

Azure OpenAI Batch API Accelerator

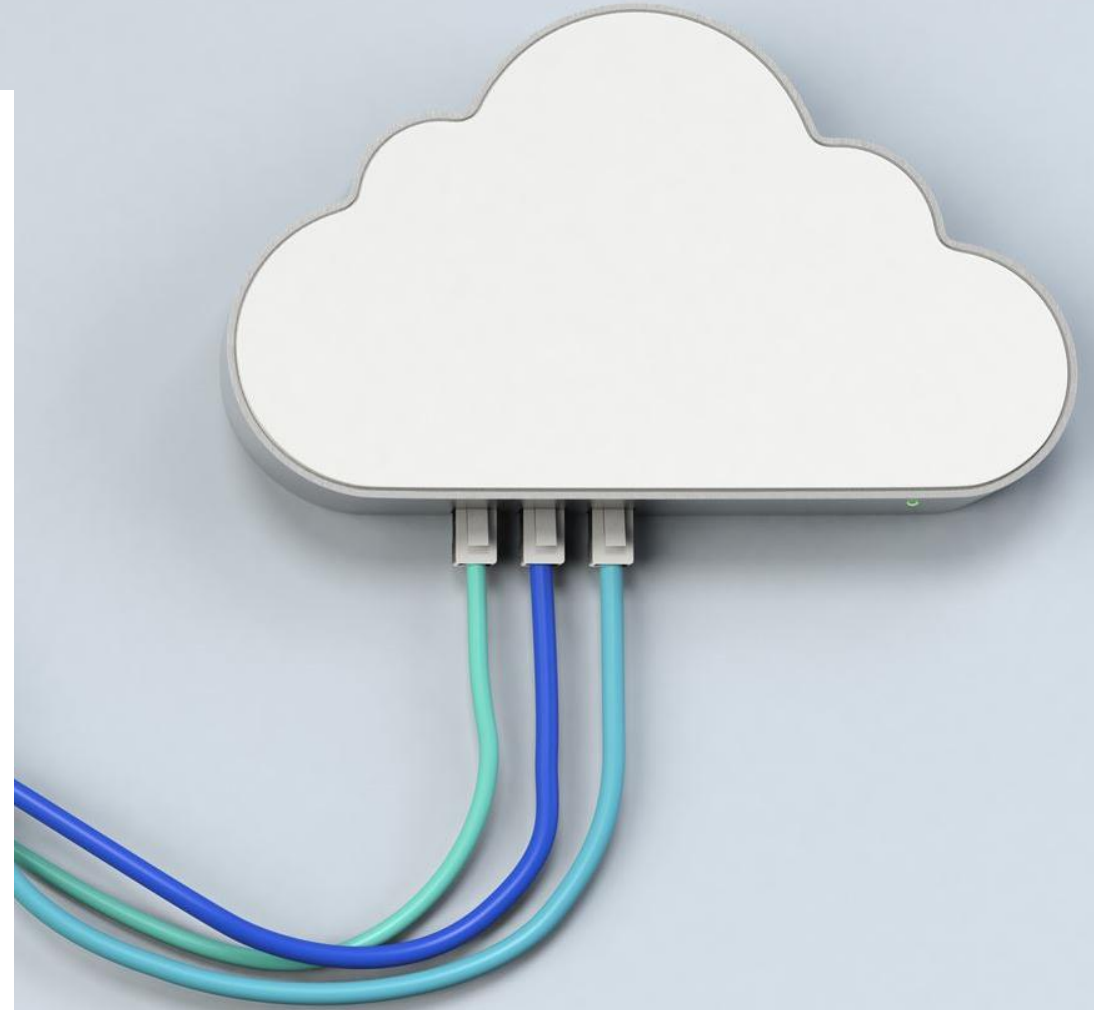
DJ Dean – Principal Cloud Solution Architect

Amit Mukherjee – Prin. Technical Specialist – GenAI

October 2nd, 2024

Why We Need the AOAI Batch API

- The Batch API significantly reduces costs for non-time sensitive workloads.
- Implementing multi-threaded/multi-process functionality can be difficult and time-consuming.
- The Batch API provides significant performance benefits when processing large quantities of files.



Customer Problem statement

- How to process tens of thousands of files to extract information
- How do track errors at the time of processing
- How to automate entire process
- How to make cost effective



AOAI Bath Accelerator

Automated Batch Job Submission and Creation

Multi-threaded Async Processing to Reduce Overall Processing Time


Automated Error Tracking

Multi-directory Hierarchy Support

Configurable Micro-batch Support

Automated Post-job Cleanup

Voice of Customer



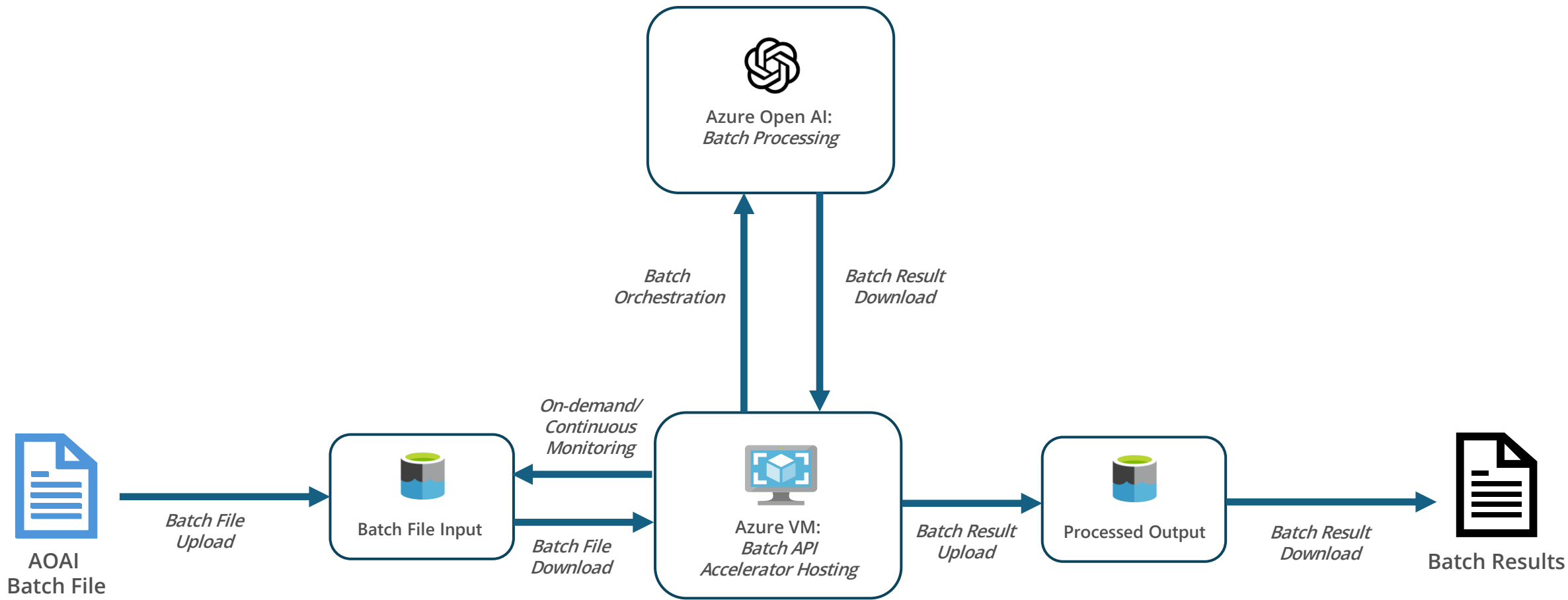
ontada[®]

*"Ontada is at the unique position of serving providers, patients and life science partners with data-driven insights. **We leverage the Azure OpenAI batch API to process tens of millions of unstructured documents efficiently, enhancing our ability to extract valuable clinical information. What would have taken months to process now takes just a week. This significantly improves evidence-based medicine practice and accelerates life science product R&D.** Partnering with Microsoft, we are advancing AI-driven oncology research, aiming for breakthroughs in personalized cancer care and drug development."*

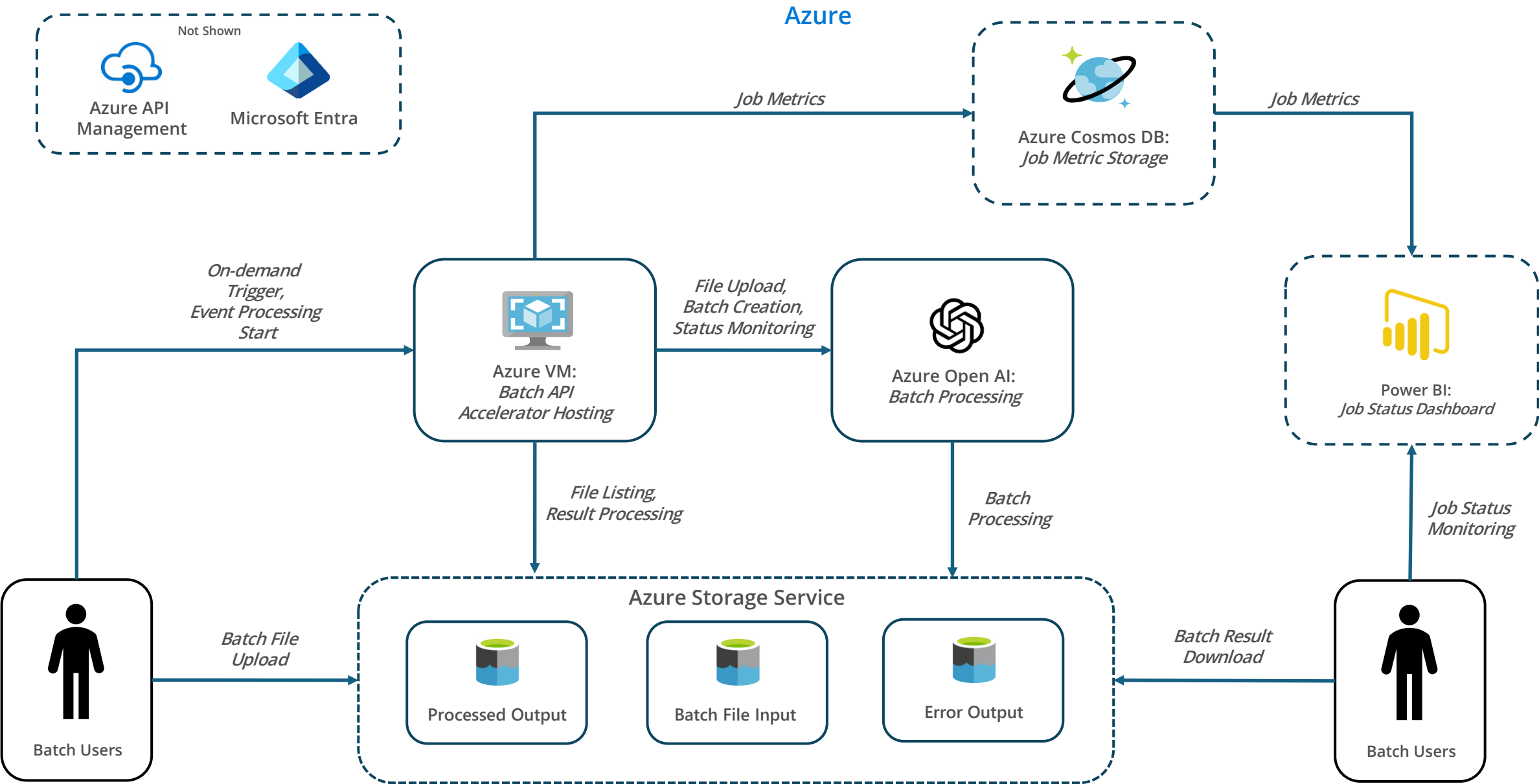
Sagran Moodley, Chief Innovation and Technology Officer, Ontada

AOAI Batch Accelerator Overview

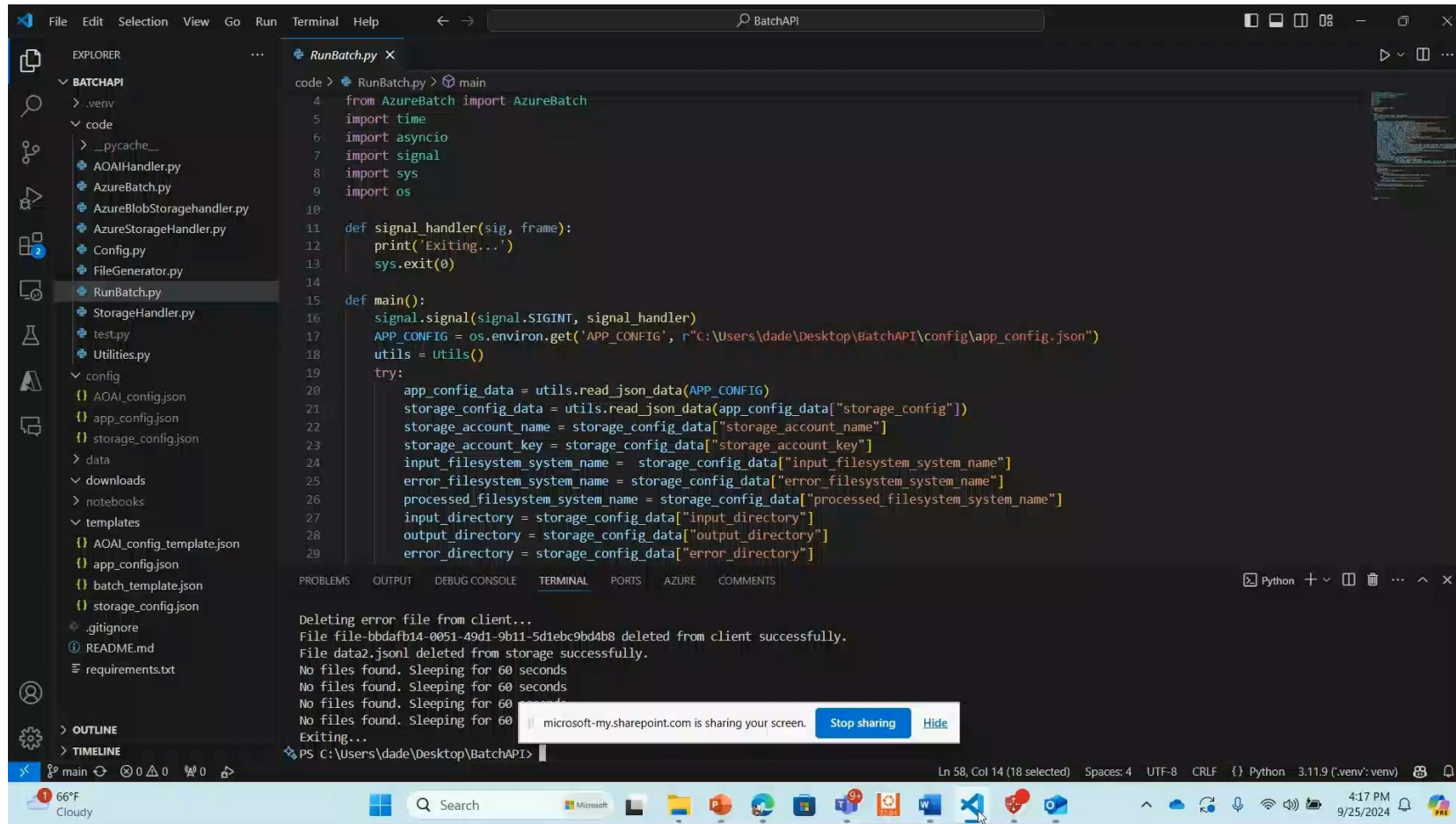
Azure



High-level Architecture (Detailed) | Data Flow and Consumption



Demo



The screenshot displays a Visual Studio Code editor window with a dark theme. The Explorer pane on the left shows a project structure for 'BatchAPI' with folders like '.venv', 'code', and 'config'. The file 'RunBatch.py' is selected and open in the editor. The code in the editor is a Python script that imports 'AzureBatch', 'time', 'asyncio', 'signal', 'sys', and 'os'. It defines a 'signal_handler' function and a 'main' function. The 'main' function reads JSON data from 'app_config.json' and 'storage_config.json', processes it, and prints the results. The terminal at the bottom shows the output of the script, which includes messages about deleting error files and sleeping for 60 seconds. A Windows taskbar is visible at the bottom of the screen.

```
code > RunBatch.py > main
4  from AzureBatch import AzureBatch
5  import time
6  import asyncio
7  import signal
8  import sys
9  import os
10
11 def signal_handler(sig, frame):
12     print('Exiting...')
13     sys.exit(0)
14
15 def main():
16     signal.signal(signal.SIGINT, signal_handler)
17     APP_CONFIG = os.environ.get('APP_CONFIG', r"C:\Users\dade\Desktop\BatchAPI\config\app_config.json")
18     utils = Utils()
19     try:
20         app_config_data = utils.read_json_data(APP_CONFIG)
21         storage_config_data = utils.read_json_data(app_config_data["storage_config"])
22         storage_account_name = storage_config_data["storage_account_name"]
23         storage_account_key = storage_config_data["storage_account_key"]
24         input_filesystem_system_name = storage_config_data["input_filesystem_system_name"]
25         error_filesystem_system_name = storage_config_data["error_filesystem_system_name"]
26         processed_filesystem_system_name = storage_config_data["processed_filesystem_system_name"]
27         input_directory = storage_config_data["input_directory"]
28         output_directory = storage_config_data["output_directory"]
29         error_directory = storage_config_data["error_directory"]
```

Deleting error file from client...
File file-bbdafb14-0051-49d1-9b11-5d1ebc9bd4b8 deleted from client successfully.
File data2.jsonl deleted from storage successfully.
No files found. Sleeping for 60 seconds
No files found. Sleeping for 60 seconds
No files found. Sleeping for 60 seconds
No files found. Sleeping for 60 seconds
Exiting...

PS C:\Users\dade\Desktop\BatchAPI>