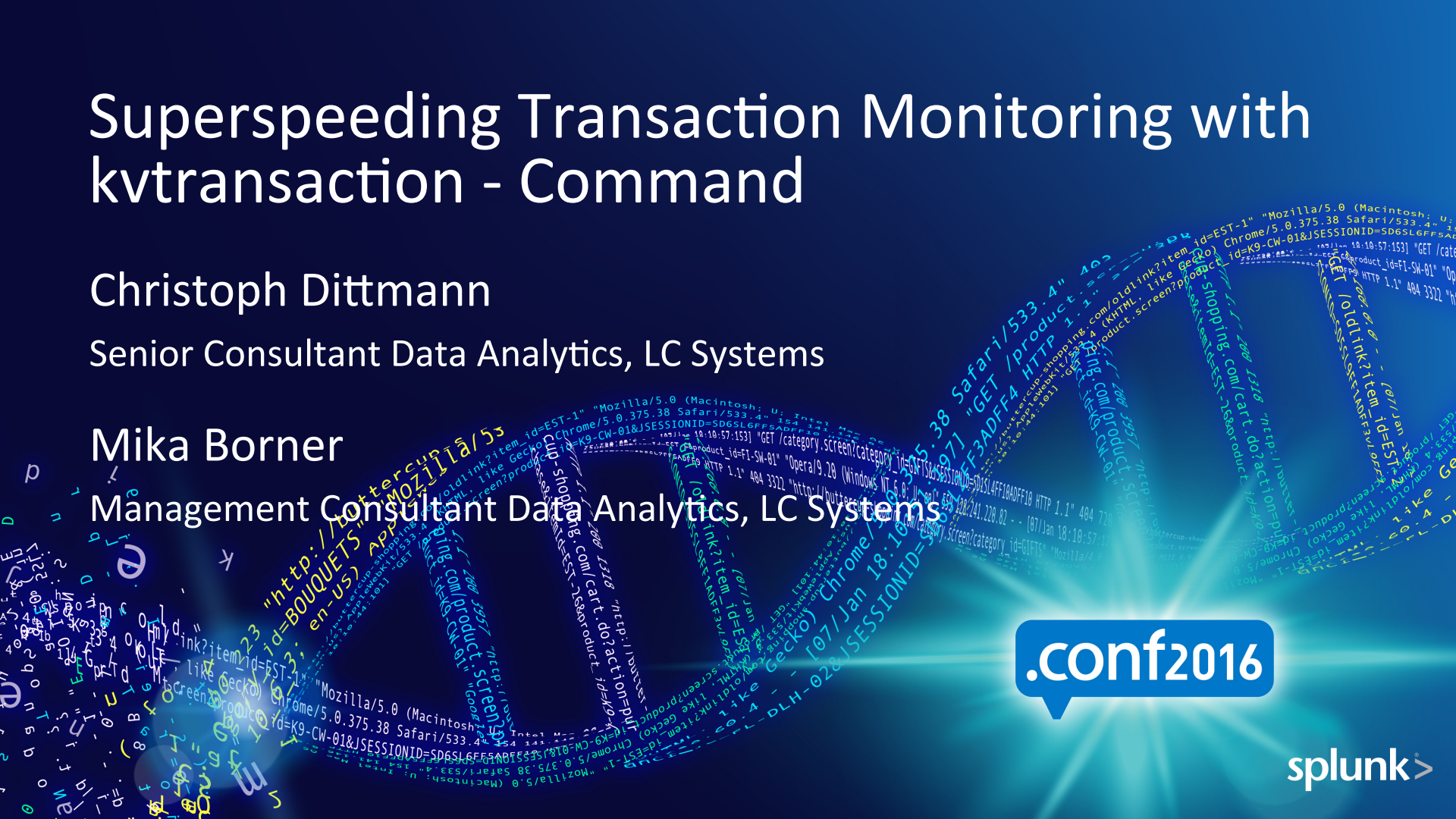
Superspeeding Transaction Monitoring with kvtransaction - Command

Christoph Dittmann

Senior Consultant Data Analytics, LC Systems

Mika Borner

Management Consultant Data Analytics, LC Systems



Agenda

- Intro
- Short Story About Long Transactions
- Where are Transactions Used?
- Anatomy of a Transaction
- Transaction Monitoring with Splunk
- Limitation of Built-In Splunk Commands
- New Approach
- “kvtransaction “– Command
- “kvtransactionoutput” – Command
- Demo
- Wrap up

Presenters

Christoph Dittmann

Learning new Splunk tricks every day.

Twitter: @mulibu_flyingk

Mika Borner

One decade of Splunk experience. Still not enough ☺

Twitter: @my2ndhead

Co-Developer

Harun Küssner

Diving deep, almost drowning in Splunk's possibilities.

Unfortunately but proudly still not using social media.

.conf2016

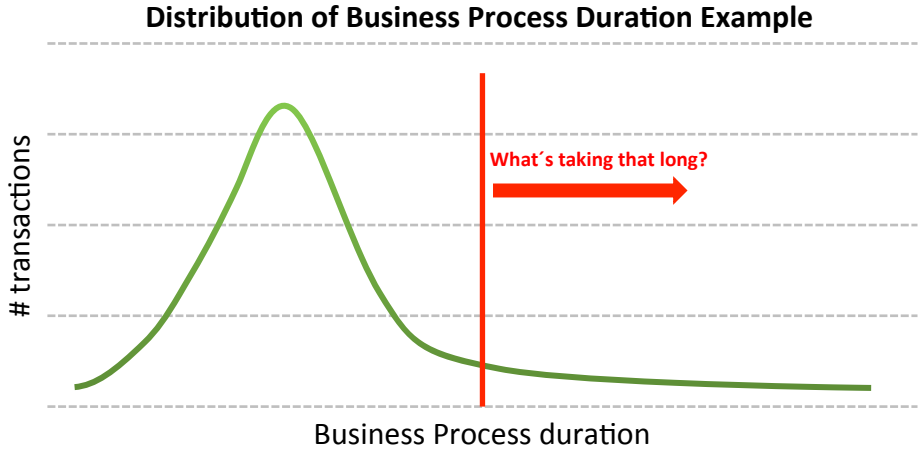
splunk>

Short Story About Long Transactions

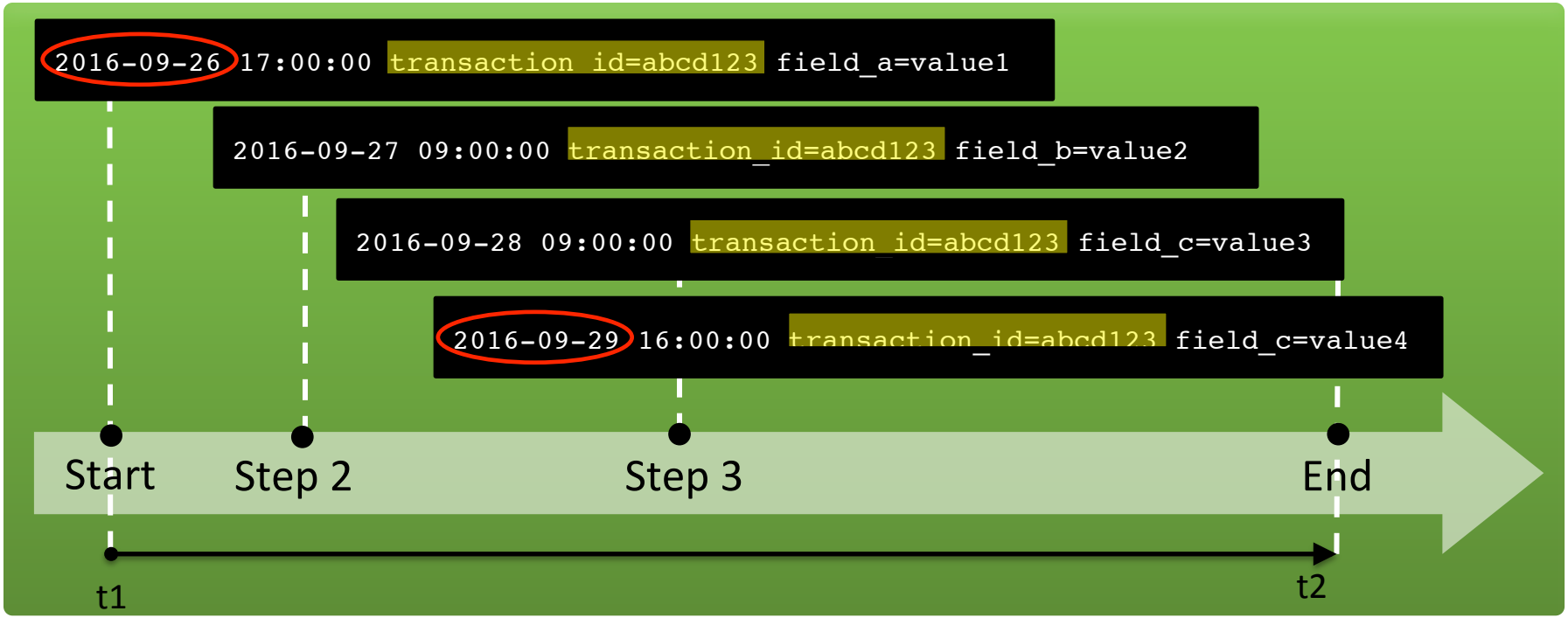
- The number of projects for **transaction monitoring** has been growing constantly at our customers
- Transaction monitoring is doable within Splunk, but the performance and complexity of the implementation is often not satisfying
- A new way how to monitor transactions had to be found
- We came up with a new custom search command that uses the KV store as a state table to track transactions
- The commands should make transaction monitoring fly!

Where Are Transactions Used?

- Session Monitoring
  - Connections (SMTP, IMAP4, POP3)
  - Sessions (Web Application, Login Session, DHCP)
- Process Monitoring
  - Business Processes



Anatomy of a Transaction



Transaction Monitoring with Splunk

- Splunk provides two commands to group events:
  - transaction
  - stats
- General rule of thumb:
  - Use stats when possible, as it is faster (Map-Reducible)
  - Use transaction, when unique field values are not sufficient to discriminate between discrete transactions

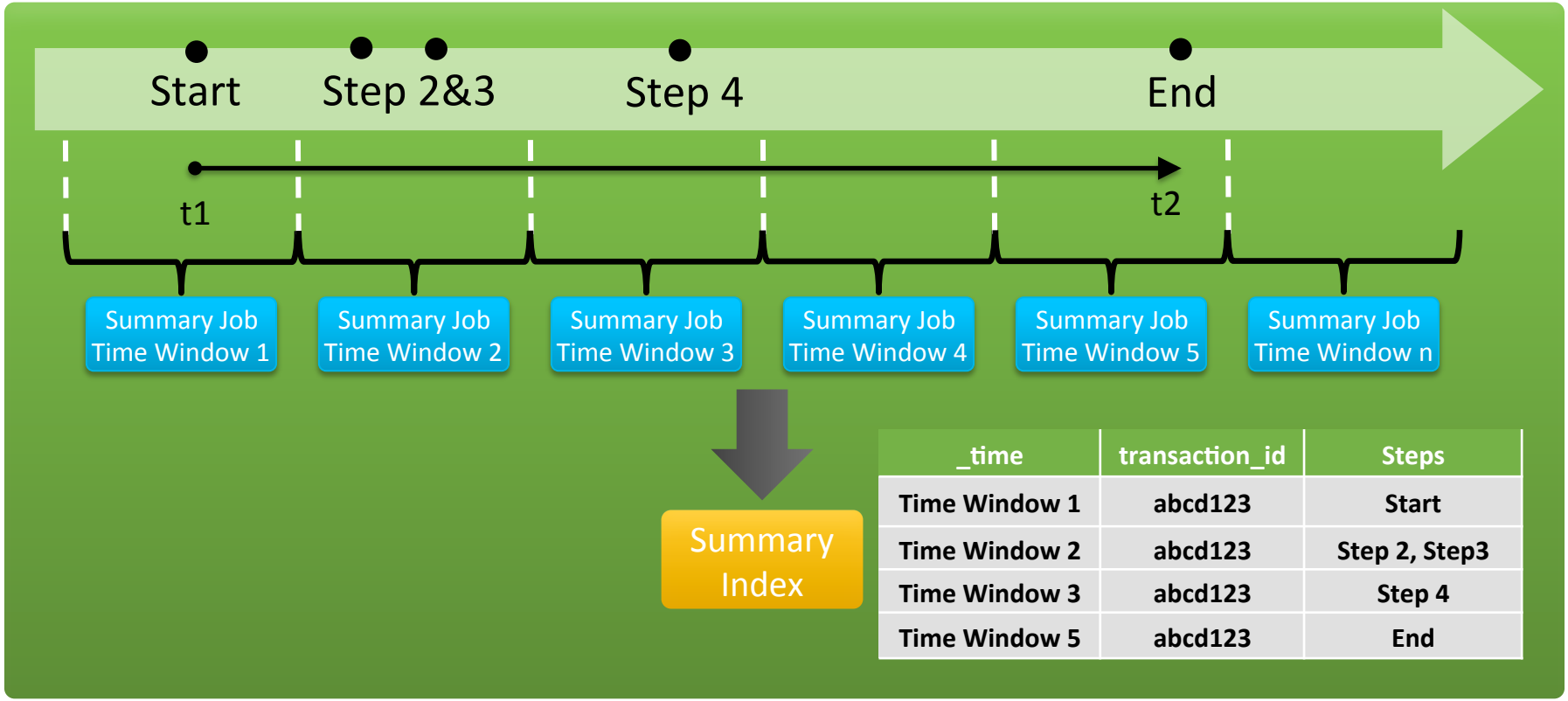
Limitation of Built-In Splunk Commands

- The transaction and stats commands usually work well, when the number of events is not too high
- Number of events can be very high when transactions go over hours or even days!
- The transaction command is usually factors slower than stats, as a lot of work is done on the search head
- The stats command is a streaming command and typically map reduces pretty well
- The stats command still has it's limitations
  - Memory limits can be hit (limits.conf has to be tuned)
  - Reduce phase is single-threaded and can be slow with millions of pre-stats results

Limitation of Built-In Splunk Commands (2)

- Summary Indexing has been used to speed up reporting with mixed success
- Each summarization job runs periodically over a moving time window
- The larger the time window, the longer it takes to aggregate the transactions
- The shorter the time window, the more transactions are cut in half. Instead of writing one complete event for a transaction into the summary index, we may get multiple partial transactions. These partial transactions have to be again aggregated at search time.

Summary Indexing with Stats/Transaction



_time	transaction_id	Steps
Time Window 1	abcd123	Start
Time Window 2	abcd123	Step 2, Step3
Time Window 3	abcd123	Step 4
Time Window 5	abcd123	End

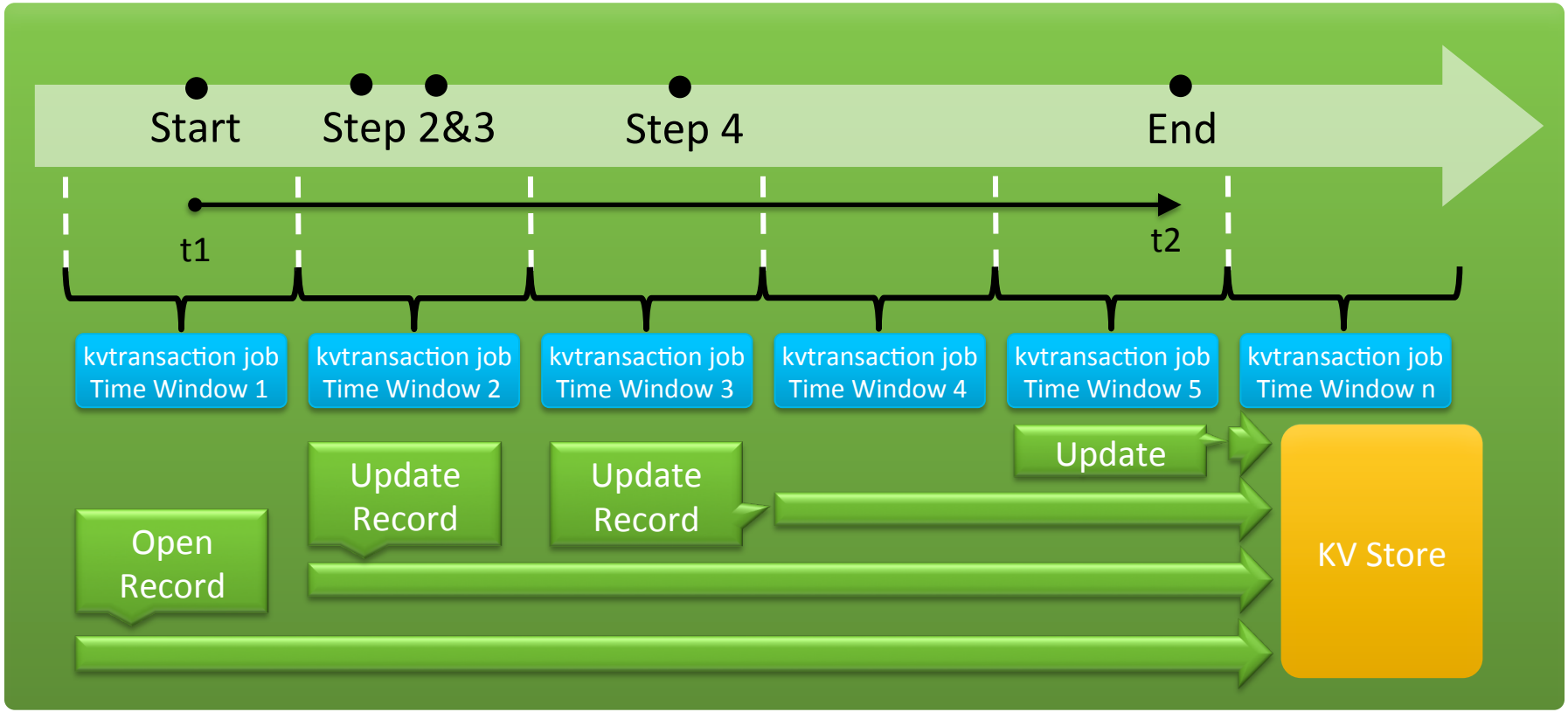
New Approach

- Even though Summary Indexing may mitigate performance issues, it is not possible to aggregate the transaction in the most dense way as possible
- The summarized data has again be aggregated during search
- This is because data can only be added to the summary index and not updated
- The only way to work around this, is to store transactions in a state table that keeps track of new transactions and their state

kvtransaction Command

- The new kvtransaction command runs periodically over a moving time window
- kvtransaction Searches for new/existing transaction identifiers
- If a new transaction identifier is seen, it will open a new record in a KV store collection
- If a new event for an existing transaction is found, the existing KV store record will be updated

kvtransaction Command



kvtransaction Command Syntax

kvtransaction [testmode=*boolean*] [mvlist=<boolean|list>]
[mvdedup=*boolean*]
transaction_id=*fieldname* collection=*collection* [fields]

testmode = If false, write to KV Store collection
mvlist = If true, keeps all seen field values. If false, keep the last non-null value
mvdedup = If true, only keep unique field values
transaction_id = Specifies the field containing the transaction identifier
collection = The kv store collection where transactions are stored
fields = Optionally filter fields

kvtransaction Command Example 1

```
2016-09-26 17:00:00 transaction_id=abcd123 field_a=value1
2016-09-27 09:00:00 transaction_id=abcd123 field_b=value2
2016-09-28 09:00:00 transaction_id=abcd123 field_c=value3
2016-09-29 16:00:00 transaction_id=abcd123 field_d=value4
```

<myevents>

|kvtransaction transaction_id=transaction_id collection=mytxncollection

_time	transaction_id	field_a	field_b	field_c	field_d	duration	eventcount
2016-09-26	abcd123	value1	value2	value3	value4	342000	4

kvtransaction Command Example 2

```
2016-09-26 17:00:00 transaction_id=abcd123 field_a=aaa
2016-09-27 09:00:00 transaction_id=abcd123 field_a=bbb
2016-09-28 09:00:00 transaction_id=abcd123 field_b=ccc
2016-09-29 16:00:00 transaction_id=abcd123 field_b=ccc
```

<myevents>
|kvtransaction **mvlist=true mvdedup=true**
transaction_id=transaction_id collection=mytxncollection

_time	transaction_id	field_a	field_b	duration	eventcount
2016-09-26	abcd123	aaa bbb	ccc	342000	4

kvtransactionoutput Command

- The KV Store collection will grow indefinitely
- kvtransactionoutput mitigates the problem of an ever growing KV
  - Reading from KV Store and writing closed transactions to an index
  - Purging transactions written to index from KV store

Syntax*

kvtransactionoutput [testmode=*boolean*] [minevents=*integer*] [action=<copy|move|flush>] index=*index* collection=*collection*

* please read the documentation for a complete list of options

Demo

Wrap Up

- kvtransaction Command does have limitations as well
  - Maximum amount of events per minute
  - Maximum amount of (open) transactions in KV Store
- kvtransaction Is well suited for business processes which tend to take longer
- Reporting over longer time periods benefits from the use of kvtransaction command
- kvtransaction output can be a great input for Business Process Dashboards or other applications (e.g. Splunk ITSI, Splunk ES)

What's Next

- Currently in closed beta
- Making app available on splunkbase
- Applying to be a certified community support app
- Uploading source code to github

THANK YOU

splunk>