Revealing the Magic

How Splunk 7.2 improved search performance by 50%

Kellen Green | Senior Software Engineer

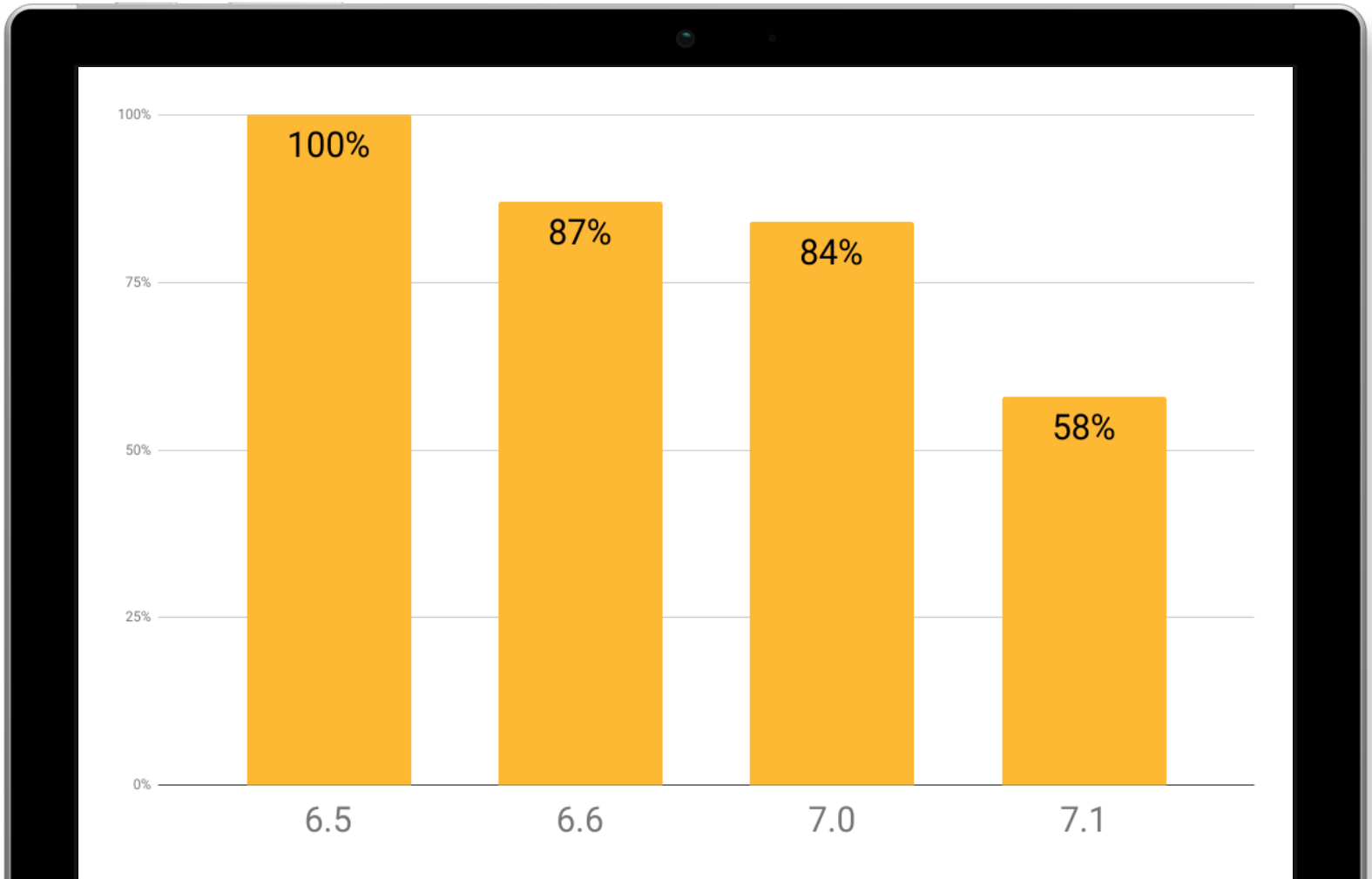
October 2018

A little bit about myself

6+ years at Splunk

Historical Performance

Since .conf2016



- We're nearing a 50% decrease in search times since Splunk 6.5
- Based out off internal test suite of 80 searches ranging in:
  - Density
  - Commands
  - Duration
    - 1s to 30 minutes

splunk> .conf18

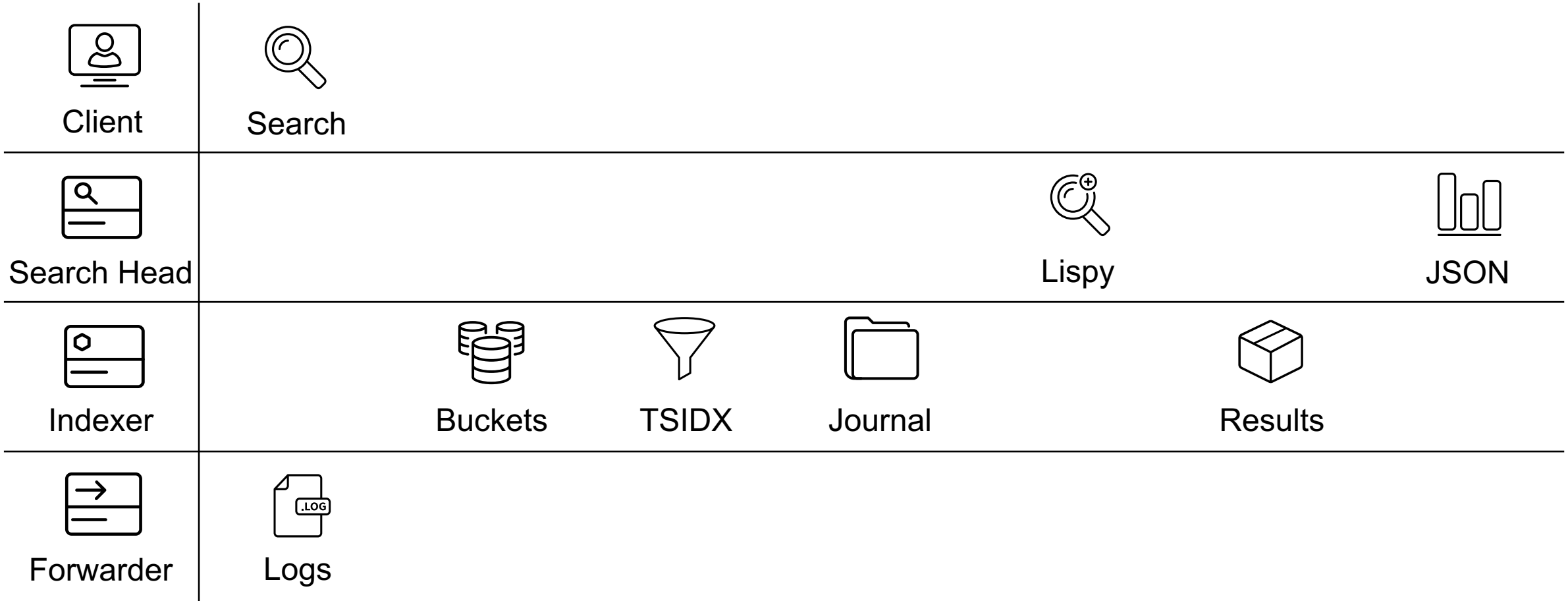
Todays Agenda

Pretty Simple

1. Splunk is way faster in 2018 than 2016
2. How did we achieve this
3. How can you take advantage of it

Quick Recap

A Splunk Search in Action



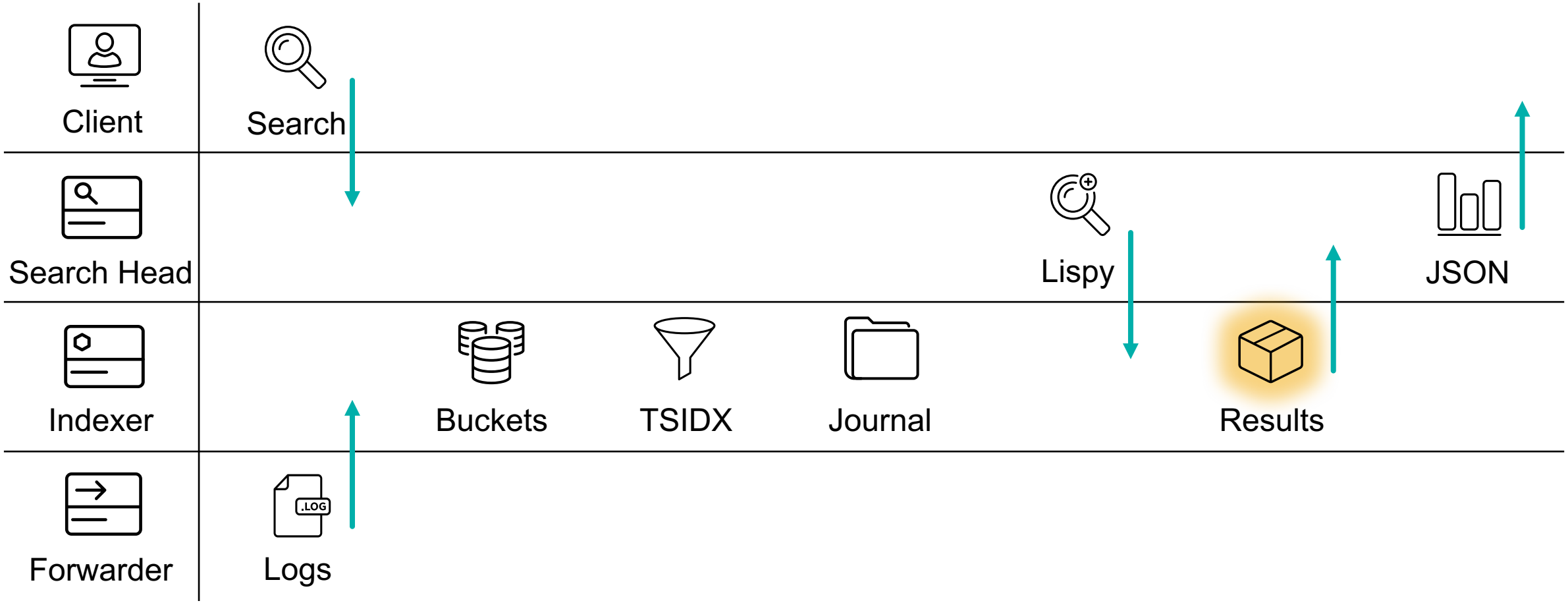
Extra Credit

Shameless Plug

- A lot of detail was glossed over in the previous slide
- Check out my .conf17 talk for a deep dive on each stage of the search process
- https://youtu.be/vo900c8nmKs

Improvement #1

Data Serialization



Data Serialization

In Splunk 7.1

- Indexers forward search results to the Search Head utilizing the CSV file format
- CSV is text based
  - Everything needs to be converted into strings
- CSV is verbose
  - Those commas can add up, not the kind of BIG DATA we're after
  - Yes it's compressed but this eats into encoding time

Search Results Serialization

New to Splunk 7.2

- Introducing the SRS file format
- New binary serialization format for search results
- Aims to solve the short comings of CSV files
  - Transmission file size
  - Encoding performance
  - Time to first read

Search Results Serialization

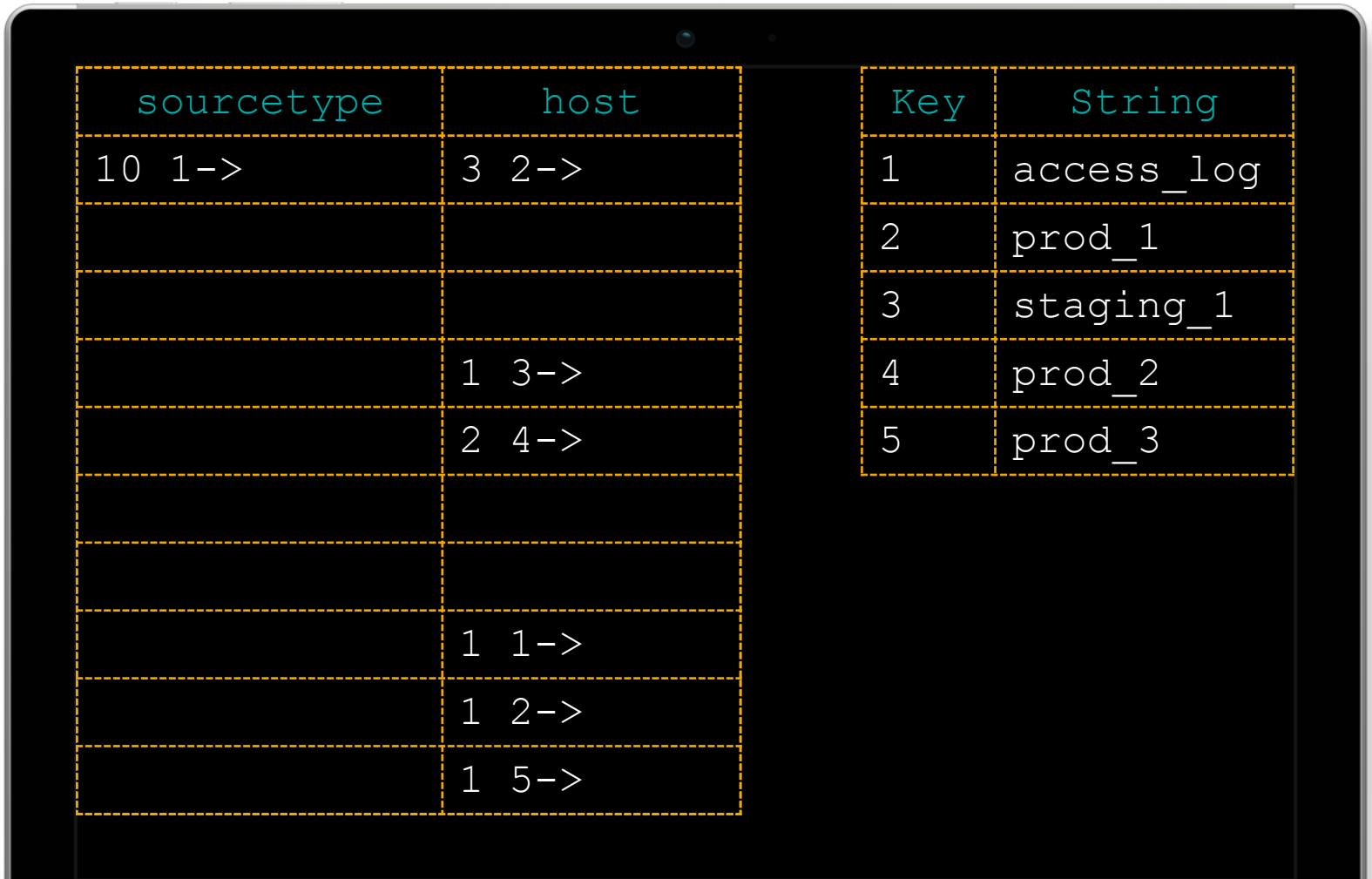
Repeating Rows

time	sourcetype	host	path
153231524	10 access_log	3 prod_1	2 /cart
153232352			
153233432			1 /about
153243455		1 staging_1	3 /home
153235768		3 prod_2	
153226454			
153237564			1 /cart
153238098		2 prod_1	1 /info
153239887			1 /item/9
153240854		1 prod_3	1 /item/2

- Values within fields often repeat
  - `status=200`
  - `method=GET`
- SRS dedupes repeated values
- Leads to the compressor having to do less work

Search Results Serialization

String Pools



sourcetype	host
10 1->	3 2->
	1 3->
	2 4->
	1 1->
	1 2->
	1 5->

Key	String
1	access_log
2	prod_1
3	staging_1
4	prod_2
5	prod_3

- To further reduce file size SRS utilizes string pools
- Pools contain unique strings by lookup id
- Events then point pool for string reference
- Leads to much less ram utilization during decode

Search Results Serialization

Event Type Support

- SRS retains type information for event values
  - Strings, Integers, Floats, Multivalued
- Encodes numbers with variable length encoding (LEB128)
  - 1532355229 encodes to 40 bits as a number
  - "1532355229" needs 80 bits when left as a string
- Leads to large gains in encode and decode performance
  - Float types will see the greatest benefits

Search Results Serialization

Column Major

time	host	sourcetype	path
1532315243	prod_1	access_log	/cart
1532323529	prod_1	access_log	/cart
1532334325	prod_1	access_log	/about
1532343455	staging_1	access_log	/home
1532357686	prod_2	access_log	/home
1532264543	prod_2	access_log	/home
1532375644	prod_2	access_log	/cart
1532380988	prod_1	access_log	/info
1532398876	prod_1	access_log	/item/987
1532408547	prod_3	access_log	/item/125

- SRS serializes in column-major fashion
- Splunk doesn't need to wait for all rows to be sent before working on a column
- Analogous nature of column data leads to compression benefits

Search Results Serialization

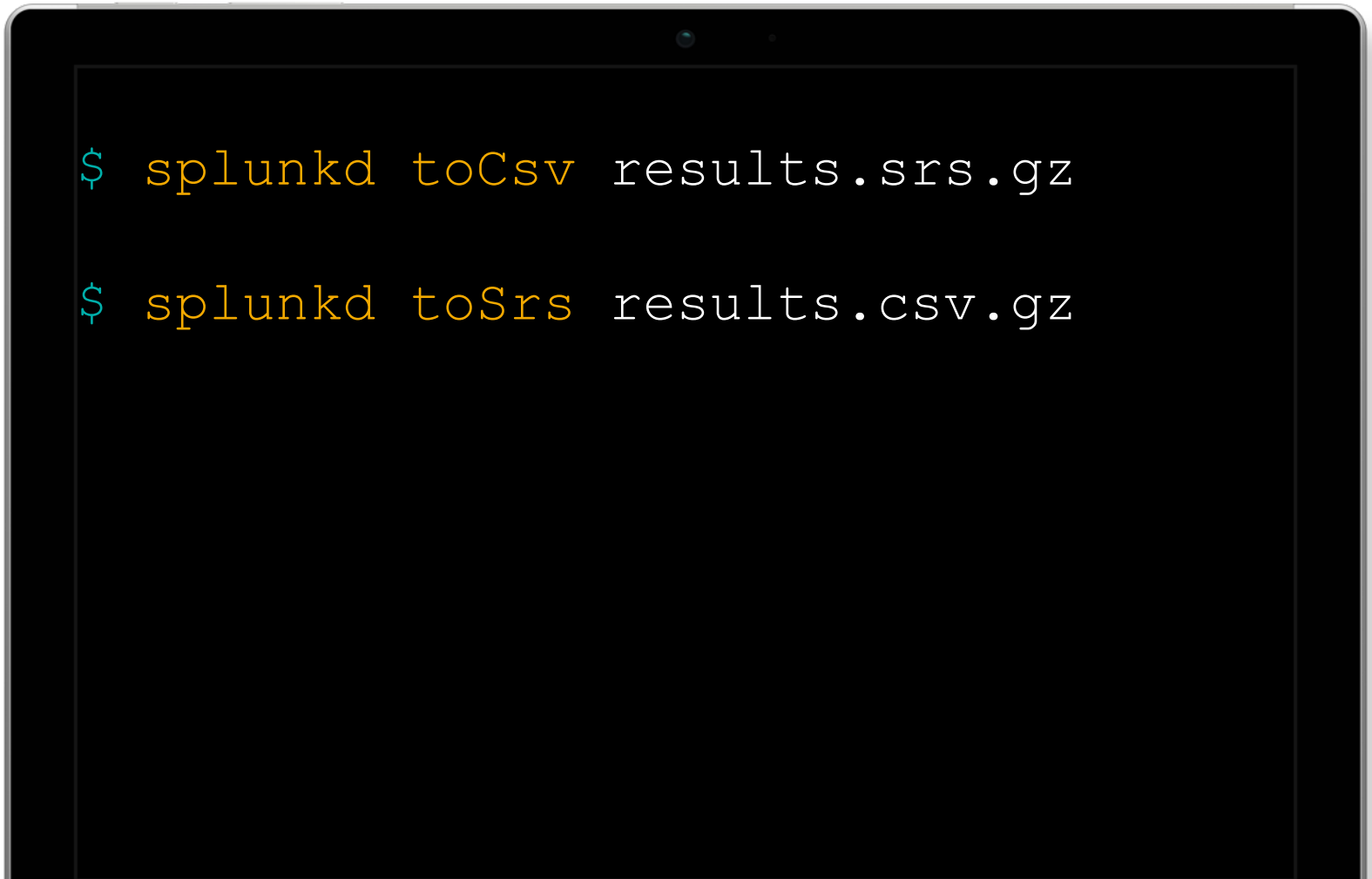
See for Yourself

```
$ cd var/run/splunk/dispatch/$sid
$ ls -l
    events
    generate_preview
    info.csv
    metadata.csv
    peers.csv
    remote_logs
    results.csv.gs > results.srs.gs
    search.log
```

- SRS can be seen in the dispatch folder on the Search Head
- `csv` extension from results is replaced with `srs`

Search Results Serialization

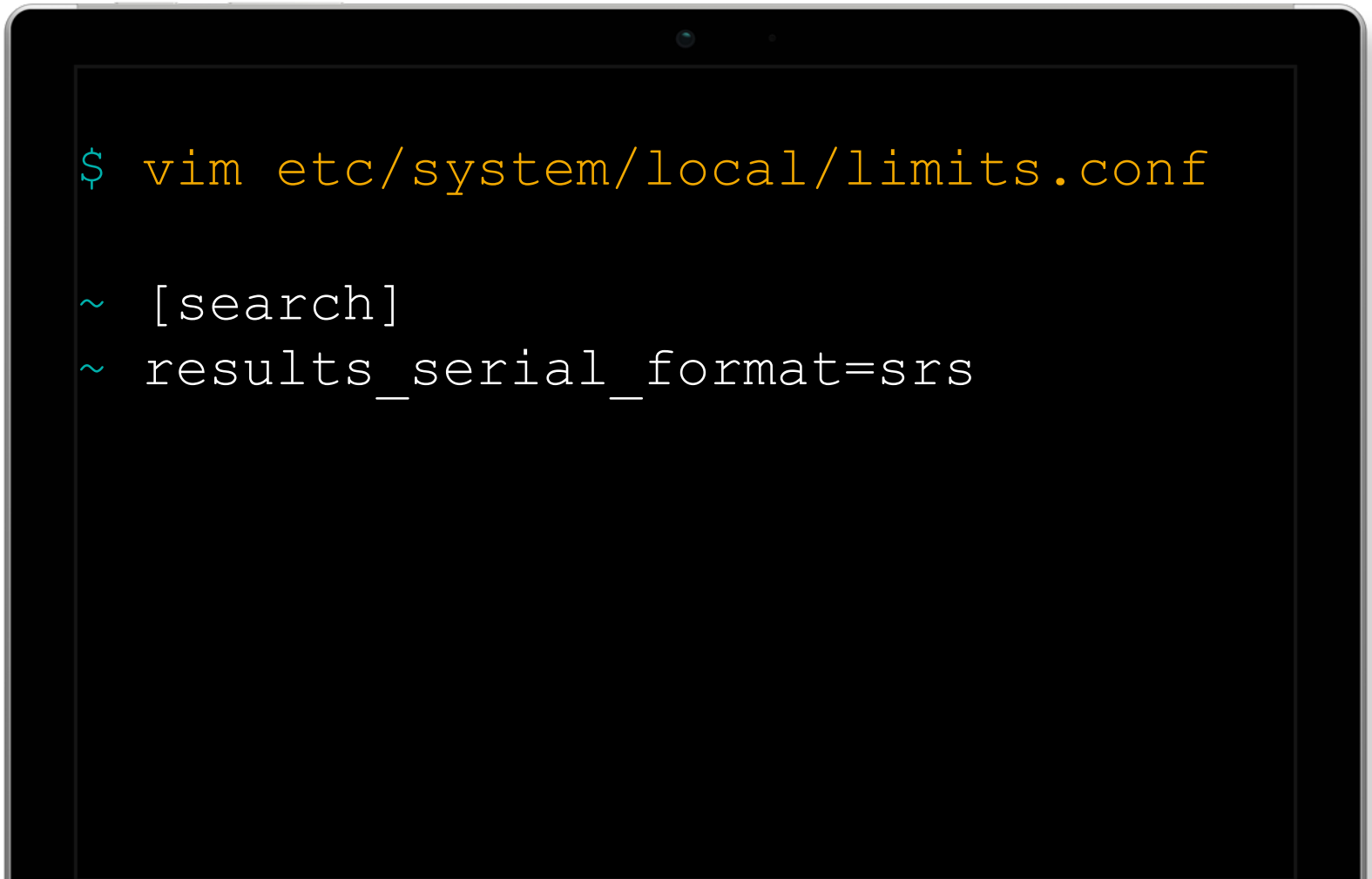
New Commands



- SRS is not readily parsable
- New commands to convert between the two result types
- Useful for scripts which did post search processing on CSV results

Search Results Serialization

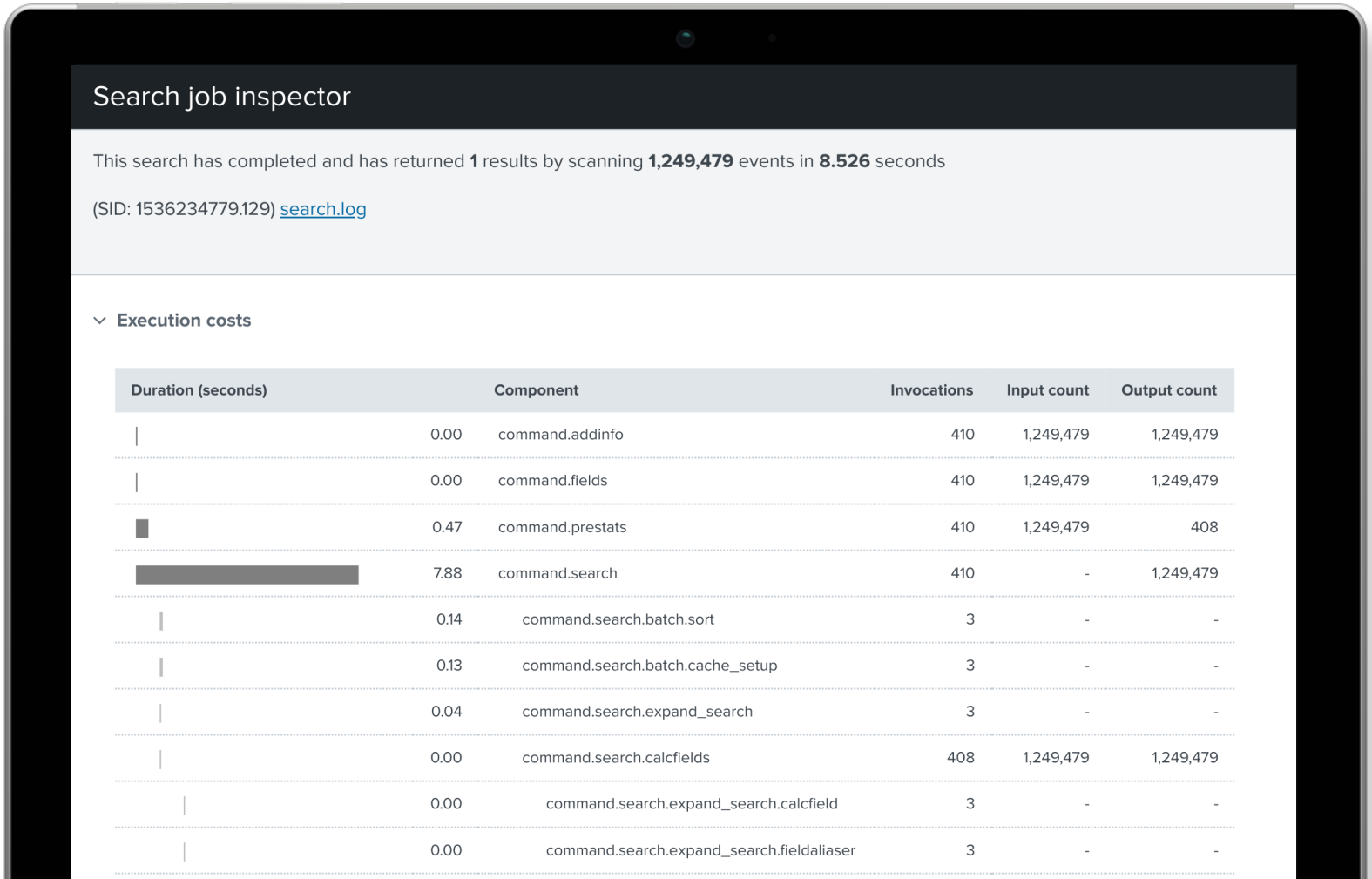
Enabled by Default



- SRS is enabled by default in Splunk 7.2
- Togglable through `limits.conf` file
- Both Indexer and Search Head need to be enabled
  - Graceful fallback to csv otherwise

Search Results Serialization

Benchmarks



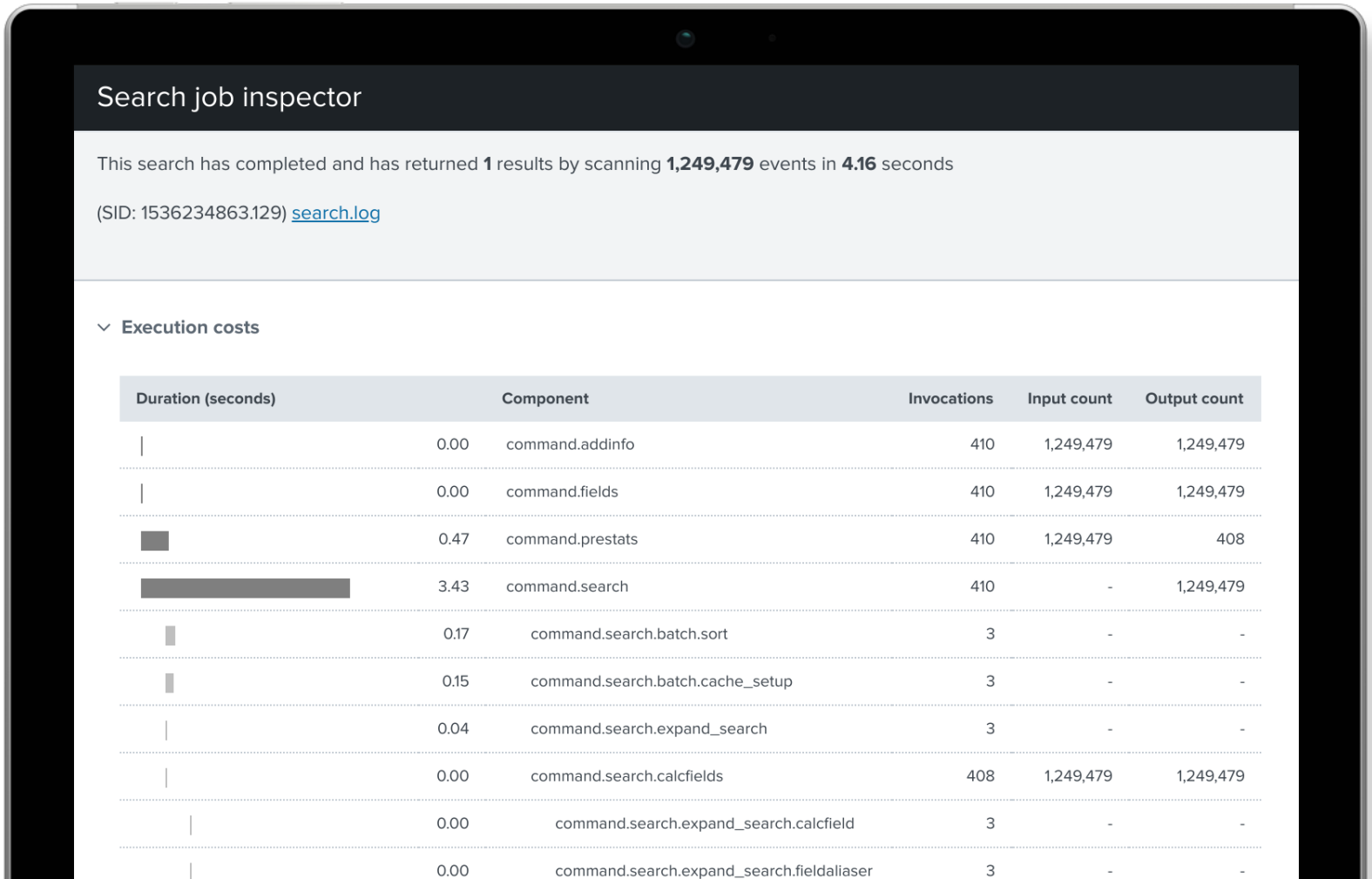
Duration (seconds)	Component	Invocations	Input count	Output count
0.00	command.addinfo	410	1,249,479	1,249,479
0.00	command.fields	410	1,249,479	1,249,479
0.47	command.prestats	410	1,249,479	408
7.88	command.search	410	-	1,249,479
0.14	command.search.batch.sort	3	-	-
0.13	command.search.batch.cache_setup	3	-	-
0.04	command.search.expand_search	3	-	-
0.00	command.search.calcfields	408	1,249,479	1,249,479
0.00	command.search.expand_search.calcfield	3	-	-
0.00	command.search.expand_search.fieldaliaser	3	-	-

- Basic `|stats sum` search over 1 million events
- 8 seconds to complete on Splunk 7.2 with SRS disabled

splunk> .conf18

Search Results Serialization

Benchmarks



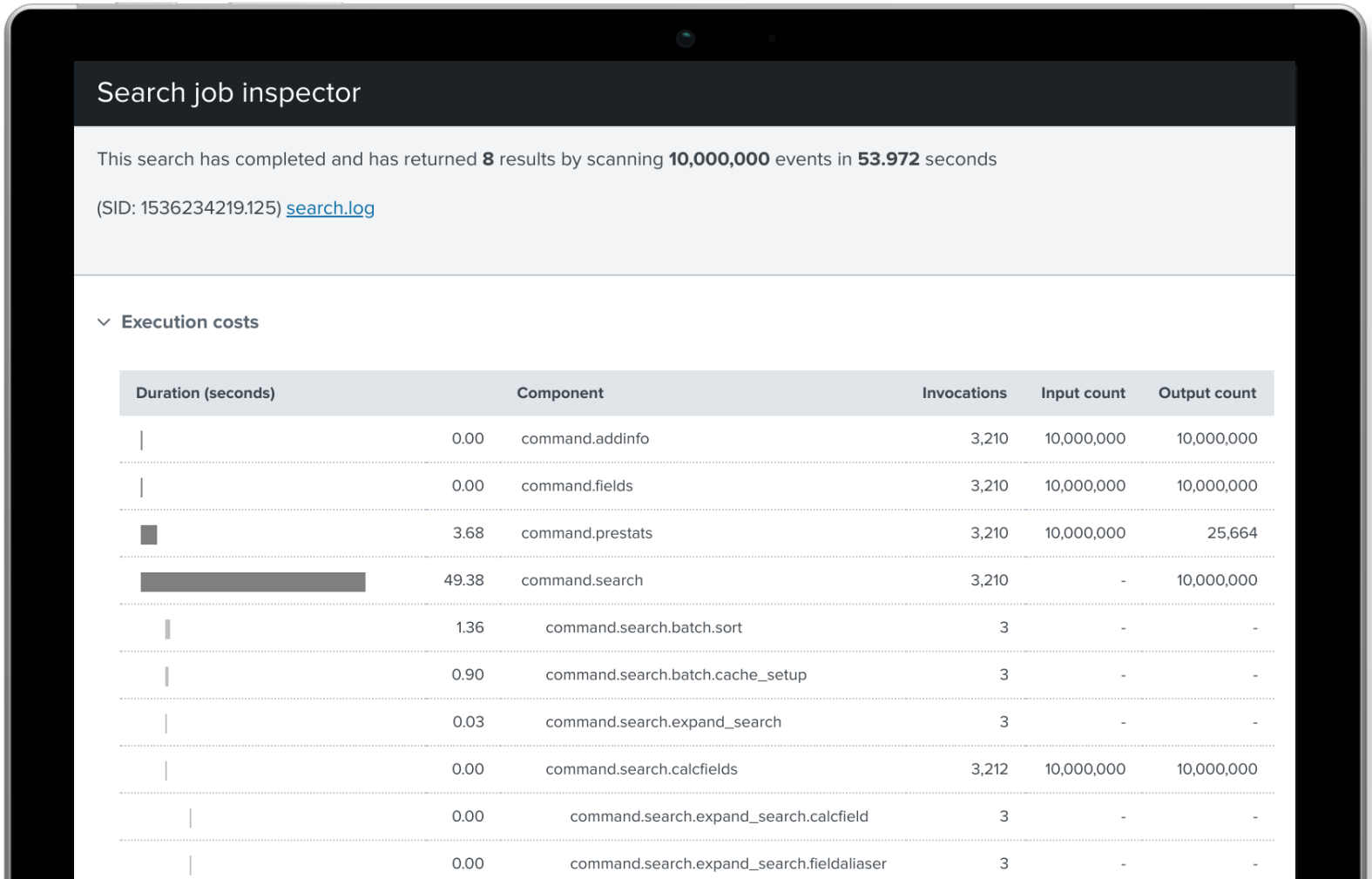
Duration (seconds)	Component	Invocations	Input count	Output count
0.00	command.addinfo	410	1,249,479	1,249,479
0.00	command.fields	410	1,249,479	1,249,479
0.47	command.prestats	410	1,249,479	408
3.43	command.search	410	-	1,249,479
0.17	command.search.batch.sort	3	-	-
0.15	command.search.batch.cache_setup	3	-	-
0.04	command.search.expand_search	3	-	-
0.00	command.search.calcfields	408	1,249,479	1,249,479
0.00	command.search.expand_search.calcfield	3	-	-
0.00	command.search.expand_search.fieldaliaser	3	-	-

- SRS cuts this same search down to 4 seconds
- Over a 50% reduction in search time

splunk> .conf18

Search Results Serialization

Benchmarks



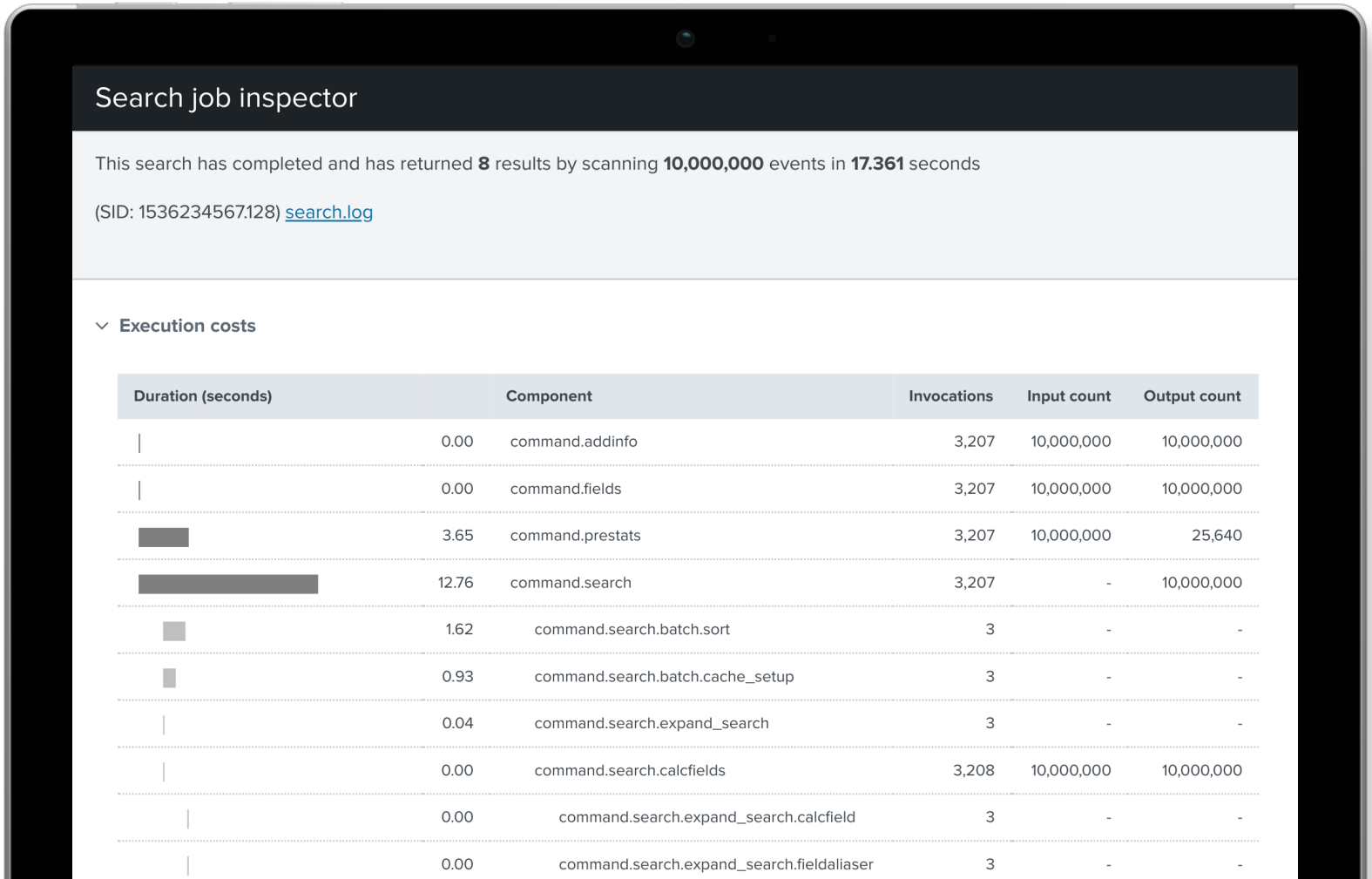
Duration (seconds)		Component	Invocations	Input count	Output count
	0.00	command.addinfo	3,210	10,000,000	10,000,000
	0.00	command.fields	3,210	10,000,000	10,000,000
	3.68	command.prestats	3,210	10,000,000	25,664
	49.38	command.search	3,210	-	10,000,000
	1.36	command.search.batch.sort	3	-	-
	0.90	command.search.batch.cache_setup	3	-	-
	0.03	command.search.expand_search	3	-	-
	0.00	command.search.calcfields	3,212	10,000,000	10,000,000
	0.00	command.search.expand_search.calcfield	3	-	-
	0.00	command.search.expand_search.fieldaliaser	3	-	-

- Same search as before but this time with 10 million events
- Nearing a minute to complete with SRS disabled

splunk> .conf18

Search Results Serialization

Benchmarks



Duration (seconds)		Component	Invocations	Input count	Output count
	0.00	command.addinfo	3,207	10,000,000	10,000,000
	0.00	command.fields	3,207	10,000,000	10,000,000
	3.65	command.prestats	3,207	10,000,000	25,640
	12.76	command.search	3,207	-	10,000,000
	1.62	command.search.batch.sort	3	-	-
	0.93	command.search.batch.cache_setup	3	-	-
	0.04	command.search.expand_search	3	-	-
	0.00	command.search.calcfields	3,208	10,000,000	10,000,000
	0.00	command.search.expand_search.calcfield	3	-	-
	0.00	command.search.expand_search.fieldaliaser	3	-	-

- 17 seconds…
- :)
- :0
- :p

splunk> .conf18

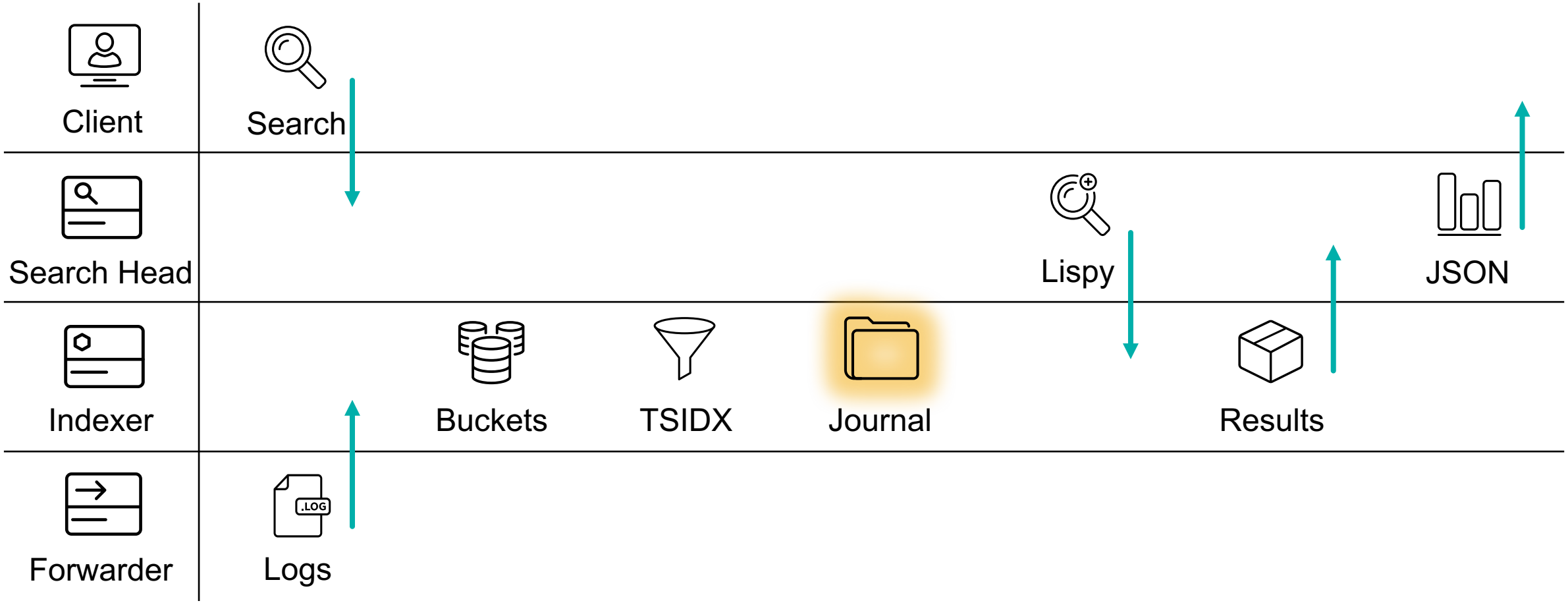
Search Results Serialization

Performance Summary

- Average 10% reduction in search times across the board
  - I think marketing is being modest, 50% seems readily achievable
- Huge performance benefits on searches with high cardinality
  - `| stats`
  - `| tstats`
- Performance will grow along with the size of your dataset

Improvement #2

Compression



Compression in Splunk

Gzip From the Beginning

- Splunk compresses raw event data on disk within the Journal
- Gzip has always been the default
  - First introduced in 1992
- L4Z compression option was added in Splunk 6.3
  - Increases compression rates by 6x
  - Increases decompression rates by 9x
  - Comes at the cost of 30% larger files
    - This tended to nullify any performance benefit

Zstandard Compression

New Compression Option

- Announcing Zstandard compression support for Splunk 7.2
  - Developed by Facebook in 2015
- Average 5% space savings over Gzip
- Compression rate 4x improvement over Gzip
- Decompression rate 3x improvement over Gzip

Zstandard Compression

Let's Enable

```
$ cd etc/apps/$app/local
$ vim index.conf

~ [$index]
~ journalCompression=zst
```

- Disabled by default for initial release
- Can be enabled in `index.conf`
- Only changes the compression algorithm for newly created buckets

Zstandard Compression

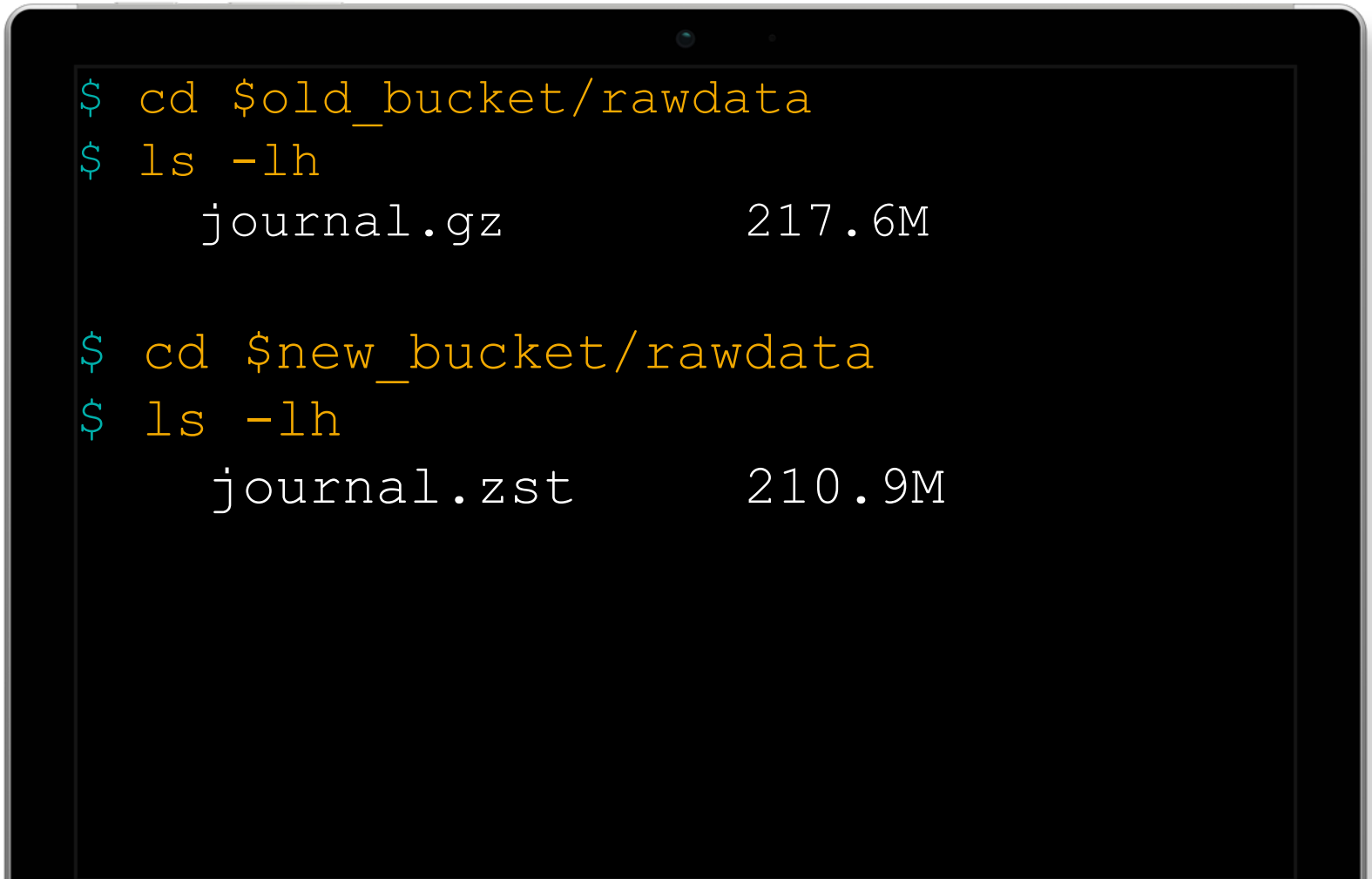
See for Yourself

```
$ cd $index/db/$bucket/rawdata
$ ls -l
    journal.gz > journal.zst
    slicemin.dat
    slicesv2.dat
```

- New journals can be seen in the buckets rawdata folder on the Indexer
- `gz` extension from journal is replaced with `zst`

Zstandard Compression

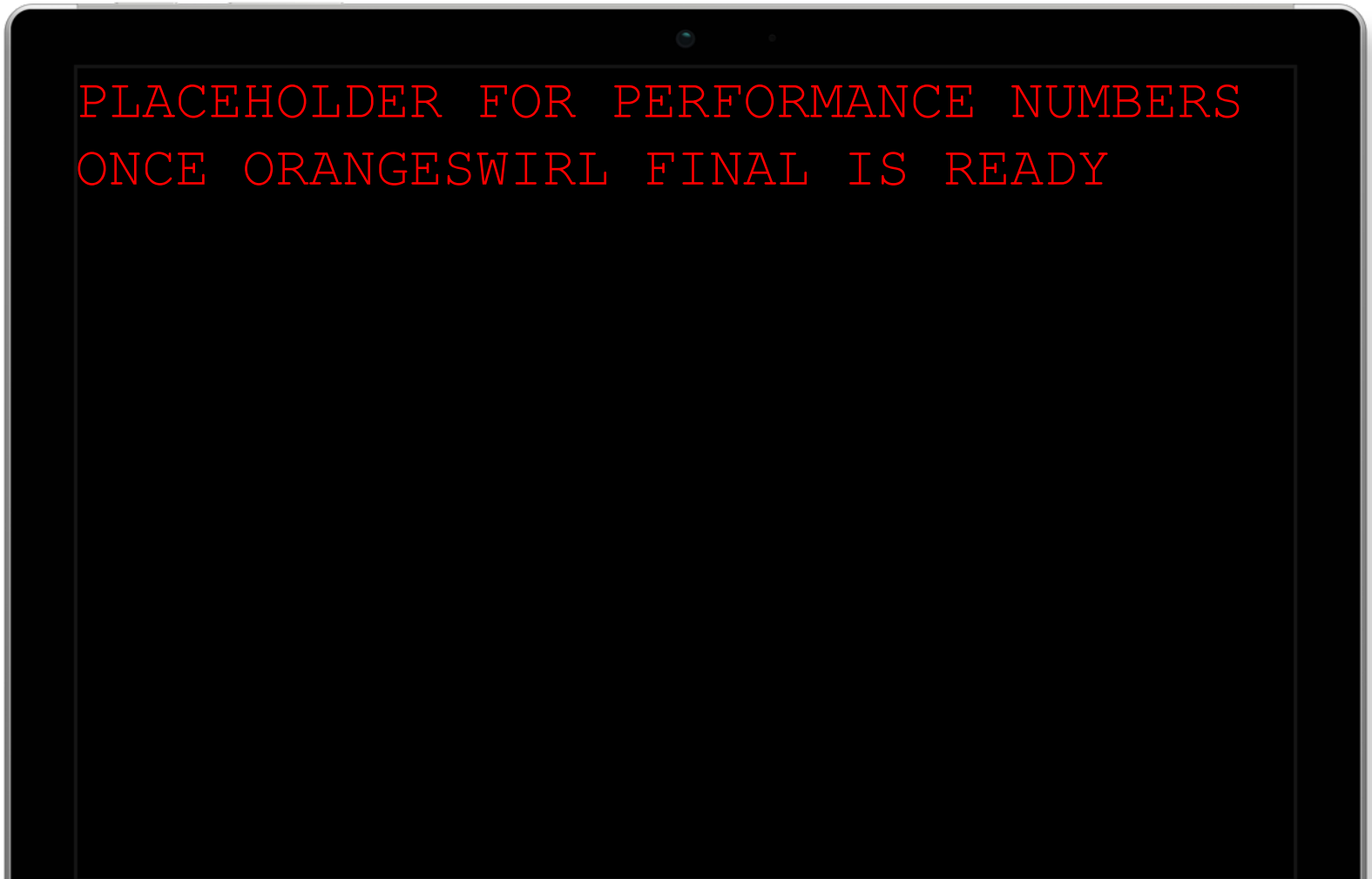
Journal Size



- Using our data set of 10 million events from earlier
- Spanning 3 buckets the total journal size dropped by 3%

Zstandard Compression

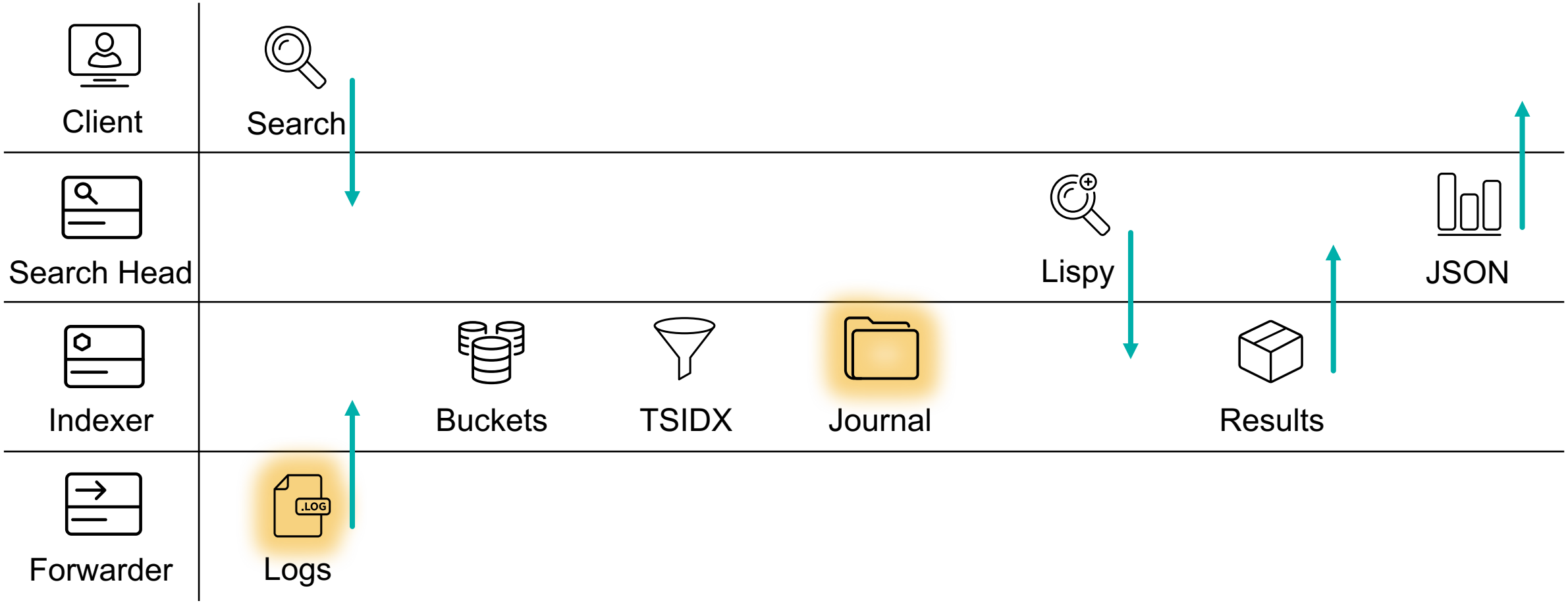
By the Numbers



- Search times reduced by XX%
- Search times reduced by XX%

Improvement #3

Eval



Enhanced Eval

New Engine

- New eval engine for Splunk 7.2
- New features for ingress evaluations
- Performance benefits for both search and ingress operations

Enhanced Eval

New Ingress Capabilities

```
$ vim transforms.conf

~ INGEST_EVAL =
    <comma separated list of eval expressions>
```

- Save extracted fields as types other than strings
  - float, int, multival
- Can delete or replace existing fields
- Change the type of an existing field
  - "200 OK" > 200

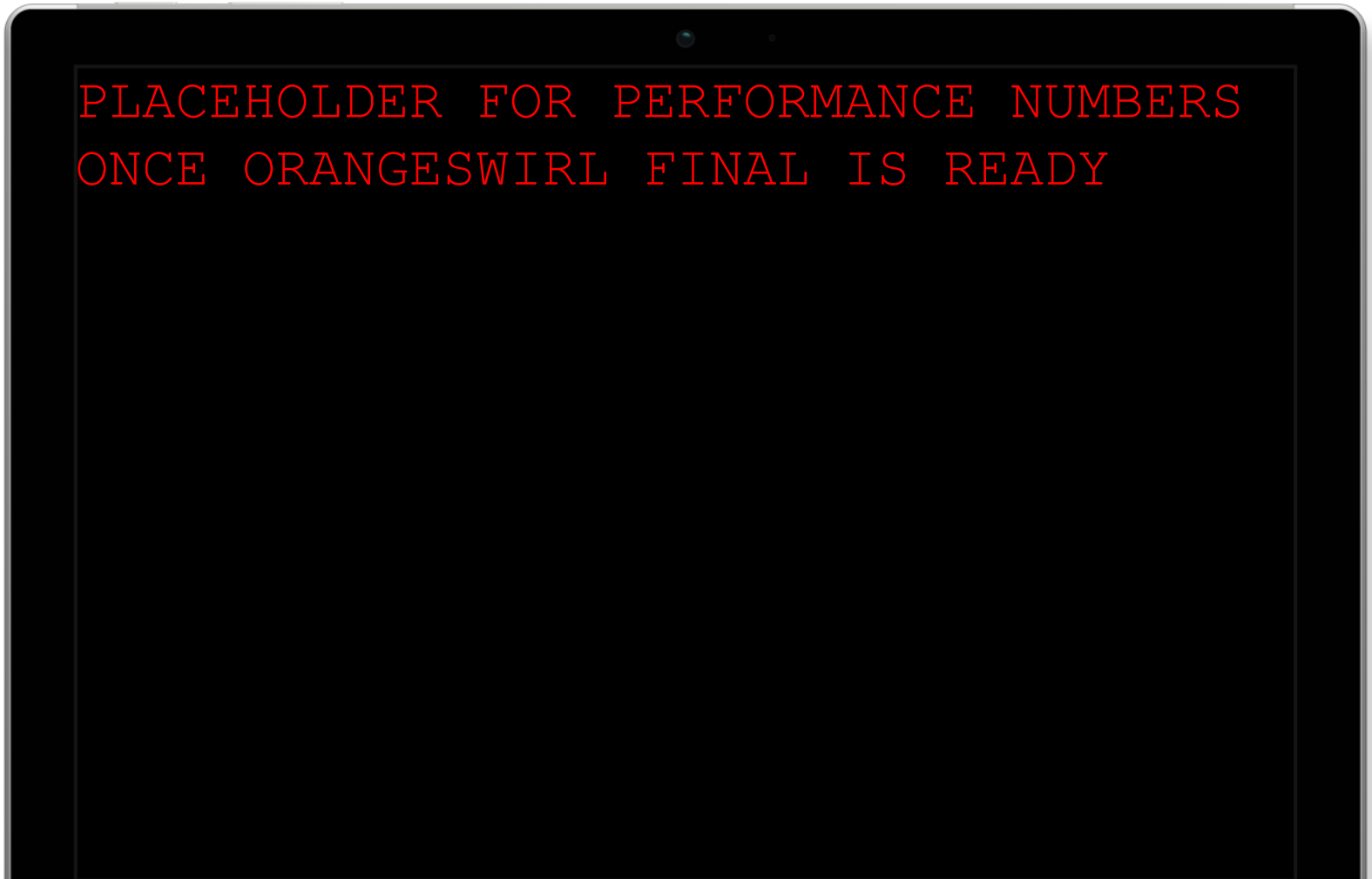
Enhanced Eval

Landing in Splunk 7.2

- Typically memory allocation is incredibly quick
  - However over millions events this can actually become a bottleneck
- Memory management improvements to eval engine
  - Reduced the need to copy results from one process to another
- Performance boosts will grow as the size of the data set increases
  - `| search len(_raw) > 1000`

Eval Enhancements

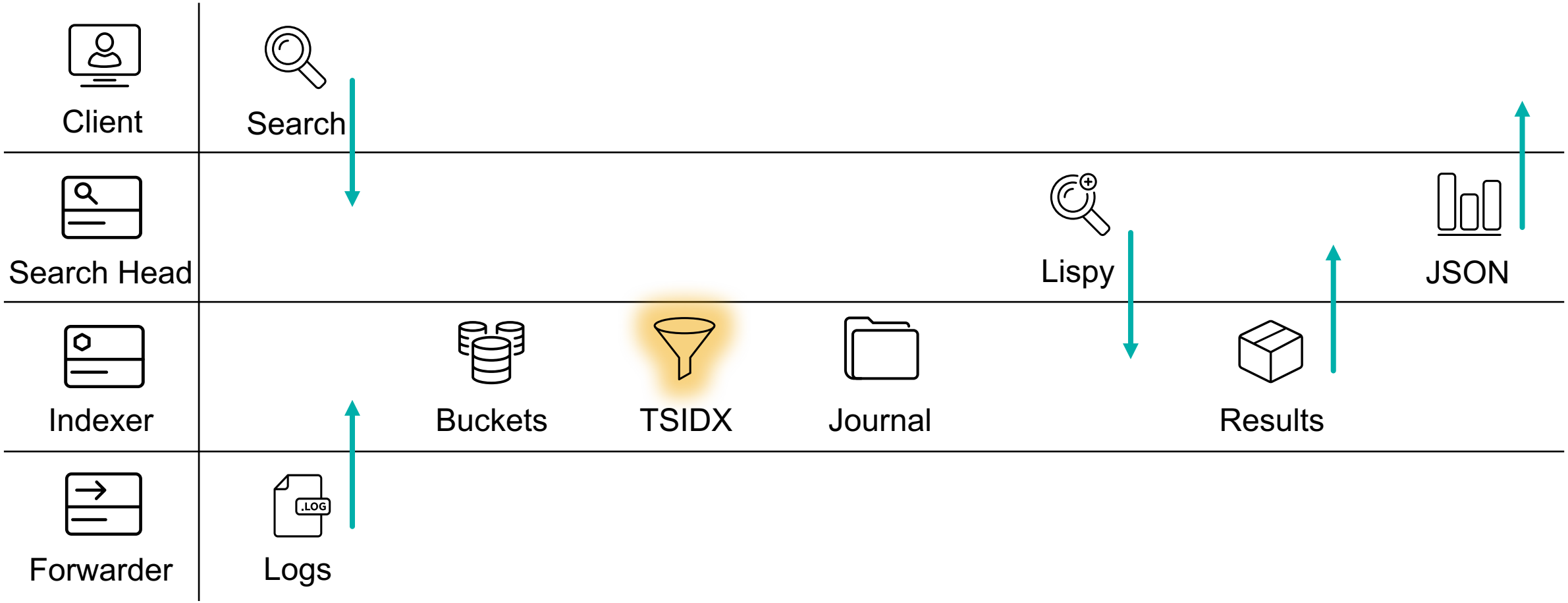
By the Numbers



- X% reduction in memory consumption
- X% drop in CPU utilization

Improvement #4

Metric Store



Metric Store

Historical Overview

- First shipped in Splunk 7.0 as a new storage type
- Leverages structured data to improve search performance
  - Nearly a 200x improvement over Splunk 6.6 in certain instances
- Comes at the cost of a more restrictive query interface via `|mstats` command

Extra Credit

Less Shameless Plug

- Great .conf talk on Metric Store from earlier in the week

- FN1096 - Metric Indexes: Architecture and Usage
  - Video and PowerPoints will be online shortly at conf.splunk.com

splunk> .conf18

Why Metrics

Splunk Seemed Pretty Good Already

- TSIDX comes in two forms
  - "Classic" edition for basic `|search` queries
  - MSIDX for `|mstats` commands
- TSIDX only supported strings
- MSIDX addressed this with support for 64-bit floats

MSIDX V2

Shipped in Splunk 7.1

- Splunk 7.0 shipped with MSIDX V1
  - Implemented with minor modifications to existing TSIDX logic
- MSIDX V2 introduced in Splunk 7.1
  - Major reworking of the data structure

MSIDX V2

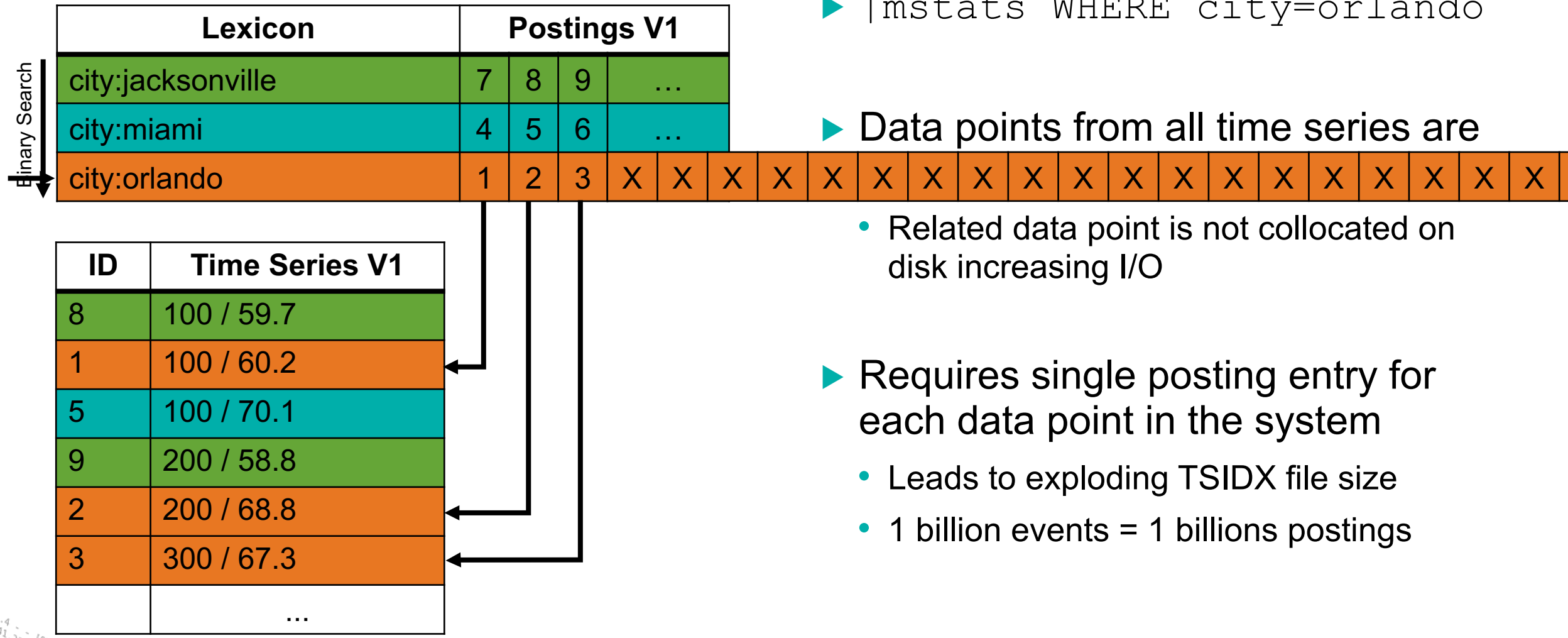
Metrics Example

metric_name	_value	_time	city
temperature	60.2	100	orlando
temperature	68.8	200	orlando
temperature	67.3	300	orlando
temperature	70.1	100	miami
temperature	69.3	200	miami
temperature	97.0	300	miami
temperature	59.7	100	jacksonville
temperature	58.8	200	jacksonville
temperature	62.4	300	Jacksonville

- Required Dimension
  - `metric_name=<str>`
- Measurement
  - `_value=<float>`
- Timestamp
  - `_time=<int>`
- Optional Dimensions
  - `<key>=<str>`

MSIDX V2

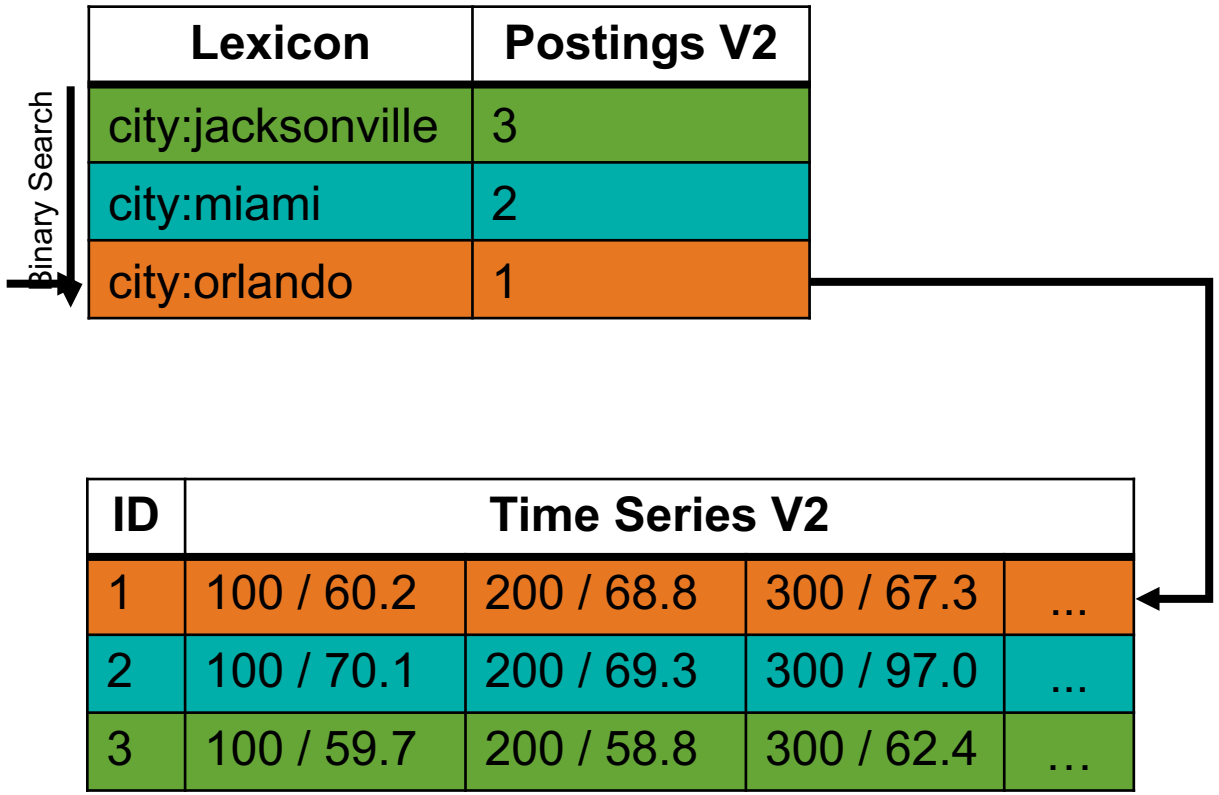
Previous Challenges



- `|mstats WHERE city=orlando`
- Data points from all time series are
  - Related data point is not collocated on disk increasing I/O
- Requires single posting entry for each data point in the system
  - Leads to exploding TSIDX file size
  - 1 billion events = 1 billions postings

MSIDX V2

Improvements



- For V2 a single posting now points to entire time series store
- Cuts the number of postings by a few order of magnitudes
- Reduces IO by moving time series data points closer together on disk
- Allows for better compression techniques

MSIDX V2

Let's Enable

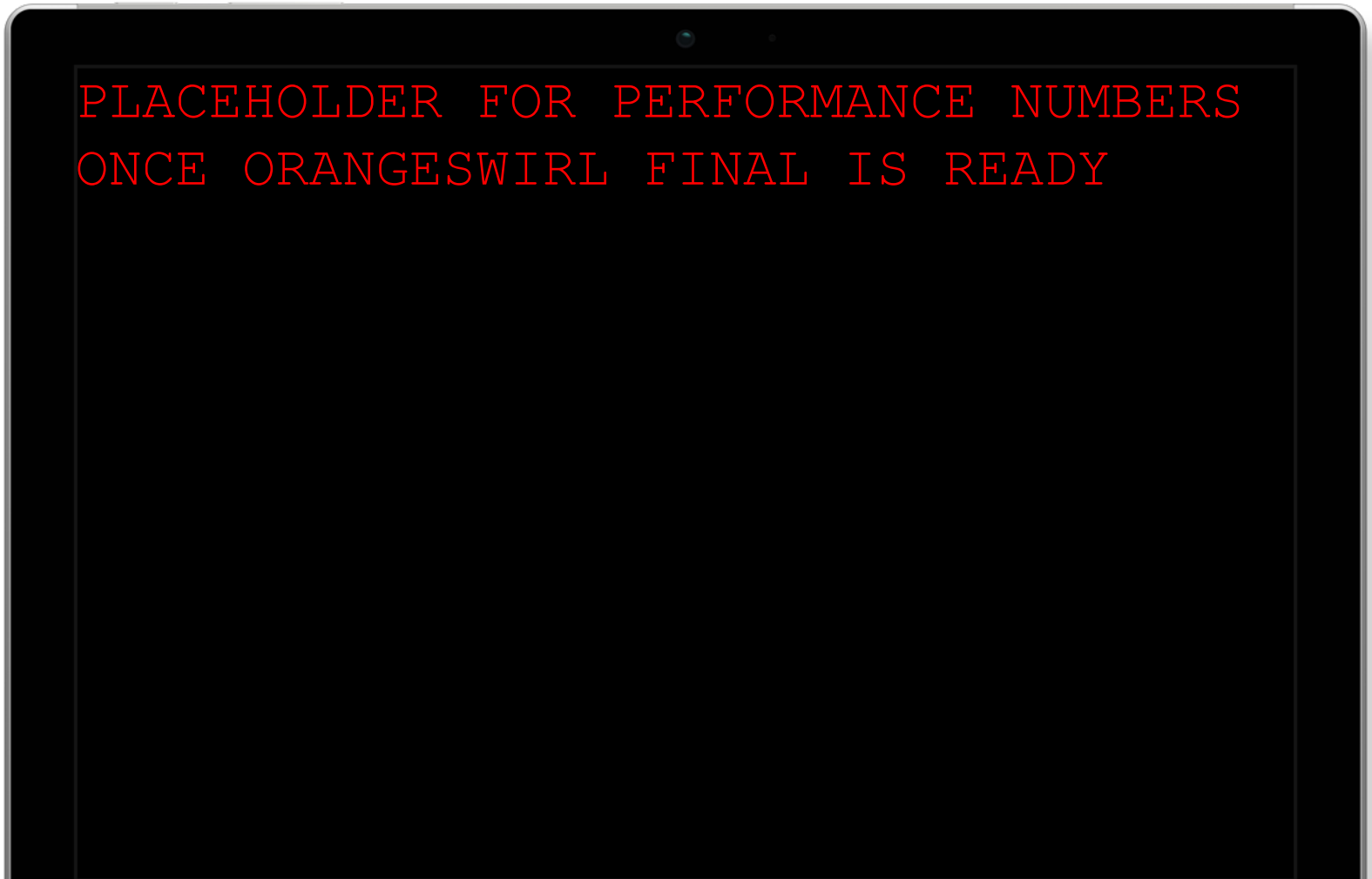
```
$ vim etc/system/local/indexes.conf

~ [my_metric_index]
~ datatype = metric
```

- Enabled by default for Splunk 7.1 onwards
- Newly ingested metrics will automatically be stored as V2
  - Existing metrics will be need to be re-ingested to move to V2
- Splunk supports searching both V1 and V2 data in the same query

MSIDX V2

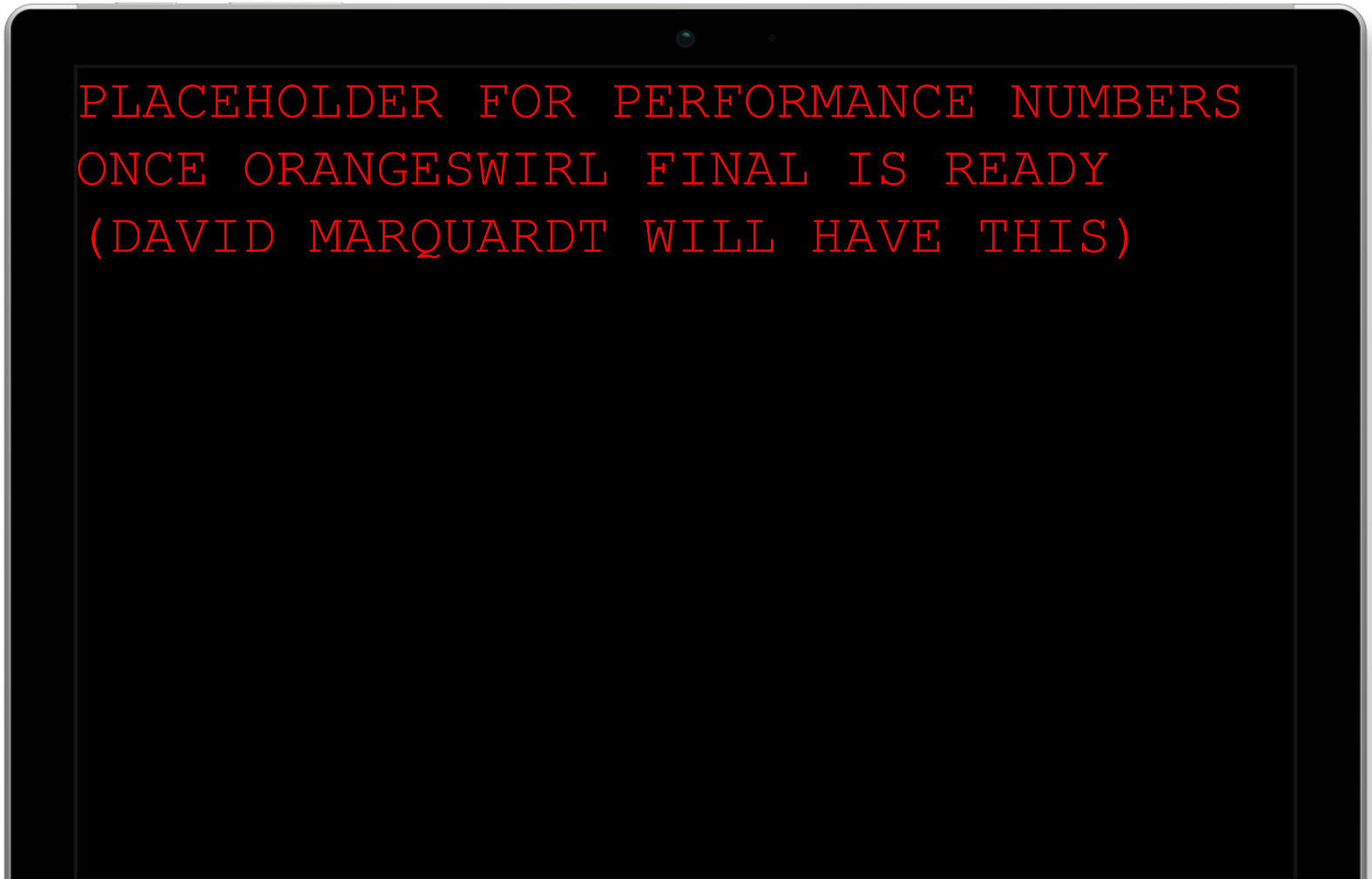
By the Numbers



- Aims to improve Metric performance by 10x over MSIDX V1
- X% improvement over Splunk 6.6
- X% improvement over Splunk 7.0

We've Been Busy

Since .conf2016



- Going back to our internal test suite
- +XX% search improvement cross the board in the last 2 years

I'm Just the Messenger

The Real Heroes

Ian Link

Vishal Patel

Justin Lin

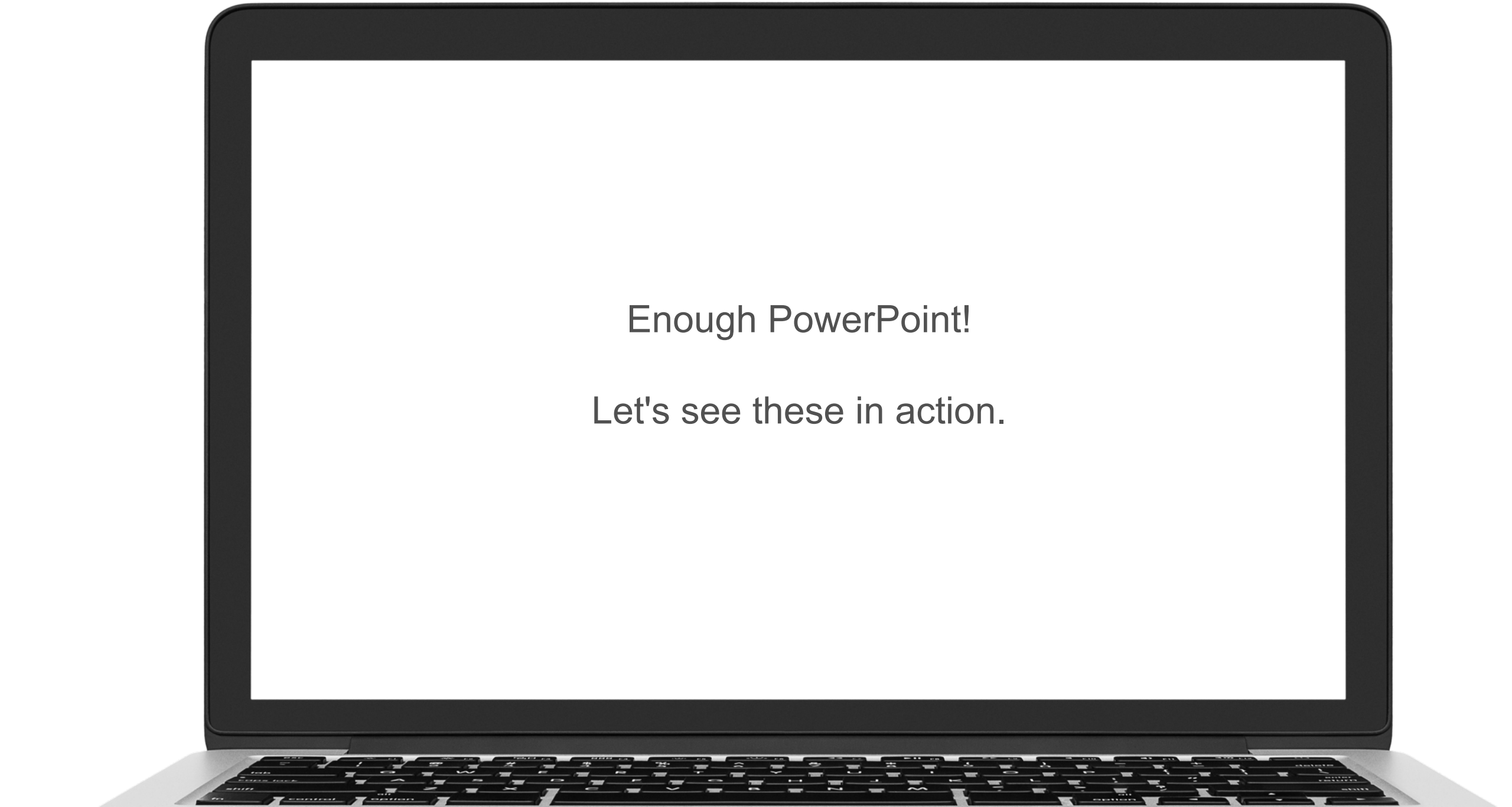
Manu Jose

Nick Romito

Karthik Sabhanatarajan

Mitch Blank

David Marquardt



Thank You

Don't forget to rate this session in the .conf18 mobile app

Q&A

Kellen Green | Senior Software Engineer