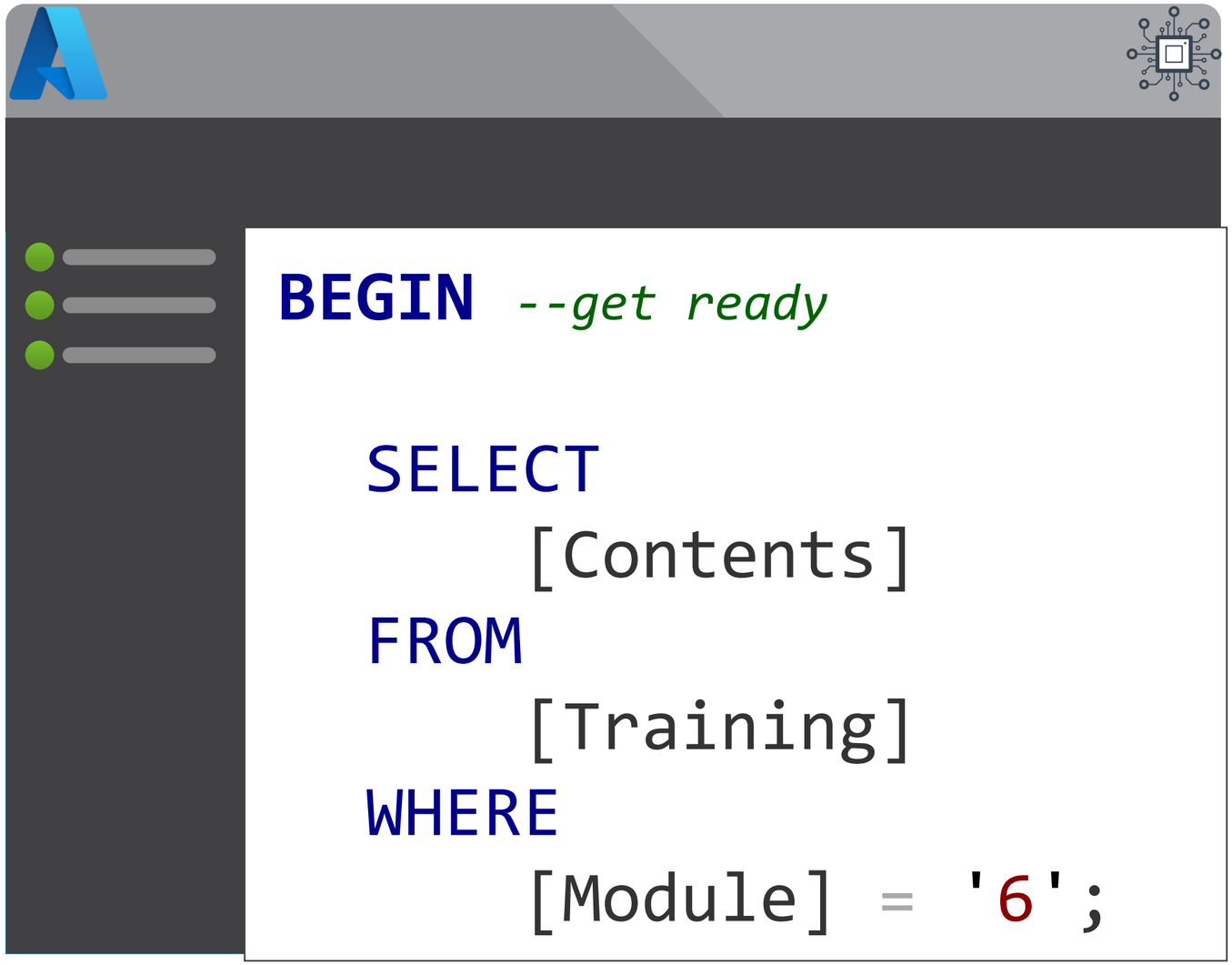
Module 6

Execution Parallelism



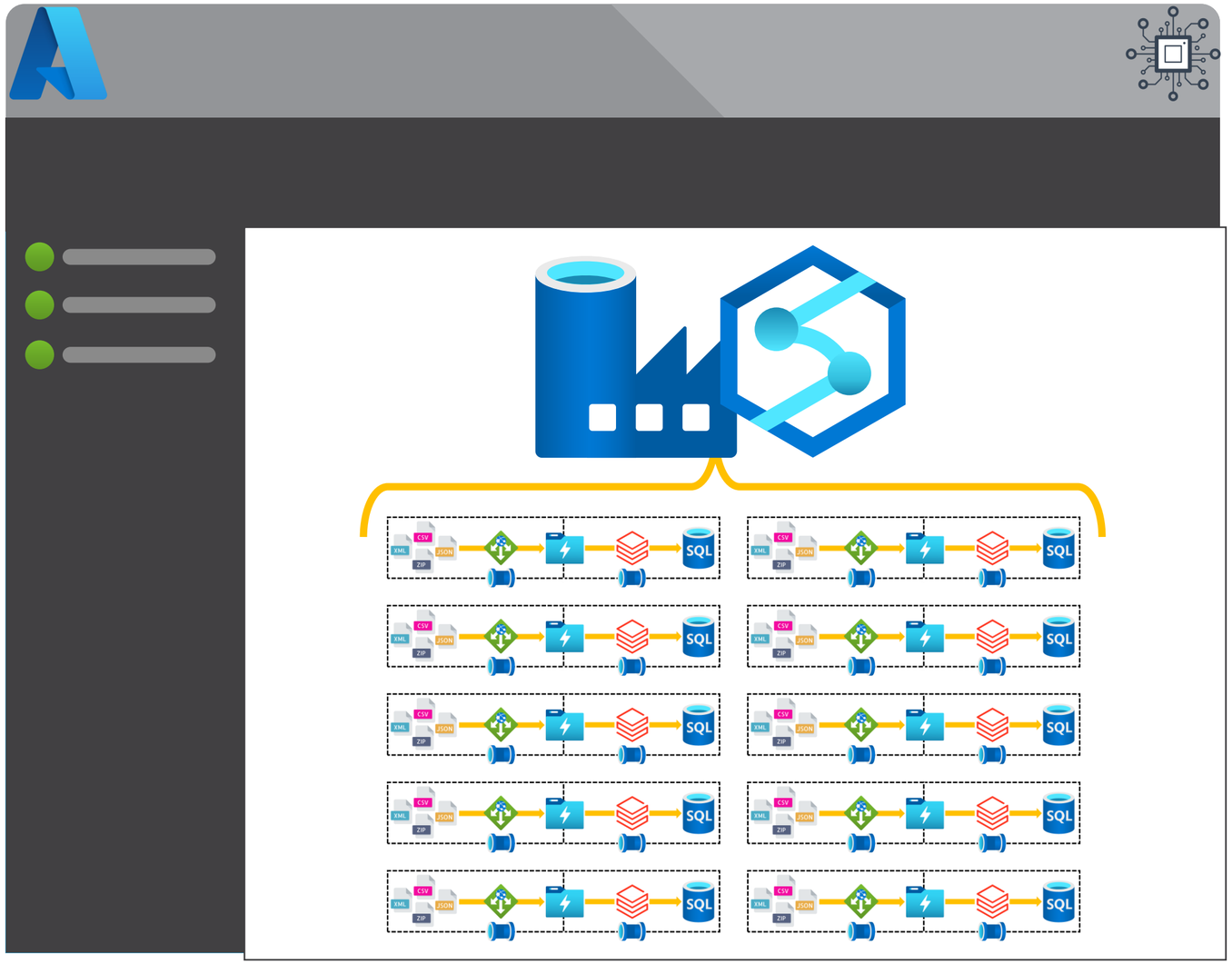
```
BEGIN --get ready

    SELECT
        [Contents]
    FROM
        [Training]
    WHERE
        [Module] = '6';
```

- Control Flow Scale Out
- Concurrency Limitations
- Internal vs External Activities
- Orchestration Framework - http://procfwk.com

Module 6

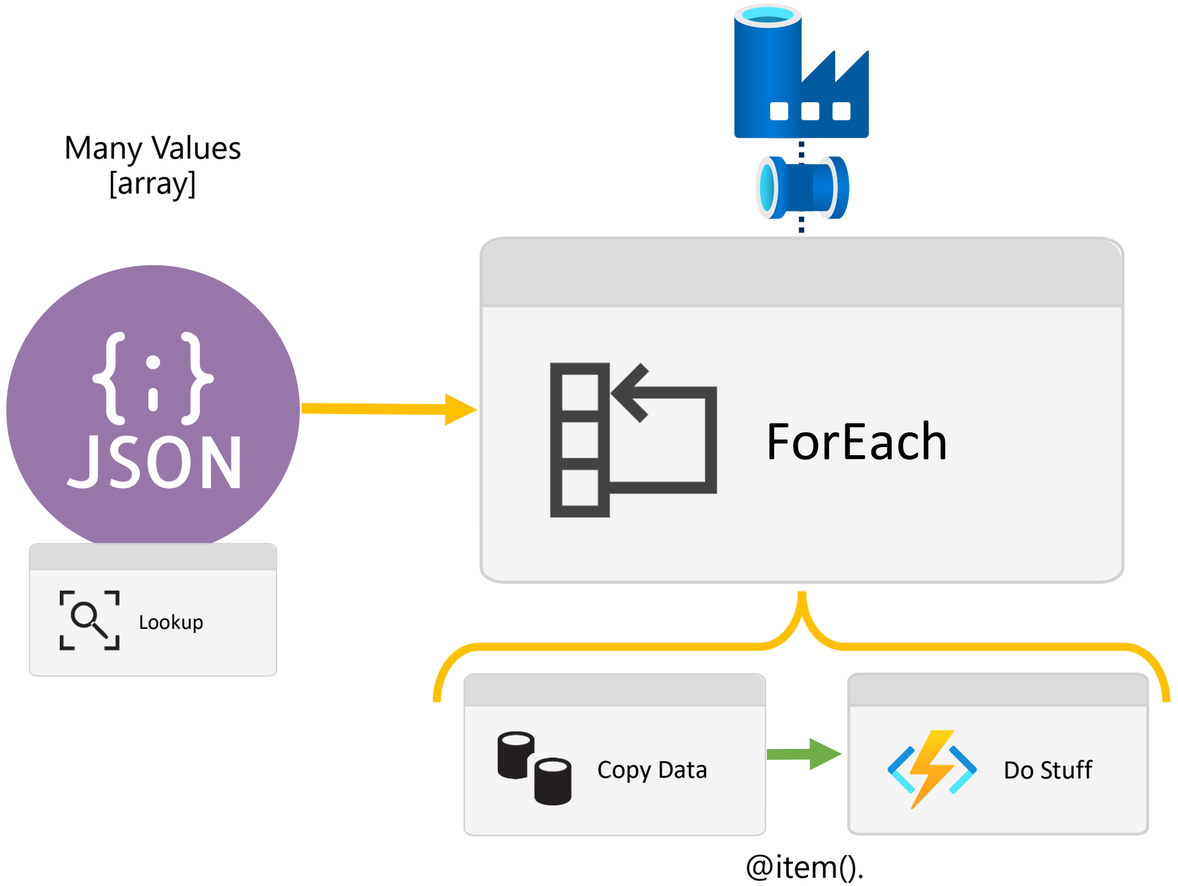
Execution Parallelism



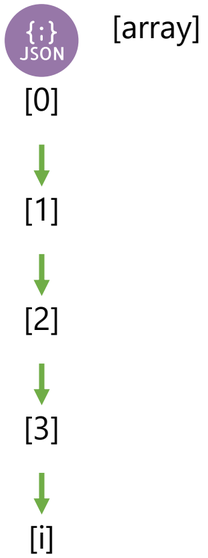
- Control Flow Scale Out
- Concurrency Limitations
- Internal vs External Activities
- Orchestration Framework - http://procfwk.com

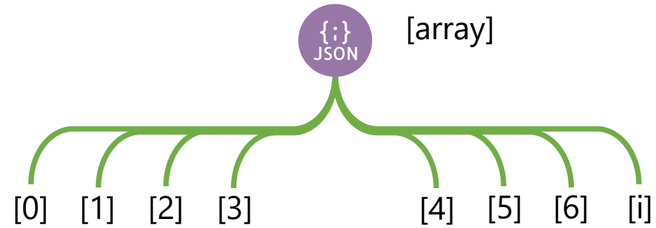
For Each Activity

Scaling Out Control Flow Activities



IsSequential: true



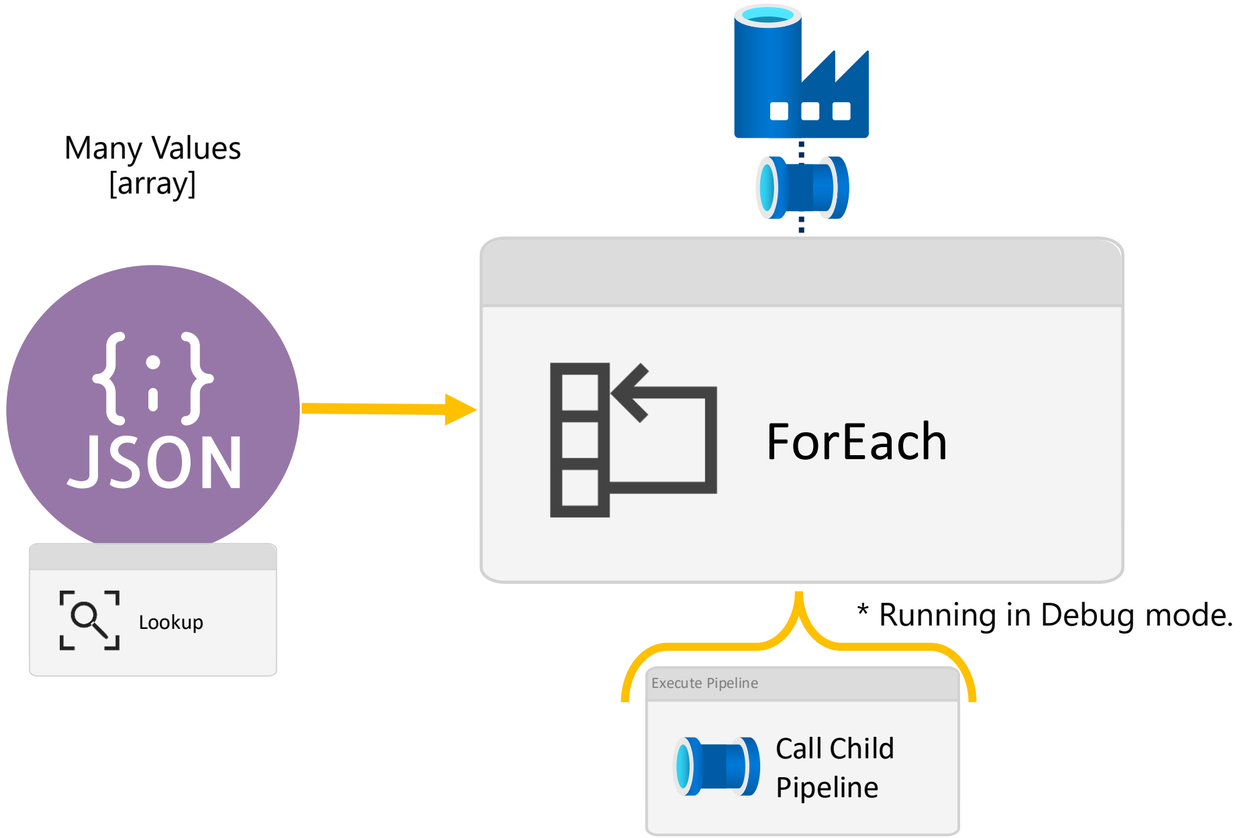


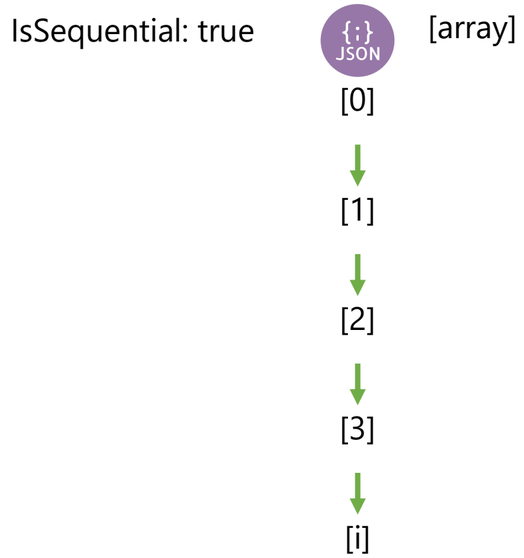
Batch Count Default: 20

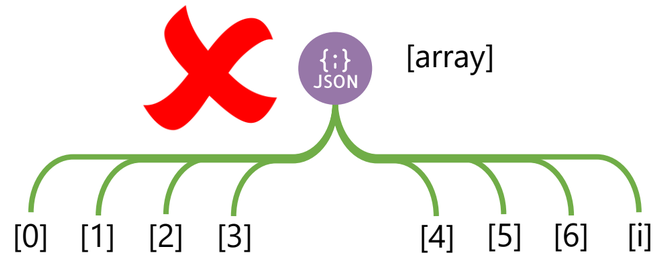
Batch Count Max: 50

For Each Activity

Scaling Out Control Flow Activities





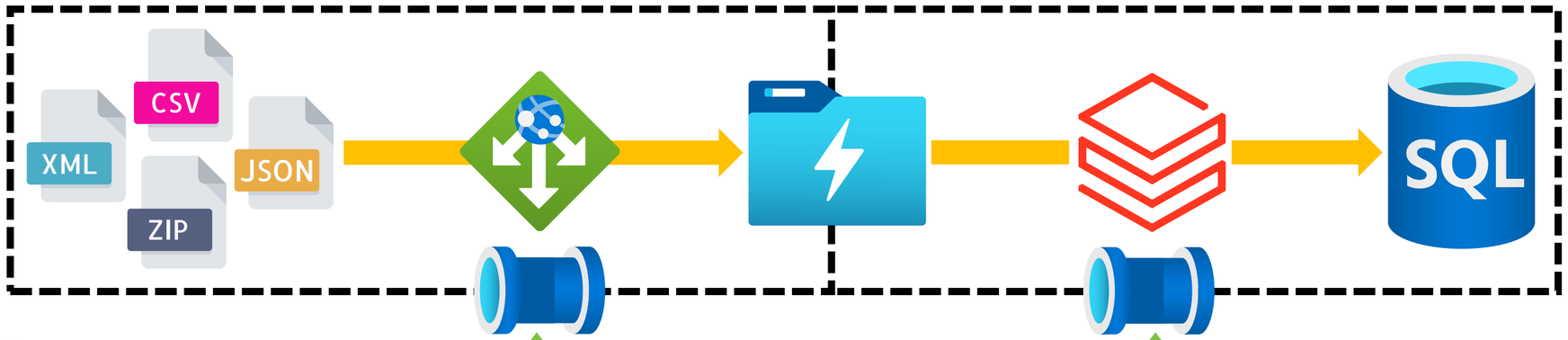


Batch Count Default: 20

Batch Count Max: 50

Integration Pipelines as Data Engineers

Control Flow



1 Linked Services

2 Datasets

3 Activities

4 Pipelines

5 Triggers

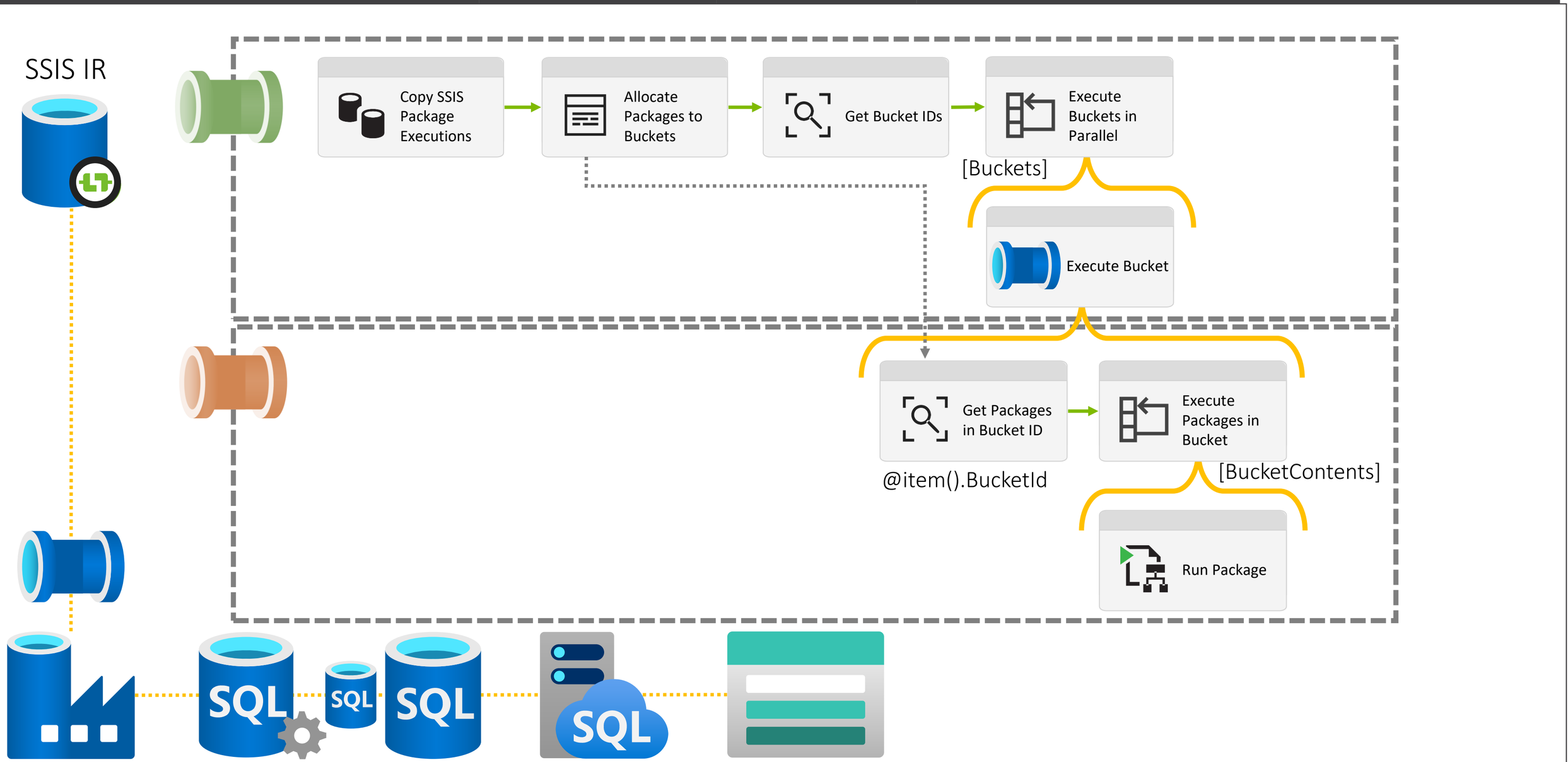
Integration Runtimes 6

Azure IR

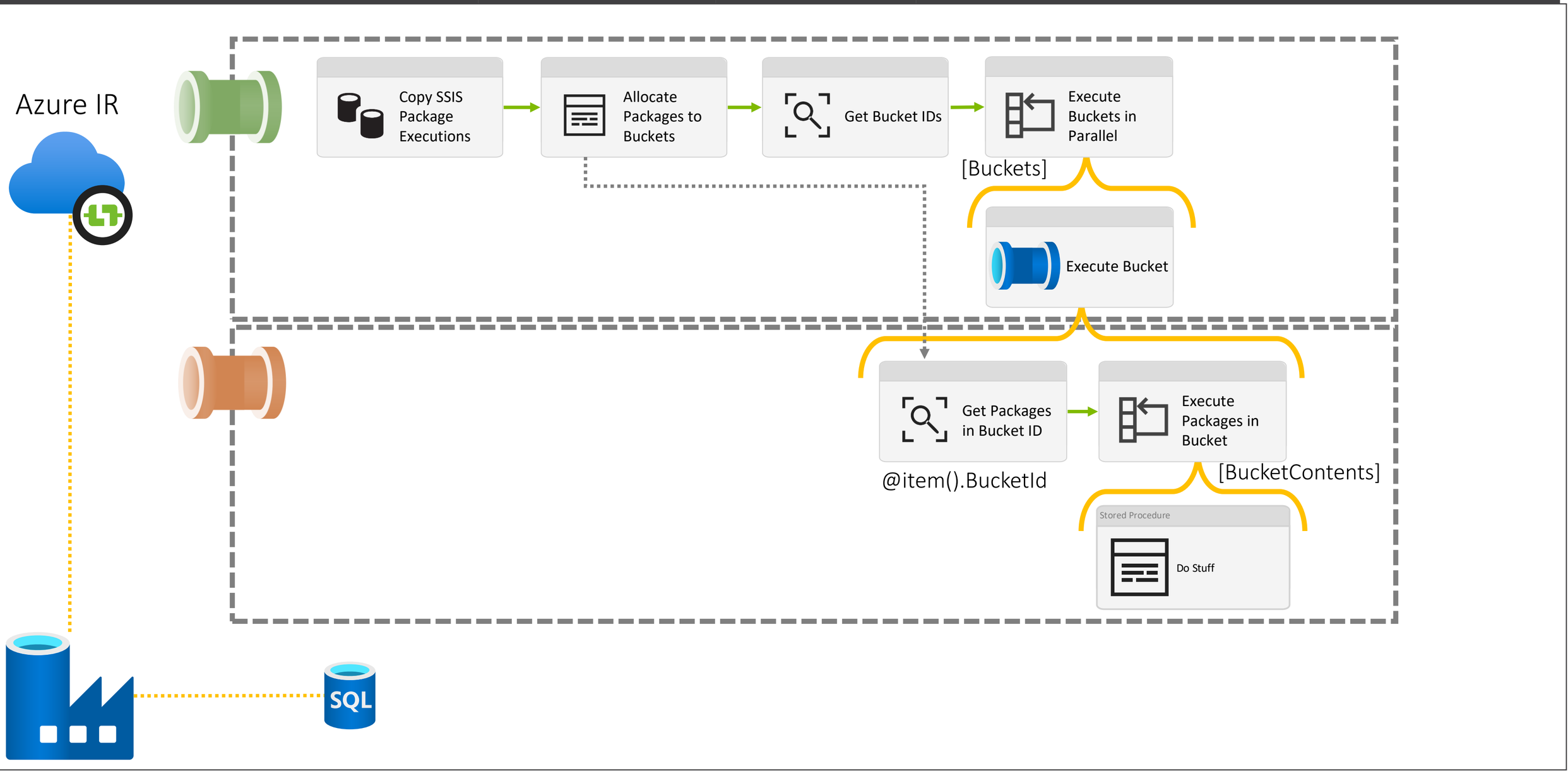
Hosted IR

SSIS IR

Solution 4: Nested ForEach Activities & Bucket Metadata



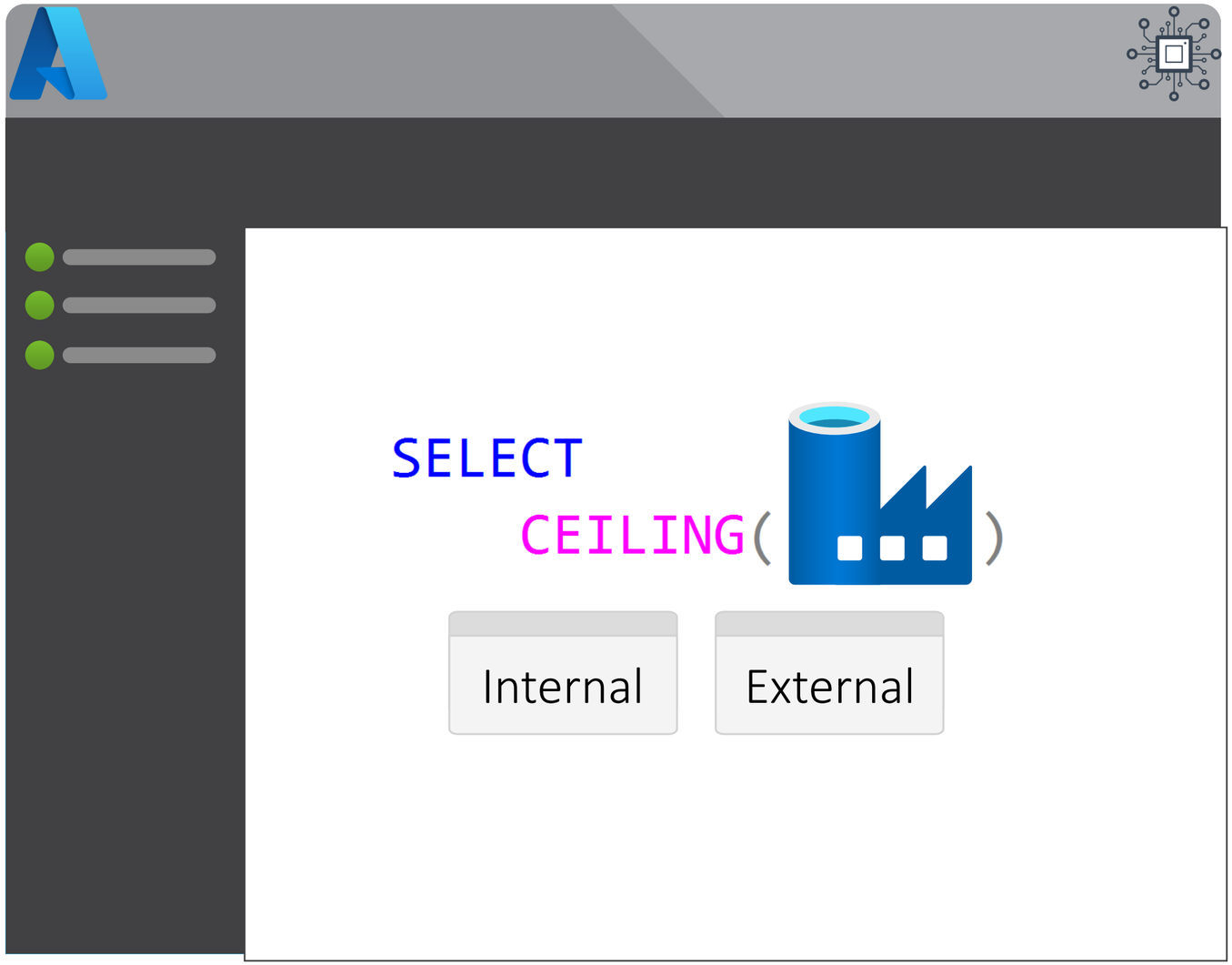
A General Pattern for Scaling Out



Demo

Module 6

Execution Parallelism



- Control Flow Scale Out
- Concurrency Limitations
- Internal vs External Activities
- Orchestration Framework - http://procfwk.com

Data Factory Limitations

Resource Limits

mrpaulandrew.com/2020/01/29/azure-data-factory-resource-limitations/

https://github.com/MicrosoftDocs/azure-docs/blob/main/includes/azure-data-factory-limits.md

800 Data Factory Instances per Subscription

40 Activities per Data Factory Pipeline

3 Active Data Flow Debug Sessions per Data Factory

5,000 Rows or 4MB of Data Returned per Lookup (No Error if More)

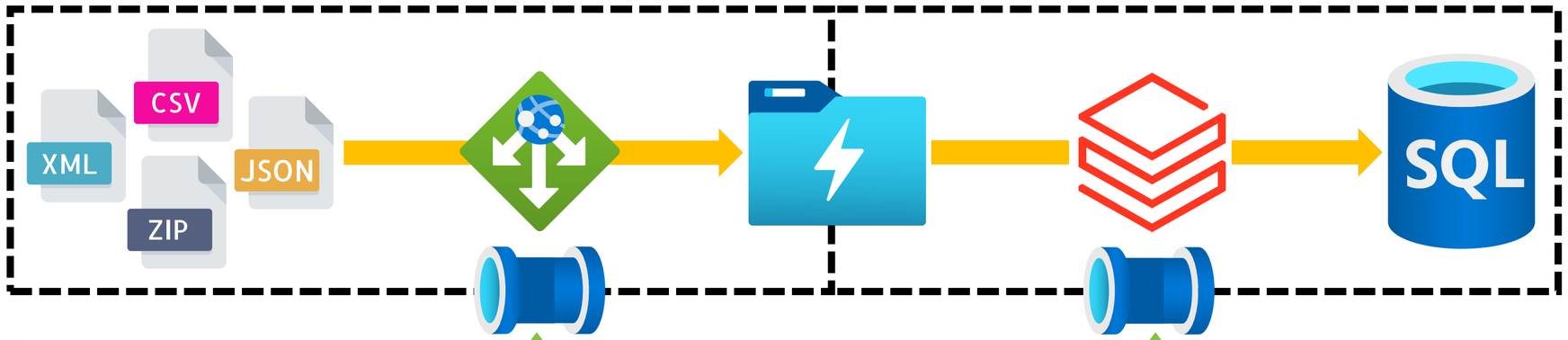
Minimum Tumbling Window Trigger – 15mins

4min Client Response Timeout Using Azure Functions Activity

5,000 Entities (Components) per Data Factory Instance

Integration Pipelines as Data Engineers

Control Flow



1 Linked Services

2 Datasets

3 Activities

4 Pipelines

5 Triggers

Integration Runtimes 6

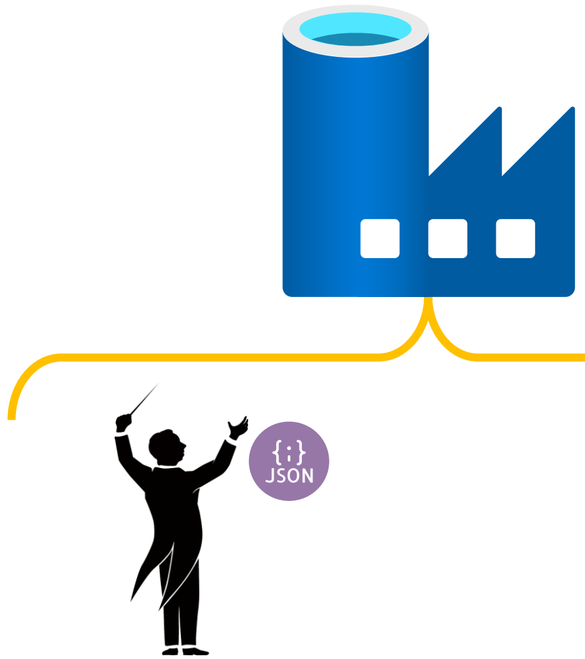
Azure IR

Hosted IR

SSIS IR

Azure Integration Runtime

Compute Concurrency



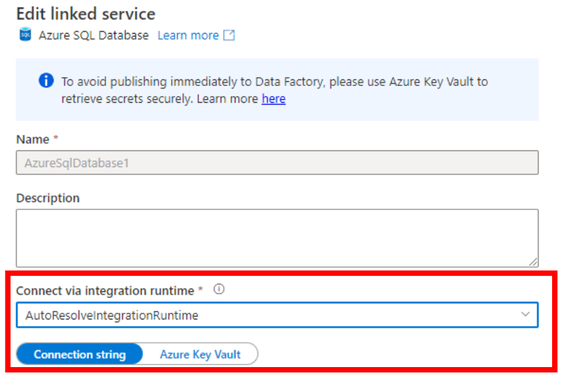
Orchestrator

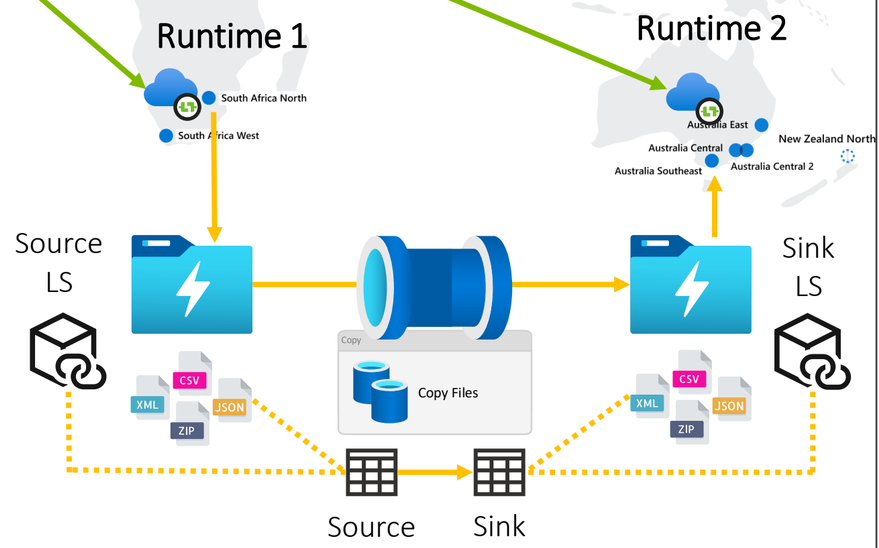
Fixed Region

Runtime

Flexible Location

AutoResolveIntegrationRuntime

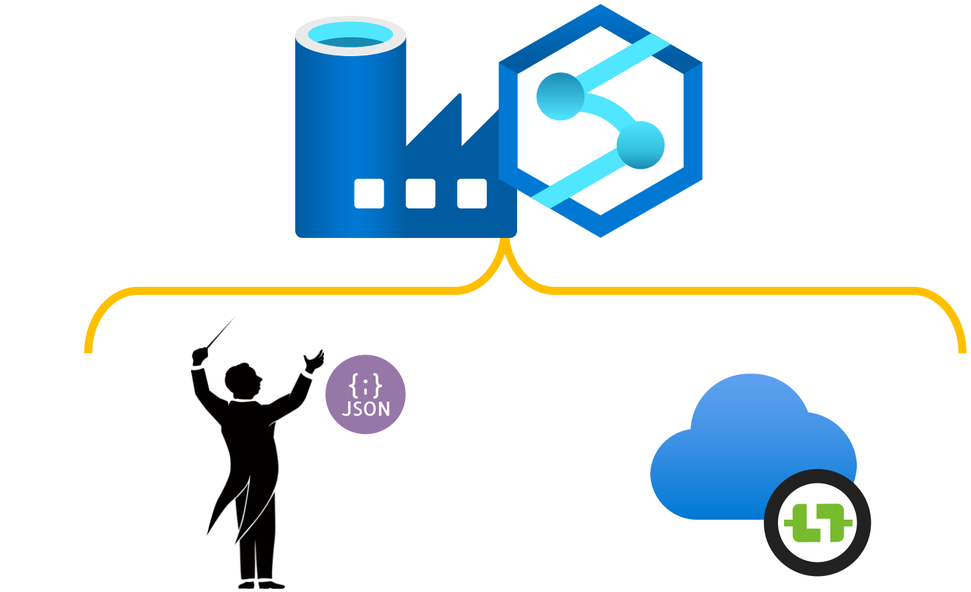




Compute Concurrency

Internal vs External Activities

https://mrpaulandrew.com/2020/12/22/pipelines-understanding-internal-vs-external-activities/



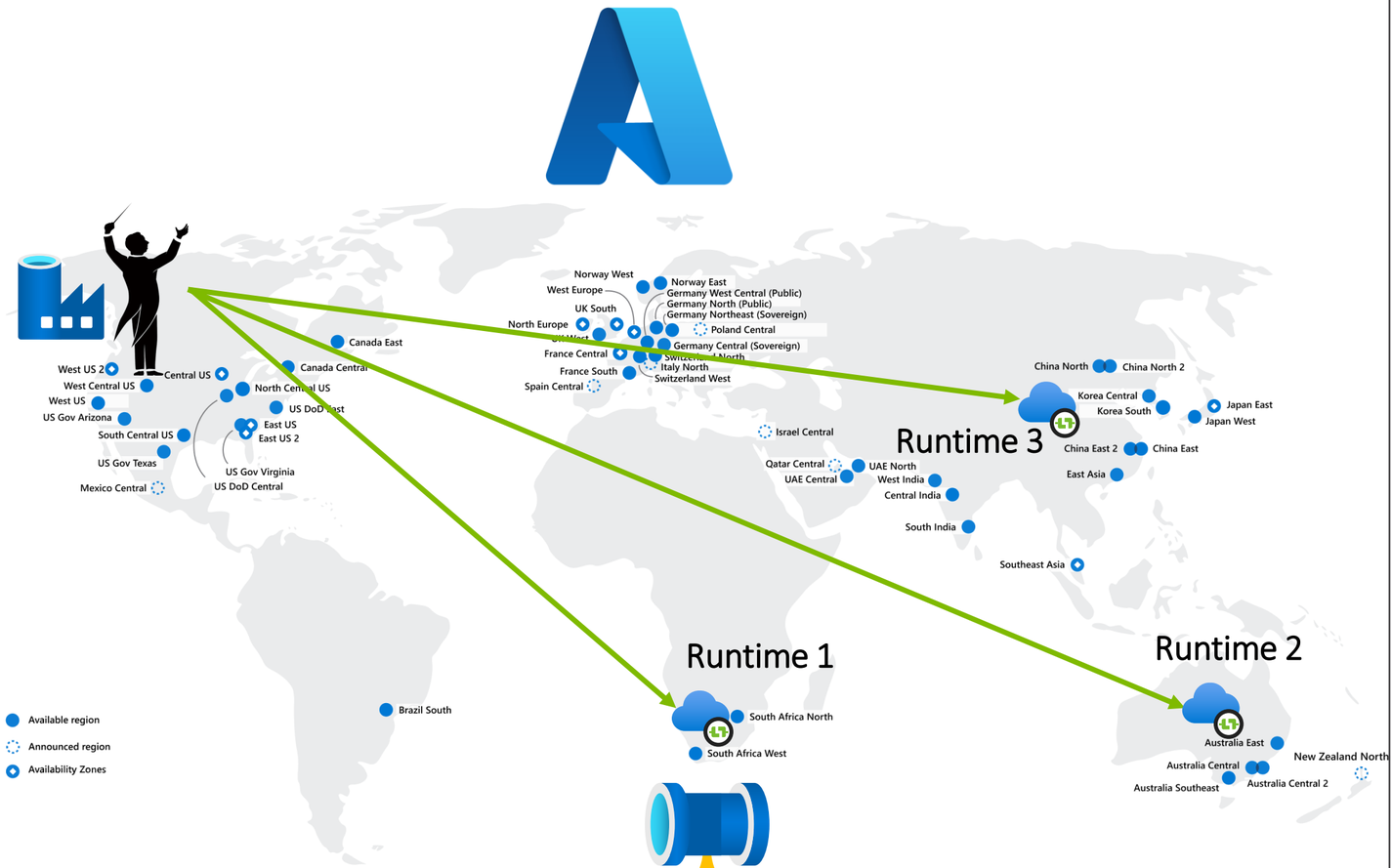
Orchestrator

Fixed Region

Runtime

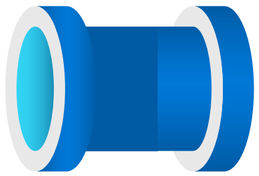
Flexible Region

AutoResolve
IntegrationRuntime



Concurrency – Pipelines vs Activities

Per Subscription, per IR Region



10,000

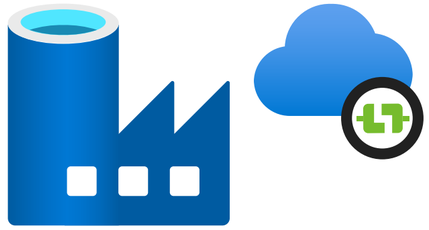
Internal 1,000

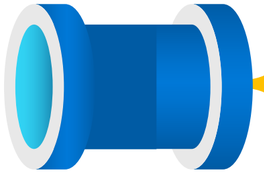
External 3,000

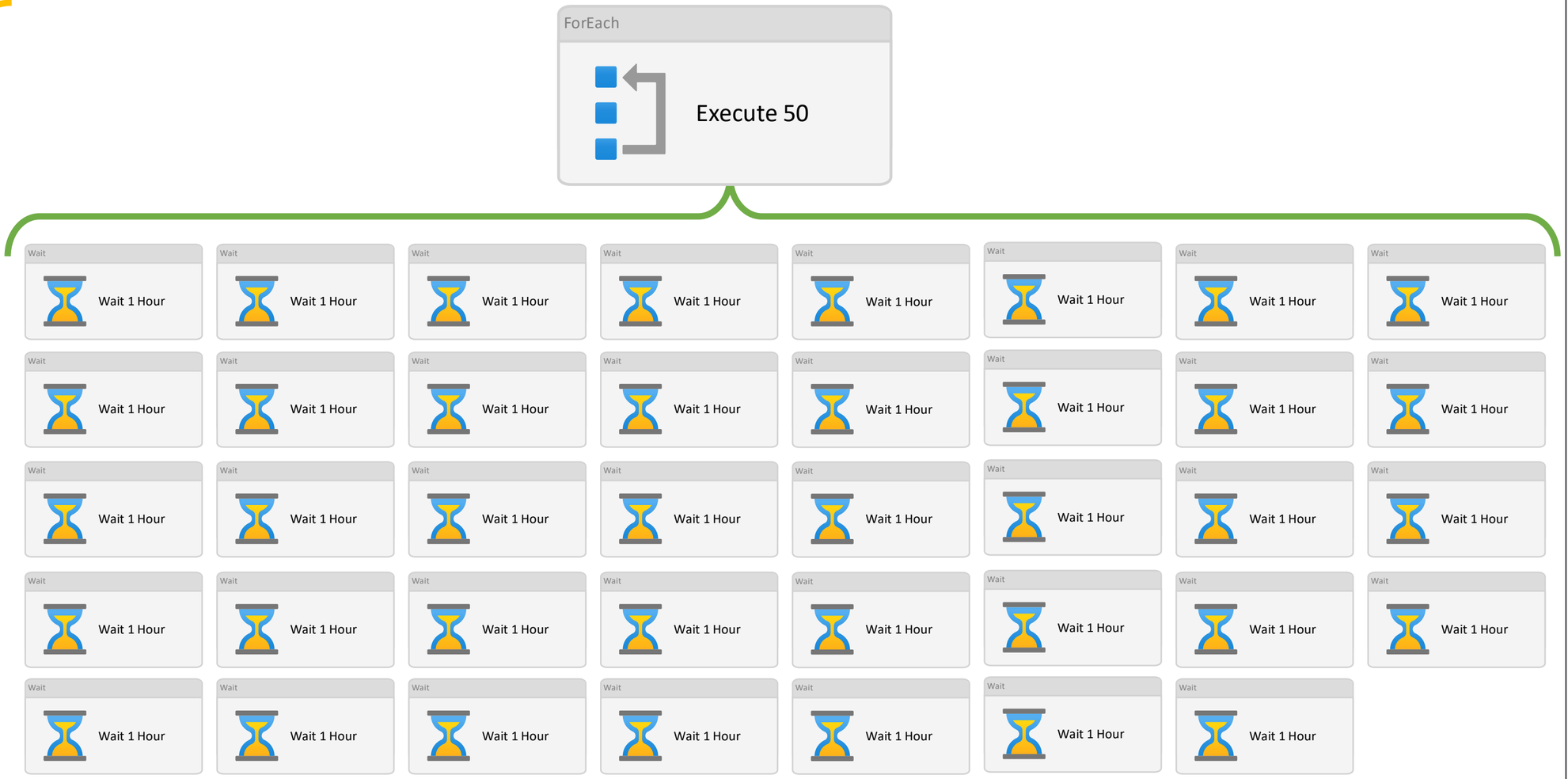
https://mrpaulandrew.com/2020/01/29/azure-data-factory-resource-limitations/

Concurrency – Pipelines vs Activities

Example 1







1 IR

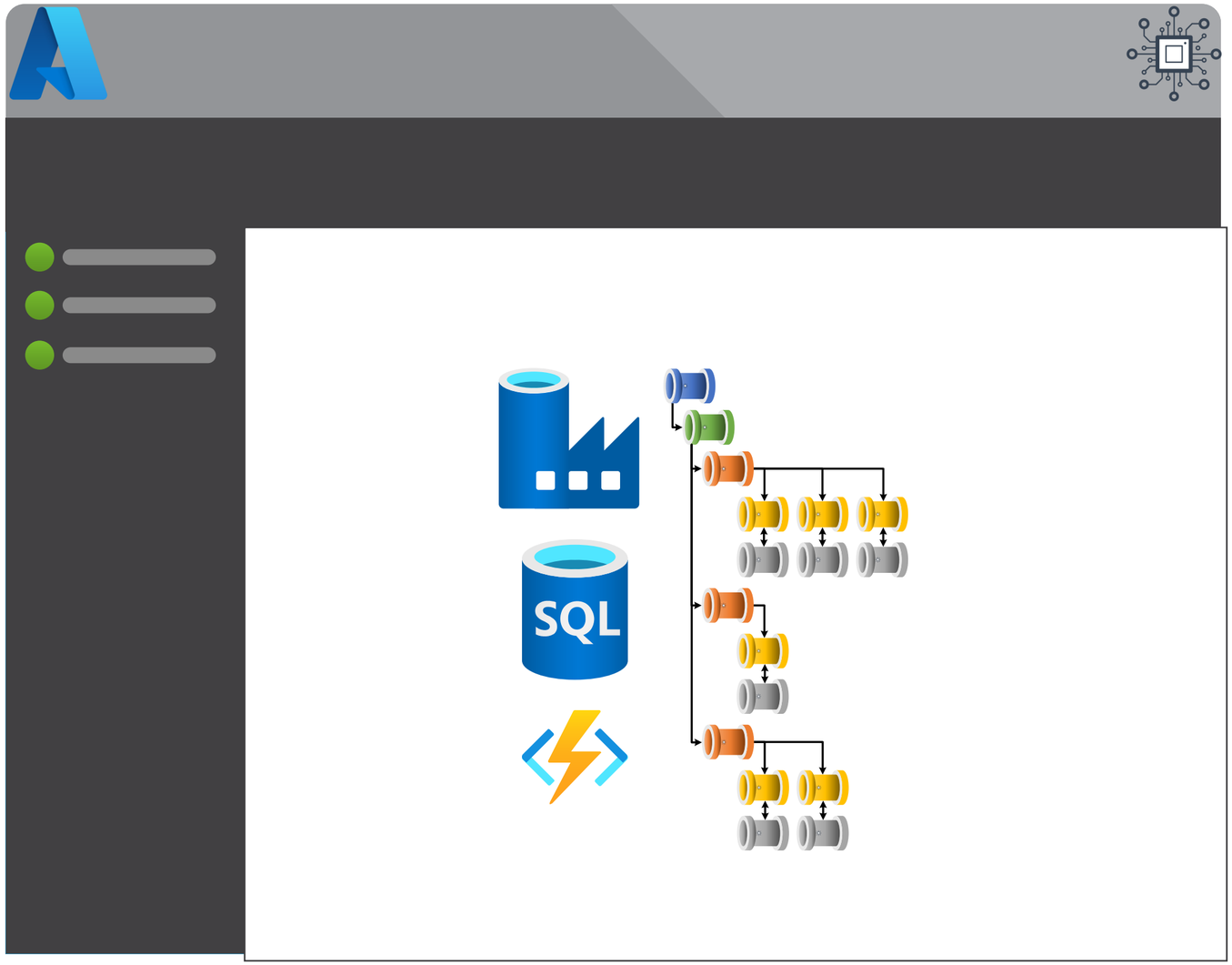
1 Pipeline

1 ForEach + (50 Batches X 39 Wait Activities) = 1951 Concurrent Activities

Demo

Module 6

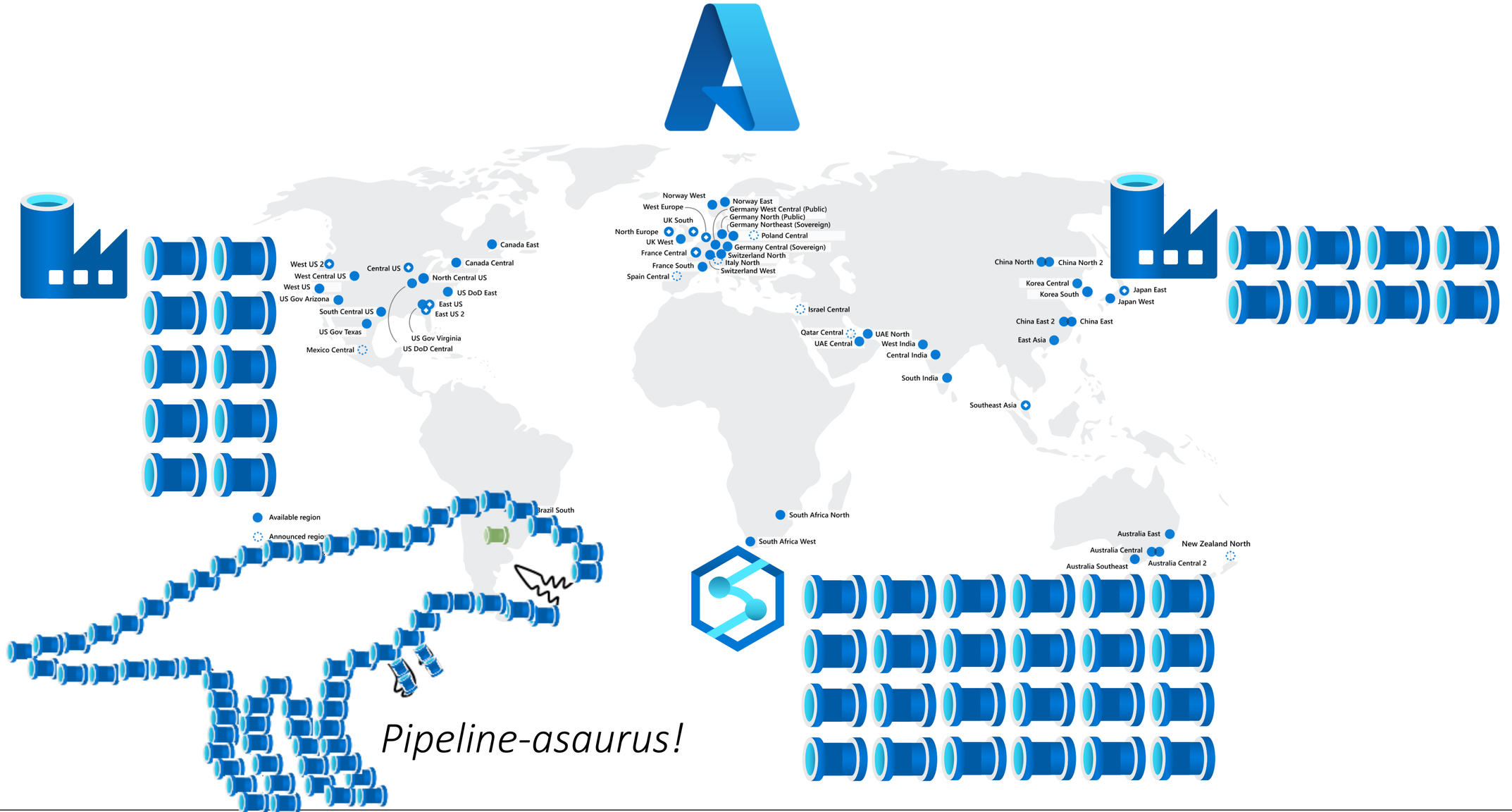
Execution Parallelism



- Control Flow Scale Out
- Concurrency Limitations
- Internal vs External Activities
- Orchestration Framework - http://procfwk.com

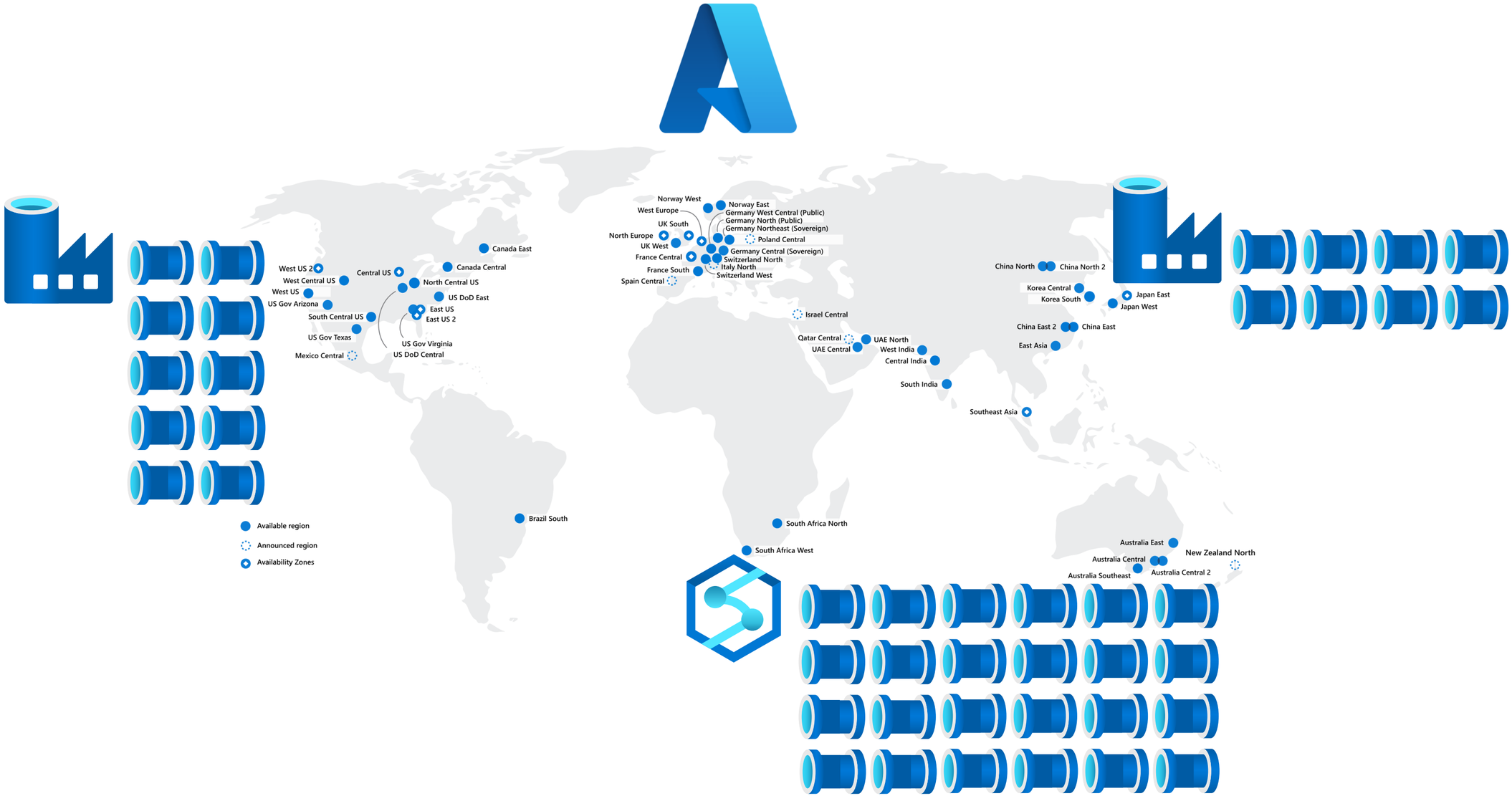
Problem

How should we structure and trigger our Integration Pipelines?

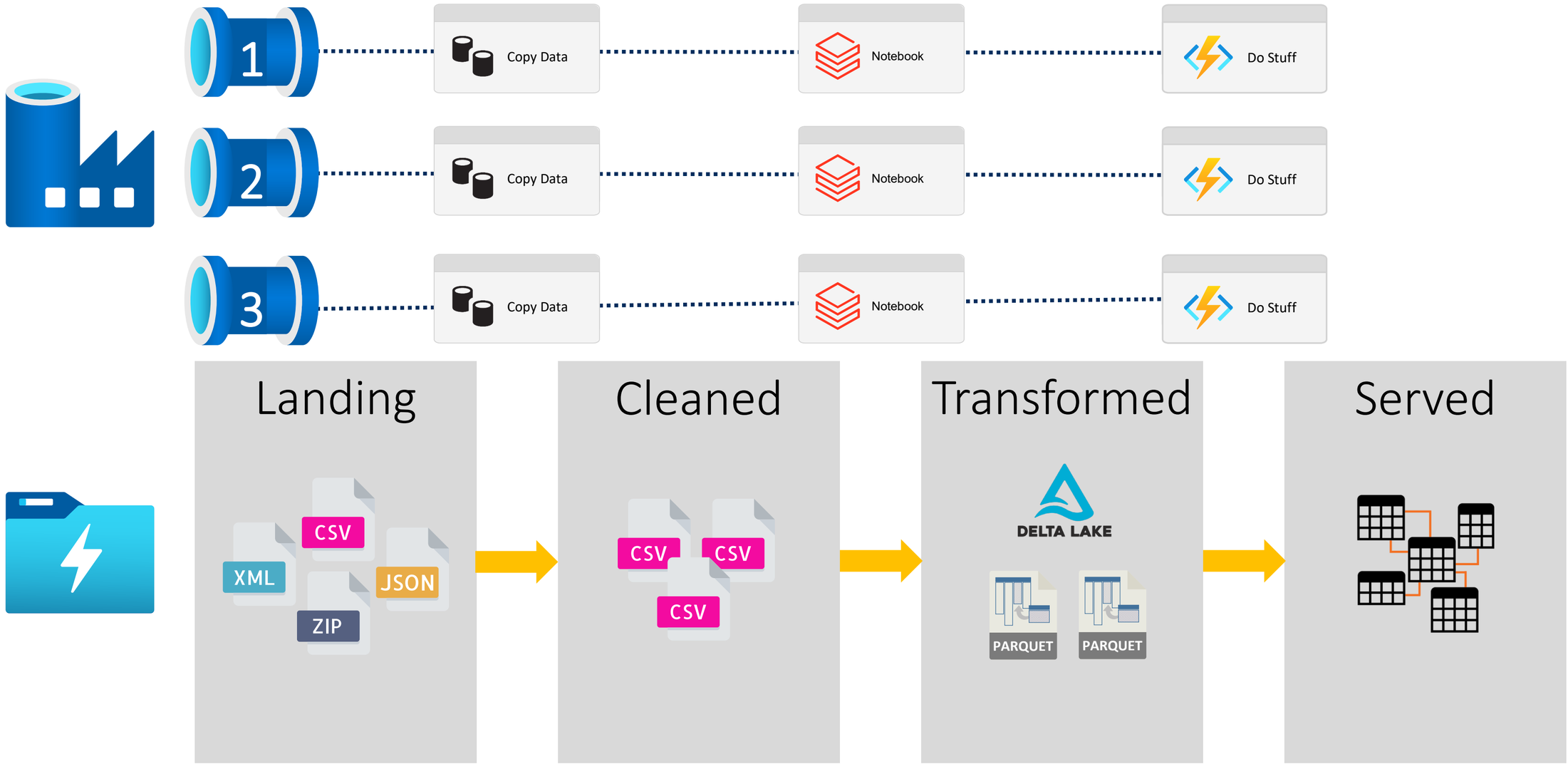


Problem

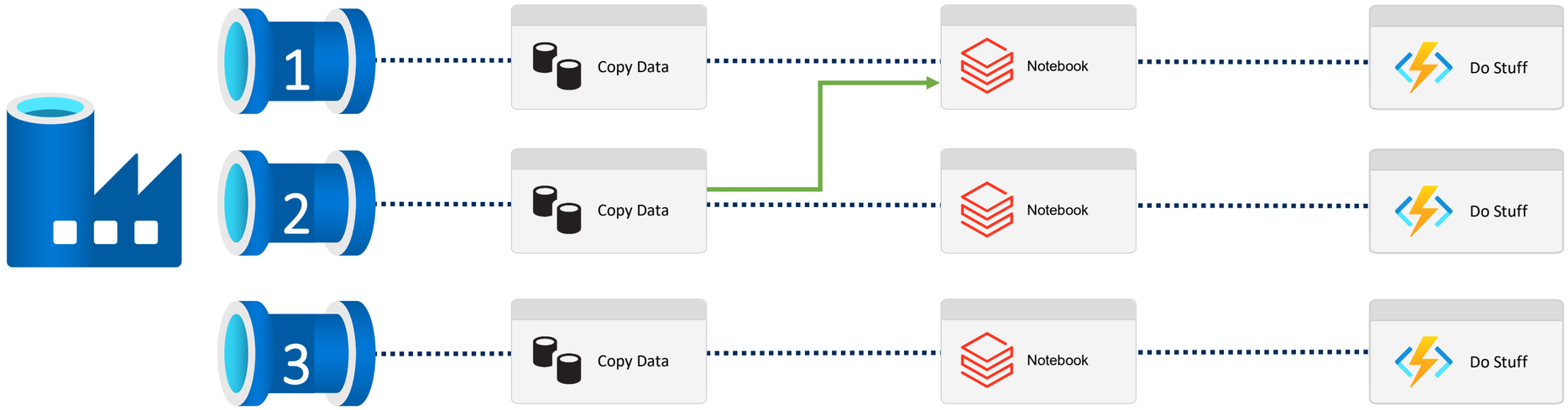
How should we structure and trigger our Integration Pipelines?

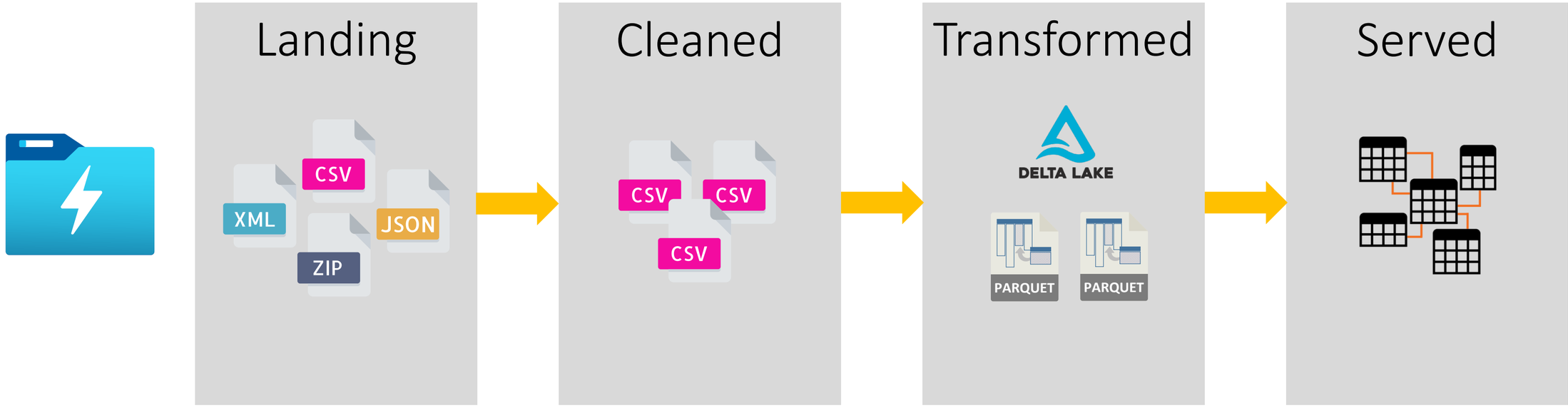


Problem

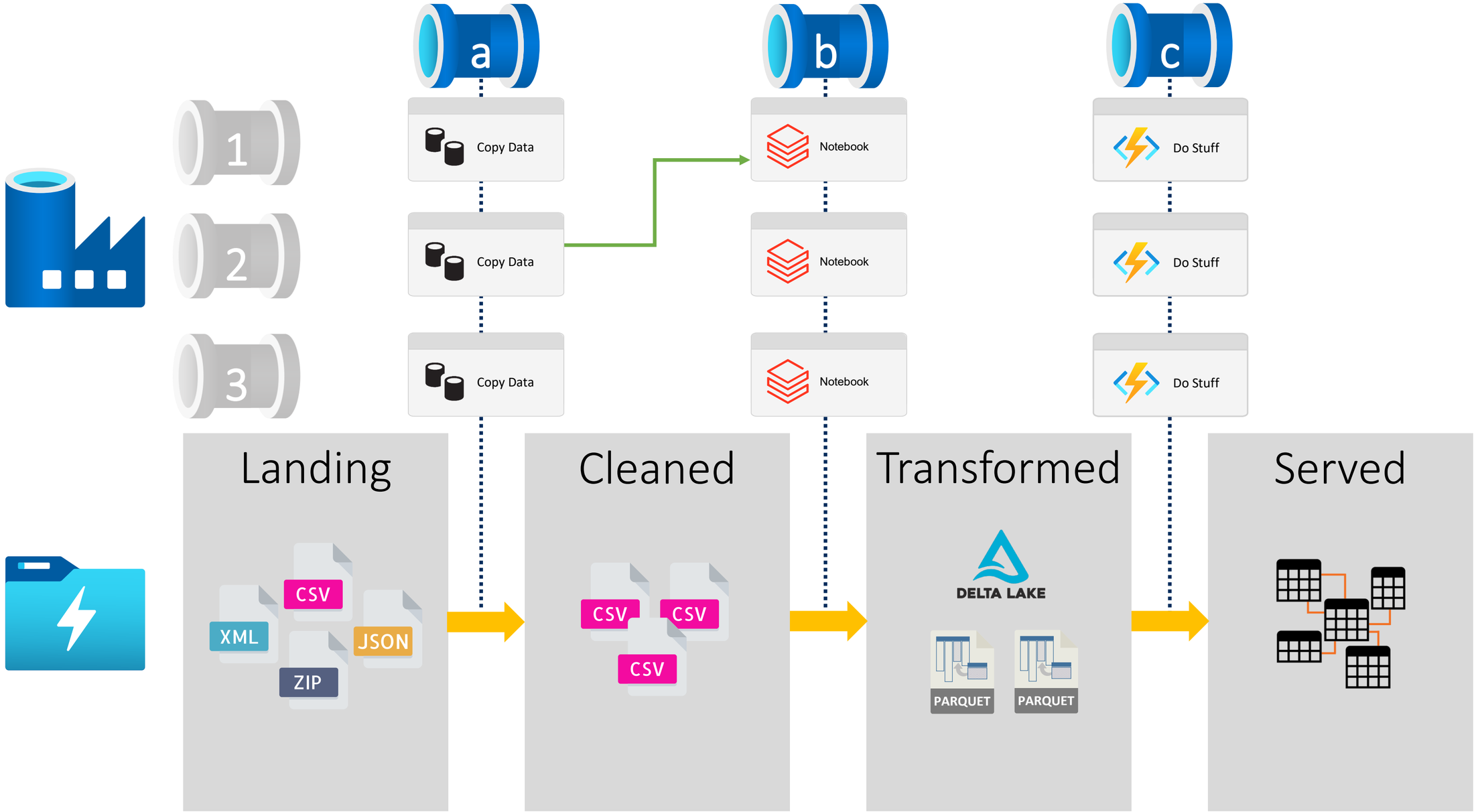

1
Copy Data
Notebook
Do Stuff
2
Copy Data
Notebook
Do Stuff
3
Copy Data
Notebook
Do Stuff
Landing
CSV
XML
JSON
ZIP
Cleaned
CSV
CSV
CSV
Transformed
DELTA LAKE
PARQUET
PARQUET
Served

Problem

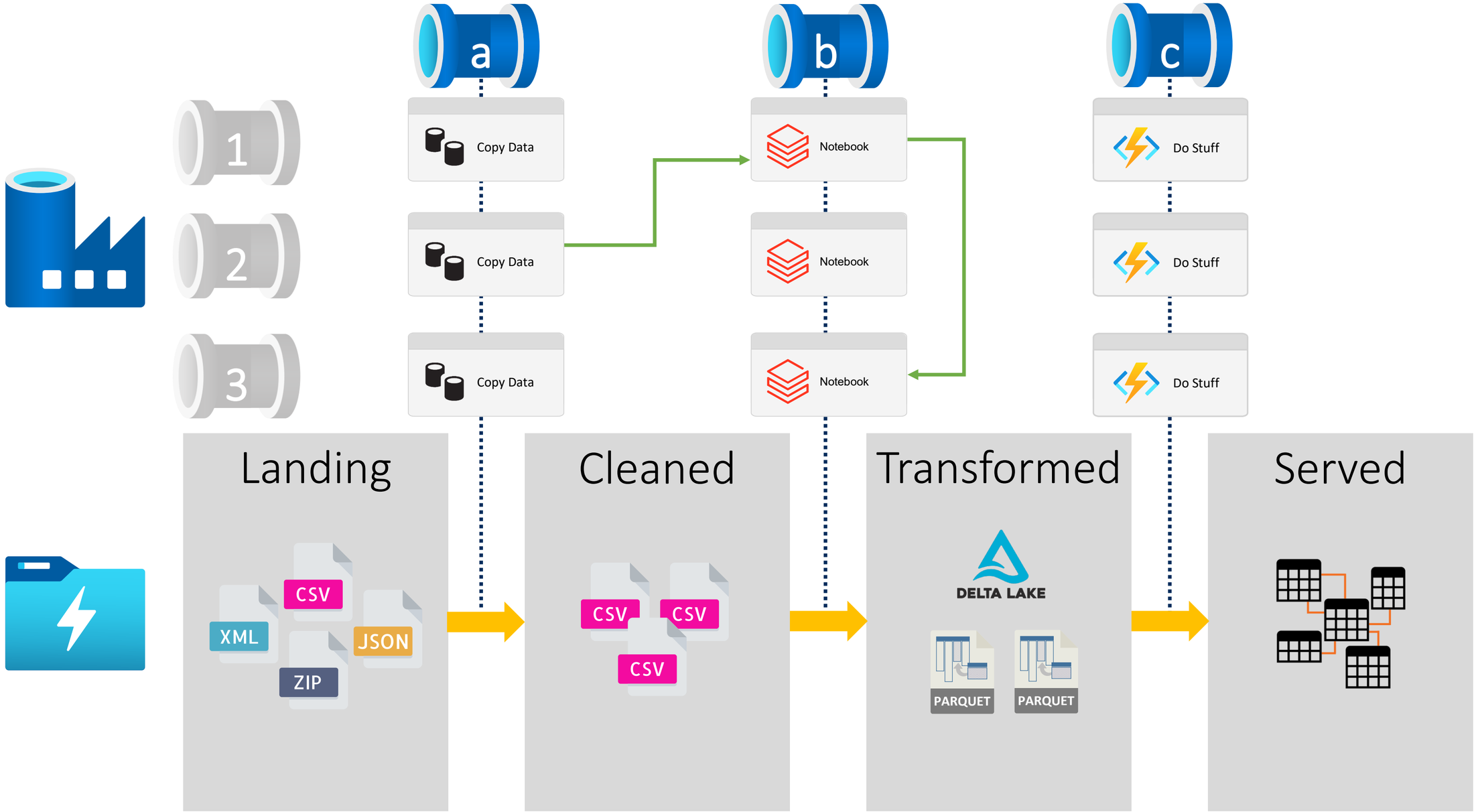

1
2
3
Copy Data
Copy Data
Copy Data
Notebook
Notebook
Notebook
Do Stuff
Do Stuff
Do Stuff


Landing
CSV
XML
JSON
ZIP
Cleaned
CSV
CSV
CSV
Transformed
DELTA LAKE
PARQUET
PARQUET
Served

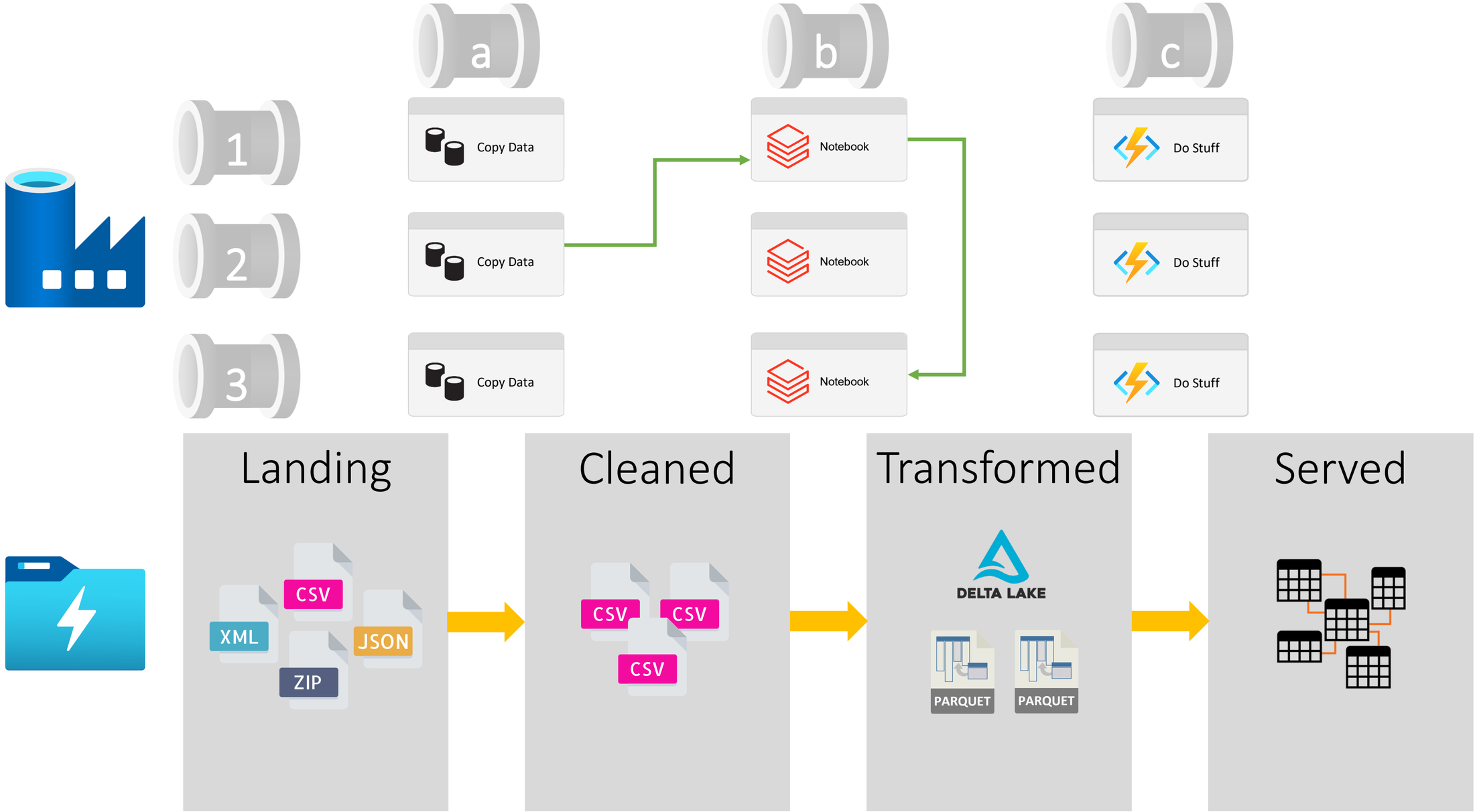
Problem


a
b
c
1
2
3
Copy Data
Copy Data
Copy Data
Notebook
Notebook
Notebook
Do Stuff
Do Stuff
Do Stuff
Landing
Cleaned
Transformed
Served
CSV
XML
JSON
ZIP
CSV
CSV
CSV
DELTA LAKE
PARQUET
PARQUET

Problem

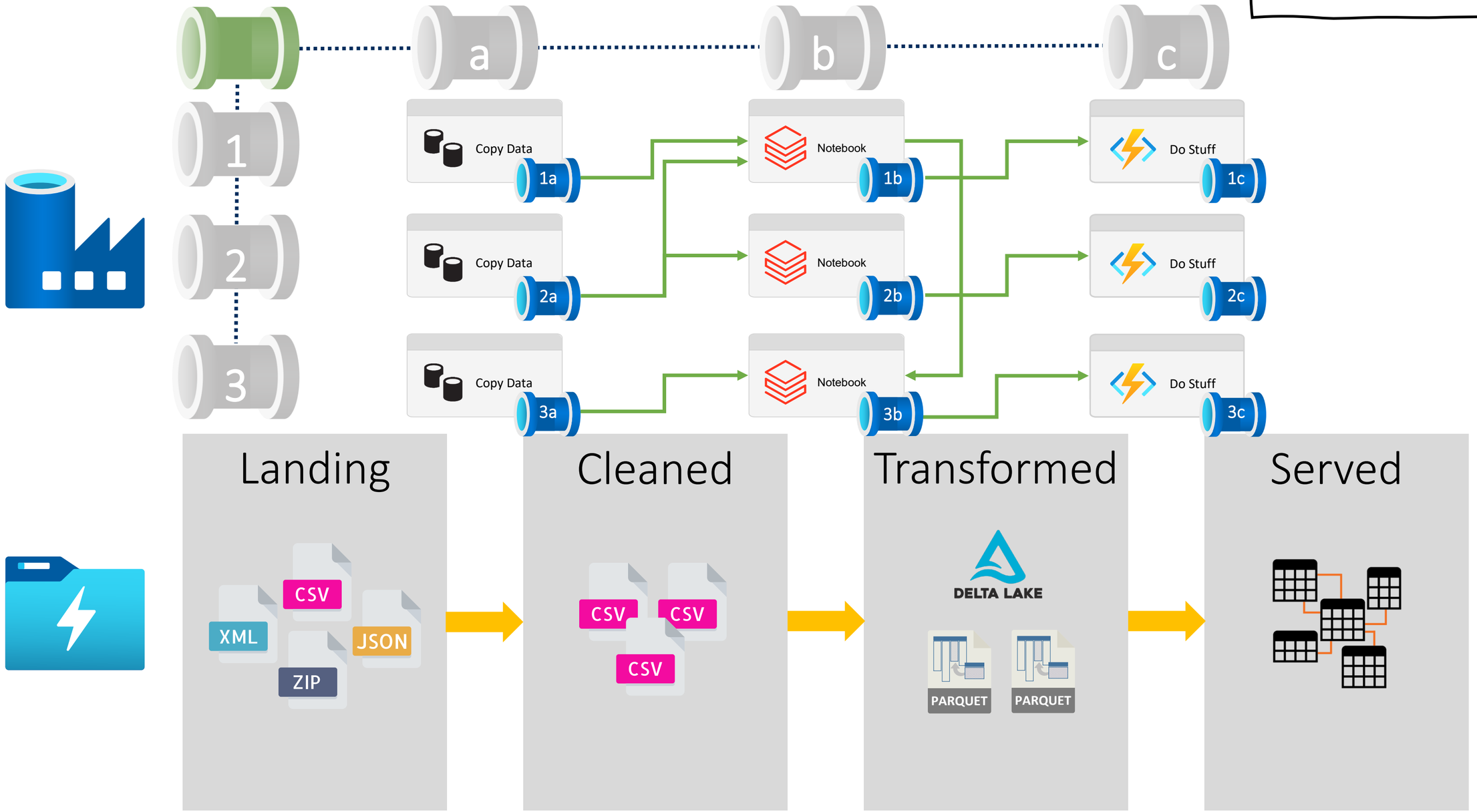

a
b
c
1
2
3
Copy Data
Copy Data
Copy Data
Notebook
Notebook
Notebook
Do Stuff
Do Stuff
Do Stuff
Landing
Cleaned
Transformed
Served
CSV
XML
JSON
ZIP
CSV
CSV
CSV
DELTA LAKE
PARQUET
PARQUET

Problem

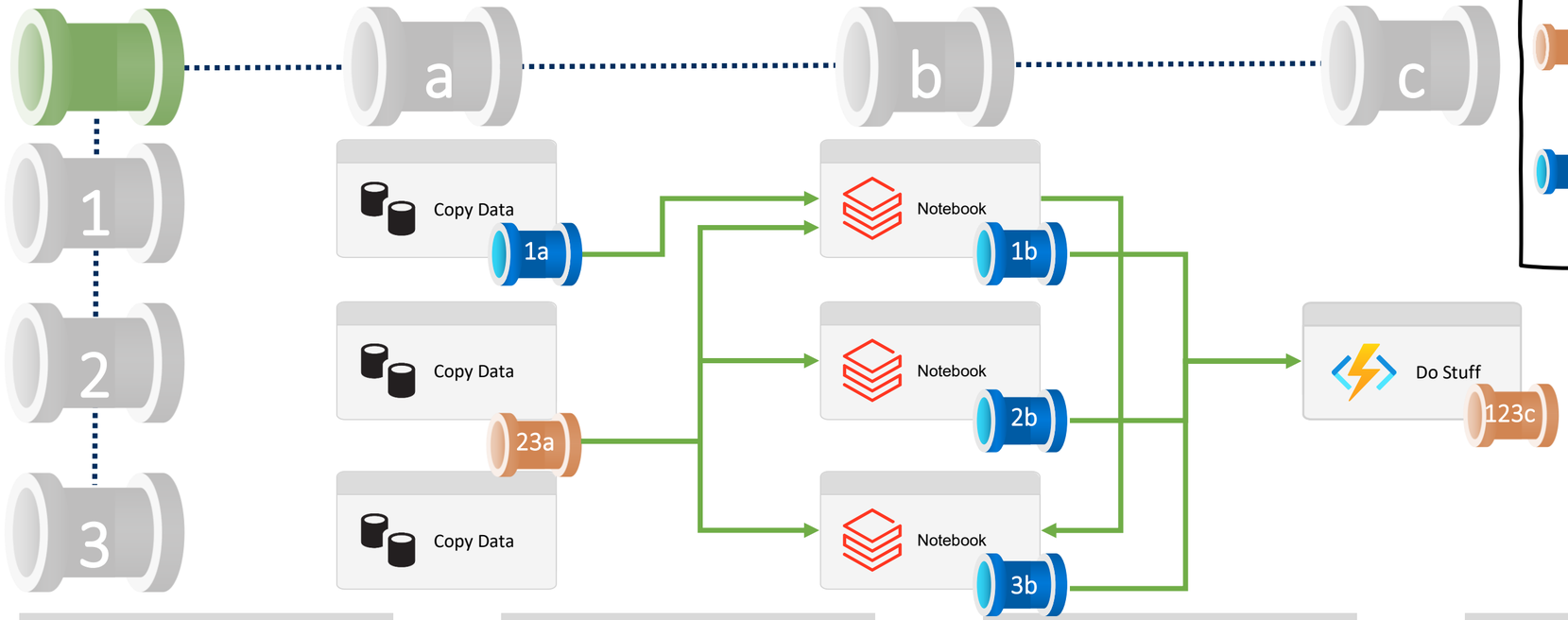

a
b
c
1
2
3
Copy Data
Copy Data
Copy Data
Notebook
Notebook
Notebook
Do Stuff
Do Stuff
Do Stuff
Landing
Cleaned
Transformed
Served
CSV
XML
JSON
ZIP
CSV
CSV
CSV
DELTA LAKE
PARQUET
PARQUET

Problem

Only 40 Activities per Pipeline.



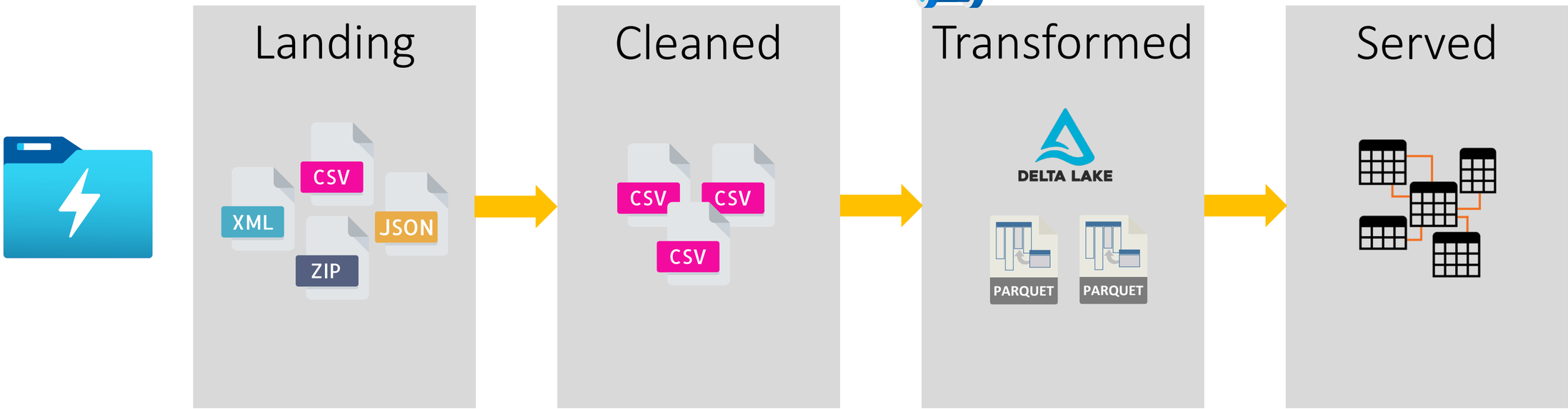
Problem



Grandparent pipeline for all processing.

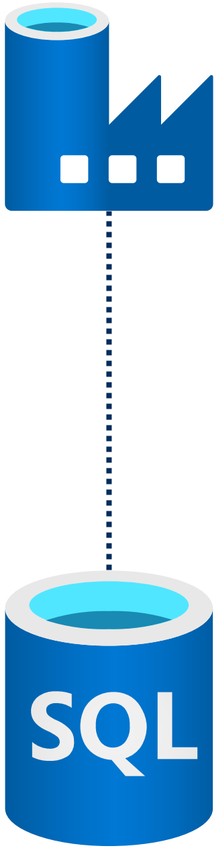
Parent pipeline to consolidate work.

Child pipelines for low level dependencies.

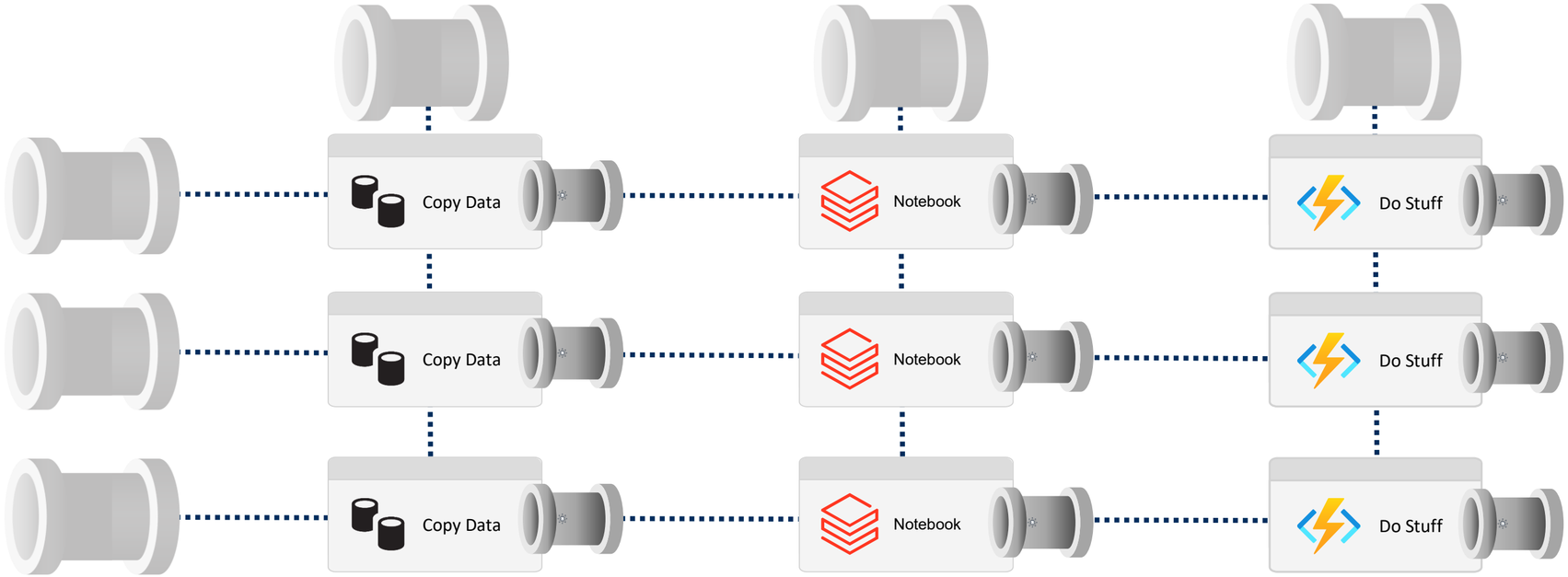


Solution

Use Metadata to Drive Data Factory Pipelines



Solution



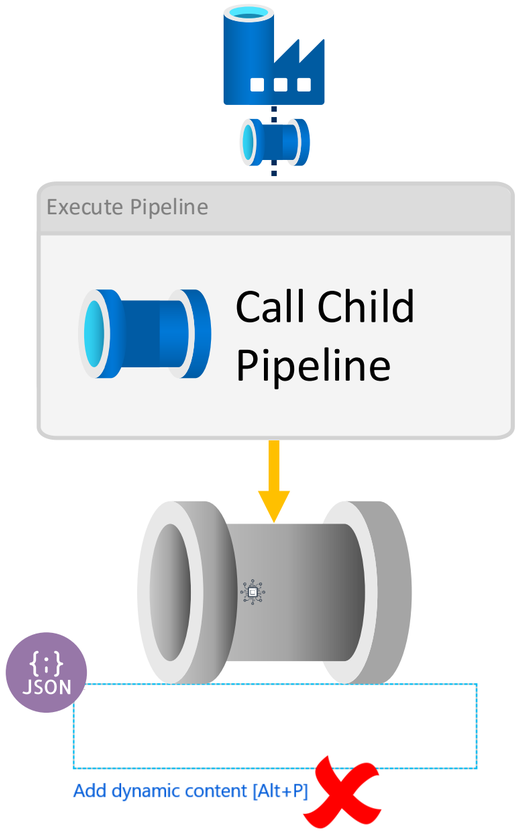
Stages
1
2
3

Pipelines
a
b
c
d
e
f
g
h
i

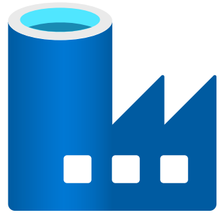
Stage	Pipeline
1	a
1	b
1	c
2	d
2	e
3	f
3	g
3	h
3	i

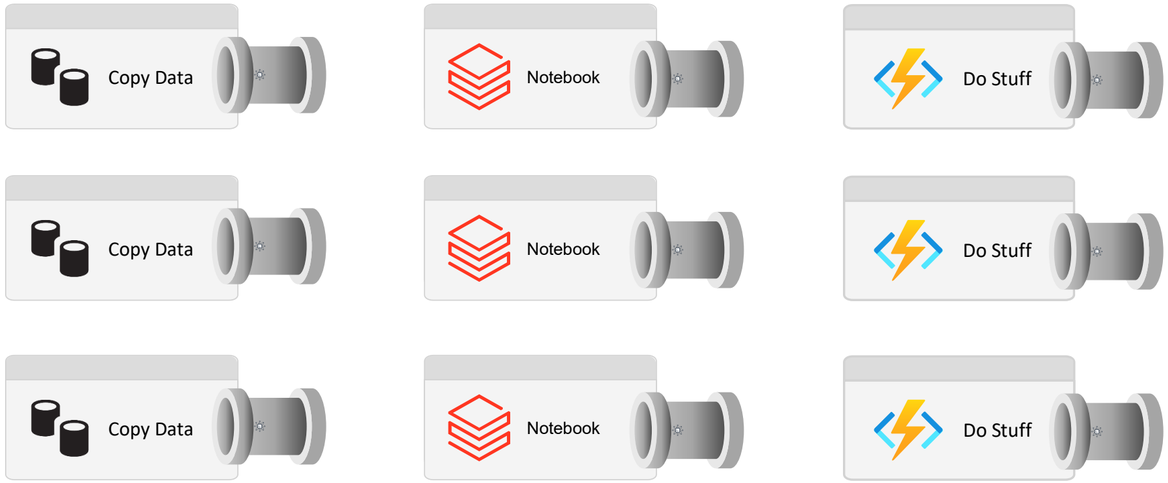
Calling Our Worker Pipelines

One More Problem to Consider



Calling Our Worker Pipelines



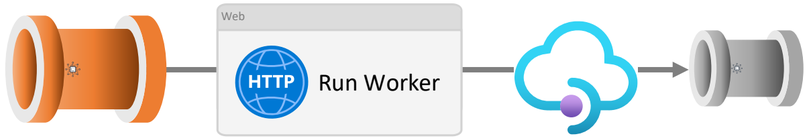


Stages
1
2
3

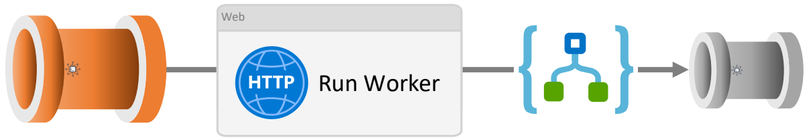
Pipelines
a
b
c
d
e
f
g
h
i

Stage	Pipeline
1	a
1	b
1	c
2	d
2	e
3	f
3	g
3	h
3	i

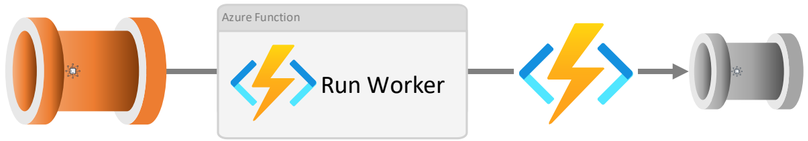
Option 1:



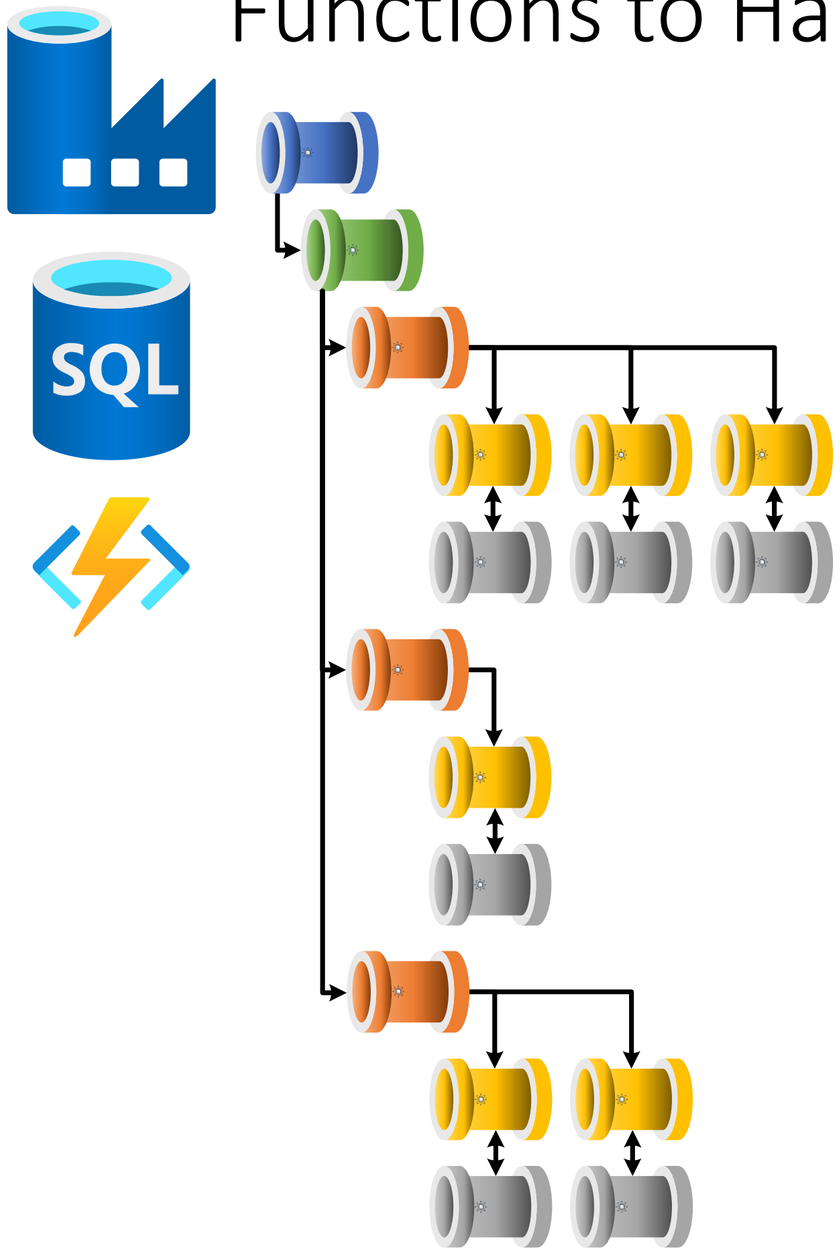
Option 2:



Option 3:



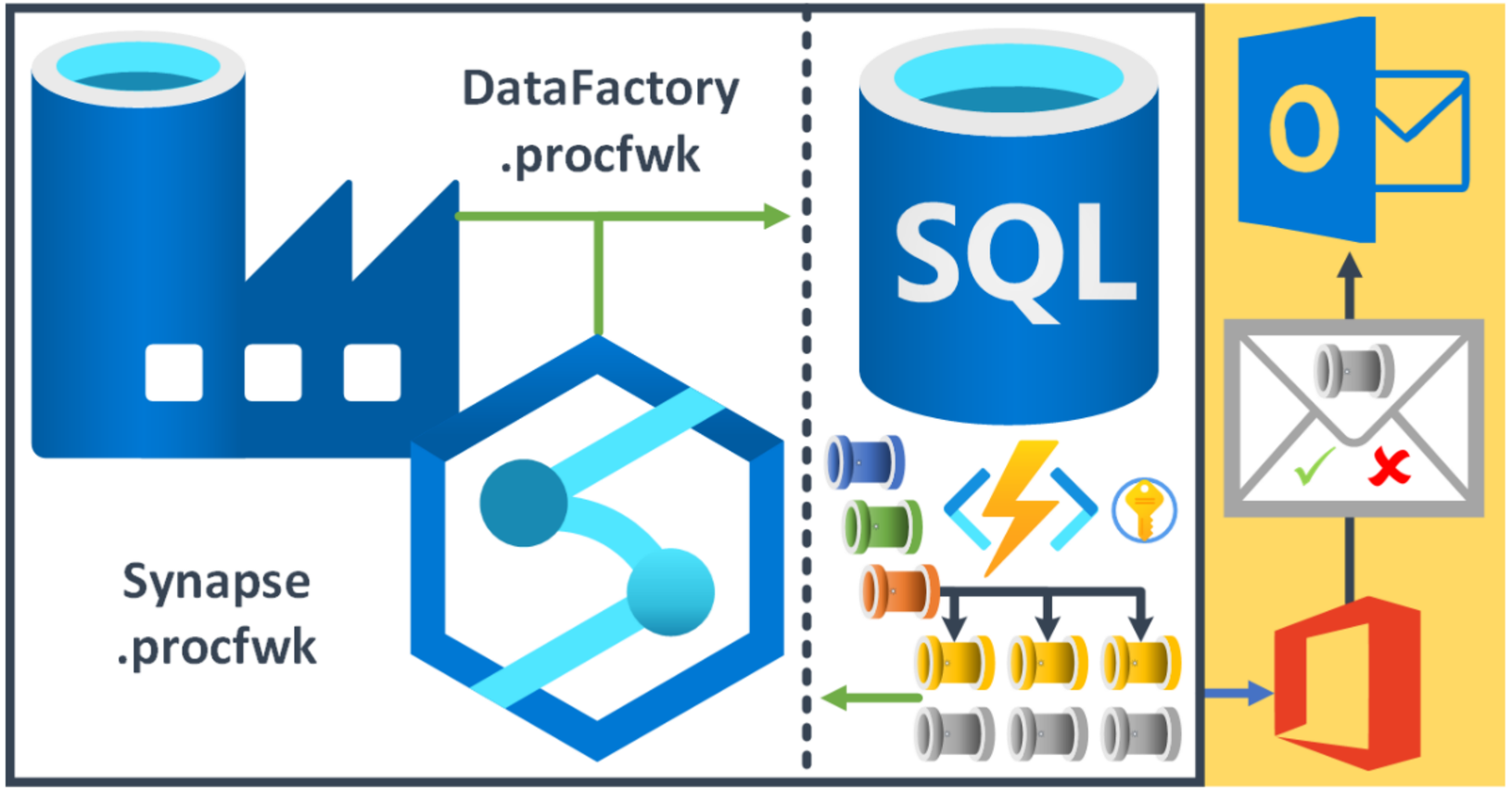
Solution: Use Metadata to Drive Data Factory Pipelines & Functions to Handle the Worker Execution



Introducing ProcFwk.com

stackoverflow

procfwk

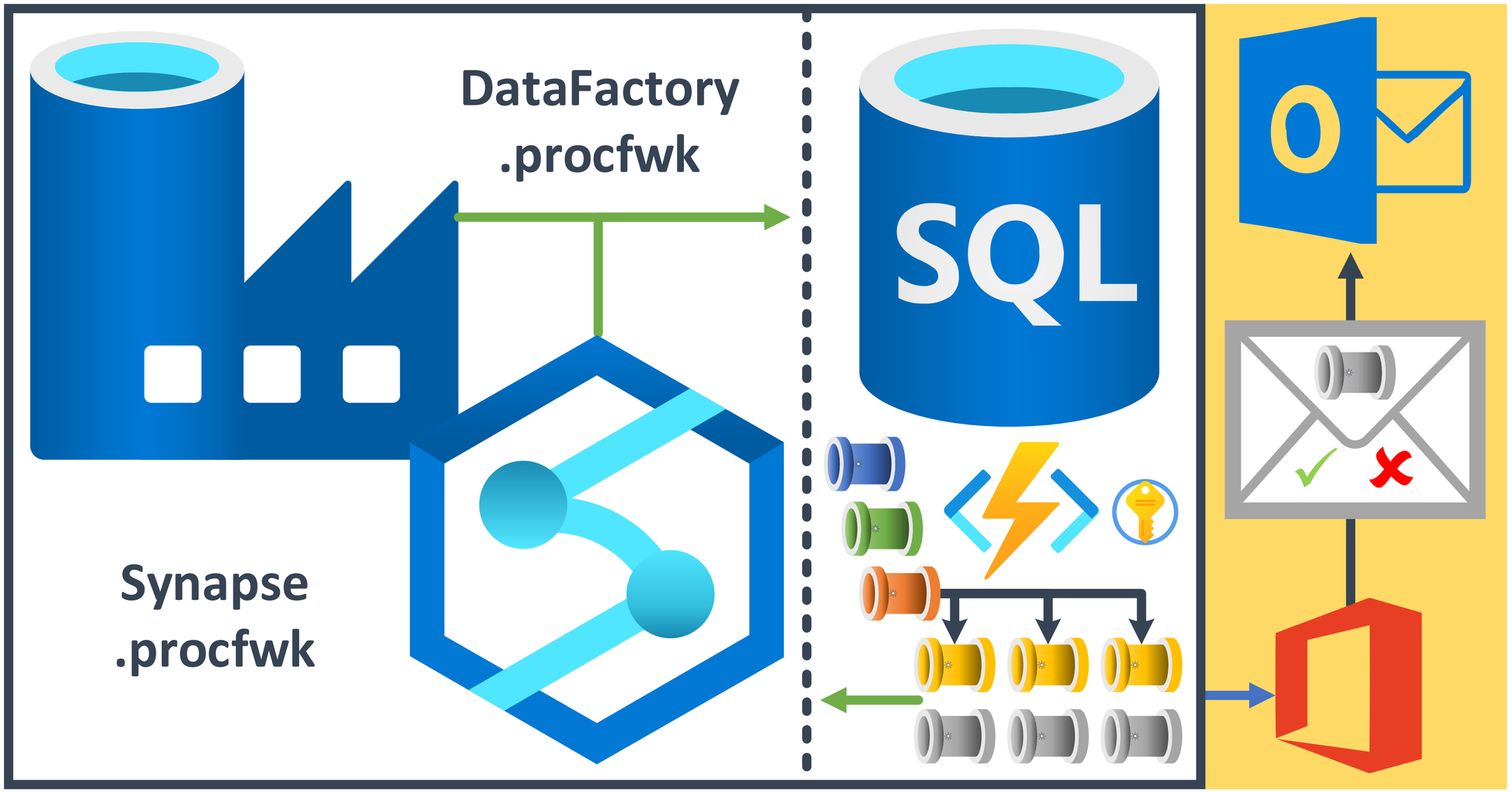


Introducing ProcFwk.com

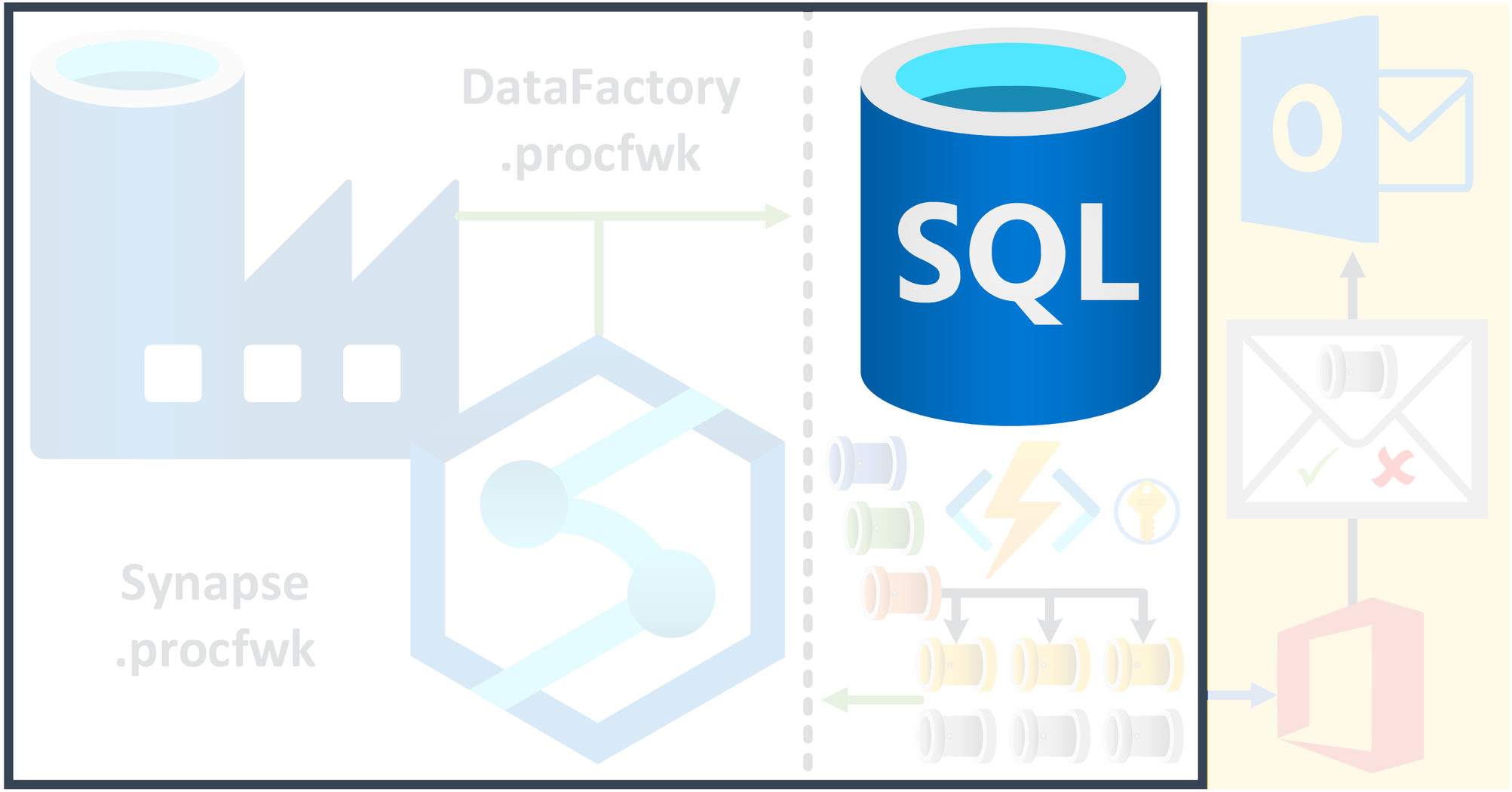
Features:

- Granular metadata control.
- Metadata integrity checking.
- Global properties.
- Complete pipeline dependency chains.
- Concurrent batch executions.
- Execution restart-ability.
- Parallel execution stages.
- Full execution and error logs.
- Operational dashboarding.
- Low-cost orchestration.
- Disconnection between framework and worker pipelines.
- Cross Tenant/Subscription/Data Factory control flows.
- Pipeline parameter support.
- Simple troubleshooting.
- Easy deployment.
- Email alerting.
- Automated testing.
- Azure Key Vault integration.
- Is pipeline already running checks.

ProcFwk Tour



ProcFwk Tour – Database (Metadata)



Framework Database

AlertOutcomes (procfwk)
- OutcomeBitPosition
- PipelineOutcomeStatus
- BitValue

PipelineAlertLink (procfwk)
- AlertId
- PipelineId
- RecipientId
- OutcomesBitValue
- Enabled

Recipients (procfwk)
- RecipientId
- Name
- EmailAddress
- MessagePreference
- Enabled

Batches (procfwk)
- BatchId
- BatchName
- BatchDescription
- Enabled

PipelineParameters (procfwk)
- ParameterId
- PipelineId
- ParameterName
- ParameterValue
- ParameterValueLastUsed

Pipelines (procfwk)
- PipelineId
- OrchestratorId
- StageId
- PipelineName
- LogicalPredecessorId
- Enabled

BatchStageLink (procfwk)
- BatchId
- StageId

PipelineDependencies (procfwk)
- DependencyId
- PipelineId
- DependantPipelineId

Stages (procfwk)
- StageId
- StageName
- StageDescription
- Enabled

Orchestrators (procfwk)
- OrchestratorId
- OrchestratorName
- OrchestratorType
- IsFrameworkOrchestrator
- ResourceGroupName
- SubscriptionId
- Description

PipelineAuthLink (procfwk)
- AuthId
- PipelineId
- OrchestratorId
- CredentialId

Subscriptions (procfwk)
- SubscriptionId
- Name
- Description
- TenantId

ServicePrincipals (dbo)
- CredentialId
- PrincipalName
- PrincipalId
- PrincipalSecret
- PrincipalIdUrl
- PrincipalSecretUrl

Tenants (procfwk)
- TenantId
- Name
- Description

Properties (procfwk)
- PropertyId
- PropertyName
- PropertyValue
- Description
- ValidFrom
- ValidTo

BatchExecution (procfwk)
- BatchId
- ExecutionId
- BatchName
- BatchStatus
- StartDateTime
- EndDateTime

CurrentExecution (procfwk)
- LocalExecutionId
- StageId
- PipelineId
- CallingDataFactoryName
- ResourceGroupName
- DataFactoryName
- PipelineName
- StartDateTime
- PipelineStatus
- LastStatusCheckDateTime
- EndDateTime
- IsBlocked
- AdfPipelineRunId
- PipelineParamsUsed

ExecutionLog (procfwk)
- LogId
- LocalExecutionId
- StageId
- PipelineId
- CallingDataFactoryName
- ResourceGroupName
- DataFactoryName
- PipelineName
- StartDateTime
- PipelineStatus
- EndDateTime
- AdfPipelineRunId
- PipelineParamsUsed

ErrorLog (procfwk)
- LogId
- LocalExecutionId
- AdfPipelineRunId
- ActivityRunId
- ActivityName
- ActivityType
- ErrorCode
- ErrorType
- ErrorMessage

Framework Database

AlertOutcomes (procfwk)
- OutcomeBitPosition
- PipelineOutcomeStatus
- BitValue

PipelineAlertLink (procfwk)
- AlertId
- PipelineId
- RecipientId
- OutcomesBitValue
- Enabled

Recipients (procfwk)
- RecipientId
- Name
- EmailAddress
- MessagePreference
- Enabled

Batches (procfwk)
- BatchId
- BatchName
- BatchDescription
- Enabled

PipelineParameters (procfwk)
- ParameterId
- PipelineId
- ParameterName
- ParameterValue
- ParameterValueLastUsed

Pipelines (procfwk)
- PipelineId
- OrchestratorId
- StageId
- PipelineName
- LogicalPredecessorId
- Enabled

BatchStageLink (procfwk)
- BatchId
- StageId

PipelineDependencies (procfwk)
- DependencyId
- PipelineId
- DependantPipelineId

Stages (procfwk)
- StageId
- StageName
- StageDescription
- Enabled

Orchestrators (procfwk)
- OrchestratorId
- OrchestratorName
- OrchestratorType
- IsFrameworkOrchestrator
- ResourceGroupName
- SubscriptionId
- Description

PipelineAuthLink (procfwk)
- AuthId
- PipelineId
- OrchestratorId
- CredentialId

Subscriptions (procfwk)
- SubscriptionId
- Name
- Description
- TenantId

ServicePrincipals (dbo)
- CredentialId
- PrincipalName
- PrincipalId
- PrincipalSecret
- PrincipalIdUrl
- PrincipalSecretUrl

Tenants (procfwk)
- TenantId
- Name
- Description

Configuration & Behaviour

Properties (procfwk)
- PropertyId
- PropertyName
- PropertyValue
- Description
- ValidFrom
- ValidTo

Core Metadata

Execution Handling

Location & Authentication

Email Alerting

Runtime & Logging

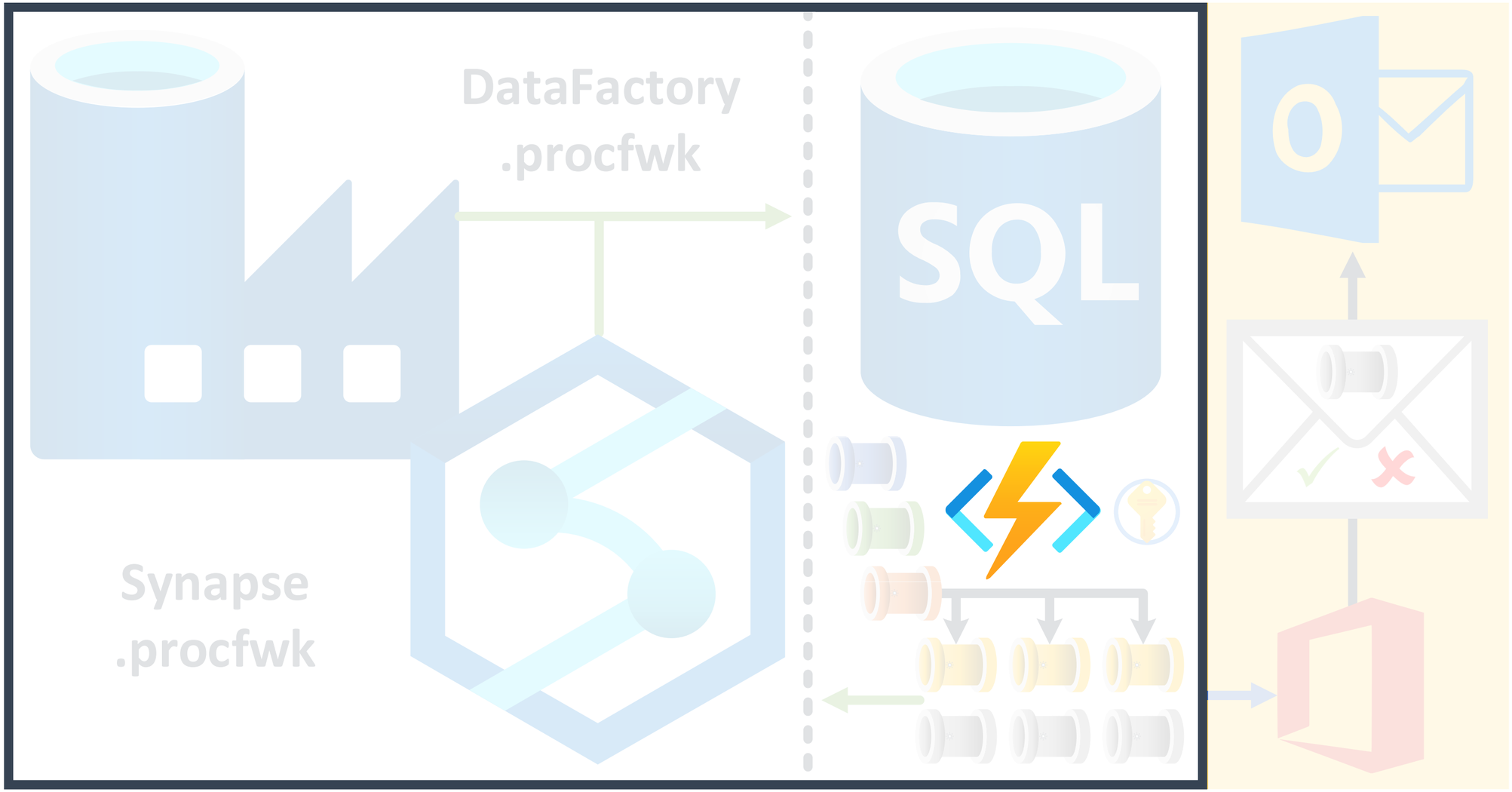
BatchExecution (procfwk)
- BatchId
- ExecutionId
- BatchName
- BatchStatus
- StartDateTime
- EndDateTime

ExecutionLog (procfwk)
- LogId
- LocalExecutionId
- StageId
- PipelineId
- CallingDataFactoryName
- ResourceGroupName
- DataFactoryName
- PipelineName
- StartDateTime
- PipelineStatus
- EndDateTime
- AdfPipelineRunId
- PipelineParamsUsed

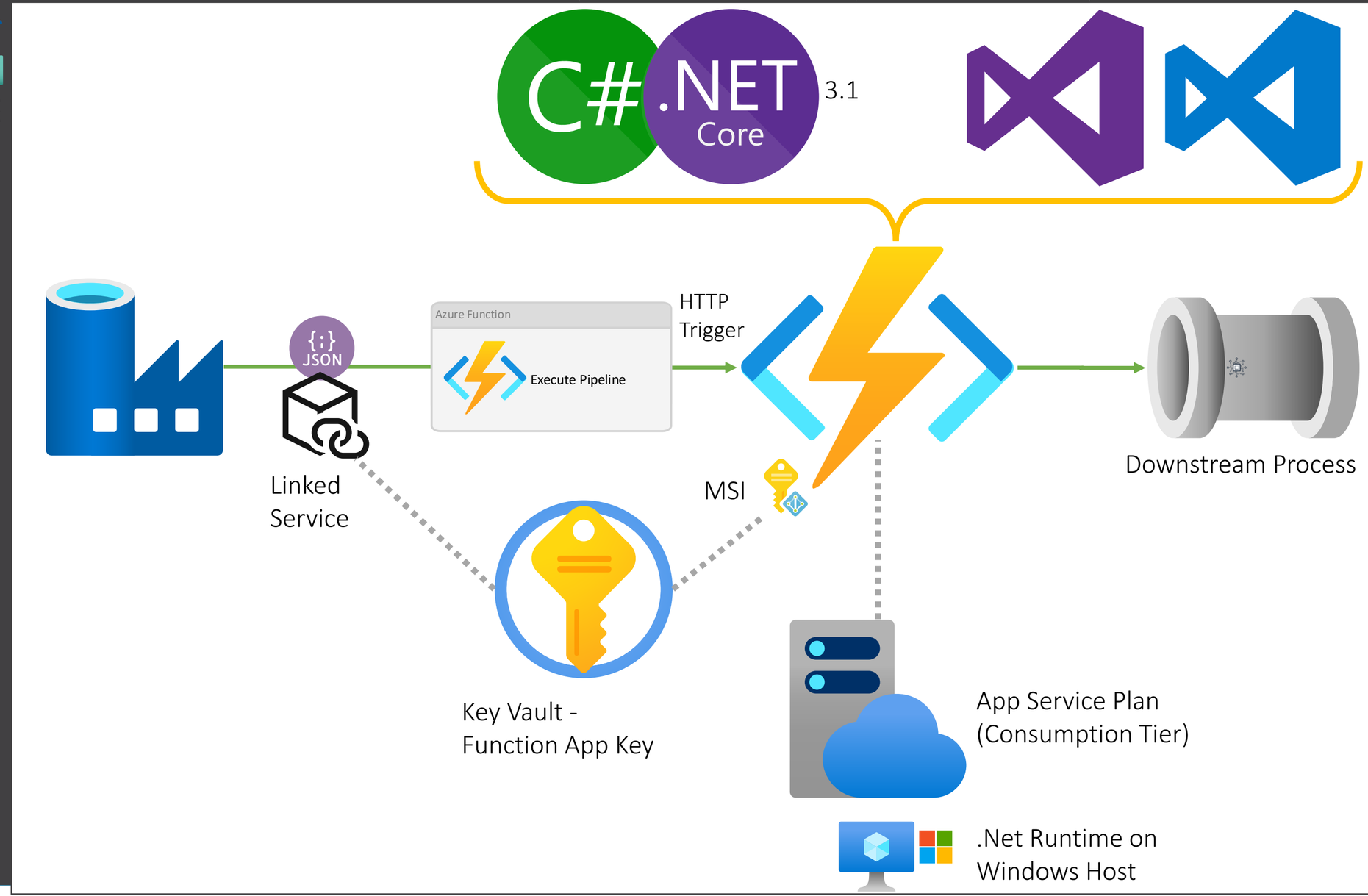
CurrentExecution (procfwk)
- LocalExecutionId
- StageId
- PipelineId
- CallingDataFactoryName
- ResourceGroupName
- DataFactoryName
- PipelineName
- StartDateTime
- PipelineStatus
- LastStatusCheckDateTime
- EndDateTime
- IsBlocked
- AdfPipelineRunId
- PipelineParamsUsed

ErrorLog (procfwk)
- LogId
- LocalExecutionId
- AdfPipelineRunId
- ActivityRunId
- ActivityName
- ActivityType
- ErrorCode
- ErrorType
- ErrorMessage

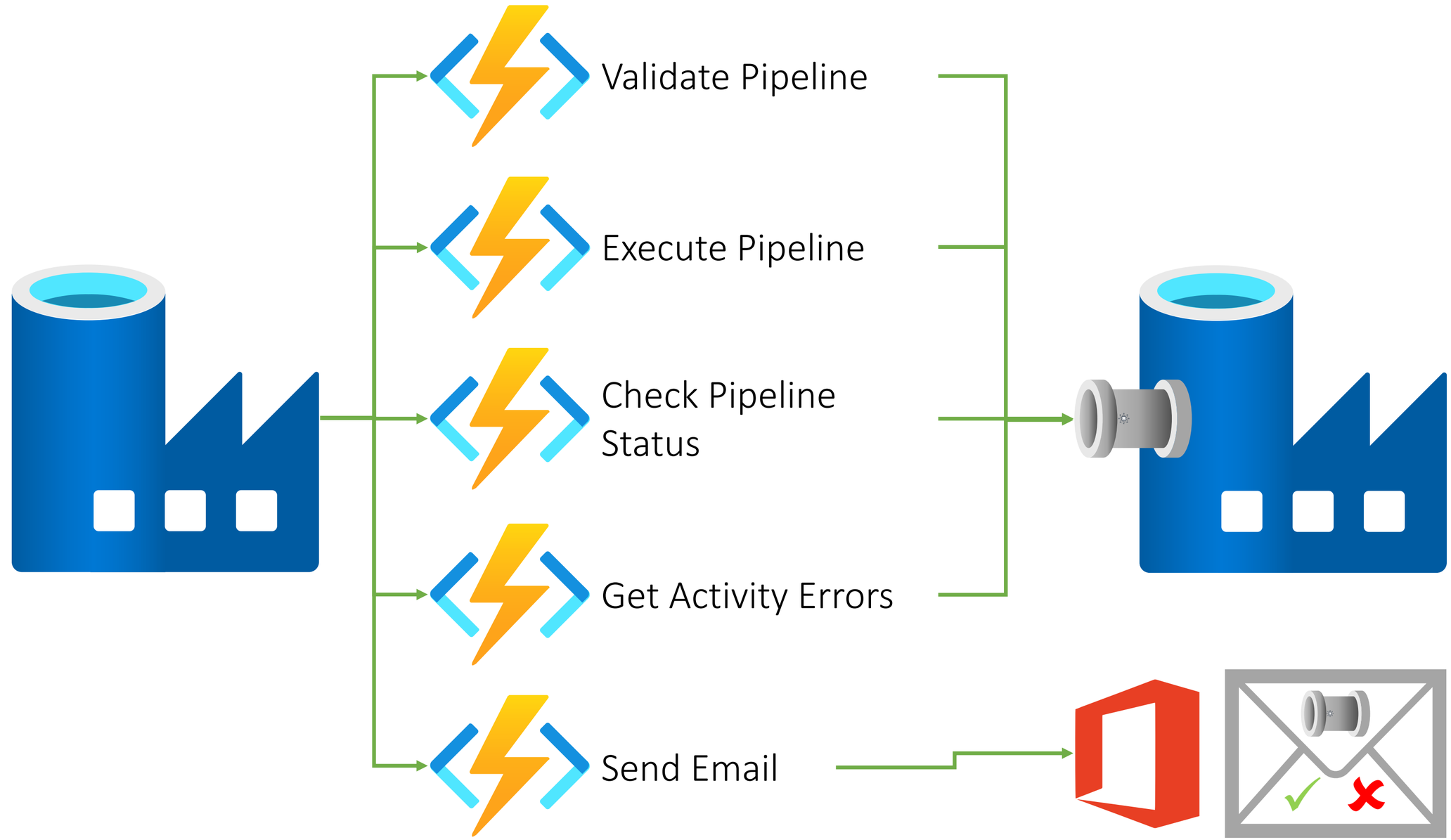
ProcFwk Tour – Functions



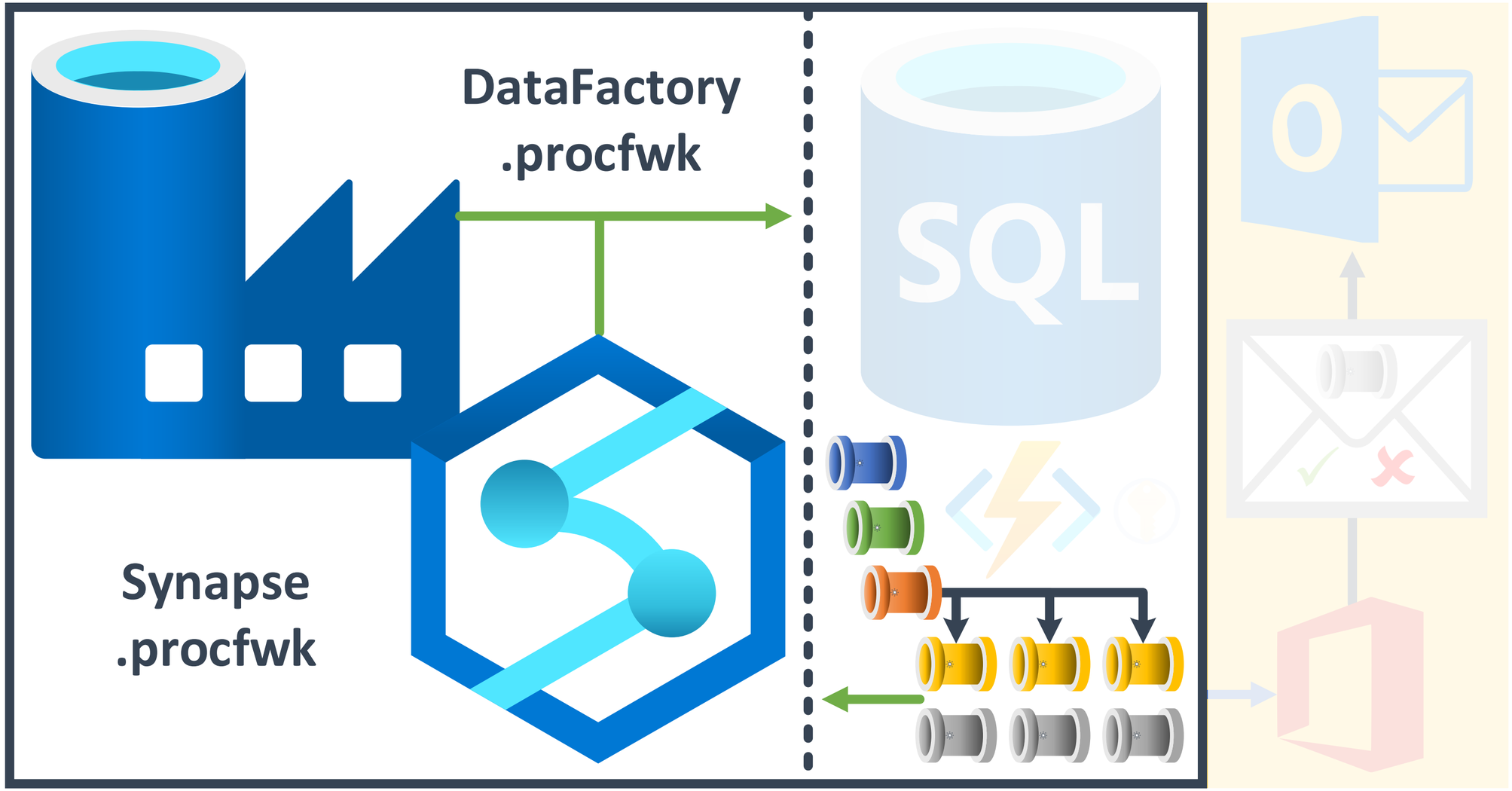
Functions Creation & Configuration



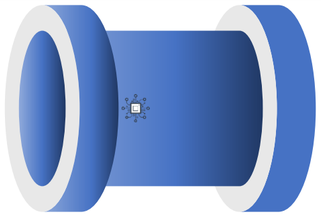
Function Roles



ProcFwk Tour – Integration Pipelines

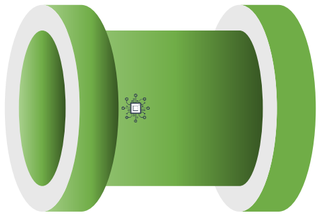


Framework Pipeline Hierarchy



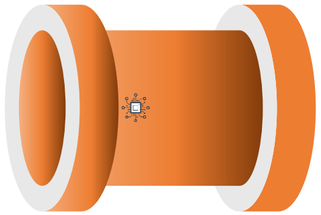
- Grandparent

Role: Optional level platform setup, for example, scale up/out compute services ready for the framework to run.



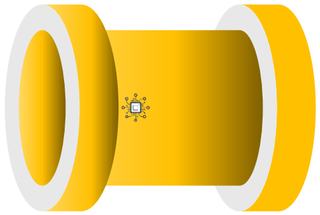
- Parent

Role: Execution run wrapper for batches and execution stage iterator.



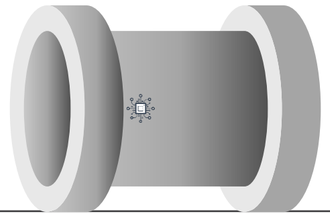
- Child

Role: Scale out triggering of worker pipelines within the execution stage(s).



- Infant

Role: Worker validator, executor, monitor and reporting of the outcome for the single worker pipeline.

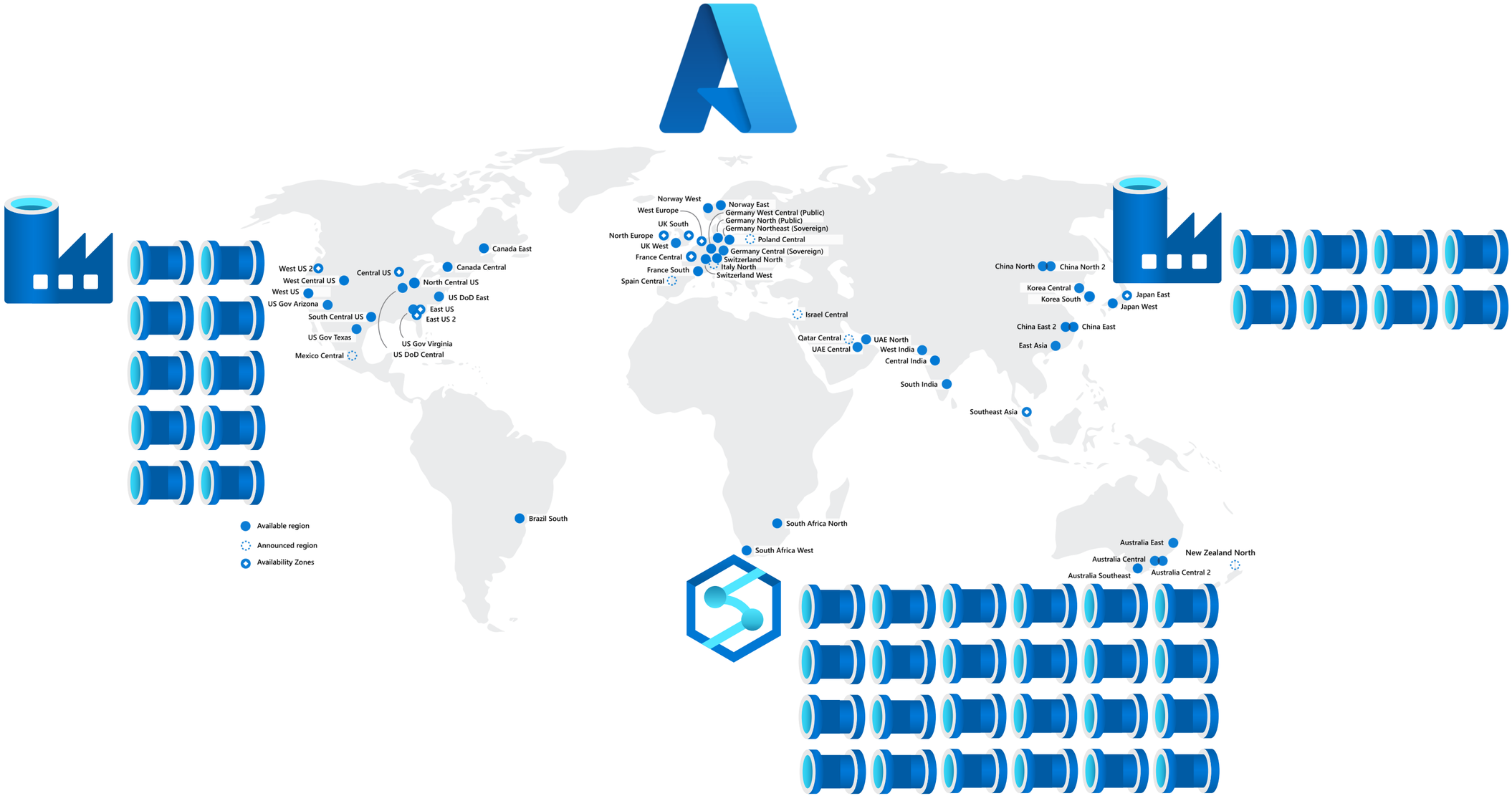


- Worker

Role: Anything specific to the process needing to be performed.

Problem

How should we structure and trigger our Integration Pipelines?

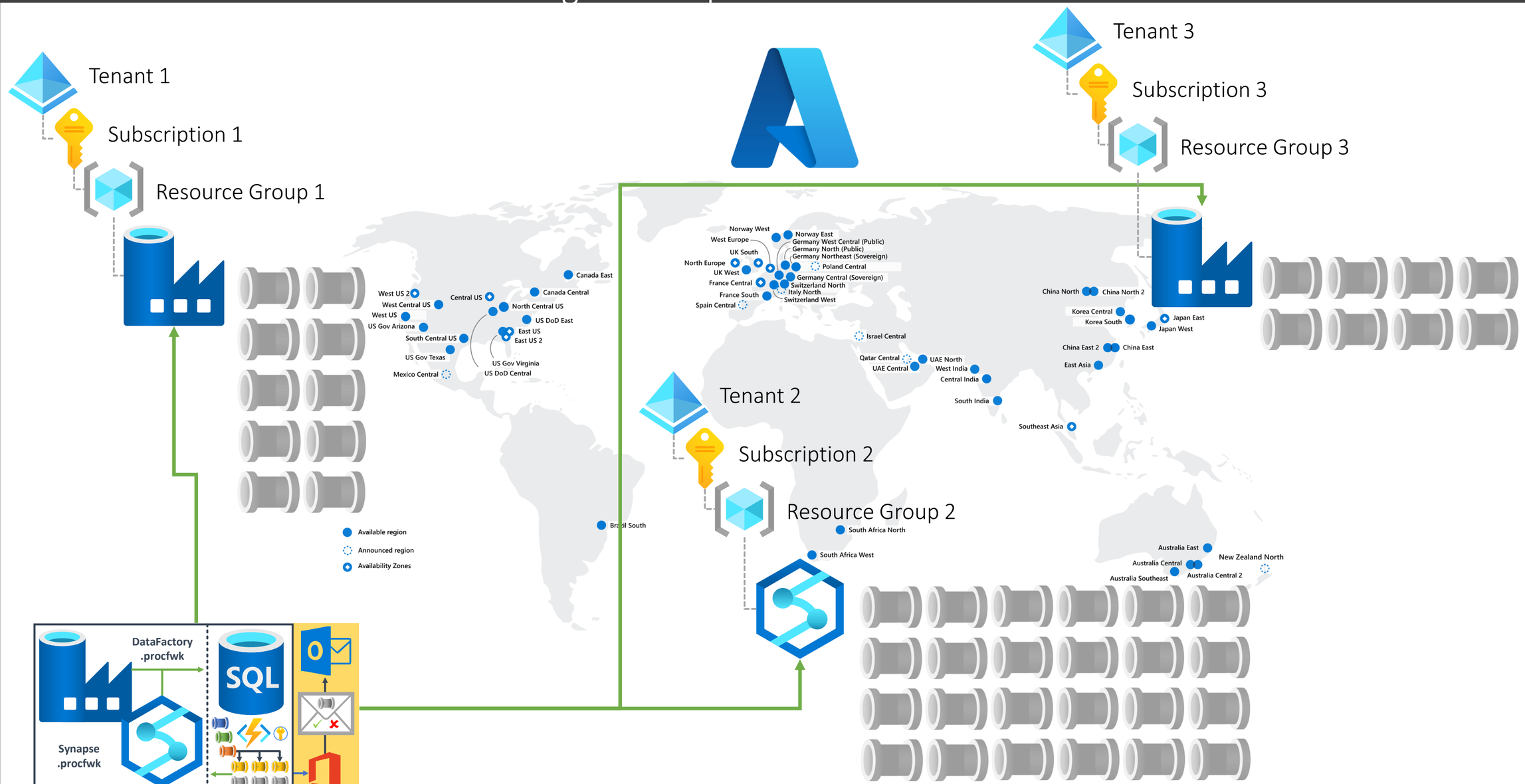


Solution

Use Metadata to Drive Integration Pipeline execution

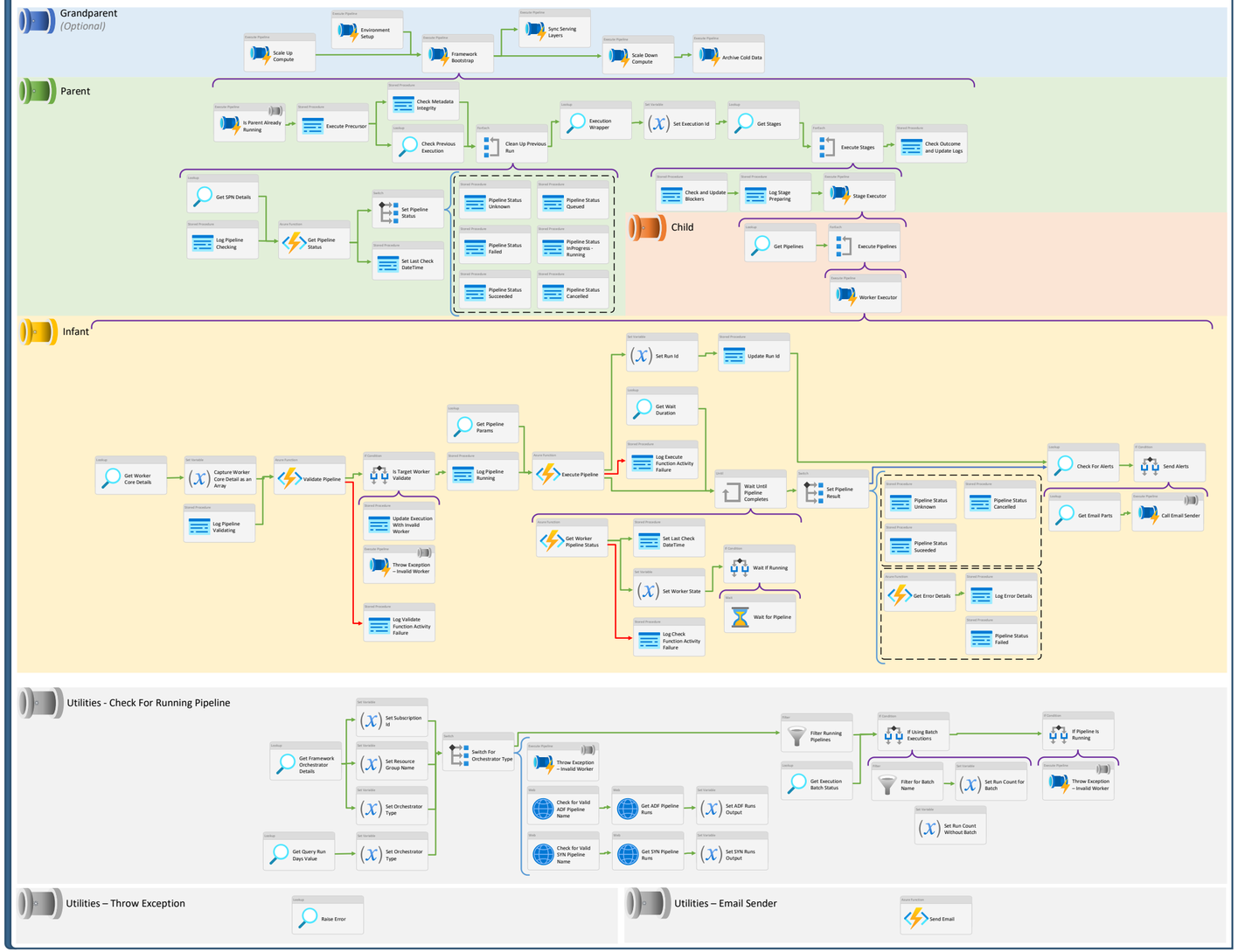
Solution

Use Metadata to Drive Integration Pipeline execution



ProcFwk.com Activity Chain

Orchestration Framework



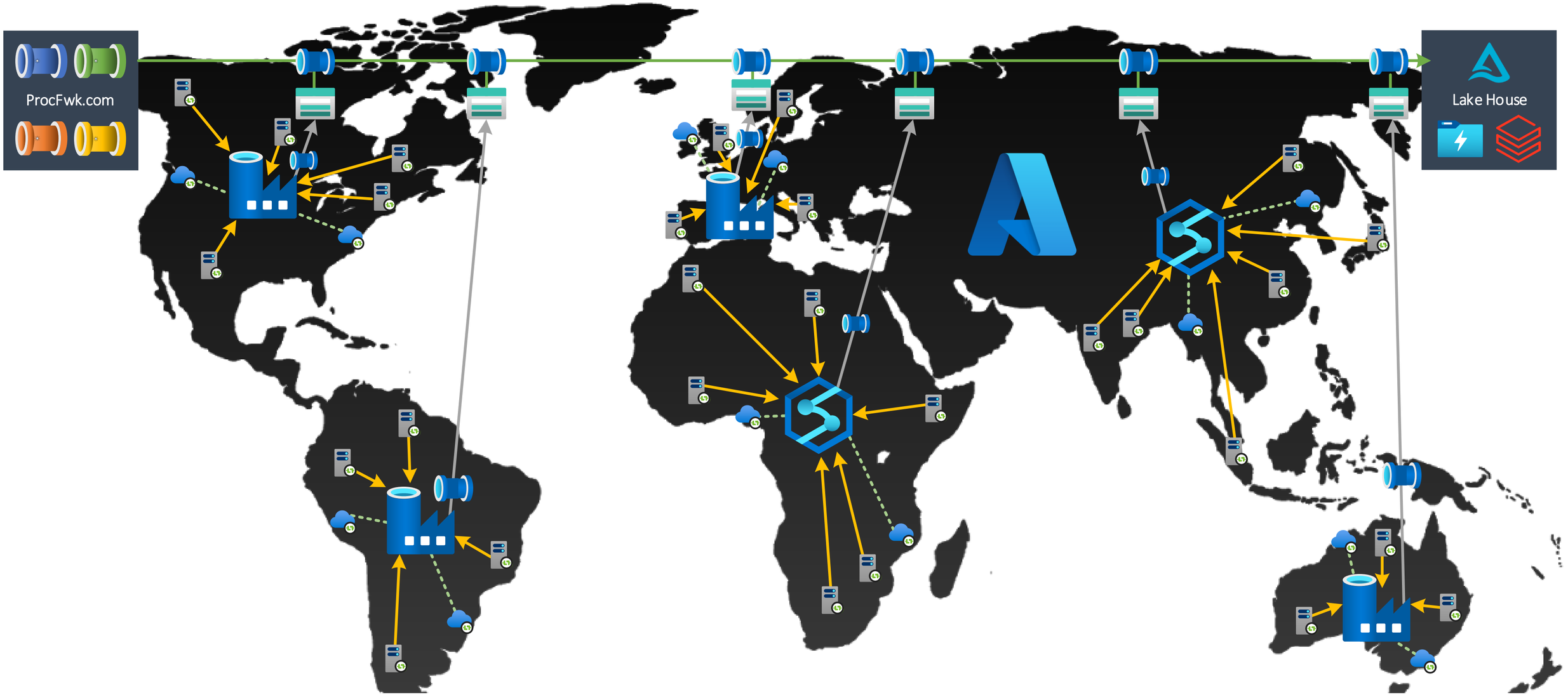
Worker Pipelines

- Worker 1 - Extract
- Worker 2 - Clean
- Worker 3 - Transform
- Worker 4 - Load
- Worker 5 - Serve
- Worker n - .....

Go to Visio file in GitHub:
github.com/mrpaulandrew/procfwk/blob/master/Images

Demo

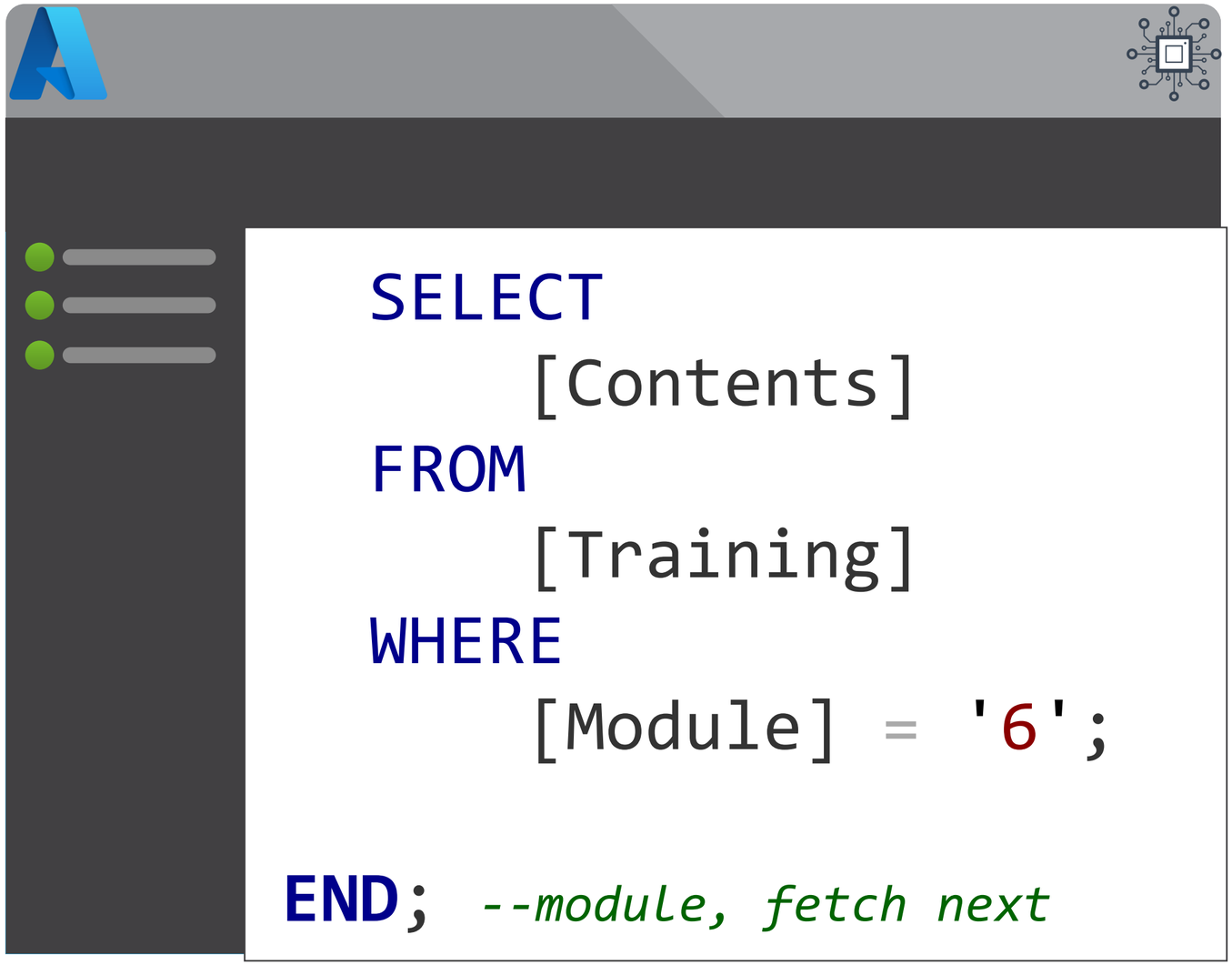
Hub & Spoke Integration Architecture



https://mrpaulandrew.com/2021/08/10/scaling-azure-data-integration-pipelines-decoupling-data-extract-and-transform/

Module 6

Execution Parallelism



```
    SELECT
        [Contents]
    FROM
        [Training]
    WHERE
        [Module] = '6';

END; --module, fetch next
```

- Control Flow Scale Out
- Concurrency Limitations
- Internal vs External Activities
- Orchestration Framework - http://procfwk.com