Using Flink to inspect live data as it flows through a data pipeline

Matt Dailey | @matthew_dailey1

April 2, 2019

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

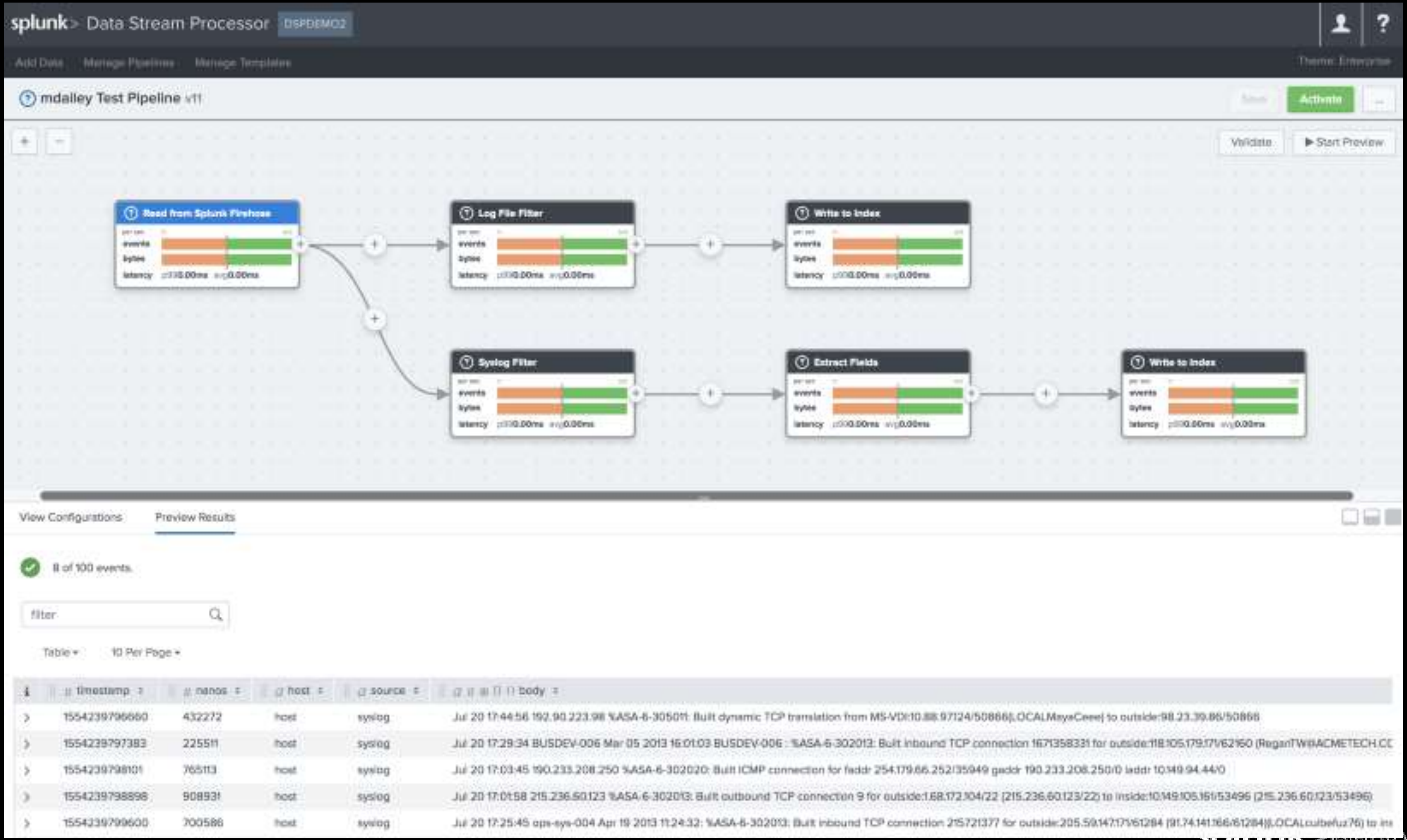
The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> listen to your data

About Me

- Principal Software Engineer @ Splunk
- Using Apache Flink for about 2 years
- Before that, Hadoop ecosystem for 6 years

@matthew_dailey1



timestamp	nanos	host	source	body
1554239796660	432272	host	syslog	Jul 20 17:44:56 192.90.223.98 %ASA-6-305011: Built dynamic TCP translation from MS-VDI:10.88.97.124/50866(LOCAL\MayaCeee) to outside:98.23.39.86/50866
1554239797383	225511	host	syslog	Jul 20 17:29:34 BUSDEV-006 Mar 05 2013 16:01:03 BUSDEV-006 : %ASA-6-302013: Built inbound TCP connection 1671358331 for outside:118.105.179.171/62160 (ReganTW@ACMETECH.CC
1554239798101	765113	host	syslog	Jul 20 17:03:45 190.233.208.250 %ASA-6-302020: Built ICMP connection for faddr 254.179.66.252/35949 gaddr 190.233.208.250/0 laddr 10.149.94.44/0
1554239798898	908931	host	syslog	Jul 20 17:01:58 215.236.60.123 %ASA-6-302013: Built outbound TCP connection 9 for outside:1.68.172.104/22 (215.236.60.123/22) to inside:10.149.105.161/53496 (215.236.60.123/53496)
1554239799600	700586	host	syslog	Jul 20 17:25:45 ops-sys-004 Apr 19 2013 11:24:32: %ASA-6-302013: Built inbound TCP connection 215721377 for outside:205.59.147.171/61284 (91.74.141.166/61284)(LOCAL\culbefuz76) to ins

splunk> listen to your data'

Authors of pipelines would like to know...

Why does my data look like *that* at the end of my pipeline?

What do we need to build to answer this?

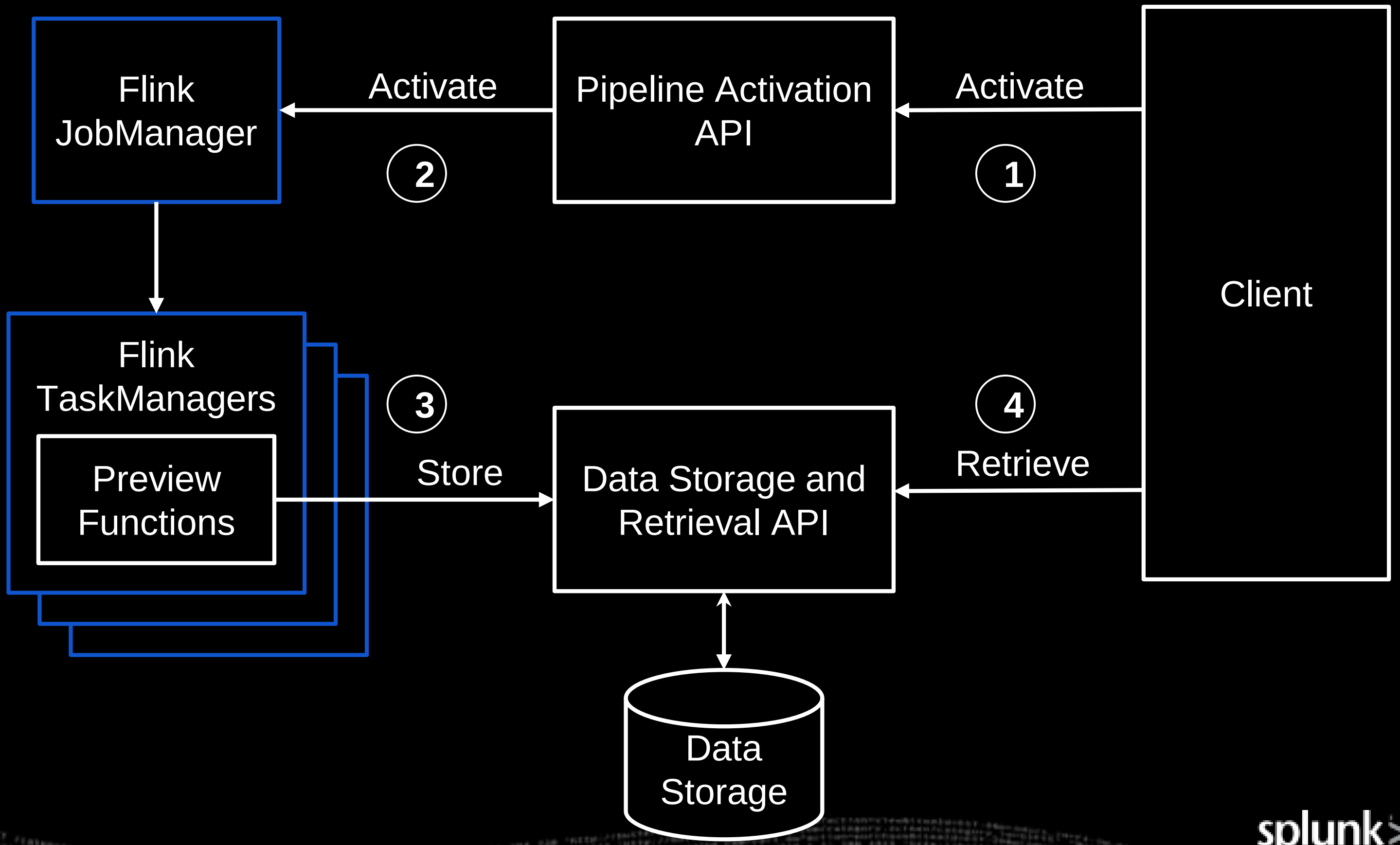
What we need

- A place to display the data
- A way to store the data for display
- A way for a pipeline to send the data to storage
- A way to test the pipelines during authoring
- A name for the feature

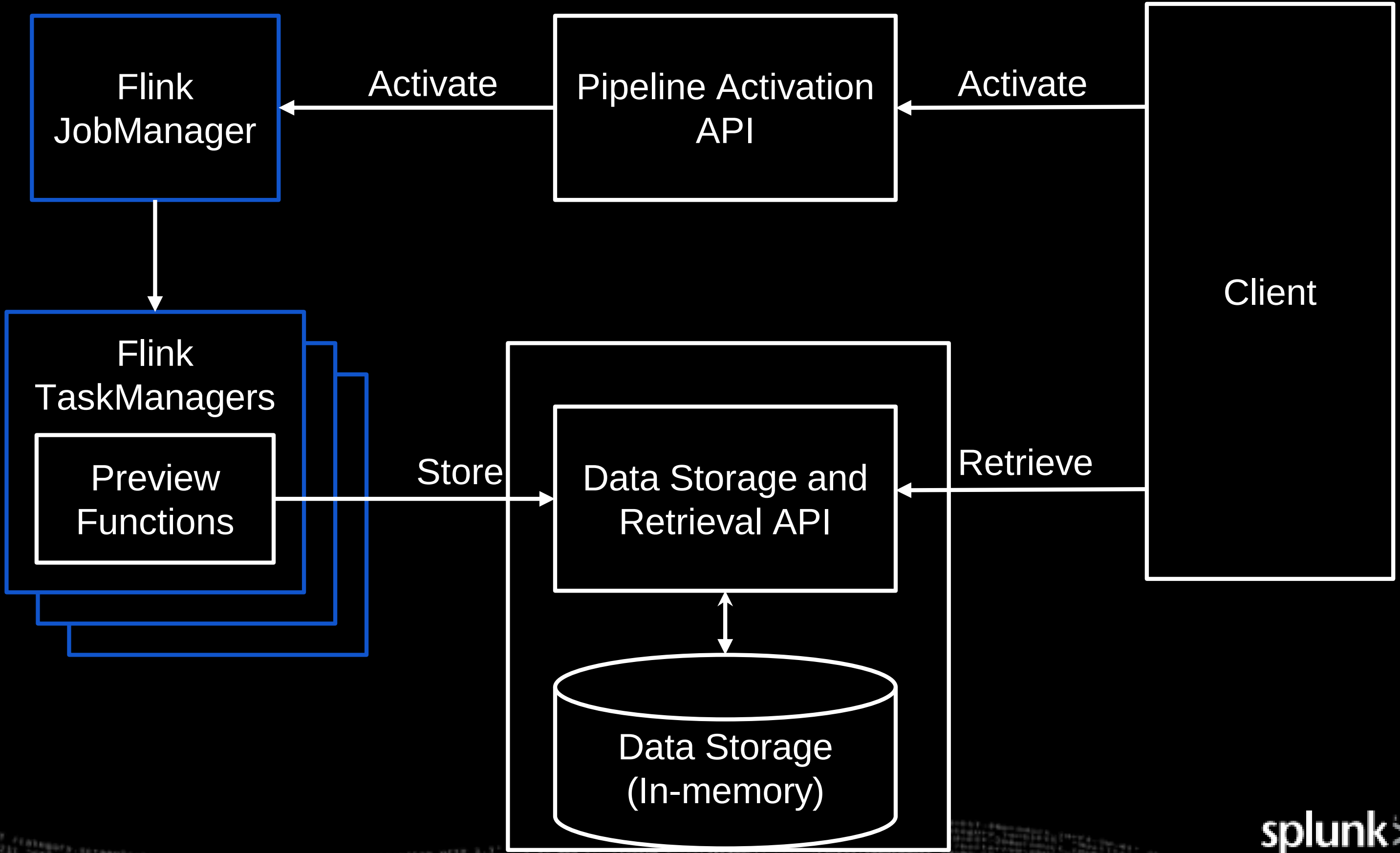
Design

- A place to display the data
  - API and User Interface
- A way to store the data for display
  - Ephemeral storage for ephemeral data
- A way for a pipeline to send the data to storage
  - Function in the pipeline sends data, limits data volume
- A way to test the pipelines during authoring
  - Run a separate pipeline for debugging
- A name for the feature
  - Preview

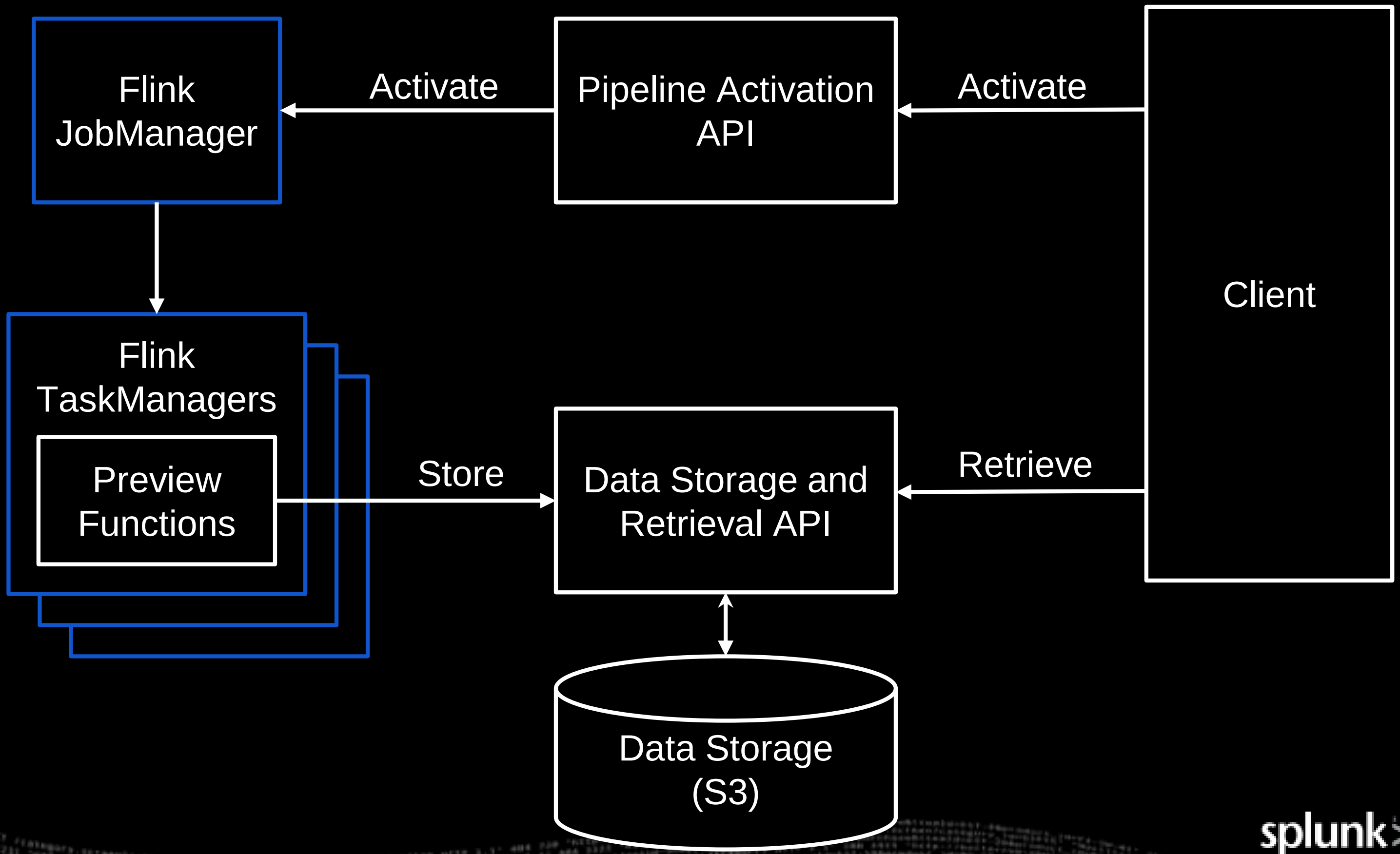
General Architecture



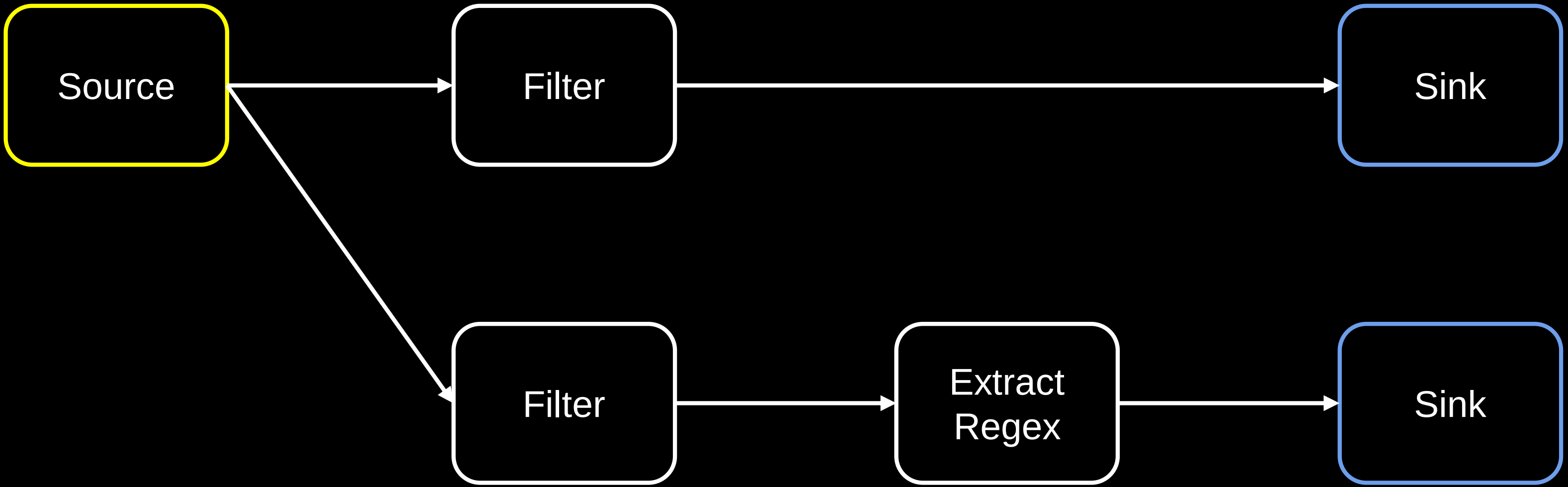
Implementation 1.0



Implementation 1.1

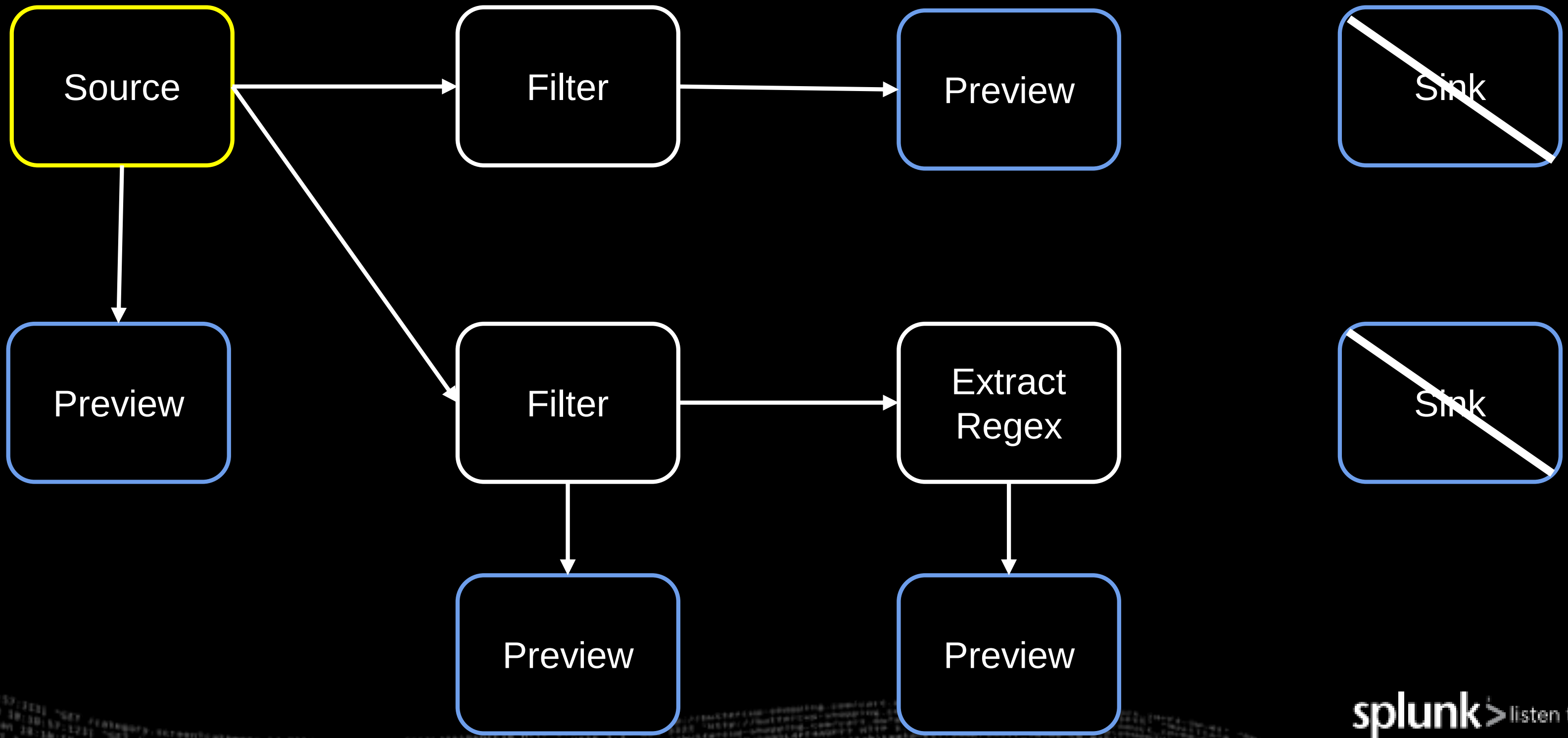


Inserting Preview functions



Inserting Preview functions

*Preview nodes are also sinks



Preview, the function

```
PreviewSink extends RichSinkFunction
open():
  create (HTTP) connection to the Data Storage service

close():
  close connection to the Data Storage service

invoke():
  increment count of records seen
  if preview has hit its upper limit (100), then do nothing
  else, (JSON) serialize the input record and send to the Data Storage service
```

What about checkpoints?

- Checkpoints allow functions to recover state following a failure and restart
- Preview could use checkpoints to store the count of input records
- Without checkpoints, preview functions could send duplicate data

What about savepoints?

- Savepoints are checkpoints you can use to stop and resume your pipeline
- Preview could use savepoints to test against the same data multiple times

What about low-cardinality data?

- Low-cardinality data = data that arrives infrequently, e.g., once a day
- Need to “import data” and run that data through a preview
- This could also work in a framework for automated testing

Can we retrieve data for production pipelines?

- The same architecture should work
- Don’t remove sink nodes
- Think through what data is stored and displayed
  - Probably want *most recent* 100 records

What we made

- A place to display the data
  - API and User Interface
- A way to store the data for display
  - Ephemeral storage for ephemeral data
- A way for a pipeline to send the data to storage
  - Function in the pipeline sends data, limits data volume
- A way to test the pipelines during authoring
  - Run a separate pipeline for debugging
- A name for the feature
  - Preview

Thanks for listening! Questions?

We’re hiring!

Matt Dailey

@matthew_dailey1

splunk>

Acknowledgements

- Joey Echeverria
  - Sergey Sergeev
    - Yassin Mohamed
      - Sharon Xie
        - Bashar Abdul-Jawad
          - Maria Yu