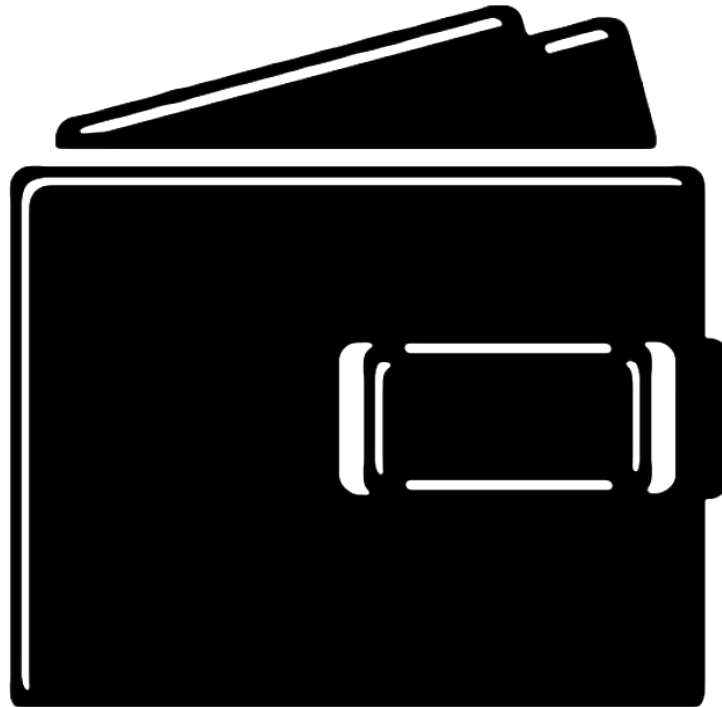


# Case Study #4

Memilavi  
[www.memilavi.com](http://www.memilavi.com)

<https://t.me/learningnets>





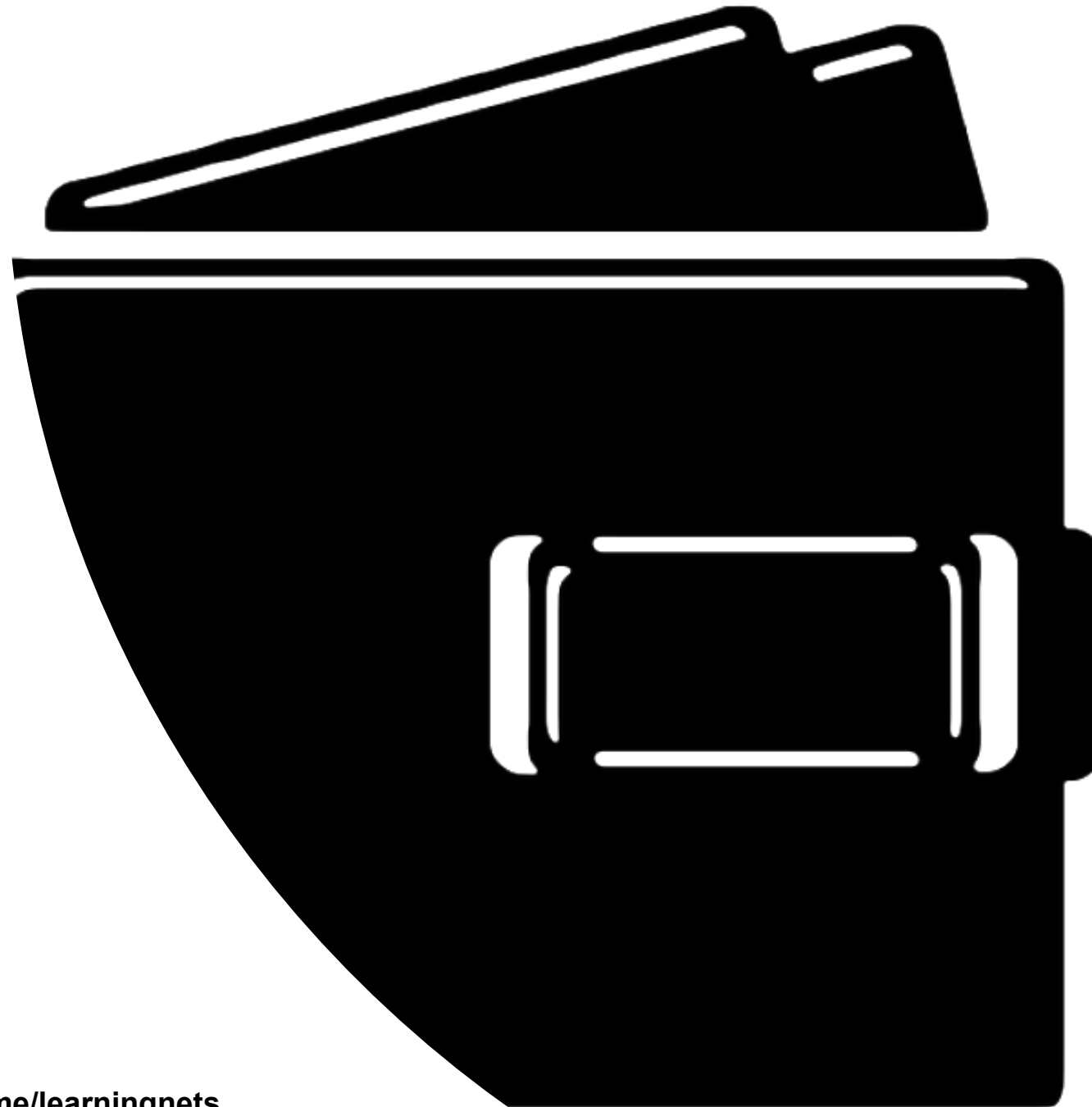
**PayRawl**

<https://t.me/learningnets>

# 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

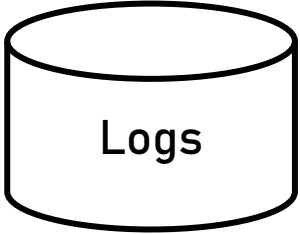
What the system should deal with

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



Convert files to unified format

Exports file to bank folder



Gets files from folders

File Handler

Files Queue

File Formatter

File Formatter

File Formatter

Formatted Files Queue

Performs calculations

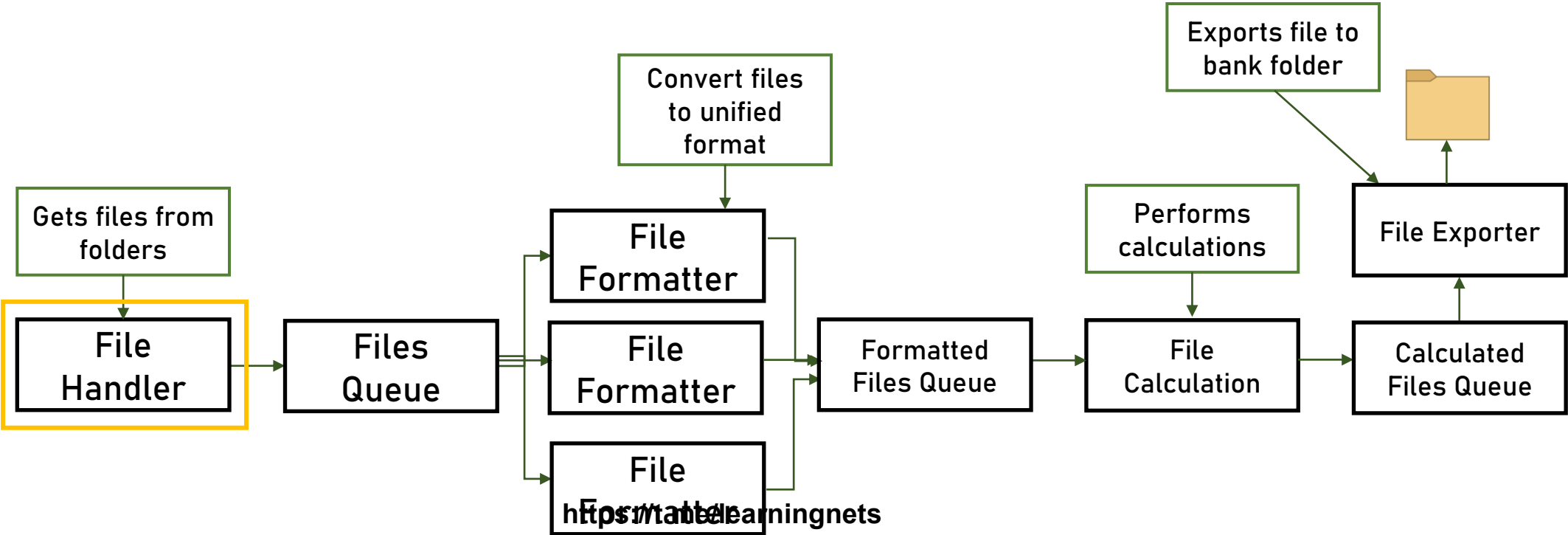
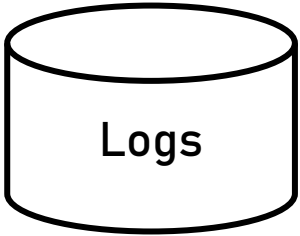
File Calculation

File Exporter

Calculated Files Queue



# Components










# 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

×

15000

Execution time (in  
milliseconds)

Executions per month

= \$0.00

### Requests

15,000

Execution count

= \$0.00

Upfront cost

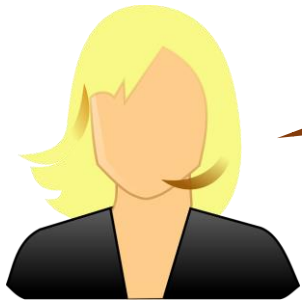
\$0.00

Monthly cost

\$0.00



# Technology Stack



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



<https://t.me/learningnets>



# 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

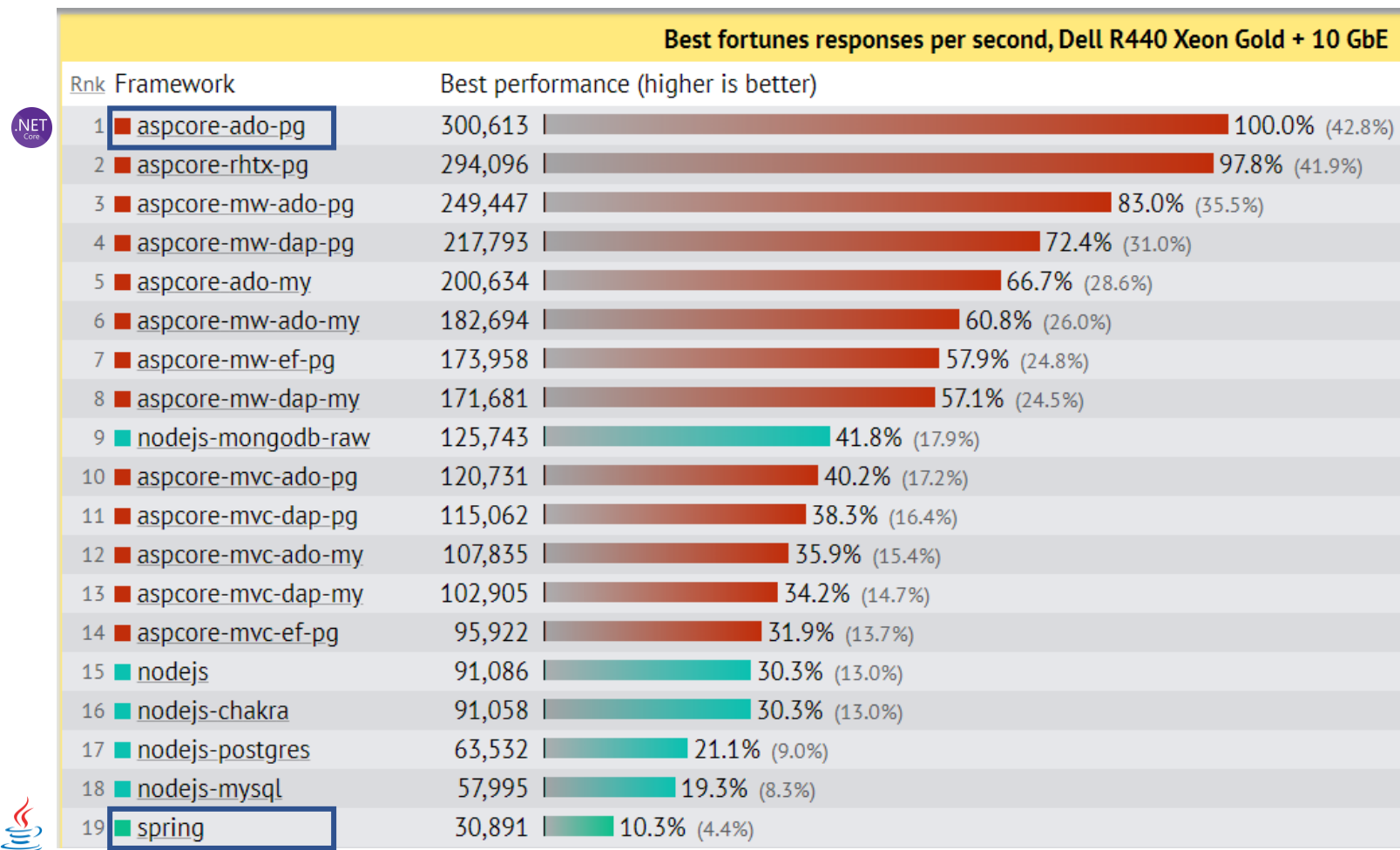
Node is mainly for web apps, our component is a service, so...





# Technology Stack

## Performance:





# 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	▼	Visual Basic .NET	5.287%	-1.17%
7	6	▼	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%



# Technology Stack

Cross Platform:





# Technology Stack

Ease to learn and use:



<https://t.me/learningnets>



# Technology Stack

Evolving?



Next versions planned until 2021



Roadmap announced until 2023



<https://t.me/learningnets>



# Technology Stack

Threading support:



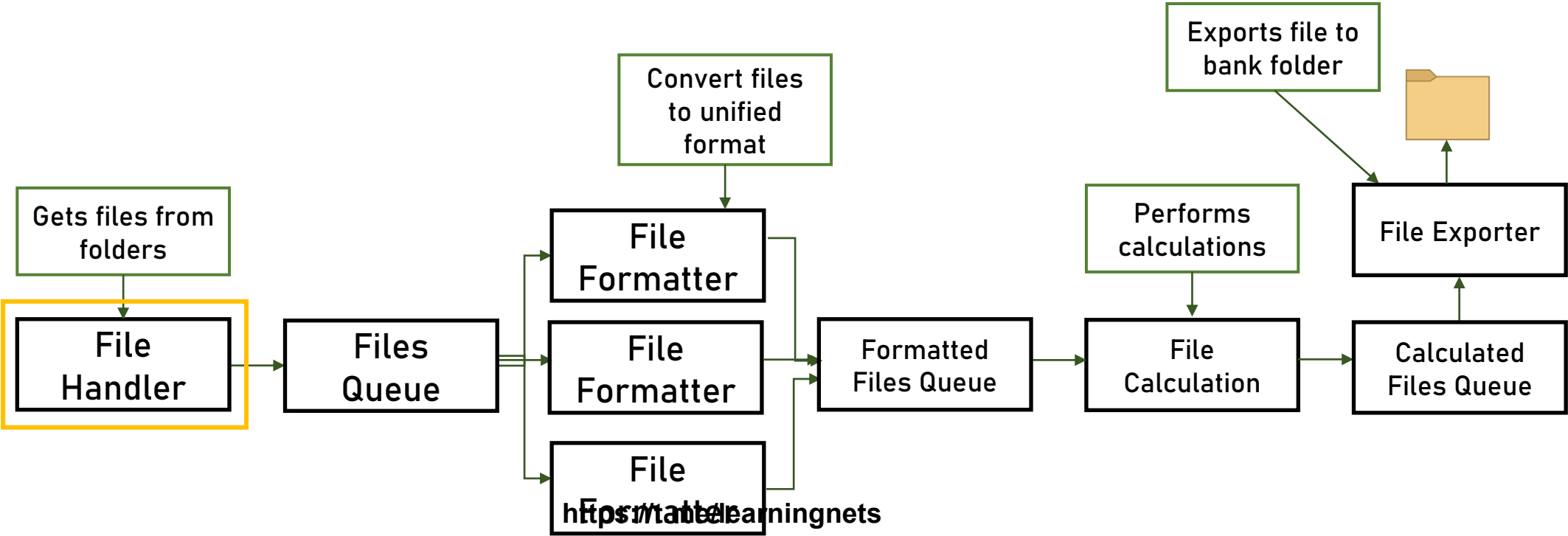
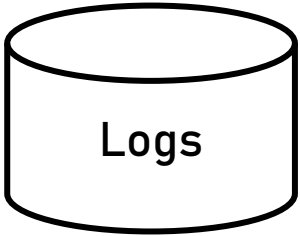


# Technology Stack - Decision



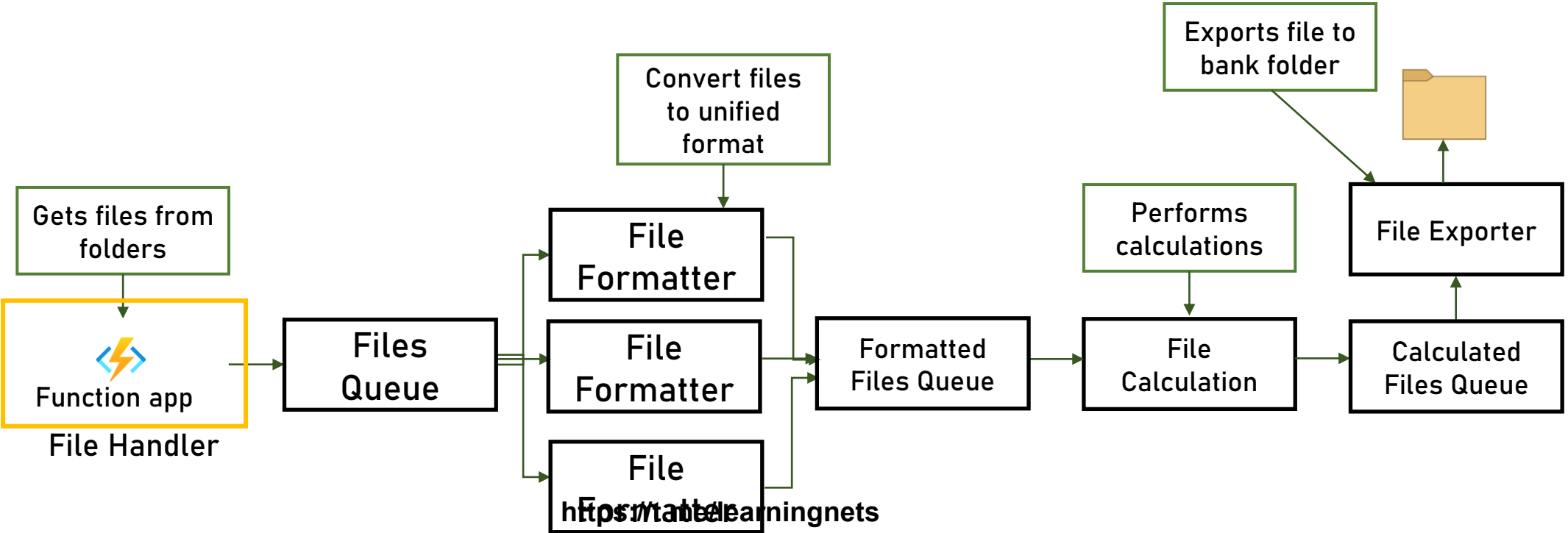
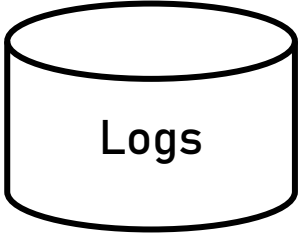


# Components



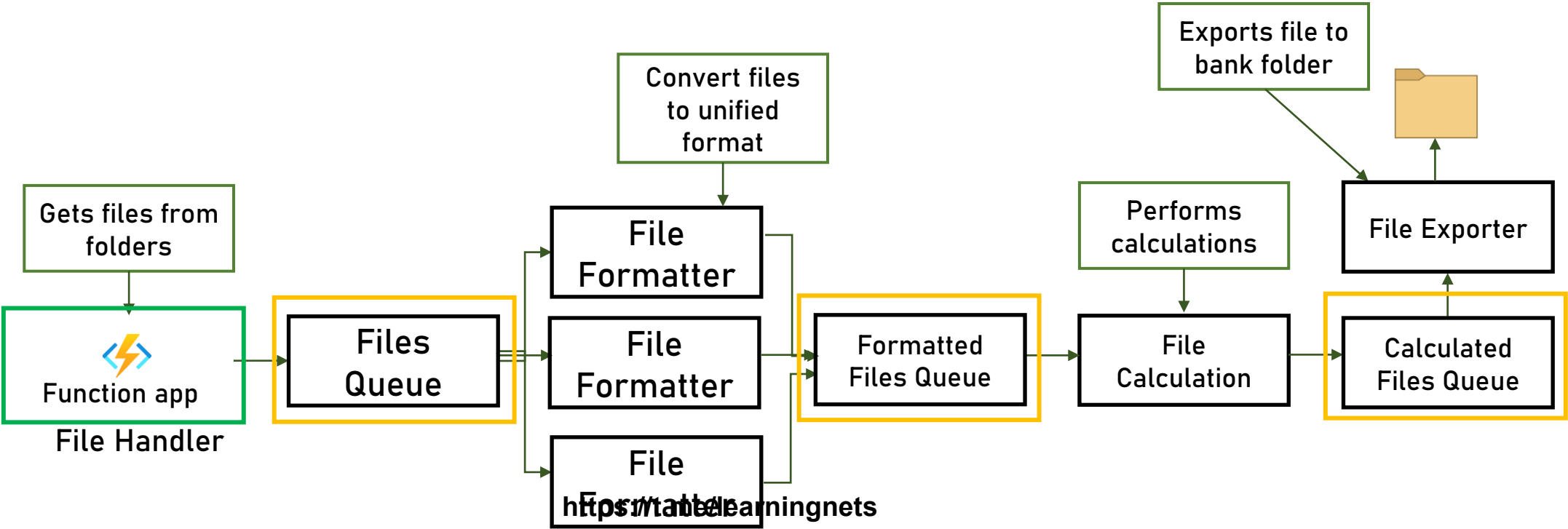
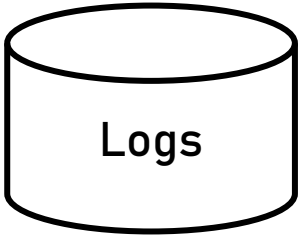


# Components





# Components





## 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



## 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

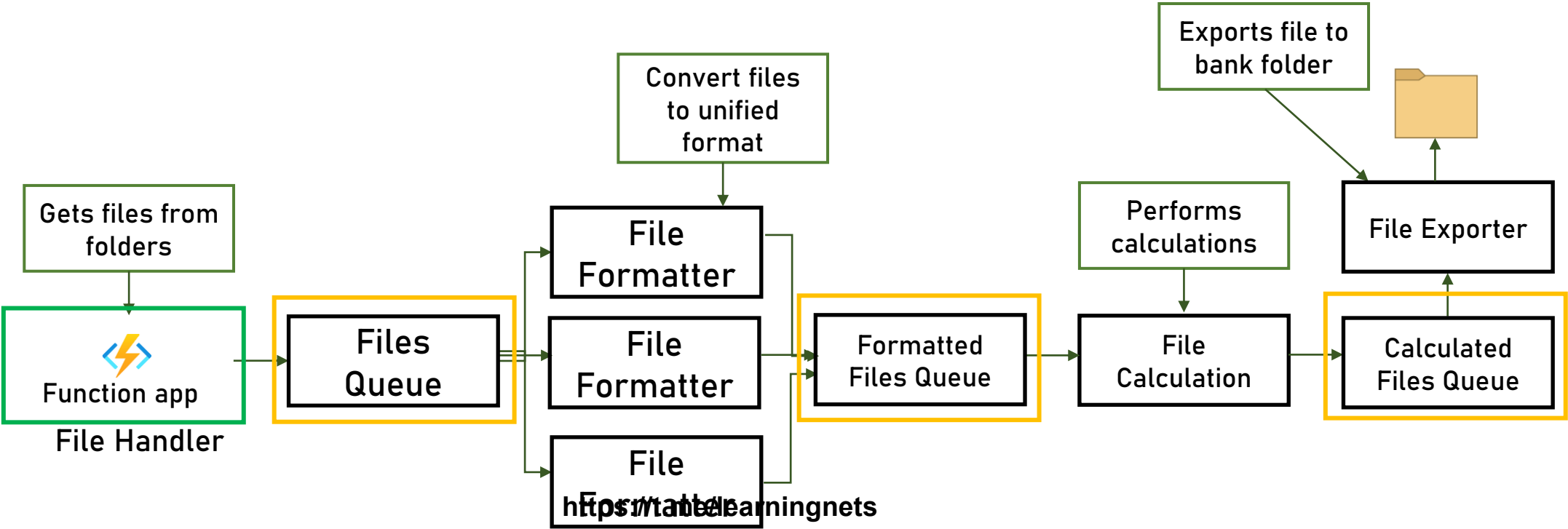
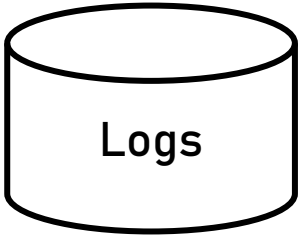
Monthly cost

\$21.90

Each TU  
supports up to  
1k msgs / sec

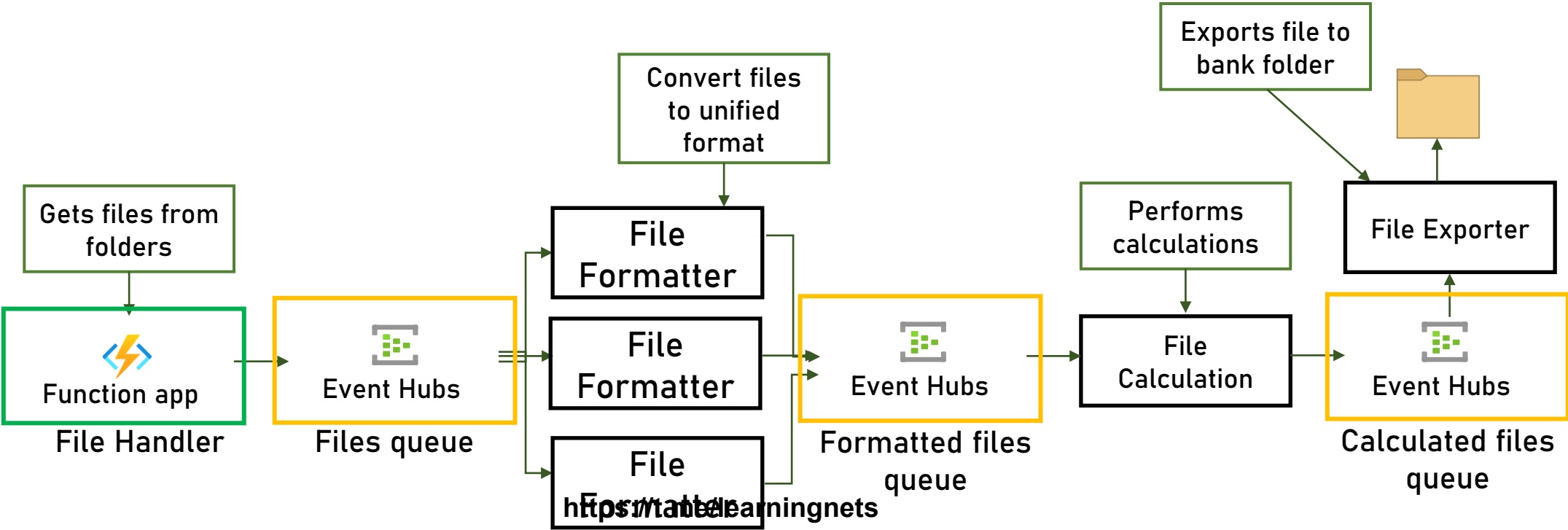
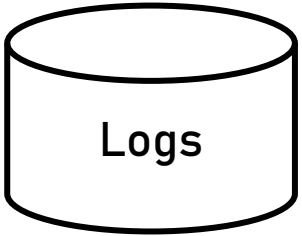


# Components



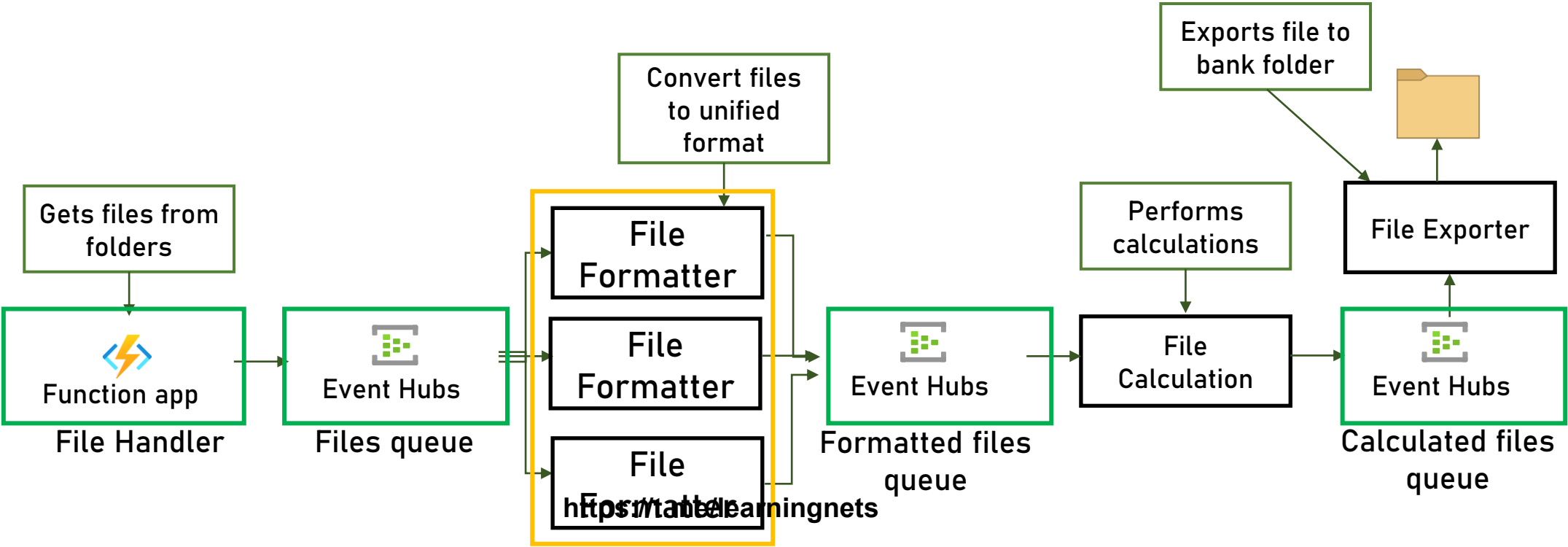
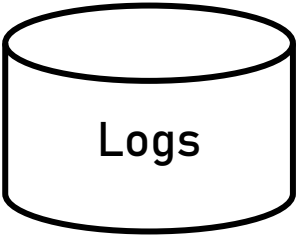


# Components





# Components





# 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

<https://t.me/learningnets>



## 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

×

15000

Execution time (in  
milliseconds)

Executions per month

= \$0.00

### Requests

15,000

Execution count

= \$0.00

Upfront cost

\$0.00

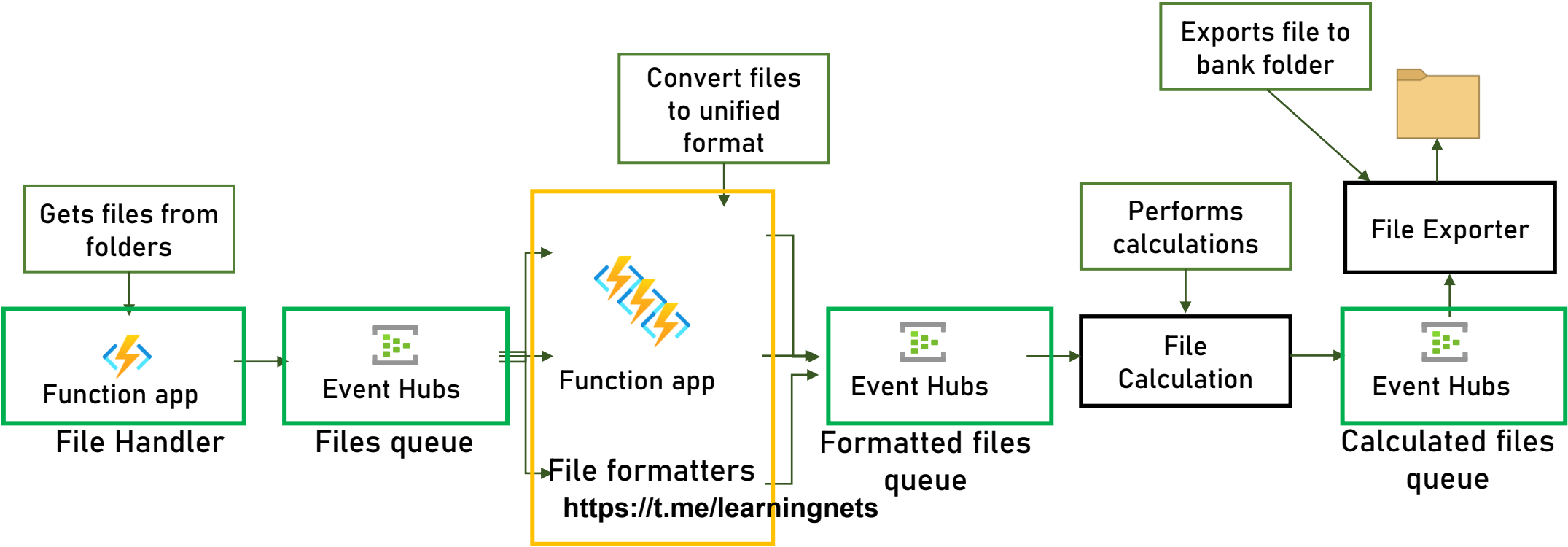
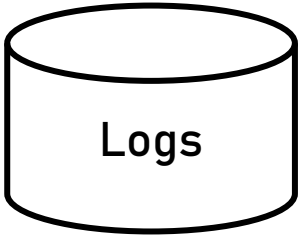
Monthly cost

\$0.00



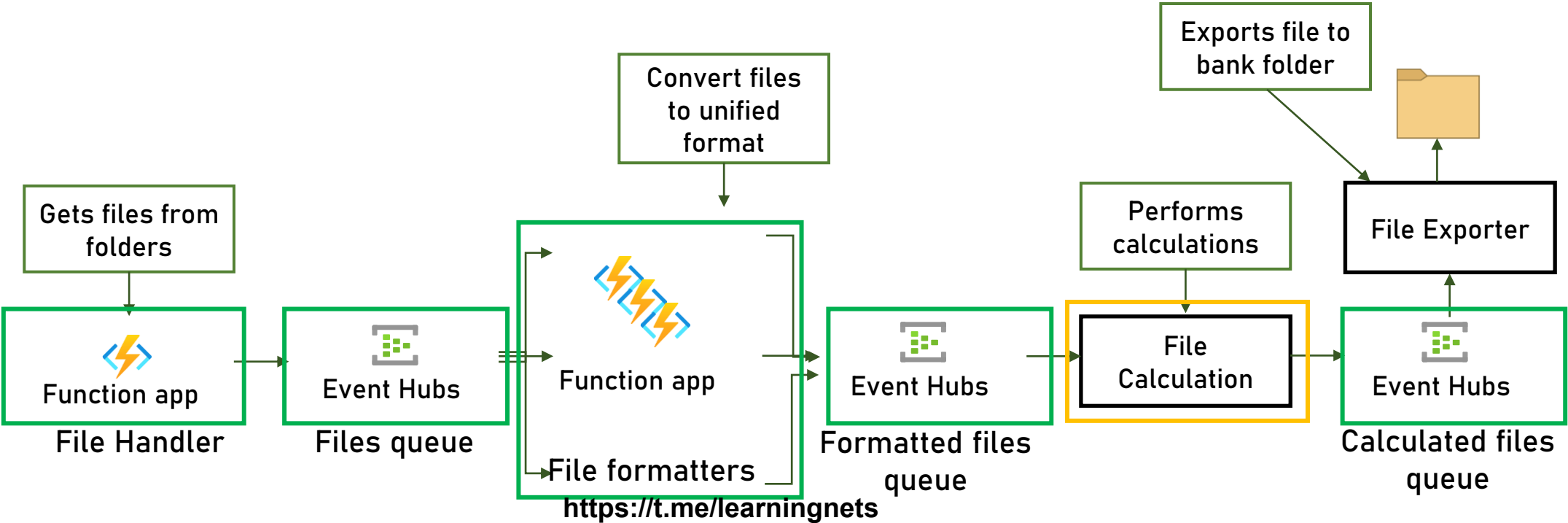
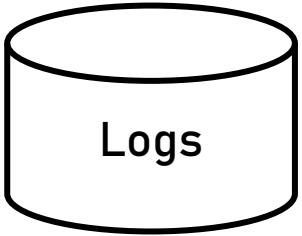


# Components





# Components





## 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

×

15000

Execution time (in  
milliseconds)

Executions per month

= \$0.00

### Requests

15,000

Execution count

= \$0.00

Upfront cost

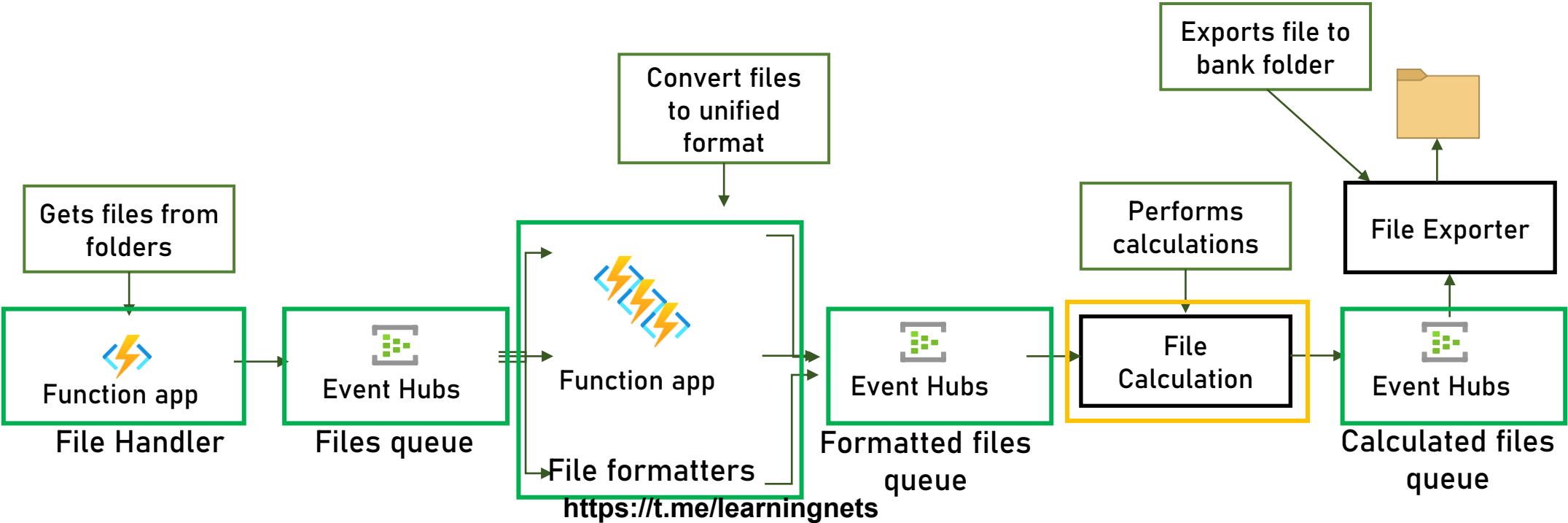
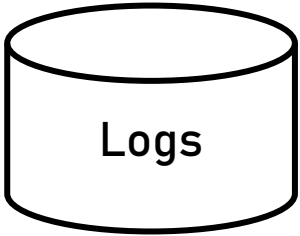
\$0.00

Monthly cost

\$0.00

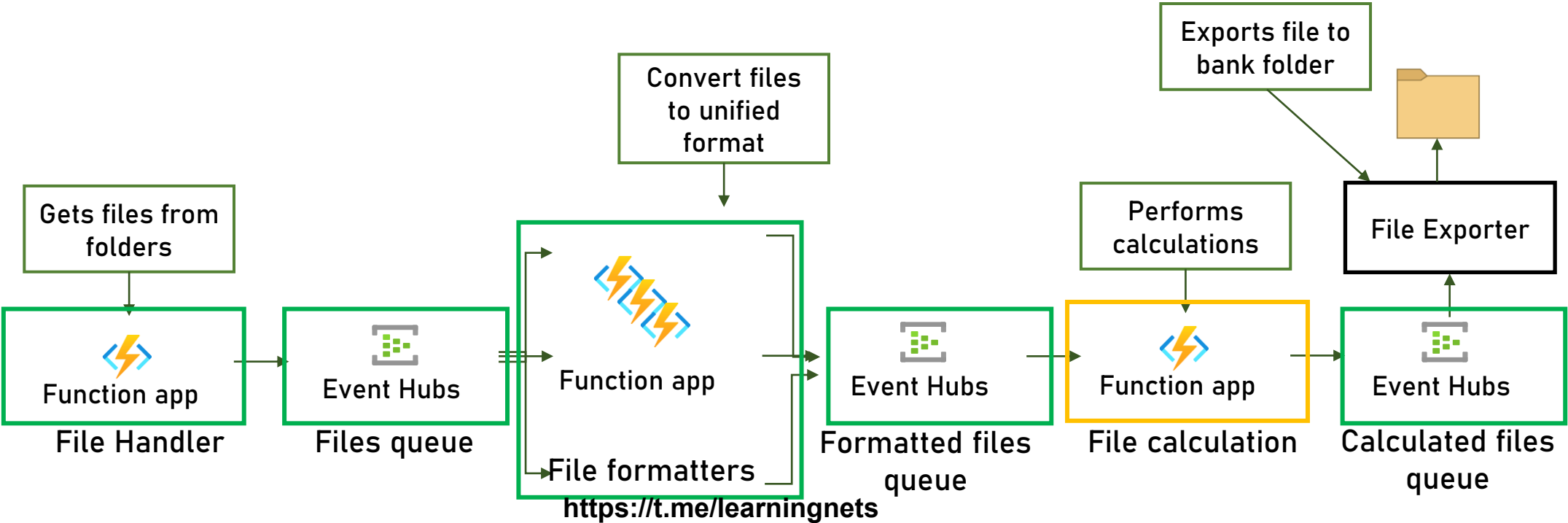
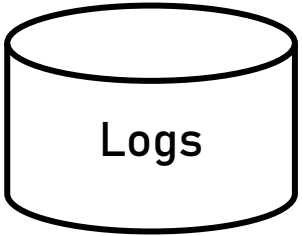


# Components



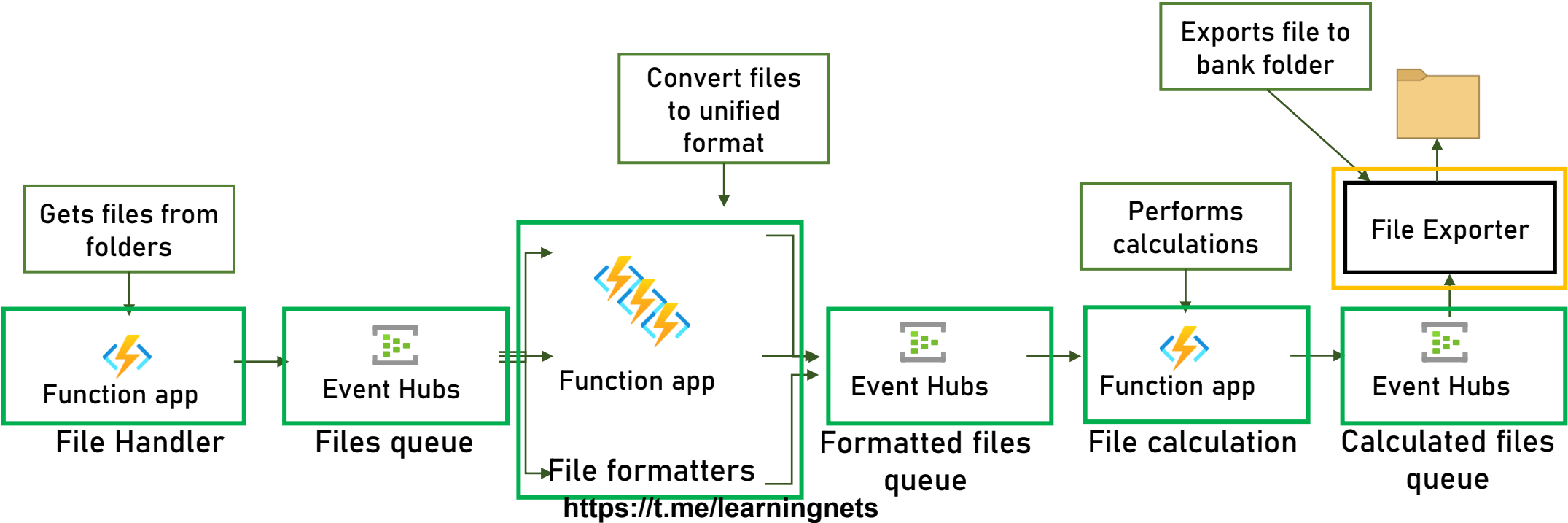
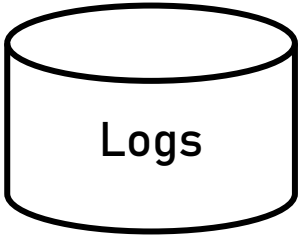


# Components





# Components





## 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

×

15000

Execution time (in  
milliseconds)

Executions per month

= \$0.00

### Requests

15,000

Execution count

= \$0.00

Upfront cost

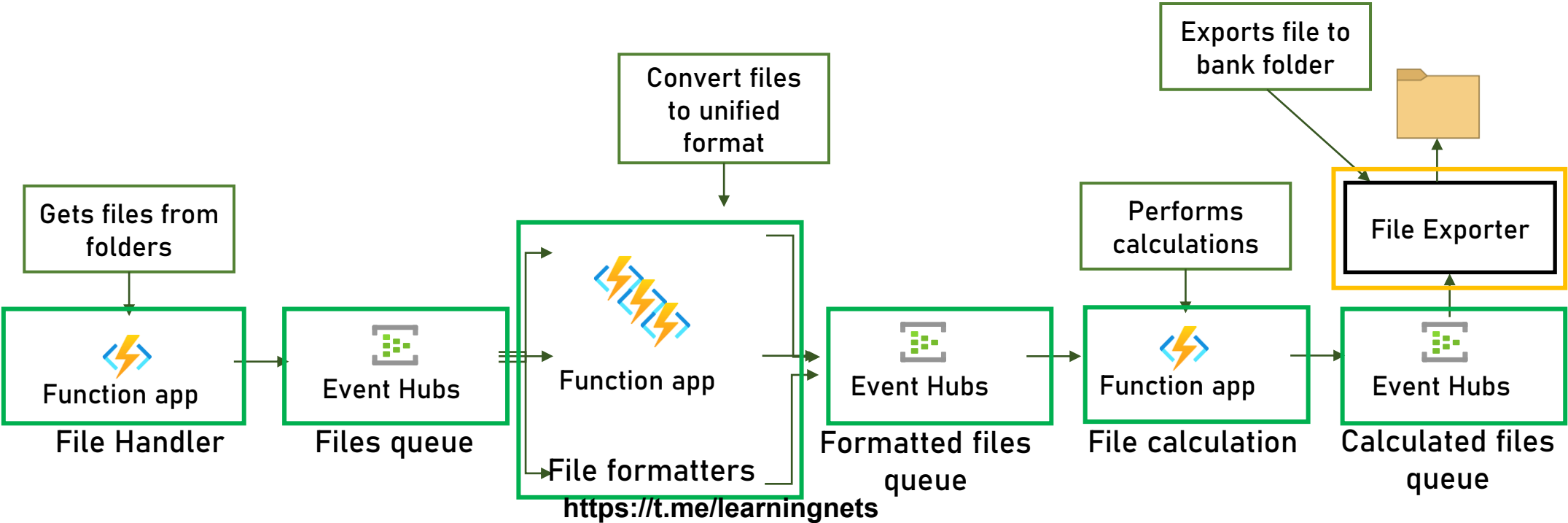
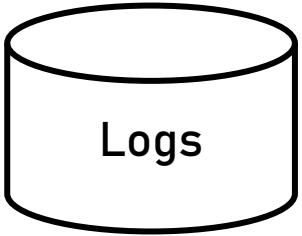
\$0.00

Monthly cost

\$0.00

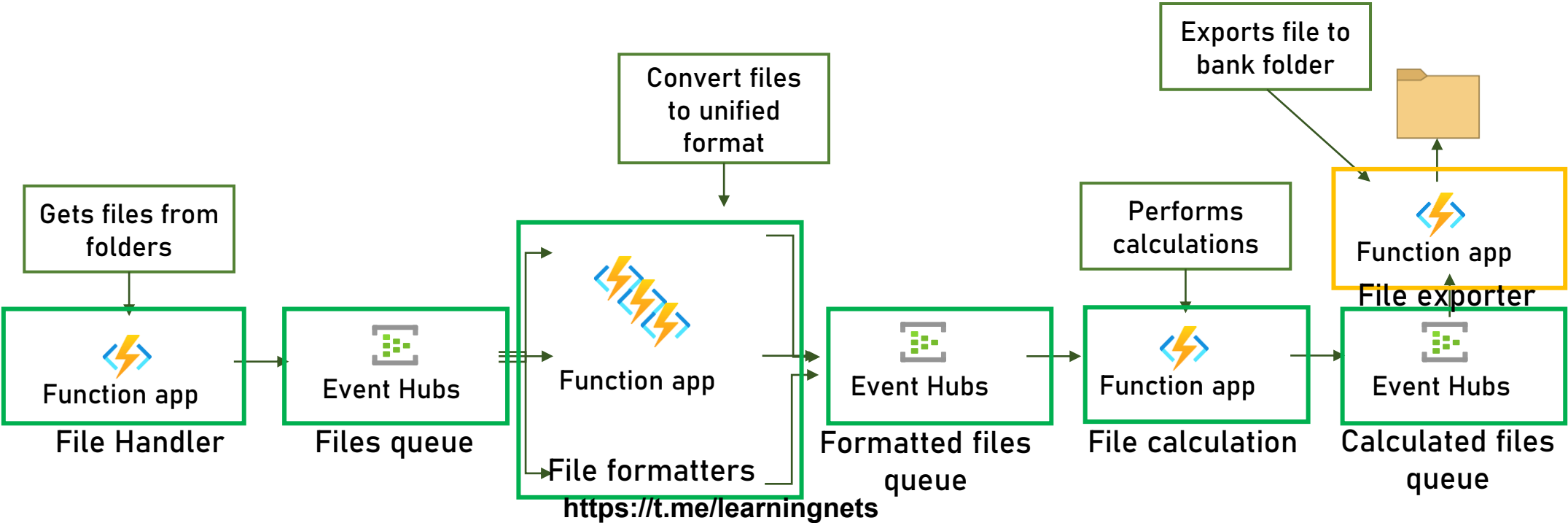
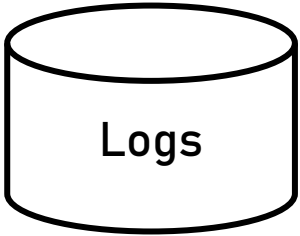


# Components



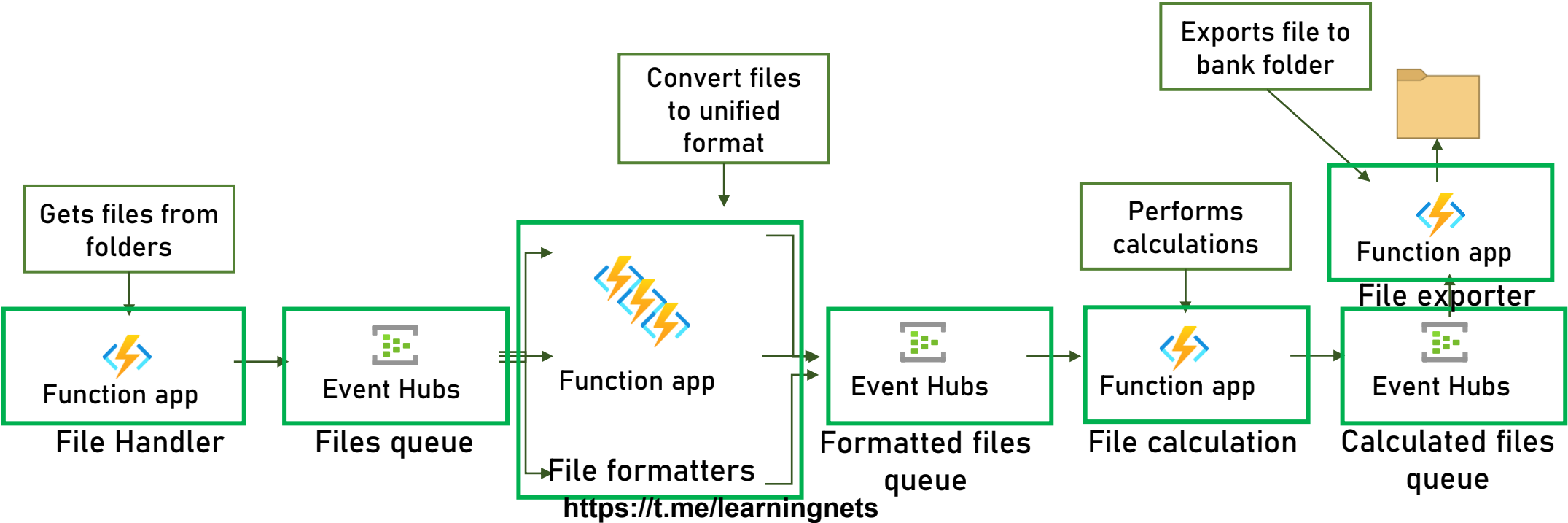
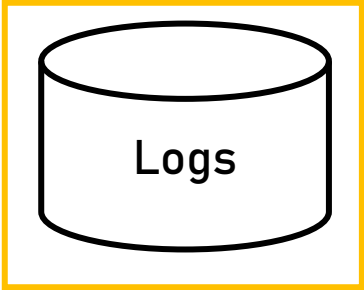


# Components





# Components





# 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

0.25  
Daily logs ingested (GB/day)

×

30

×

\$2.99

Per GB

=

\$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

7.5

Total monthly ingestion (GB)

×

23

Additional retention (months)

×

\$0.13

Per GB/month

=

\$22.43

Application Insights

\$0.00

Alert Rules

\$1.60

ITSM Connector - Ticket Creation/Update

\$0.00

Notifications

\$0.00

Upfront cost

\$0.00

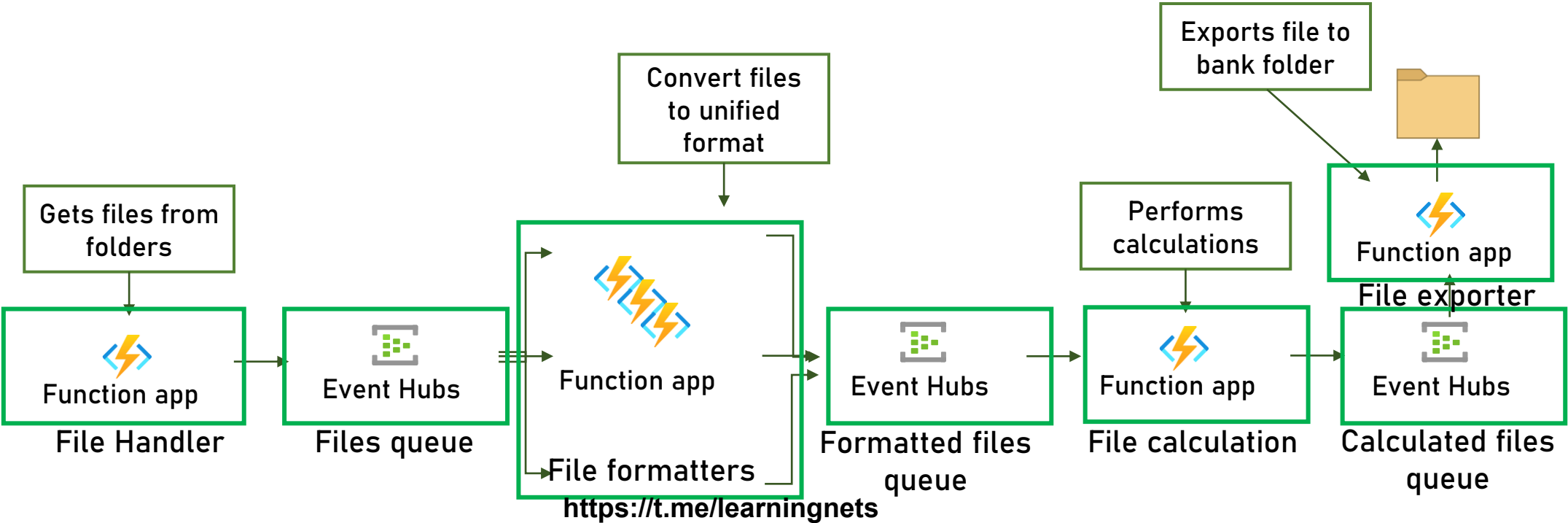
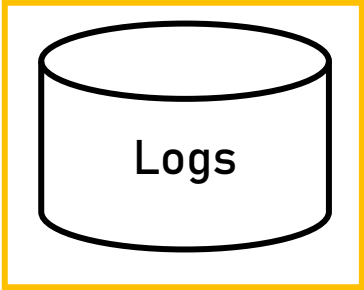
Monthly cost

\$46.45

<https://t.me/learningnets>

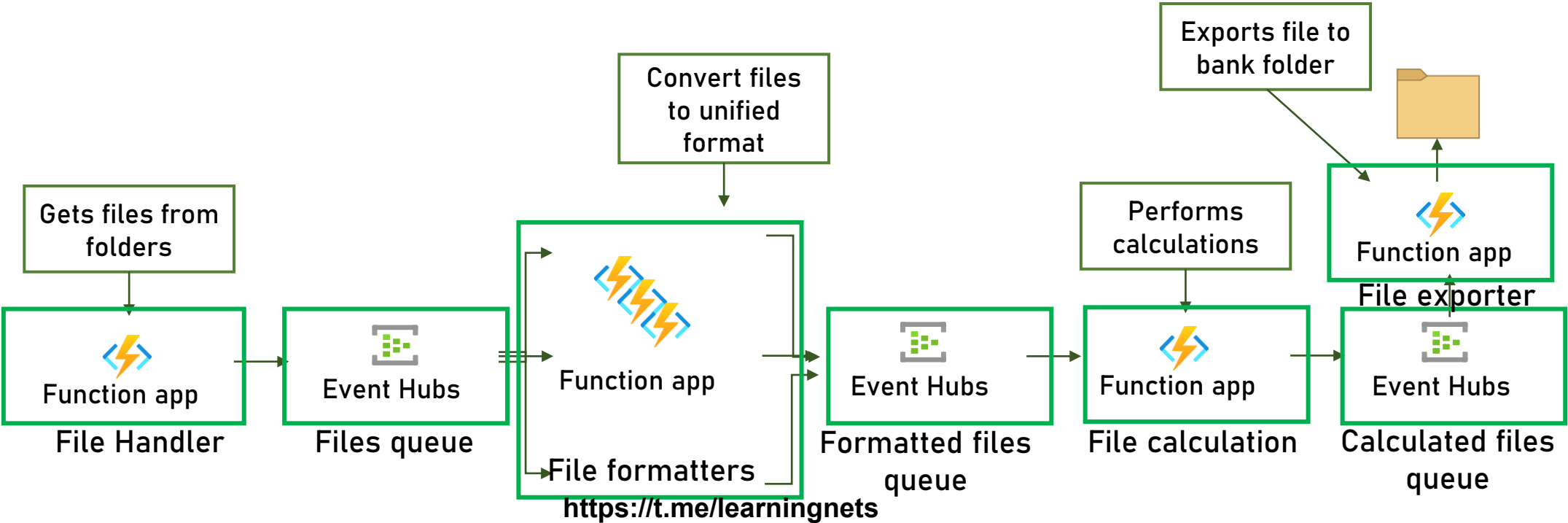
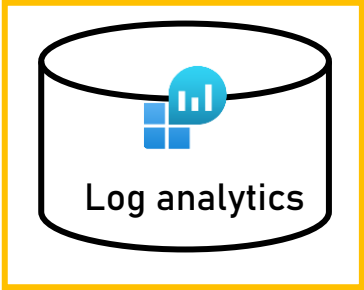


# Components





# Components





# Security

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

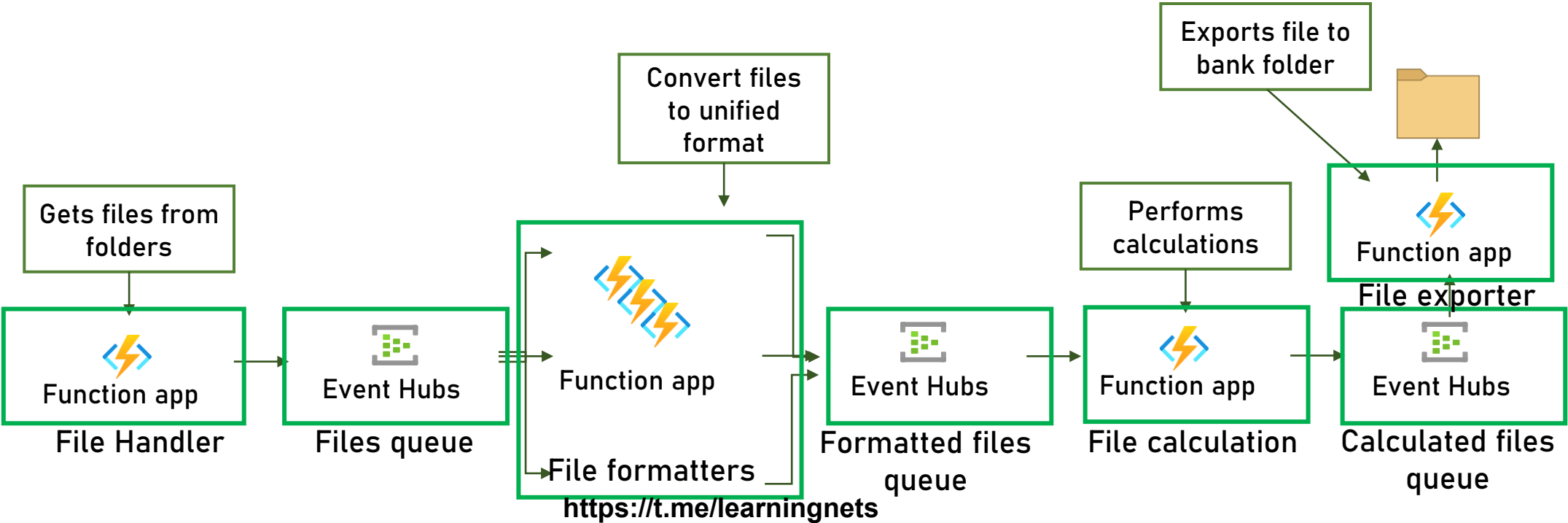
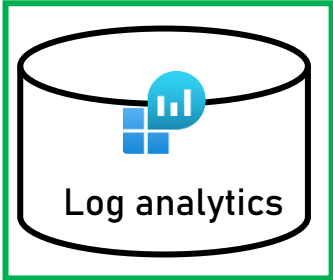


# Security

- **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

<https://t.me/learningnets>