



Solution Brief

Data-Provenance-as-a-Service with the SPARKL® Sequencing Engine and Splunk

SPARKL® Limited 2016

Inside This Brief

2

About

What is the SPARKL Sequencing Engine?

3

Data Provenance

Applications of SPARKL in Business

4-5

Tracing Data

Using Splunk and SPARKL to source and analyse data

About SPARKL

Bring Machines Together

All enterprises suffer from the black box swamp. Systems that work fine on their own, but won't play nicely with others.

It's hard to describe how a system should work - let alone how or why different systems interact.

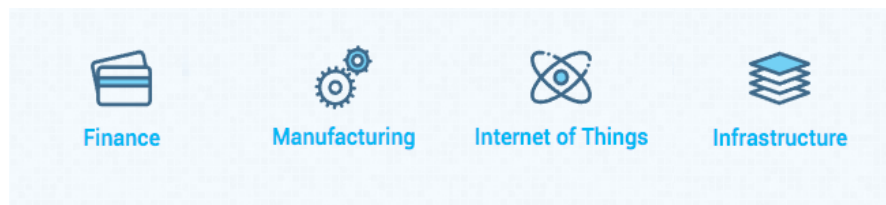
SPARKL® is powerful technology for managing the behaviour of distributed systems. The lightning fast, distributed SPARKL Sequencing Engine drives events between machines, applications and things.

It provides Distributed Intelligence for true fog computing, allowing edge devices to interact with or without the cloud.

It introduces Reasoned Provisioning which spins up secure, on-demand infrastructure to meet the need of actual business logic.

Secured by blockchain, SPARKL logs every single event in a clean, connected Audit Trail to solve compliance and regulatory reporting across machines and systems, old and new.

SPARKL designs and develops the SPARKL® Sequencing Engine in London, UK. We work with partners including Cisco and Intel to market the product to innovators and customers worldwide.



Data Provenance

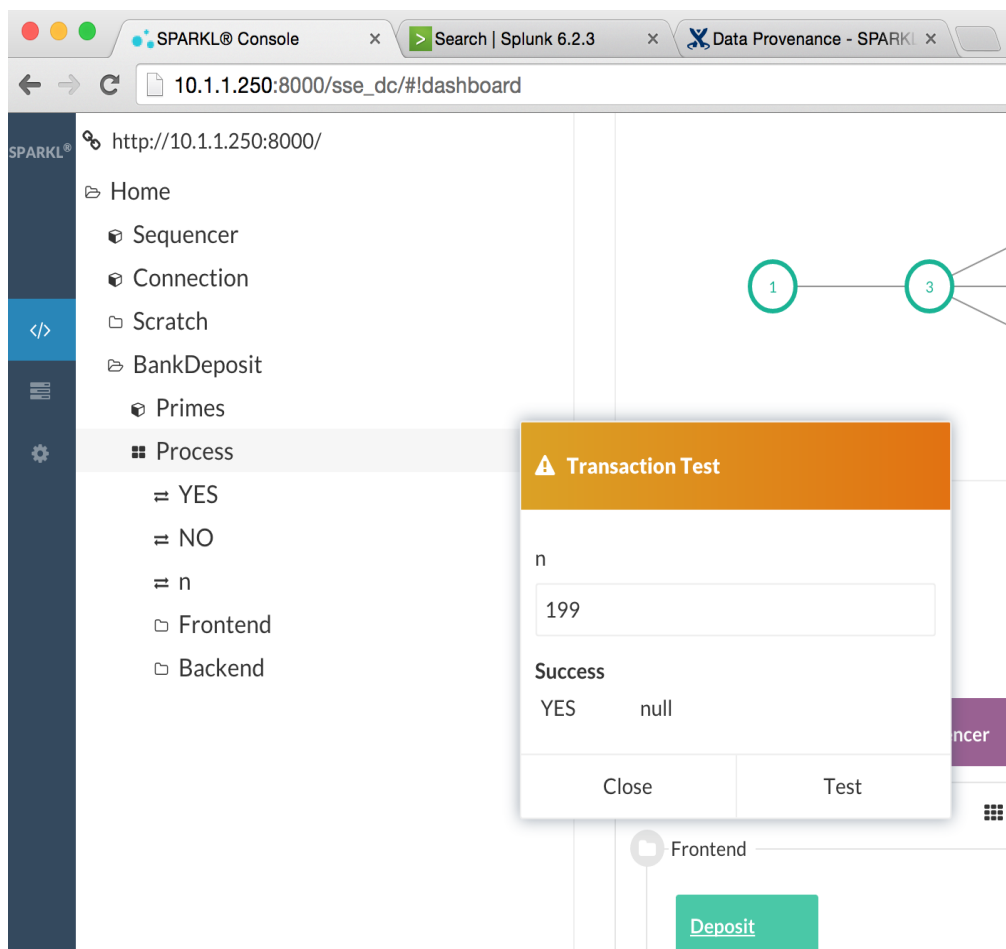
SPARKL is orchestration middleware which generates logs that can be mined to extract the provenance of data - that is, how data changes in value and moves around in an enterprise system.

Many different stakeholders are typically interested in how and where data is used, including business managers and executives, compliance auditors, technical staff and general users.

The execution of business transactions is the principal vehicle by which data is transferred between system components, and between systems and stakeholders.

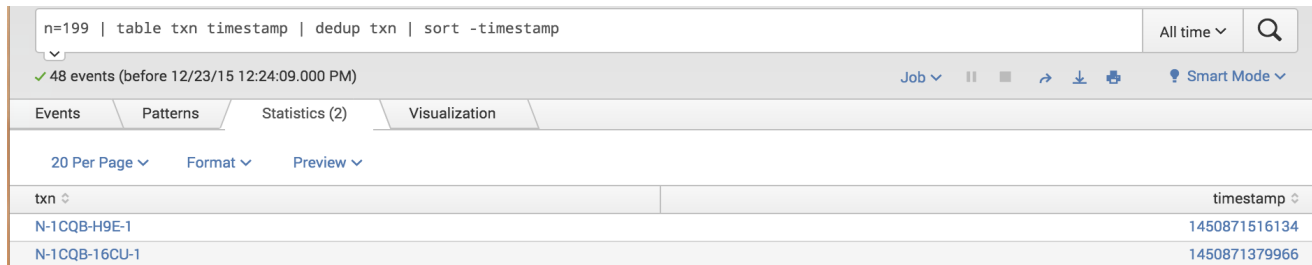
Each event logged by SPARKL is stamped with a transaction ID, as well as the service end points involved in the movement of data, thus allowing a stakeholder insight into which transactions and their types are involved in the migration of data through an enterprise and elsewhere.

In the following example, we are using SPARKL's administrator console to submit a transaction to a mock banking service for \$199 ($n=199$).



Data Provenance

Once we have done so, we can trace the provenance of this data. Firstly, let's get the transaction ID. The simplest way here would be to sort transactions including $n=199$ by timestamp using Splunk, and picking the last of these.



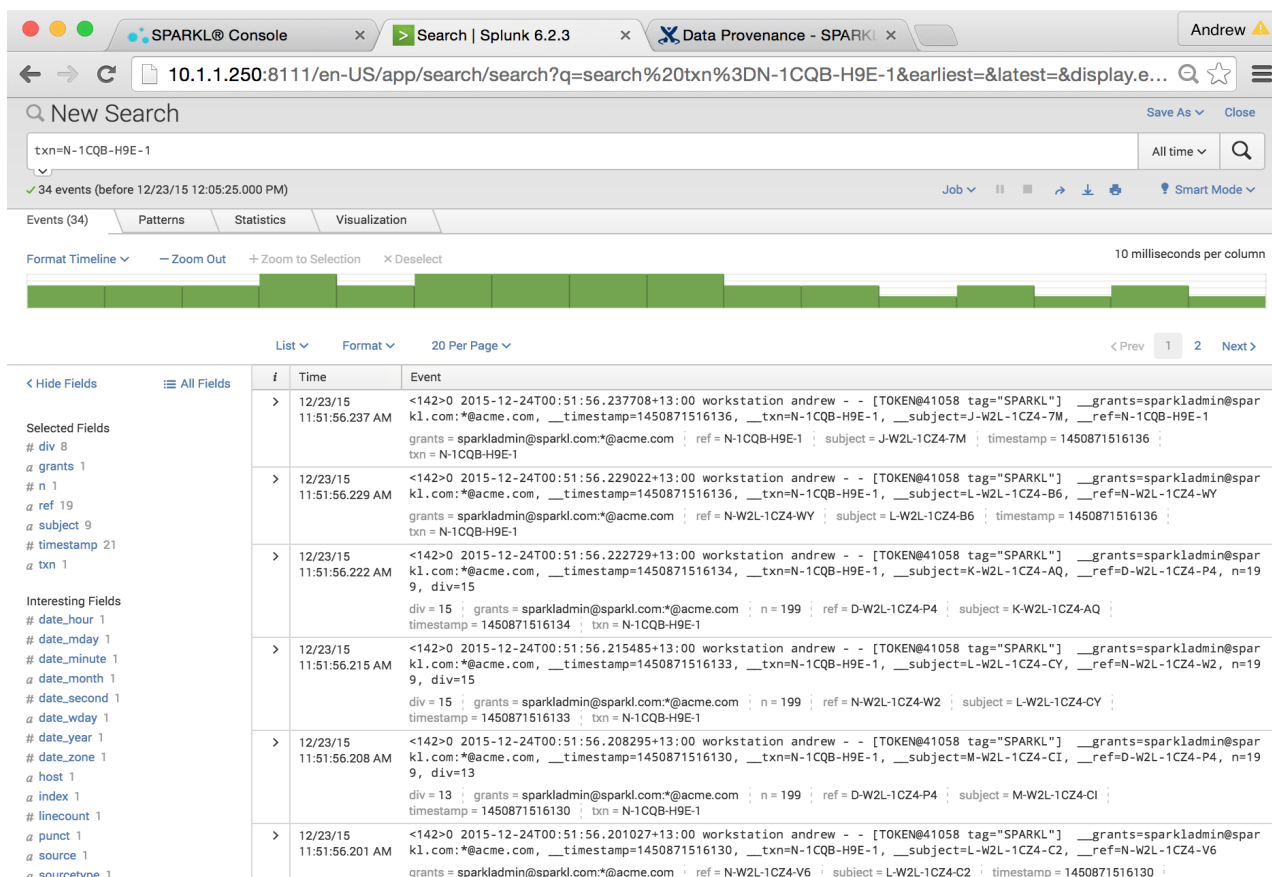
The screenshot shows a Splunk search interface with the query `n=199 | table txn timestamp | dedup txn | sort -timestamp`. It displays 48 events. The results table has two columns: `txn` and `timestamp`. The first two rows are:

txn	timestamp
N-1CQB-H9E-1	1450871516134
N-1CQB-16CU-1	1450871379966

Here, we see that the transaction ID we want is **N-1CQB-H9E-1**. By explicitly specifying this ID, we can see events relating to this transaction solely and how data pertaining to the transaction has moved around our system.

In the following screenshot, we can see events with different subjects which refer to various service end-points.

Event streams, where used by SPARKL analytics and administrators, will have **grant information** (as `__grants`, in the following), which determines who specifically may see the data event. We see that anyone from the ACME company, in this instance, may access the events.



The screenshot shows a Splunk search interface with the query `txn=N-1CQB-H9E-1`. It displays 34 events. The results table has three columns: `i`, `Time`, and `Event`. The first three rows are:

i	Time	Event
>	12/23/15 11:51:56.237 AM	<142>0 2015-12-24T00:51:56.237708+13:00 workstation andrew - - [TOKEN@41058 tag="SPARKL"] __grants=sparkladmin@sparkl.com:*@acme.com, __timestamp=1450871516136, __txn=N-1CQB-H9E-1, __subject=J-W2L-1CZ4-7M, __ref=N-1CQB-H9E-1 grants = sparkladmin@sparkl.com:*@acme.com ; ref = N-1CQB-H9E-1 ; subject = J-W2L-1CZ4-7M ; timestamp = 1450871516136 ; txn = N-1CQB-H9E-1
>	12/23/15 11:51:56.229 AM	<142>0 2015-12-24T00:51:56.229022+13:00 workstation andrew - - [TOKEN@41058 tag="SPARKL"] __grants=sparkladmin@sparkl.com:*@acme.com, __timestamp=1450871516136, __txn=N-1CQB-H9E-1, __subject=L-W2L-1CZ4-B6, __ref=N-W2L-1CZ4-WY grants = sparkladmin@sparkl.com:*@acme.com ; ref = N-W2L-1CZ4-WY ; subject = L-W2L-1CZ4-B6 ; timestamp = 1450871516136 ; txn = N-1CQB-H9E-1
>	12/23/15 11:51:56.222 AM	<142>0 2015-12-24T00:51:56.222729+13:00 workstation andrew - - [TOKEN@41058 tag="SPARKL"] __grants=sparkladmin@sparkl.com:*@acme.com, __timestamp=1450871516134, __txn=N-1CQB-H9E-1, __subject=K-W2L-1CZ4-AQ, __ref=D-W2L-1CZ4-P4, n=199, div=15 grants = sparkladmin@sparkl.com:*@acme.com ; n = 199 ; ref = D-W2L-1CZ4-P4 ; subject = K-W2L-1CZ4-AQ ; timestamp = 1450871516134 ; txn = N-1CQB-H9E-1

By refining the search, we are able to see clearly that the **n** data field touched two different service end-points for the given transaction ID.

K-W2L-1CZ4-9U	199	1450871516071
K-W2L-1CZ4-AQ	199	1450871516134
K-W2L-1CZ4-AQ	199	1450871516127
K-W2L-1CZ4-AQ	199	1450871516120