Looping without Looping

Hamish Kingsbury

Principal Integration Specialist
Abley

I Spy......

How many zipsters and lizards can you find?

Come tell me at the drinks and nibbles and there might be a prize for you…

Abley

Scenario

Getting all FME Flow users

FME Flow API

- 100+ users

- Can only get 20 users at a time

- Use the FMEREST API
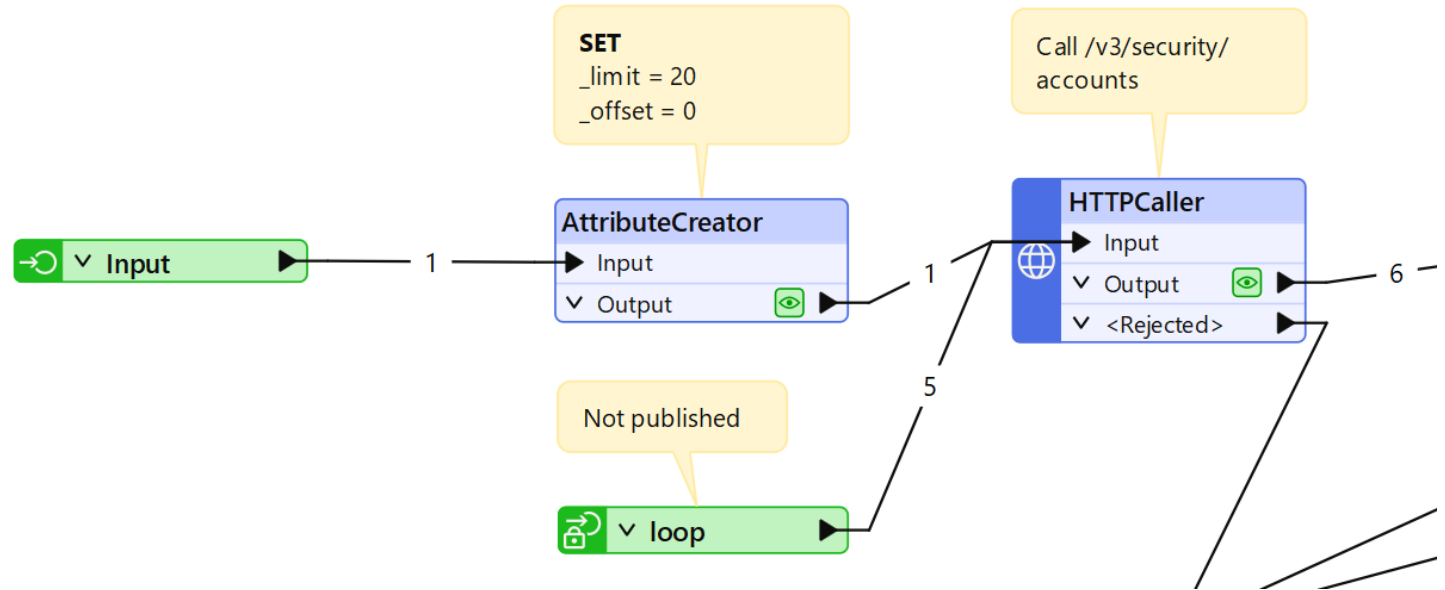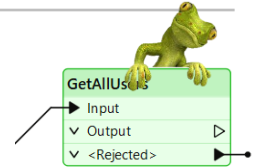
Looping Custom Transformer

Looping Custom Transformer
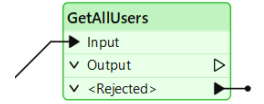
- Call the API

Looping Custom Transformer

- Call the API
- **Extract the total count of expected results**
- **Extract the actual data**
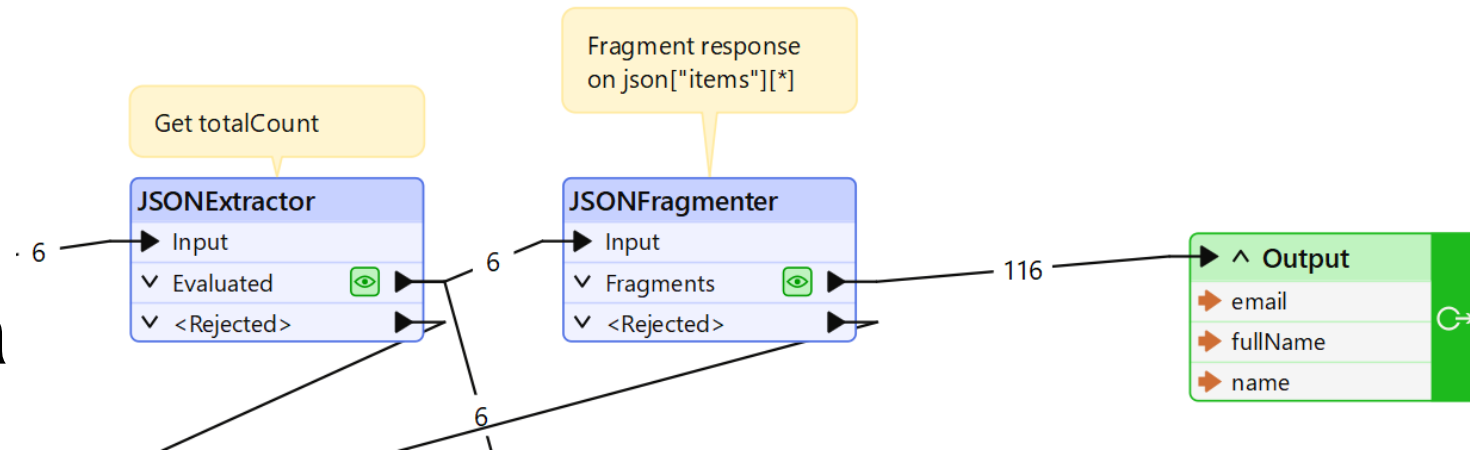
Looping Custom Transformer
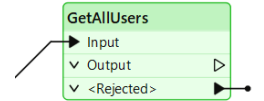
- Call the API
- Extract the total count of expected results
- Extract the actual data
- **Check if the next call will get data**
- **Update the offset and loop to the start**

GetAllUsers
Input
Output
<Rejected>

Tester
Input
Passed
Failed

AttributeCreator_2
Input
Output

Loop to loop

loop back to start with increased offset

Test if offest of next call will exceed the total count of avaliable results

SET
_limit = 20
_offest = @Evaluate(@CurrentAttribute()+@Value(_limit))
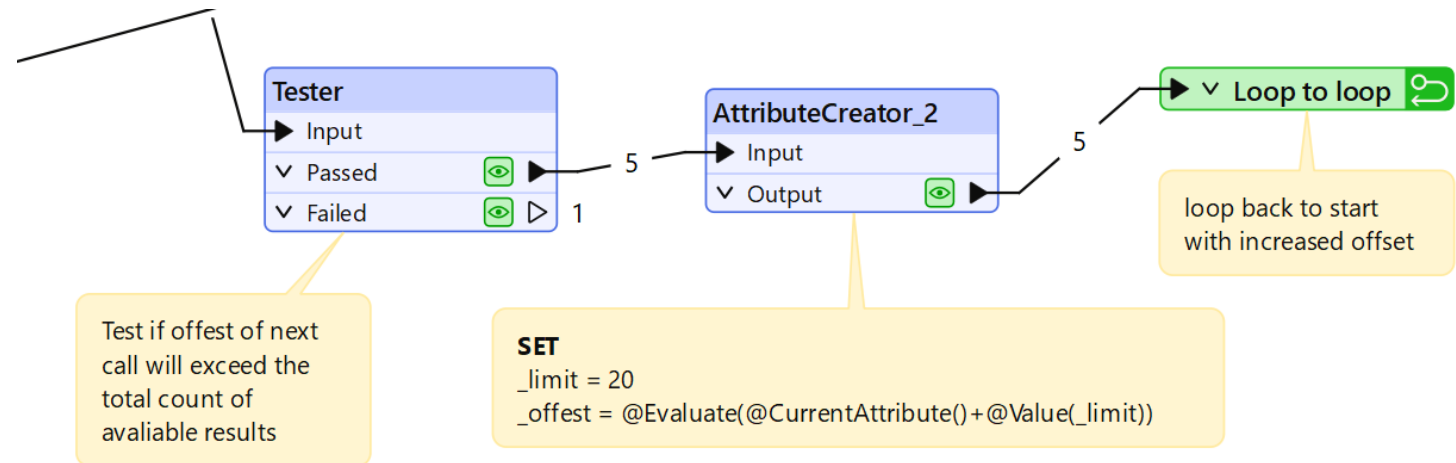
Looping Custom Transformer
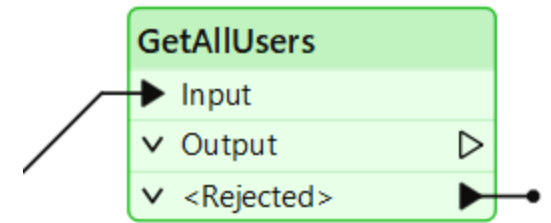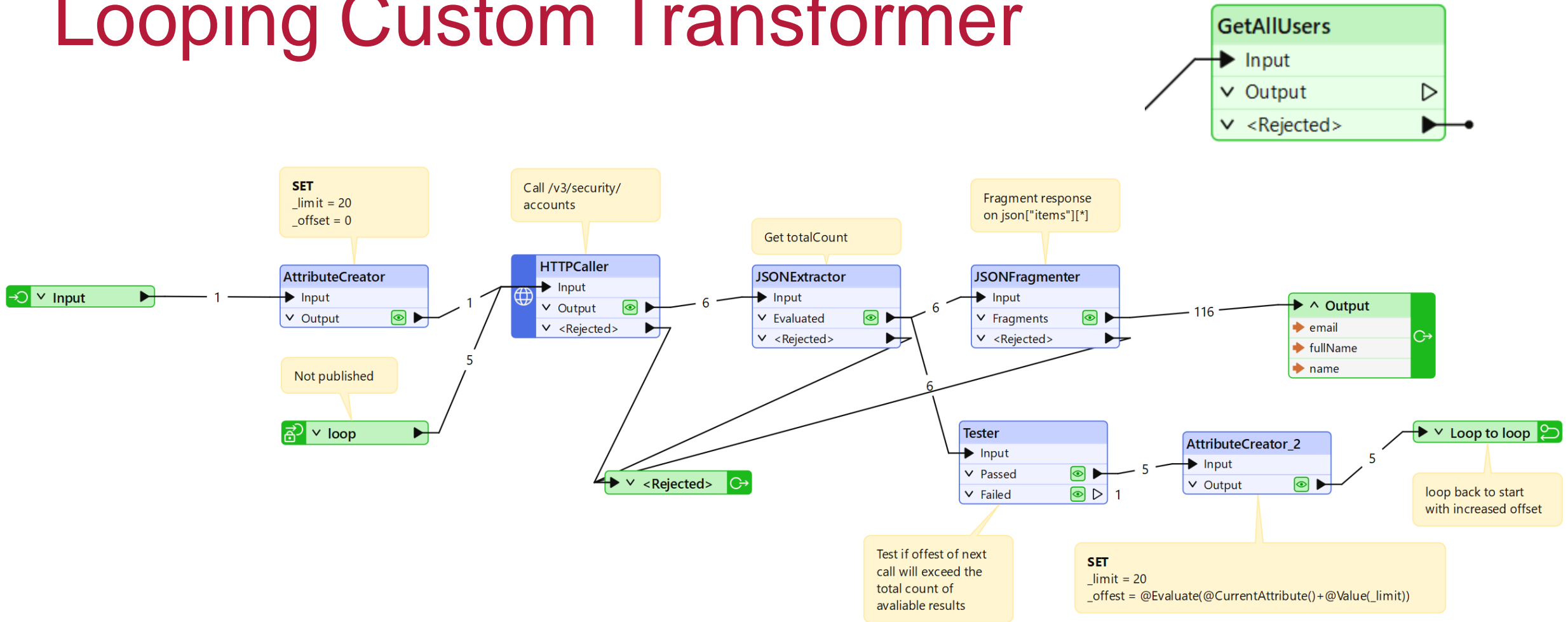
How about without looping?

Using the 'count'
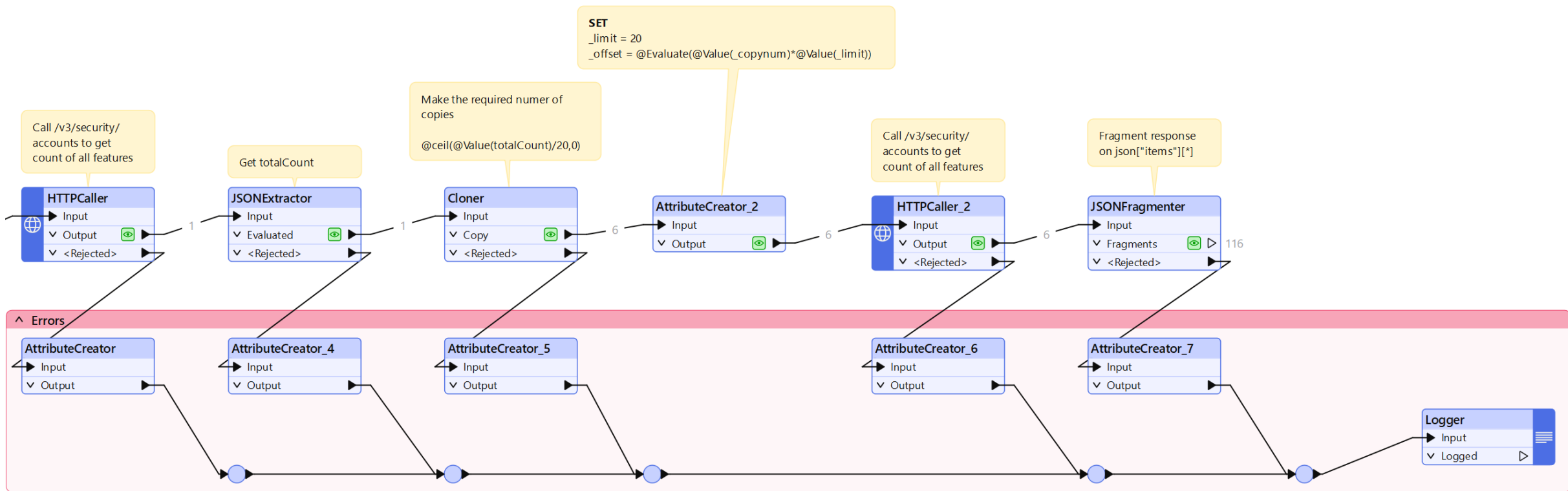
Using the 'count'

- Call the API
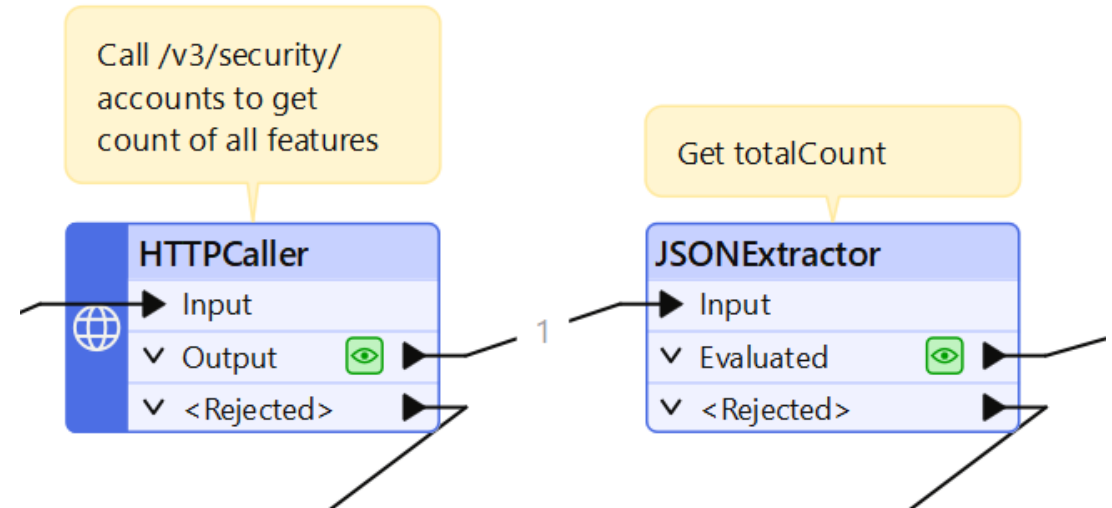
- Extract the total count of expected results

Using the 'count'

- Call the API
- Extract the total count of expected results
- Use the count to get the required number of calls

SET
_limit = 20
_offset = @Evaluate(@Value(_copynum)*@Value(_limit))

Make the required numer of copies

@ceil(@Value(totalCount)/20,0)

Cloner
- Input
- Copy
- <Rejected>

6

AttributeCreator_2
- Input
- Output

6
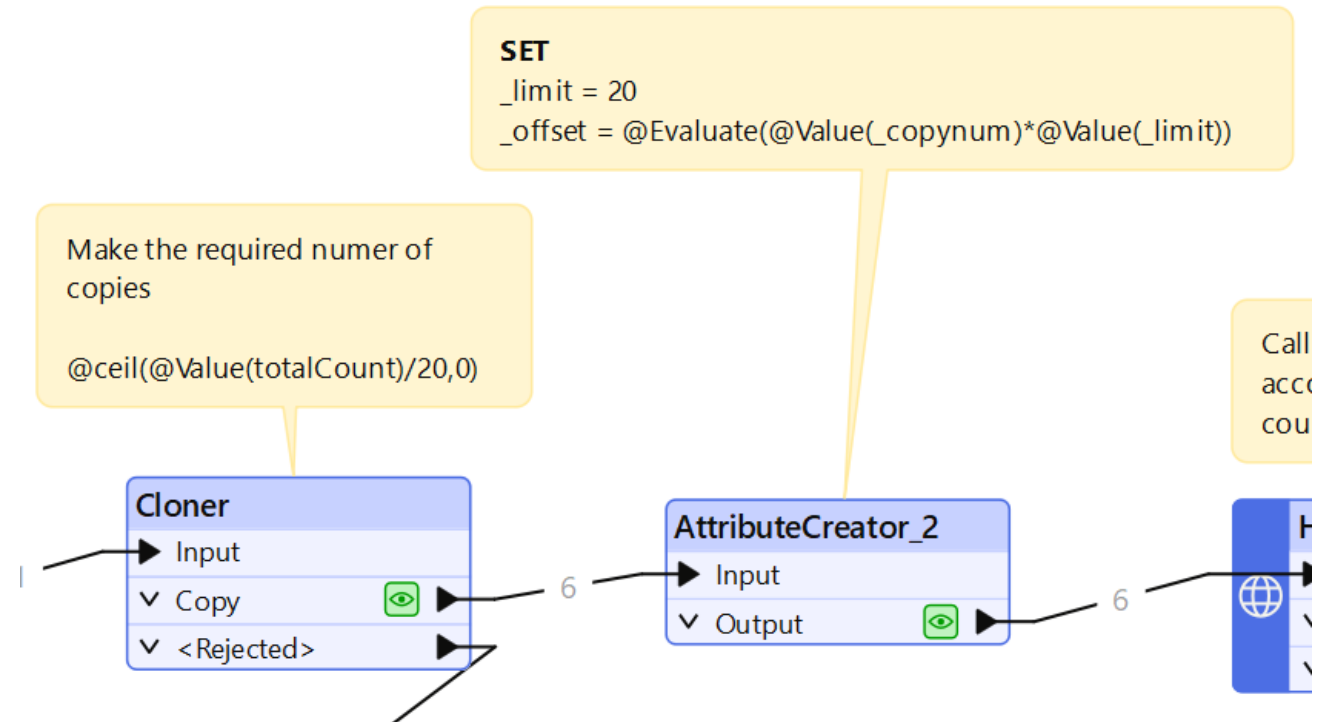
Call
acc
cou

Using the 'count'

- Call the API
- Extract the total count of expected results
- Use the count to get the required number of calls
- **Make the calls to the API and extract data**

Call /v3/security/ accounts to get count of all features

Fragment response on json["items"][*]

HTTPCaller_2
- ▶ Input
- ⌄ Output 👁 ▶
- ⌄ <Rejected> ▶

6

JSONFragmenter
- ▶ Input
- ⌄ Fragments 👁 ▷ 116
- ⌄ <Rejected> ▶

16
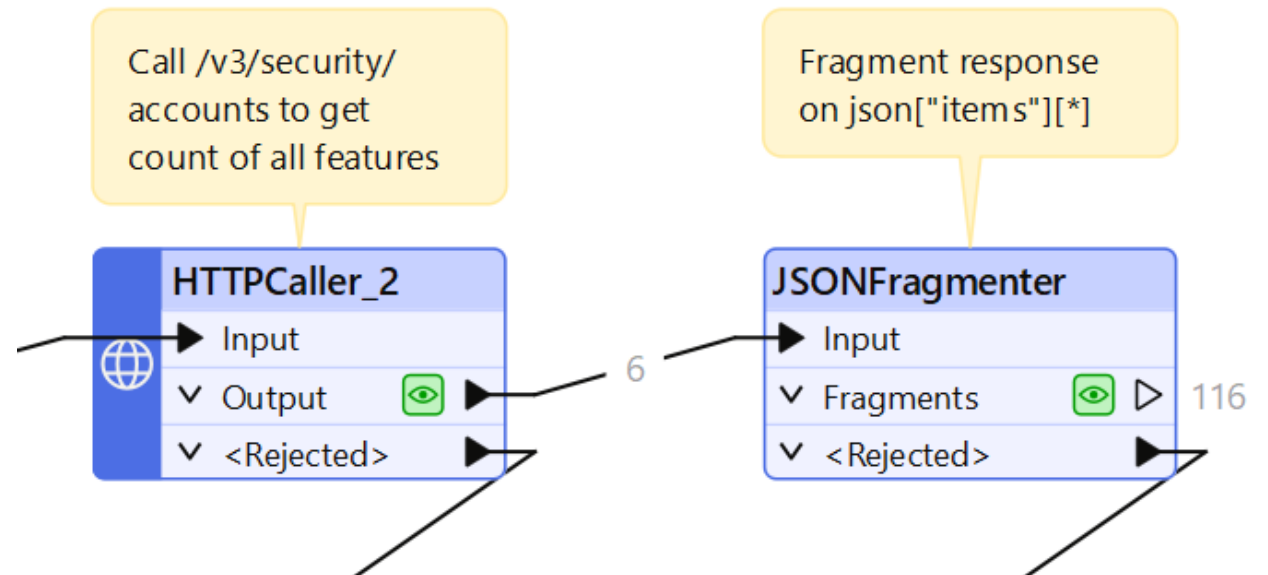
Using the 'count'
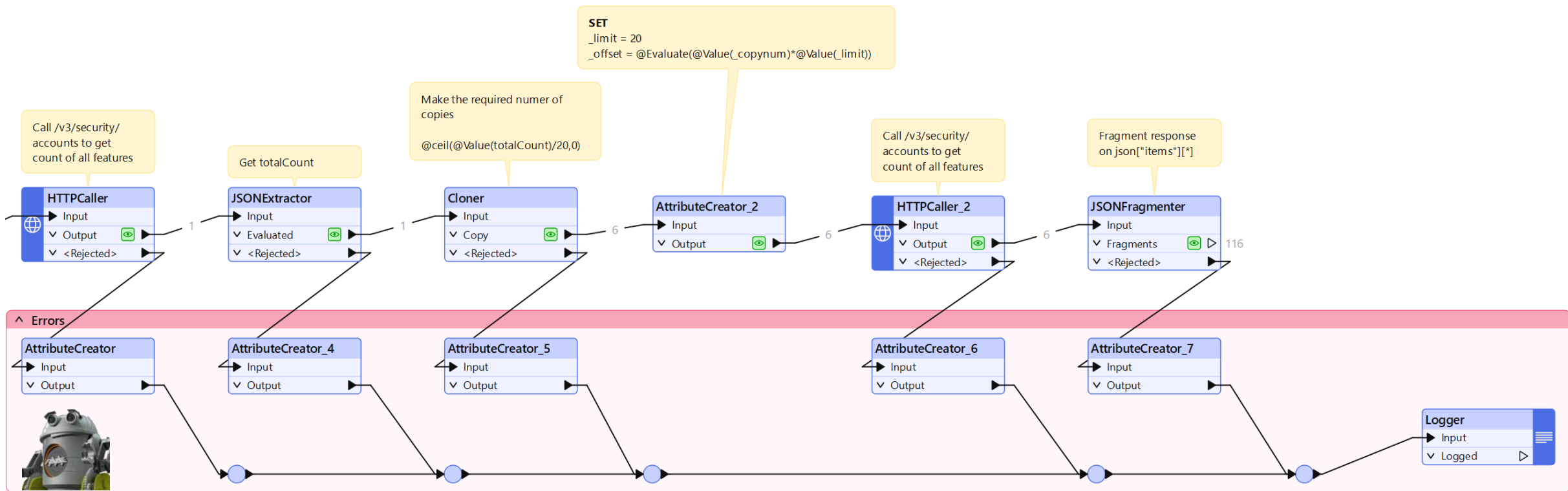
Variables?
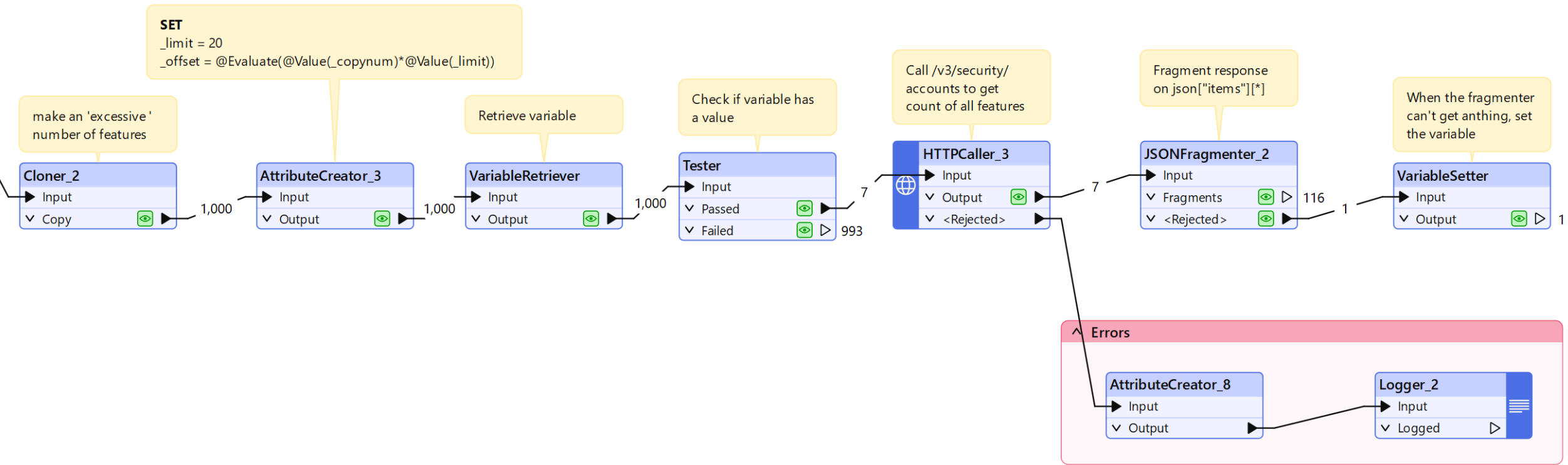
Variables?

- Make lots of features
- Set offset based on _copynum

Variables?

- Make lots of features
- Set offset based on _copynum
- **Check what the variable contains**

Retrieve variable

Check if variable has a value

VariableRetriever
▶ Input
∨ Output

1,000

1,000

Tester
▶ Input
∨ Passed
∨ Failed

7

993

20

Variables?

- Make lots of features
- Set offset based on _copynum
- Check what the variable contains
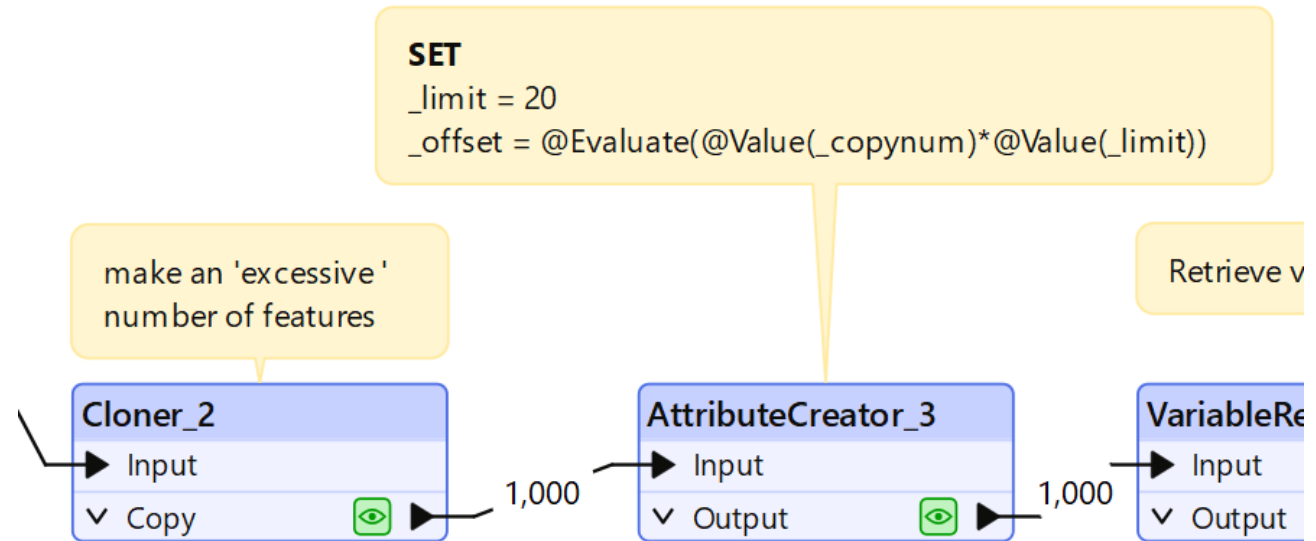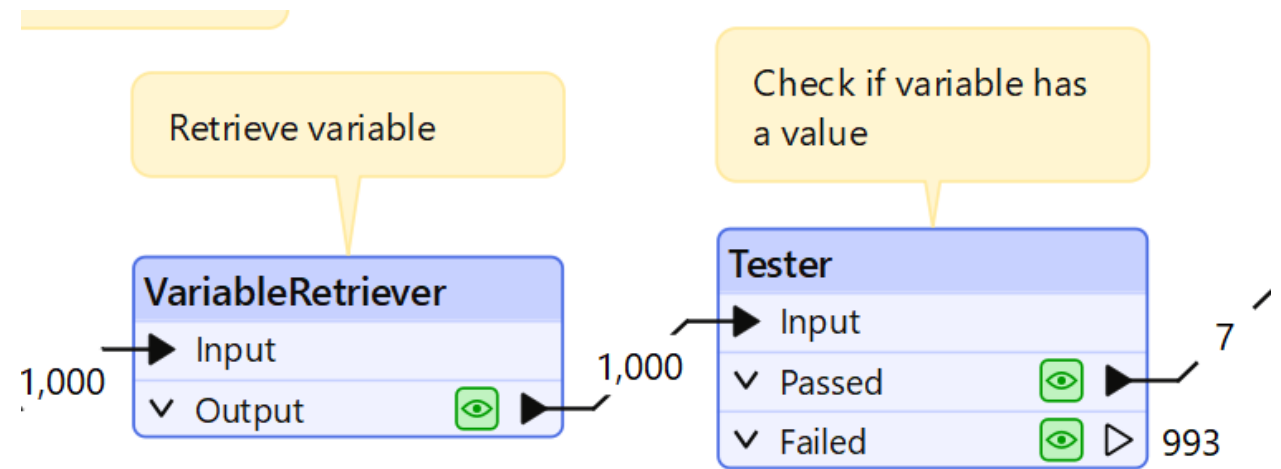- **Make API calls, extract data**
- **If the extract fails, set the variable**

Call /v3/security/ accounts to get count of all features

Fragment response on json["items"][*]

When the fragmenter can't get anthing, set the variable

HTTPCaller_3
Input
∨ Output
∨ <Rejected>

JSONFragmenter_2
Input
∨ Fragments
∨ <Rejected>

VariableSetter
Input
∨ Output

7

7

116

1

1
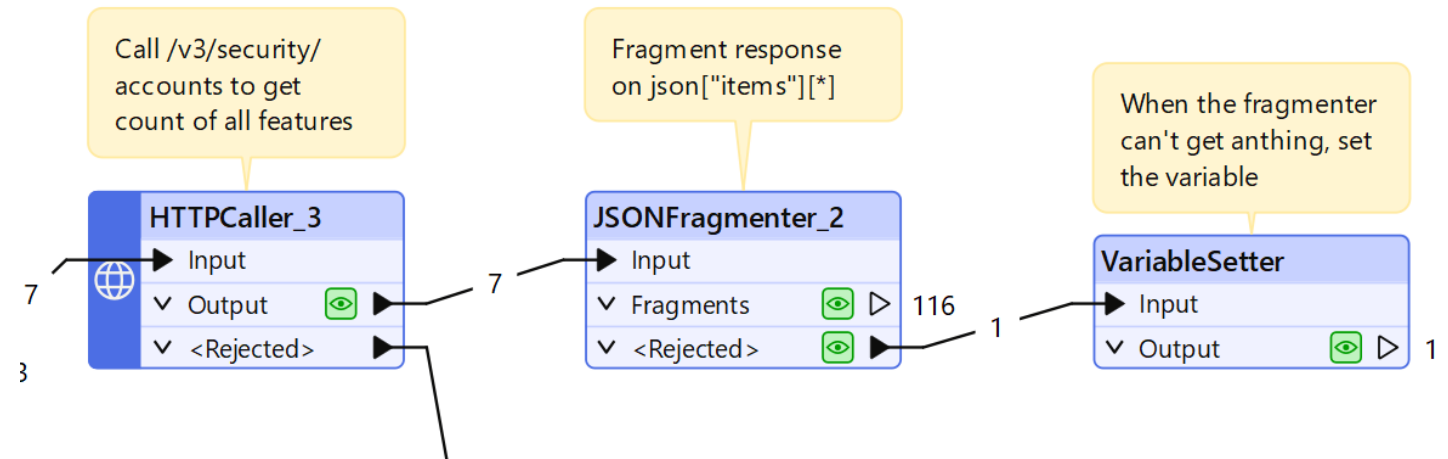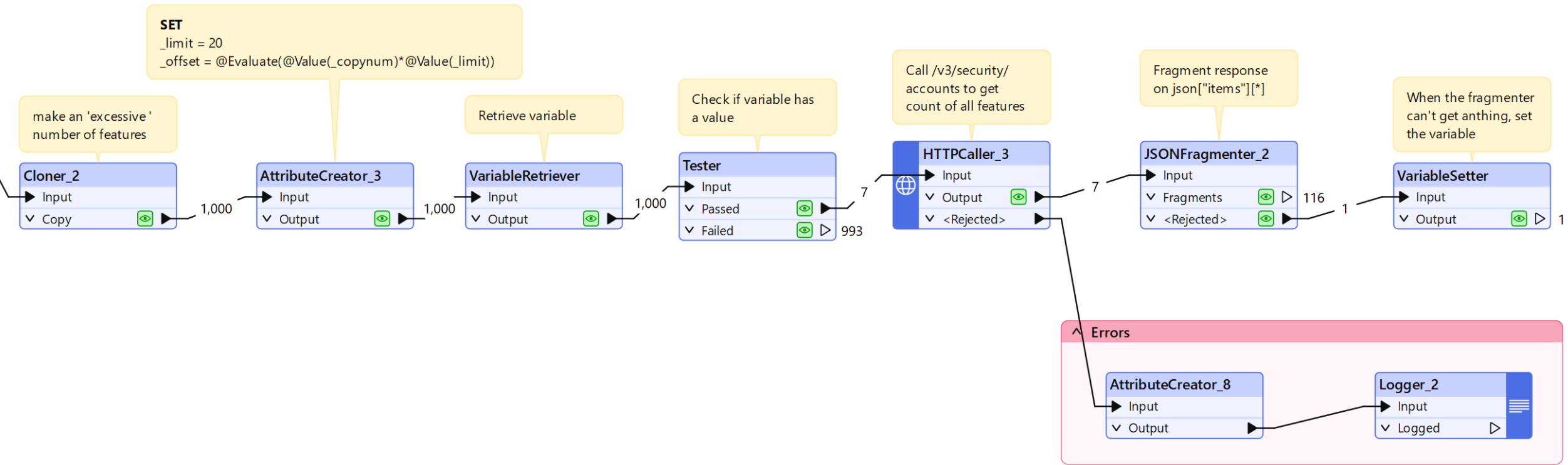
Variables?

When should each one be used?

- Custom Transformers can only have blocking transformers if you export them
- This took the longest to setup, hardest to debug

- Variable approach is better when you're polling, waiting for something to finish
- Need to break out of bulk mode

- Count approach can only work if you know how much data you're expecting
- Count can use HTTPCaller paralisation
- Probably the easiest to understand

Thank You

Hamish.kingsbury@abley.com

abley

Auckland
Level 1/70 Shortland Street
Auckland 1010
Aotearoa New Zealand

Wellington
Level 1/119-123 Featherston Street
Wellington 6011
Aotearoa New Zealand

Christchurch
Level 1/137 Victoria Street
PO Box 36446, Merivale
Christchurch 8146
Aotearoa New Zealand

hello@abley.com
+64 3 377 4703
abley.com