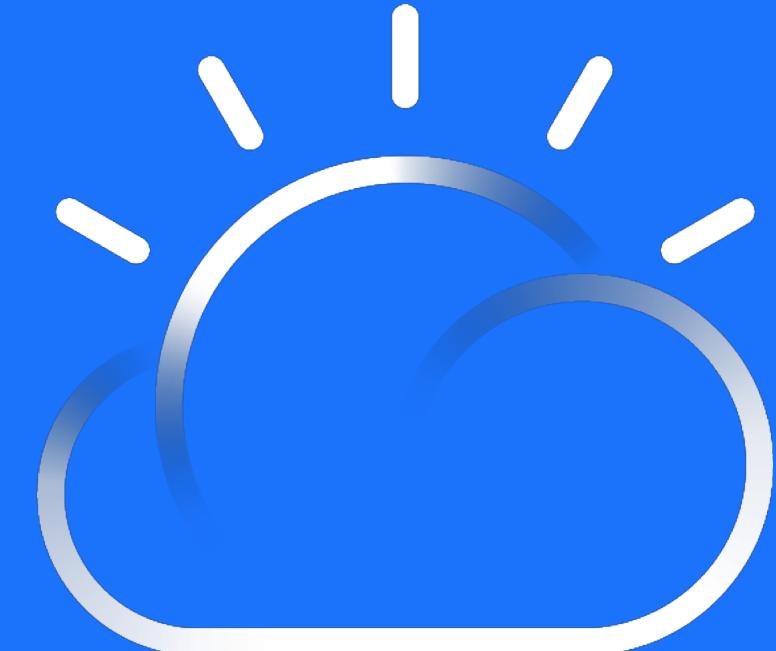


# A10: Batch Processing with IBM App Connect

Andy Garratt, IBM UK  
[andyg@uk.ibm.com](mailto:andyg@uk.ibm.com)



IBM Cloud

IBM

# Why ‘Batch Processing’?



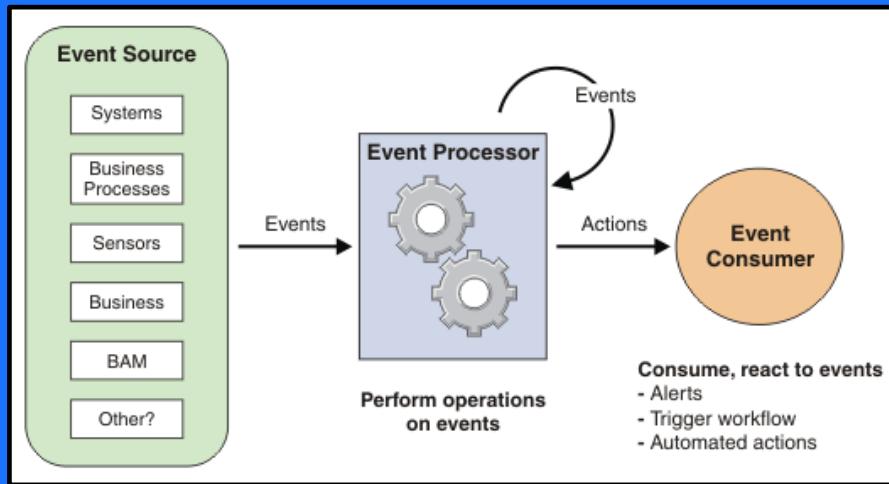
- **Initial Data Load:**  
Cutover/Go-live /Reset data.  
Used before event-driven processing starts.
- **Scheduled Data Sync**  
Control when the data is moved.  
Schedule for off-peak or when data is stable
- **Replication to cache/lakes**  
Use for read optimization or analytics  
Takes load off the SOR application

```
{ "timestamp": "2017-06-03T18:42:18.018", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "36"}, {"timestamp": "2017-06-03T18:43:335", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "7"}, {"timestamp": "2017-06-03T18:46:921.0", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "23"}, {"timestamp": "2017-06-03T18:42:18.018", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "36"}, {"timestamp": "2017-06-03T18:43:335.0", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "7"}, {"timestamp": "2017-06-03T18:46:921.0", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "23"}, {"timestamp": "2017-06-03T18:42:18.018", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "36"}, {"timestamp": "2017-06-03T18:43:335.0", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "7"}, {"timestamp": "2017-06-03T18:46:921.0", "class": "com.orgmanager.handlers.RequestHandler", "durationMillis": "23"}]
```

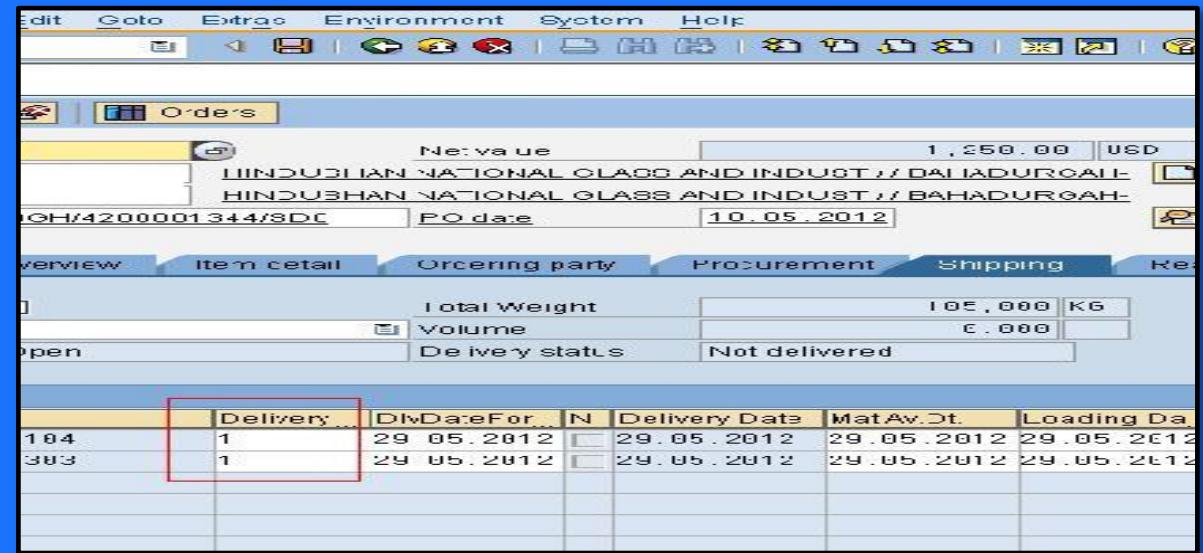
# 'Multi Item' Patterns



Application Sourced Events  
e.g. New Order Created



Parent/Child e.g. Orders/Items

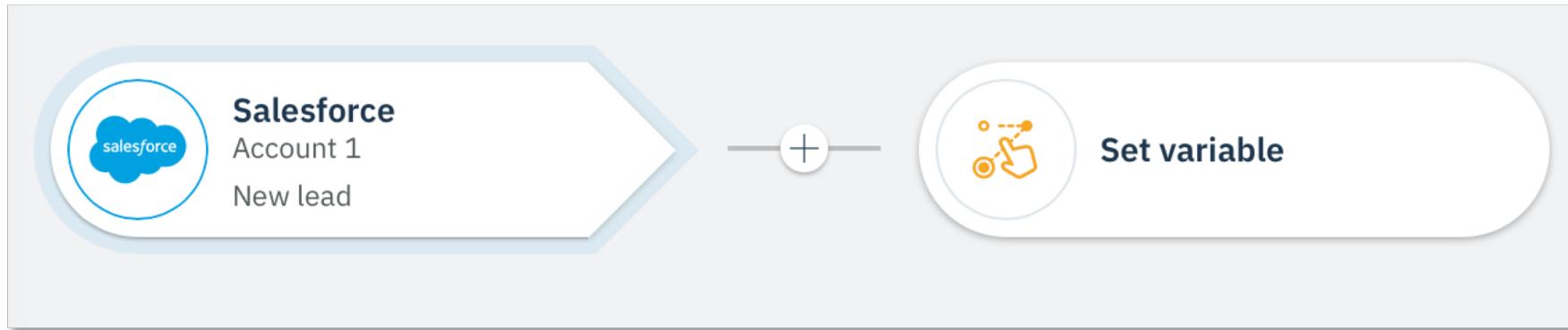


Collection of records e.g. payments

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	22488289/0	Order0001	2/7/2013	9	Payment requested	testoff	2/7/2013	Jenny Tester	100,00	EUR	CreditCard	VISA	XXXXXXXXXXXX1111		
2	22488290/0	Order0002	2/7/2013	9	Payment requested	testoff	2/7/2013	Paul McCardno	95,00	GBP	CreditCard	MasterCard	XXXXXXXXXXXX9999		
3	22488291/0	Order0003	2/7/2013	2	Authorisation declined		2/7/2013	Richard Starkey	20,50	EUR	CreditCard	VISA	XXXXXXXXXXXX1111		
4	22488292/0	Order0004	2/7/2013	9	Payment requested	testoff	2/7/2013	John Lemon	100,00	GBP	CreditCard	VISA	XXXXXXXXXXXX1111		
5	22488293/0	Order0005	2/7/2013	9	Payment requested	testoff	2/7/2013	Linda Eastman	95,00	EUR	CreditCard	MasterCard	XXXXXXXXXXXX9999		
6	22488294/0	Order0006	2/7/2013	2	Authorisation declined		2/7/2013	George Horizon	20,50	USD	CreditCard	VISA	XXXXXXXXXXXX1111		
7	22488295/0	Order0007	2/7/2013	9	Payment requested	testoff	2/7/2013	Yo Konono	100,00	EUR	CreditCard	VISA	XXXXXXXXXXXX1111		
8	22488296/0	Order0008	2/7/2013	9	Payment requested	testoff	2/7/2013	Lucy Diamonds	95,00	USD	CreditCard	MasterCard	XXXXXXXXXXXX9999		

# Application Sourced Events:

We don't treat this as a batch, no matter how many of them there are



- Smart Connectors detect the events: Then the flow runs
- One flow run per event (*counts as one flow run in your plan*)
- App Connect handles scaling when multiple flow runs are needed

# Parent-Children relationships (e.g. Orders with Multiple Products in the Order)

Order

Properties Operations

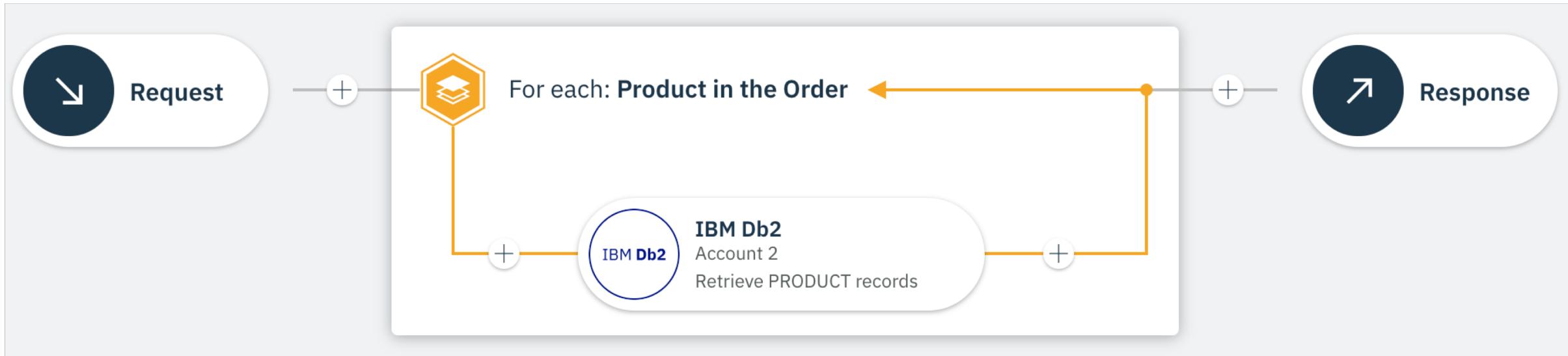
Add properties to your Order model

Property	Type	Operations
OrderNumber	Number	ID
OrderLines	Array of objects	○
LineNumber	Number	
Product	String	
Quantity	Number	
Price	Number	
+ Add property		
OrderTotal	Number	○

- E.g. We have an API to create an order.
- For each Product in the order, need to get the price from a database.
- **Need to send the whole order back as one piece of data, including the total.**

# Parent-Children (e.g. Pricing each Product in the Order)

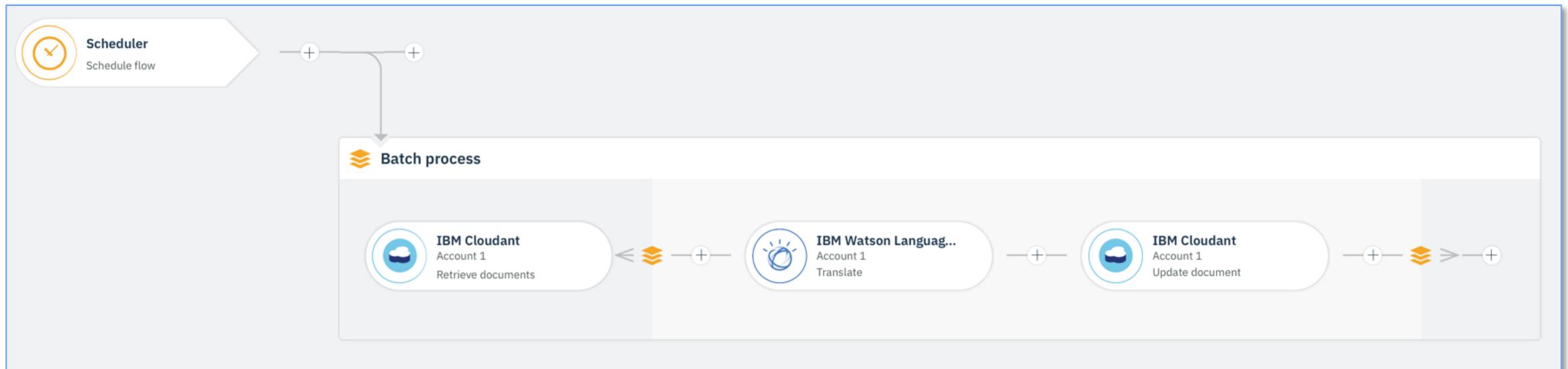
If we need to keep the whole object together in the flow:



- Use ‘For Each’
- Can process each Product in serial or parallel
- Creates an array of output items (one per Product) that can be used by the flow after the “For Each” finishes.

# A set of *Independent* Items/Records **Retrieved** from an Application

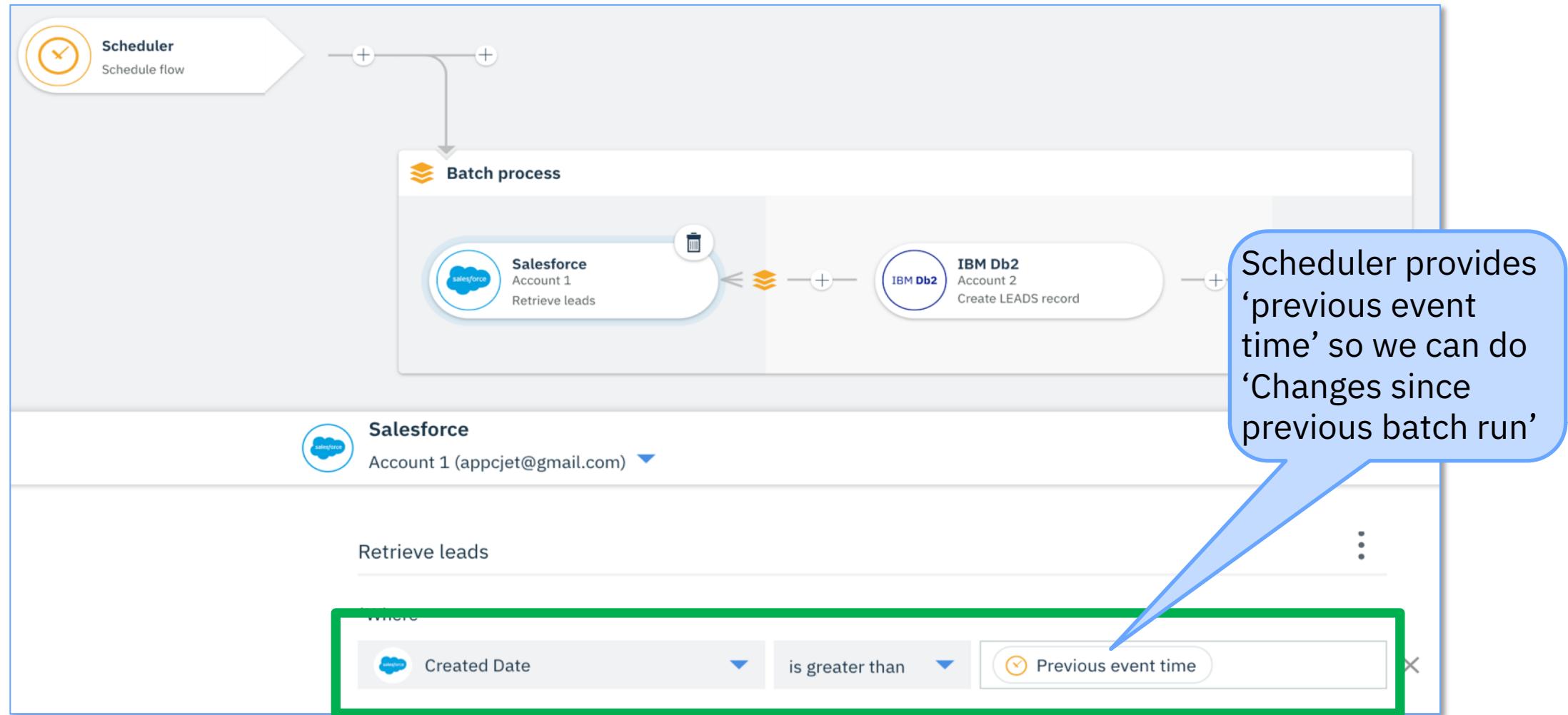
## Use Batch Processing



- The batch process manages the retrieval and processing of the data
- Each item is processed independently in batch – in parallel
- Must *not* need to use aggregated results (e.g. Order Totals) later in the flow

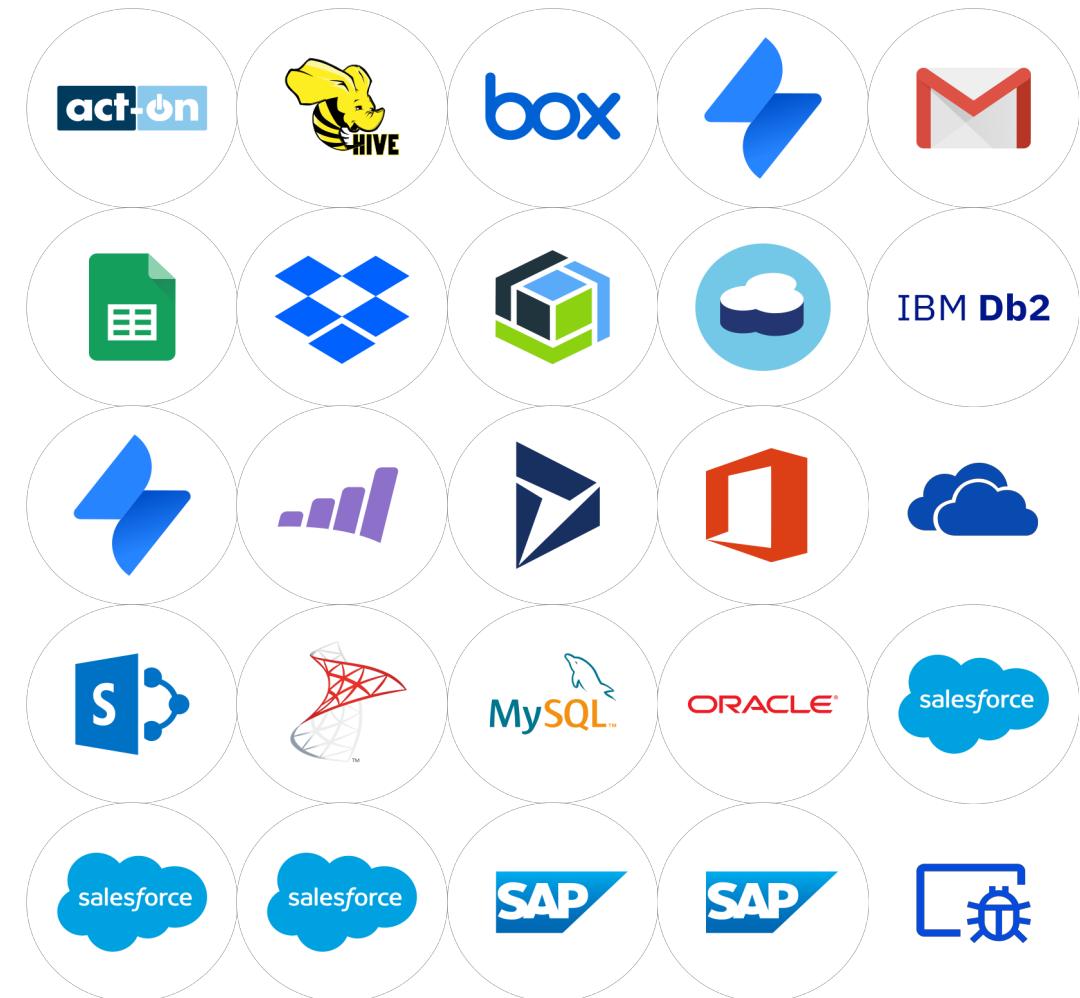
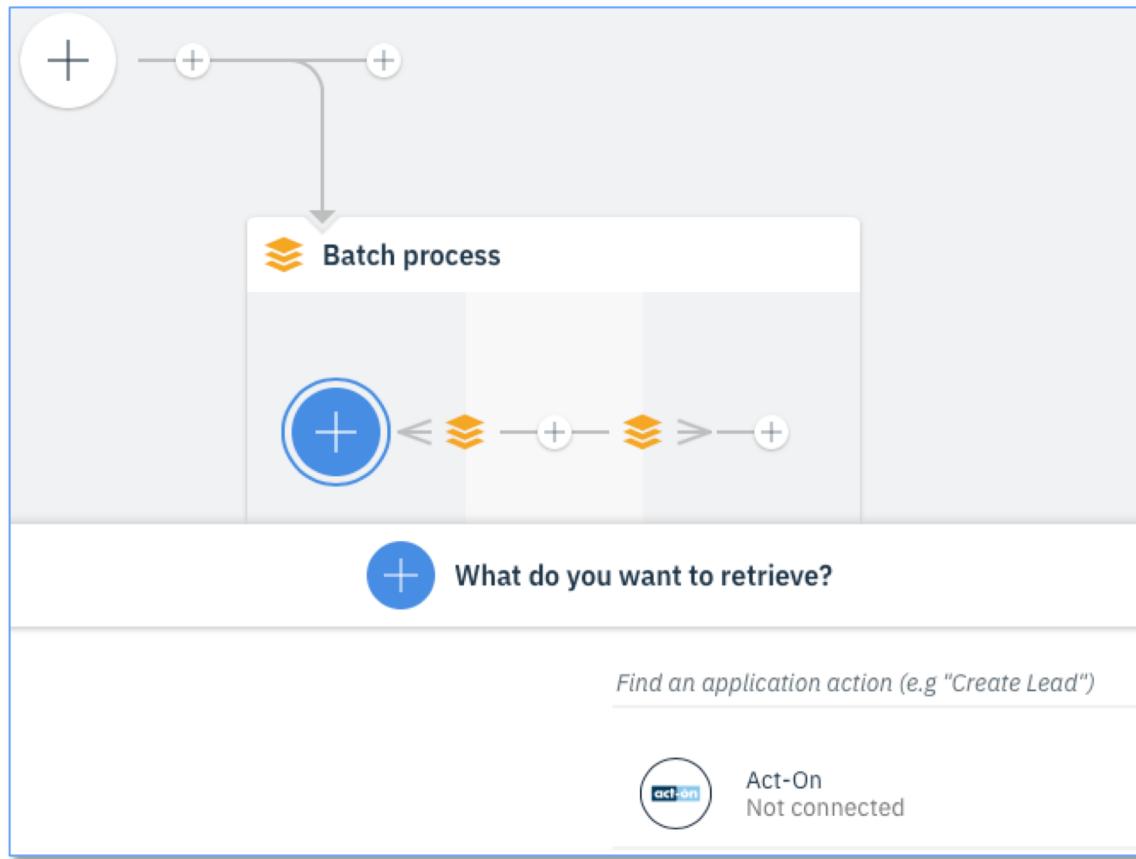
# Use Batch Processing to control *when* data sync happens

e.g. Schedule at 2am to keep the load out-of-hours



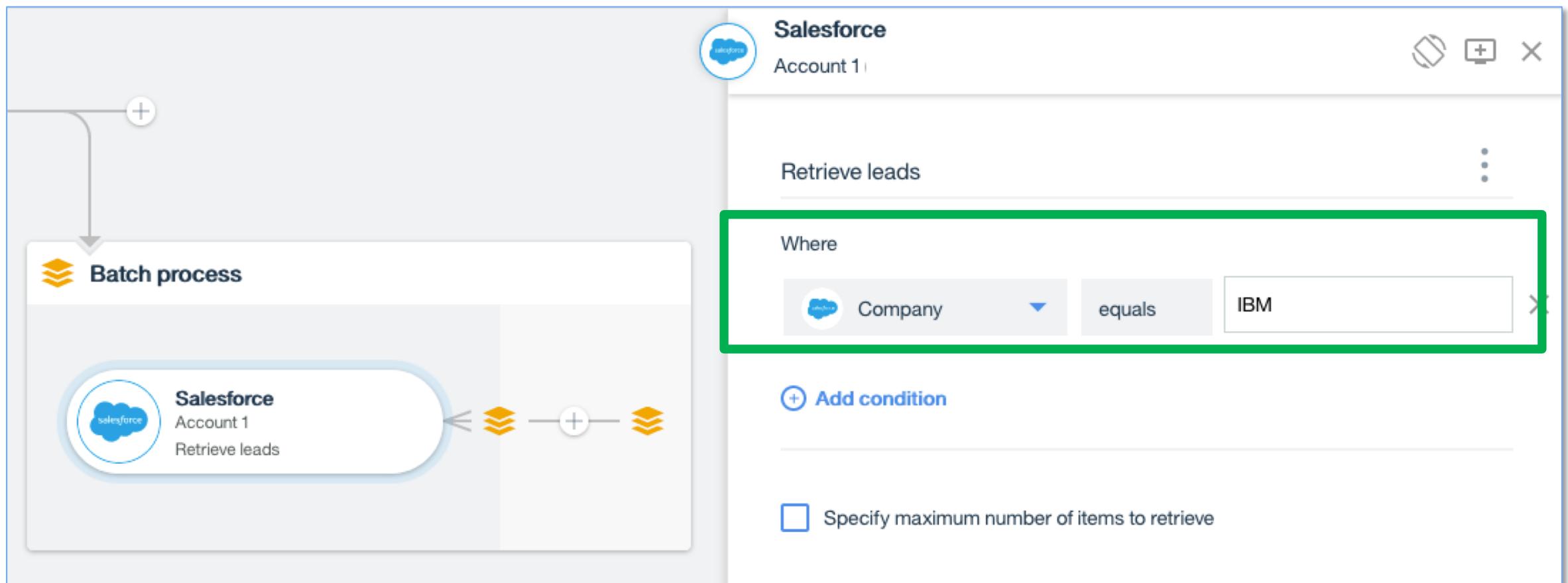
# Currently 25 connectors support the ‘Retrieve Data for Batch’

## We’re adding more and more connectors with batch support



# Different connectors have different filters:

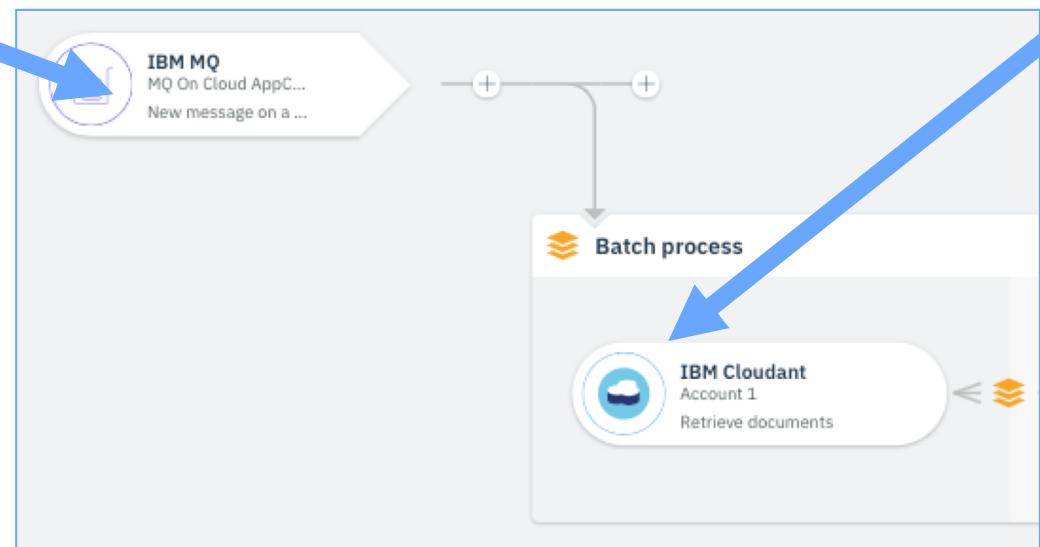
Using Salesforce to filter leads by company gives a batch result set just for that company



# Caution: Don't confuse 'batch' with Application Events

## Inbound Application events

- Events processed as they occur
- Have the 'event' node shape
- Number of events determined by source system
- Each event is a new flow
- App Connect sees each event as completely independent of others
- There is no first-class concept of the 'batch'

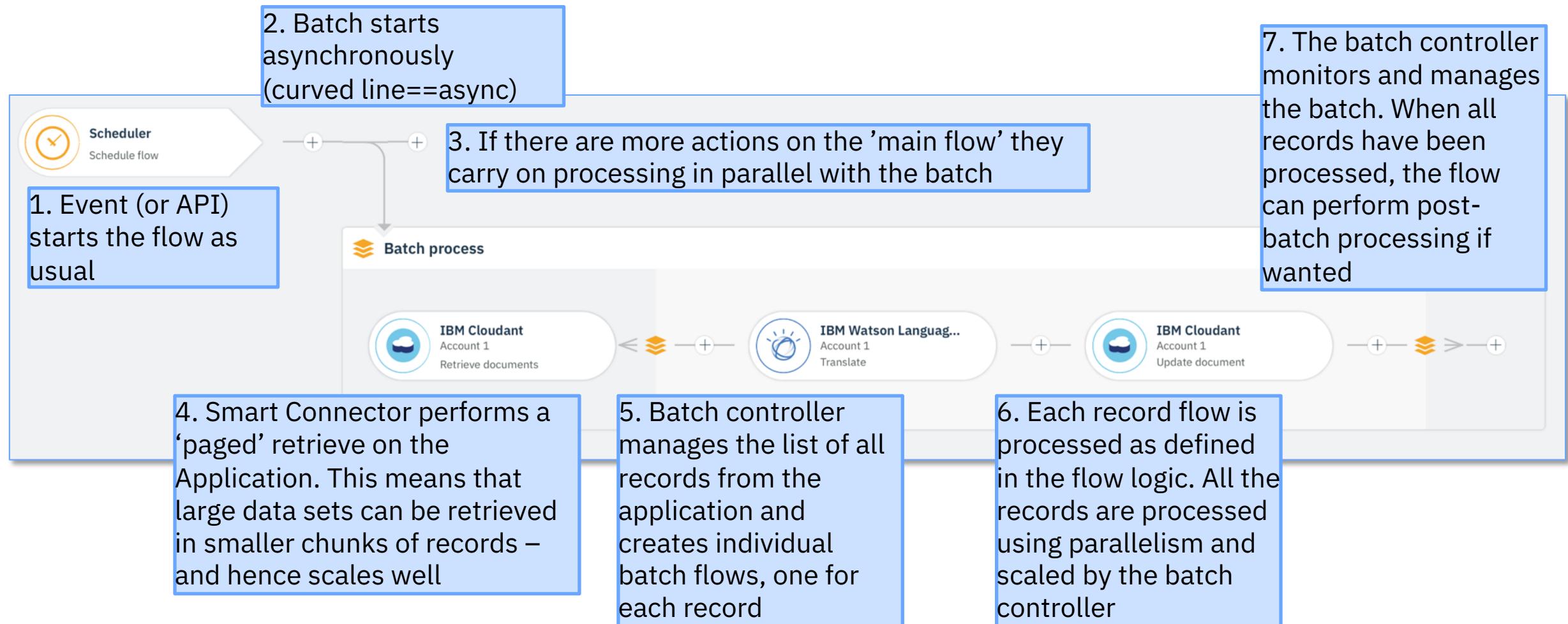


## Batch

- Data processed when the batch runs – e.g. can be scheduled
- Has the 'action' node shape
- Number of events determined by query from batch engine
- All batch items are processed in one 'flow run'.
- Batch events are processed independently but batch totals etc are managed by App Connect
- First class concept of 'the batch'

# How does ‘Batch’ work?

## What is the sequence of operations in the flow?

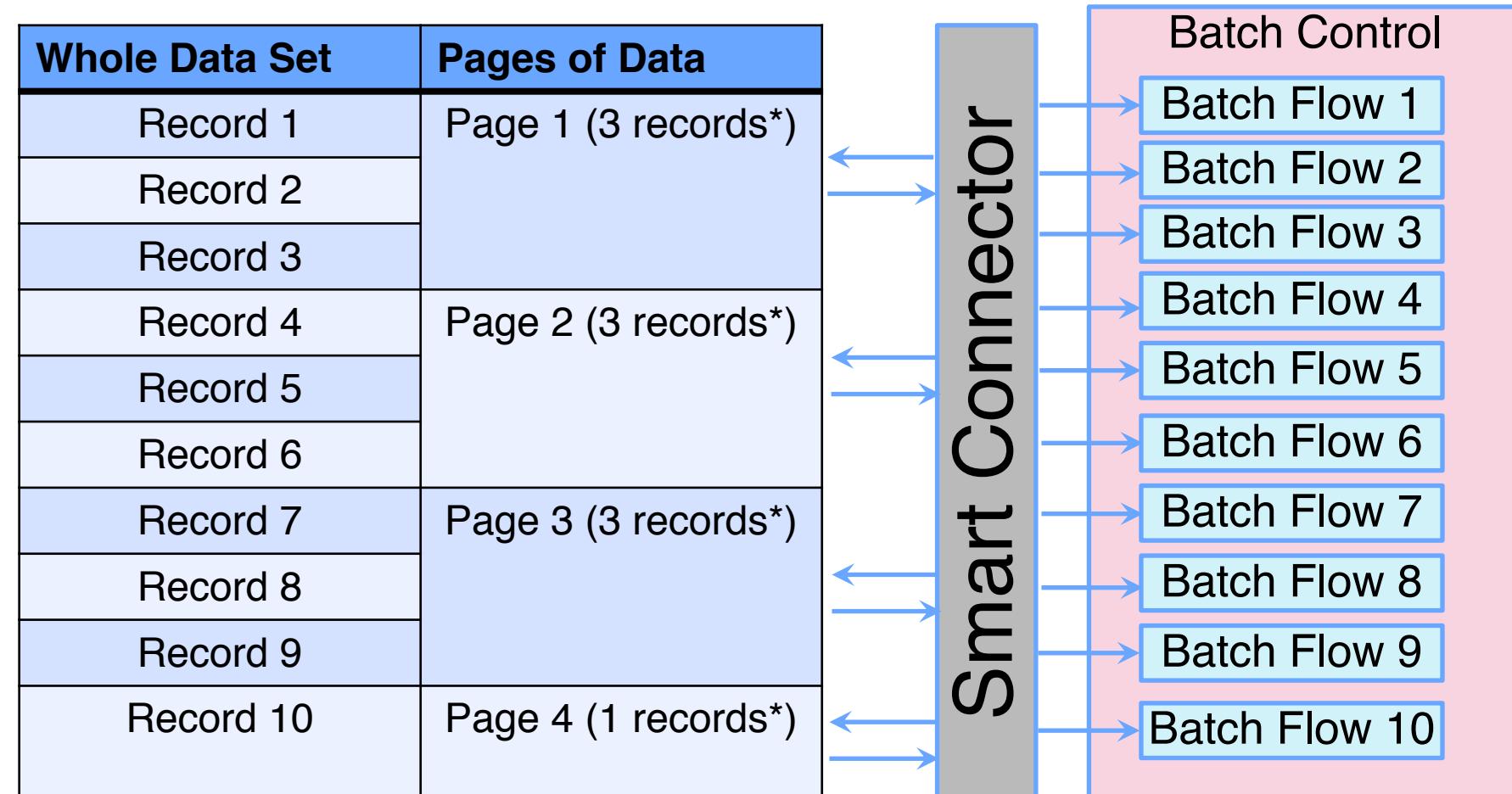


# How does the connector retrieve the data from the application in ‘Batch Mode’?

The Smart Connector is aware of how the application allows ‘paged’ data retrieval. It retrieves the entire data in smaller ‘chunks’.

The size of these ‘chunks’ is managed by the Smart Connector. (The designer does not need to worry about this.)

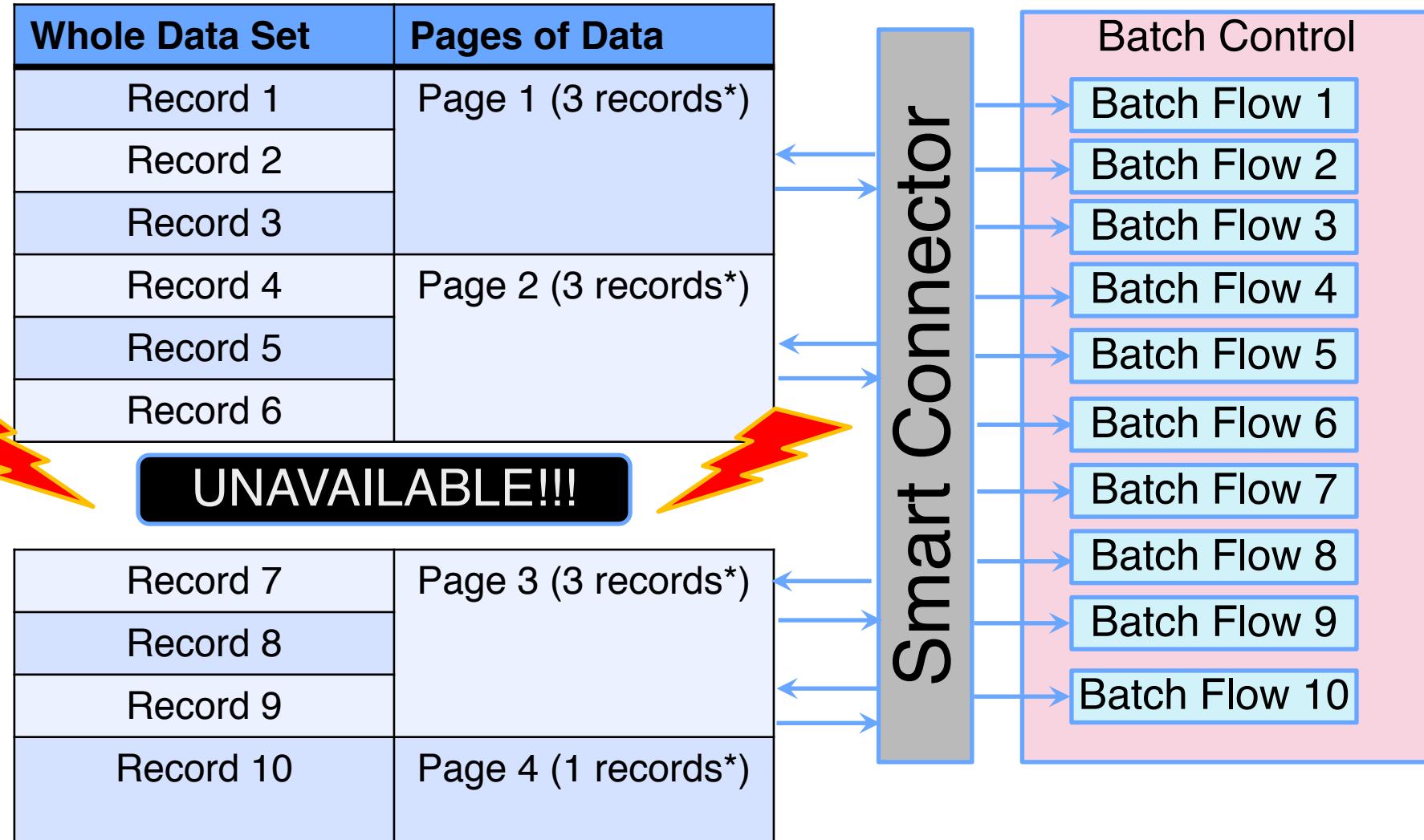
The Smart Connector and Batch Controller then take the individual records/items in those chunks and send them to the batch processing engine.



(\*Pages are normally a lot bigger than the 3 records in our example!)

# Batch processing manages any issues when retrieving data:

If there is an issue,  
the batch  
controller  
'remembers' which  
pages it has  
retrieved with  
Smart Cursors:  

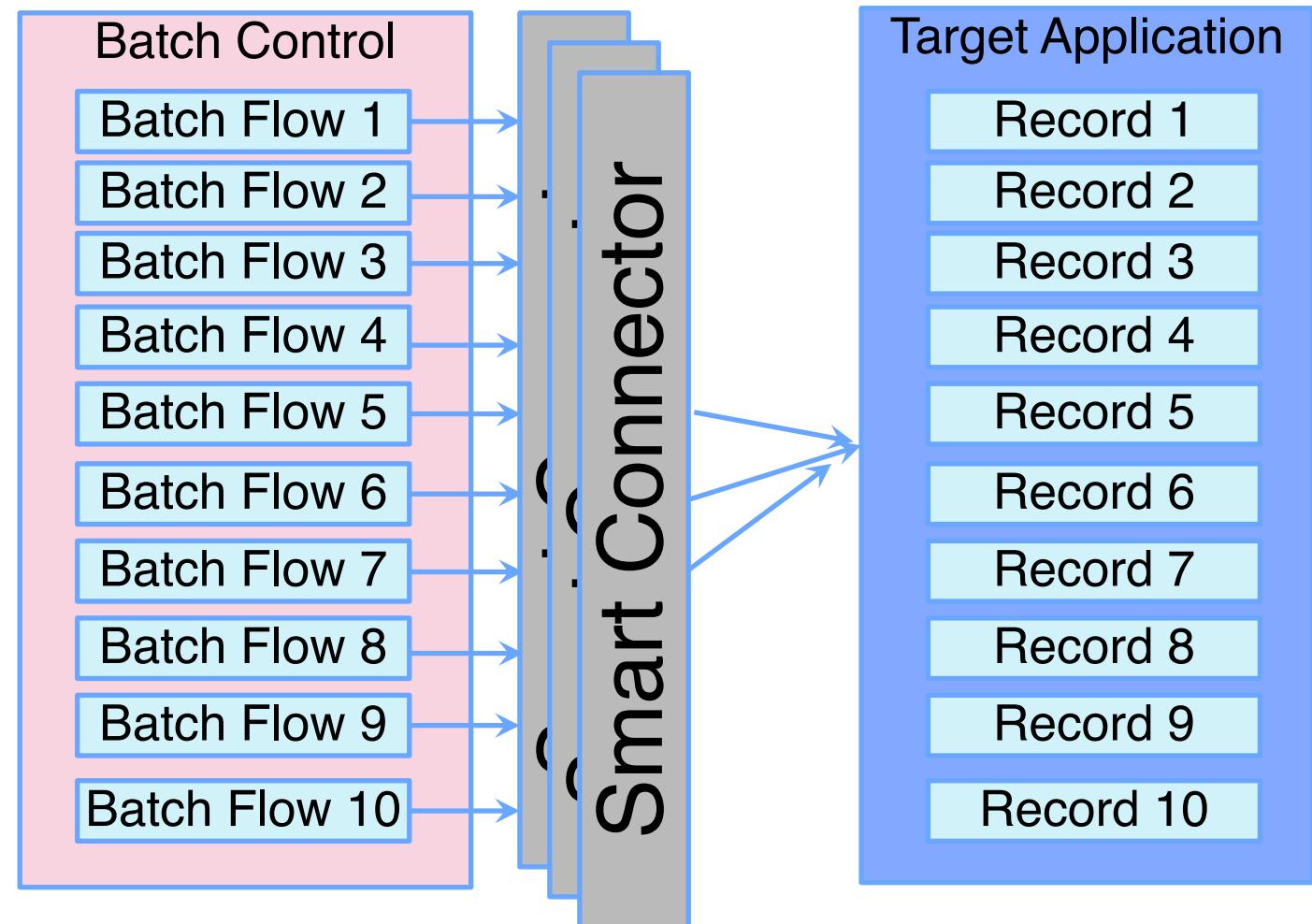



and re-starts  
where it left off:  
Automatically.

# How do we write the data from the batches?

(Our example shows one target application – you can of course have many)

- Usual Smart Connector Operation:  
Retries and error handling etc
- Still need to consider outbound data caps (if relevant)
- Scales if the target app allows for it
- Smart throttling depending on response from target system e.g. HTTP



# Some Characteristics of ‘For Each’:

If these affect your flow design, you may need to use batch processing

## Processing Time

- ForEach runs *within* the flow
- The entire flow (with all records) must complete within the 60 second flow limit

## Memory

- The entire flow has 100mb of memory
- This is shared out between the records in the For Each

## Retrieve

- A retrieve in a non-batch flow is restricted to 1000 records
- All the records retrieved must fit in the 100mb flow (along with everything else)

# Comparison between Batch and For Each:

ForEach	Batch
Operates on data already in the flow	Manages the retrieval of the batch data from the source
Restricted to 60 seconds flow time	Can run for 30 days in total. (Each batch record must run in 60 seconds though)
Allows creation of an output result set which the flow can act on	Each batch record is independent – only aggregates available are batch statistics
Can work over any object/dataset in the parent flow, retrieves, set variables, mappings . .	Uses the ‘retrieve’ data from the Smart Connector which is doing the batch retrieve in the batch async.
Retrieve limited to 1000 records	Retrieve uses Smart Connector paging – no published limits on record numbers
100mb total for all record flows	Each batch record has independent 1mb memory
Can process records serially or in parallel	Parallel only
No visibility of processing of individual records	Batch monitoring of record statistics, errors etc

# Monitoring during the batch process:

Use the dashboard to see the batch status, both in real-time and after it has run

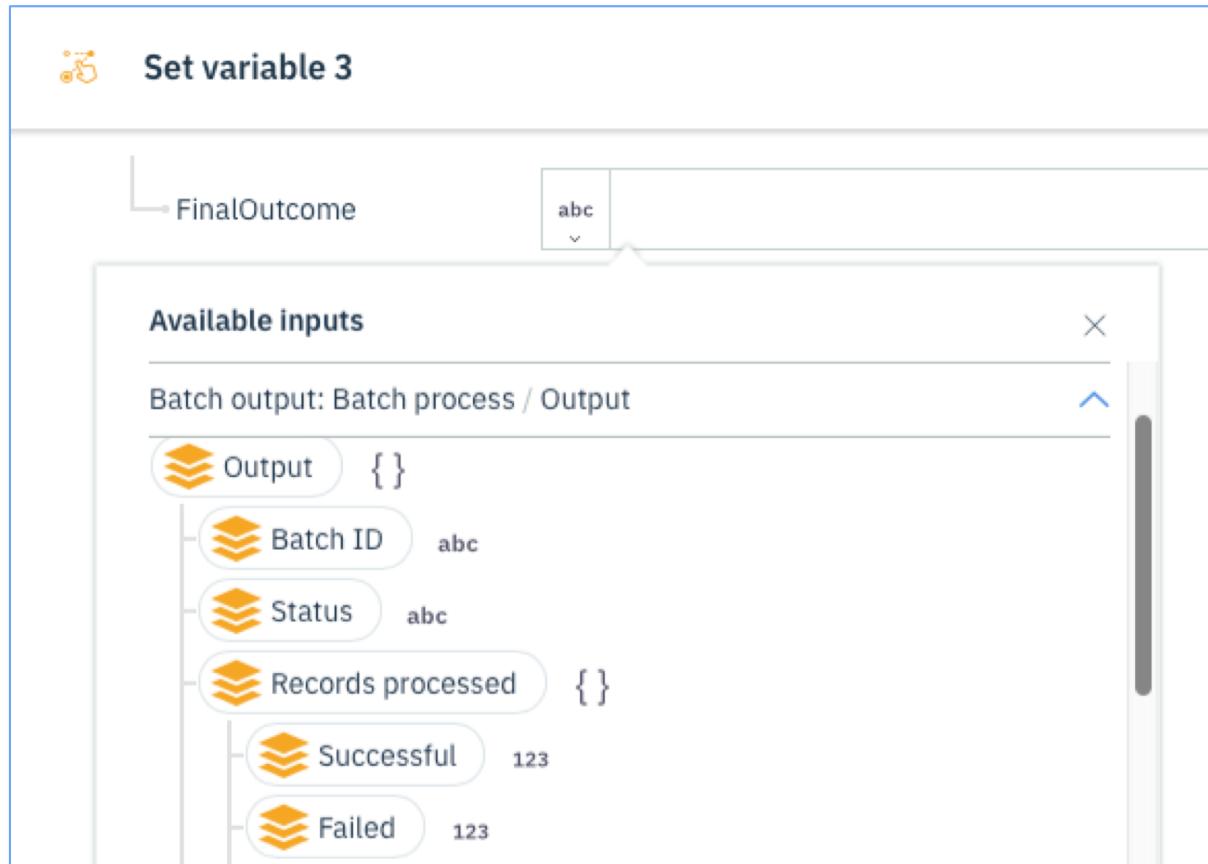
The image shows two screenshots of the Tramino application interface. The left screenshot displays a sidebar with various options: Stop, View, View errors, View batches (which is highlighted with a green box), Enable debug logging, Export, and Delete. The right screenshot shows a 'Batch status' table with the following data:

Name	Status	Start time	End time	Expiry time	Retrieved	Processed	Succeeded	Failed	Canceled	Actions
Batch process	Completed	00:00:08, Apr 1	00:00:29, Apr 1	-	1002	1002	0	1002	0	...
Batch process	Completed	23:00:08, Mar 31	23:01:15, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	22:00:08, Mar 31	22:00:59, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	21:00:08, Mar 31	21:00:54, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	20:00:08, Mar 31	20:01:21, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	19:00:08, Mar 31	19:01:17, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	18:00:08, Mar 31	18:01:17, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	17:00:08, Mar 31	17:01:21, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	16:00:08, Mar 31	16:01:19, Mar 31	-	1002	1002	0	1002	0	...
Batch process	Completed	15:00:08, Mar 31	15:01:15, Mar 31	-	1002	1002	0	1002	0	...

At the bottom right of the table is a 'Close' button.

During and after the batch, we have the batch status available

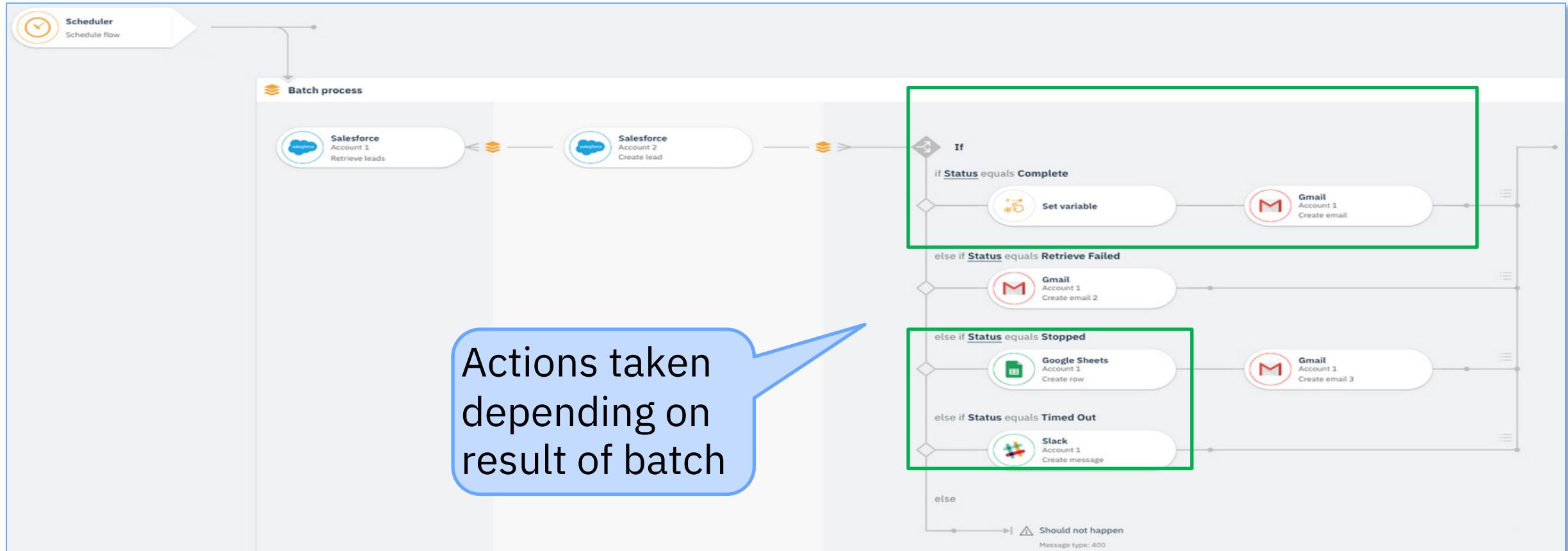
Can see this in the dashboard or use the data in the flow



### Inbound events

- Number processed
- Number failed
- Number successful
- Number Cancelled
  
- Start Time
- End Time

# After the batch runs, you can use the flow to take actions depending on the batch status.



Errors are grouped for batch:  
Use the dashboard to see which errors ‘belong’ to which batch.

The screenshot shows the IBM MQ to 2 applications dashboard. On the left, there's a sidebar with various buttons: 'Stop', 'Batch Close', 'View', 'View errors' (which is highlighted with a green box), 'View batches', 'Enable debug logging', 'Export', and 'Delete'. Below these is a section labeled 'IBM MQ to 2 applications'. On the right, there's a table titled 'Error messages' with columns for 'Timestamp', 'Message', 'User Action', and 'Actions'. The 'Actions' column is highlighted with a green box. A blue arrow points from the 'View errors' button on the left to the 'Actions' column on the right, indicating that clicking 'View errors' will lead to this detailed error list.

Timestamp	Message	User Action	Actions
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...

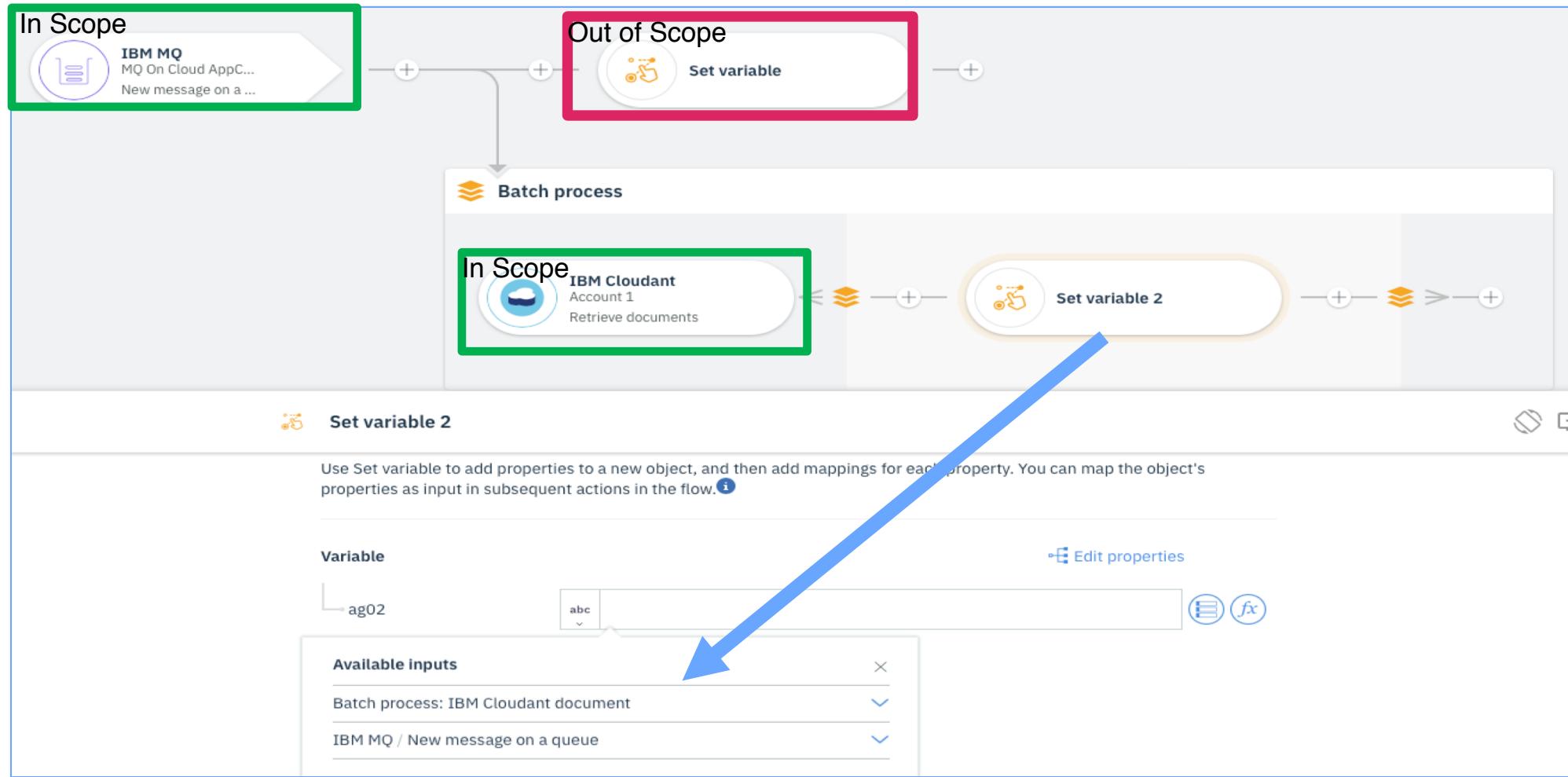
# Logs are automatically filtered by transactionID

(or you can use the transactionIDs to filter what you want)

Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	...
Apr 1, 2019 12:00:29 AM	The IBM Db2 Create ATMS record action failed to create an object.	Check for other errors, which might contain more information about the cause of the problem in the application.	<a href="#">View logs</a>

The screenshot shows the IBM Cloud Logging Kibana interface. At the top, there's a navigation bar with tabs for 'IBM Cloud Logging', 'Kibana 5', and 'Admin'. On the right, it shows the user's email 'andy.garratt@uk.ibm.com' and account details 'Andy Garratt's Account | IBM Cloud UKI | AGDev'. Below the navigation, there's a search bar with the query 'component\_str:AppConnect AND instance\_str:lqdmmayod AND transaction-id\_str:"453d9721-2a68-41a6-8f8f-04ef62711a8d/679f75ba-552c-4130-9333-3b65eeccc39c/773"'. A green box highlights this search bar. A blue arrow points from this green box down to a blue box around the 'Batch Number is' field in the bottom left. Another blue arrow points from the same green box up to another green box highlighting the 'View logs' link in the top right of the log table. The main area displays log entries with columns for timestamp, log message, and additional metadata like 'tail\_str' and 'loglevel\_str'. A blue box highlights the 'Batch record ID' field, and another blue box highlights the 'Batch Number is' field, both of which have dropdown menus open. A large blue box at the bottom contains the text: 'You can define this ID yourself in the batch flow - (max 256 chars)'. The bottom left corner shows the 'ALCH\_TENANT\_ID' field.

# Scopes in Batch Processing



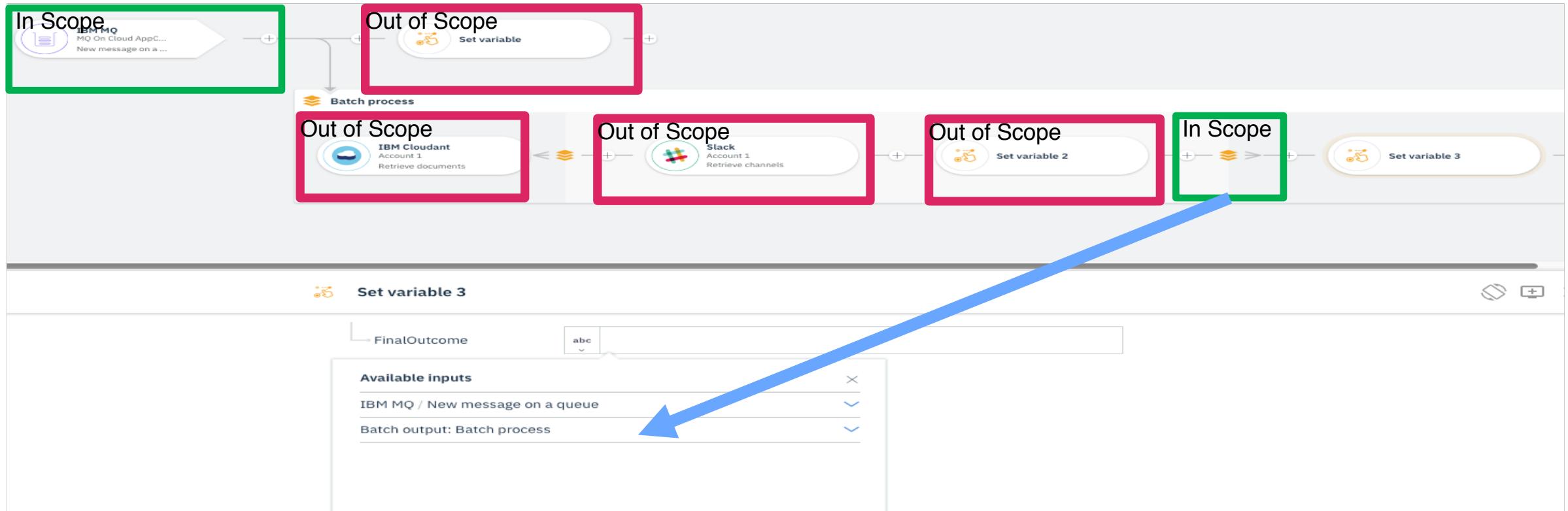
Batch is Async  
(shown by the curved line)

The rest of the flow runs as normal after the batch is started

The flow cannot 'see' the batch data.

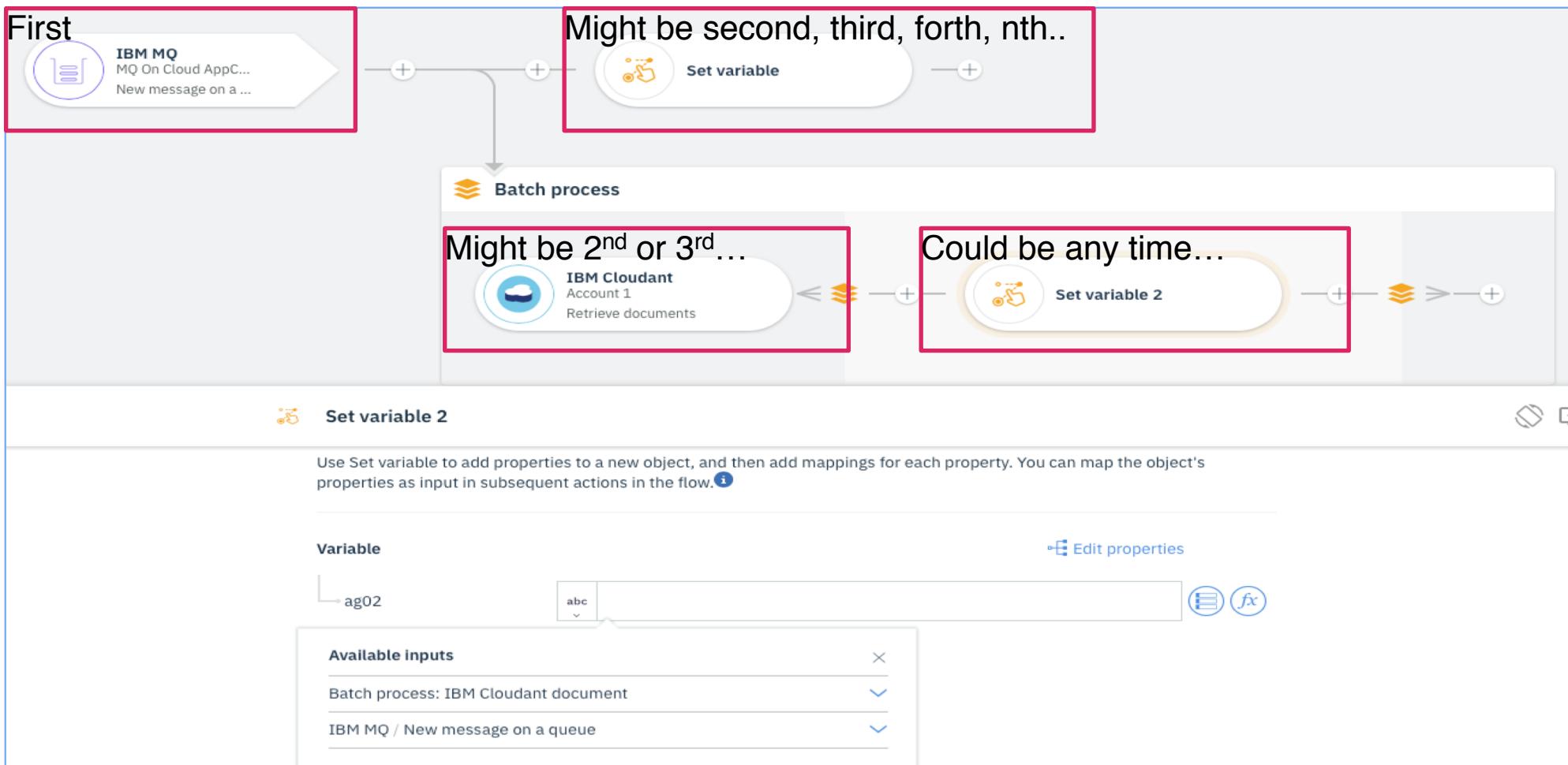
The batch record flow cannot 'see' the rest of the flow data

# The ‘After the batch completes’ flow context:



You can see ‘before batch’ flow context and the batch statistics.  
You **cannot** see data within the batch records themselves.

# Do NOT rely on a diagrammed ‘sequence’ of node execution:

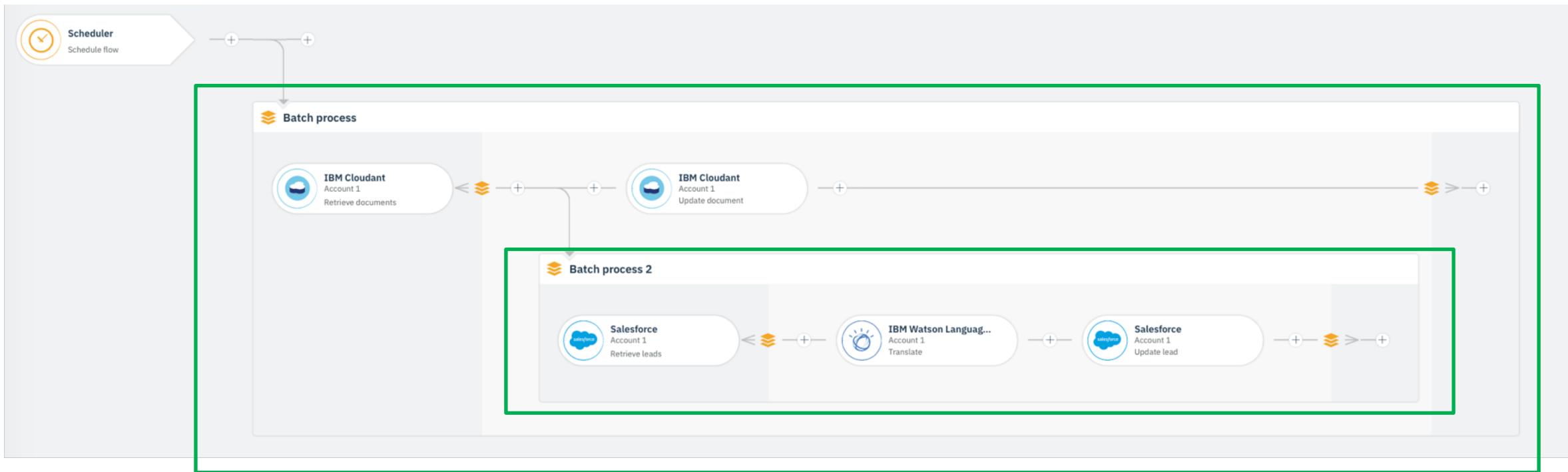


The batch run  
is async from  
the rest of  
the flow

Each batch  
record is run  
async in  
parallel too

# Batches within batches? No problem!

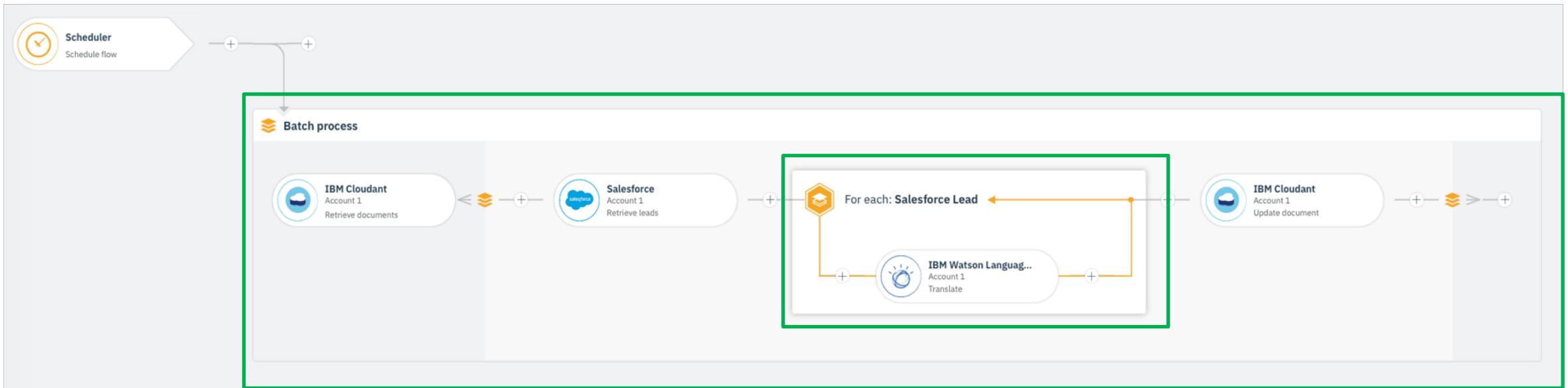
(And batch in batch in batch . . .)



- Take a list of accounts in a database:
- For every account, read all the leads in Salesforce and update them
- Then update the DB to say you did it.

# For Each within Batches? Batches within For Each?

You can do that too...



- Be aware of the processing time and async patterns
- Be aware of your memory
- Know the different ‘Retrieve’ behaviours

# Batch can handle *very* large amounts of data

## Each batch run only counts as one flow run!

No limit on number of batch records

No limit on retrieve action size (usual retrieve has limit of 1000 records) – but check each individual record size!

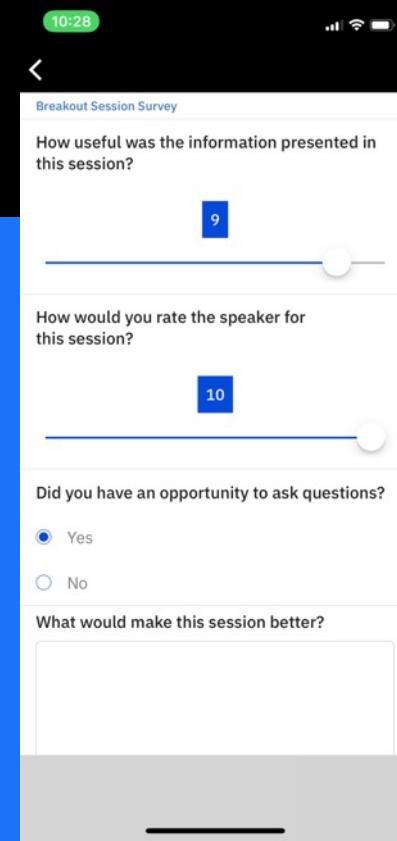
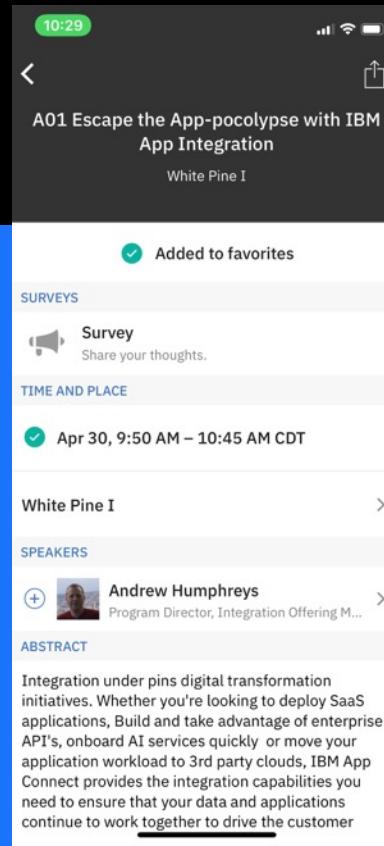
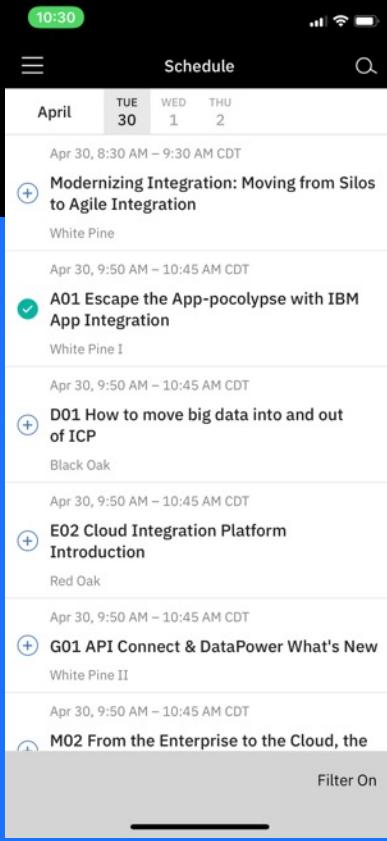
“Parent Flow” still has the 100mb limit and 60 seconds execution

All data usage counts towards your Outbound data limits

Total Batch Process duration cannot exceed 30 days (some source systems can take a long time to send large numbers of records e.g. web crawling)

Total memory **for each batch record** is 1mb – no ‘aggregated’ total of all records like in For Each

‘Lite plan’ limits the number of batches (6)



IBM

# Don't forget to fill out the survey!

Select your session, select survey, rate the session (A10) and submit!

Thank You

