



# The Environmental Data Initiative (EDI)



**Make metadata with the EML assembly line**



EDI is funded by the NSF DEB



# 5 Phases of Publishing Ecological Data

1. Assemble data and metadata
2. Format and QC data tables
3. **Create EML metadata**
4. Submit your data package (data and metadata) to repository
5. Cite your data package



# 5 Phases of Publishing Ecological Data

1. Assemble data and metadata
2. Format and QC data tables
3. **Create EML metadata**
4. Submit your data package (data and metadata) to repository
5. Cite your data package

[www.environmentaldatainitiative.org](http://www.environmentaldatainitiative.org)



# Quality metadata is **VERY** important!

- **For understanding:**
  - What the data are
  - When the data were collected
  - How the data were collected
  - Who collected the data
  - How the data have been modified
- **For assessing fitness for use**
- **Critical to:**
  - Open and reproducible science
  - Synthesis science
  - Extending the life and value of data



# Structured metadata is **VERY** important!

- **If not structured then:**
  - **Can't search for the data in repositories or aggregators**
  - **Can't search content at the dataset level**
  - **The speed of science and knowledge formation is slowed**
  - **Potential information loss**
  - **Can't convert to other metadata standards**

A decorative horizontal line with a teal segment on the left and an orange segment on the right, positioned above the main heading.

# But, making quality EML is challenging!

- **Creating quality EML requires:**
  - **Detailed info about: methods, data entities, personnel, keywords, etc.**
  - **Detailed technical info about data entities**
  - **Understanding of the EML schema (where content can go)**
  - **Understanding EML best practices (where content should go)**
  - **Construction of EML (how to create it)**

A decorative horizontal line with a teal segment on the left and an orange segment on the right, positioned above the main heading.

# But, making quality EML is challenging!

- **Creating quality EML requires:**
  - Detailed info about: methods, data entities, personnel, keywords, etc.
  - Detailed technical info about data entities
  - Understanding of the EML schema (where content can go)
  - Understanding EML best practices (where content should go)
  - Construction of EML (how to create it)



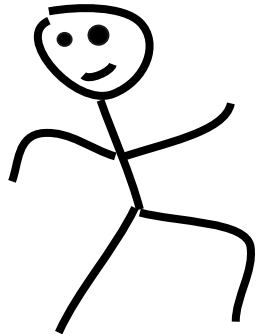
# The EMLassemblyline R code package

- **User:**
  - Supplies core information in a familiar way (tables, text files)
- **EMLassemblyline:**
  - Provides meaningful error messages and guidance
  - Extracts technical info about data entities
  - Builds and validates the EML
  - Embodies EML best practices
  - Simplifies versioning
  - Facilitates automation





# Step 1:

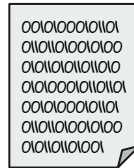
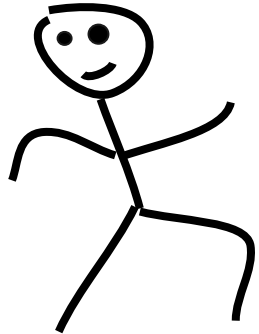


0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000



---

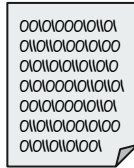
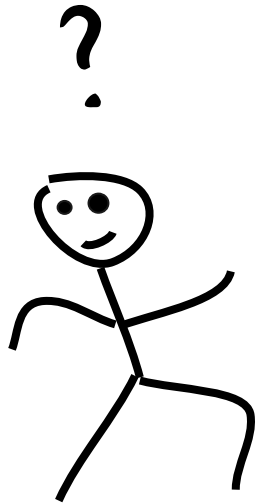
# Step 1: Identify data type(s)





---

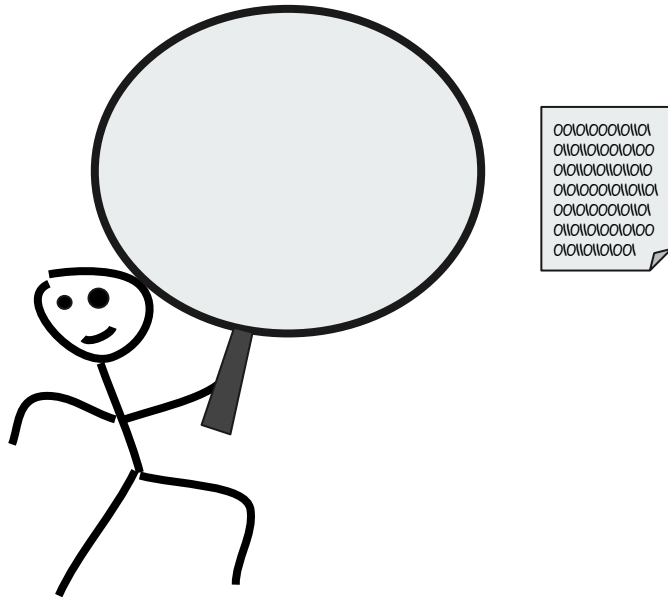
# Step 1: Identify data type(s)





---

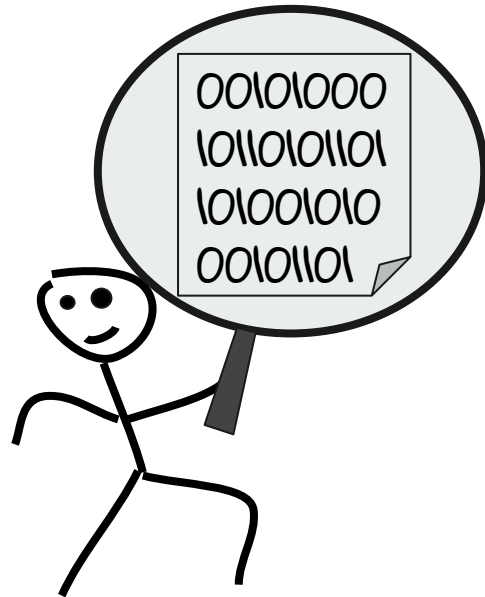
# Step 1: Identify data type(s)





---

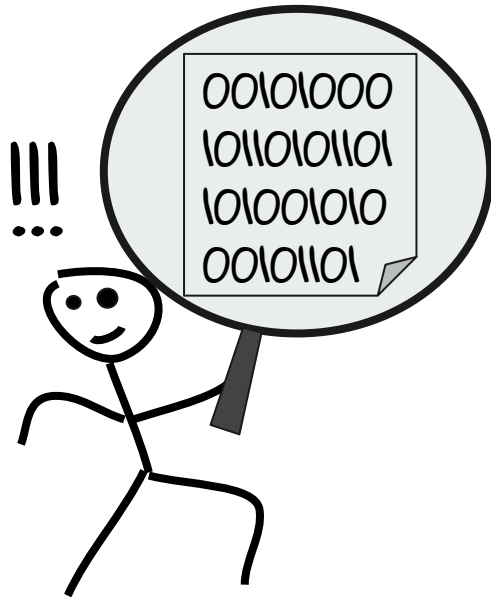
## Step 1: Identify data type(s)





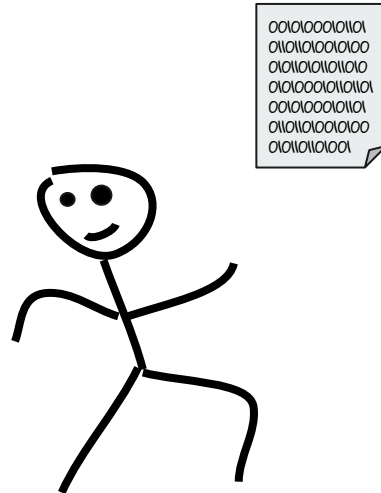
---

## Step 1: Identify data type(s)



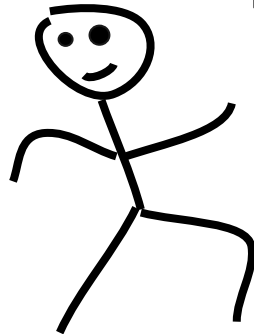


# Step 1: Identify data type(s)





## Step 2:



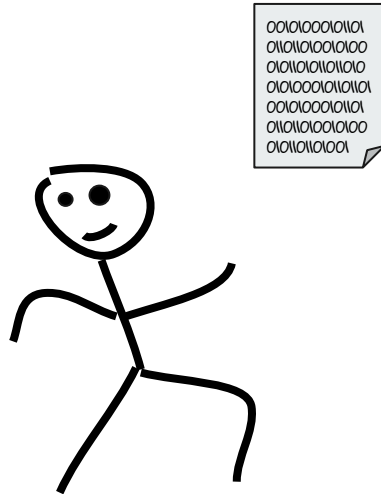
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000





---

## Step 2: Consult instructions

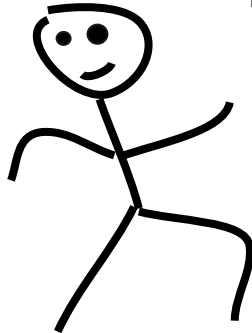


A horizontal bar with a teal-to-orange gradient.

## Step 2: Consult instructions

### Instructions

- data table
- spatial vector
- spatial raster
- other

A rectangular box with a folded bottom-right corner containing binary code.

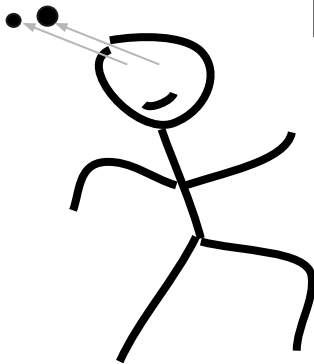
```
0010001011  
01101010100  
010101101010  
01000010111  
0010001011  
01101010100  
01010101
```

---

## Step 2: Consult instructions

### Instructions

- data table
- spatial vector
- spatial raster
- other

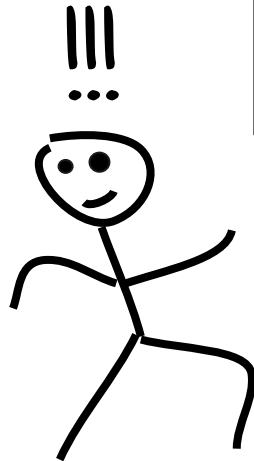


```
0010001011  
0110100100  
0101011010  
0100001011  
0010001011  
0110100100  
010101001
```

## Step 2: Consult instructions

### Instructions

- data table
- spatial vector
- spatial raster
- other

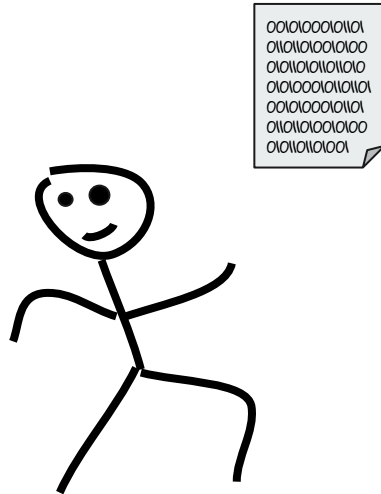


```
0010001011  
0110100100  
0101011010  
0100001011  
0010001011  
0110100100  
010101001
```



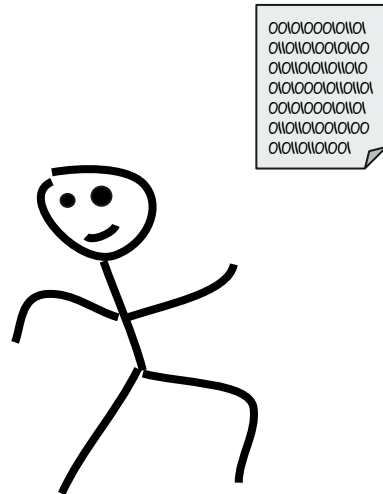
---

## Step 2: Consult instructions





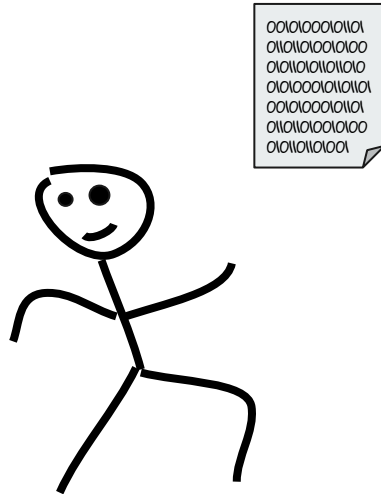
# Step 3:





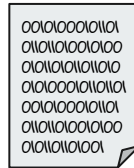
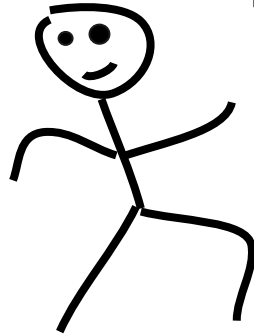
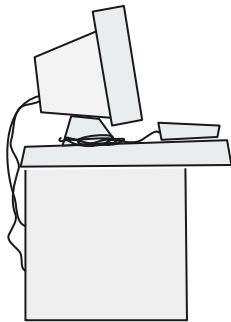
---

## Step 3: Import templates



---

## Step 3: Import templates

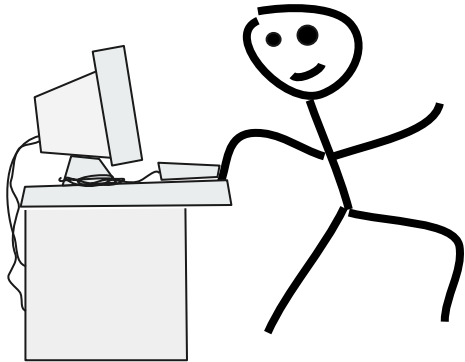






---

## Step 3: Import templates

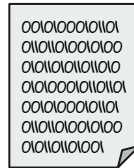
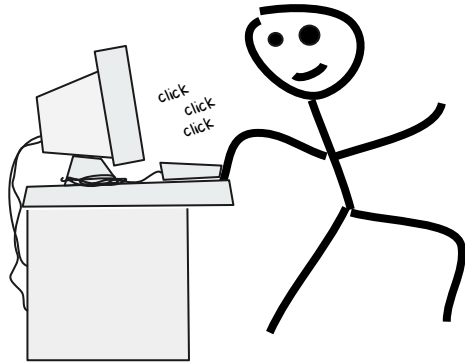


001000101  
01010010100  
01010101010  
01000010101  
0010000101  
01010010100  
0101010101



---

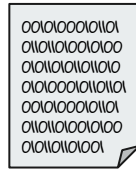
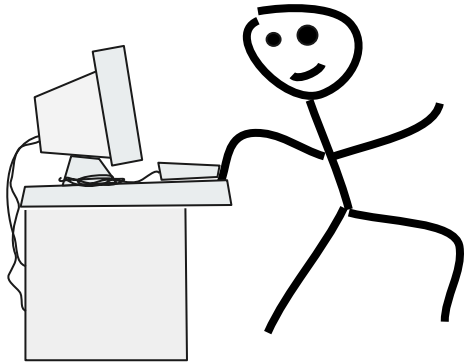
## Step 3: Import templates





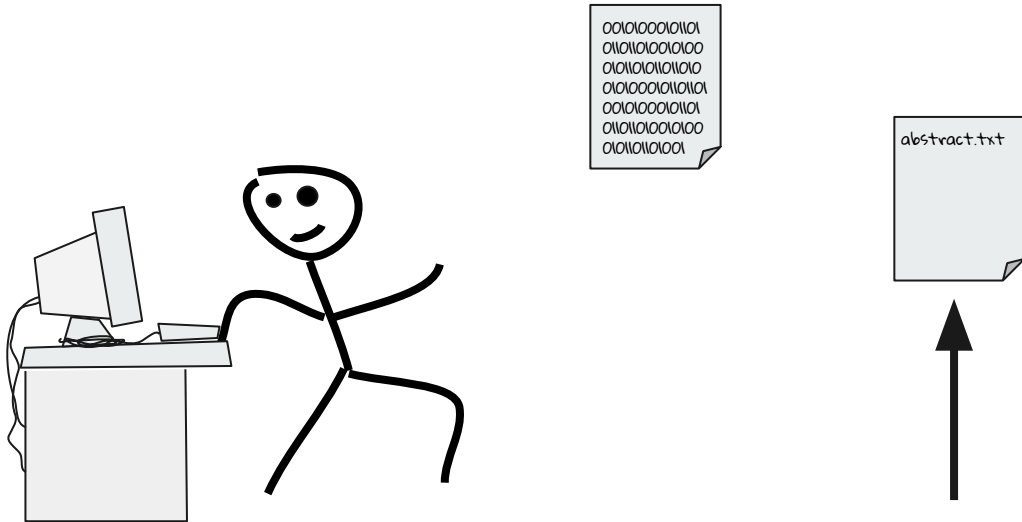
---

## Step 3: Import templates

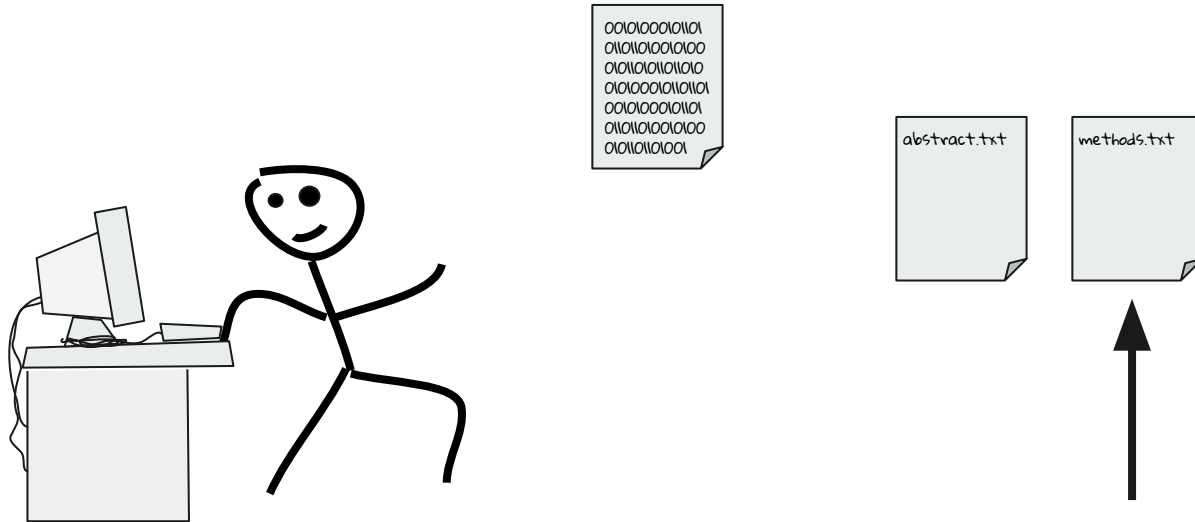


---

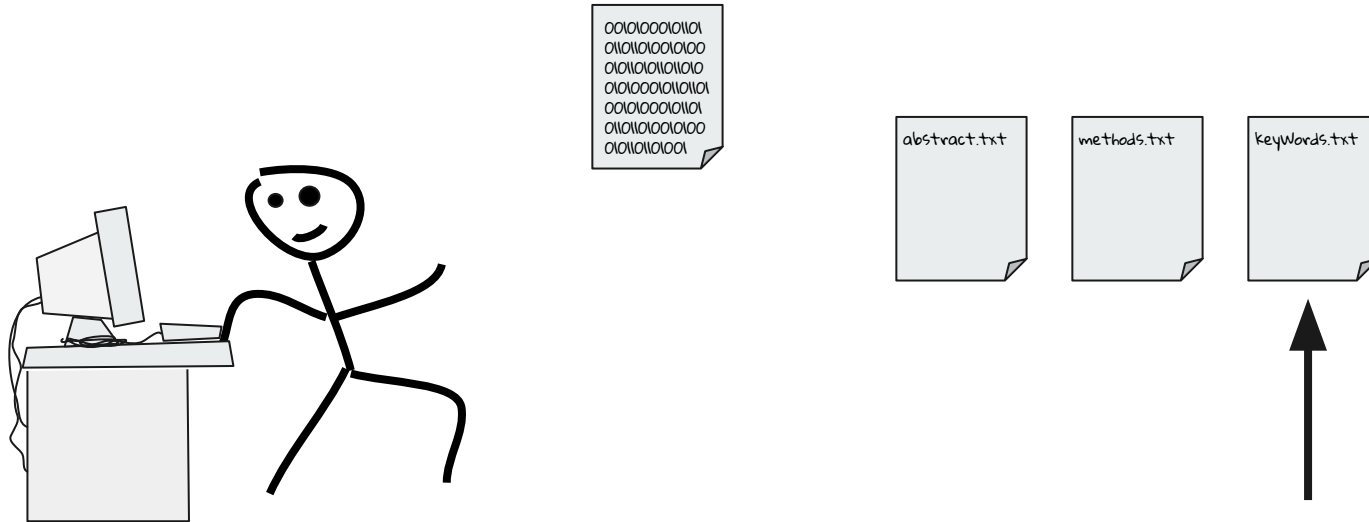
## Step 3: Import templates



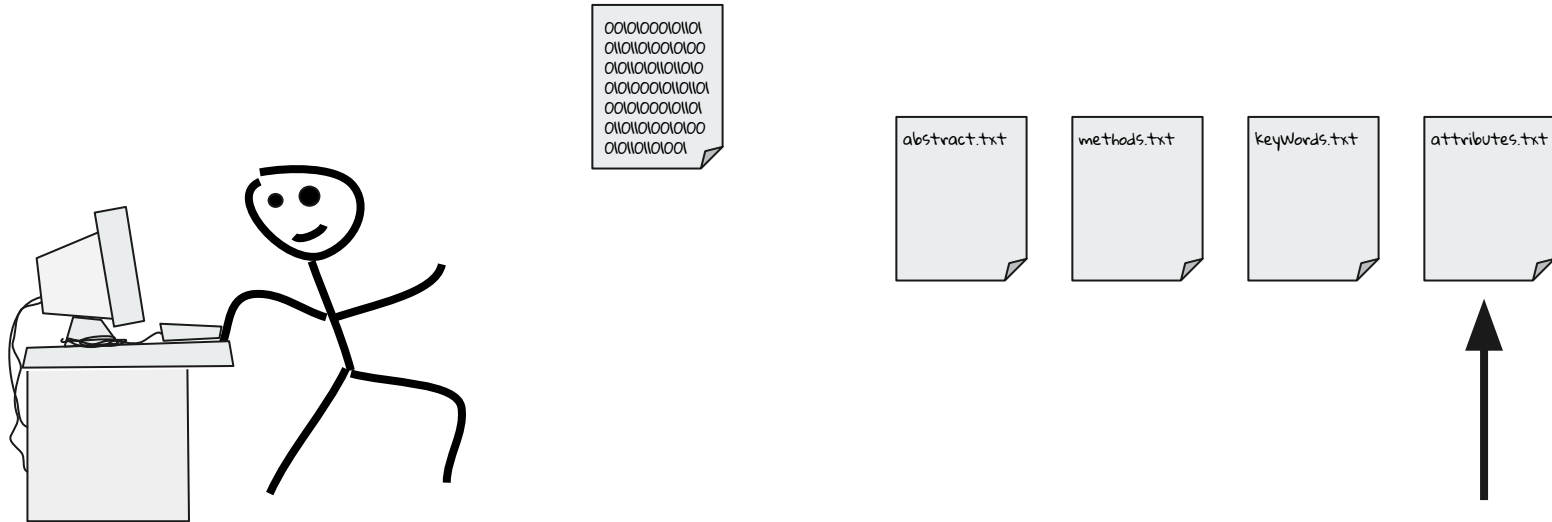
## Step 3: Import templates



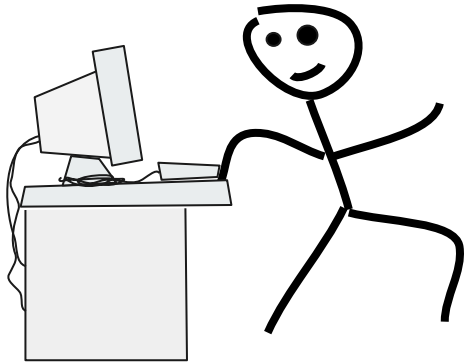
## Step 3: Import templates



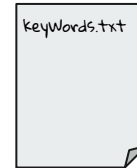
## Step 3: Import templates



## Step 3: Import templates



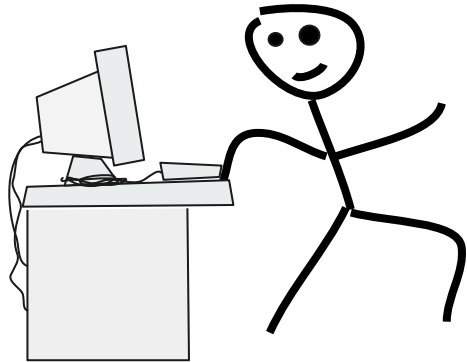
```
0010001011  
0110100100  
01010101010  
0100001011  
0010001011  
0110100100  
01010101
```



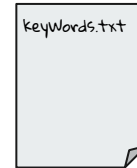


---

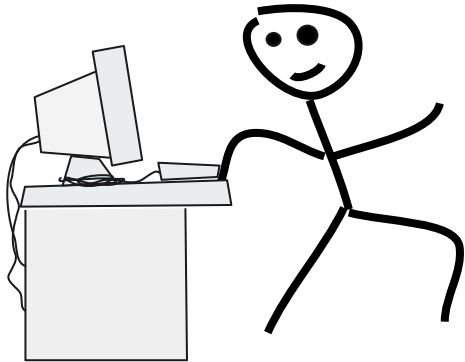
## Step 4:



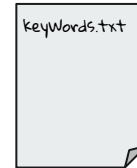
```
0010000101  
0110100100  
01010101010  
01000010101  
0010000101  
0110100100  
01010101
```



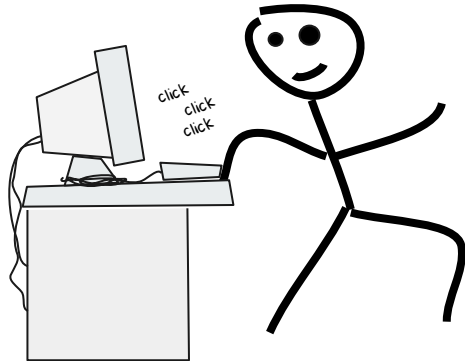
## Step 4: Complete templates



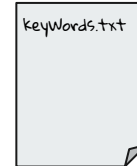
```
0010000101  
01010100100  
01010101010  
01000010101  
0010000101  
01010100100  
0101010101
```



## Step 4: Complete templates

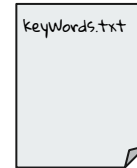
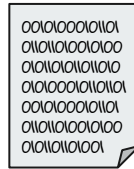
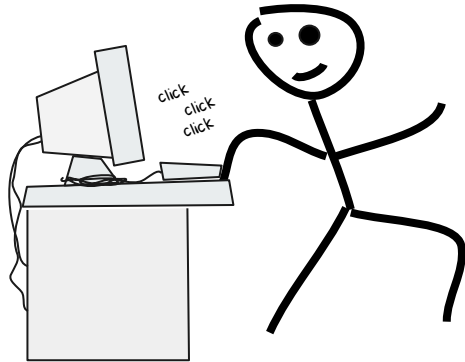


```
0010000101  
01010100100  
01010101010  
01000010101  
0010000101  
01010100100  
010101001
```

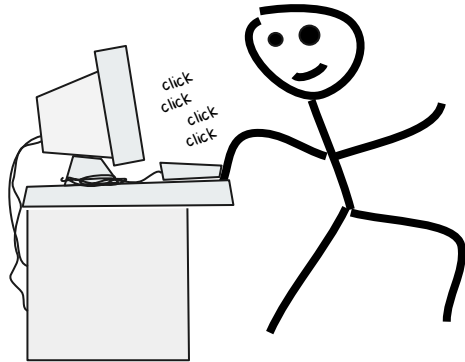




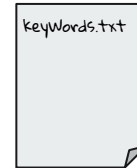
# Step 4: Complete templates



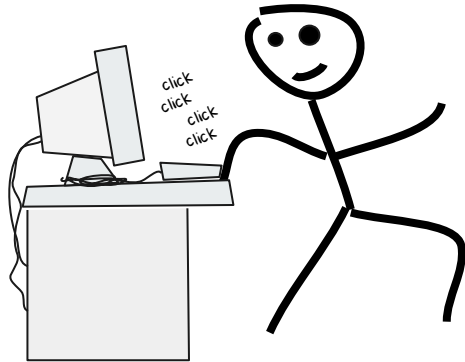
# Step 4: Complete templates



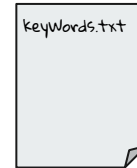
```
0010000101  
0110100100  
01010101010  
01000010101  
0010000101  
0110100100  
01010101
```



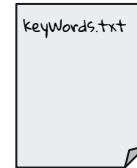
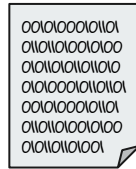
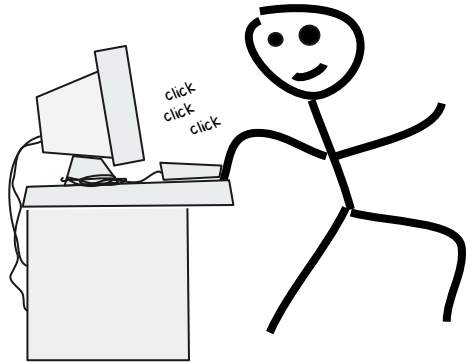
# Step 4: Complete templates



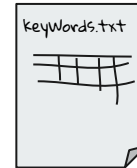
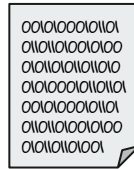
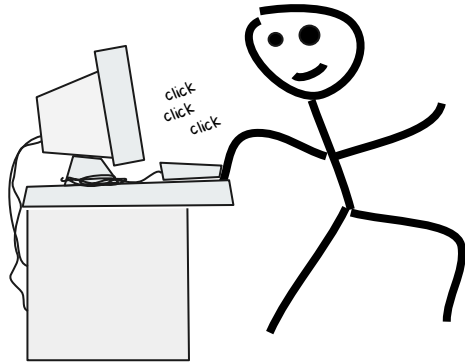
```
0010000101  
0101000100  
01010101010  
01000010101  
0010000101  
0101000100  
01010101
```



# Step 4: Complete templates

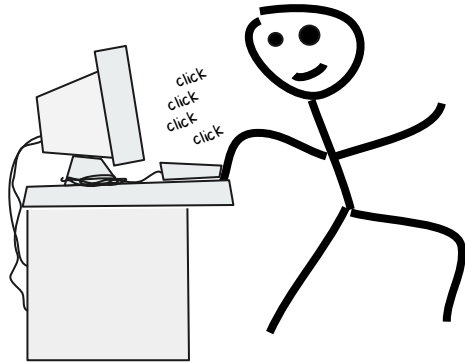


# Step 4: Complete templates

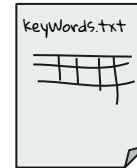




# Step 4: Complete templates

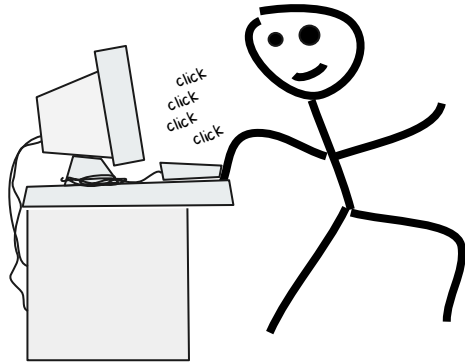


```
0010000101  
0110100100  
0101010101  
0100001010  
0010000101  
0110100100  
01010101
```

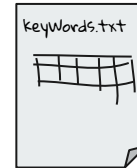




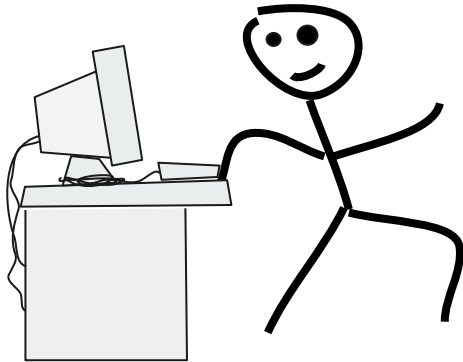
# Step 4: Complete templates



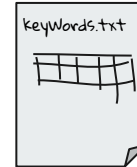
```
0010000101  
0110100100  
0101010101  
0100001010  
0010000101  
0110100100  
01010101
```



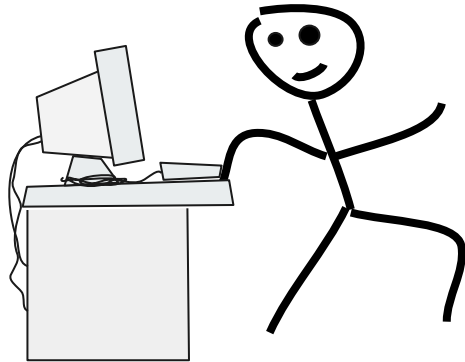
## Step 4: Complete templates



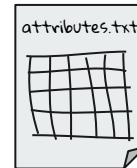
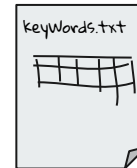
```
0010000101  
01010100100  
01010101010  
01000010101  
0010000101  
01010100100  
01010101
```



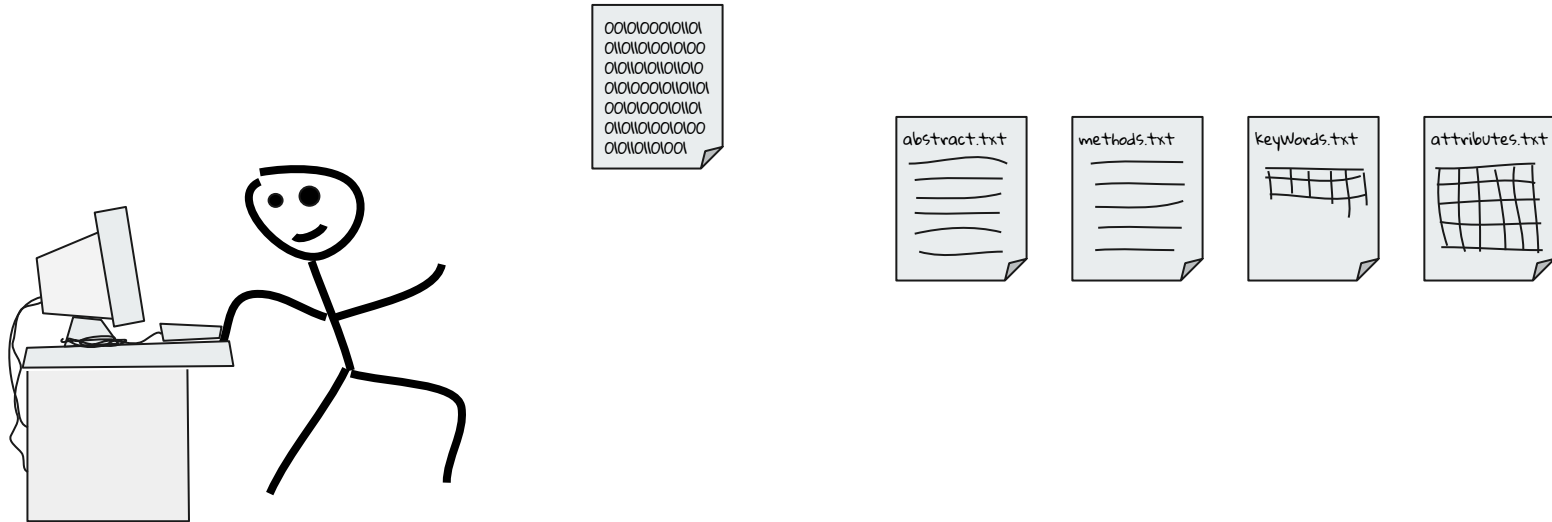
## Step 5:



```
0010001011  
0110100100  
0101011010  
0100010101  
0010001011  
0110100100  
010101001
```



## Step 5: Add custom details



## Step 5: Add custom details

### Instructions

- geographic
- taxonomic
- categorical
- other

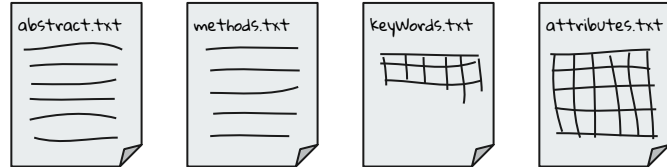
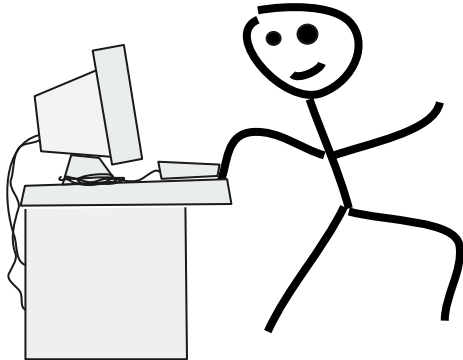
```
0010000101  
0110100100  
01010101010  
01000010101  
0010000101  
0110100100  
01010101
```

abstract.txt

methods.txt

keyWords.txt

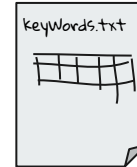
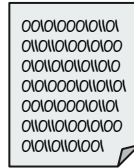
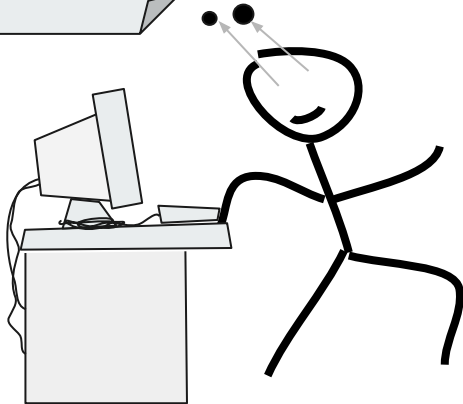
attributes.txt



## Step 5: Add custom details

**Instructions**

- geographic
- taxonomic
- categorical
- other



## Step 5: Add custom details

### Instructions

- geographic
- taxonomic
- categorical
- other

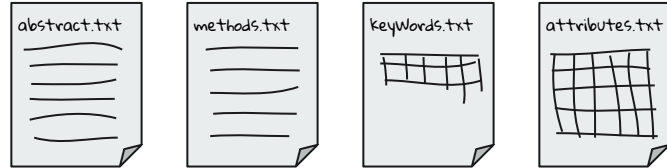
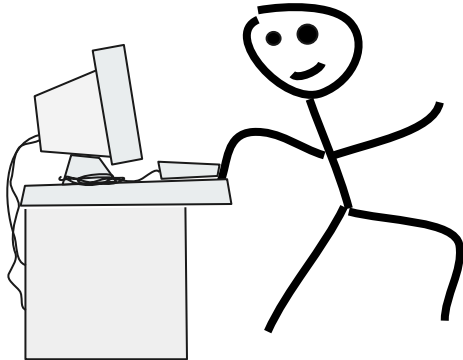
```
0010000101  
0110100100  
01010101010  
01000010101  
0010000101  
0110100100  
01010101
```

abstract.txt

methods.txt

keyWords.txt

attributes.txt

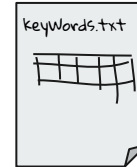
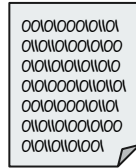
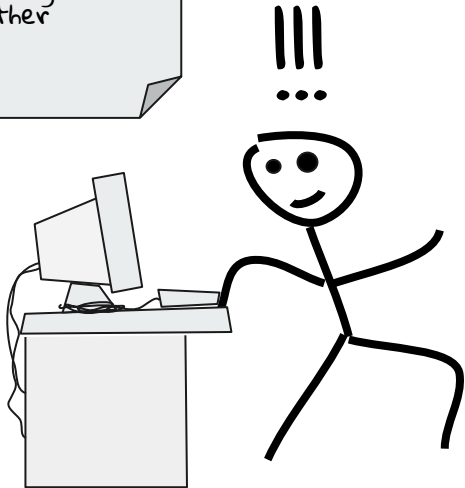




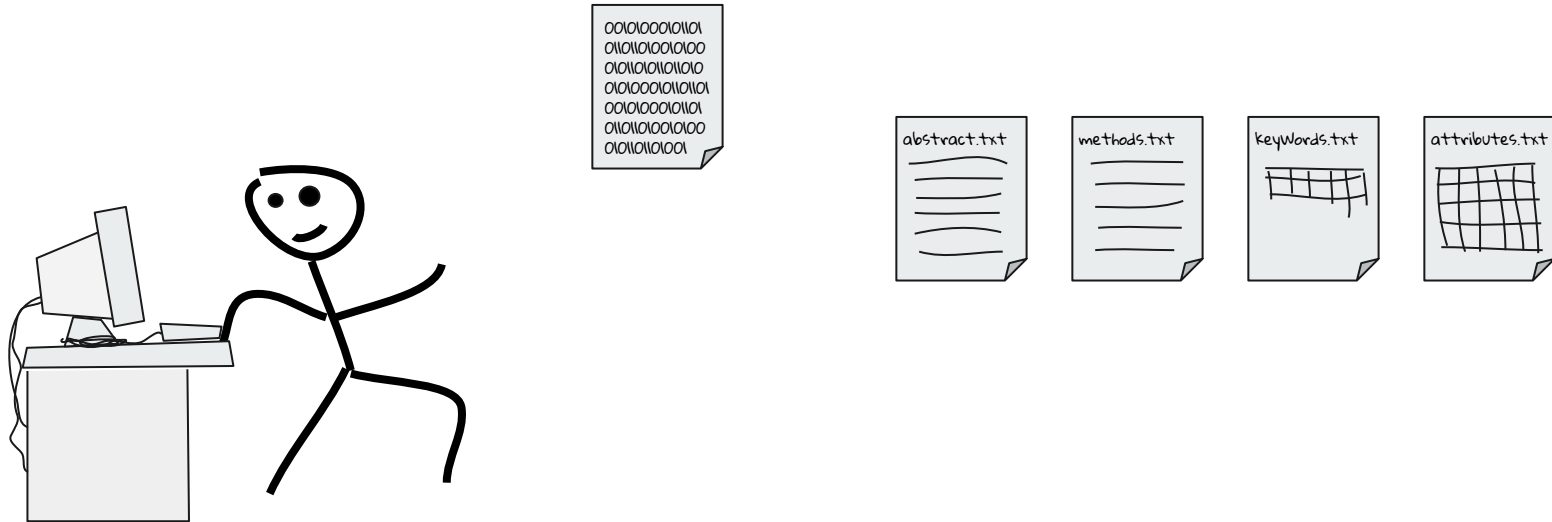
## Step 5: Add custom details

**Instructions**

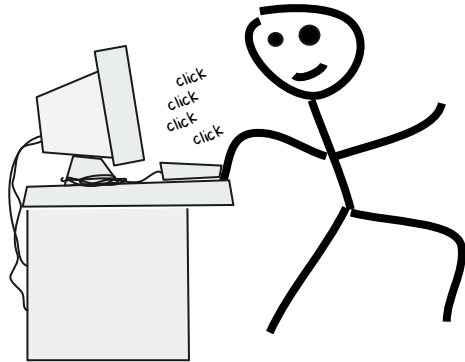
- geographic
- taxonomic
- categorical
- other



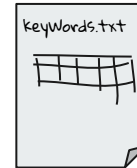
## Step 5: Add custom details



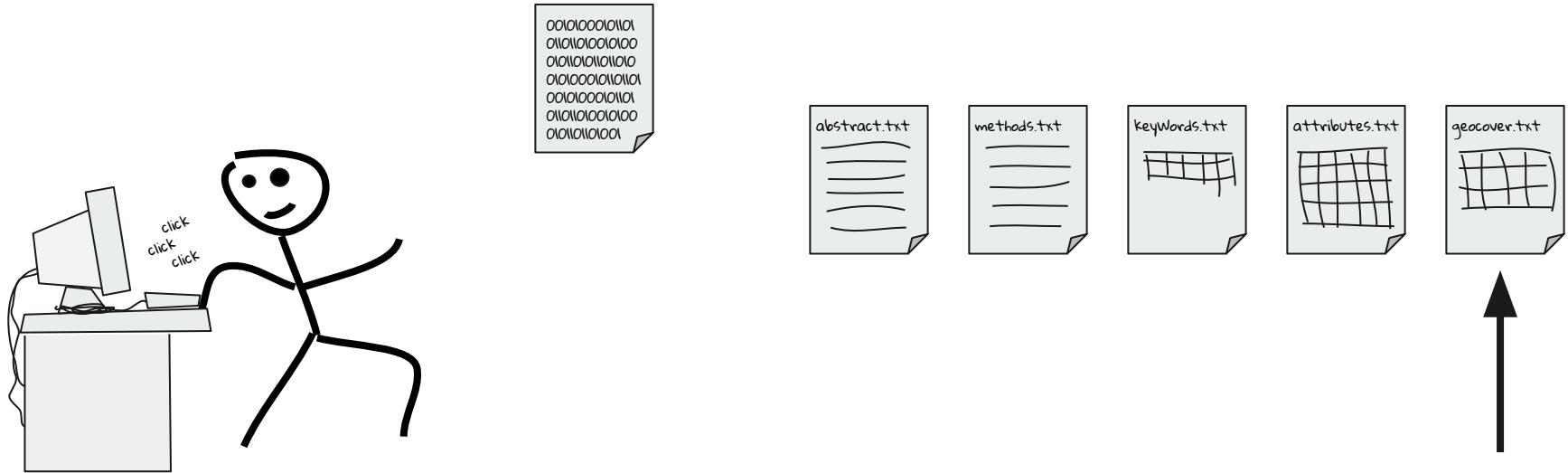
## Step 5: Add custom details



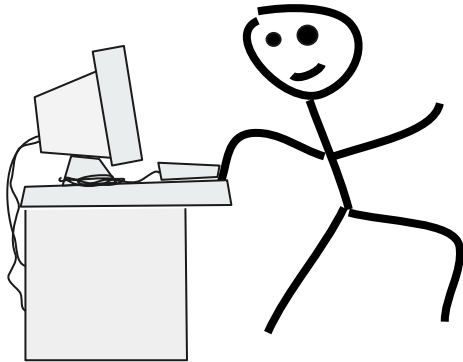
```
0010000101  
0110100100  
0101010101  
0100001010  
0010000101  
0110100100  
01010101
```



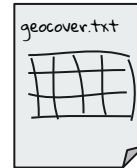
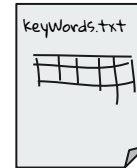
## Step 5: Add custom details



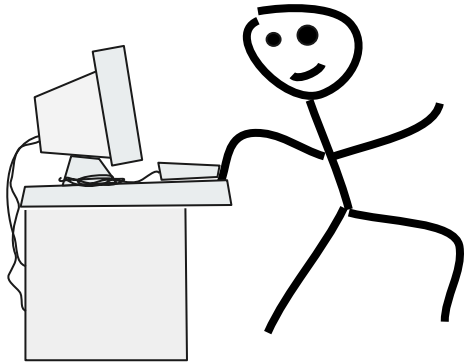
## Step 5: Add custom details



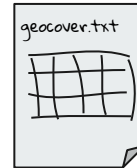
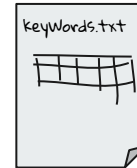
```
0010000101  
0110100100  
0101010101  
0100001011  
0010000101  
0110100100  
01010101
```



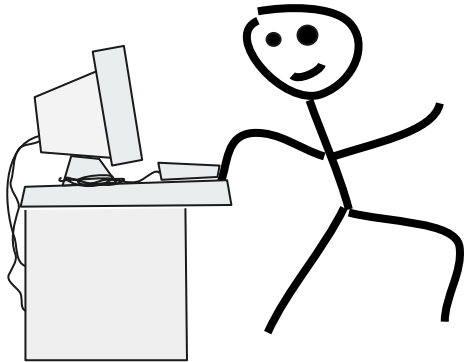
# Step 6:



```
0010000101  
0110100100  
0101010101  
0100001011  
0010000101  
0110100100  
01010101
```



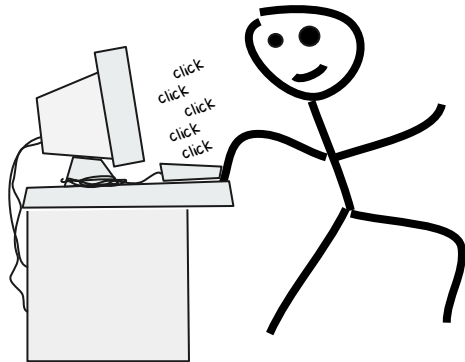
## Step 6: Translate to EML



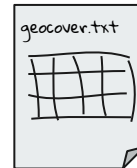
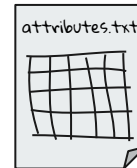
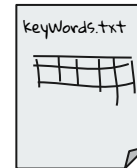
```
0010000101  
0101000100  
0101010100  
0100001011  
0010000101  
0101000100  
01010101
```



## Step 6: Translate to EML

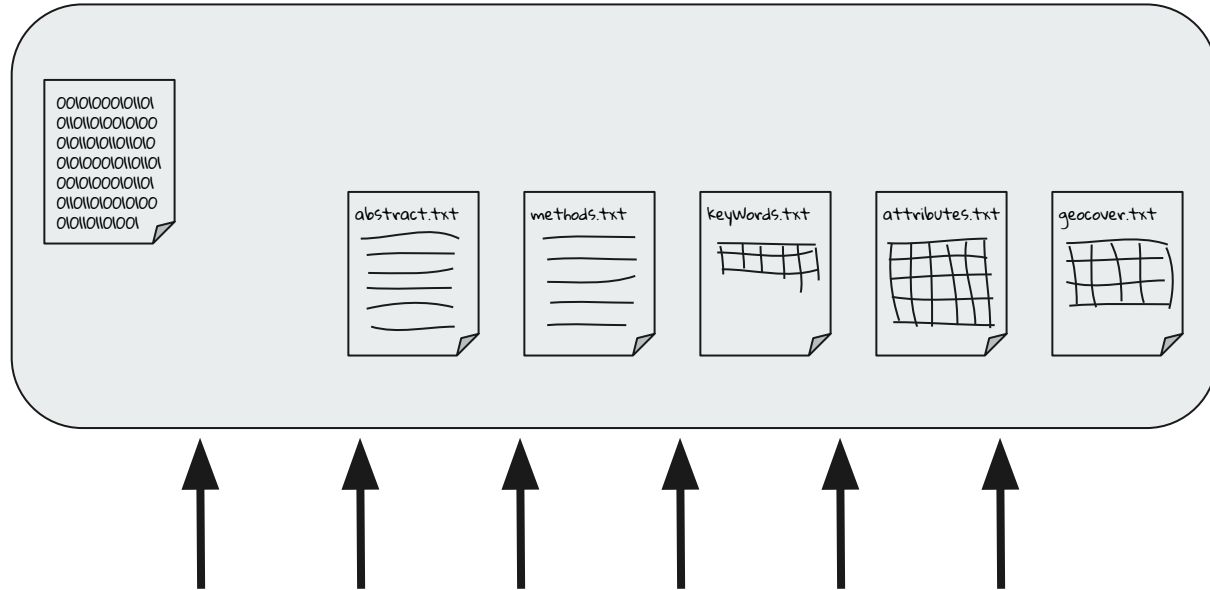
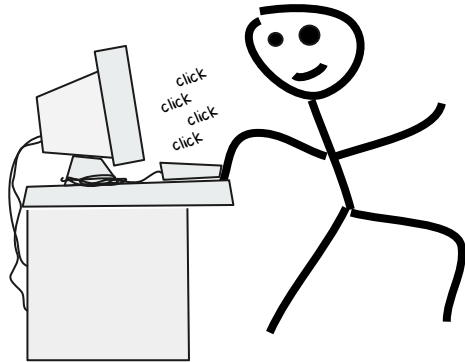


```
001010001011  
011010100100  
010101010101  
01000010101  
00101000101  
011010100100  
0101010101
```

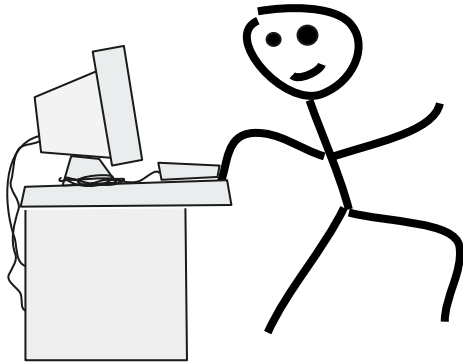




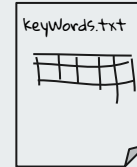
## Step 6: Translate to EML



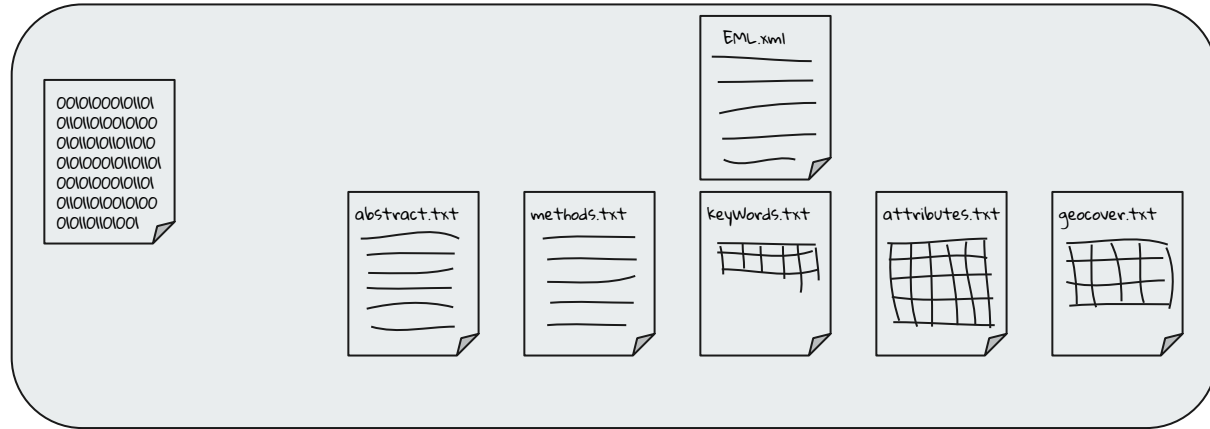
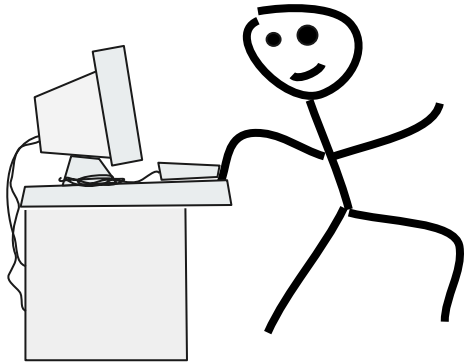
## Step 6: Translate to EML



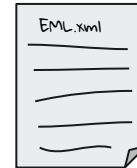
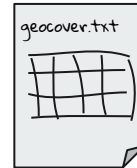
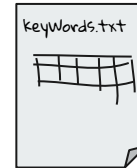
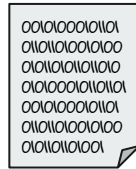
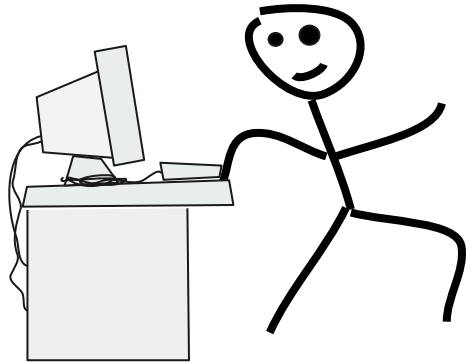
001000101  
0110100100  
01010101010  
01000010101  
0110100100  
01010101



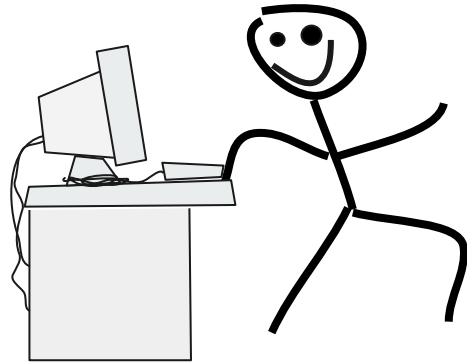
## Step 6: Translate to EML



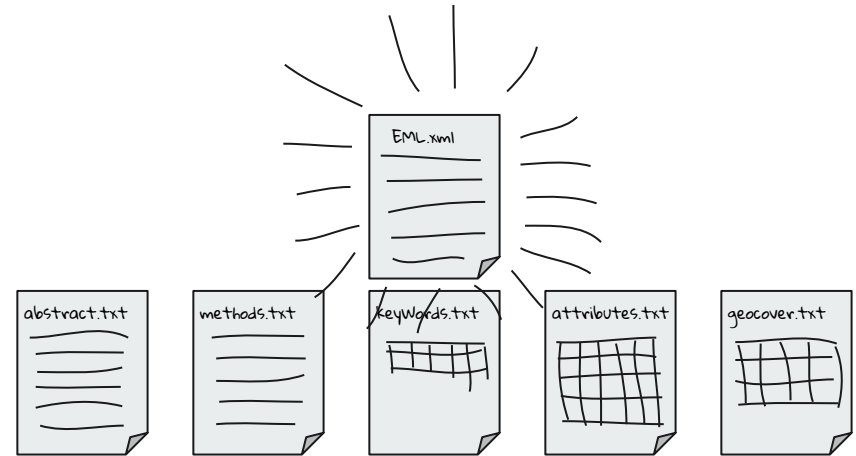
## Step 6: Translate to EML



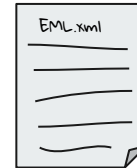
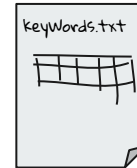
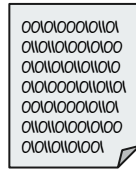
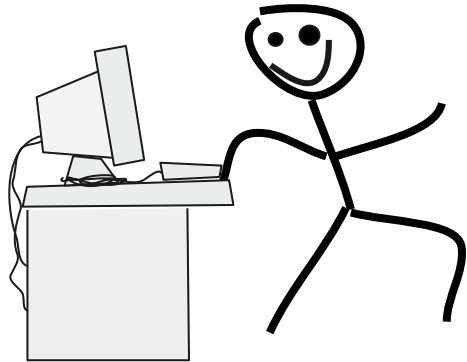
## Step 6: Translate to EML



0010000101  
0101000100  
01010101010  
01000010101  
0101000101  
0101000100  
010101001



## Step 6: Translate to EML





**Live demo!**



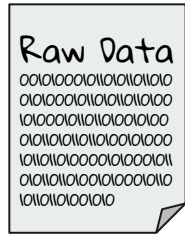
**Workflow:**



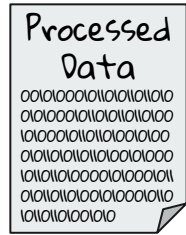


# Workflow: One-off publication

# Workflow: One-off publication

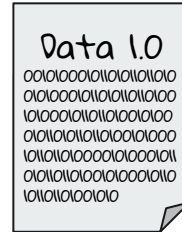


cleaning

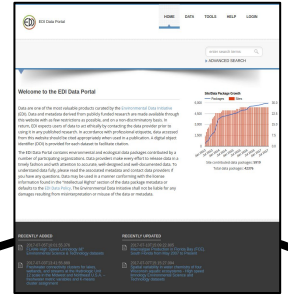
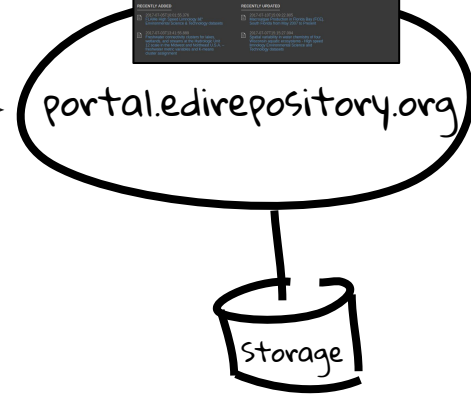


EMLassembleline

edit templates



Upload



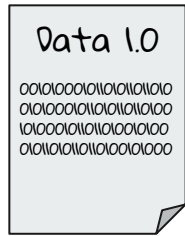


## **Workflow:**

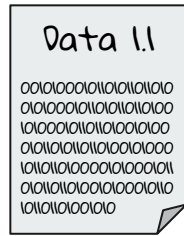


## **Workflow: Revision (same data format)**

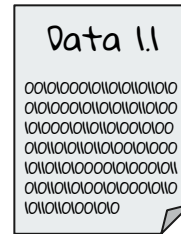
# Workflow: Revision (same data format)



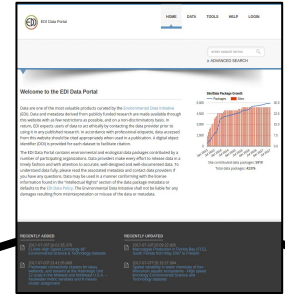
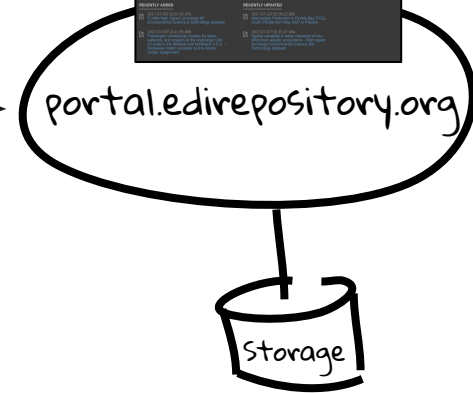
append new  
data



EMLassembleline



Upload



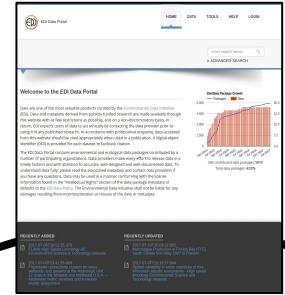
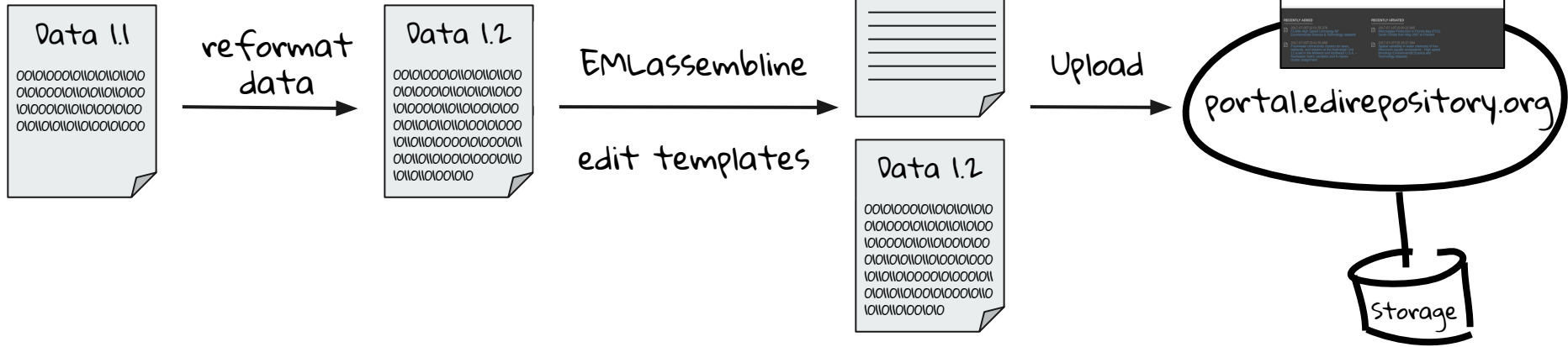


**Workflow:**



# **Workflow: Revision (new data format)**

# Workflow: Revision (new data format)





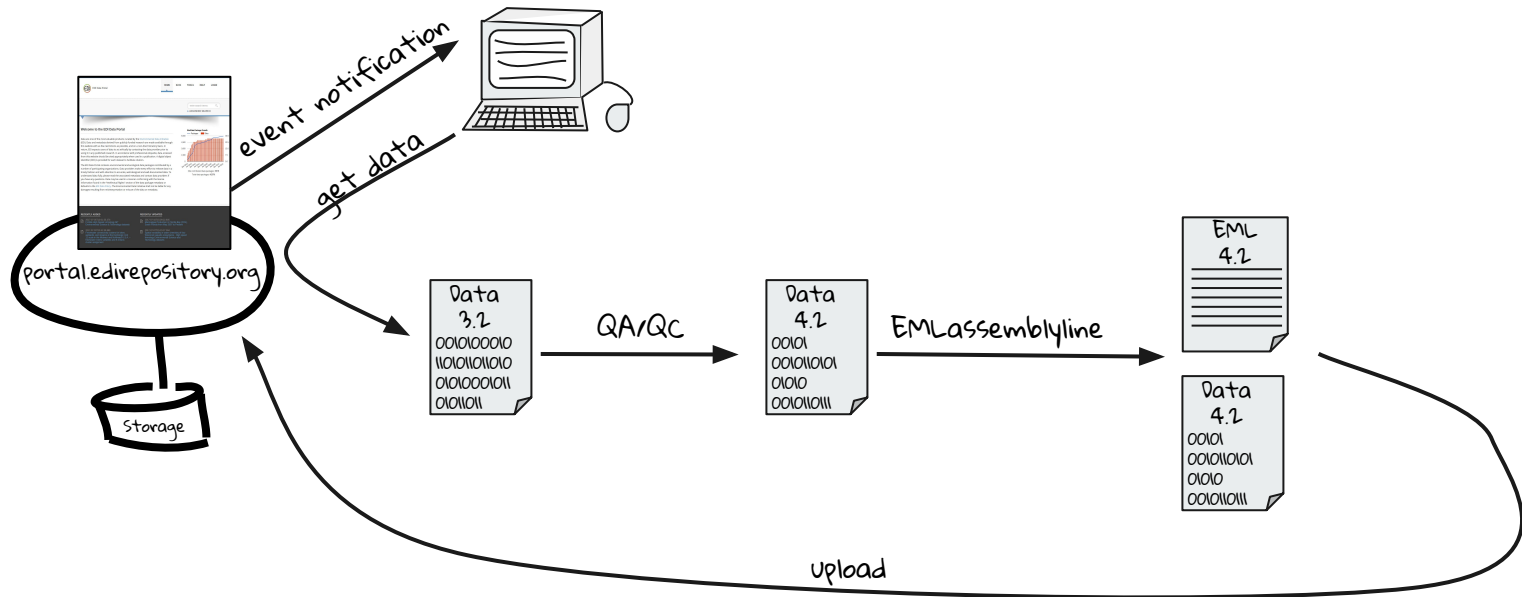


## **Workflow:**



# **Workflow: Automation (derived data)**

# Workflow: Automation (derived data)





# Road map

- **Where is the project going?**
  - **High-level functions**
  - **Auto-extracted metadata from data entities**
  - **Support for more data types**
- **How to get involved?**
  - **<https://github.com/EDlorg/EMLassemblyline>**
  - **[colin.smith@wisc.edu](mailto:colin.smith@wisc.edu)**



# EDI Resources

- **EDI website on “5 phases of data publishing”**
  - [environmentaldatainitiative.org/resources/assemble-data-and-metadata](https://environmentaldatainitiative.org/resources/assemble-data-and-metadata)
- **Contact EDI’s data curation team**
  - [info@environmentaldatainitiative.org](mailto:info@environmentaldatainitiative.org)
- **Data Portal**
  - [portal.edirepository.org/nis/home.jsp](https://portal.edirepository.org/nis/home.jsp)
- **GitHub**
  - [github.com/EDInc](https://github.com/EDInc)
- **Twitter**
  - [@EDInc](https://twitter.com/EDInc)
- **Slack**
  - [edi-got-data.slack.com](https://edi-got-data.slack.com)