

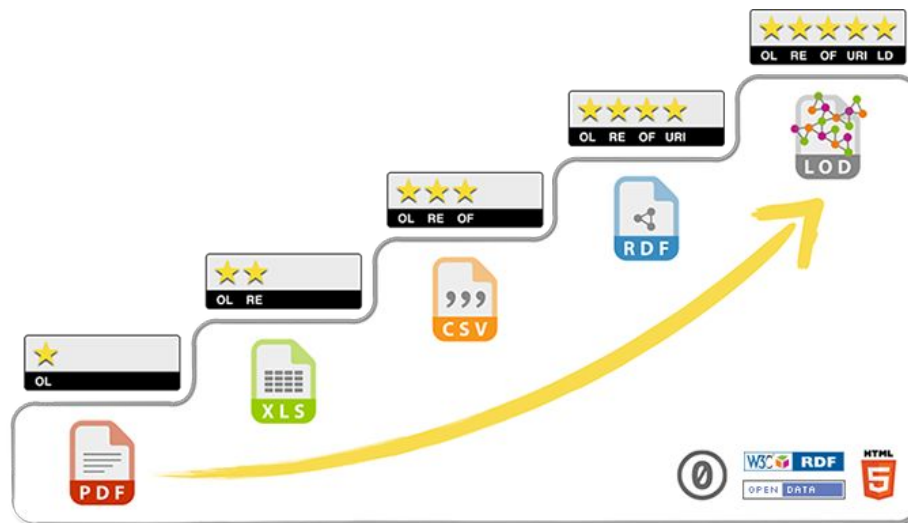
<https://goo.gl/zfcbhg>

LinkedPipes ETL

Jakub Klímek, Petr Škoda

<https://etl.linkedpipes.com/>

Publication of Linked Open Data (LOD)



1. Gather sources (RDB, XML, CSV, JSON, XLSX, proprietary)
2. Analyze them (UML class diagrams)
3. Design transformations to RDF (vocabularies to be used - LOV)
4. Implement transformations to RDF
 - Wrappers (D2RQ)
 - Transformers (RML)
5. Publish files (RDF Turtle, RDF TriG, JSON-LD, ...)
6. Upload to SPARQL endpoint (Virtuoso, Blazegraph, RDF4J, Jena, ...)
7. Register in Open Data catalog (CKAN, DKAN, ...)

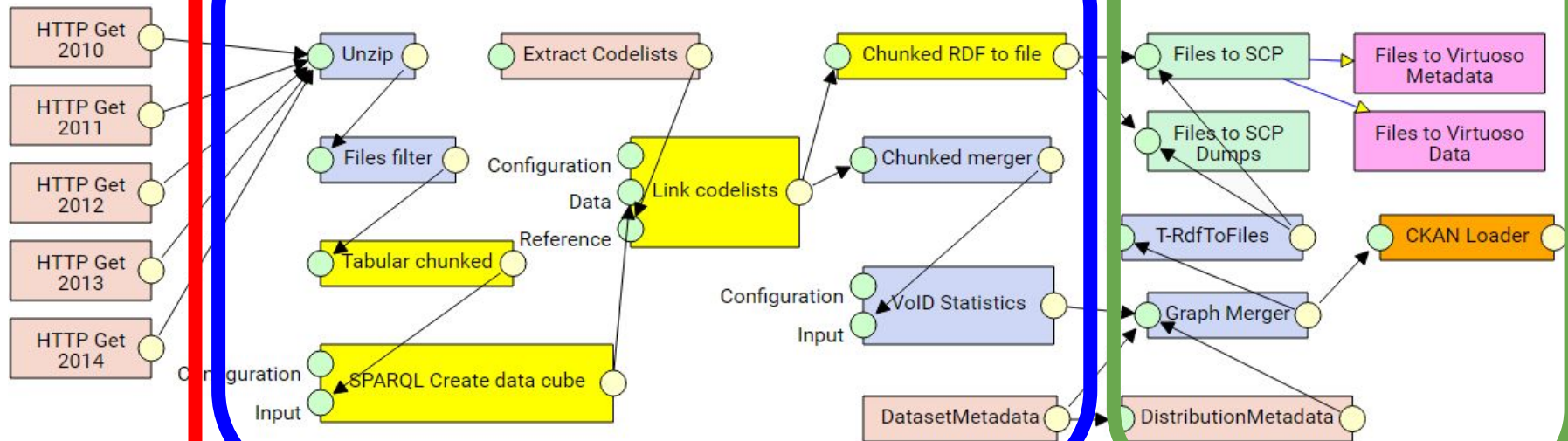
LinkedPipes ETL - **E**xtract **T**ransform **L**oad for LOD

components in a pipeline

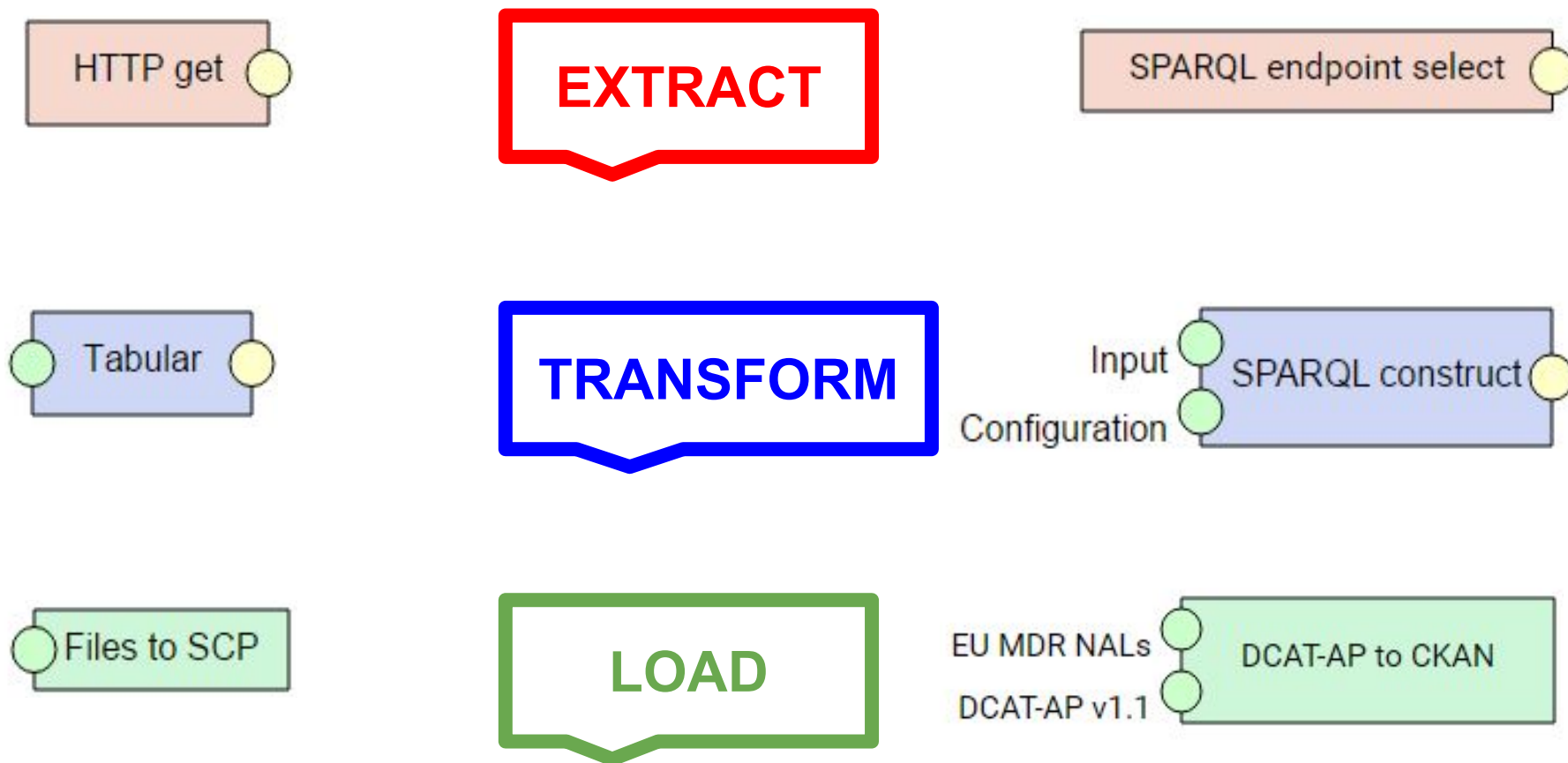
EXTRACT

TRANSFORM

LOAD



LinkedPipes ETL - examples of components



LinkedPipes ETL - Passing Files

1. Component 1 creates files
2. Component 2 consumes files
3. Data is passed as a directory in local file system



LinkedPipes ETL - Passing RDF data

1. Component 1 creates data
2. Component 2 consumes data
3. Data is passed as an RDF4J repository (Native or In-Memory)



Designing a pipeline in LP-ETL

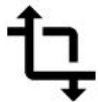
<https://etl.linkedpipes.com/>

LP-ETL Component types



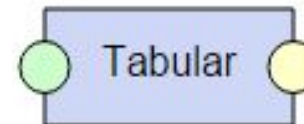
Extractors

Bring data from the outside into the pipeline



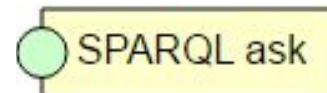
Transformers

Take data from the pipeline, change it and pass it on



Quality Assessment

Check data passing through the pipeline and stop execution in case of problems



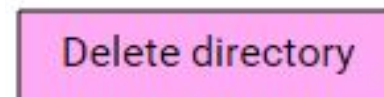
Loaders

Take data from the pipeline and load it to an external place

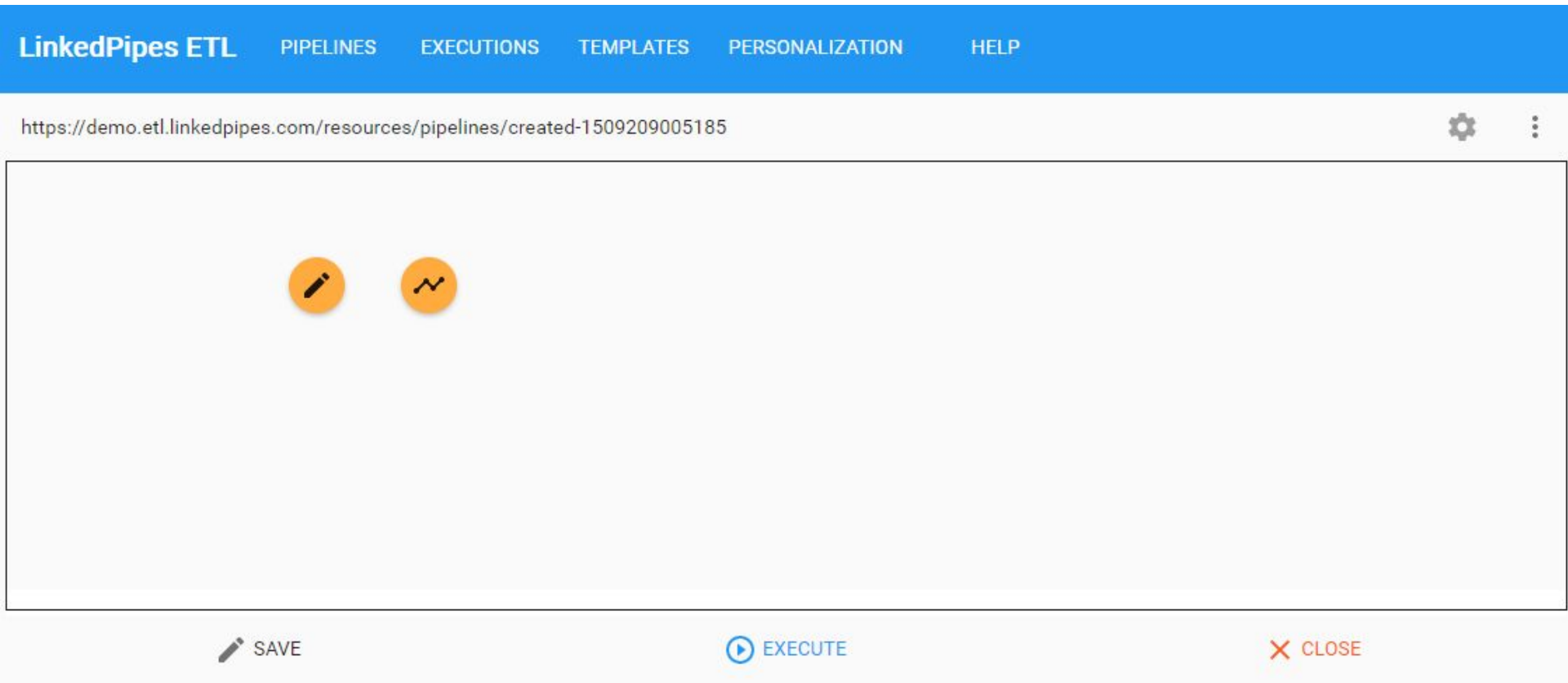


Special

They do something else, e.g. execute a remote command



LP-ETL Pipeline designer



LP-ETL Pipeline designer

The screenshot displays the LP-ETL Pipeline designer web application. The interface features a blue header bar with the following navigation links: **LinkedPipes ETL**, PIPELINES, EXECUTIONS, TEMPLATES, PERSONALIZATION, and HELP. Below the header, a URL bar shows <https://demo.etl.linkedpipes.com/resources/pipelines/>. On the left side of the main workspace, a red rectangular box contains a list:

- label
- function

 A large red arrow points from this box towards the center. In the center, a 'Select component' dialog box is open, featuring a search bar with the text 'get' and a list of three components: 'HTTP get', 'HTTP get list', and 'HTTP get test template'. Each component is preceded by a downward-pointing arrow icon. At the bottom of the interface, there are three buttons: 'SAVE' (with a pencil icon), 'EXECUTE' (with a play button icon), and 'CLOSE' (with a red 'X' icon). The background workspace is currently empty.

LinkedPipes ETL PIPELINES EXECUTIONS TEMPLATES PERSONALIZATION HELP

<https://demo.etl.linkedpipes.com/resources/pipelines/>

- label
- function

Select component



Label search
get

- ↓ HTTP get
- ↓ HTTP get list
- ↓ HTTP get test template


SAVE EXECUTE CLOSE




LP-ETL Pipeline designer

LinkedPipes ETL PIPELINES EXECUTIONS TEMPLATES PERSONALIZATION HELP

<https://demo.etl.linkedpipes.com/resources/pipelines/created-1509209005185>  

Drag from data unit & Drop



 SAVE  EXECUTE  CLOSE

LP-ETL Pipeline designer

The screenshot displays the LP-ETL Pipeline designer web application. The top navigation bar includes links for **LinkedPipes ETL**, **PIPELINES**, **EXECUTIONS**, **TEMPLATES**, **PERSONALIZATION**, and **HELP**. The main workspace shows a pipeline with an **HTTP get** component. A red callout box with an arrow points to the component, containing the text: **Filter by:**



- compatible input
- frequent usage

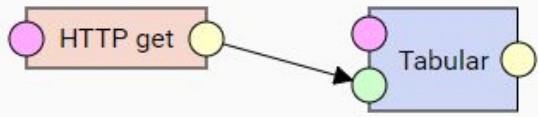
The **Select component** dialog box is open, featuring a **Label search** input field and a list of components: **Tabular ODCS**, **Files to RDF single graph**, and **Tabular**. Each component is preceded by a small icon representing a data flow. At the bottom of the interface, there are buttons for **SAVE**, **EXECUTE**, and **CLOSE**.

<https://etl.linkedpipes.com/components/>




LP-ETL Pipeline designer

LinkedPipes ETL PIPELINES EXECUTIONS TEMPLATES PERSONALIZATION HELP

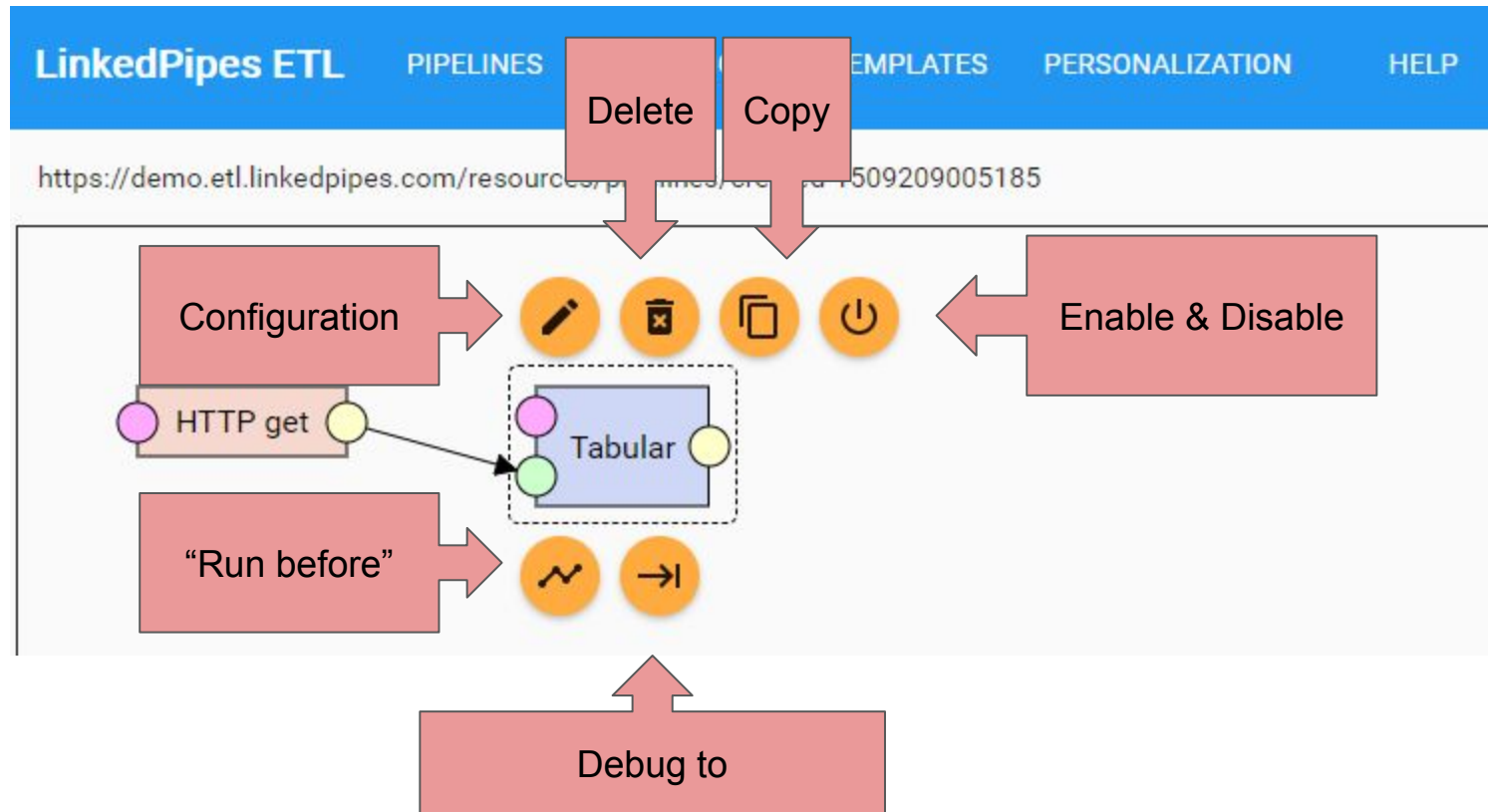
<https://demo.etl.linkedpipes.com/resources/pipelines/created-1509209005185>  



```
graph LR; A(( )) --> B(( ))
```

 SAVE  EXECUTE  CLOSE

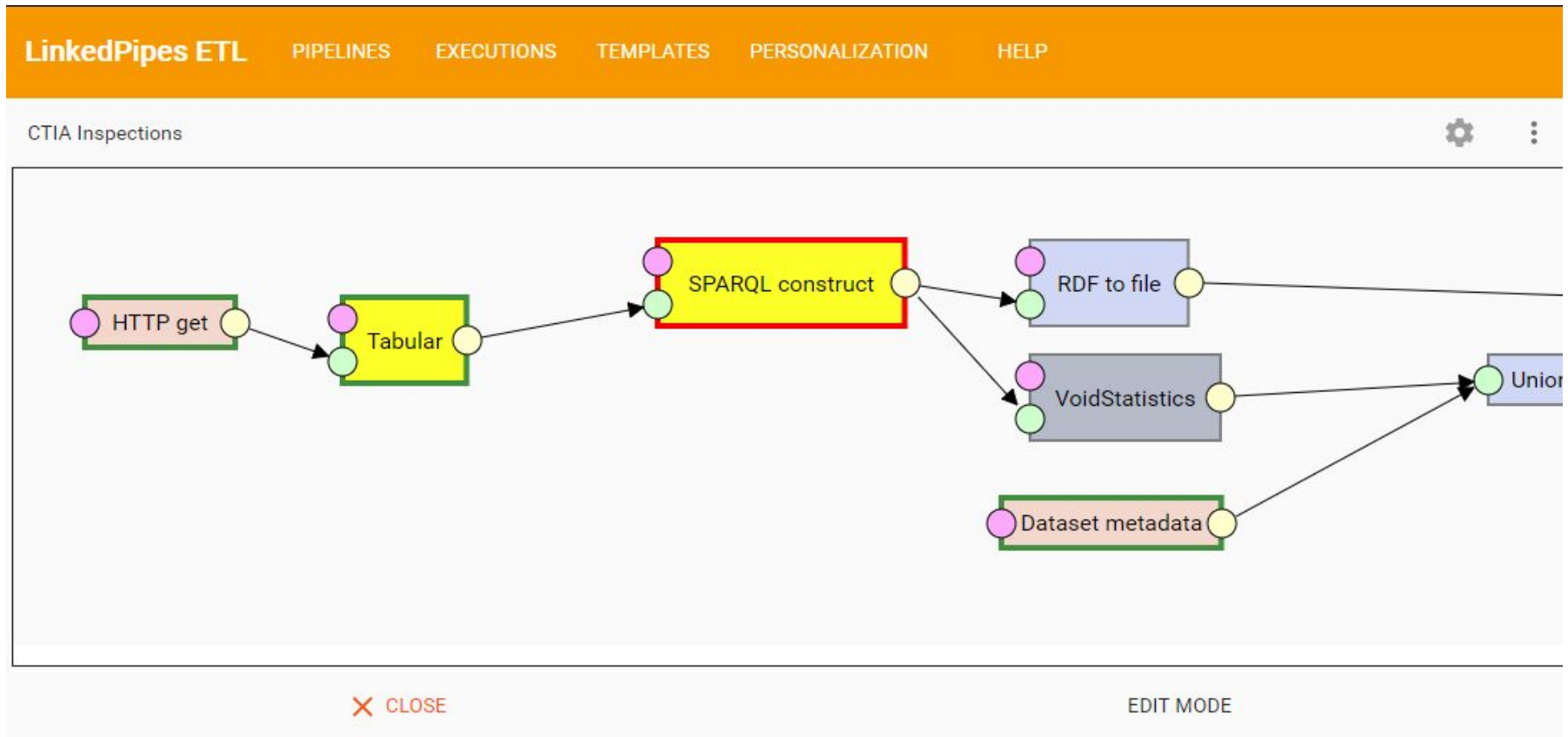
LP-ETL Pipeline designer



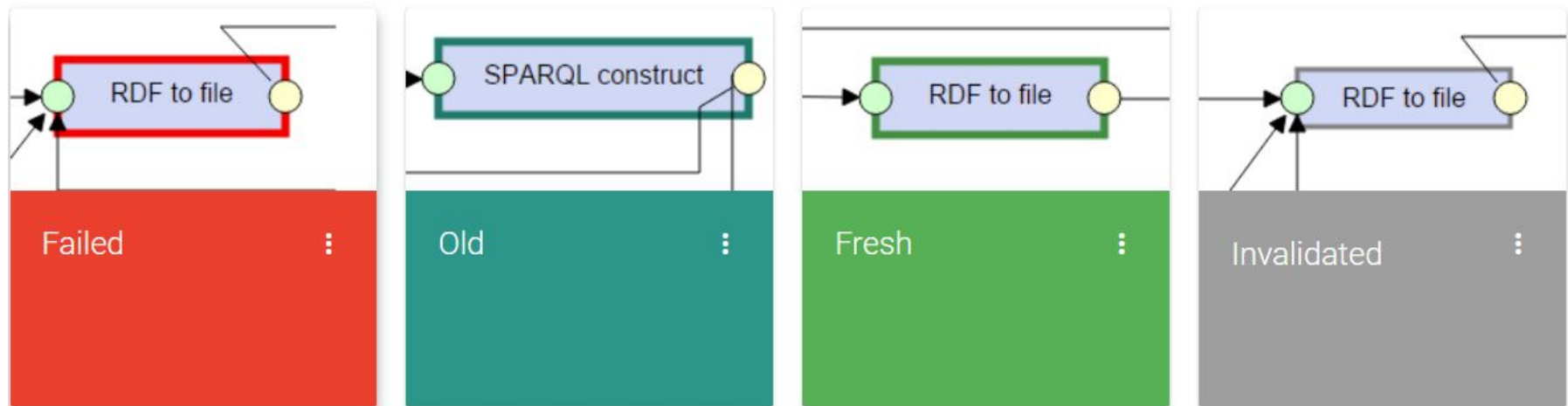
Debugging in LP-ETL

<https://etl.linkedpipes.com/>

LP-ETL Pipeline debugging - overview



LP-ETL Pipeline debugging - component states



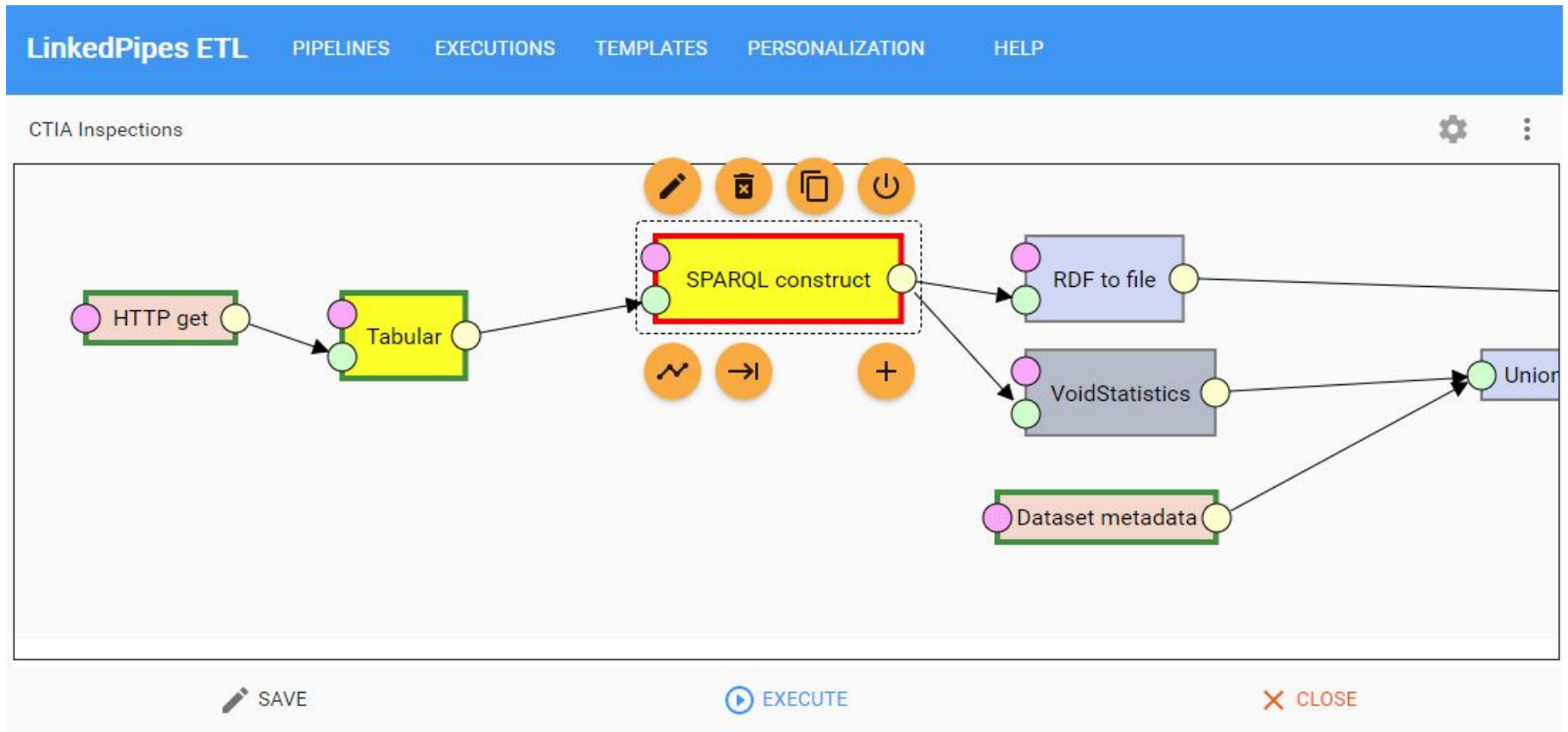
LP-ETL Pipeline debugging - error details

The screenshot displays the LinkedPipes ETL interface. At the top is an orange navigation bar with the following menu items: **LinkedPipes ETL**, PIPELINES, EXECUTIONS, TEMPLATES, PERSONALIZATION, and HELP. Below the navigation bar, the main area is titled 'CTIA Inspections'. On the left, a pipeline diagram shows a component labeled 'HTTP get' (represented by a pink circle and a yellow circle) with an arrow pointing to a 'Union' component (represented by a green circle and a blue rectangle). A modal window titled 'Component execution detail' is open, displaying the following information:

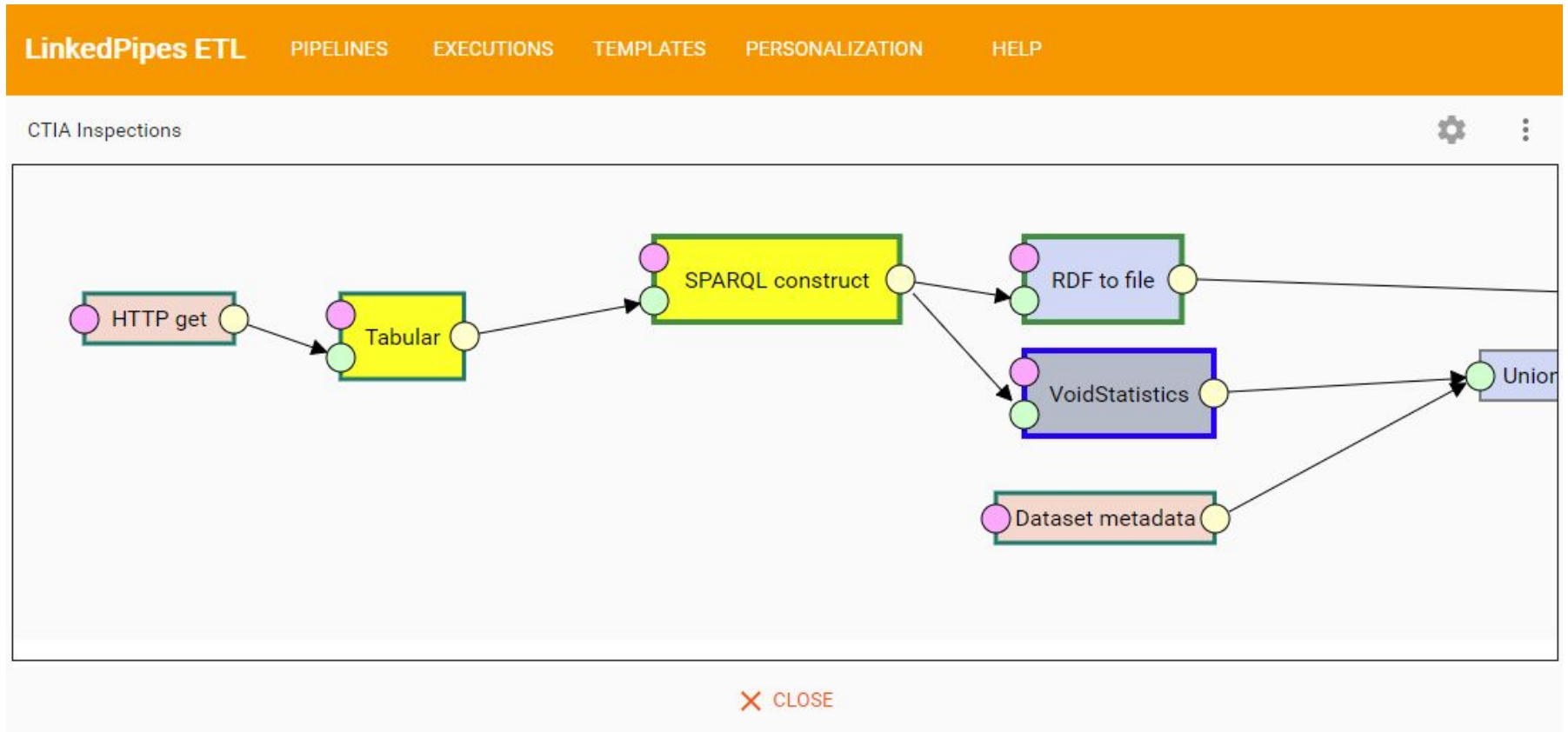
- End: 2017-10-30 07:31:58
- Duration: 00:01:16
- Cause:
Component execution failed.
- Root cause:
TokenMgrError : Lexical error at line 12, column 11. Encountered: " " (32), after : "X"
- Messages:

At the bottom of the modal window is a 'CLOSE DETAIL' button. Below the modal window, there is a red 'X CLOSE' button and an 'EDIT MODE' button.

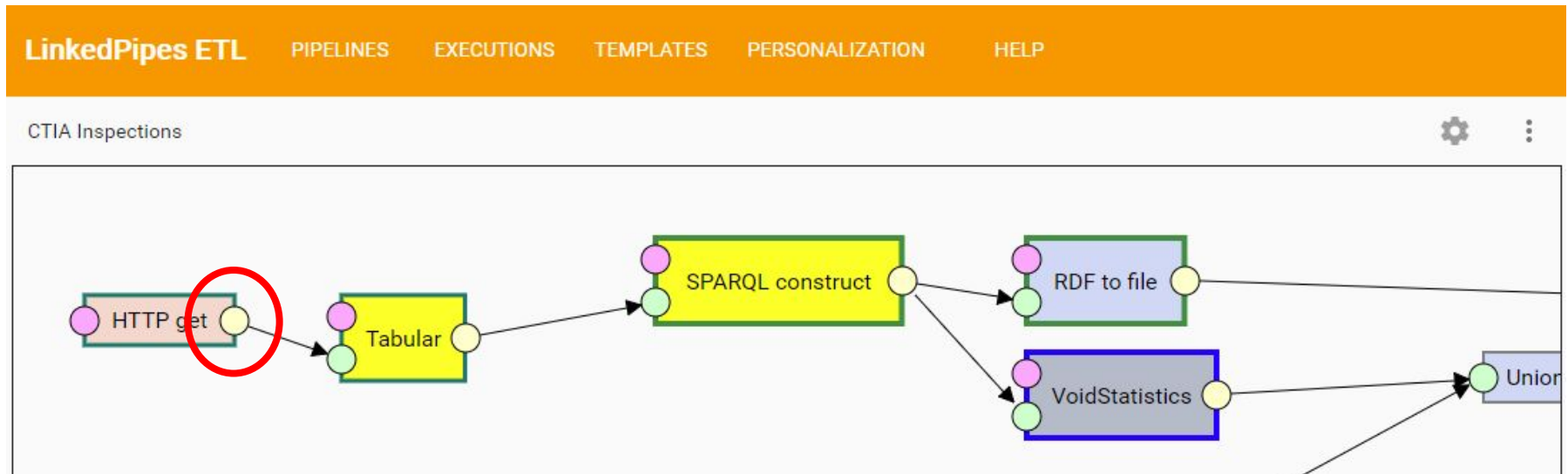
LP-ETL Pipeline debugging - fixing components



LP-ETL Pipeline debugging - “debug from”



LP-ETL Pipeline debugging - debug data - files

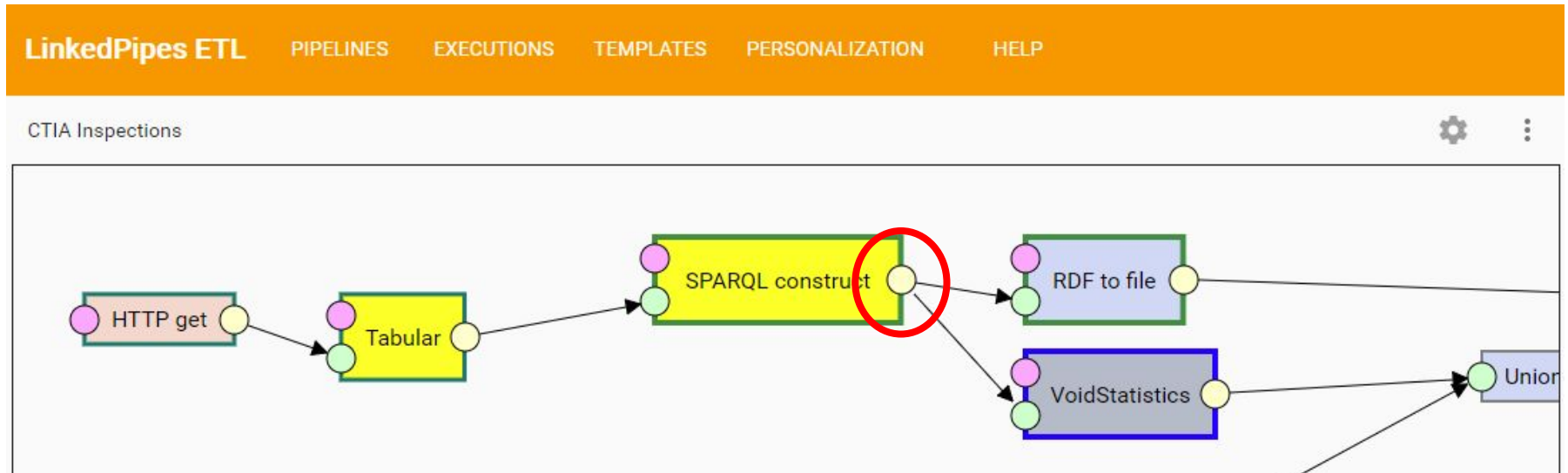


Index of /ff1f83a8-5d68-4bbd-82ea-e5cbbfa129b8/001

[📁 \[parent directory\]](#)

Name	Size	Date Modified
📄 kontroly.csv	13.1 MB	10/30/17, 8:29:00 AM

LP-ETL Pipeline debugging - debug data - RDF

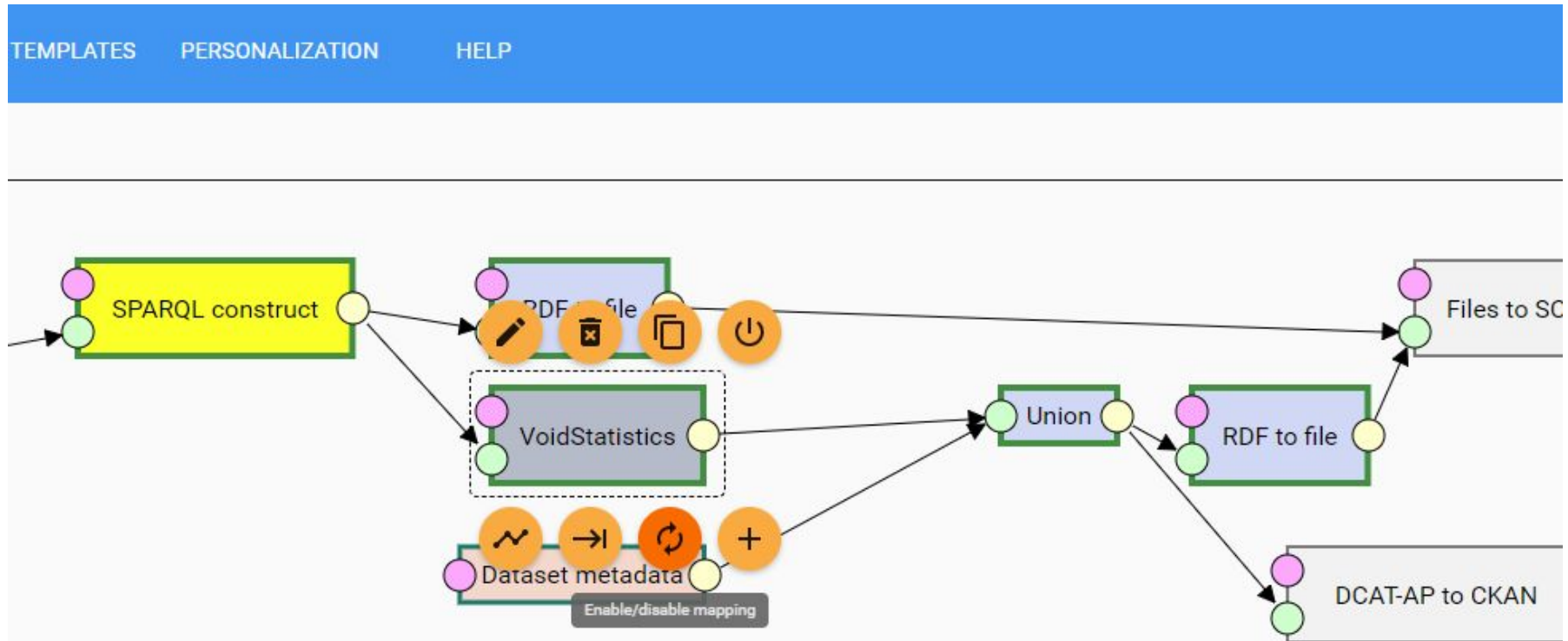


Index of /ff1f83a8-5d68-4bbd-82ea-e5cbbfa129b8/009

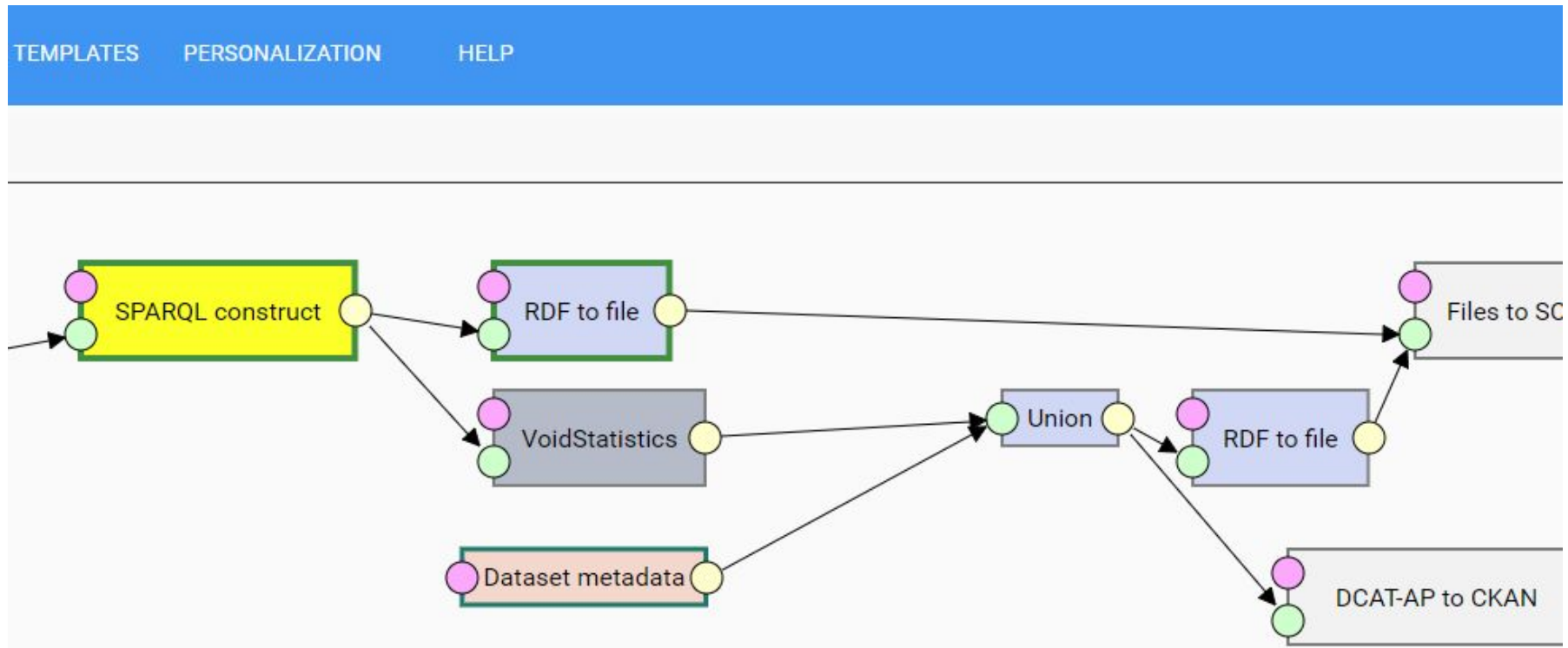
[\[parent directory\]](#)

Name	Size	Date Modified
data.ttl	152 MB	10/30/17, 8:40:00 AM

LP-ETL Pipeline debugging - data invalidation



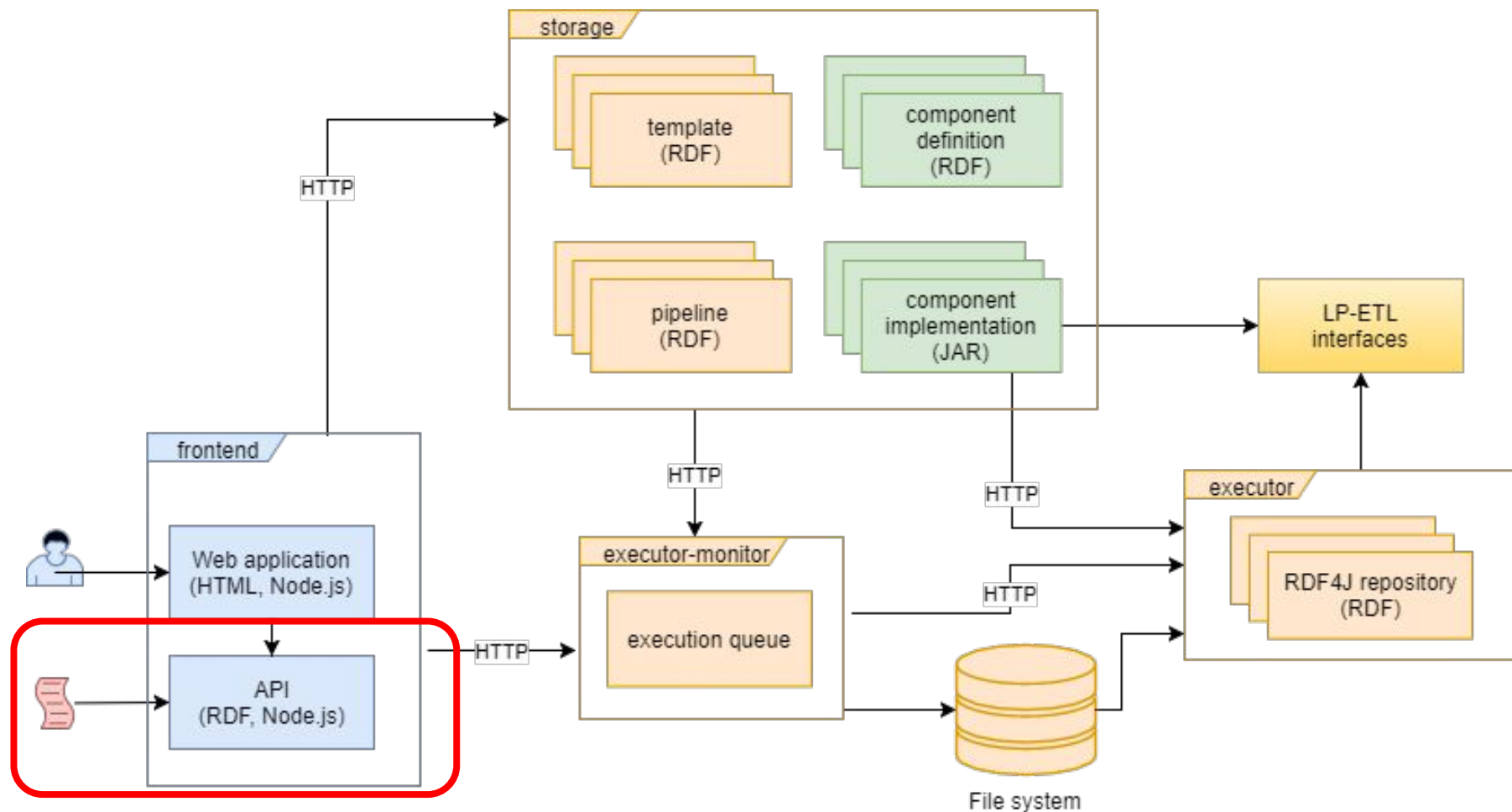
LP-ETL Pipeline debugging - data invalidation



LP-ETL Architecture & API

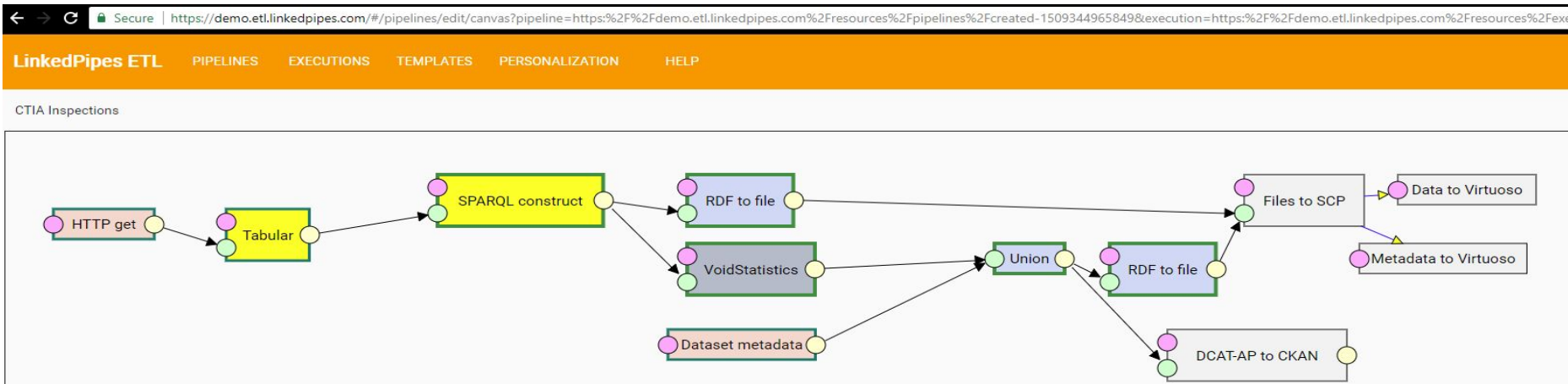
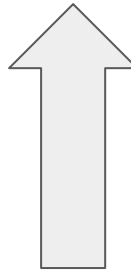
<https://etl.linkedpipes.com/>

LP-ETL Architecture



LP-ETL API

<https://demo.etl.linkedpipes.com/#/pipelines/edit/canvas?pipeline=https:%2F%2Fdemo.etl.linkedpipes.com%2Fresources%2Fpipelines%2Fcreated-1509344965849&execution=https:%2F%2Fdemo.etl.linkedpipes.com%2Fresources%2Fexecution%2F11f83a8-5d68-4bbd-82ea-e5cbbfa129b8>



LP-ETL API - execute a pipeline

```
curl -i -X POST  
https://demo.etl.linkedpipes.com/resources/executions?pipeline=https://demo.etl.linkedp  
ipes.com/resources/pipelines/created-1509344965849
```

```
HTTP/1.1 200 OK  
Date: Mon, 30 Oct 2017 07:27:45 GMT  
Content-Type: application/json; charset=UTF-8
```

```
{"iri": "https://demo.etl.linkedpipes.com/resources/executions/f38208c0-a6a9-410e-8ae2-7  
9c06e27642f"}
```

LP-ETL API - monitor execution

- See pipeline
 - <https://demo.etl.linkedpipes.com/resources/pipelines/created-1509344965849>
- See pipeline execution
 - <https://demo.etl.linkedpipes.com/resources/executions/ff1f83a8-5d68-4bbd-82ea-e5cbbfa129b8>
- Monitor status of a pipeline execution
 - <https://demo.etl.linkedpipes.com/resources/executions/ff1f83a8-5d68-4bbd-82ea-e5cbbfa129b8/overview>

LP-ETL Components Overview

<https://etl.linkedpipes.com/>

LP-ETL Components overview - more than 60

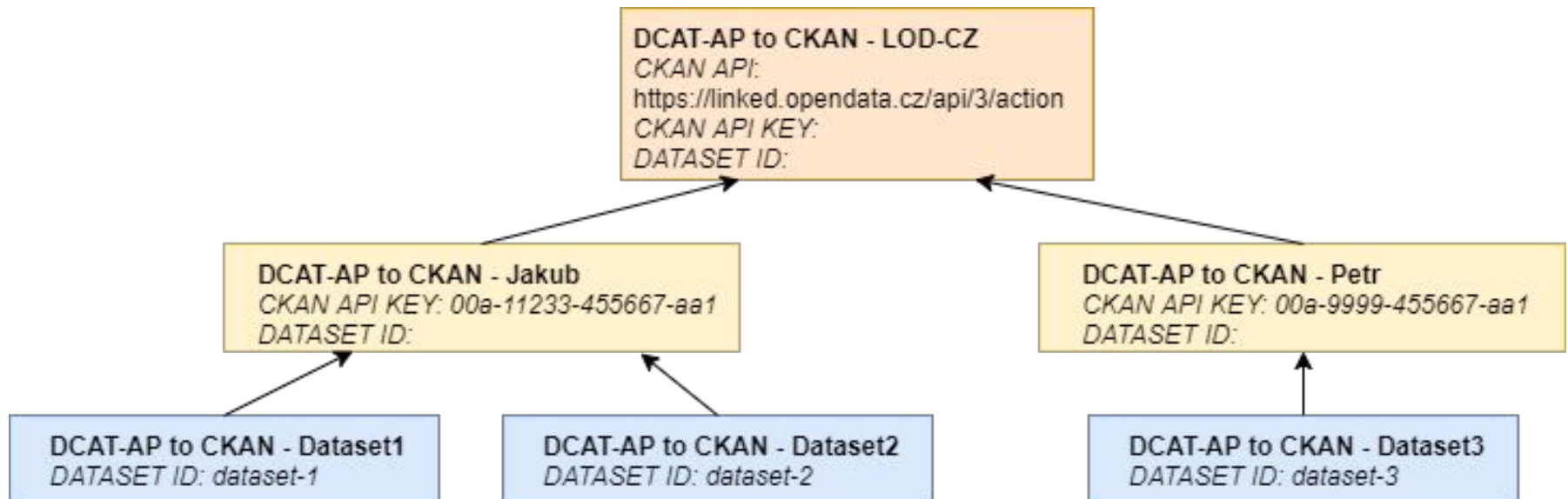
- Extractors
 - HTTP GET, SPARQL endpoint (and variants)
 - Text holder
 - Extract from FTP
 - Pipeline input
 - Files from local
- Transformers
 - SPARQL Construct, SPARQL Update
 - Files to RDF, RDF to file
 - XSLT, JSON to JSON-LD, Tabular, Excel to CSV
 - Bing translator, Decode base64, File hasher, Compress, Decompress, Mustache
- Loaders
 - Files to SCP
 - SPARQL Update, SPARQL Graph Store Protocol
 - DCAT-AP to CKAN
 - Files to local

<https://etl.linkedpipes.com/components/>

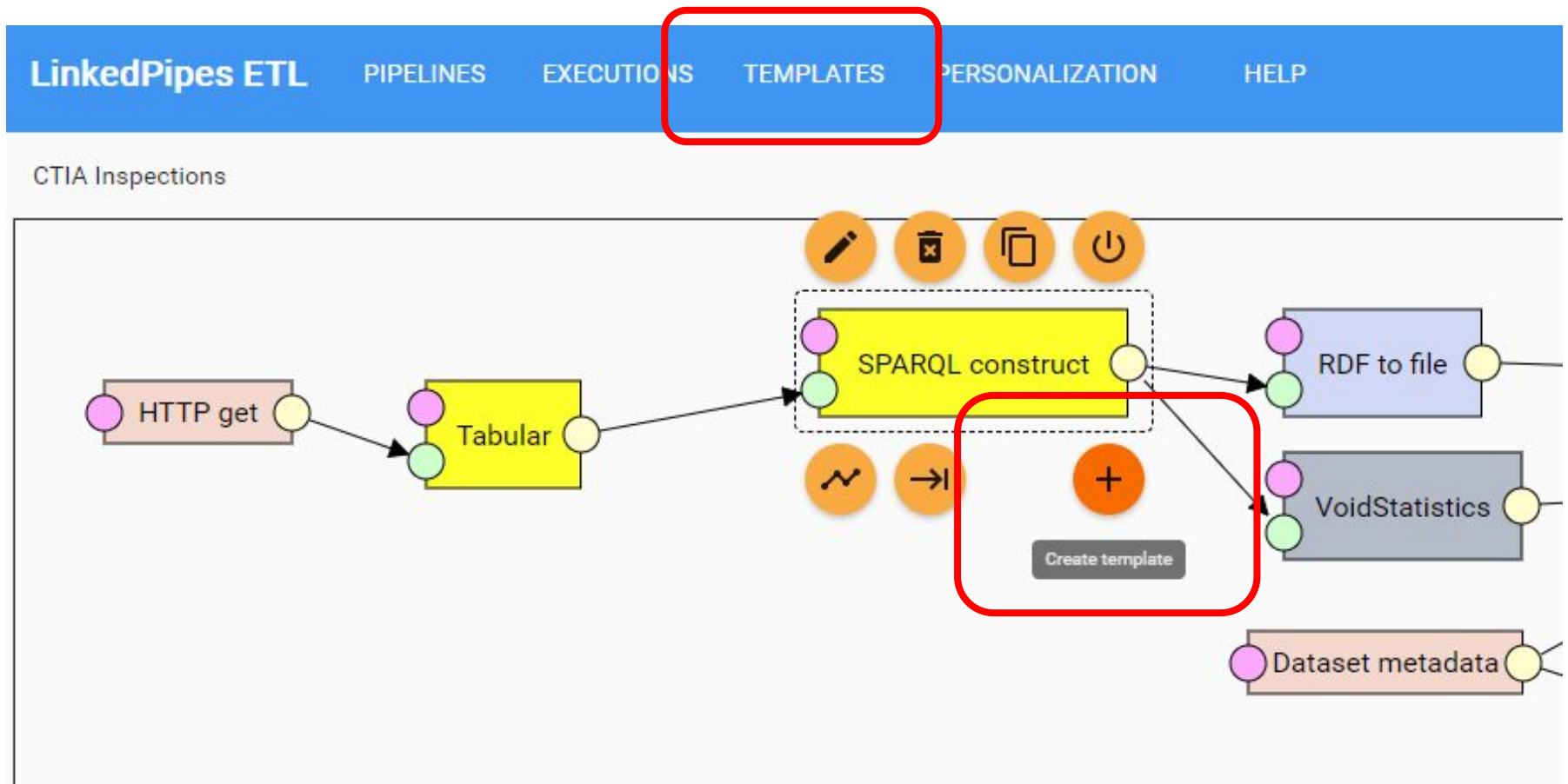
LP-ETL Component Templates

<https://etl.linkedpipes.com/>

LP-ETL Component Templates



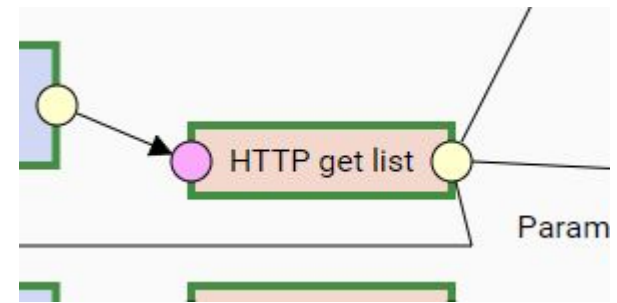
LP-ETL Component Templates



Runtime configuration of LP-ETL components

<https://etl.linkedpipes.com/>

LP-ETL Runtime configuration



```
@prefix httpList: <http://plugins.linkedpipes.com/ontology/e-httpGetFiles#> .
```

```
<http://localhost/resource/configuration> a httpList:Configuration ;  
    httpList:reference <http://localhost/resource/ref/2015-04-30> ,  
<http://localhost/resource/ref/2015-05-18> , <http://localhost/resource/ref/2015-06-16> ,  
<http://localhost/resource/ref/2015-07-21> .
```

```
<http://localhost/resource/ref/2015-04-30> a httpList:Reference ;  
    httpList:fileName "2015-04-30.xml" ;  
    httpList:fileUri "http://portal.gov.cz/portal/rejstriky/data/97898/index-2015-04.xml" .
```

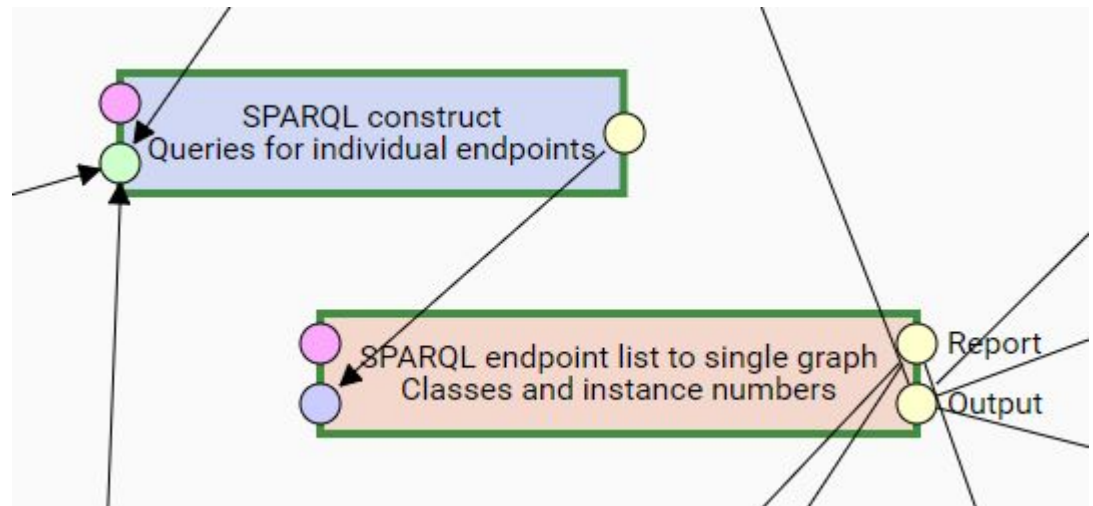
```
<http://localhost/resource/ref/2015-05-18> a httpList:Reference ;  
    httpList:fileName "2015-05-18.xml" ;  
    httpList:fileUri "http://portal.gov.cz/portal/rejstriky/data/97898/index-2015-05.xml" .
```

```
<http://localhost/resource/ref/2015-06-16> a httpList:Reference ;  
    httpList:fileName "2015-06-16.xml" ;  
    httpList:fileUri "http://portal.gov.cz/portal/rejstriky/data/97898/index-2015-06.xml" .
```

```
<http://localhost/resource/ref/2015-07-21> a httpList:Reference ;  
    httpList:fileName "2015-07-21.xml" ;  
    httpList:fileUri "http://portal.gov.cz/portal/rejstriky/data/97898/index-2015-07.xml" .
```

LP-ETL Tasks

```
<https://nkod.opendata.cz/sparql> a <http://plugins.linkedpipes.com/ontology/e-sparqlEndpointList#Task> ;
    <http://plugins.linkedpipes.com/ontology/e-sparqlEndpointList#endpoint> "https://nkod.opendata.cz/sparql" ;
    <http://plugins.linkedpipes.com/ontology/e-sparqlEndpointList#group> "https://nkod.opendata.cz/sparql" ;
    <http://plugins.linkedpipes.com/ontology/e-sparqlEndpointList#query> ""PREFIX adhoc:
<http://linked.opendata.cz/ontology/adhoc/>
CONSTRUCT {
  [] adhoc:class ?Class ;
    adhoc:endpointUri \"https://nkod.opendata.cz/sparql\";
    adhoc:numberOfInstances ?numberOfInstances .
} WHERE {
  {
    SELECT ?Class (COUNT(?resource) AS ?numberOfInstances)
    WHERE {
      ?resource a ?Class.
    }
    GROUP BY ?Class
  }
}"" .
```



Processing larger data in LP-ETL with RDF data chunking

<https://etl.linkedpipes.com/>

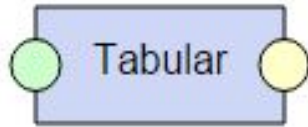
LinkedPipes ETL - Passing RDF data

1. Component 1 creates data
2. Component 2 consumes data
3. Data is passed as an RDF4J repository (Native or In-Memory)



- Small data => OK
- Larger data (e.g. 1 GB CSV file and larger) => **OutOfMemoryException**

LP-ETL - CSV to RDF conversion example



according to
[Generating RDF from Tabular Data on the Web](#) W3C Recommendation

Company name	ID
My First Company	00122344
Unlimited Ltd.	11334499
ACME	99778811
Trading One	19971375
Trading Two	99771133
The Company	00990099

_:1 :Company+name "My First Company" ;
:ID "00122344" .

_:2 :Company+name "Unlimited Ltd." ;
:ID "11334499" .

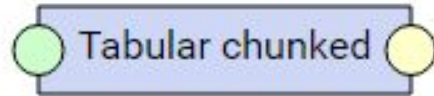
_:3 :Company+name "ACME" ;
:ID "99778811" .

_:4 :Company+name "Trading One" ;
:ID "19971375" .

_:5 :Company+name "Trading Two" ;
:ID "99771133" .

_:6 :Company+name "The Company" ;
:ID "00990099" .

LP-ETL - RDF data chunk size: 6 rows ~ 12 triples



according to
[Generating RDF from Tabular Data on the Web](#) W3C Recommendation

Company name	ID
My First Company	00122344
Unlimited Ltd.	11334499
ACME	99778811
Trading One	19971375
Trading Two	99771133
The Company	00990099

_:1 :Company+name "My First Company" ;
:ID "00122344" .

_:2 :Company+name "Unlimited Ltd." ;
:ID "11334499" .

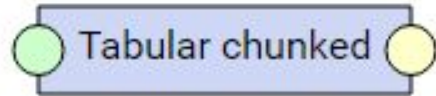
_:3 :Company+name "ACME" ;
:ID "99778811" .

_:4 :Company+name "Trading One" ;
:ID "19971375" .

_:5 :Company+name "Trading Two" ;
:ID "99771133" .

_:6 :Company+name "The Company" ;
:ID "00990099" .

LP-ETL - RDF data chunk size: 2 rows ~ 4 triples



according to
[Generating RDF from Tabular Data on the Web](#) W3C Recommendation

Company name	ID
My First Company	00122344
Unlimited Ltd.	11334499
ACME	99778811
Trading One	19971375
Trading Two	99771133
The Company	00990099

_:1 :Company+name "My First Company" ;
:ID "00122344" .

_:2 :Company+name "Unlimited Ltd." ;
:ID "11334499" .

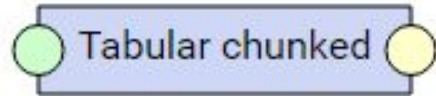
_:3 :Company+name "ACME" ;
:ID "99778811" .

_:4 :Company+name "Trading One" ;
:ID "19971375" .

_:5 :Company+name "Trading Two" ;
:ID "99771133" .

_:6 :Company+name "The Company" ;
:ID "00990099" .

LP-ETL - RDF data chunk size: 1 row ~ 2 triples



according to
[Generating RDF from Tabular Data on the Web](#) W3C Recommendation

Company name	ID
My First Company	00122344
Unlimited Ltd.	11334499
ACME	99778811
Trading One	19971375
Trading Two	99771133
The Company	00990099

_:1 :Company+name "My First Company" ;
:ID "00122344" .

_:2 :Company+name "Unlimited Ltd." ;
:ID "11334499" .

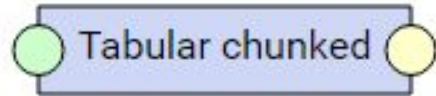
_:3 :Company+name "ACME" ;
:ID "99778811" .

_:4 :Company+name "Trading One" ;
:ID "19971375" .

_:5 :Company+name "Trading Two" ;
:ID "99771133" .

_:6 :Company+name "The Company" ;
:ID "00990099" .

LP-ETL - RDF data chunk: **bad** chunking



according to
[Generating RDF from Tabular Data on the Web](#) W3C Recommendation

Company name	ID
My First Company	00122344
Unlimited Ltd.	11334499
ACME	99778811
Trading One	19971375
Trading Two	99771133
The Company	00990099

_:1 :Company+name "My First Company" ;
:ID "00122344" .

_:2 :Company+name "Unlimited Ltd." ;
:ID "11334499" .

_:3 :Company+name "ACME" ;
:ID "99778811" .

_:4 :Company+name "Trading One" ;
:ID "19971375" .

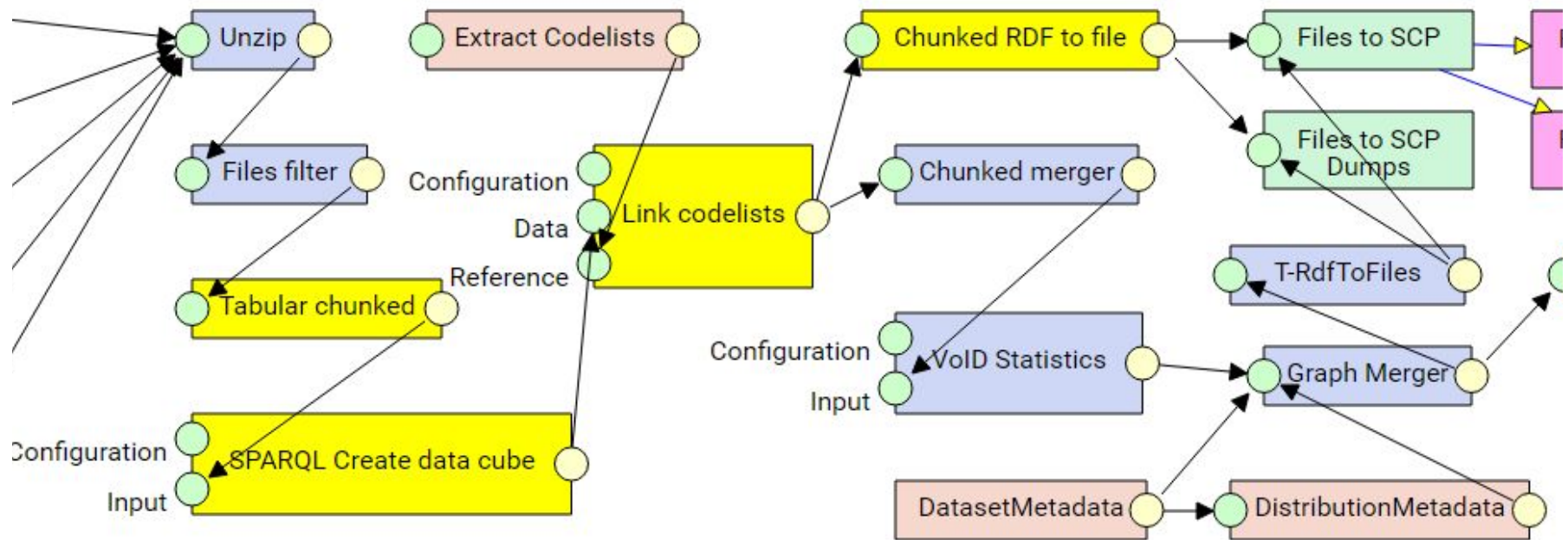
_:5 :Company+name "Trading Two" ;
:ID "99771133" .

_:6 :Company+name "The Company" ;
:ID "00990099" .

LP-ETL - Chunked components in a pipeline

Yellow = Chunked

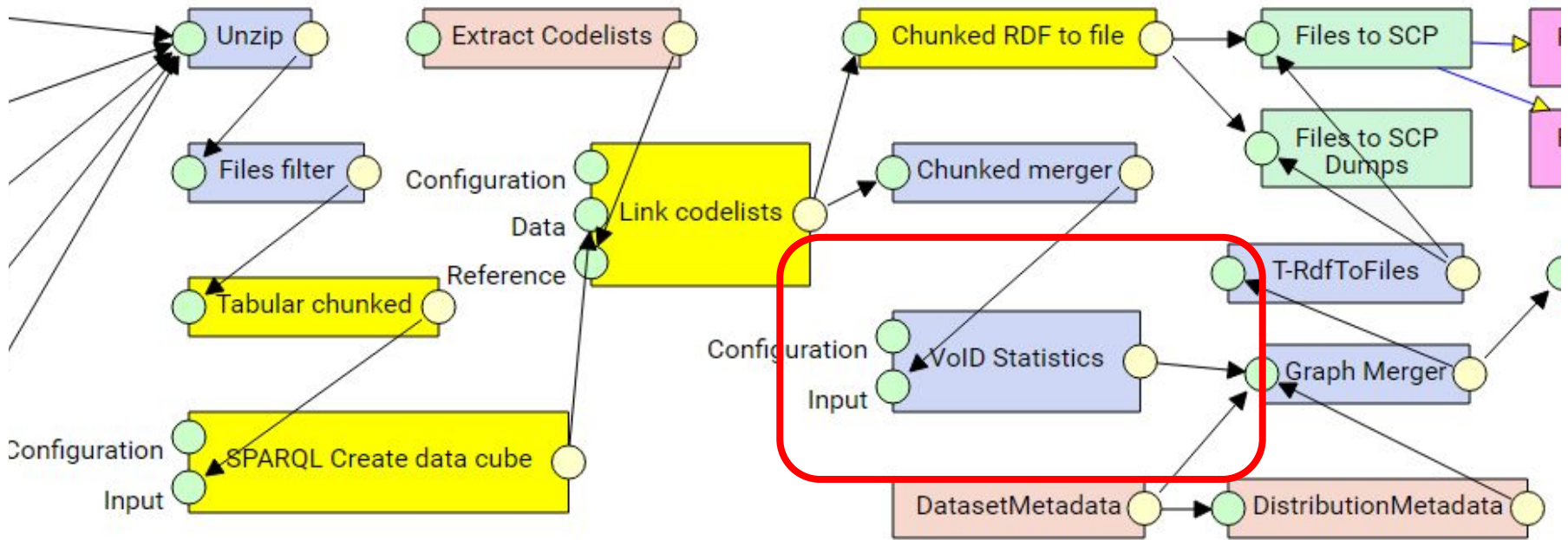
Only N chunks in memory at one time



LP-ETL - Not everything can be solved by chunking

```
SELECT (COUNT (DISTINCT ?s) AS ?dsubjects)
WHERE { ?s ?p ?o }
```

• • •

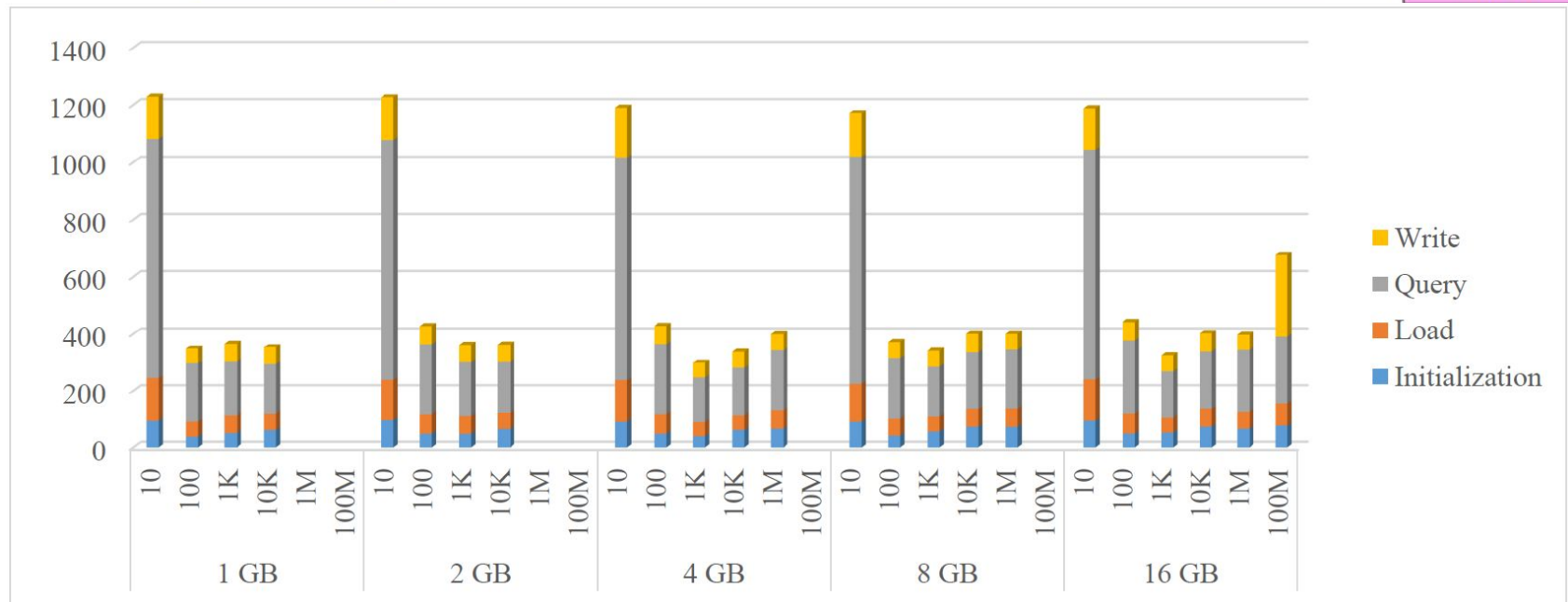
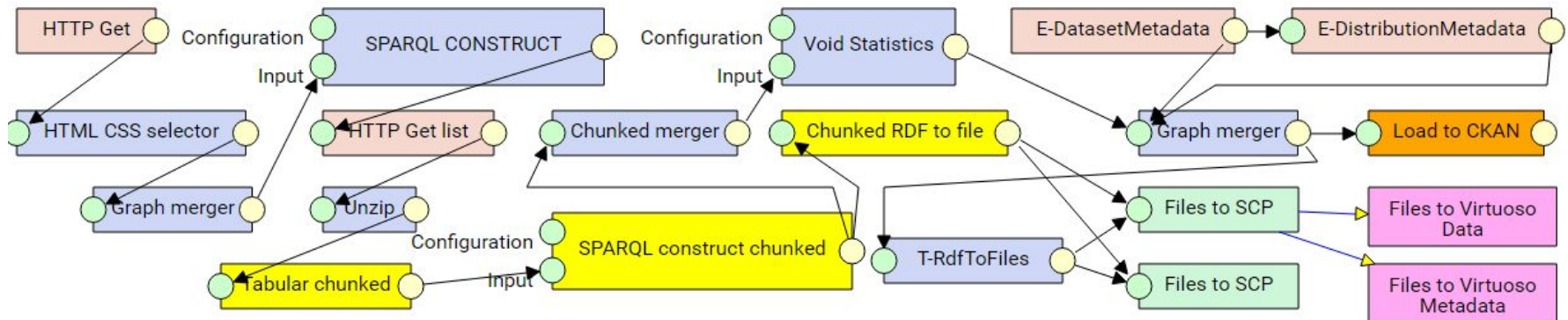


LinkedPipes ETL - Chunked components

- Extractors
 - SPARQL endpoint
- Transformers
 - Creating chunks
 - Tabular
 - Files To RDF
 - Transforming chunks
 - SPARQL Construct
 - SPARQL Update
 - Chunked To Files - each chunk forms an RDF file
 - Mustache - files from data + text templates
 - SPARQL Linker
 - Merging chunks
 - Merger
- Loaders
 - SPARQL Update

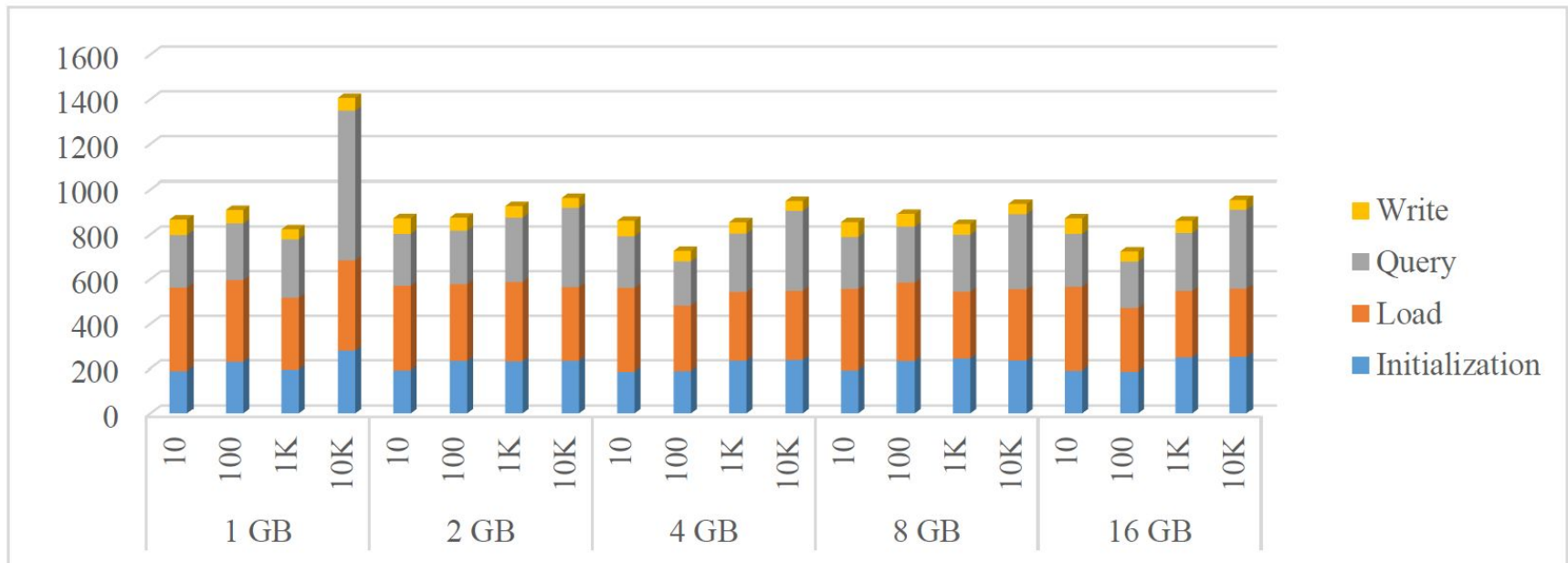
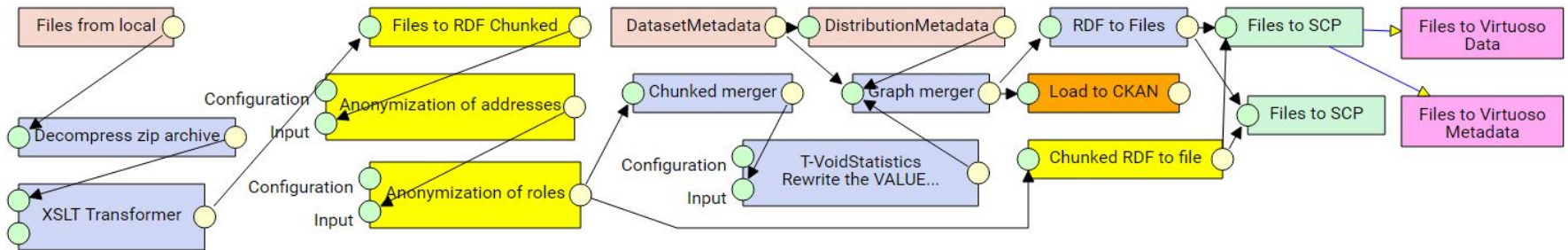
Datasets: List of Czech business entity IDs

- 4.3M rows, 200MB CSV file => 14.5M RDF triples, avg 3.3 triples per record
- 5h 30m = 19 800 seconds w/o chunking in UnifiedViews
- 48m = 2 880 seconds using TARQL



Datasets: Sample of Czech Business Registry

- 220K RDF files = 26M triples => 10M triples (anonymization)
- avg 119.8 triples per record, 856 max, 29 min
- Did not finish in UnifiedViews



LP-ETL Documentation

<https://etl.linkedpipes.com/>

LP-ETL Component Documentation

The screenshot displays the LinkedPipes ETL interface with a modal window titled "SPARQL construct" open. The window has a blue header bar with a red-outlined information icon (i) on the right. Below the header, there are three tabs: "CONFIGURATION" (selected), "INHERITANCE", and "GENERAL". The main content area shows a SPARQL query editor with the following text:

```
SPARQL CONSTRUCT query
PREFIX adhoc: <http://linked.opendata.cz/ontology/adhoc/>
PREFIX ss: <http://plugins.linkedpipes.com/ontology/t-sparqlSelectMultiple#>

CONSTRUCT {
  ?configiri a ss:Configuration ;
    ss:query ?query ;
    ss:fileName ?filename .
} WHERE
{
  {SELECT DISTINCT ?endpoint WHERE {[] adhoc:endpointUri ?
```

At the bottom of the window, there are three buttons: "DISCARD CHANGES", "SAVE CHANGES" (highlighted in blue), and "CLOSE".

In the background, the ETL pipeline editor is visible. It shows a flow starting with an "HTTP get Tabulka endpointu" component, followed by a "Tabular" component, and then a "SPARQL select nodes.csv" component. A "SAVE" button is located at the bottom left of the pipeline editor.

LP-ETL Component Documentation

<https://etl.linkedpipes.com/components/t-sparqlconstruct>

LinkedPipes > ETL > Components > SPARQL construct



SPARQL construct

Transformer, allows the user to transform RDF data using a **CONSTRUCT query**. This means that based on the input data, output data will be generated using this query.

SPARQL CONSTRUCT query

Query for transformation of triples

Characteristics

ID

t-sparqlconstruct

Type

The SPARQL construct component queries input data using a **SPARQL CONSTRUCT** query producing new

LP-ETL Component Documentation

Mustache template

The `{{ mustache }}` template

Characteristics

ID

t-mustache

Type

transformer

Inputs

RDF data

Outputs

Files

Look in pipeline

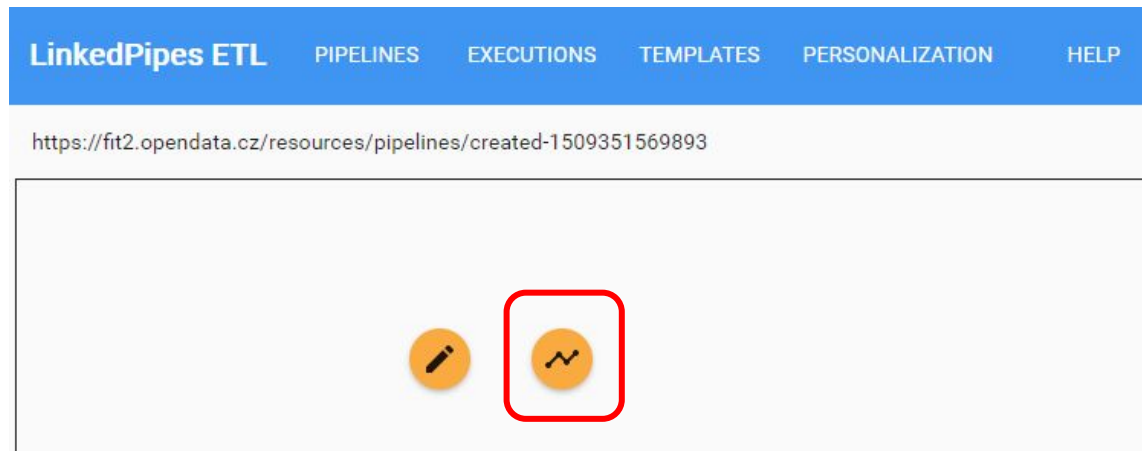


Sample pipeline

[available](#)

First, we recommend the potential user of this component to get to know the library itself and [its demo](#). There, a template and a sample JSON file with data are shown and the resulting text can be generated. In LinkedPipes ETL, we work with RDF, therefore, the template placeholders will be IRIs and the data will be stored as RDF.

LP-ETL Pipeline fragments



This screenshot shows the 'From URL' section of the LinkedPipes ETL interface. The top navigation bar is identical to the previous image. Below it, there are three options: 'Upload file' (unselected), 'From URL' (selected, indicated by an orange circle), and 'Update existing templates' (disabled, indicated by a greyed-out toggle). Below these options, the text 'Pipeline URL *' is followed by the URL <https://etl.linkedpipes.com/assets/pipelines/t-mustache-1.jsonld>. At the bottom left, there is a blue button labeled 'UPLOAD'.

LP-ETL Pipeline fragments

LinkedPipes ETL

PIPELINES

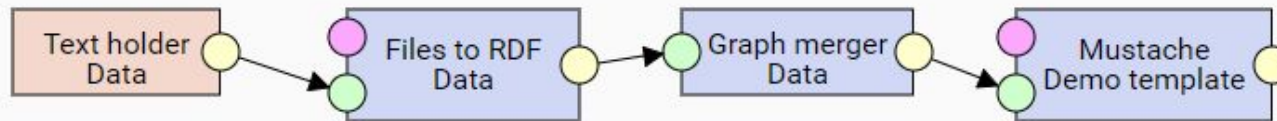
EXECUTIONS

TEMPLATES

PERSONALIZATION

HELP

<https://fit2.opendata.cz/resources/pipelines/created-1509351569893>



 SAVE

 EXECUTE

 CLOSE

LP-ETL Tutorials & How-Tos

LinkedPipes > ETL

ETL: Extract Transform Load for Linked Data

DEMO

INSTALL

GITHUB

READ

LEARN

What's new

2017-10-16: [LinkedPipes ETL @ ODBASE 2017!](#)

[Data chunking](#) in LinkedPipes ETL will be presented at [ODBASE 2017!](#)
See you in Rhodes, Greece!

2017-08-17: [A set of simple how-tos added](#)

2017-07-31: [Geocoding with nominatim tutorial available](#)

2017-06-30: [Tabular data to RDF tutorial available](#)

Featured component

Pipeline input

Pipeline input

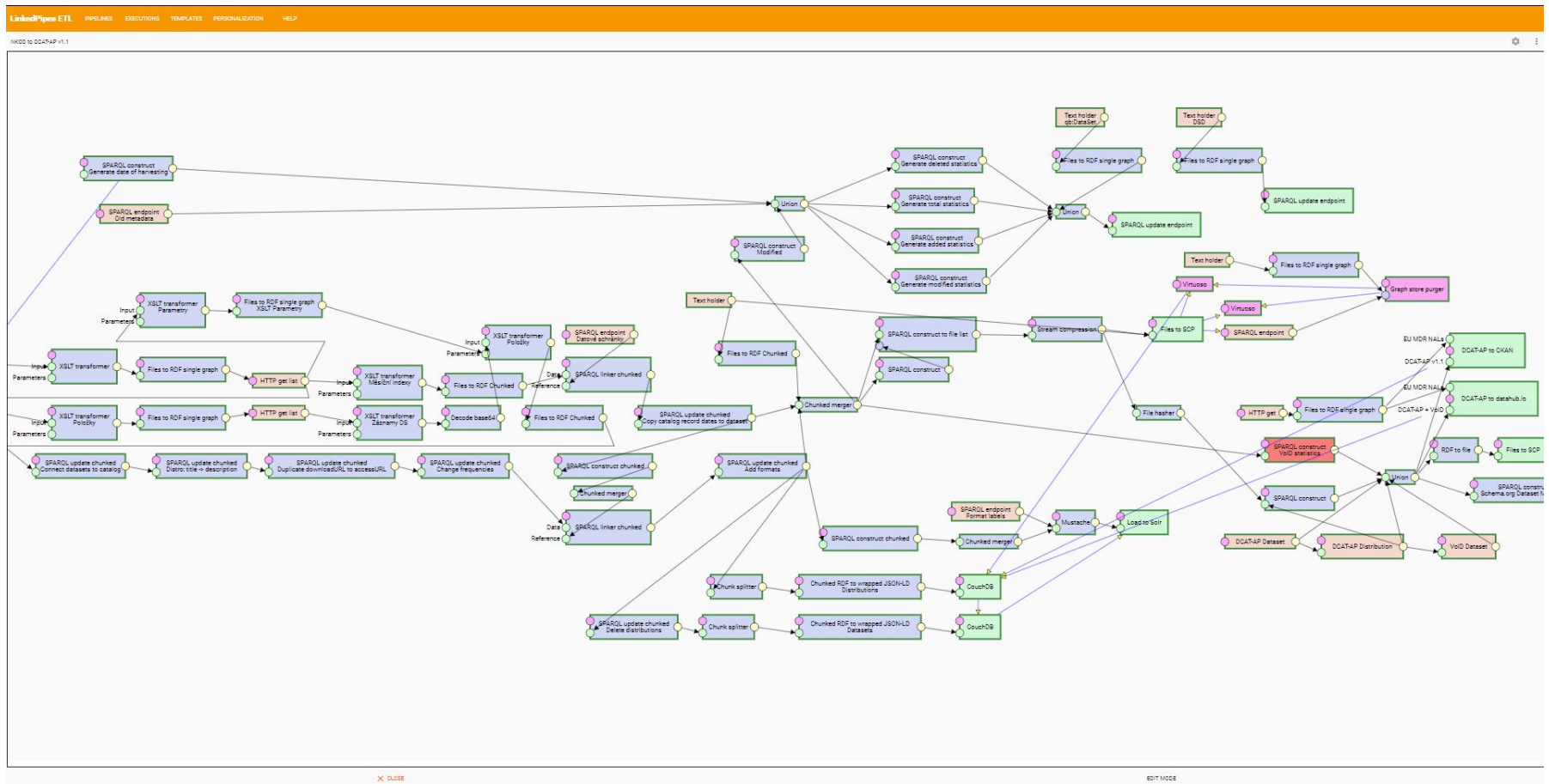
Passes files from an HTTP POST pipeline execution request

MORE COMPONENTS

LP-ETL Tutorials & How-Tos

- <https://etl.linkedpipes.com/tutorials/>
- Tutorials
 - Converting tabular data to RDF
 - Geocoding with Nominatim
- How-Tos
 - How to generate SPARQL queries via Mustache
 - How to cache linked data
 - How to map RDF properties using SPARQL Update
 - How to test data
 - How to process large RDF data
 - How to load data to Virtuoso
 - How to compress and decompress data
 - How to input fixed data
 - How to map JSON to RDF
 - How to convert CSV to RDF
 - How to convert RDF to CSV
 - How to convert XML to RDF

LP-ETL NKOD Pipeline



LP-ETL Known instances

- Charles University - teaching LOD, research projects, OpenData.cz datasets
- Czech Technical University - teaching LOD
- University of Economics - teaching LOD, research projects, data publishing
- Ministry of the Interior of the Czech Republic - National Open Data Catalog transformation and data quality
- Datlowe - working with semantic models for text processing
- Municipality of Brno - data publishing and catalogization
- Greek National Library
- Fraunhofer IAIS - OpenBudgets.eu platform

Advanced RDF data consumption using LP-ETL

<https://etl.linkedpipes.com/>

LP-ETL and larger data

```
CONSTRUCT {  
  ?point a gml:Point;  
    gml:pos ?o .  
}  
WHERE  
{  
  ?point a gml:Point;  
    gml:pos ?o .  
}
```

LP-ETL - **bad** chunking ~ bad paging

```
CONSTRUCT {  
  ?point a gml:Point;  
    gml:pos ?o .  
}  
WHERE  
{  
  ?point a gml:Point;  
    gml:pos ?o .  
}
```

LIMIT 3

_:1 :Company+name "My First Company" ;
:ID "00122344" .

_:2 :Company+name "Unlimited Ltd." ;
:ID "11334499" .

_:3 :Company+name "ACME" ;
:ID "99778811" .

_:4 :Company+name "Trading One" ;
:ID "19971375" .

_:5 :Company+name "Trading Two" ;
:ID "99771133" .

_:6 :Company+name "The Company" ;
:ID "00990099" .

LP-ETL and larger data

```
SELECT ?point
WHERE {
    [] ruian:adresniBod ?point.
}
```

```
SELECT ?point
WHERE {
    [] ruian:adresniBod ?point.
}
```

```
LIMIT 100
OFFSET 10200
```

HttpException: 500 SPARQL Request Failed

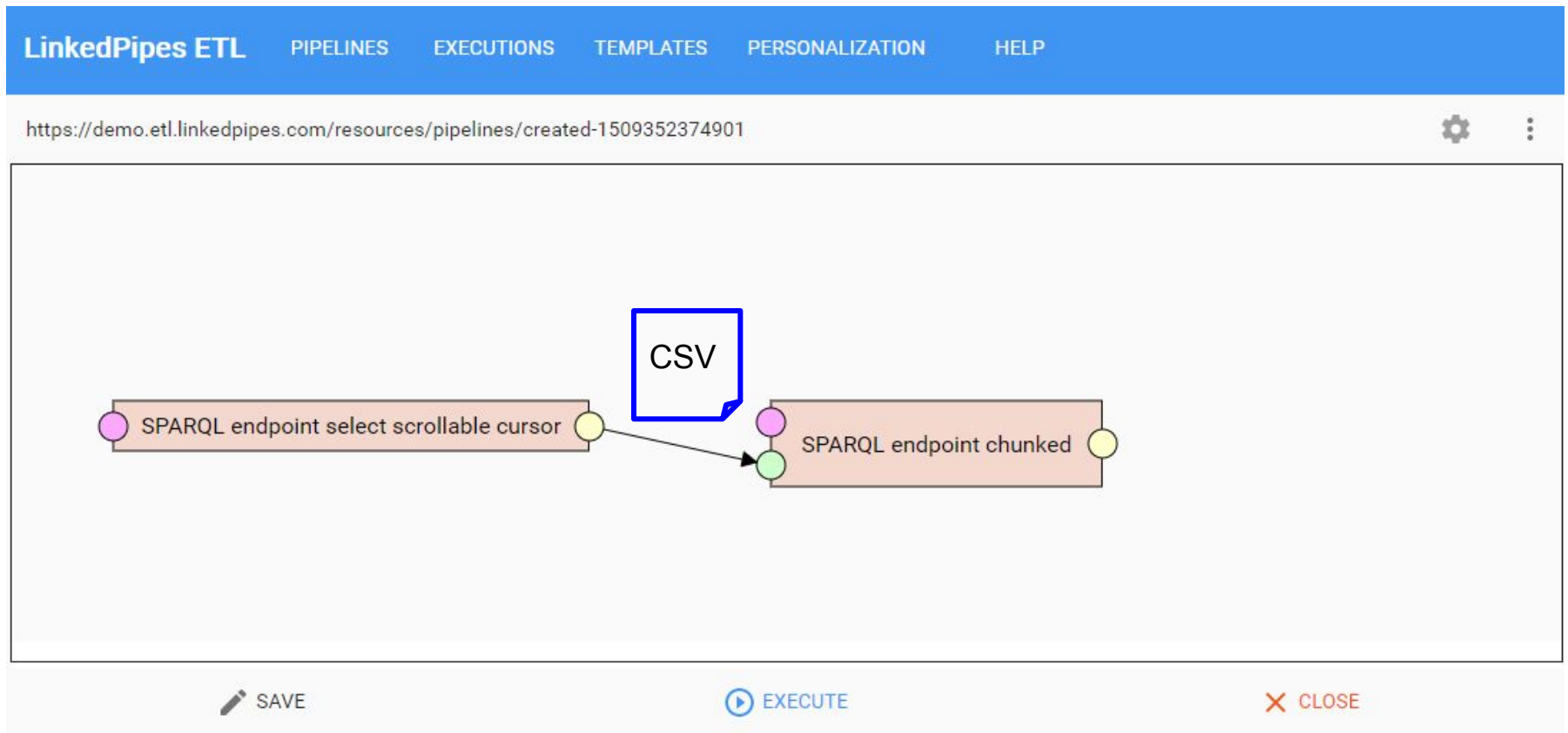
Virtuoso 22023 Error SR353: Sorted TOP clause specifies more then 41000 rows to sort.
Only 40000 are allowed.

Either decrease the offset and/or row count or use a scrollable cursor

LP-ETL and Virtuoso Scrollable Cursor

```
SELECT ?point
WHERE
{
    {
        SELECT ?point
        WHERE {
            [] ruian:adresniBod ?point.
        }
        ORDER BY ASC(?point)
    }
}
LIMIT 100
OFFSET 10200
```


LP-ETL and larger data



LP-ETL and larger data - 1 record

```
CONSTRUCT {  
  my:Point a gml:Point;  
    gml:pos ?o .  
}  
WHERE  
{  
  my:Point a gml:Point;  
    gml:pos ?o .  
}
```

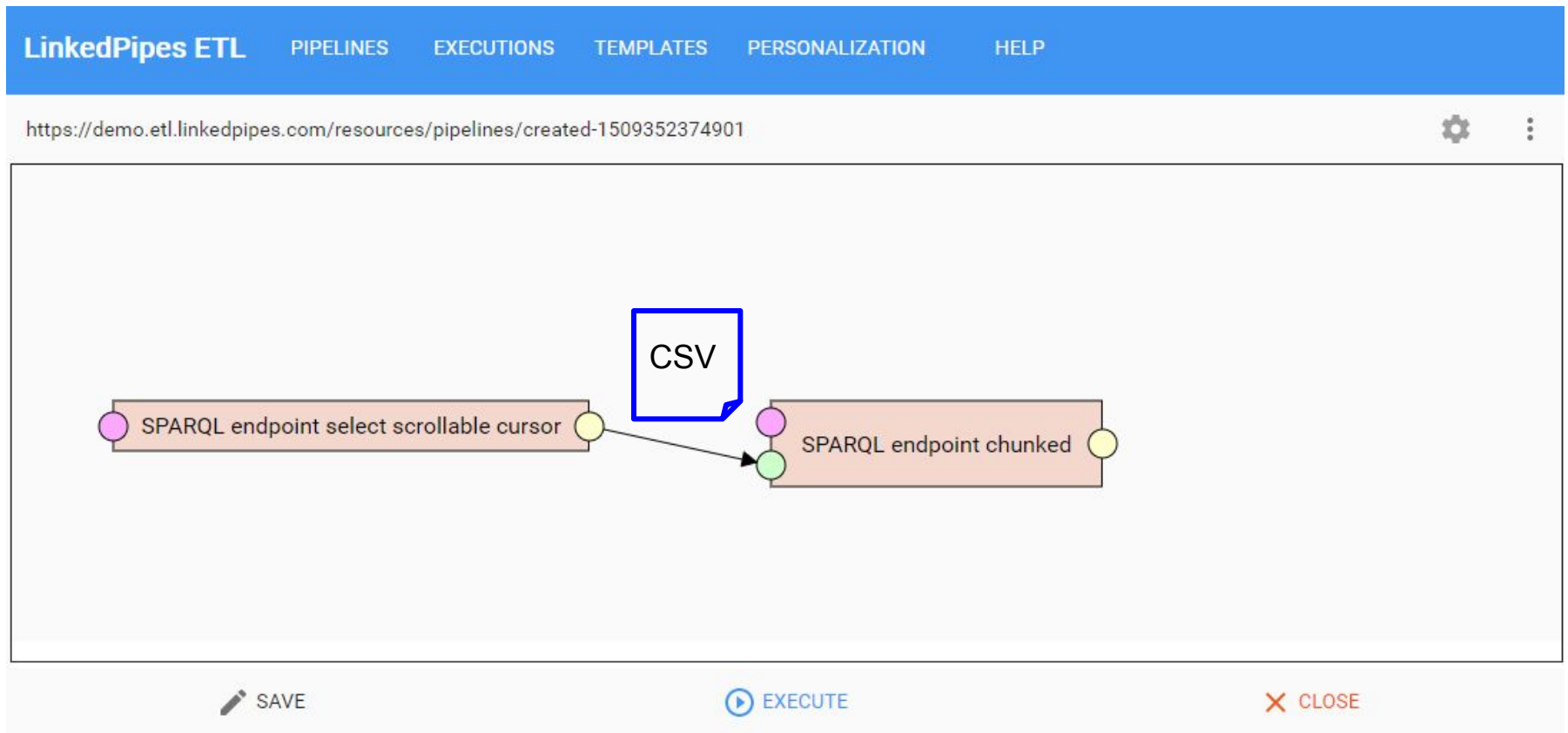
LP-ETL and larger data - 2 records

```
CONSTRUCT {  
  ?point a gml:Point;  
    gml:pos ?o .  
}  
WHERE  
{  
  ?point a gml:Point;  
    gml:pos ?o .  
VALUES ?point { my:Point1 my:Point2 }  
}
```

LP-ETL and larger data - all records

```
CONSTRUCT {  
  ?point a gml:Point;  
    gml:pos ?o .  
}  
WHERE  
{  
  ?point a gml:Point;  
    gml:pos ?o .  
  ${VALUES}  
}
```

LP-ETL and larger data



<https://goo.gl/zfcbhg>

LinkedPipes ETL

Jakub Klímek, Petr Škoda

<https://etl.linkedpipes.com/>