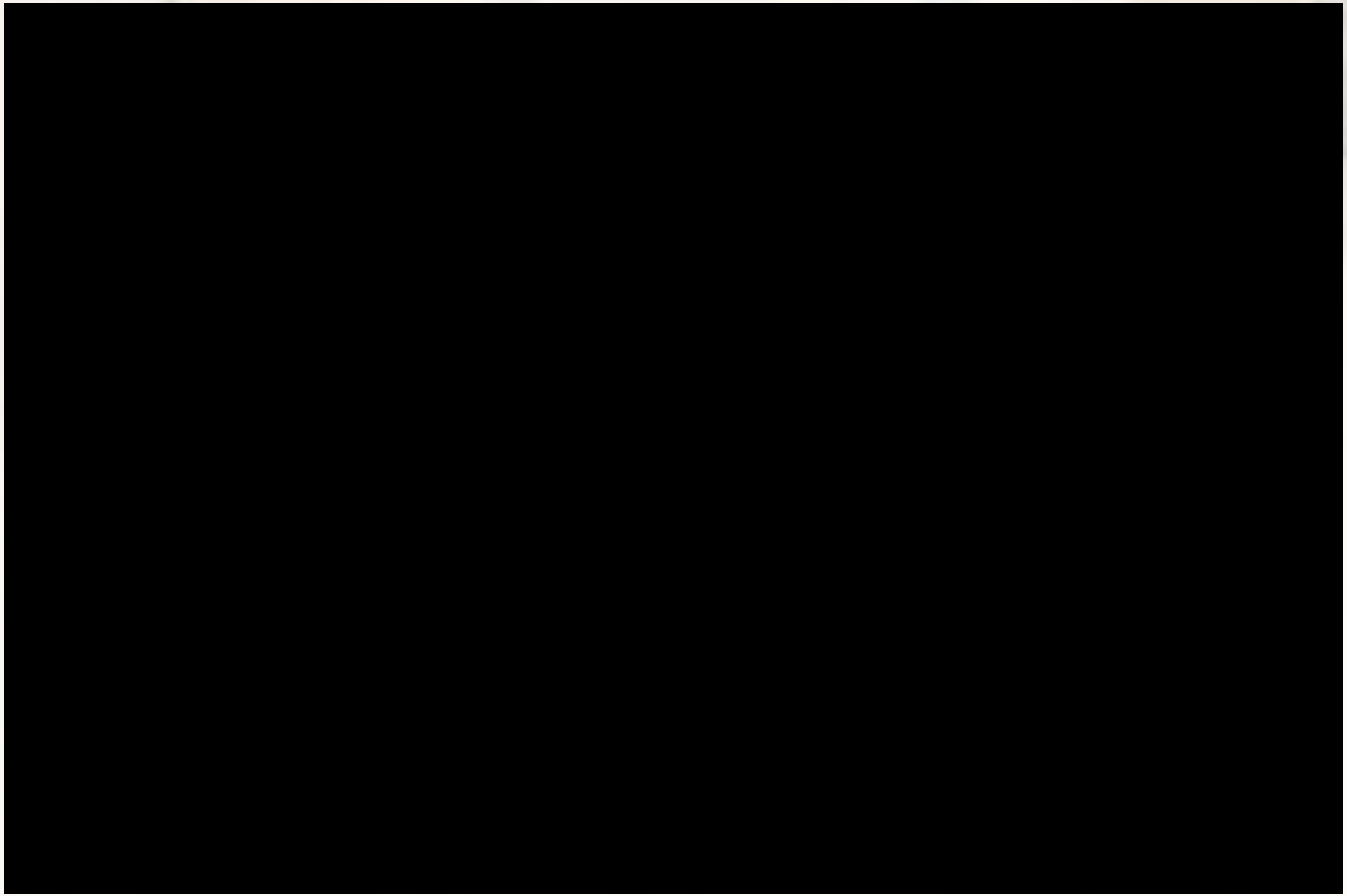
AZURE LOGIC APPS OVERVIEW

Intuitive User Interface
Includes a simplified interface with drag and drop features and heavy guidance

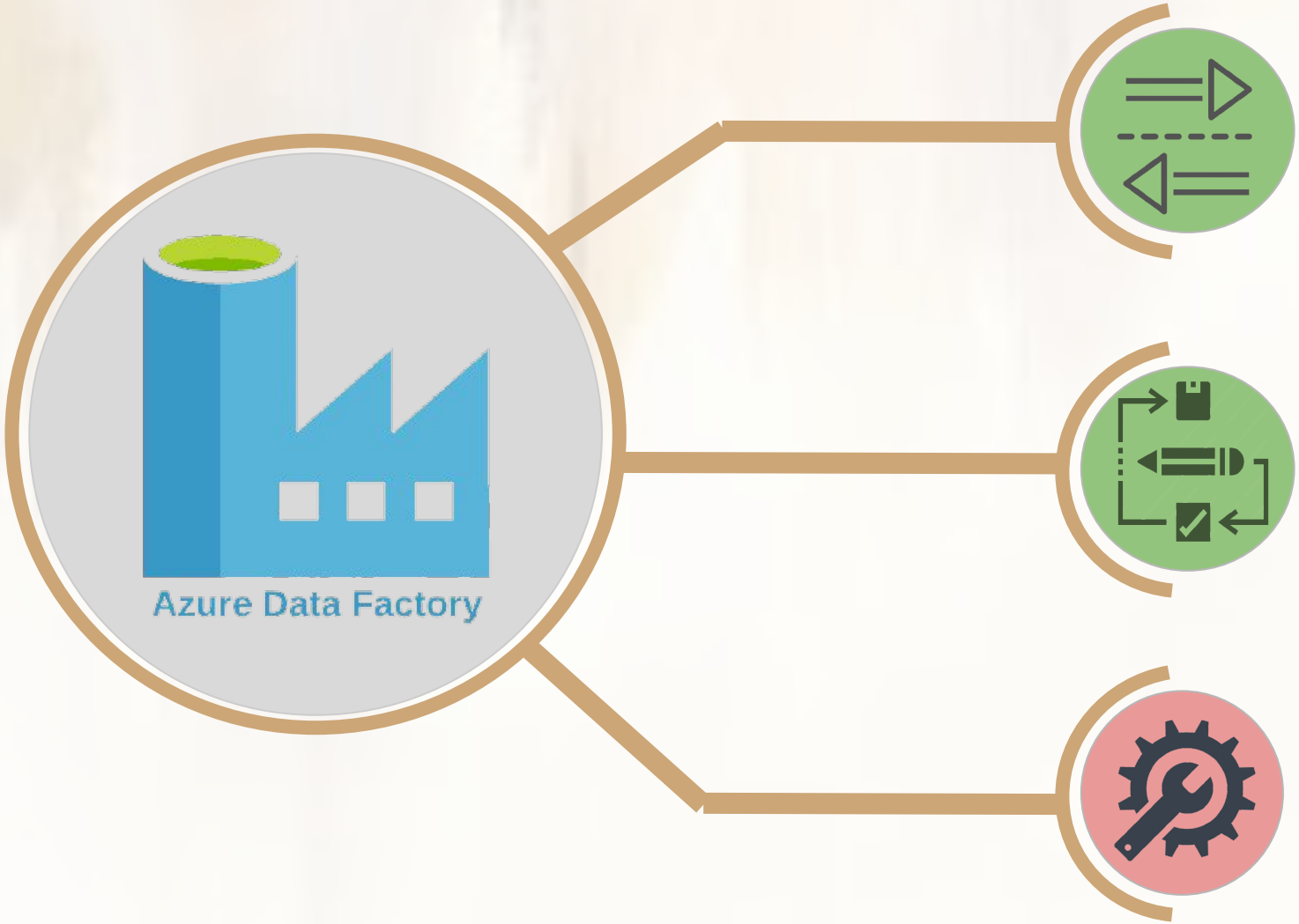
High Level of Connectivity
Provides built-in connectors that allow for integration with a variety of softwares (Outlook, Local Storage, etc.)

Lacks Capacity for Large Files
Not suited to process large amounts of data efficiently

AZURE DATA FACTORY OVERVIEW



AZURE DATA FACTORY OVERVIEW



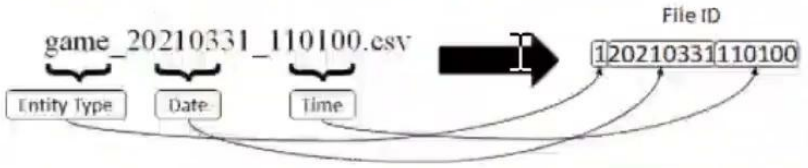
Facilitated Conversion
Efficiently copy CSV files into our SQL database in just one step.

Increased Modification Capabilities
Very easy to change the sequence of actions in a single workflow without having to re-do completed configuration.

Lacking Trigger Functionality
Did not offer functionality to trigger the pipeline when a file was added to a local directory.

Deliverable- Documentation

entity type (game, player, retailer), file date creation, and file time creation which are all extracted from the file name, as seen in figure _.



Since the file name was passed into the workflow as a parameter from Azure Logic Apps, and is stored in the pipeline parameter 'blobName'. Figure _ shows how all components required to construct the File ID are retrieved by utilizing Azure Data Factory's dynamic content capabilities along with advanced string manipulation, where the pipeline parameters are highlighted to show they are utilized.

Time component

```
@substring( , add(1,sub(indexof( , '_'), 1) , 6)
```

Date component

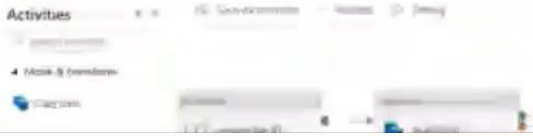
```
@substring( , add(1,indexOf( , '_'), 1), 8)
```

Overall File ID

```
@concat(if(startswith( , 'player'),1,
if(startswith( , 'game'),2, 3)), variables('date'), variables('time'))
```

Step 7: Configure Copy Data Activity to copy data from blob storage to SQL database

In the Activities Menu, expand Move & Transform to drag and drop the "Copy Data" activity to the pipeline.



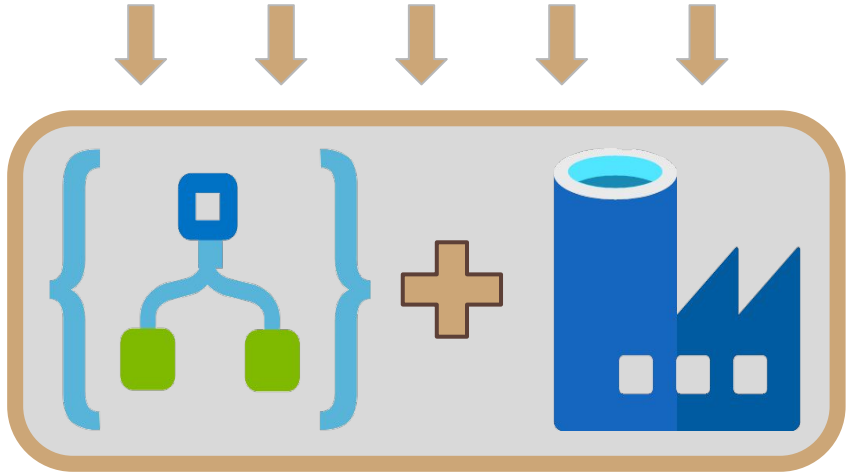
Final Recommendation

Features	Azure Logic Apps	Azure Data Factory
Simplified User Interface (e.g. accessible task panes, run buttons, code views)	☑	☑
Drag and Drop Configuration	☑	☑
Intuitiveness for Beginner Programmers	☑	☐
Efficient Modificiation of Workflow Sequence	☐	☑
Extensive Documentation available	☑	☑
Post Execution Analysis and Debugging	☑	☑
Windows Compatible	☑	☑
Ability to Connect to Microsoft Outlook	☑	☐
Ability to Connect to SQL Server	☑	☑
Wide access to third-party applications	☑	☑
Access Azure Storage Accounts	☑	☑
Ability to generate a structured PDF	☑	☑
View previous runs and triggers	☑	☑
Detailed dashboards of log and metric data through Azure Monitor	☑	☑
Alert and notification monitering through Azure Monitor	☑	☑
File System Trigger Capability	☑	☐
Stored Procedure Execution	☑	☑
Multiple File Processing	☑	☐
Dynamic Content Usage	☑	☑
Capacity to handle a large amount of data in single activity	☐	☑
Copy data directly from flat file to structured data store	☐	☑
Data Element Mapping (Explicit Mapping)	☐	☑
Efficiently copies data in one step	☐	☑
User Defined and System Variables	☑	☑
Parameterization	☑	☑
Facilitated CSV to JSON Transformation	☐	☑

Scored from 1 (lowest)- 5(highest)

Criteria	Weight	Azure Logic Apps	Azure Data Factory
User Friendliness	20%	4	3
Connectivity	30%	5	4
Supportability	10%	5	5
Functionality	40%	3	5
Total Score		4	4.3

Max Score= 5



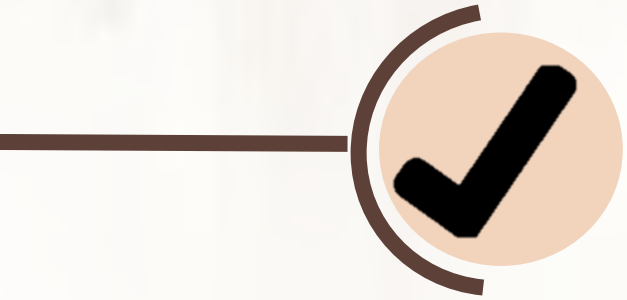
Our recommendation ⟶ Combine platforms in order to utilize both of their strengths.

Future Work

01	CSV → JSON Conversion in Logic Apps	• Utilize an API to facilitate conversion • Use BizTalk Server
02	Multiple File Processing	• Using a schedule-based trigger vs. event-based trigger
03	Azure Monitor	• Dashboards • Smart alerts for errors
04	Generating a PDF within the workflow	• PDF4Me Connector (Logic Apps) • Using Tabula in Data Factory
05	Azure DevOps/ Resource Templates	• Import/ Export Feature • ARM template
06	Explicit Mapping in Data Factory	• Utilizing user defined schemas

OBJECTIVES RECAP

Research Azure (Logic Apps & Data Factory)	Built Logic Apps Prototype to mimic ETL process	Build Data Factory Prototype to mimic ETL Process	Compare the two platforms via a comparison/ decision matrix	Give Final Recommendation
01/2021 →		2/21-3/21 →		4/21 →



Research and Implement Microsoft Azure Applications (Logic Apps & Data Factory) to automate the Florida Lottery Department's file ETL processes by providing two prototypes & decision matrices by April 2021.

THANK YOU!

ANY QUESTIONS?

jessicasilang@ufl.edu | ndanner97@ufl.edu | nicholasgonzalez@ufl.edu