DocOps.io AsciiDoctorJ Architecture Decision Record(ADR)

Steve D Roach

Table of Contents

1. What is it?

An Architectural Decision (AD) is a software design choice that addresses a functional or non-functional requirement that is architecturally significant. ADR

2. How to use?

```
[adr,test2,border=false] ① ② ③
----
include::123.adr[] ④
----
```

① adr—name of the extension

② test2—name of the generated file

③ border-use shadowed border or not

④ file that contains the adr format

Top

2.1. ADR Format

- Using colons as separators for the file layout.

```
Title:Arch decision Title ①
Date: November 24th, 2010 ②
Status: Proposed ③
Context: Setting the context here ④
Consequences: What are the consequences of the decision ⑤
Participants: Architect,Engineer ⑥
```

① Title for the record

② Date for the record

③ Status can be 1 of (Proposed, Accepted, Superseded, Deprecated, Rejected)

④ Context

⑤ Consequences

⑥ Participants (optional)

Top

2.2. First Example

- Contents of 123.adr

```
Title:Use Solr for Structured Data Search
Date: November 24th, 2010
Status: Accepted
Context:  Solr and Elasticsearch are both open source search engines. Both can be used
to search
large amounts of data quickly and accurately. While Solr uses a SQL-like query
language, Elasticsearch has a full-text search engine and is designed for distributed
search and analytics. Elasticsearch also allows for faster indexing and more advanced
search replicas. Both technologies have strengths and weaknesses and are often used in
combination for enterprise-level search. There is a need of having an API exposed
which can be used to search structured
data. The Data currently resides in RDBMS, it is difficult to expose micro-service
directly querying out of RDBMS databases since the application runs out of the same
environment.

There are options like [[https://www.elastic.co ElasticSearch]] and Solr where data
can be replicated.

These solutions provide out of the box capabilities that can be leveraged by
developers without needed to build RESTful or GraphQL type APIs.
Decision:Use [[https://solr.apache.org/ Solr]] for data indexing. This use is because
Solr has high performance throughput with large volume of data.

Unstructured data can also be supported.
If this decision does not meet the need then additional PoC will be created.
Consequences:Data Needs to be replicated across the solr cloud instances.
This Solr cloud needs maintenance.

Near realtime data replication is required Additional Cost of maintaining the Solr
Cloud environment.
Participants:Roach,Rose,Duffy
```

Top

2.3. Resulting ADR

Use Solr for Structured Data Search

Status: Accepted **Date:** November 24th, 2010

Context

Solr and Elasticsearch are both open source search engines. Both can be used to search large amounts of data quickly and accurately. While Solr uses a SQL-like query language, Elasticsearch has a full-text search engine and is designed for distributed search and analytics. Elasticsearch also allows for faster indexing and more advanced search replicas. Both technologies have strengths and weaknesses and are often used in combination for enterprise-level search. There is a need of having an API exposed which can be used to search structured data. The Data currently resides in RDBMS, it is difficult to expose micro-service directly querying out of RDBMS databases since the application runs out of the same environment.

There are options like and Solr where data can be replicated.

These solutions provide out of the box capabilities that can be leveraged by developers without needed to build RESTful or GraphQL type APIs.

Decision

Use for data indexing. This use is because Solr has high performance throughput with large volume of data.

Unstructured data can also be supported. If this decision does not meet the need then additional PoC will be created.

Consequences

Data Needs to be replicated across the solr cloud instances. This Solr cloud needs maintenance.

Near realtime data replication is required Additional Cost of maintaining the Solr Cloud environment.

Participants

Roach,Rose,Duffy

Top

2.3.1. New Window Test

Use Solr for Structured Data Search

Status: Deprecated **Date:** November 24th, 2010

Context

Solr and Elasticsearch are both open source search engines. Both can be used to search large amounts of data quickly and accurately. While Solr uses a SQL-like query language, Elasticsearch has a full-text search engine and is designed for distributed search and analytics. Elasticsearch also allows for faster indexing and more advanced search replicas. Both technologies have strengths and weaknesses and are often used in combination for enterprise-level search. There is a need of having an API exposed which can be used to search structured data. The Data currently resides in RDBMS, it is difficult to expose micro-service directly querying out of RDBMS databases since the application runs out of the same environment. There are options like and Solr where data can be replicated. These solutions provide out of the box capabilities that can be leveraged by developers without needed to build RESTful or GraphQL type APIs.

Decision

Use for data indexing. This use is because Solr has high performance throughput with large volume of data. Unstructured data can also be supported. If this decision does not meet the need then additional PoC will be created.

Consequences

Data Needs to be replicated across the solr cloud instances. This Solr cloud needs maintenance. Near realtime data replication is required Additional Cost of maintaining the Solr Cloud environment.

Participants

Roach,Rose,Duffy

Top

Source for Non New Window

```
[adr,test4,border=true, role="left", newWin=false, increaseWidth="80"]
----
include::456.adr[] ①
----
```

Use Solr for Structured Data Search

Status: Proposed **Date:** November 24th, 2010

Context

Solr and Elasticsearch are both open source search engines. Both can be used to search large amounts of data quickly and accurately. While Solr uses a SQL-like query language, Elasticsearch has a full-text search engine and is designed for distributed search and analytics. Elasticsearch also allows for faster indexing and more advanced search replicas. Both technologies have strengths and weaknesses and are often used in combination for enterprise-level search. There is a need of having an API exposed which can be used to search structured data. The Data currently resides in RDBMS, it is difficult to expose micro-service directly querying out of RDBMS databases since the application runs out of the same environment. There are options like and Solr where data can be replicated.

These solutions provide out of the box capabilities that can be leveraged by developers without needed to build RESTful or GraphQL type APIs.

Decision

Use for data indexing. This use is because Solr has high performance throughput with large volume of data. Unstructured data can also be supported. If this decision does not meet the need then additional PoC will be created.

Consequences

Data Needs to be replicated across the solr cloud instances. This Solr cloud needs maintenance. Near realtime data replication is required Additional Cost of maintaining the Solr Cloud environment.

Participants

Roach,Rose,Duffy

Top

Use Solr for Structured Data Search

Status: Superseded **Date:** November 24th, 2010

Context

Solr and Elasticsearch are both open source search engines. Both can be used to search large amounts of data quickly and accurately. While Solr uses a SQL-like query language, Elasticsearch has a full-text search engine and is designed for distributed search and analytics. Elasticsearch also allows for faster indexing and more advanced search replicas. Both technologies have strengths and weaknesses and are often used in combination for enterprise-level search. There is a need of having an API exposed which can be used to search structured data. The Data currently resides in RDBMS, it is difficult to expose micro-service directly querying out of RDBMS databases since the application runs out of the same environment. There are options like and Solr where data can be replicated. These solutions provide out of the box capabilities that can be leveraged by developers without needed to build RESTful or GraphQL type APIs.

Decision

Use for data indexing. This use is because Solr has high performance throughput with large volume of data. Unstructured data can also be supported. If this decision does not meet the need then additional PoC will be created.

Consequences

Data Needs to be replicated across the solr cloud instances. This Solr cloud needs maintenance. Near realtime data replication is required Additional Cost of maintaining the Solr Cloud environment.

Participants

Roach,Rose,Duffy

Top

Use Solr for Structured Data Search

Status: Rejected **Date:** November 24th, 2010

Context

Solr and Elasticsearch are both open source search engines. Both can be used to search large amounts of data quickly and accurately. While Solr uses a SQL-like query language, Elasticsearch has a full-text search engine and is designed for distributed search and analytics. Elasticsearch also allows for faster indexing and more advanced search replicas. Both technologies have strengths and weaknesses and are often used in combination for enterprise-level search. There is a need of having an API exposed which can be used to search structured data. The Data currently resides in RDBMS, it is difficult to expose micro-service directly querying out of RDBMS databases since the application runs out of the same environment. There are options like and Solr where data can be replicated. These solutions provide out of the box capabilities that can be leveraged by developers without needed to build RESTful or GraphQL type APIs.

Decision

Use for data indexing. This use is because Solr has high performance throughput with large volume of data. Unstructured data can also be supported. If this decision does not meet the need then additional PoC will be created.

Consequences

Data Needs to be replicated across the solr cloud instances. This Solr cloud needs maintenance. Near realtime data replication is required Additional Cost of maintaining the Solr Cloud environment.

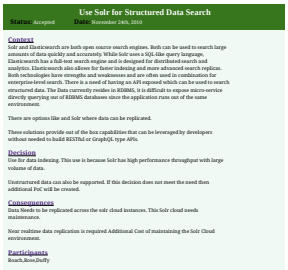
Participants

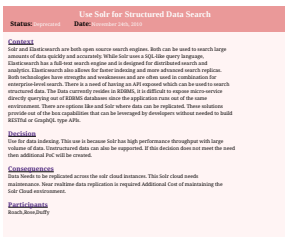
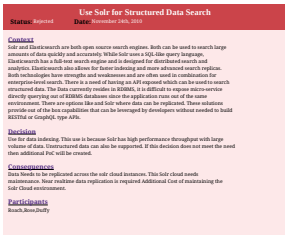
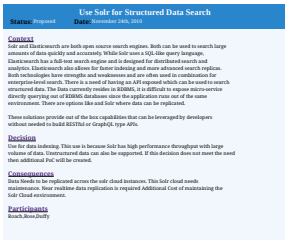
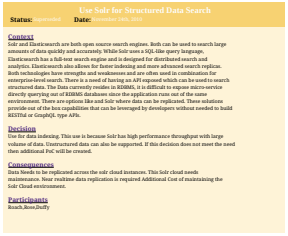
Roach,Rose,Duffy

Top

① Included adr file

3. ADR Summary Example

Title	ADR Contents	Participants	Date
Use Solr for Structured Data Search	▼ *View ADR Details* 	Roach,Rose,Duffy ,	November 24th, 2010

Title	ADR Contents	Participants	Date
Use Solr for Structured Data Search	▼ *View ADR Details* 	Roach,Rose,Duffy ,	November 24th, 2010
Use Solr for Structured Data Search	▼ *View ADR Details* 	Roach,Rose,Duffy ,	November 24th, 2010
Use Solr for Structured Data Search	▼ *View ADR Details* 	Roach,Rose,Duffy ,	November 24th, 2010
Use Solr for Structured Data Search	▼ *View ADR Details* 	Roach,Rose,Duffy ,	November 24th, 2010
http://127.0.0.1:8000/src/main/docs/456.adr does not exist			
http://127.0.0.1:8000/src/main/docs/123.adr does not exist			

3.1. ADR Summary Source

Only adr files can be processed. Recommend using adr files and include as you can version control this file.

```
[adrsummary]
----
123.adr ①
http://127.0.0.1:8000/src/main/docs/456.adr ②
http://127.0.0.1:8000/src/main/docs/123.adr
----
```

① Relative path to current document

② Full URL to adr

Header 1	Header 2
Column 1, row 1	Column 2, row 1
▼ *ADR*	Column 2, row 2