Microsoft Power Platform

Best Practices in Building Power Automate Cloud Flows

Rasika Chaudhary

Principal Program Manager
Power CAT, Microsoft

Reza Dorrani

Principal Program Manager
Power CAT, Microsoft

Who are we?

Power CAT

Power Customer Advisory Team (Power CAT)
A strategic function of Product Engineering at Microsoft

Rasika Chaudhary

Principal Program Manager
Power CAT

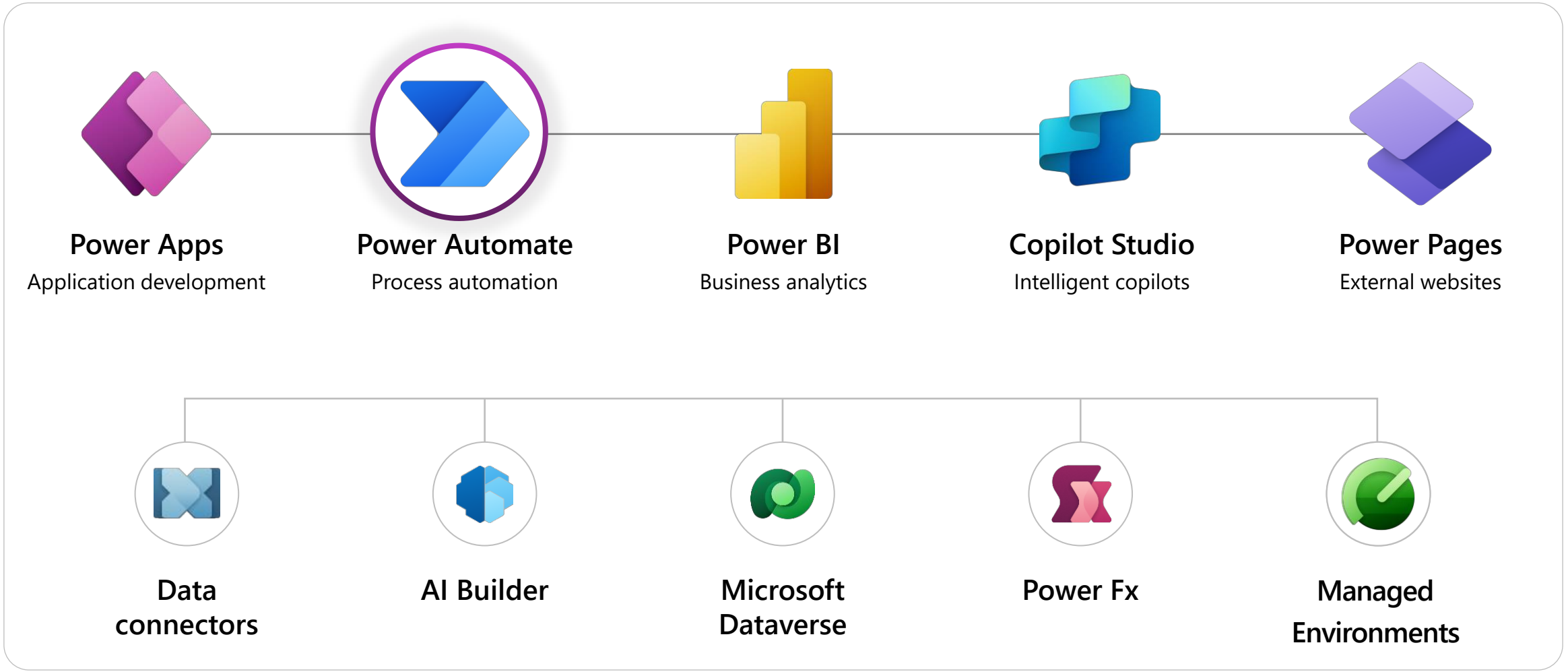
Reza Dorrani

Principal Program Manager
Power CAT

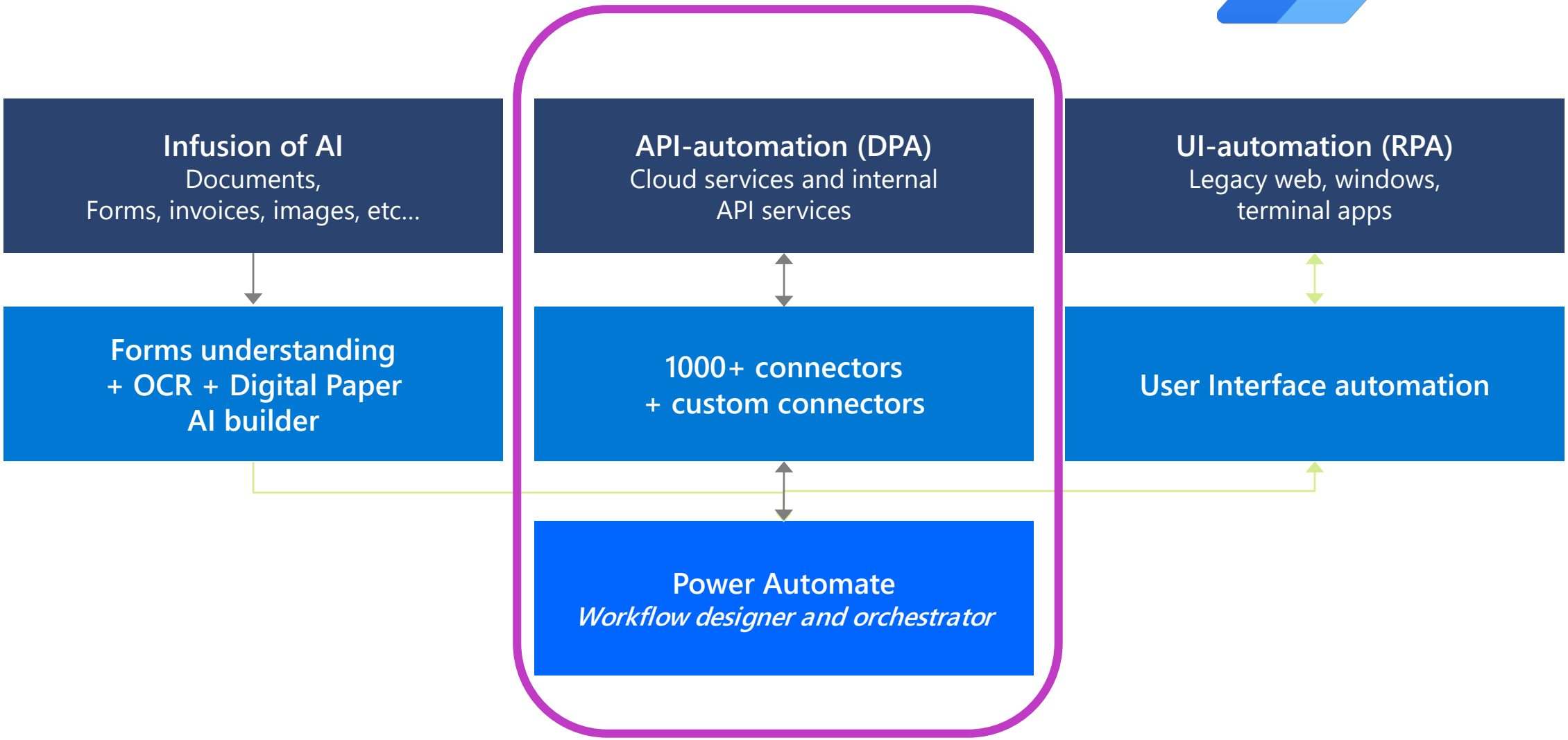
aka.ms/whoispowercat

Microsoft Power Platform

An all software development platform across Microsoft 365, Azure, Dynamics 365, and standalone applications



Power Automate: Automation Portfolio



How do I?

Make my flow successful

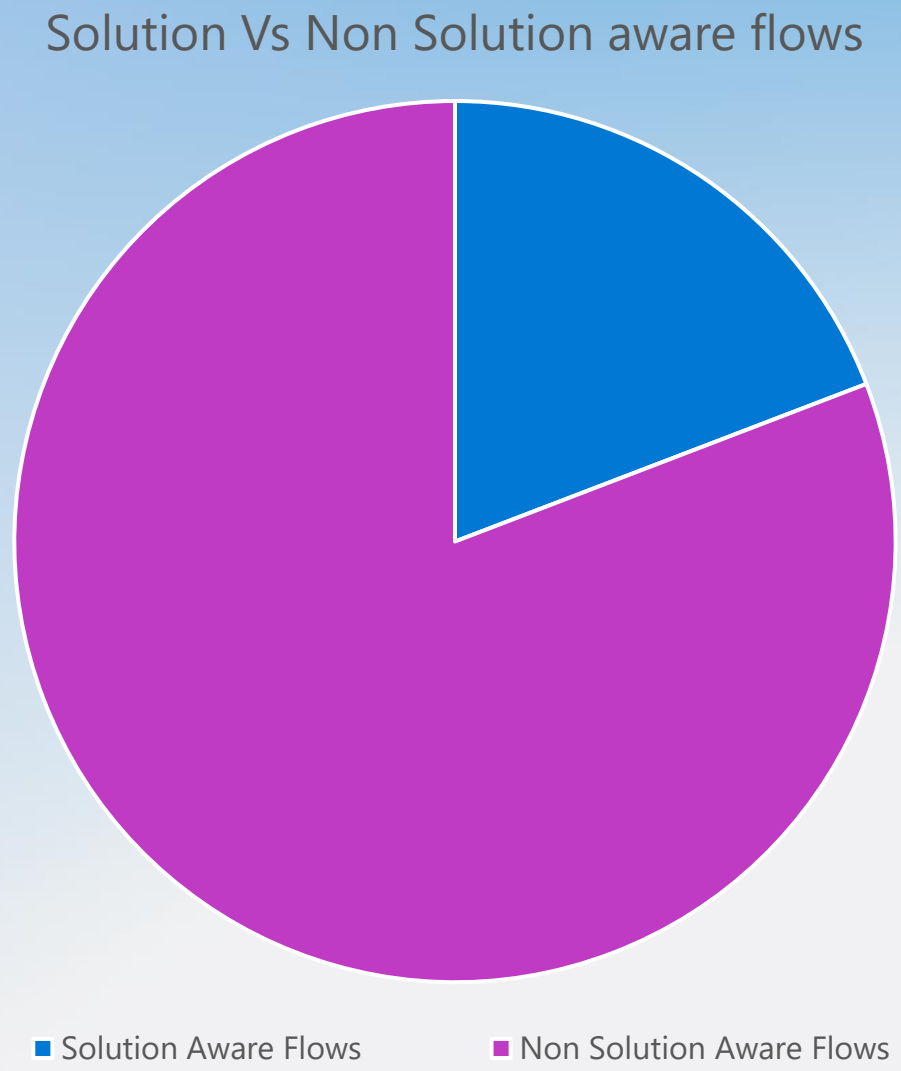
Make my flow performant

Make my flow reusable

Make my flow scalable and reliable

Why Build Cloud flows in a Solution?

Currently, around **23%** of Power Automate Cloud flows are solution aware flows



Advantages of Building Solution aware flows

- Enhanced Organization and Management

  More visibility and version control
- Enhanced ALM

  Ready to deploy to your downstream environments via pipelines
- Overcome Flow Creation limits

  Flow Creation limits per user (max of 600 flows) doesn't apply to solution aware flows
- Easy Maintenance

  Use connection references to update flow connections at the time of solution import.
- More…

  Change Primary Owner, Enables use of environment variables, **longer** Run History and more..

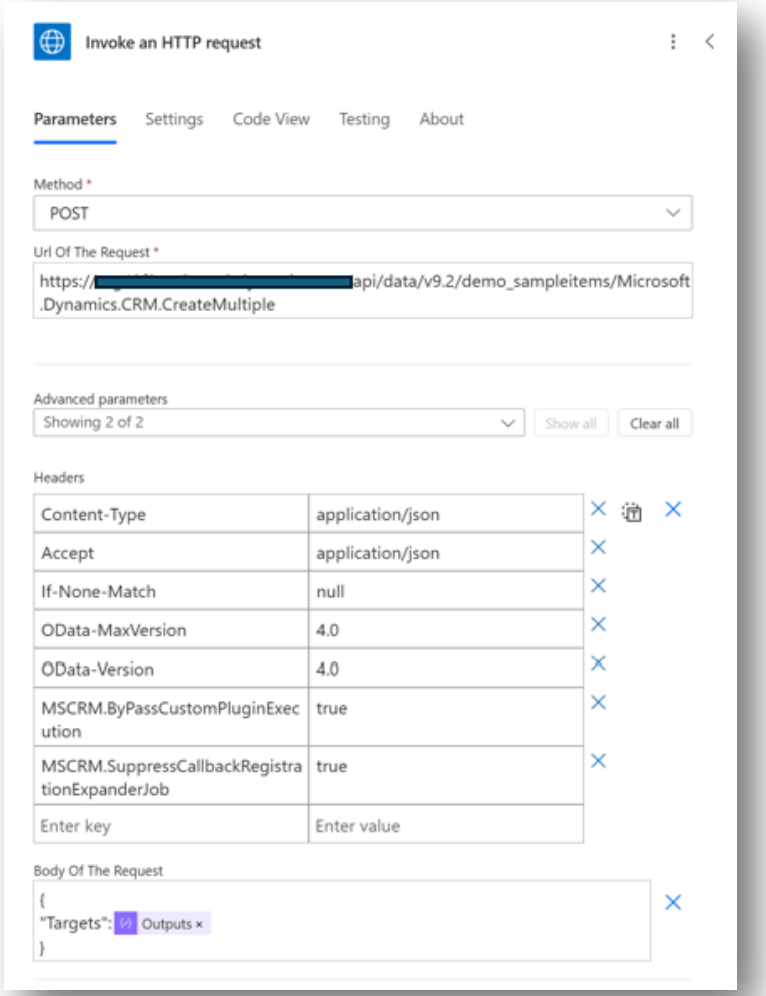
Performing Bulk Operations

Use Batch/Bulk operations to optimize throughput

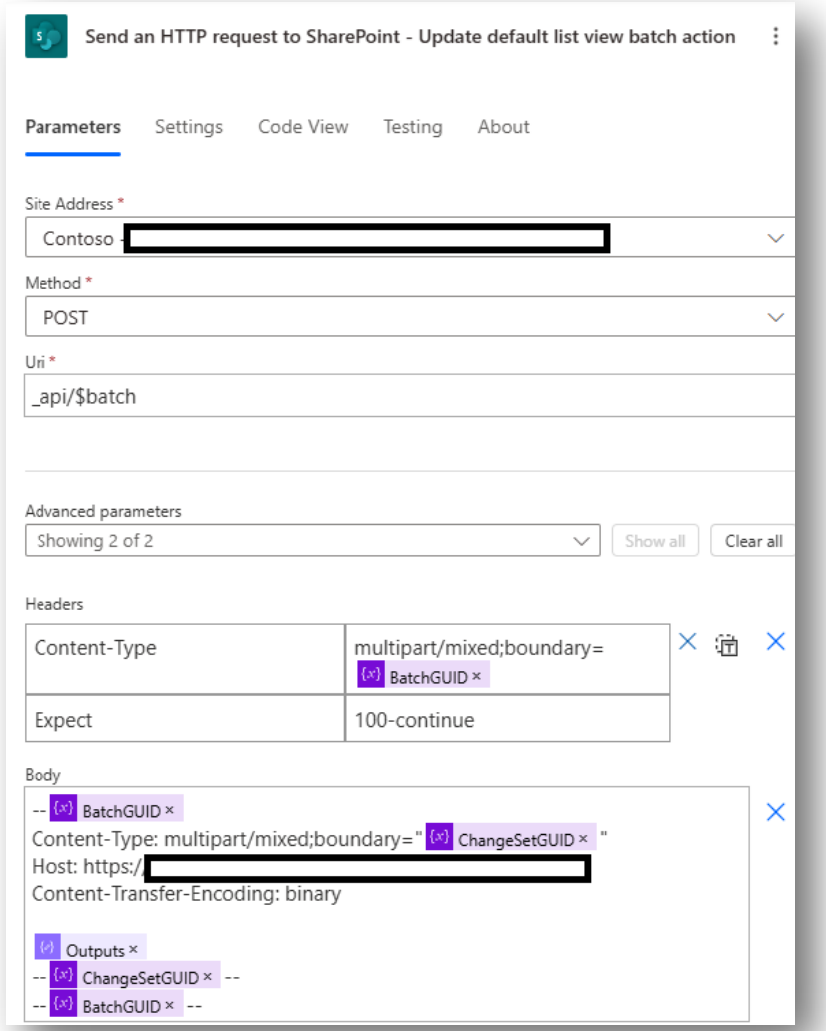
- When to use this?
  - To create/update large data, popular approach is to use For-Each loop.
- How to use BATCH?
  - Connectors/services expose API to post requests in BATCH.
- Why to use BATCH?
  - To reduce API calls and execution time
  - Process datasets in Bulk

More BATCH examples

Dataverse BATCH API



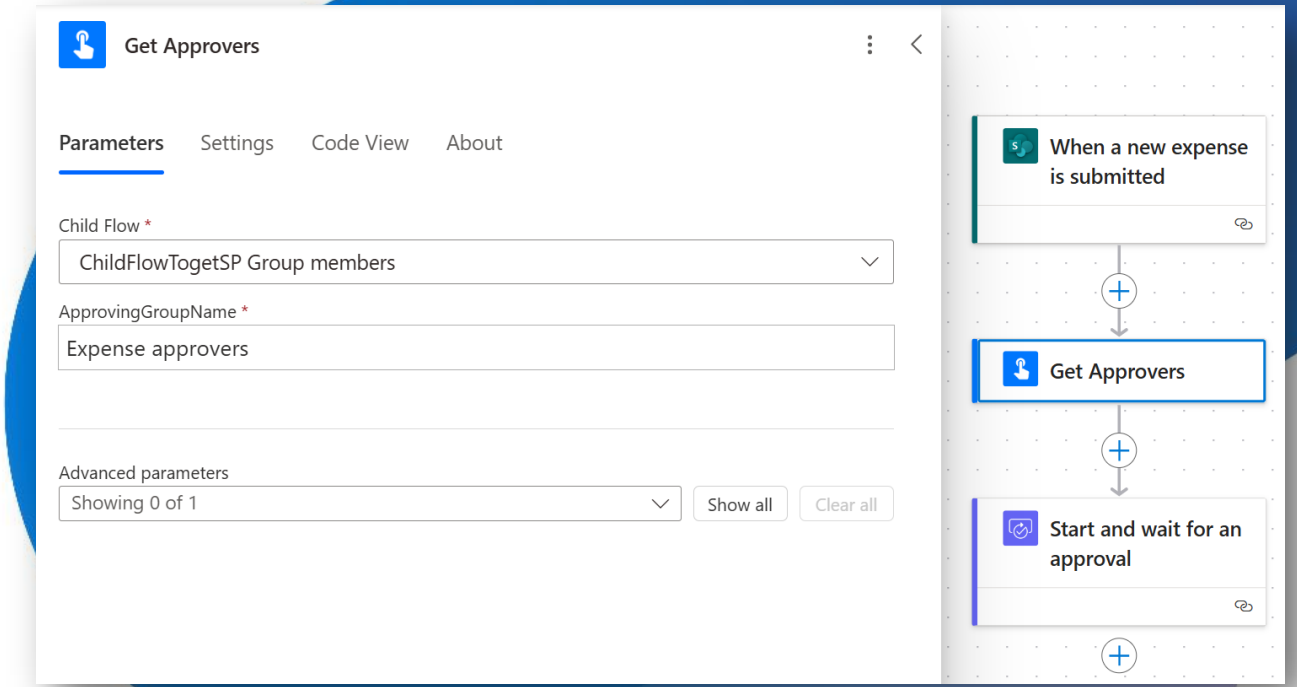
SharePoint BATCH API request



Creating reusable Flows

Why use Child flows?

- Modularity
  - Allows building reusable components that can be plugged into different workflows
- Scalability
  - Breakdown flow logic into smaller and manageable pieces
- Collaboration
  - allows team members to work on separate components of a large automation project simultaneously



Other Best Practices

What's more??

Here are more best practices to consider while building cloud flows

- Leave complex business logic out – low code plugins, custom connectors etc.
- Work only with relevant data – Use Select columns, OData filter conditions
- Use trigger conditions – To trigger your flow only when required
- Leverage Parallel branches to do more processing at the same time

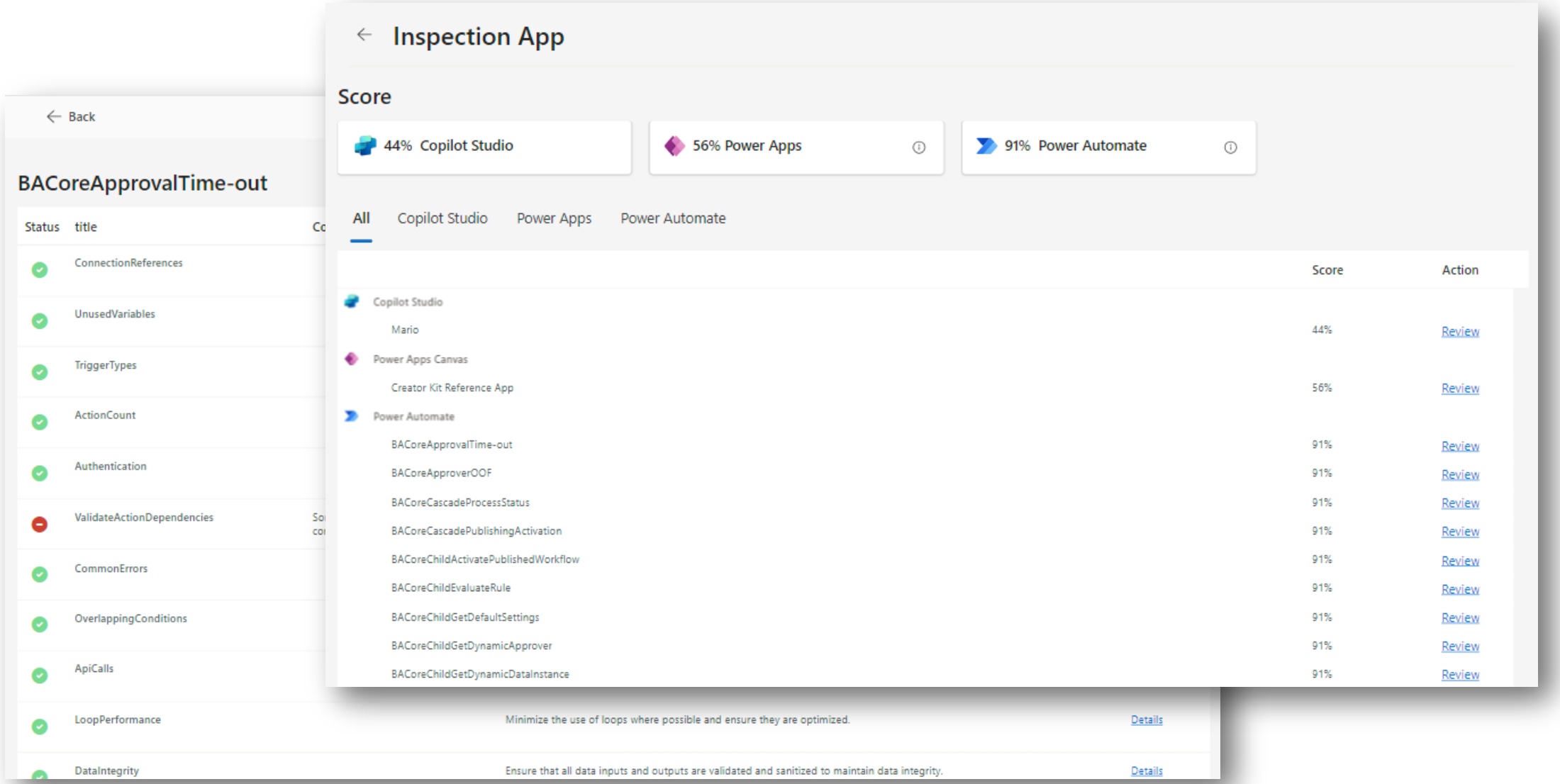
If we had a few more hours or days or weeks, we would have covered a lot more…

BUT we couldn’t and therefore we are proud to announce the latest addition to Power Automate Learn documentation (Coming soon)

Our very first..

Power Automate Best Practices White Paper !!!

Power Platform Review Tool COMING SOON JAN 2025



Microsoft Power Platform

Thank you!!!

© Copyright Microsoft Corporation. All rights reserved.

Appendix

Connection References vs Connections

- ## Connections

  Connections are configurations that establish link between Power Automate and external services like Dataverse, SharePoint etc.

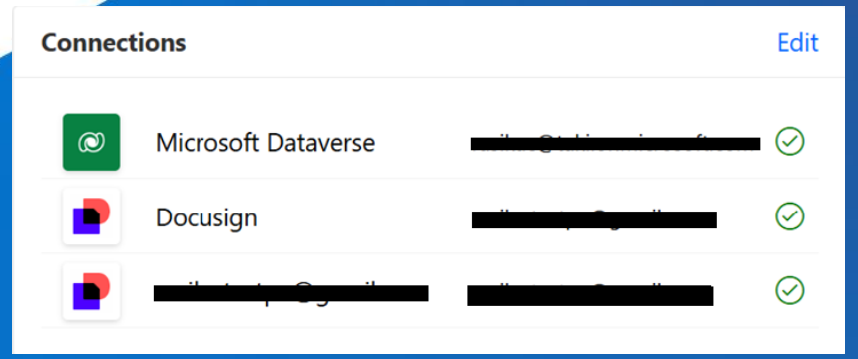
  If you were to change a connection for a cloud flow, every action associated with the connection needs to be updated.

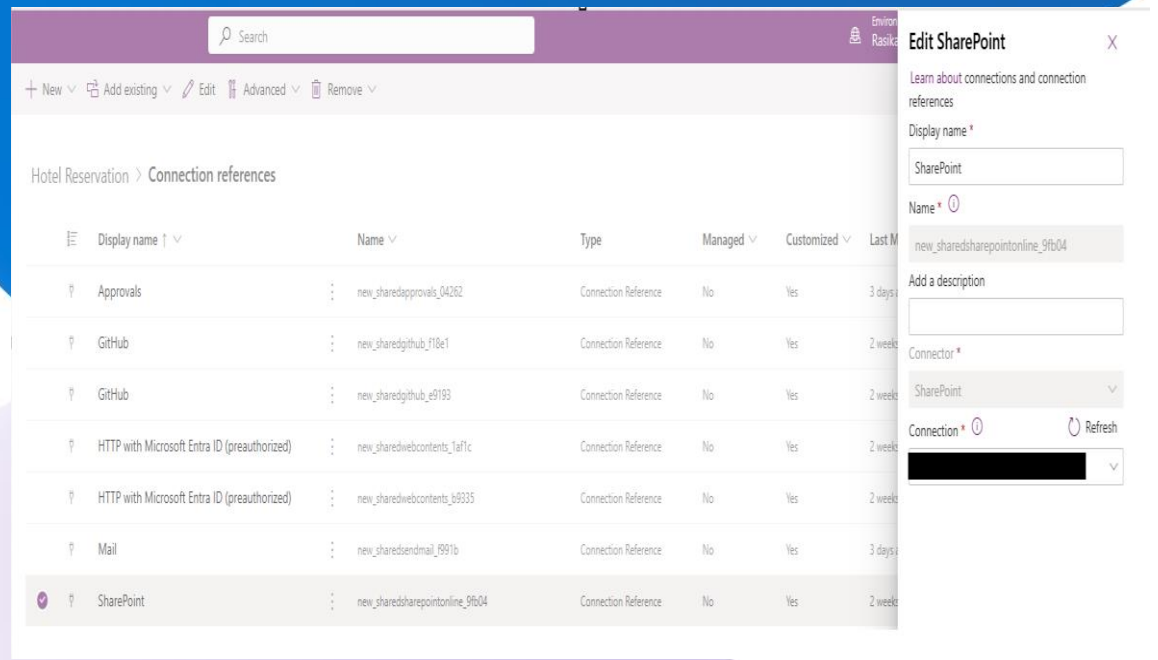
- ## Connection References

  Are wrappers that act as placeholders or pointers to actual connection

  Separates flow design from specific connection instances

  Only connection references need to be updated instead of all the actions



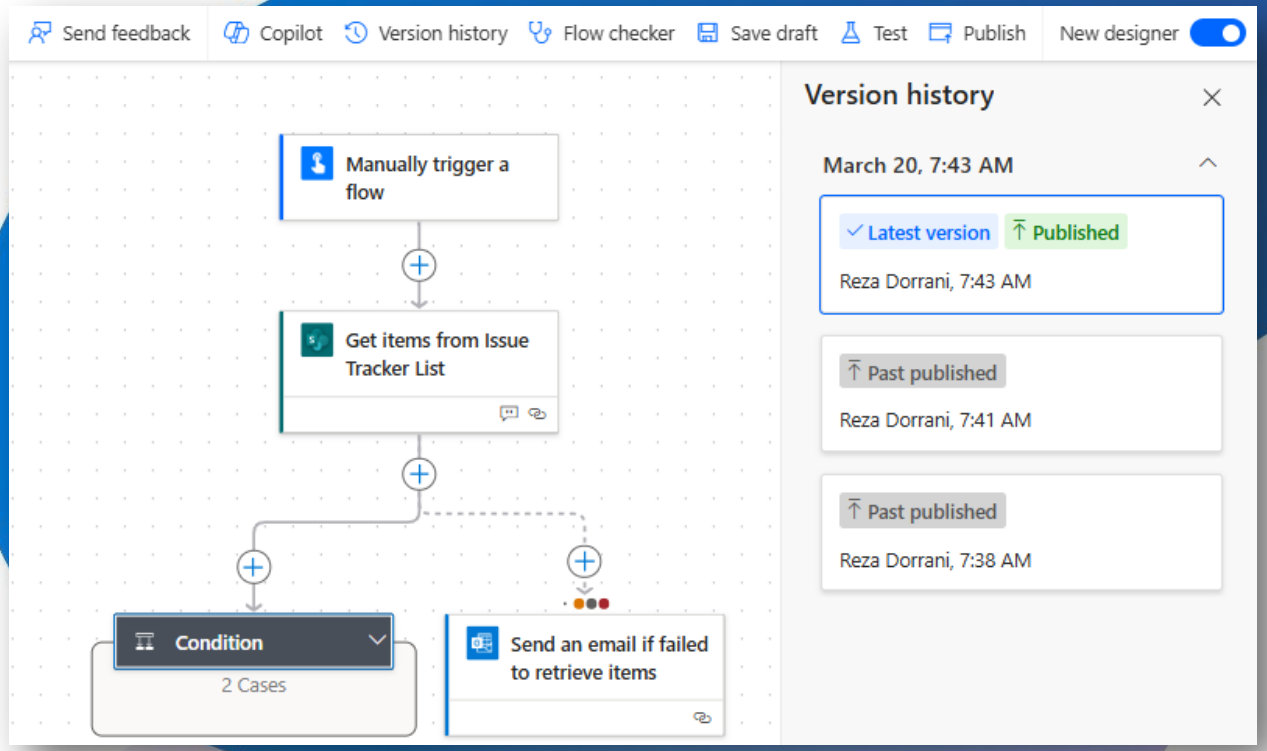


Flow versioning

Users can save flows as draft as they continue to evolve the flow logic and publish once complete

Versioning is tracked via storing flow metadata information in Dataverse

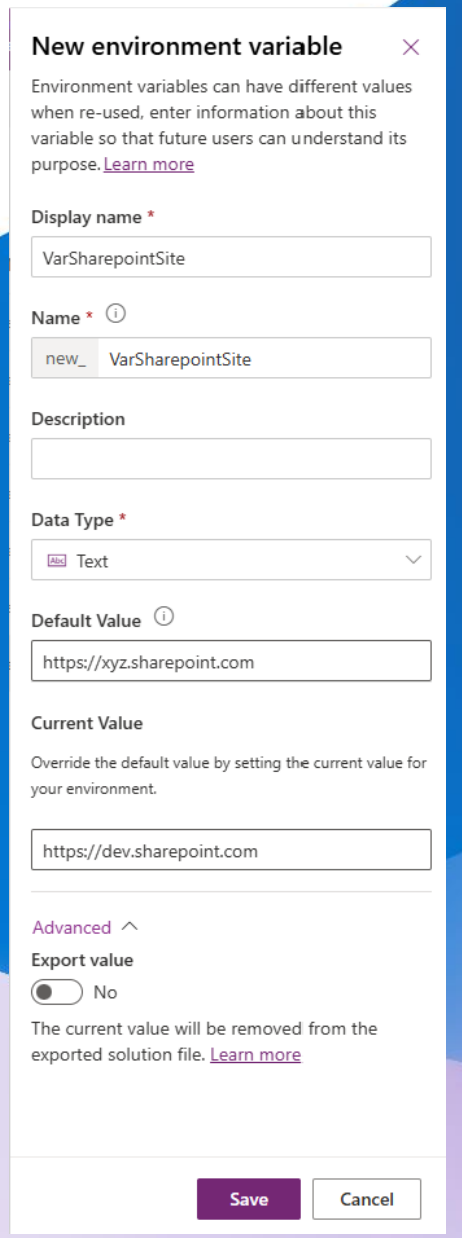
Allows reverting to previous versions

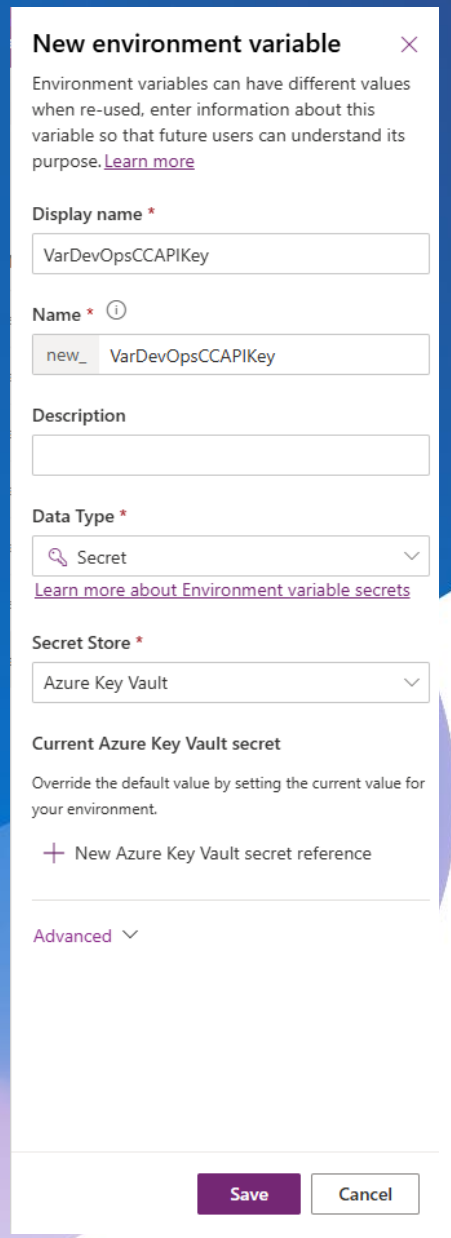


Environment variables

Helps define dynamic values that can change between source and target environments, without changing flow definition

Gives ability to store sensitive information like passwords, API Keys etc. stored in Azure Key Vault



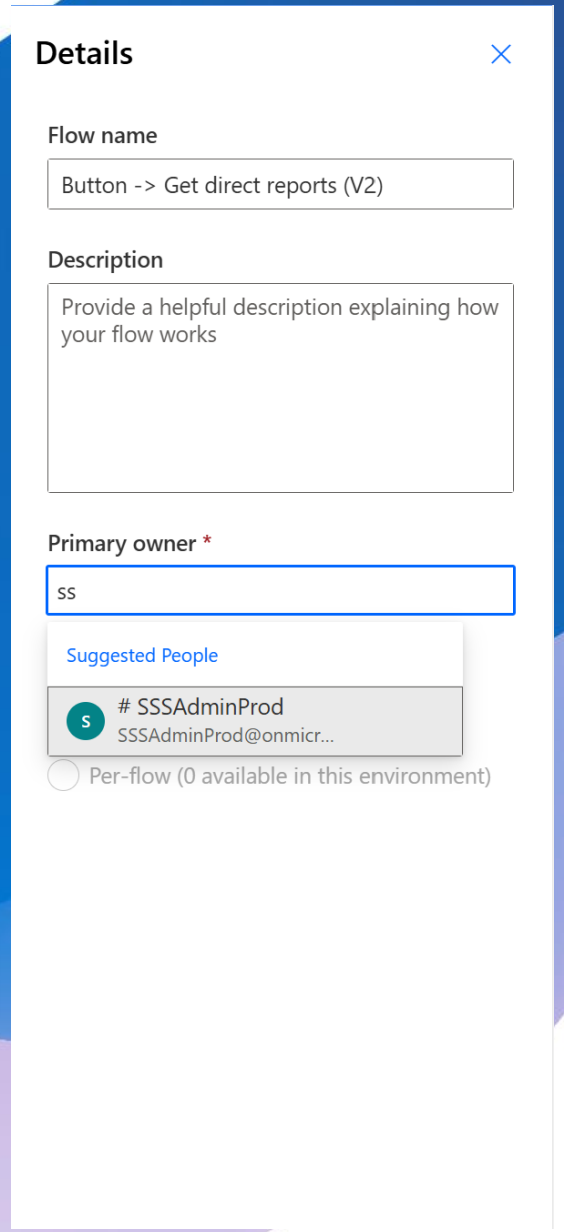


Primary Owner Change

Ensures smooth transition of flow ownership from one user to other.

Flow owners responsibilities include

- Flow Management to have full control over the flow
- Permissions and Sharing to determine who can access
- Monitoring and troubleshooting to address any issues
- Licensing is determined based on the flow owner license



Manage Run History

Solution Flows run history can be stored in Dataverse

Ability to store flow run history data beyond 28 days (define your own custom duration)

Use automation center to provide comprehensive monitoring and troubleshooting experiences

The following table lists selected properties for the Flow Run (flowrun) table.

Expand table

Property	Value
DisplayName	Flow Run
DisplayCollectionName	Flow Runs
SchemaName	`flowrun`
CollectionSchemaName	`flowruns`
EntitySetName	`flowruns`
LogicalName	`flowrun`
LogicalCollectionName	`flowruns`
PrimaryIdAttribute	`flowrunid`
PrimaryNameAttribute	`name`
TableType	`Elastic`
OwnershipType	`UserOwned`

Best Practice

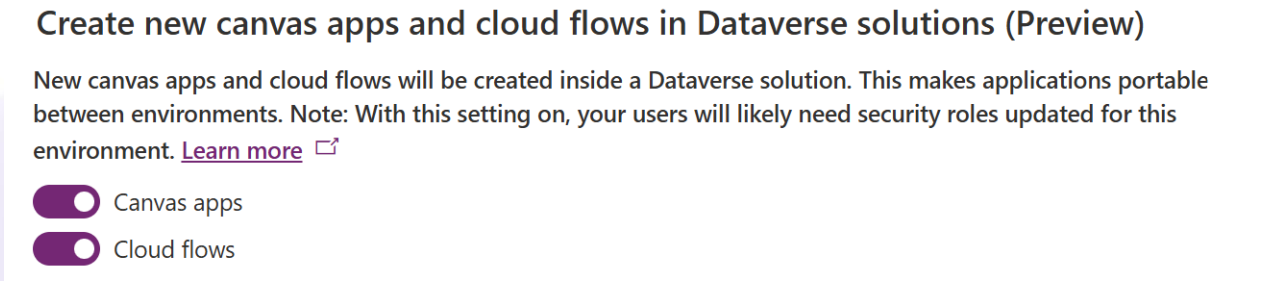
Encourage your makers to build artifact
in solutions.

Turn on creating solution by default

The following table lists selected properties for the Flow Run (flowrun) table.

Expand table

Property	Value
DisplayName	Flow Run
DisplayCollectionName	Flow Runs
SchemaName	`flowrun`
CollectionSchemaName	`flowruns`
EntitySetName	`flowruns`
LogicalName	`flowrun`
LogicalCollectionName	`flowruns`
PrimaryIdAttribute	`flowrunid`
PrimaryNameAttribute	`name`
TableType	`Elastic`
OwnershipType	`UserOwned`



Licensing Limits

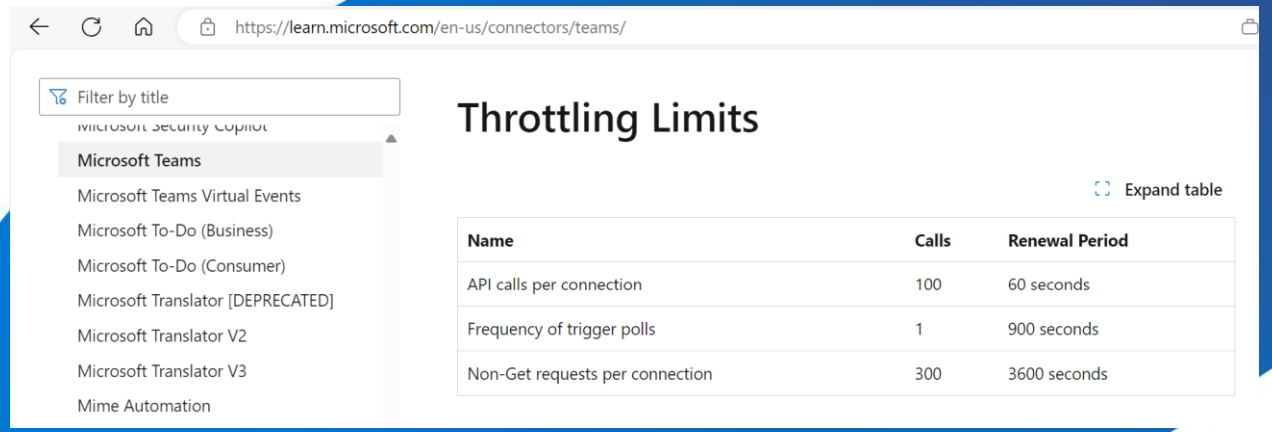
- User limits are applied across all flows
  - Automatic flows which they created
  - Manual flows which they executed
- Flow/Process limits are applied to the flows
  - Per-Process can apply to group of related flows
  - Per flow applies to a single flow
- Pay-Go environments
  - Flows don’t have daily quota as you pay for consumption

License	Power platform actual limits per 24 hours	Power platform transition period limits per 24 hours
Power Automate premium	40K/user	200K/flow
Power Automate process plan	250K/process	500K/process
Office 365	6K/user	10K/flow
Power Apps premium	40K/user	200K/flow
Dynamics 365 professional	40K/user	200K/flow
Dynamics 365 Enterprise applications	40K/user	200K/flow
Dynamics 365 Team member	6K/user	10K/flow

Plan	Limits per 24 hours	Data consumption per day
Office 365 Flow licenses, Power Apps per app, and Dynamics team member and trials	Final limit: 6,000 actions across all flows created by a single user. Transition limit: 10,000 actions per flow	1 GB across all flows created by a single user.
Power Automate Premium, Power Apps Premium, Power Automate Per user, Power Automate Per user with attended RPA, and Power Apps per user	40,000 actions across all flows created by a single user. Transition limit: 200,000 actions per flow	10 GB across all flows created by a single user.
Dynamics Professional licenses	40,000 actions across all flows created by a single user. Transition limit: 200,000 actions per flow	10 GB across all flows created by a single user.
Dynamics Enterprise Application licenses	40,000 actions across all flows created by a single user. Transition limit: 200,000 actions per flow	10 GB across all flows created by a single user.
Power Automate Process license, and Power Automate per flow license	250,000 actions per process. Transition limit: 500,000 actions per flow	50 GB storage per flow.

Connector API Limits

- These limits are specific to, per-connection, each connector
- When API limits are reached, 429 (Too many requests), error code will be returned.
- For Dataverse
  - Service limits are evaluated per user
  - When called by a flow the “user” is whoever is associated with the action. Usually this is the flow owner but can be the invoking user if using invoking user context in the action



Name	Calls	Renewal Period
API calls per connection	100	60 seconds
Frequency of trigger polls	1	900 seconds
Non-Get requests per connection	300	3600 seconds

DATAVERSE API LIMITS

Measure	Description	Limit per web server
Number of requests	The cumulative number of requests made by the user.	6000 within the 5 minute sliding window
Execution time	The combined execution time of all requests made by the user.	20 minutes (1200 seconds) within the 5 minute sliding window
Number of concurrent requests	The number of concurrent requests made by the user	52 or higher

Flow Concurrency Limits

- Designing scalable, efficient flows includes understanding of the concurrency, looping and de-batching limits to help avoid unnecessary delays
- Limits for a single run are as follows

Name	Limit	Notes
Concurrent runs	- Unlimited for flows with Concurrency Control turned off - 1 to 100 when Concurrency Control is turned on (defaults to 25)	This is the limit for how many runs a flow can have at the same time. **Note**: Concurrency Control is set in the flow's trigger settings and is off by default. Turning on Concurrency Control can't be undone without deleting and re-adding the trigger.
Waiting runs	- Not applicable when Concurrency Control is off - 10 plus the degree of parallelism (1-100) when Concurrency Control is on	This limit describes the highest number of flow runs that can be queued when the flow is at its maximum number of concurrent runs. **Note**: Additional triggers that arrive while the waiting runs limit is met might be re-tried by the connector. However, the retry attempts might not succeed if the maximum waiting limit continues to be met for an extended period of time. To ensure all triggers result in flow runs, leave the Concurrency Control setting off in the flow's trigger.
Apply to each array item	5,000 for Low, 100,000 for all others	This limit describes the highest number of array items that an "apply to each" loop can process. To filter larger arrays, you can use the query action.
Apply to each concurrency	1 is the default limit. You can change the default to a value between 1 and 50 inclusively.	This limit is highest number of "apply to each" loop iterations that can run at the same time, or in parallel.
Split on items	- 5,000 for Low without trigger concurrency - 100,000 for all others without trigger concurrency - 100 with trigger concurrency	For triggers that return an array, you can specify an expression that uses a 'SplitOn' property that splits or debatches array items into multiple workflow instances for processing, rather than use a "Foreach" loop. This expression references the array to use for creating and running a workflow instance for each array item. **Note**: When concurrency is turned on, the Split on limit is reduced to 100 items.
Until iterations	- Default: 60 - Maximum: 5,000	
Paginated items	5,000 for Low, 100,000 for all others	To process more items, trigger multiple flow runs over your data.

Action Burst Limits

- There is a per-flow cap of 100,000 actions per 5 minutes
- Bursts of triggering/loops that cause more than the above limit will cause throttling and delay executions of flows
- To avoid reaching these limits, consider
  - Distributing the work between multiple flows
  - Use trigger conditions to avoid unnecessary flow runs
  - Design your loops efficiently
  - Consider split between parent/child flows

Additional Limits

Flow Design and Creation limits

Name	Limit	Notes
Actions per workflow	500	Flows with a large number of actions may encounter performance issues while you edit them, even if they have fewer than 500. Consider using child flows to reduce the number of actions in a single flow or if you need more than 500.
Allowed nesting depth for actions	8	Add child flows if you need more than eight levels of nesting.
Switch scope cases limit	25	
Variables per workflow	250	
Length of `action` or `trigger` name	80 characters	
Characters per expression	8,192	

The following table describes the limit for the My flows and Team flows tabs.

Expand table

Name	Limit	Notes
Number of flows owned by a single user	600	Use flows under solutions if you need more than 600.

Flow Timeout Limits

The following table describes the duration limits for a single flow run.

Expand table

Name	Limit	Notes
Run duration	30 days	Run duration is calculated using a run's start time and includes flows with pending steps like approvals. After 30 days, any pending steps time out.
Run retention in storage	30 days	Run retention is calculated using a run's start time.
Minimum recurrence interval	60 seconds	
Maximum recurrence interval	500 days	
Minimum postpone interval	5 seconds for Low, 1 second for all other performance profiles	

Know your cloud flows limits !

Limits to consider

- Licensing limits for daily per-flow action quotas
- API throughput limits on connectors
- Dataverse API limits
- Flow concurrency limits
- Burst throughput limits for flow actions

- Limits Documentation
- Licensing FAQ