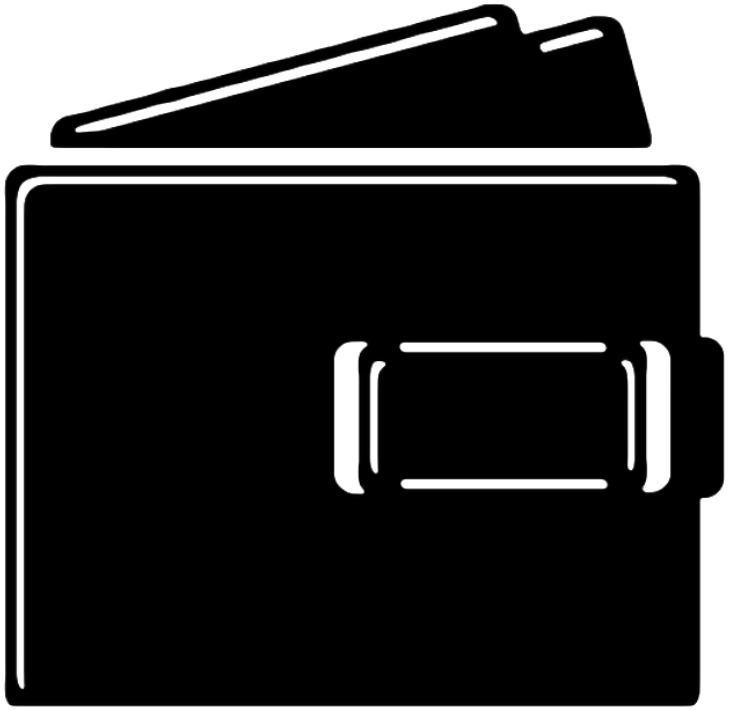
Case Study #4

Memi Lavi
www.memilavi.com



PayRawl

PayRawl

- Payment processing system
- Receives files from various sources
- Validates and processes the files
- Sends instruction files to banks
- Fully automatic, no UI

Requirements

Functional

What the system should do

1. Receive file to be processed
2. Validate and process the file
3. Work with various file formats
4. Perform various calculations on the file
5. Create bank payment file
6. Put the payment file in a designated folder
7. Keep log of all the activity for 7 years

Non-Functional

What the system should deal with

NFR - What We Ask

1. *"How many files per day?"* — 500
2. *"How long should the process take?"* — 1 min
3. *"What is the average size of a file?"* — 1MB
4. *"Can we tolerate data loss?"* — Absolutely Not!

Data Volume - Files

- 1 File = 1MB
- 500 files / day = 500MB / day

  => ~182GB / year

  => ~1.3TB / 7 years

Data Volume - Log

- Assuming each processing generates 500KB log data
- 500 files / day = 250MB log data / day

  => ~91GB log data / year

  => ~638GB log data / 7 years

Requirements

Functional

What the system should do

1. Receive file to be processed
2. Validate and process the file
3. Work with various file formats
4. Performs various calculations on the file
5. Create bank payment file
6. Put the payment file in a designated folder
7. Keep log of all the activity for 7 years

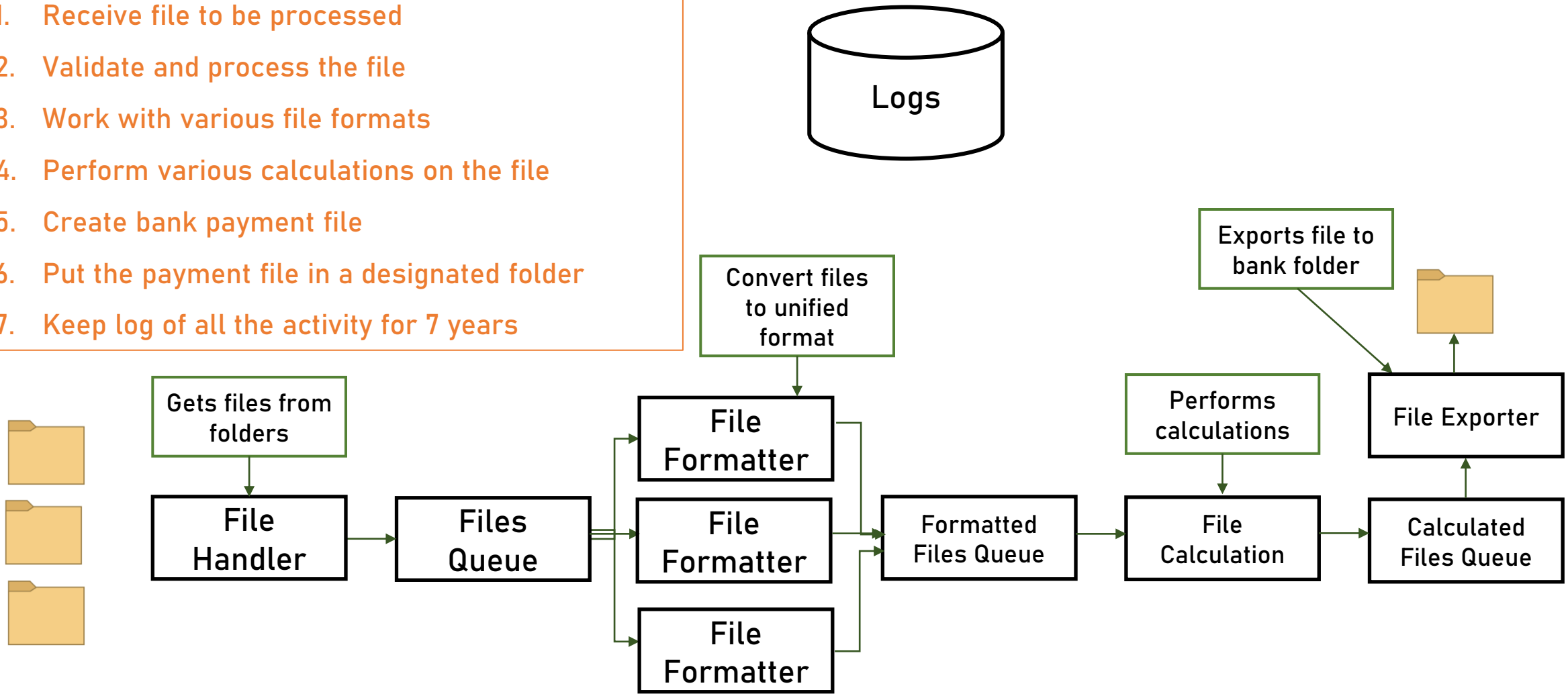
Non-Functional

What the system should deal with

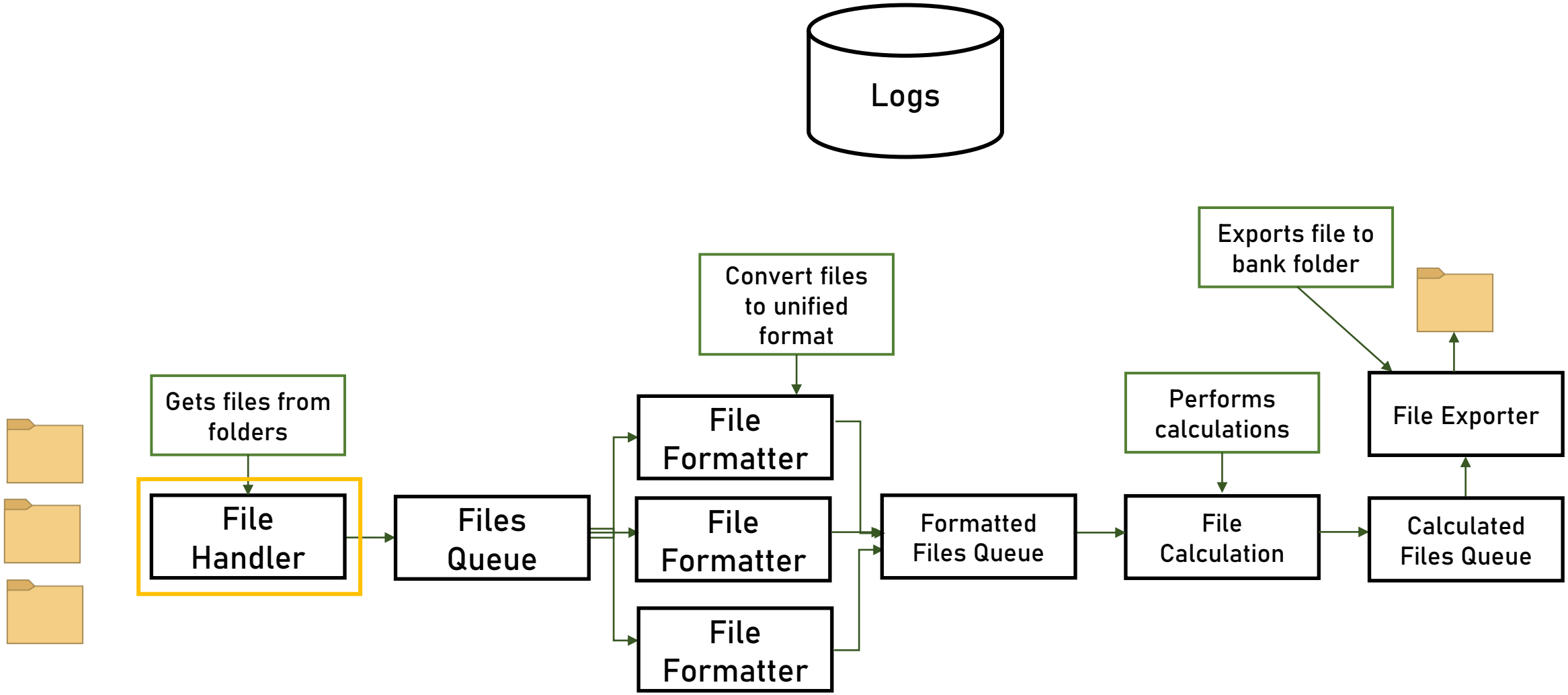
1. 500 files / day
2. No data loss
3. 1 min processing time
4. Activity log for 7 years
5. ~2TB / 7 years

Components

1. Receive file to be processed
2. Validate and process the file
3. Work with various file formats
4. Perform various calculations on the file
5. Create bank payment file
6. Put the payment file in a designated folder
7. Keep log of all the activity for 7 years



Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
File Exporter
File Formatter
File Handler
Files Queue
File Formatter
Formatted Files Queue
File Calculation
Calculated Files Queue
File Formatter

File Handler

What it does:

- Pulls payment files from folders

- Put the files in the queue

Application Type

- Web App & Web API ✗
- Mobile App ✗
- Console ✓
- Service ✓
- Desktop App ✗

Technology Stack

Considerations:

- Should be able to pull files from folders
- Should be able to connect to queue
- Not much else…

File Handler

Function App

- Designed for lightweight operations
- Great, built-in integration with many queue implementations
- Cost effective
- Autoscaling

Azure Functions

REGION: West Europe

TIER: Consumption

The first 400,000 GB/s of execution and 1,000,000 executions are free.

Executions

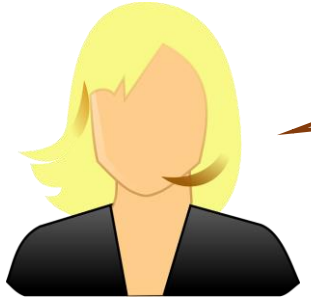
Memory size: 128 × 100 (Execution time (in milliseconds)) × 15000 (Executions per month) = $0.00

Requests

15,000 Execution count = $0.00

Upfront cost	$0.00
Monthly cost	$0.00

Technology Stack



This is a brand new company, we don’t have existing knowledge. What would you recommend?

Technology Stack

What we're looking for:

- Performance
- Community
- Cross Platform
- Easy to learn and use
- Evolving
- Great threading support

Technology Stack

Our candidates:

Technology Stack

Node is mainly for web apps, our component is a service, so…

Technology Stack

Performance:

Best fortunes responses per second, Dell R440 Xeon Gold + 10 GbE

Rnk	Framework	Best performance (higher is better)	
1	aspcore-ado-pg	300,613	100.0% (42.8%)
2	aspcore-rhtx-pg	294,096	97.8% (41.9%)
3	aspcore-mw-ado-pg	249,447	83.0% (35.5%)
4	aspcore-mw-dap-pg	217,793	72.4% (31.0%)
5	aspcore-ado-my	200,634	66.7% (28.6%)
6	aspcore-mw-ado-my	182,694	60.8% (26.0%)
7	aspcore-mw-ef-pg	173,958	57.9% (24.8%)
8	aspcore-mw-dap-my	171,681	57.1% (24.5%)
9	nodejs-mongodb-raw	125,743	41.8% (17.9%)
10	aspcore-mvc-ado-pg	120,731	40.2% (17.2%)
11	aspcore-mvc-dap-pg	115,062	38.3% (16.4%)
12	aspcore-mvc-ado-my	107,835	35.9% (15.4%)
13	aspcore-mvc-dap-my	102,905	34.2% (14.7%)
14	aspcore-mvc-ef-pg	95,922	31.9% (13.7%)
15	nodejs	91,086	30.3% (13.0%)
16	nodejs-chakra	91,058	30.3% (13.0%)
17	nodejs-postgres	63,532	21.1% (9.0%)
18	nodejs-mysql	57,995	19.3% (8.3%)
19	spring	30,891	10.3% (4.4%)

Source: https://www.techempower.com/benchmarks/#section=data-r18&hw=ph&test=fortune&l=zik0ot-f&p=zik0zj-zijocf-zijocf-4atpfj

Technology Stack

Community:

Jan 2020	Jan 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.896%	-0.01%
2	2		C	15.773%	+2.44%
3	3		Python	9.704%	+1.41%
4	4		C++	5.574%	-2.58%
5	7	^	C#	5.349%	+2.07%
6	5	v	Visual Basic .NET	5.287%	-1.17%
7	6	v	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%

Source: https://www.tiobe.com/tiobe-index/

Technology Stack

Cross Platform:

Technology Stack

Ease to learn and use:

Technology Stack

Evolving?

Java

Next versions planned until 2021

.NET Core

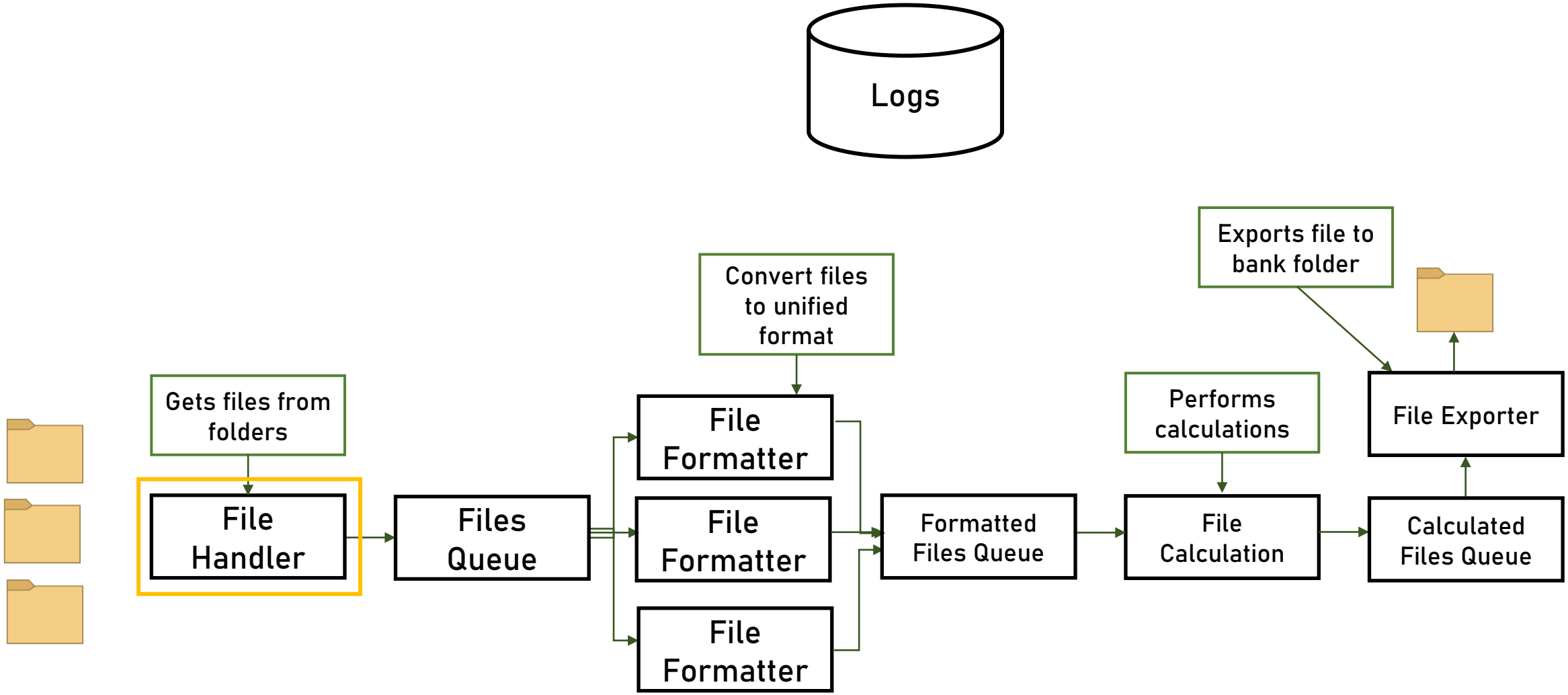
Roadmap announced until 2023

Technology Stack

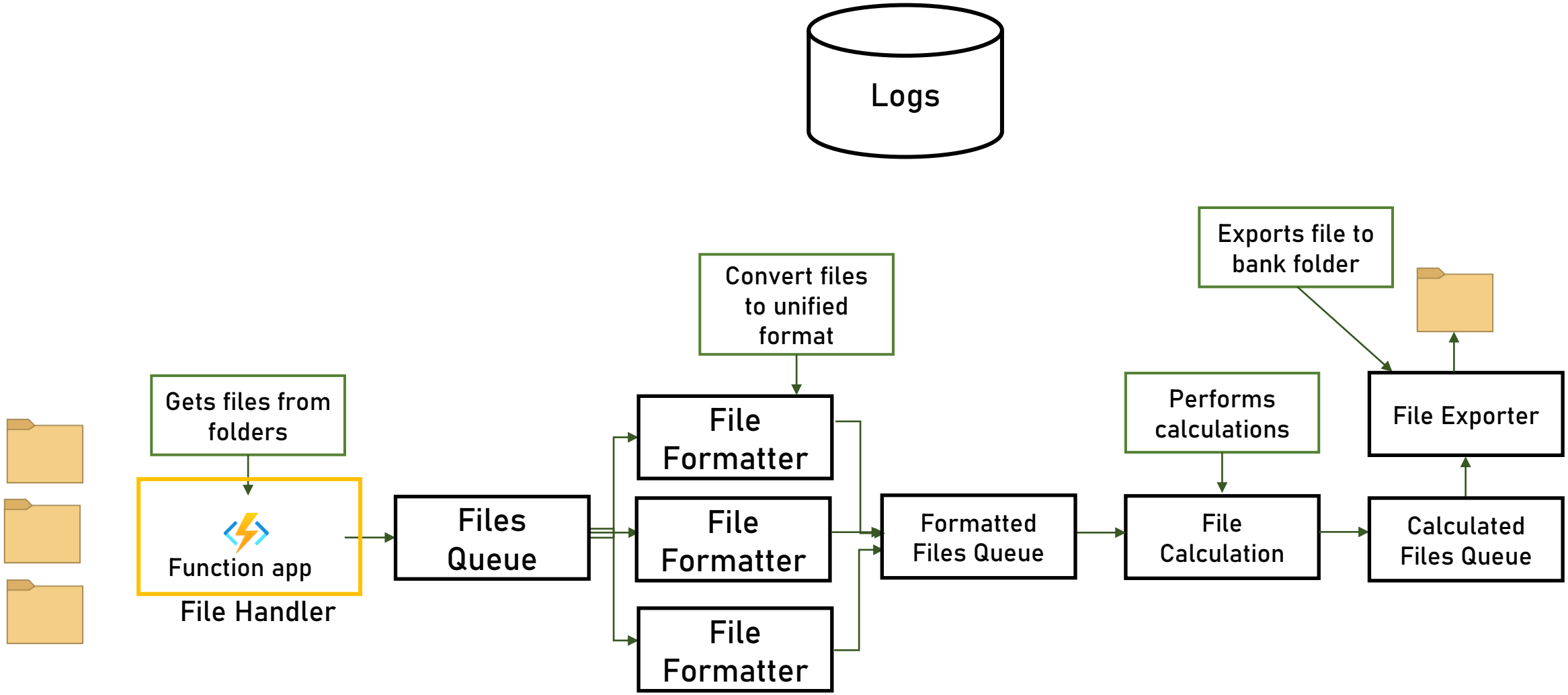
Threading support:

Technology Stack - Decision

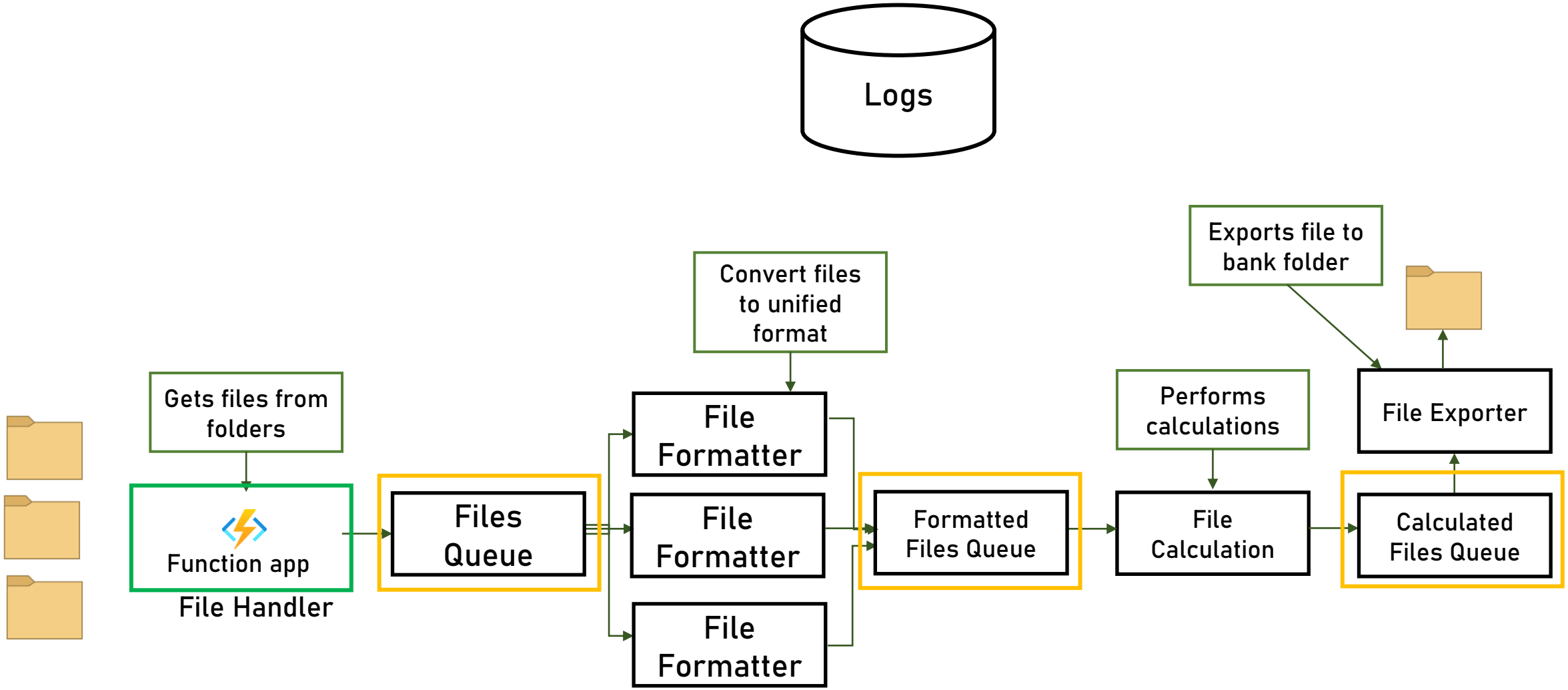
Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
File Exporter
File Formatter
File Handler
Files Queue
File Formatter
Formatted Files Queue
File Calculation
Calculated Files Queue
File Formatter

Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
File Exporter
File Formatter
Function app
Files Queue
File Formatter
Formatted Files Queue
File Calculation
Calculated Files Queue
File Formatter
File Handler

Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
File Exporter
File Formatter
Function app
Files Queue
File Formatter
Formatted Files Queue
File Calculation
Calculated Files Queue
File Formatter
File Handler

The Queue

- Passes payloads from logic unit to another
- Balances load
- Persists messages (Durability!)
- No high load expected (~500 messages / day)
- Message size ~1MB

The Queue

- Asynchronous
  - Which is good since we don’t have UI

Messaging in Azure

Service	Used For…	Guarantees Order	Max Msg Size	And also…
Storage Queue	Dead simple queueing	Yes	64KB	Extremely simple, no additional cost
Event Grid	Event driven architectures	No	1MB	Great integration with other services
Service Bus	Advanced queueing solutions	Yes	256KB	Advanced messaging features, durable
Event Hubs	Big data streaming	Yes	1MB	Low latency, designed for heavy load

Each TU supports up to 1k msgs / sec

Event Hubs

REGION: West Europe

TIER: Standard

Units

Maximum throughput units: 20. Up to 1 MB per second of ingress events. Up to 2 MB per second of egress events.

1 Throughput units	×	730 Hours	×	$0.030 Per unit/hour	=	$21.90

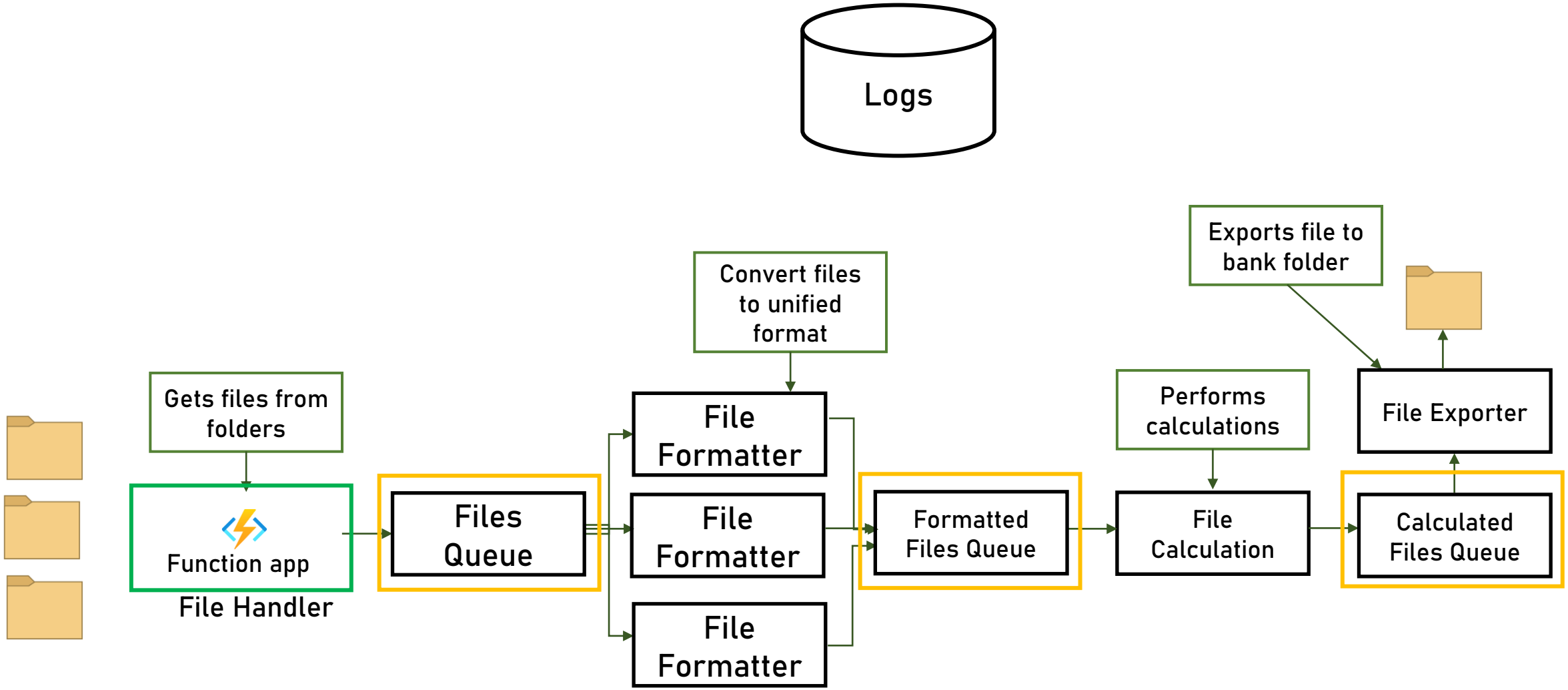
Enable Capture

Ingress

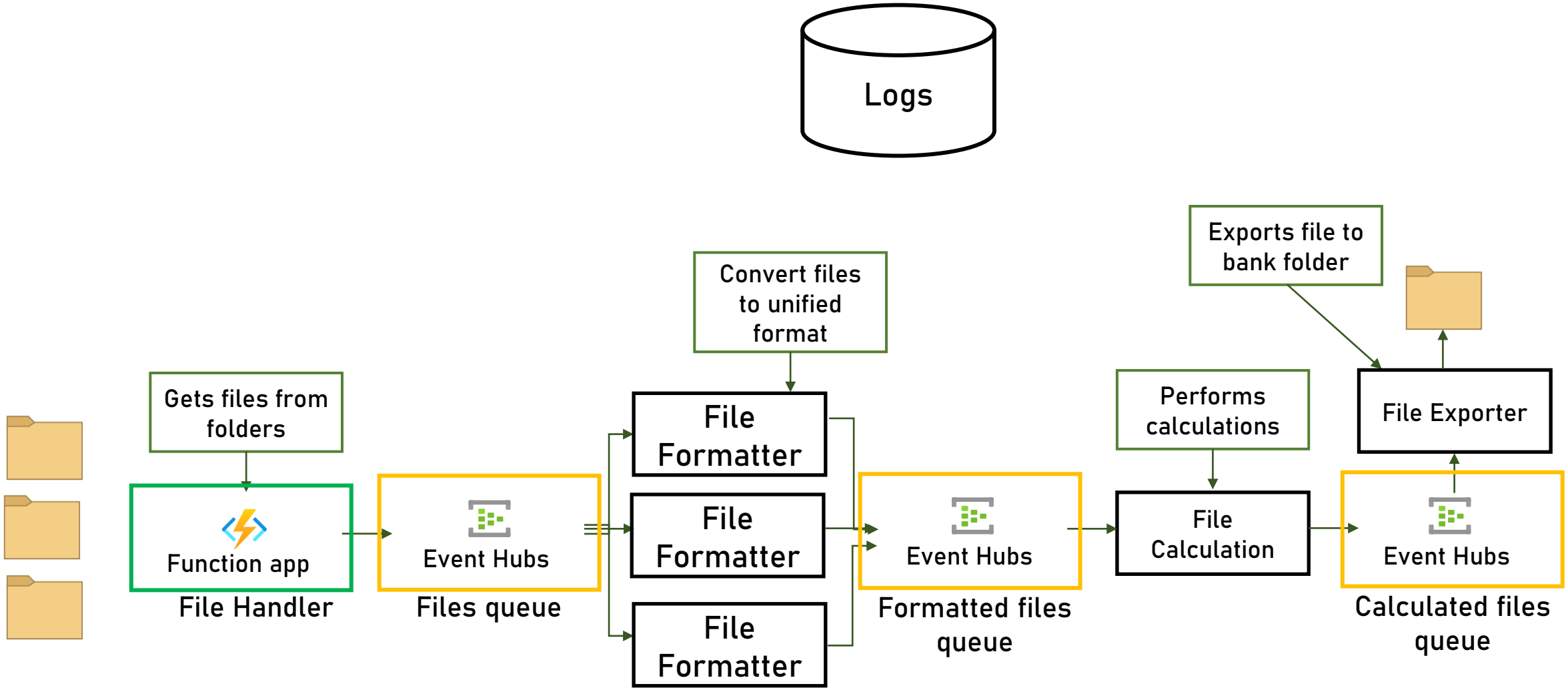
0.015 Million Events per month	×	$0.028 Per million Events / month			=	$0.01

Upfront cost	$0.00
Monthly cost	$21.90

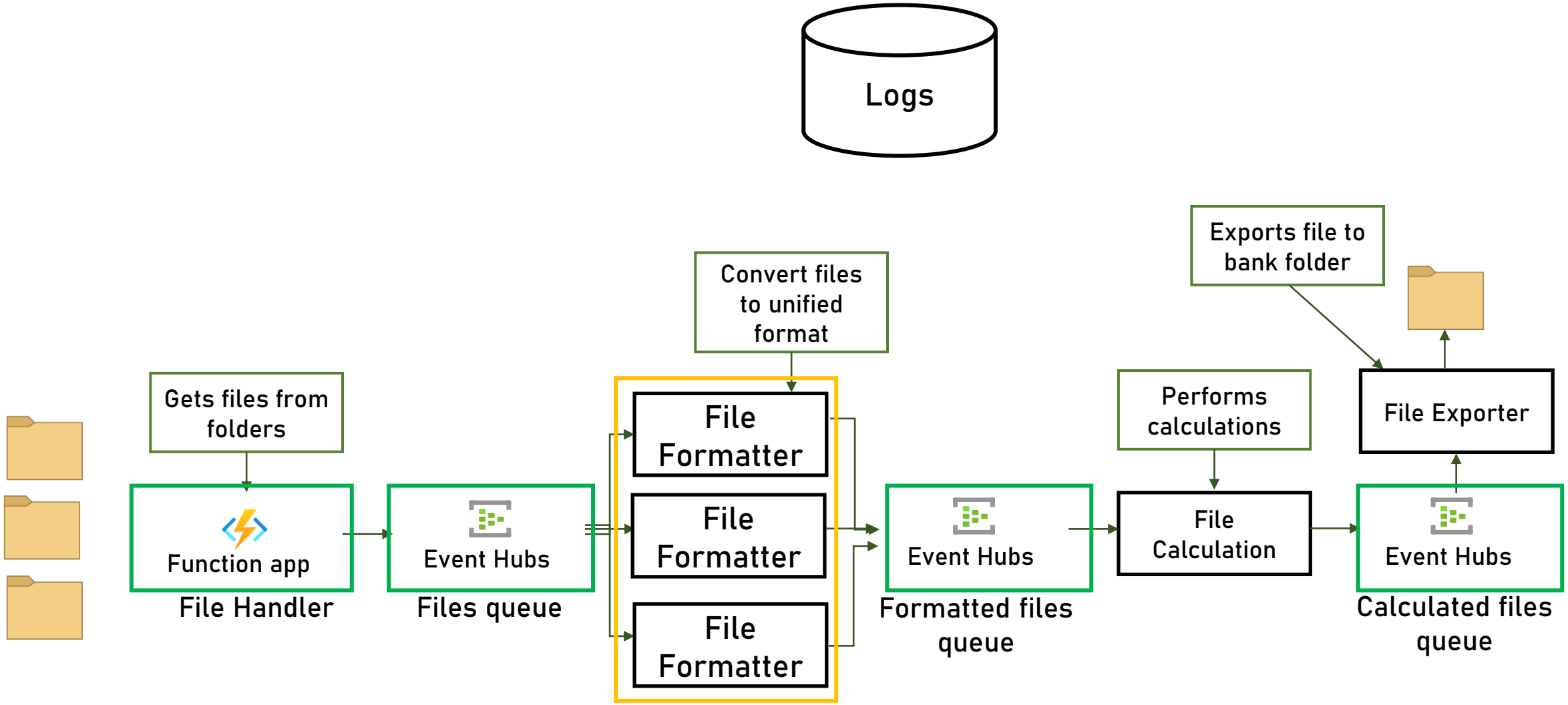
Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
File Exporter
File Formatter
Function app
Files Queue
File Formatter
Formatted Files Queue
File Calculation
Calculated Files Queue
File Formatter
File Handler

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
File Formatter
File Formatter
File Formatter
Event Hubs
Formatted files queue
Performs calculations
File Calculation
Event Hubs
Calculated files queue
Exports file to bank folder
File Exporter

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
File Formatter
File Formatter
File Formatter
Event Hubs
Formatted files queue
Performs calculations
File Calculation
Event Hubs
Calculated files queue
File Exporter
Exports file to bank folder

File Formatter

What it does:

- Receives files from its specific topic

- Validates and formats the file to unified format

- Puts the new file in a queue

- New formatters will be developed for new file formats

Application Type

- Web App & Web API ✗
- Mobile App ✗
- Console ✓
- Service ✓
- Desktop App ✗

File Formatter

Function App

- Designed for lightweight operations
- Great, built-in integration with many queue implementations
- Cost effective
- Autoscaling

Azure Functions

REGION: West Europe

TIER: Consumption

The first 400,000 GB/s of execution and 1,000,000 executions are free.

Executions

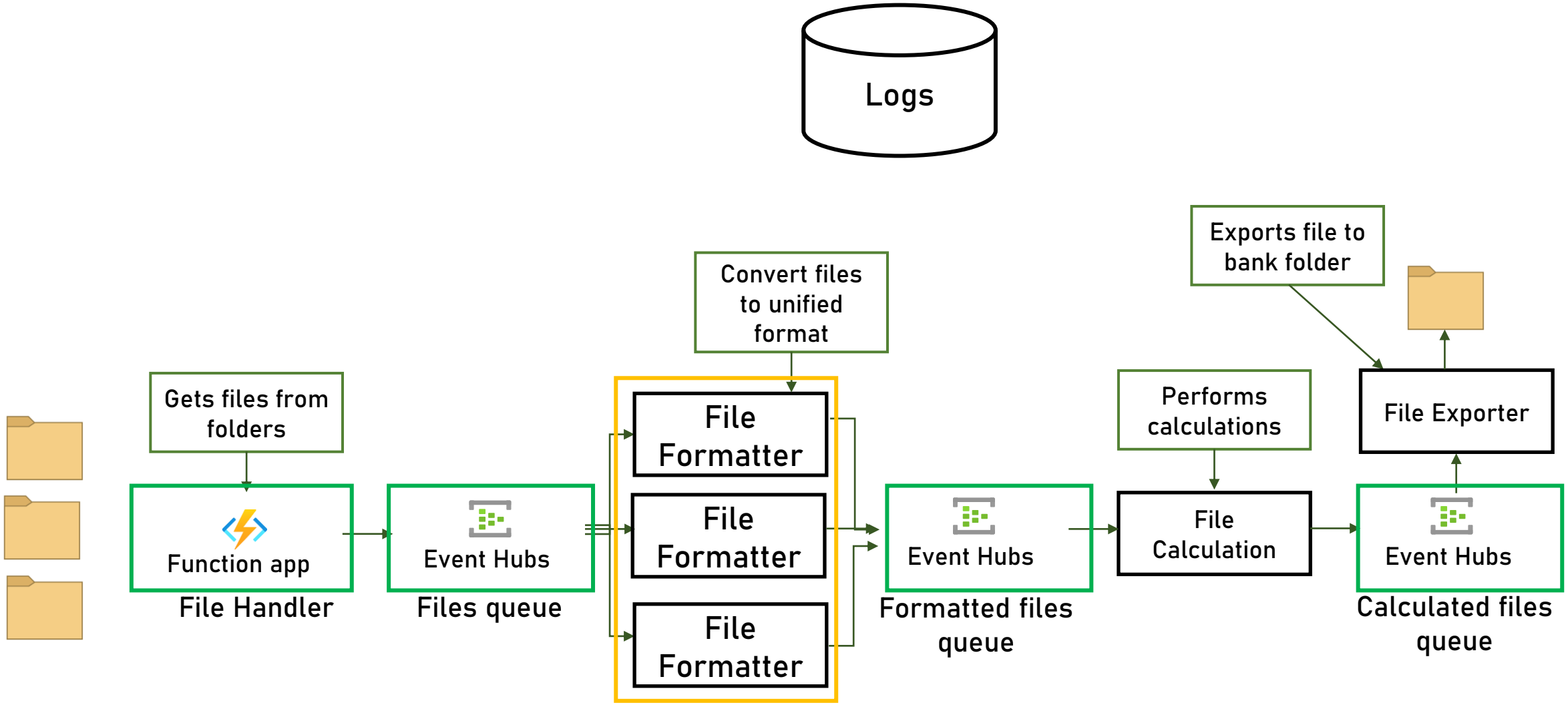
Memory size: 128 × 100 (Execution time (in milliseconds)) × 15000 (Executions per month) = $0.00

Requests

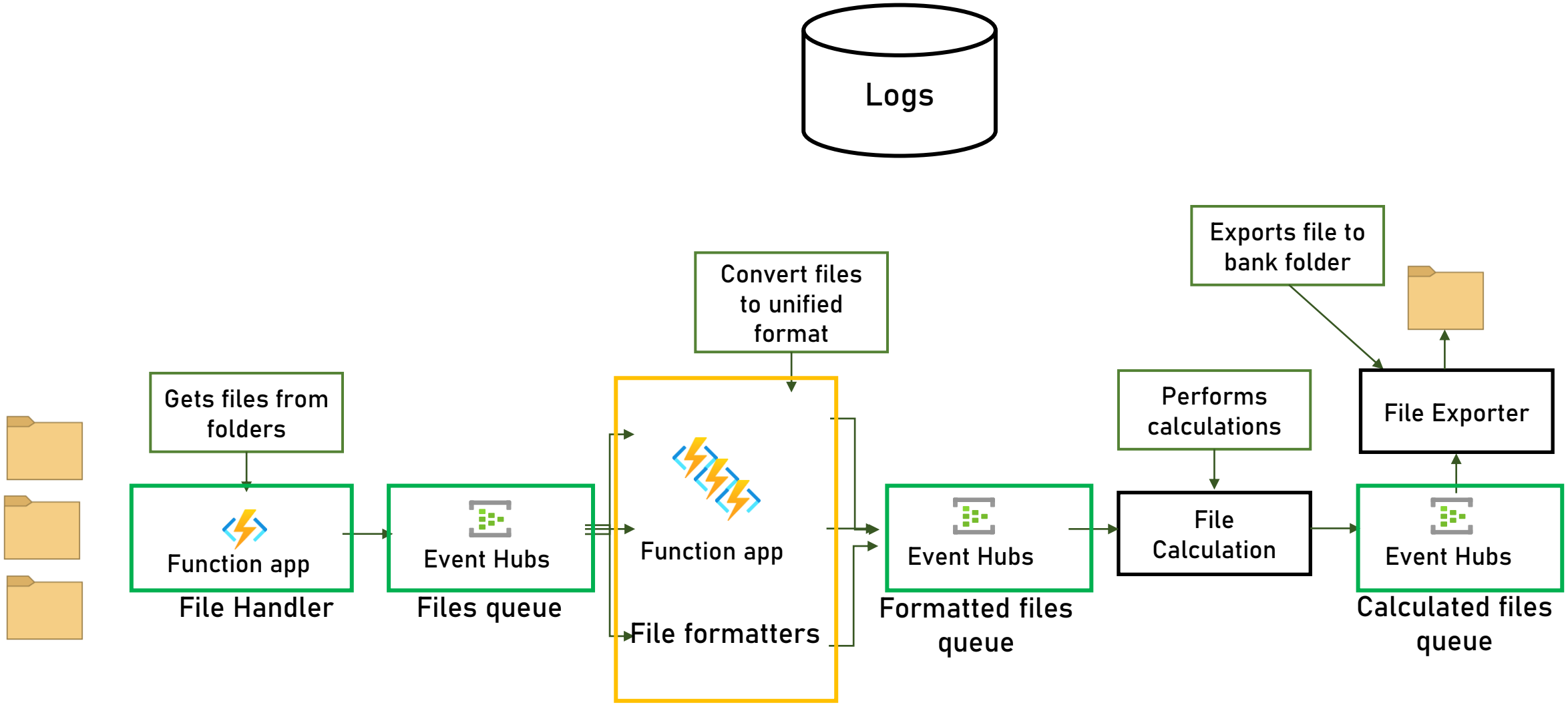
15,000 Execution count = $0.00

Upfront cost	$0.00
Monthly cost	$0.00

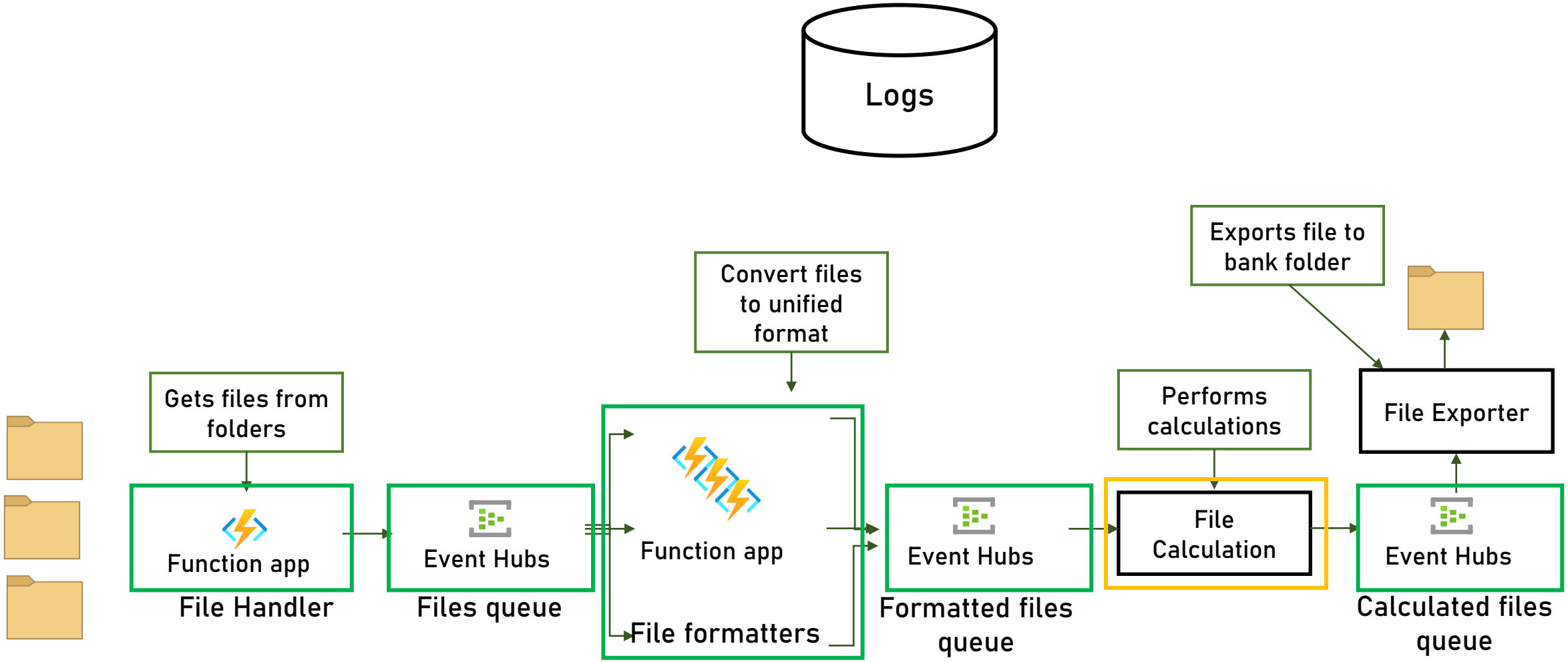
Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
File Formatter
File Formatter
File Formatter
Event Hubs
Formatted files queue
Performs calculations
File Calculation
Event Hubs
Calculated files queue
File Exporter
Exports file to bank folder

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
File Calculation
Exports file to bank folder
File Exporter
Event Hubs
Calculated files queue

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
File Calculation
Exports file to bank folder
File Exporter
Event Hubs
Calculated files queue

File Calculation

What it does:

- Receives files from the queue

- Performs some calculations on the data

- Puts the new file in a queue

File Calculation

Quite similar to the file formatter, so:

Function App

Azure Functions

REGION: West Europe

TIER: Consumption

The first 400,000 GB/s of execution and 1,000,000 executions are free.

Executions

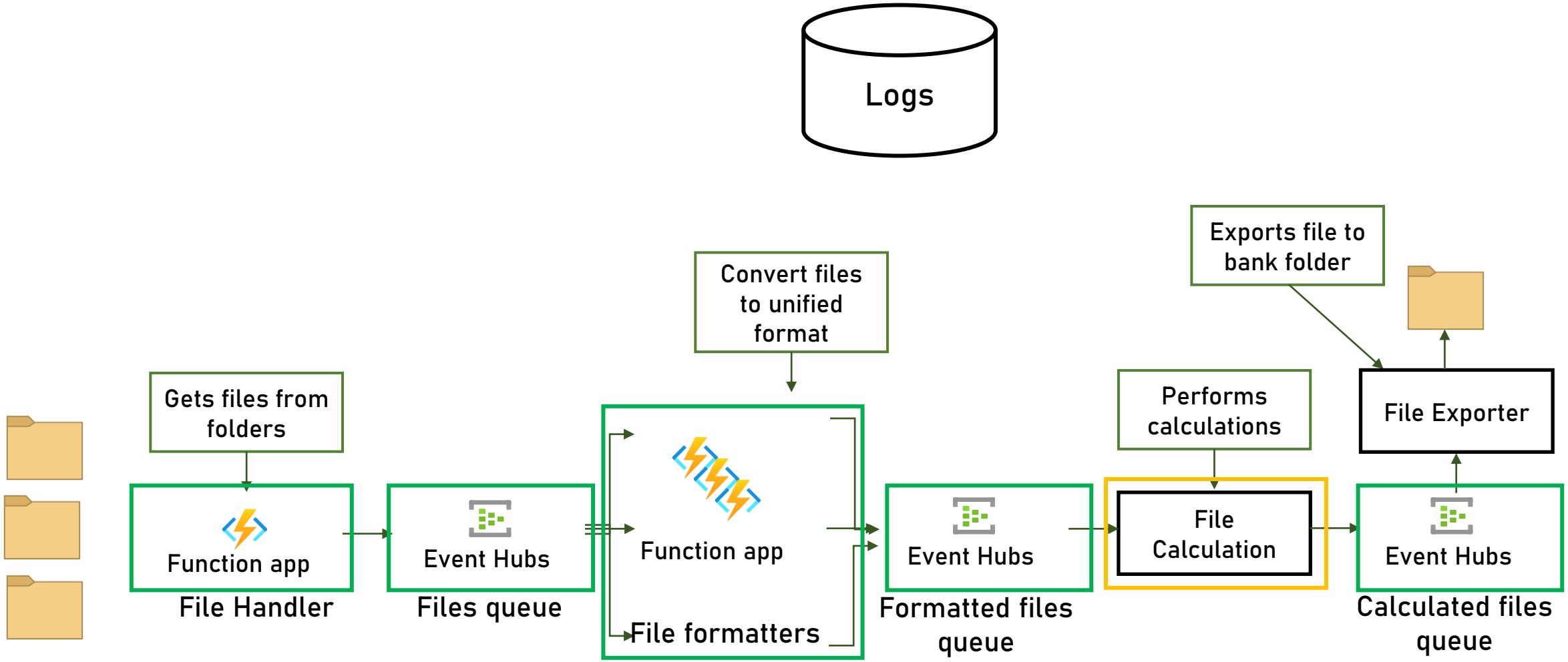
Memory size: 128 × 100 (Execution time (in milliseconds)) × 15000 (Executions per month) = $0.00

Requests

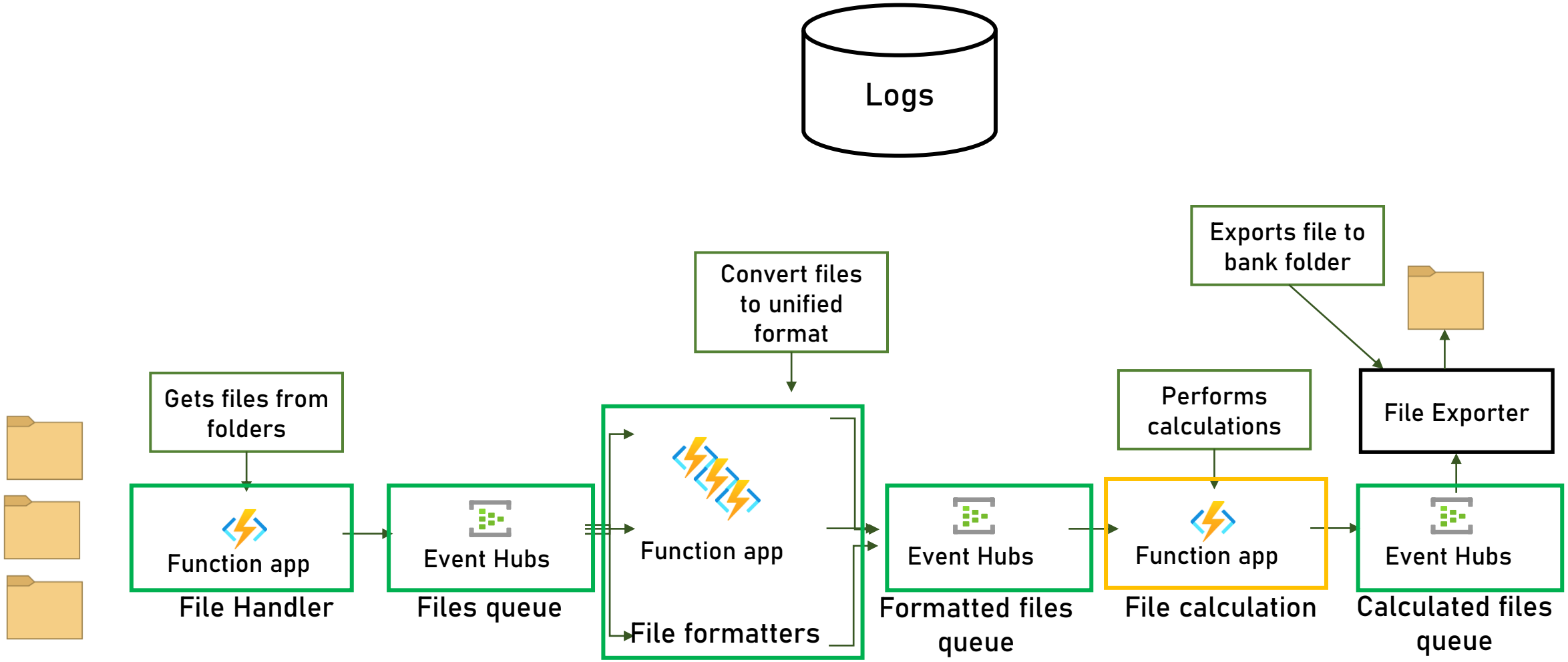
15,000 Execution count = $0.00

Upfront cost	$0.00
Monthly cost	$0.00

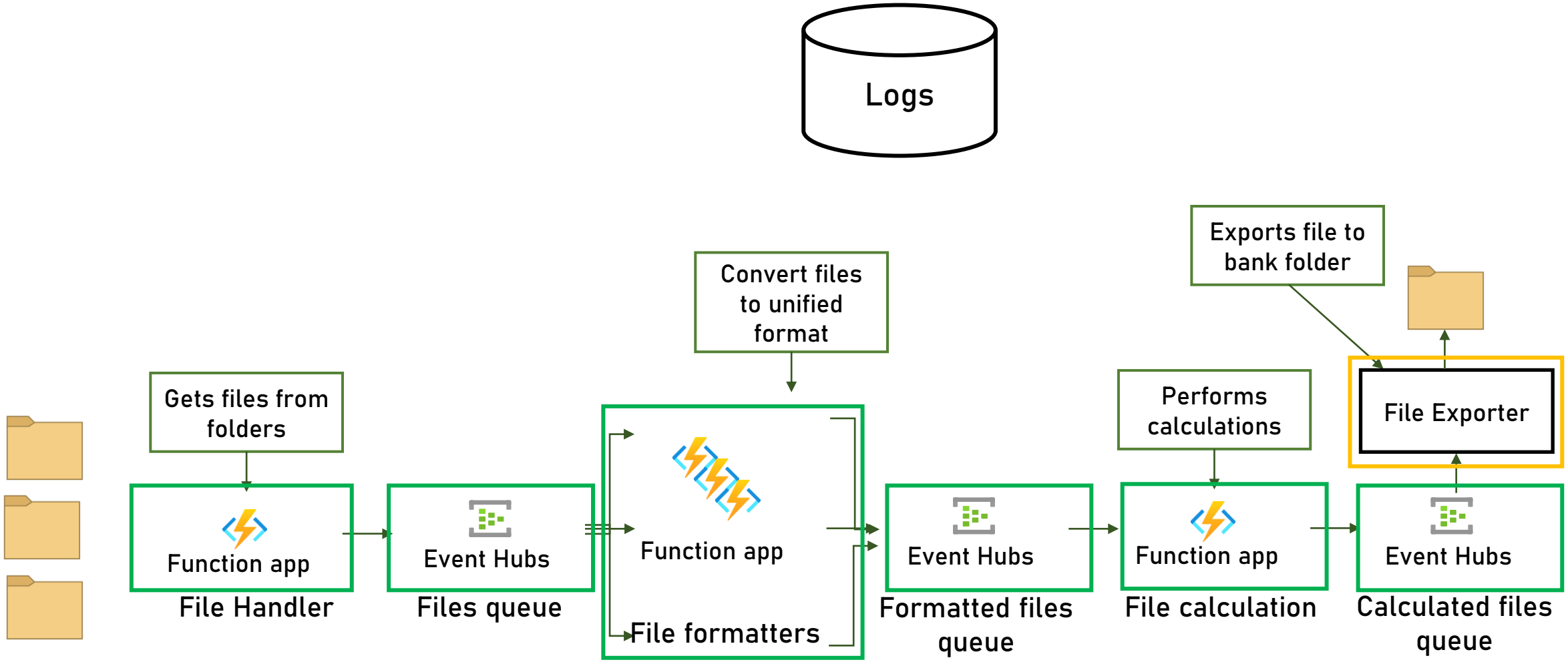
Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
File Calculation
Exports file to bank folder
File Exporter
Event Hubs
Calculated files queue

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
Function app
File calculation
Exports file to bank folder
File Exporter
Event Hubs
Calculated files queue

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
Function app
File calculation
Exports file to bank folder
File Exporter
Event Hubs
Calculated files queue

File Exporter

What it does:

- Receives files from the queue

- Puts the file in the bank's folder

File Exporter

Quite similar to the file calculation, so:

Function App

Azure Functions

REGION: West Europe

TIER: Consumption

The first 400,000 GB/s of execution and 1,000,000 executions are free.

Executions

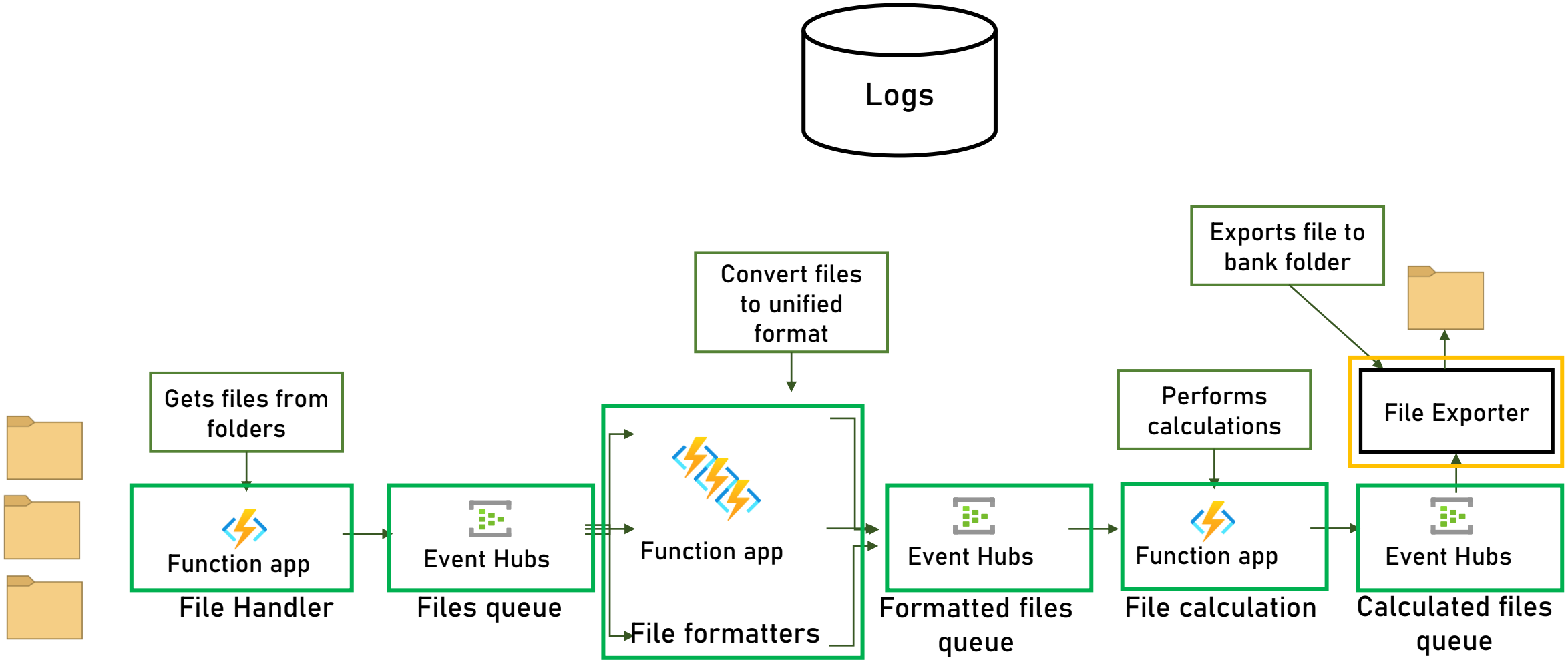
Memory size: 128 × 100 (Execution time (in milliseconds)) × 15000 (Executions per month) = $0.00

Requests

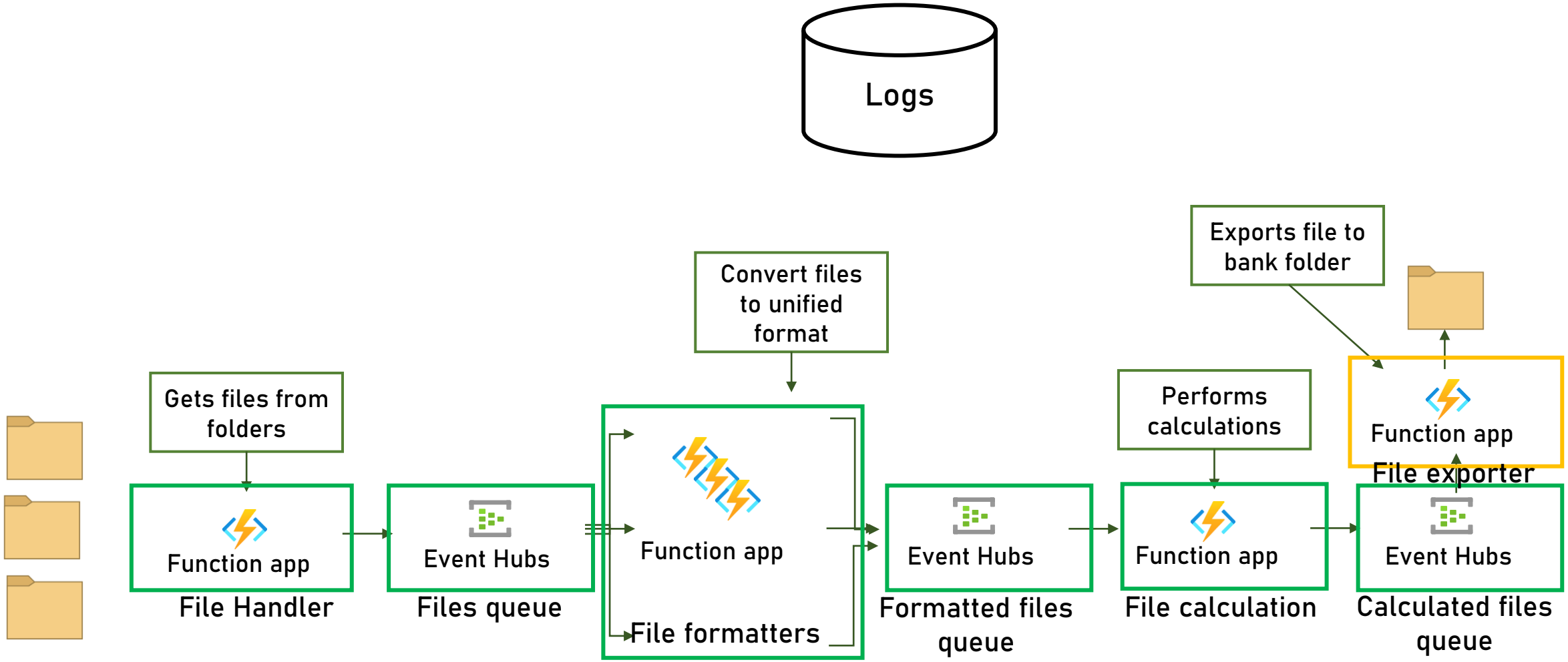
15,000 Execution count = $0.00

Upfront cost	$0.00
Monthly cost	$0.00

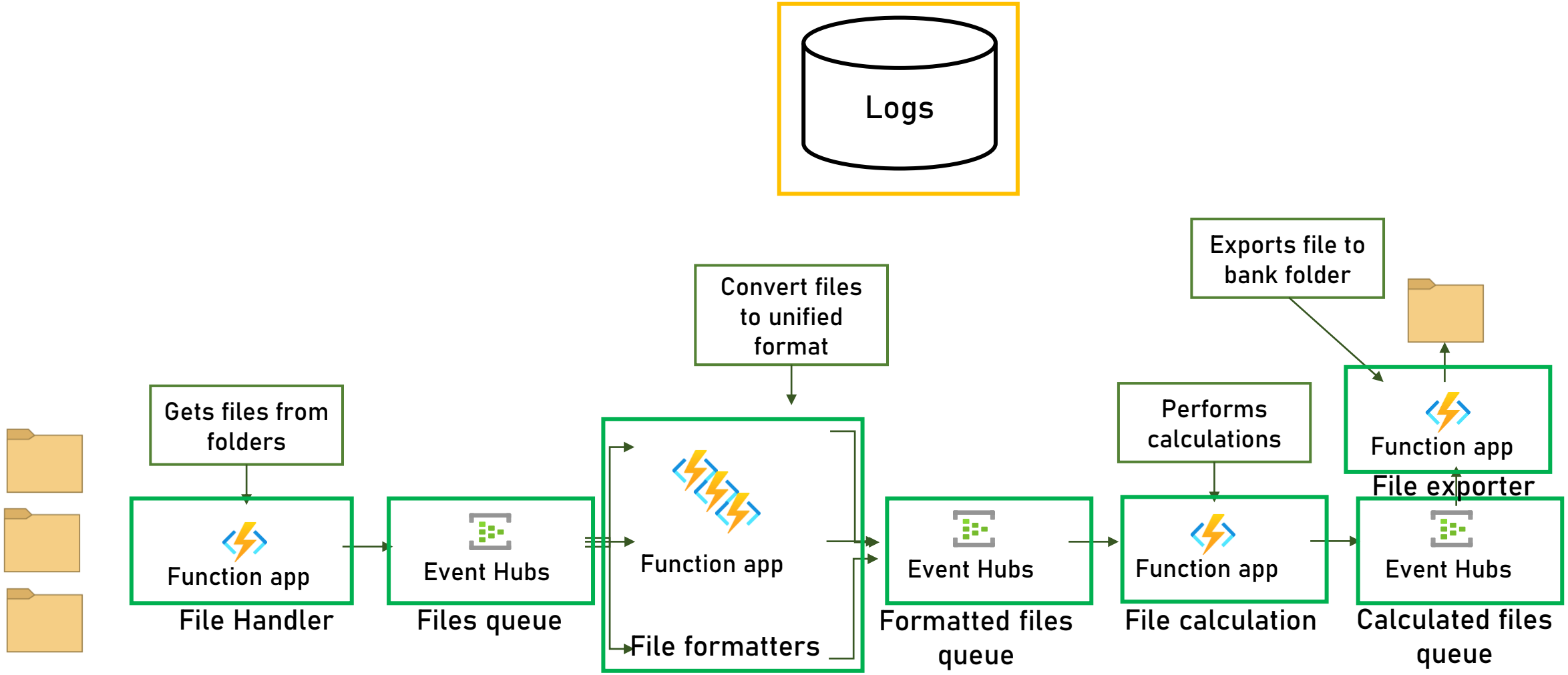
Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
Function app
File calculation
Exports file to bank folder
File Exporter
Event Hubs
Calculated files queue

Components


Logs
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
Function app
File calculation
Exports file to bank folder
Function app
File exporter
Event Hubs
Calculated files queue

Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
Function app
File exporter
Function app
Function app
File Handler
Event Hubs
Files queue
Function app
File formatters
Event Hubs
Formatted files queue
Function app
File calculation
Event Hubs
Calculated files queue

Requirements

Functional

What the system should do

1. Receive file to be processed
2. Validate and process the file
3. Work with various file formats
4. Performs various calculations on the file
5. Create bank payment file
6. Put the payment file in a designated folder
7. Keep log of all the activity for 7 years

Non-Functional

What the system should deal with

1. 500 files / day
2. No data loss
3. 1 min processing time
4. Activity log for 7 years
5. ~2TB / 7 years

Logging

What we need:

- Write a lot of log records

- Allow easy visualizations and analytics

- Preferably – based on existing platform

Logging in Azure

- Azure log analytics
  - Part of Azure Monitor
  - Great integration with a lot of services
  - Handles huge amounts of data
  - Offers query language for analysis
  - Can be streamed to log analytics tools (Power BI etc.)

Azure Monitor

REGION:

West Europe

Log Analytics $44.85

Daily log data ingested will depend on what you are monitoring with Log Analytics. Learn more about estimating data volumes.

Data Ingestion

Daily logs ingested (GB/day)	×	Days	×	Per GB	=	
0.25	×	30	×	$2.99	=	$22.43

This estimate is calculated using the most optimal pricing tier for the data ingestion. This calculation uses **Pay-As-You-Go tier**. Learn more about the pricing tiers

Data Retention

The first month of retention is free

24
Total retention (Months)

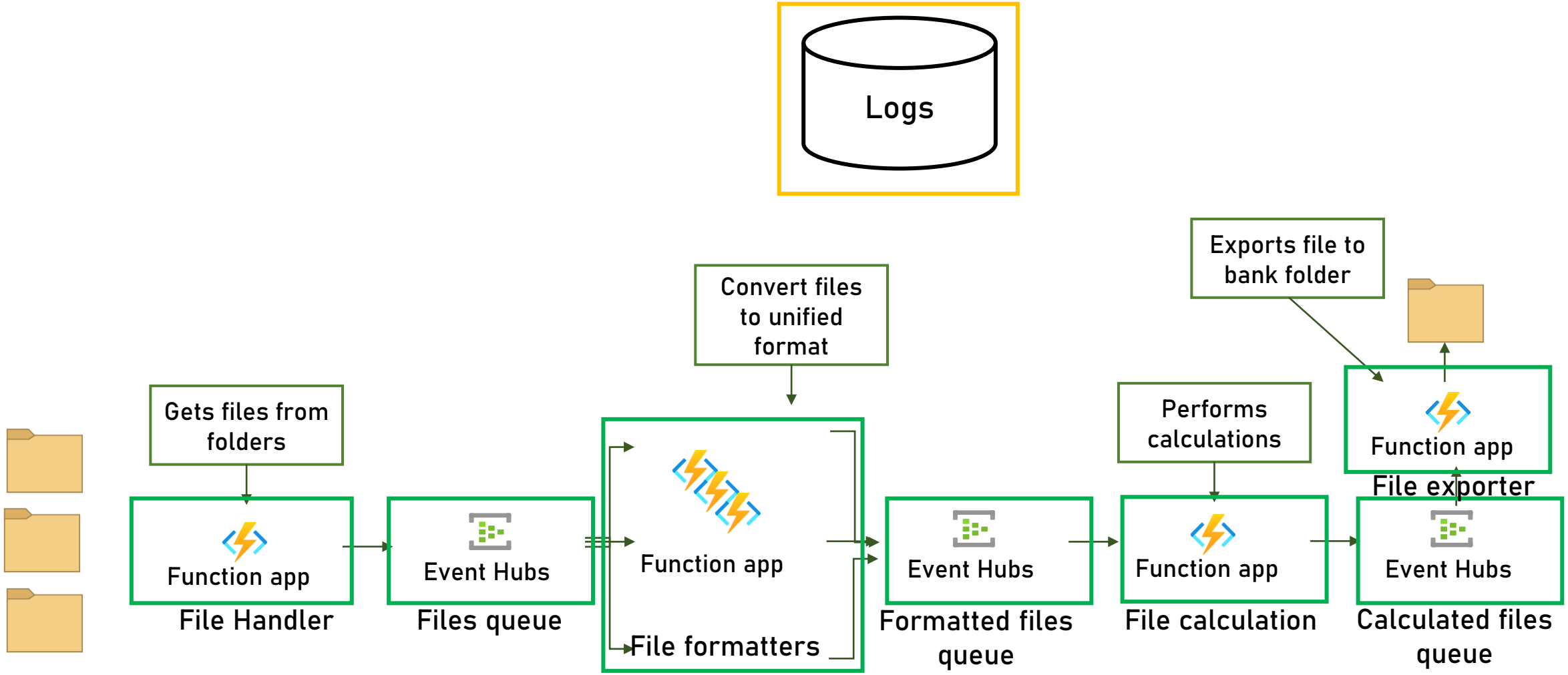
Max retention is 2 years. Stream to storage or re-discuss NFR

Total monthly ingestion (GB)	×	Additional retention (months)	×	Per GB/month	=	
7.5	×	23	×	$0.13	=	$22.43

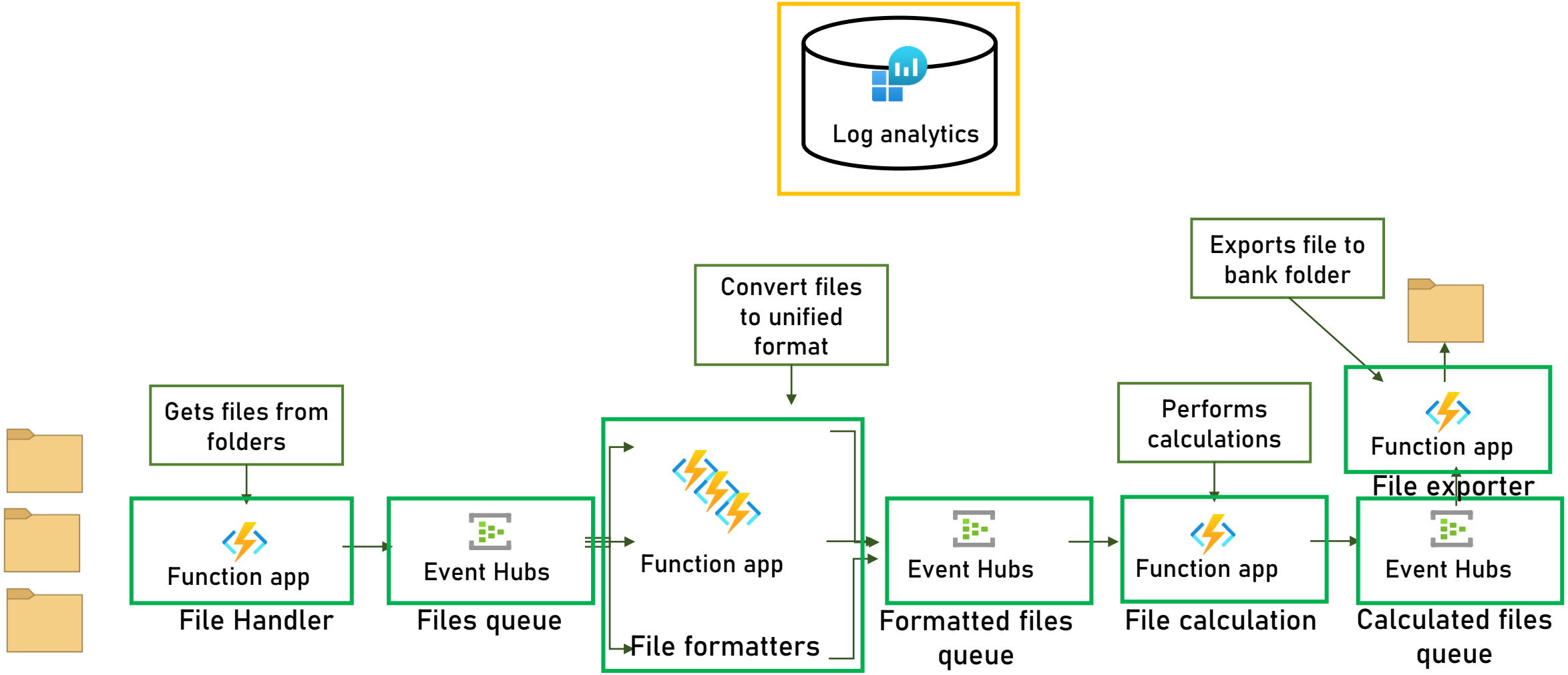
Application Insights	$0.00
Alert Rules	$1.60
ITSM Connector - Ticket Creation/Update	$0.00
Notifications	$0.00

Upfront cost	$0.00
Monthly cost	$46.45

Components


Logs
Exports file to bank folder
Convert files to unified format
Gets files from folders
Performs calculations
Function app
File exporter
Function app
Function app
File Handler
Event Hubs
Files queue
Function app
File formatters
Event Hubs
Formatted files queue
Function app
File calculation
Event Hubs
Calculated files queue

Components


Log analytics
Gets files from folders
Function app
File Handler
Event Hubs
Files queue
Convert files to unified format
Function app
File formatters
Event Hubs
Formatted files queue
Performs calculations
Function app
File calculation
Exports file to bank folder
Function app
File exporter
Event Hubs
Calculated files queue

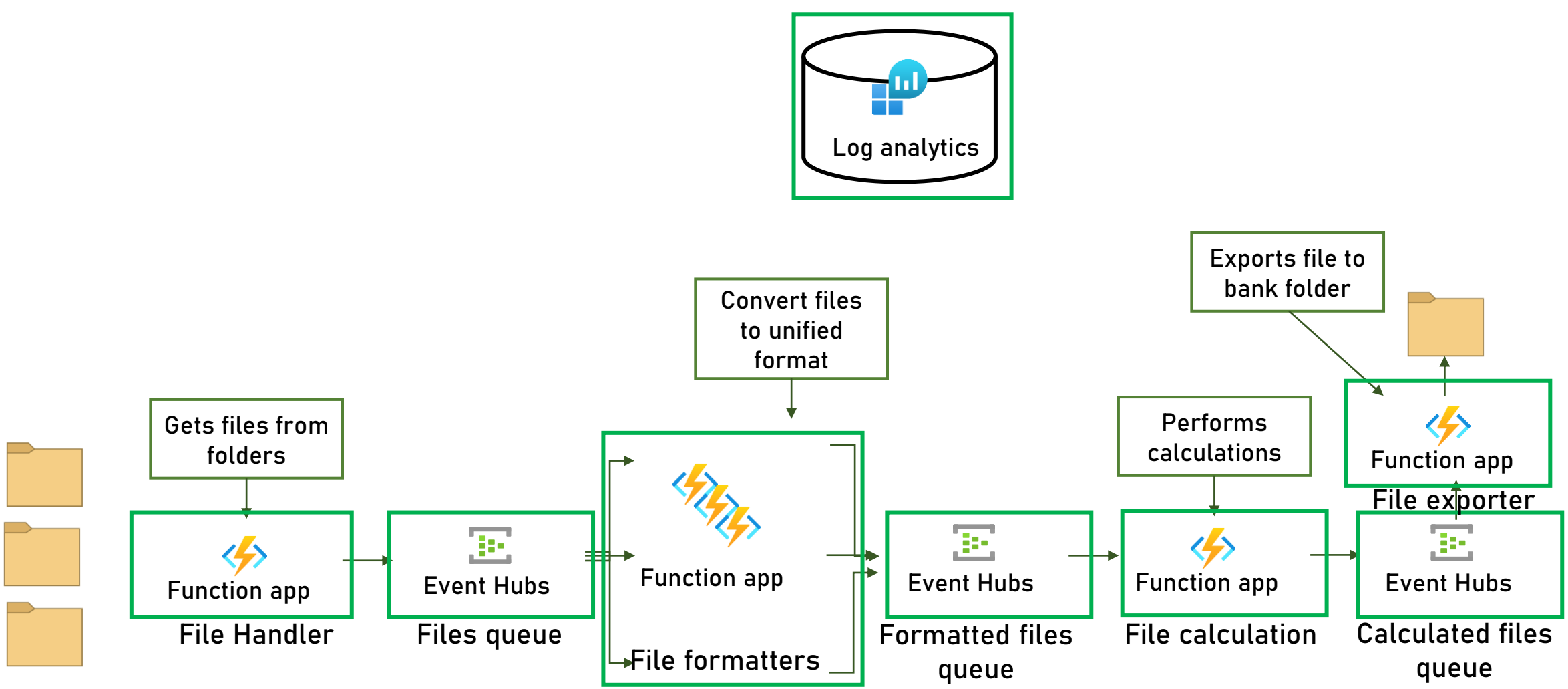
Security

- System is internal only so no substantial security risks expected
- Pay attention to:
  - Data encryption
  - Data validation

Security

- To-Do:
  - Make sure database encryption is on (should be the default)
  - Make sure the file handler performs validation on the files

Architecture Diagram



Cost

Estimated upfront cost	$0.00
Estimated monthly cost	$68.35

Download detailed cost estimation from the lecture's resources