Transformation as a Service

Paul Ireifej/ Principal Member of Technical Staff
Mohammad Omar Khalid Mirza/ Professional Big Data Engineer
April 19, 2021

AT&T Business

Agenda

Let your customers be your partners; let your vendors be your employees. What's necessary in this transformation more than anything else is courage and willingness to change.

Safra Catz (CEO of Oracle)

1. Overview

Managing data flow across systems is challenging

Enabling data flow is a manual process

Need to define a schema

Managing data flow across systems is challenging

Enabling data flow is a manual process

Need to define a schema

Develop code specific to a target system to implement the schema

Managing data flow across systems is challenging

Enabling data flow is a manual process

Need to define a schema

Develop code specific to a target system to implement the schema

Time-intensive, costly, error-prone

Managing data flow across systems is challenging

Enabling data flow is a manual process

Need to define a schema

Develop code specific to a target system to implement the schema

Time-intensive, costly, error-prone

Field format changes

Managing data flow across systems is challenging

Enabling data flow is a manual process

Need to define a schema

Develop code specific to a target system to implement the schema

Time-intensive, costly, error-prone

Field format changes

Schema change and redefinition

Managing data flow across systems is challenging
Enabling data flow is a manual process

Need to define a schema

Develop code specific to a target system to implement the schema

Time-intensive, costly, error-prone

Field format changes

Schema change and redefinition

Target system code change and testing

Transformation as a Service

Overview

Transforms data input

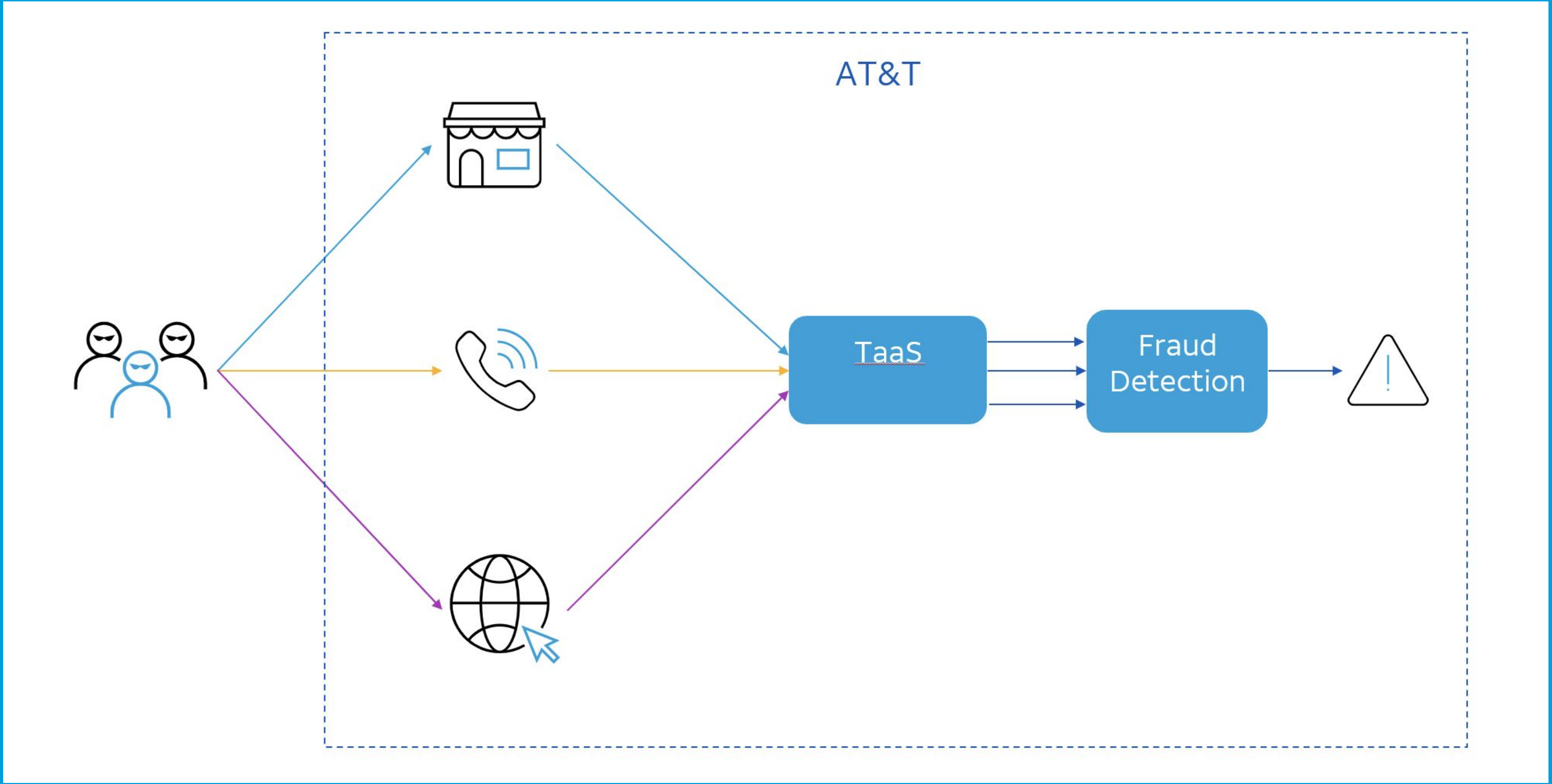
User interface for

self-service

Different data sources and formats

Transformation as a Service / April 19, 2021 / © 2021 AT&T Intellectual Property - AT&T Proprietary

Transformation as a Service
Overview



User Interface

- **Identify** input data and data source
- **Define** a target output – key/value mapping pairs and data types
- **View** (in near real-time) the transformation outputs produced
- **Enables** user to **configure** and **visualize** transformation output

User Interface

Simplifies intercommunication and use of data across systems

Reduces complexity

Reduces lead times

Reduces cost

Enable flexible data injections and customization of transformation output

User Interface

Distribution

Receive input from **different** services and distribute transformation outputs to **target** systems

Kafka Integration

User Interface

Distribution

Receive input from **different** services and distribute transformation outputs to **target** systems

Kafka

Kafka Integration

Azure Event Hub

Azure Event Hub Integration

User Interface
Distribution

Receive input from **different** services and distribute transformation outputs to **target** systems

Kafka

Kafka Integration

Azure Event Hub

Azure Event Hub Integration

Redis

Redis Integration

User Interface
Distribution

Receive input from **different** services and distribute transformation outputs to **target** systems

Kafka

Azure Event Hub

Redis

Rest APIs

Kafka Integration

Azure Event Hub Integration

Redis Integration

REST API Integration

AT&T

User Interface
Data Injection / Enrichment

- **Regular expression** releating to data labeling, categorization, values, calculations, address normalization, name standardization to create new fields (**virtual fields**) for a transformation output
- Generate new data points in real-time
- **Merge** multiple source data feeds into a single transformation output
- **Transform** data objects from one format to another

User Interface
Validation

- **Mandatory** Is the field required to be included in the transformaton?
- **Data Type** Is the value supplied the expected data type (STRING, BOOLEAN, INTEGER, etc.)?
- **Not Null** Does a value exist and is it NULL?
- **Available Values** Does the value fall into the given list of expected values?

Backend Processor
Transformation Engine

Type Validation

Type Validation performed as part of Transformation

Incoming data types and target data types are evaluated based on the UI.

Backend Processor
Transformation Engine

Type Validation

Regex

Type Validation performed as part of Transformation

Incoming data types and target data types are evaluated based on the UI.

Regular expression evaluation

Regular expression defined in the UI are evaluated against the incoming data source to create new fields.

Backend Processor
Transformation Engine

Type Validation

Regex

Data Transfer Technology Integration

Type Validation performed as part of Transformation

Incoming data types and target data types are evaluated based on the UI.

Regular expression evaluation

Regular expression defined in the UI are evaluated against the incoming data source to create new fields.

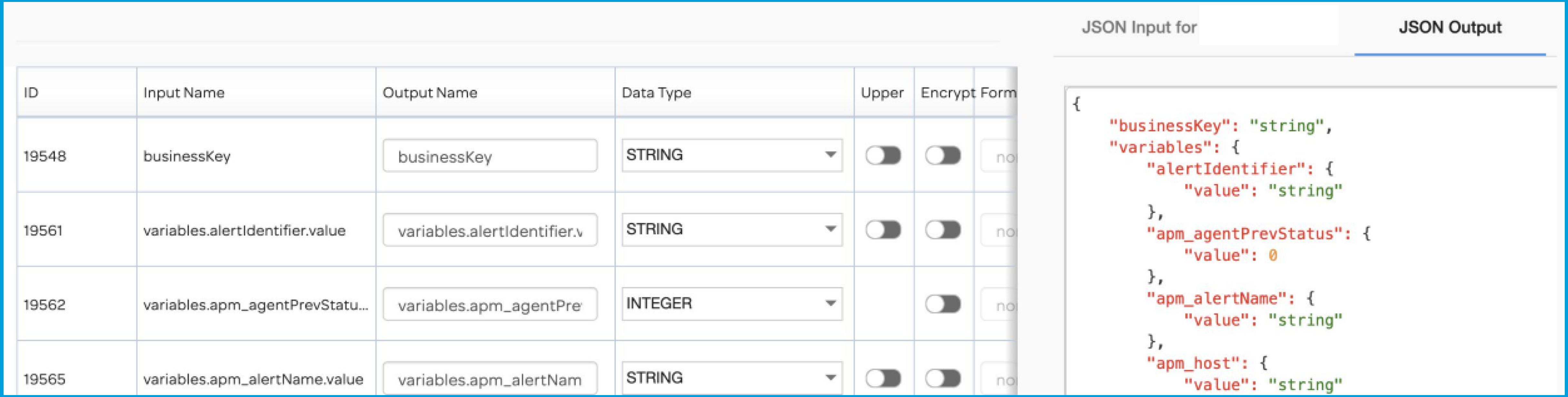
Output Integration

Ability to connect to Kafka topics, Event Hub, Redis and Rest APIs.

2. User Interface

Input Data Interface

- Input text box
- User provides input data
- JSON is validationed
- Data type determined automatically



JSON Input for | JSON Output

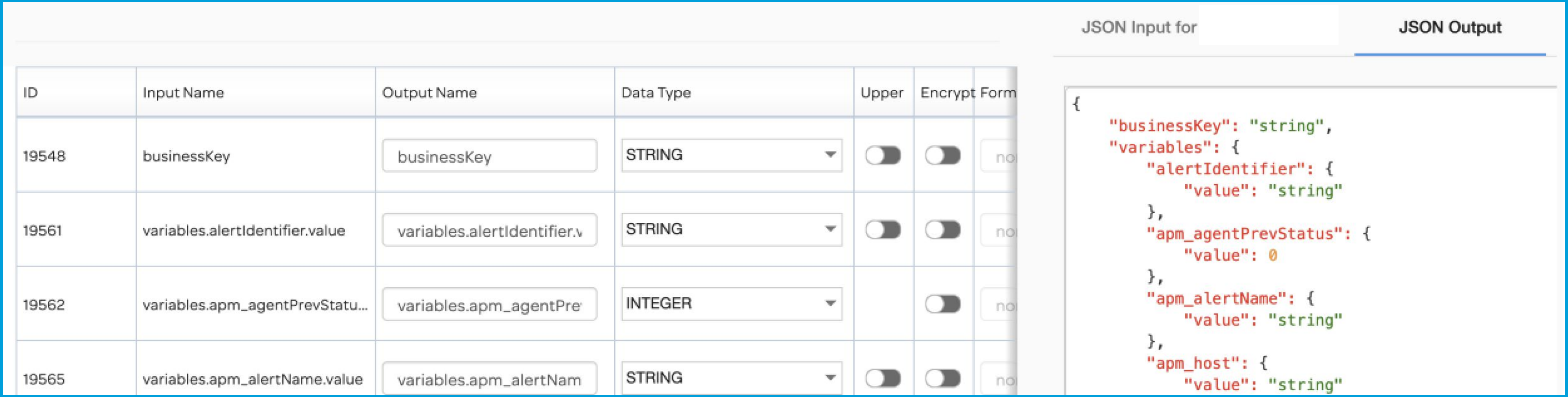
ID	Input Name	Output Name	Data Type	Upper	Encrypt	Form
19548	businessKey	businessKey	STRING			no
19561	variables.alertIdentifier.value	variables.alertIdentifier.v	STRING			no
19562	variables.apm_agentPrevStatu...	variables.apm_agentPre	INTEGER			no
19565	variables.apm_alertName.value	variables.apm_alertNam	STRING			no

```
{
    "businessKey": "string",
    "variables": {
        "alertIdentifier": {
            "value": "string"
        },
        "apm_agentPrevStatus": {
            "value": 0
        },
        "apm_alertName": {
            "value": "string"
        },
        "apm_host": {
            "value": "string"
```

AT&T Business

Output Data Display

- **Preview** of the transformation output
- Editing input and output name will **update** the output JSON in real-time
- Read-Only
- Capable of **merging** multiple data source feeds into single output



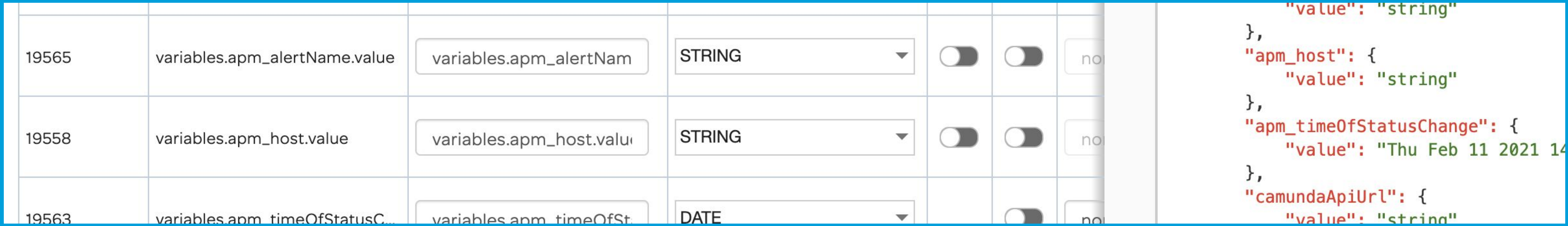
ID	Input Name	Output Name	Data Type	Upper	Encrypt	Form
19548	businessKey	businessKey	STRING			no
19561	variables.alertIdentifier.value	variables.alertIdentifier.v	STRING			no
19562	variables.apm_agentPrevStatu...	variables.apm_agentPre	INTEGER			no
19565	variables.apm_alertName.value	variables.apm_alertNam	STRING			no

JSON Input for | JSON Output

```
{
    "businessKey": "string",
    "variables": {
        "alertIdentifier": {
            "value": "string"
        },
        "apm_agentPrevStatus": {
            "value": 0
        },
        "apm_alertName": {
            "value": "string"
        },
        "apm_host": {
            "value": "string"
```

Transformation Object Table

- Key/value pairs witout nesting (flattened JSON)
- **Unique keys:** reach row represents one transformation object
- Determine data type, input name, output name, case, encrypt, virtual
- **Input Name:** flattened key, entering a period will create a structure with nested values. Entering a square bracket will create an array.
- **Output Name:** final name of a transformation

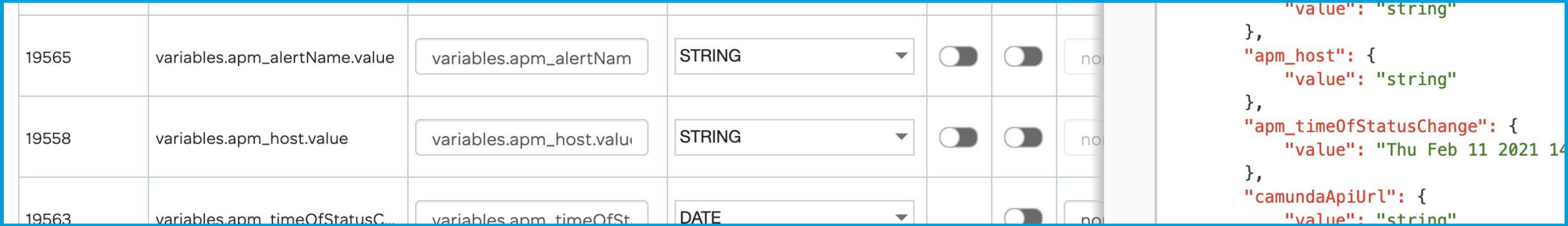


19565	variables.apm_alertName.value	variables.apm_alertNam	STRING	○	○	no
19558	variables.apm_host.value	variables.apm_host.valu	STRING	○	○	no
19563	variables.apm_timeOfStatusC...	variables.apm_timeOfSt	DATE		○	no

```
        "value": "string"
    },
    "apm_host": {
        "value": "string"
    },
    "apm_timeOfStatusChange": {
        "value": "Thu Feb 11 2021 14
    },
    "camundaApiUrl": {
        "value": "string"
```

Transformation Object Table

- **Data Type:** INTEGER, DOUBLE, DATE, STRING. Changing the data type will generate a default value automatially.
- **Upper Case:** change the value to upper case or lower case.
- **Encrypt**: cause the corresponding to become encrypted in the transformation configuration data when stored.



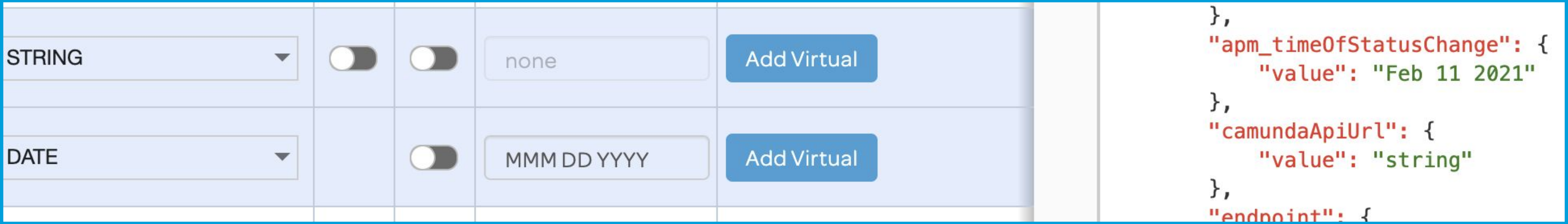
19565	variables.apm_alertName.value	variables.apm_alertNam	STRING	○	○	no
19558	variables.apm_host.value	variables.apm_host.valu	STRING	○	○	no
19563	variables.apm_timeOfStatusC...	variables.apm_timeOfSt	DATE		○	no

```
        "value": "string"
    },
    "apm_host": {
        "value": "string"
    },
    "apm_timeOfStatusChange": {
        "value": "Thu Feb 11 2021 14
    },
    "camundaApiUrl": {
        "value": "string"
```

AT&T Business

Transformation Object Table

- **Format** update a date/time format for the corresponding date value. FOr example, MMM DD YYYY will update the date to **Feb 11 2021**.

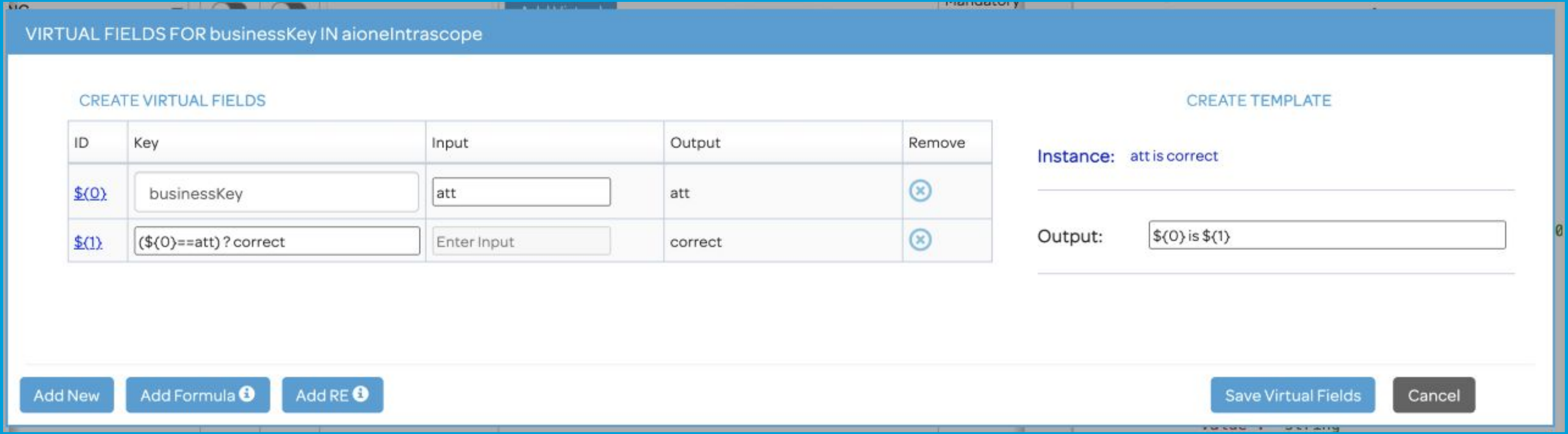


STRING			none	Add Virtual
DATE			MMM DD YYYY	Add Virtual

```
},
"apm_timeOfStatusChange": {
    "value": "Feb 11 2021"
},
"camundaApiUrl": {
    "value": "string"
},
```

Virtual Fields

- Field based on other fields
- Regular, Formula, Regular Expression
- Select a key for the virtual field to be based on



VIRTUAL FIELDS FOR businessKey IN aioneIntrascope

CREATE VIRTUAL FIELDS

ID	Key	Input	Output	Remove
${0}	businessKey	att	att	⊗
${1}	(${0}==att) ? correct	Enter Input	correct	⊗

CREATE TEMPLATE

Instance: att is correct

Output: ${0} is ${1}

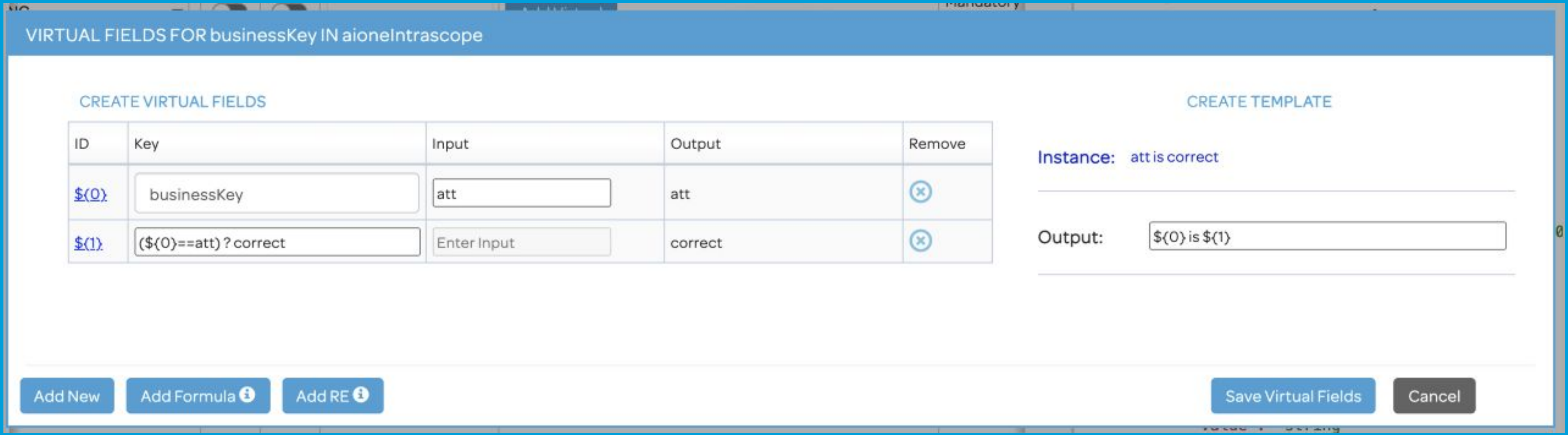
AT&T Business

Virtual Fields

Regular Expression

FirstName + LastName = FirstNameLastName

John + " " + Smith = "John Smith"



VIRTUAL FIELDS FOR businessKey IN aioneIntrascope

CREATE VIRTUAL FIELDS

ID	Key	Input	Output	Remove
${0}	businessKey	att	att	⊗
${1}	(${0}==att) ? correct	Enter Input	correct	⊗

CREATE TEMPLATE

Instance: att is correct

Output: ${0} is ${1}

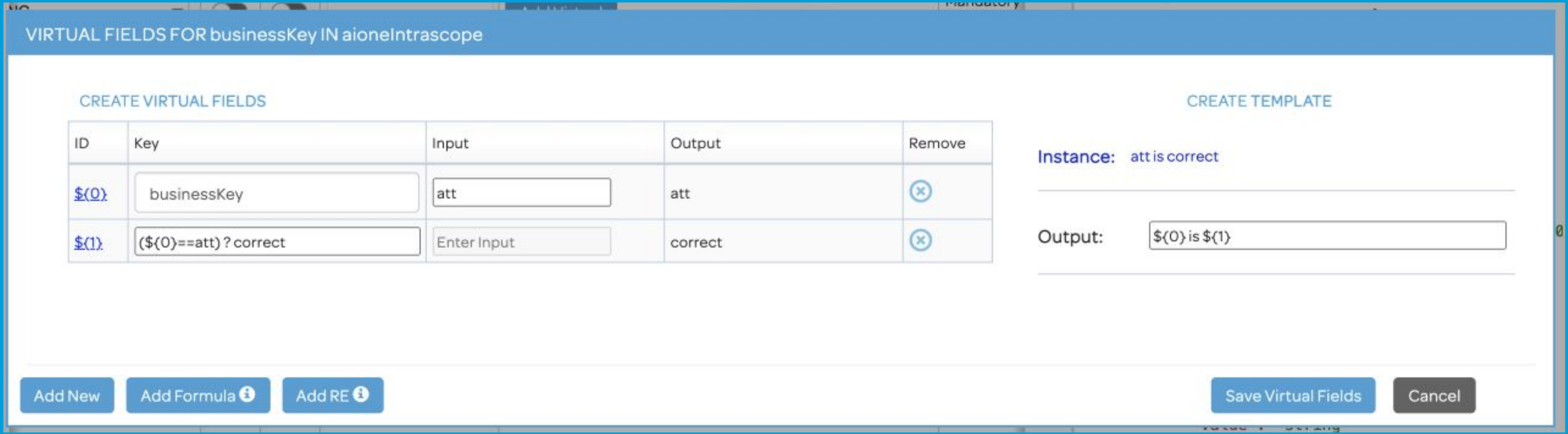
AT&T Business

Virtual Fields

Regular Expression

[A-Za-z]-->X match all uppercase and lowercase letters, replace with X

"123 Main Street" translates to "123 XXXX XXXXXX"



VIRTUAL FIELDS FOR businessKey IN aioneIntrascope

CREATE VIRTUAL FIELDS

ID	Key	Input	Output	Remove
${0}	businessKey	att	att	⊗
${1}	(${0}==att) ? correct	Enter Input	correct	⊗

CREATE TEMPLATE

Instance: att is correct

Output: ${0} is ${1}

AT&T Business

Virtual Fields

Regular Expression using example "john@email.com"

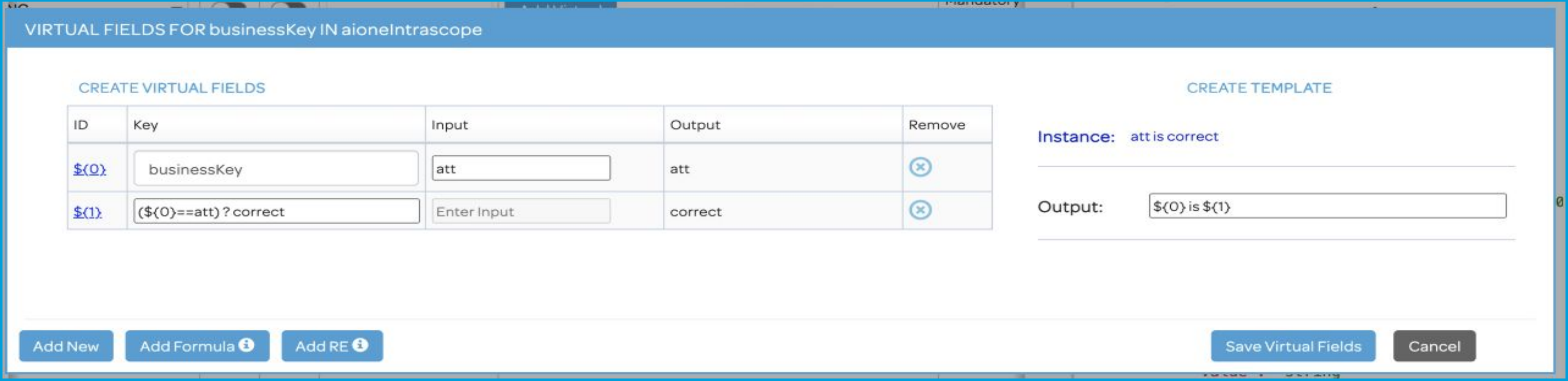
(.+)@(.+) --> $1 **john**

(.+)@(.+) --> $2 **email.com**

Allows for groups.

Dollar sign 1 represents first match group (all text before @ symbol).

Dollar sign 2 represents second match group (all text after @ symbol).



VIRTUAL FIELDS FOR businessKey IN aioneIntrascope

CREATE VIRTUAL FIELDS

ID	Key	Input	Output	Remove
${0}	businessKey	att	att	⊗
${1}	(${0}==att) ? correct	Enter Input	correct	⊗

CREATE TEMPLATE

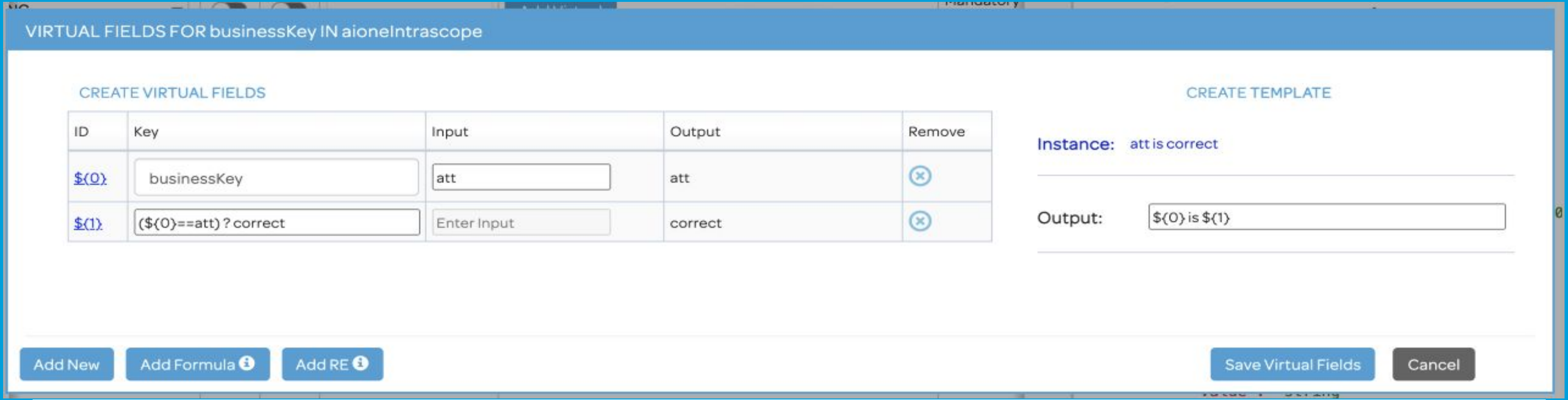
Instance: att is correct

Output: ${0} is ${1}

Virtual Fields

Template

Insert the selected index into an output area. It will present a result, after real values are substitued, either from input data or as the result of a calculation.



VIRTUAL FIELDS FOR businessKey IN aioneIntrascope

CREATE VIRTUAL FIELDS

ID	Key	Input	Output	Remove
${0}	businessKey	att	att	⊗
${1}	(${0}==att) ? correct	Enter Input	correct	⊗

CREATE TEMPLATE

Instance: att is correct

Output: ${0} is ${1}

AT&T Business

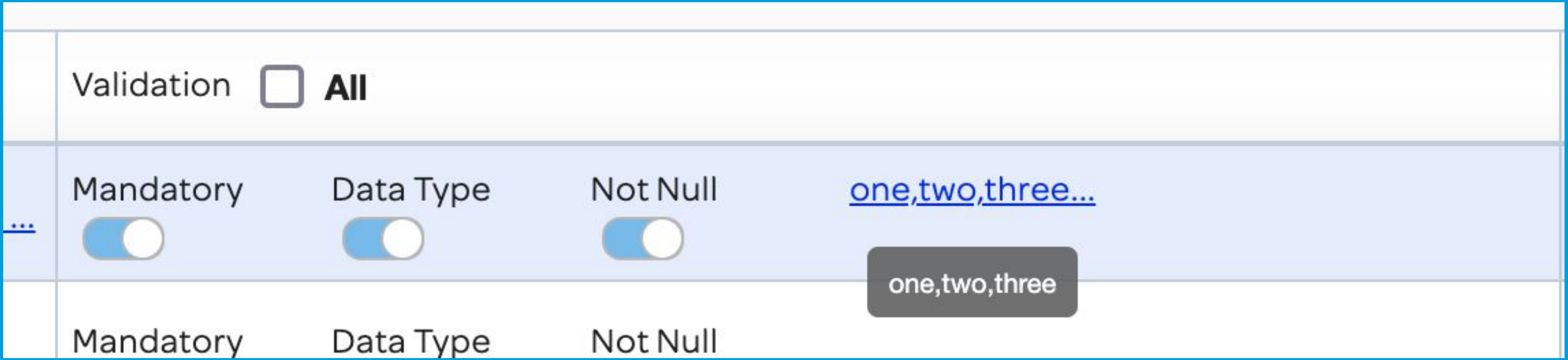
Validation

Mandatory: This key/value pair must exist in the JSON.

Data Type: This value must reflect the assigned data type.

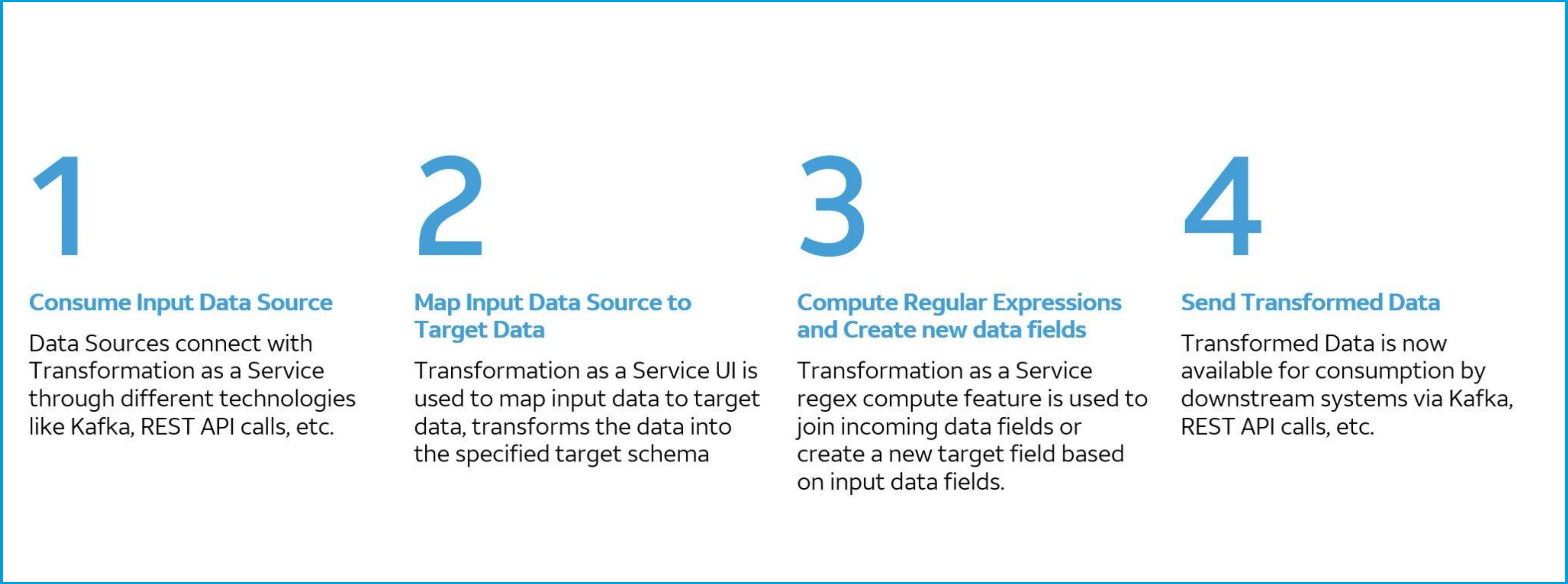
Not Null: This value must not be NULL.

Available Values: This value must be one of the values in the user-defined list.



3. Process Flow

Transformation as a Service
Flow



1

Consume Input Data Source

Data Sources connect with Transformation as a Service through different technologies like Kafka, REST API calls, etc.

2

Map Input Data Source to Target Data

Transformation as a Service UI is used to map input data to target data, transforms the data into the specified target schema

3

Compute Regular Expressions and Create new data fields

Transformation as a Service regex compute feature is used to join incoming data fields or create a new target field based on input data fields.

4

Send Transformed Data

Transformed Data is now available for consumption by downstream systems via Kafka, REST API calls, etc.

THANK YOU

Questions?

AT&T Business

AT&T Business