SpringOne

Application Modernization: Migrating mainframe apps to the cloud using Spring

September 1–2, 2021

springone.io

Application Modernization: Migrating mainframe apps to the cloud using Spring

Forward-Looking Statements

This presentation contains forward-looking statements including, among other things, statements regarding VMware's strategic and growth opportunities; planned spin-off from its parent company, Dell Technologies Inc. (Dell); expansion of VMware's offerings; and potential benefits to VMware and its customers and partners of the same. These forward-looking statements are subject to applicable safe harbor provisions under federal securities laws, such as the Private Securities Litigation Reform Act of 1995. Actual results could differ materially from those projected in the forward-looking statements as a result of certain risk factors, including but not limited to: (1) the delay or failure to consummate the spin-off from Dell; (2) the impact of the COVID-19 pandemic on our operations, financial condition, our customers, the business environment and global and regional economies; (3) adverse changes in general economic or market conditions; (4) delays or reductions in consumer, government and information technology spending; (5) competitive factors, such as pricing pressures, industry consolidation, entry of new competitors into the industries in which we compete, as well as new product and marketing initiatives by VMware's competitors; (6) rapid technological changes in the virtualization software, cloud, end user, security and mobile computing industries; (7) VMware's ability to enter into, maintain and extend strategically effective partnerships, collaborations and alliances; (8) VMware's relationship with Dell, Dell's ability to control matters requiring VMware stockholder approval and matters relating to Dell's investment in VMware; and (9) changes in VMware's financial condition. These forward-looking statements are made as of the initial date of this presentation, are based on current expectations and are subject to uncertainties and changes in condition, significance, value and effect as well as other risks detailed in documents filed with the Securities and Exchange Commission, including VMware's most recent reports on Form 10-K and Form 10-Q and current reports on Form 8- K that we may file from time to time, which could cause actual results to vary from expectations. VMware assumes no obligation to, and does not currently intend to, update any such forward-looking statements after the initial date of this presentation.

GLENN RENFRO

Works on – Spring Cloud Data Flow, Spring Cloud Task, & Spring Batch

Board of Directors – AJUG & DevNexus

Twitter - @cppwfs

Agenda

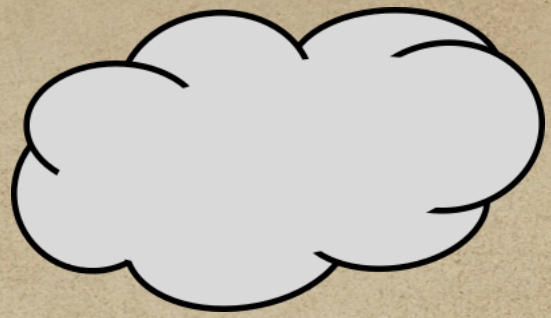
- Mainframe will surely be gone by now
- Strategies (the path)
- Streaming info out of the mainframe
- Batch apps
- Accessing the data
- Keeping the footprint small

My Story

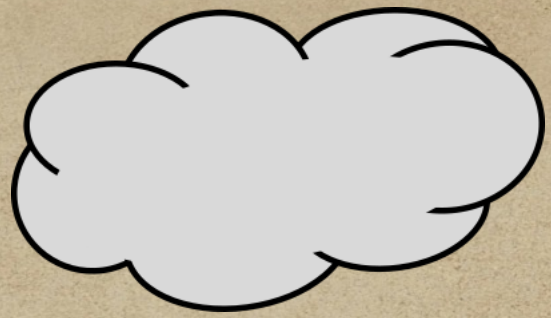
Why Migrate?

Big Bang

Strangler Pattern



Strangler Pattern




Our Company Island
Java Sea
Existing Data Mountains
Mainframe-ville
Spring Cloud Stream
Data Lake
Flat File Forest
Batchington
Query Bay
Graphql-town

Our Quest

- Stream data from mainframe to new datastore
- Migrate batch jobs
- Give services access to data to services


Our Company Island
Existing Data Mountains
Java Sea
Mainframe-ville
Spring Cloud Stream
Flat File Forest
Data Lake
Batchington
Query Bay
Graphql-town

First Steps

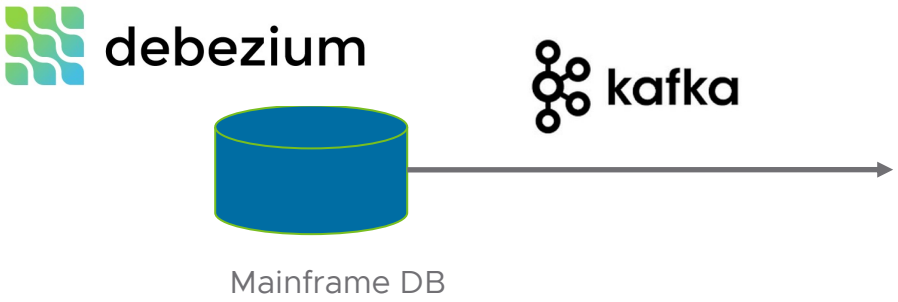
How do we get our data from mainframe's DB?

How do we keep our data in sync from the mainframe?

Chapter One: Following The Stream

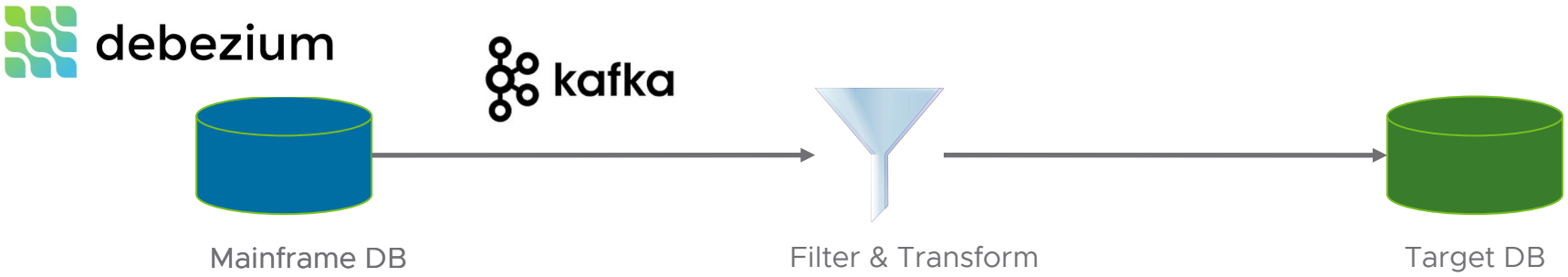

Our Company Island
Existing Data Mountains
Java Sea
Spring Cloud Stream
Flat File Forest
Data Lake
Query Bay

What are the components to our stream?

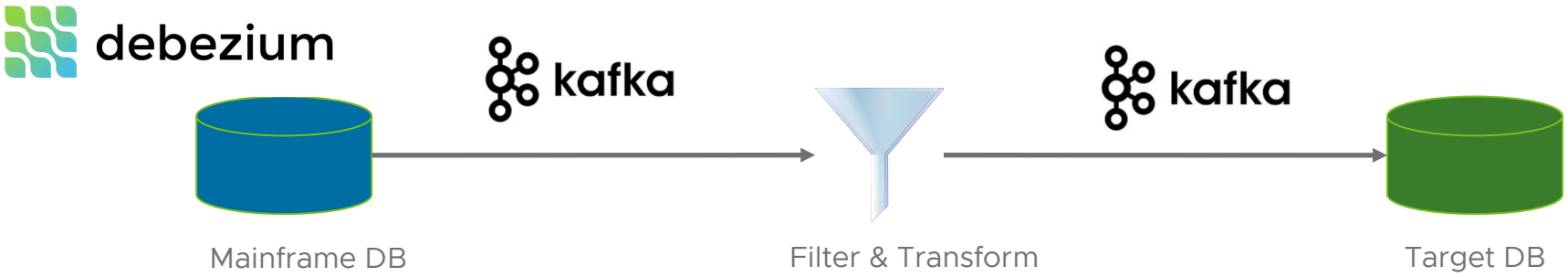


What are the components to our stream?

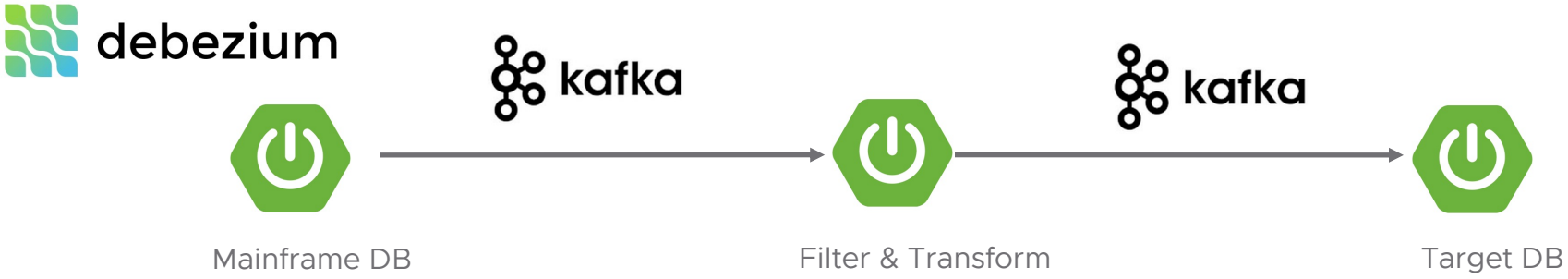
What are the components to our stream?



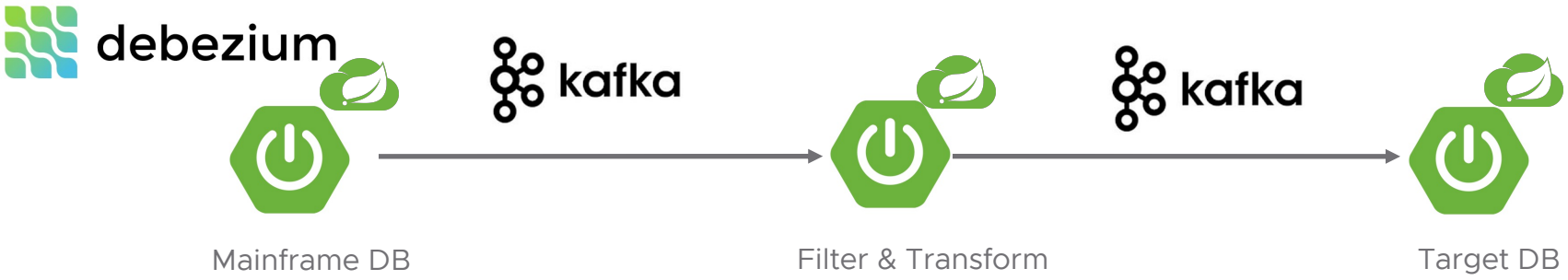
What are the components to our stream?



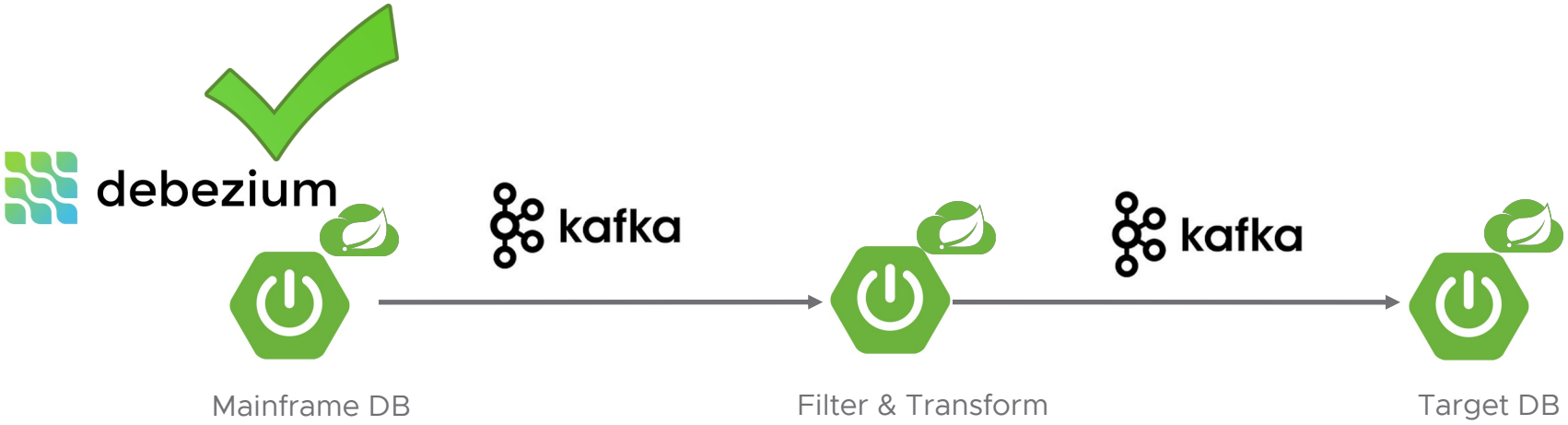
What are the components to our stream?



Spring Cloud Stream to the rescue



Spring Cloud Stream to the rescue



Spring Cloud Stream to the rescue

```
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-stream</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-stream-binder-kafka</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud.fn</groupId>
        <artifactId>filter-function</artifactId>
        <version>1.0.3</version>
    </dependency>
```

Spring Cloud Stream to the rescue

```xml
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-stream</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-stream-binder-kafka</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud.fn</groupId>
    <artifactId>filter-function</artifactId>
    <version>1.0.3</version>
</dependency>
```

```java
@Bean
public Function<String, String> transform() {
    return (payload) -> {
        JsonNode jsonNode = getJsonNode(payload);
        String result;
        String tableName = jsonNode.get("source").get("table").asText();
        if (tableName == null || jsonNode.get("after") == null) {
            result = translateOthers(jsonNode);
        }
        else {
            if (tableName.equals("plans")) {
                result = translatePlan(jsonNode);
            }
            else if (tableName.equals("users")) {
                result = translateUser(jsonNode);
            }
            else if (tableName.equals("call_usage")) {
                result = translateCallUsage(jsonNode);
            }
            else if (tableName.equals("data_usage")) {
                result = translateDataUsage(jsonNode);
            }
            else {
                result = translateOthers(jsonNode);
            }
        }
        return result;
    };
}
```

Spring Cloud Stream to the rescue

```xml
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-stream</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-stream-binder-kafka</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud.fn</groupId>
    <artifactId>filter-function</artifactId>
    <version>1.0.3</version>
</dependency>
```

```yaml
spring:
  cloud:
    stream:
      function:
        definition: transform|filterFunction
        bindings:
          transformfilterFunction-out-0: output
          transformfilterFunction-in-0: input

filter:
  function:
    expression: headers[usage]=='true'
```

```java
@Bean
public Function<String, String> transform() {
    return (payload) -> {
        JsonNode jsonNode = getJsonNode(payload);
        String result;
        String tableName = jsonNode.get("source").get("table").asText();
        if (tableName == null || jsonNode.get("after") == null) {
            result = translateOthers(jsonNode);
        }
        else {
            if (tableName.equals("plans")) {
                result = translatePlan(jsonNode);
            }
            else if (tableName.equals("users")) {
                result = translateUser(jsonNode);
            }
            else if (tableName.equals("call_usage")) {
                result = translateCallUsage(jsonNode);
            }
            else if (tableName.equals("data_usage")) {
                result = translateDataUsage(jsonNode);
            }
            else {
                result = translateOthers(jsonNode);
            }
        }
        return result;
    };
}
```

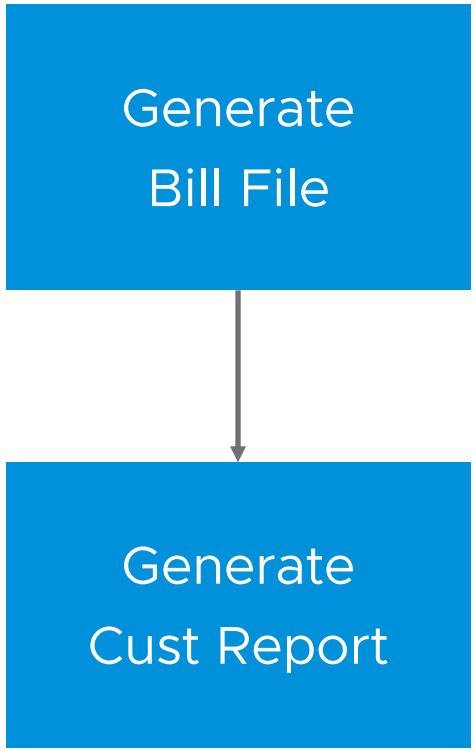

Demo


Java Sea
Existing Data Mountains
Mainframe-ville
TapTown
Tap
Cloud Stream
Spring
Flat File Forest
Data Lake
Batchington
Query Bay
Graphql-town

Chapter Two: Off to Batchington

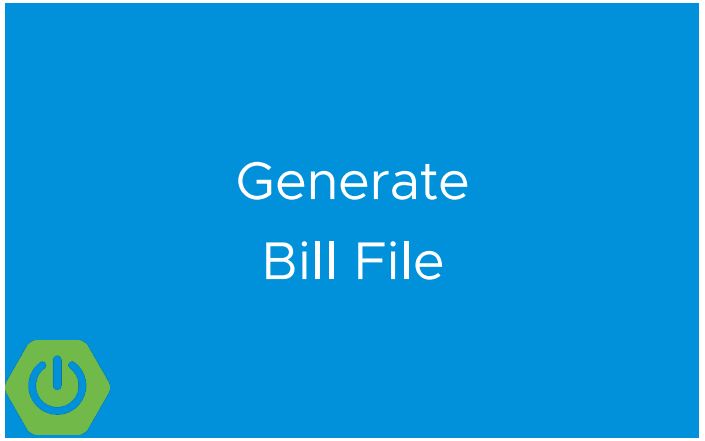

Our Company Island
Existing Data Mountains
Java Sea
Mainframe-ville
Spring Cloud Stream
Flat File Forest
Data Lake
Batchington
Query Bay
Graphql-town

Batch Jobs

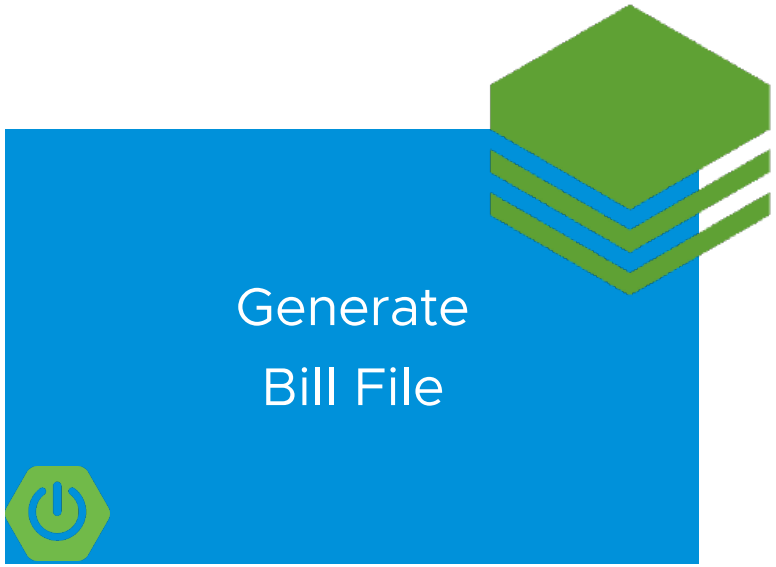



```
//MYJCL JOB ,'GLENN',CLASS=6,NOTIFY=&SYSUID
//*
//STEP1 EXEC PGM=GENBILL
//INPUT1 …
//OUTPUT1 DD DSN=BILL.OUTPUT,…
//*
//STEP2 EXEC PGM=BILLREP
//INPUT2 DD DSN=BILL.OUTPUT,…
//OUTPUT2 DD DSN=BILL.REPORT,…
```

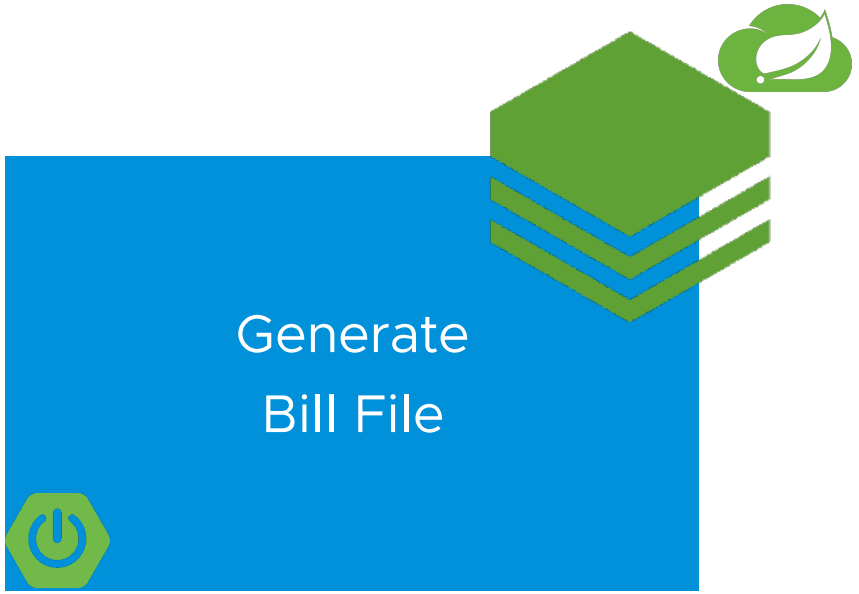
Batch Jobs



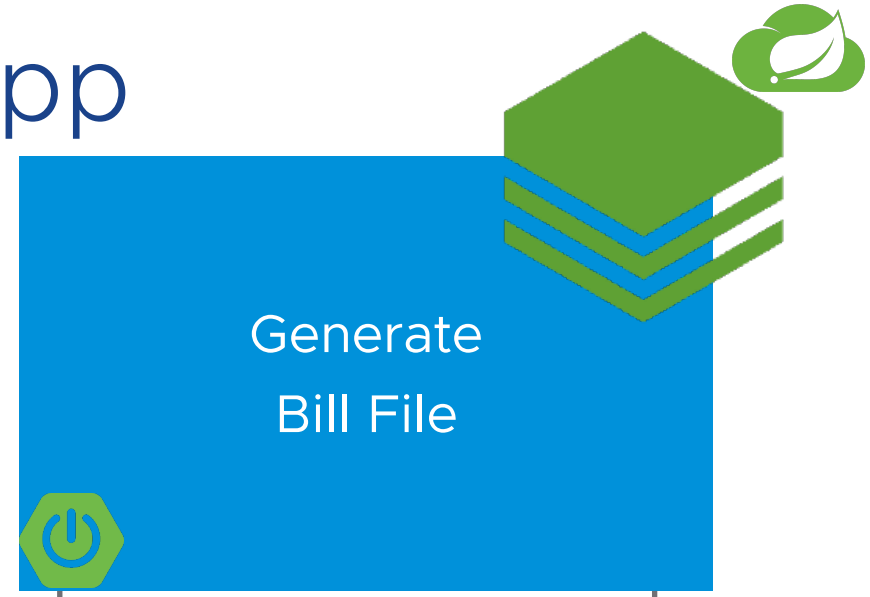
Batch Jobs

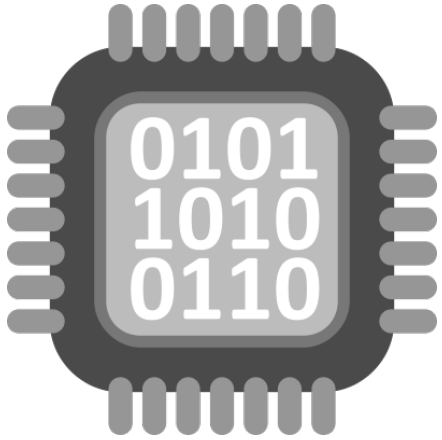


Batch Jobs

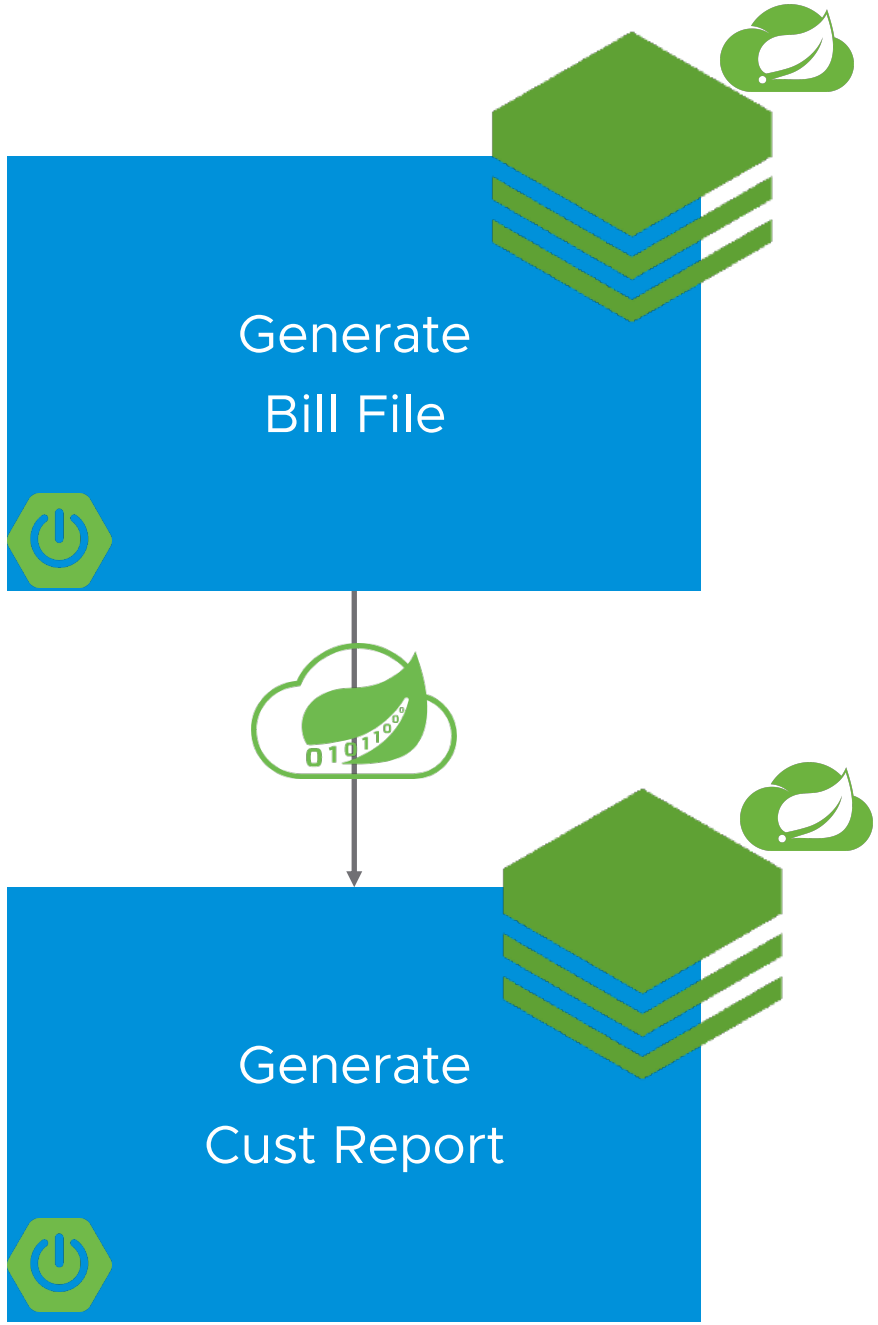


Focus on BillRun App





Batch Jobs




Demo

Chapter Three: Query Bay


Our Company Island
Existing Data Mountains
Java Sea
Mainframe-ville
Spring Cloud Stream
Data Lake
Flat File Forest
Batchington
Query Bay
Graphql-town
userId
planId
firstName
lastName

GraphQL

GraphQL is a relatively new but well-established alternative to REST for exposing web APIs that is rapidly becoming a popular choice for web and mobile clients due to its ease of use and degree of control.

GraphQL Syntax - Query

Retrieve the first 10 user ids and their associated plan ids.

```
query{users(count: 10){ id,planId }}
```


```
{
  "data": {
    "users": [
      {
        "id": 1,
        "planId": 1
      },
      {
        "id": 2,
        "planId": 1
      },
      {
        "id": 3,
        "planId": 1
      }
    ]
  }
}
```

GraphQL Syntax - Query

Retrieve the first 10 user ids and their associated plan ids.

```
query{users(count: 10){ firstName,lastName }}
```


```
{
  "data": {
    "users": [
      {
        "firstName": "doe",
        "lastName": "jane"
      },
      {
        "firstName": "doe",
        "lastName": "john"
      },
      {
        "firstName": "doe",
        "lastName": "pat"
      }
    ]
  }
}
```

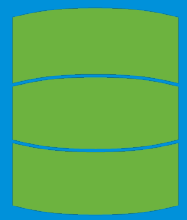
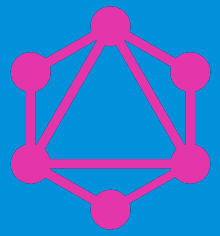
GraphQL Syntax - Mutate

Create a new User and have the result contain the new userId

```
mutation {createUser( lastName: \"doe\", firstName: \"ruby\", planId: 1) { id } }"}
```

```
{
  "data": {
    "createUser": {
      "id": 1162
    }
  }
}
```

Spring GraphQL

	
Entities	Queries
Repositories	Mutations
Services	GraphQLS Configuration


Demo

Application Size

Stay Connected.

Spring For Architects

https://springone.io/2021/sessions/spring-for-architects

Proactively Designing for Diversity, Equity, and Inclusion

https://springone.io/2021/sessions/proactively-designing-for-diversity-equity-and-inclusion

@s1p #springone

Thank You!

https://github.com/cppwfs/S12021Demo