

# Data migration to Plone 5.2 and Volto

Rodrigo Ferreira de Souza

October, 2019

# Where we are

1 Knoledgements

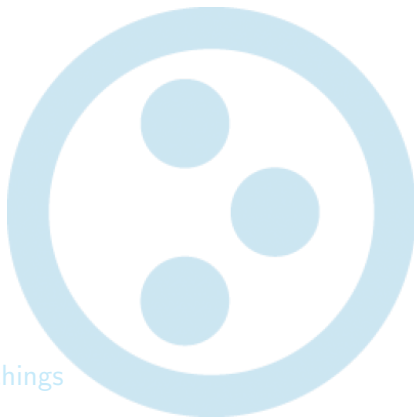
2 Use cases

3 The challenge

4 Our way

5 Live coding

6 Details on how we did things



# Options to Migrate

- Plone 4.3 → Plone 5+
  - Collective Transmogrifier



# Options to Migrate

- Plone 4.3 → Plone 5+
  - [Collective Transmogrifier](#)
- Plone 5.1 → Plone 5.2+
  - [Migrate a ZODB from Python 2.7 to Python 3](#)

# Options to Migrate

- Plone 4.3 → Plone 5+
  - Collective Transmogrifier
- Plone 5.1 → Plone 5.2+
  - Migrate a ZODB from Python 2.7 to Python 3
  - Collective Transmogrifier

# Options to Migrate

- Plone 4.3 → Plone 5+
  - Collective Transmogrifier
- Plone 5.1 → Plone 5.2+
  - Migrate a ZODB from Python 2.7 to Python 3
  - Collective Transmogrifier

## Options to Migrate

Why we use Transmogrifier?

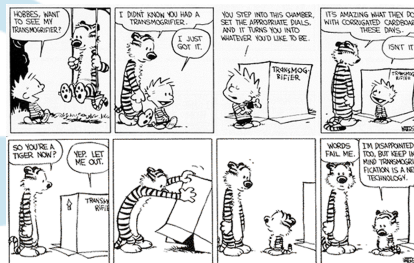


Figure: A transmogrifier is fictional device used for transforming one object into another object. The term was coined by Bill Waterson of Calvin and Hobbes fame.

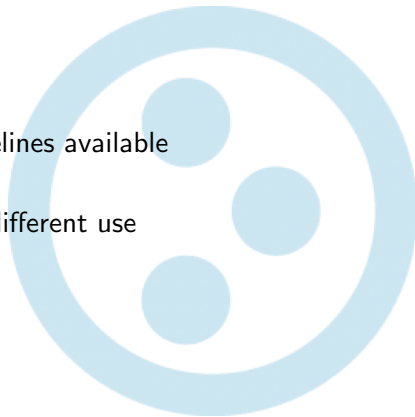
# Why we use Transmogrifier?

- Have many generic Pipelines available for common cases



# Why we use Transmogrifier?

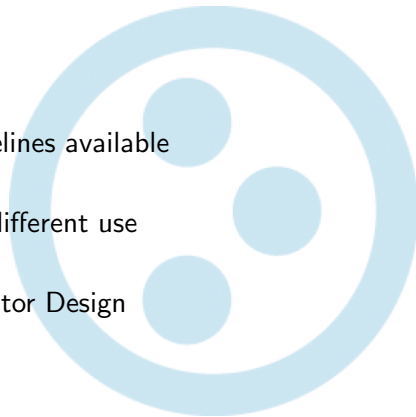
- Have many generic Pipelines available for common cases
- Flexibility to deal with different use cases





# Why we use Transmogrifier?

- Have many generic Pipelines available for common cases
- Flexibility to deal with different use cases
- Brilliant way to use Iterator Design Pattern!



## Why we use Transmogrifier?

- Have many generic Pipelines available for common cases
- Flexibility to deal with different use cases
- Brilliant way to use Iterator Design Pattern!

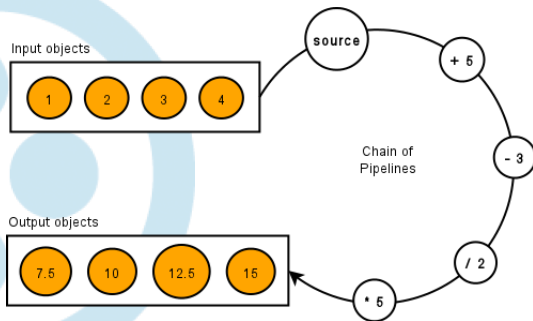


Figure: Transmogrify Diagram

## Why we use Transmogrifier?

- Have many generic Pipelines available for common cases
- Flexibility to deal with different use cases
- Brilliant way to use Iterator Design Pattern!



Figure: Modern Times – Production line

Knoledgements

Use cases

The challenge

Our way

Live coding

Details on how we did things

Large University

High-profile government client

One of the largest research institutions in Germany

# Where we are

1 Knoledgements

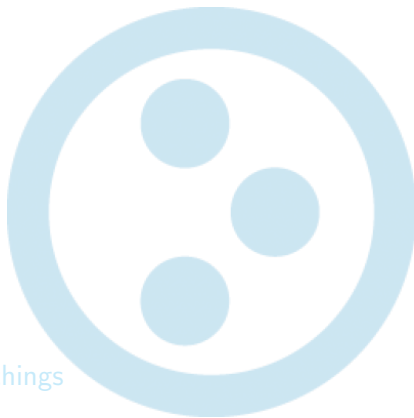
2 Use cases

3 The challenge

4 Our way

5 Live coding

6 Details on how we did things



Knoledgements  
Use cases  
The challenge  
Our way  
Live coding  
Details on how we did things

## Large University

High-profile government client  
One of the largest research institutions in Germany

# Large University

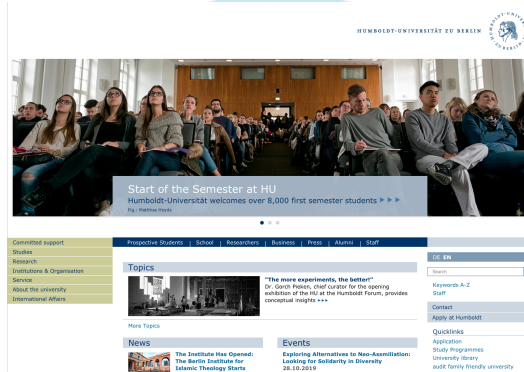


Figure: Humboldt-Universität zu Berlin

Knoledgements  
Use cases  
The challenge  
Our way  
Live coding  
Details on how we did things

Large University  
High-profile government client  
One of the largest research institutions in Germany

## High-profile government client

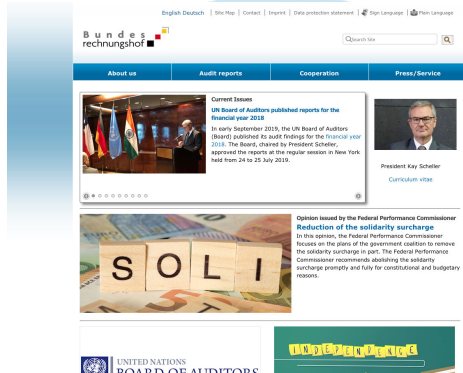


Figure: Bundesrechnungshof

Knoledgements  
Use cases  
The challenge  
Our way  
Live coding  
Details on how we did things

Large University  
High-profile government client  
One of the largest research institutions in Germany

# One of the largest research institutions in Germany

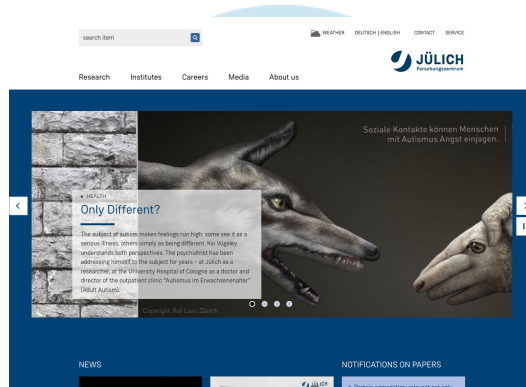


Figure: Forschungszentrum Jülich – Intranet

## Where we are

1 Knoledgements

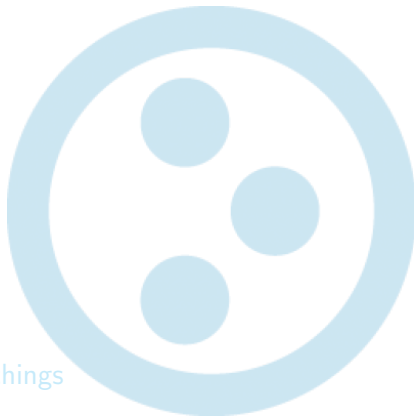
2 Use cases

3 **The challenge**

4 Our way

5 Live coddling

6 Details on how we did things





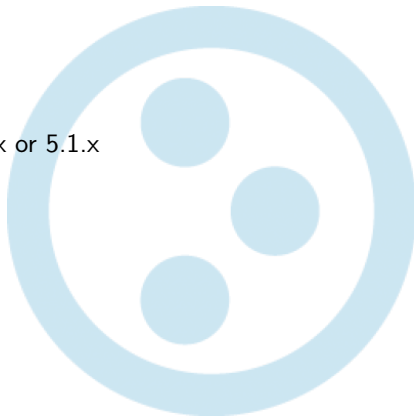
# The challenge

- From:
  - Python 2.x



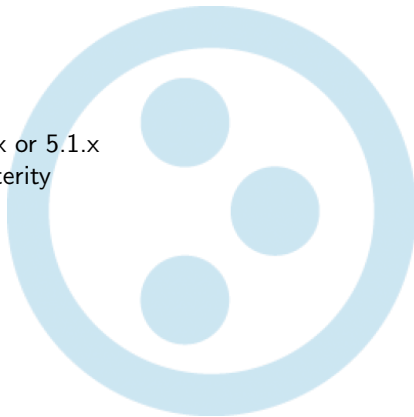
# The challenge

- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x



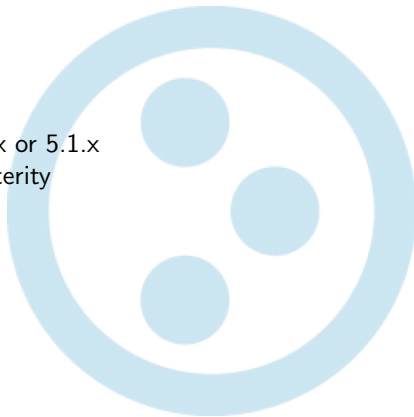
# The challenge

- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x
  - Archetypes or Dexterity



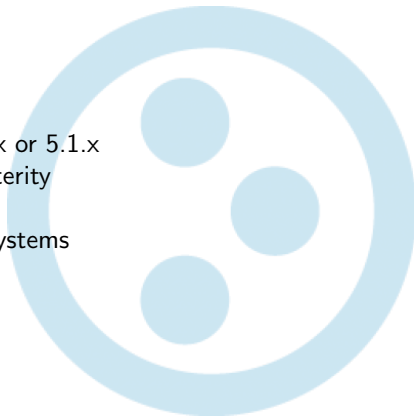
# The challenge

- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x
  - Archetypes or Dexterity
  - Old Products



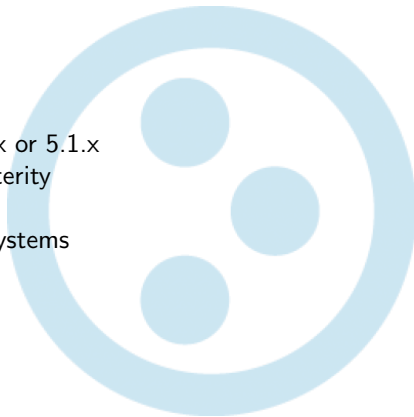
# The challenge

- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x
  - Archetypes or Dexterity
  - Old Products
  - Sometimes other systems



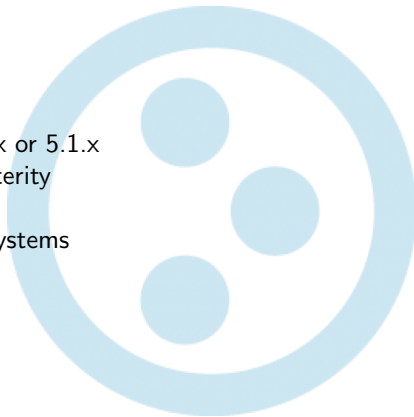
# The challenge

- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x
  - Archetypes or Dexterity
  - Old Products
  - Sometimes other systems
- To:
  - Python 3



# The challenge

- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x
  - Archetypes or Dexterity
  - Old Products
  - Sometimes other systems
- To:
  - Python 3
  - Plone 5.2



- From:
  - Python 2.x
  - Plone 4.3.x or 5.0.x or 5.1.x
  - Archetypes or Dexterity
  - Old Products
  - Sometimes other systems
- To:
  - Python 3
  - Plone 5.2
  - Volto



# Advantages for the clients

- They spare a migration from Plone 5 to Plone 6



## Advantages for the clients

- They spare a migration from Plone 5 to Plone 6
- At least part of it

## Advantages for plone solutions providers

- A way to sell clients the Python 3 upgrade

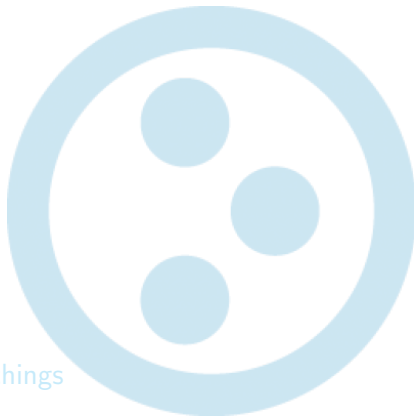


## Advantages for plone solutions providers

- A way to sell clients the Python 3 upgrade
- Which is costly but does not gain the client anything in terms of functionality

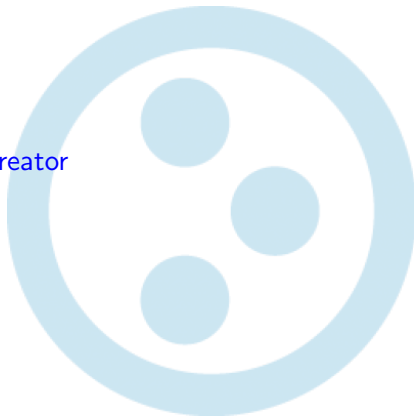
# Where we are

- 1 Knoledgements
- 2 Use cases
- 3 The challenge
- 4 Our way**
- 5 Live coding
- 6 Details on how we did things



# Packages

- Kitconcept Content Creator



# Packages

- [Kitconcept Content Creator](#)
- [Kitconcept Migrator](#) (lives in Migration Package)



# Packages

- [Kitconcept Content Creator](#)
- [Kitconcept Migrator](#) (lives in Migration Package)
- [Migration Plone 5](#)





# Commander Utility

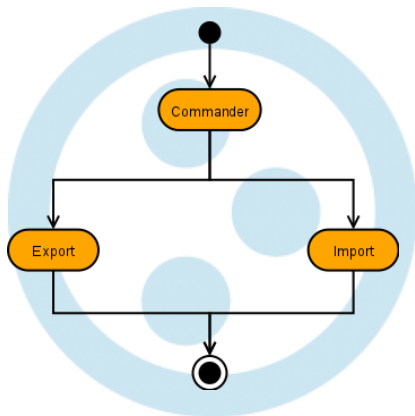


Figure: Commander Utility

Knoledgements  
Use cases  
The challenge  
**Our way**  
Live coddling  
Details on how we did things

Packages  
Commander Utility  
**Jenkins**  
Migration Server

# Jenkins

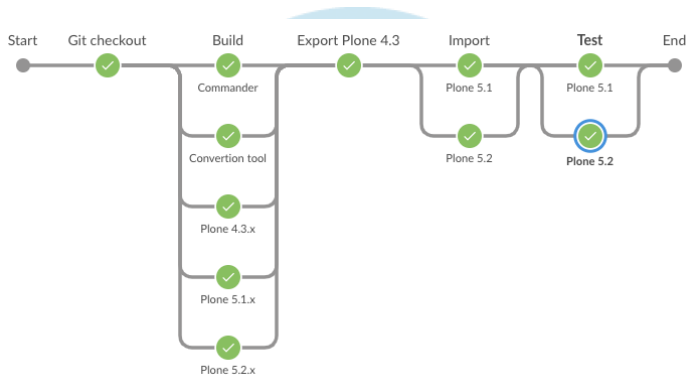


Figure: Jenkins

# Migration Server

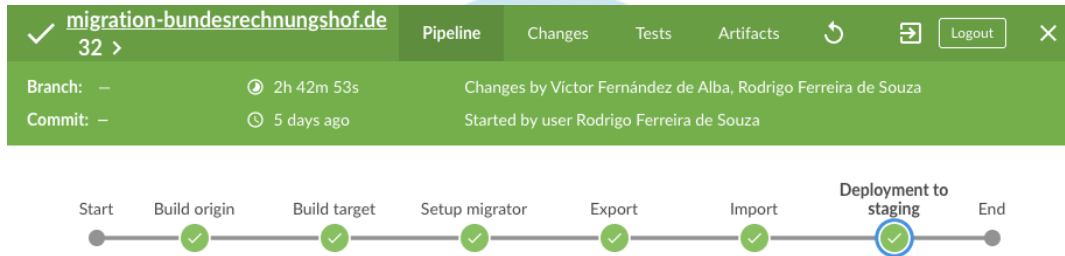
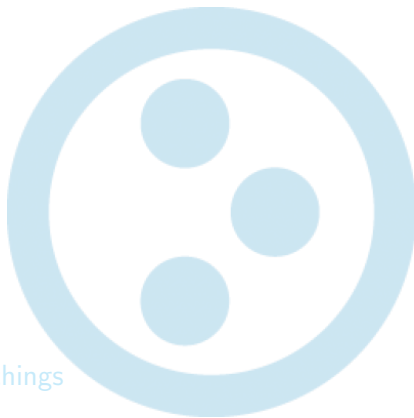


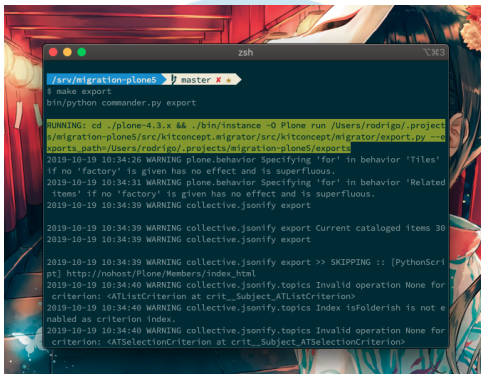
Figure: Migration Server

# Where we are

- 1 Knoledgements
- 2 Use cases
- 3 The challenge
- 4 Our way
- 5 Live coding**
- 6 Details on how we did things



## Our general approach



```
zsh
/srv/migration-plone5 master ✖
$ make export
bin/python commander.py export

RUNNING: cd ./plone-4.3.x && ./bin/instance -O Plone run /Users/rodrigo/.project
s/migration-plone5/src/kitconcept.migrator/src/kitconcept/migrator/export.py --c
xports_path=/Users/rodrigo/.projects/migration-plone5/exports

2019-10-19 10:34:26 WARNING plone.behavior Specifying 'for' in behavior 'Tiles'
if no 'factory' is given has no effect and is superfluous.
2019-10-19 10:34:31 WARNING plone.behavior Specifying 'for' in behavior 'Related
items' if no 'factory' is given has no effect and is superfluous.
2019-10-19 10:34:39 WARNING collective.jsonify export

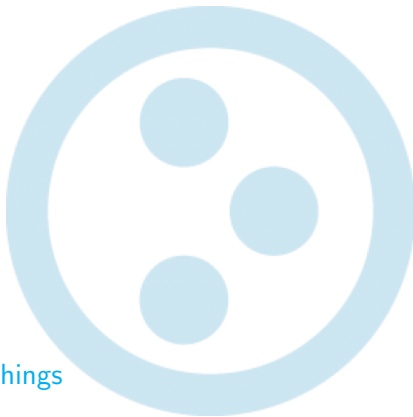
2019-10-19 10:34:39 WARNING collective.jsonify export Current cataloged items 30
2019-10-19 10:34:39 WARNING collective.jsonify export

2019-10-19 10:34:39 WARNING collective.jsonify export >> SKIPPING :: [PythonScri
pt] http://nohost/Plone/Members/index.html
2019-10-19 10:34:40 WARNING collective.jsonify.topics Invalid operation None for
criterion: <ATListCriterion at crit__Subject_ATListCriterion>
2019-10-19 10:34:40 WARNING collective.jsonify.topics Index isFolderish is not e
nabled as criterion index.
2019-10-19 10:34:40 WARNING collective.jsonify.topics Invalid operation None for
criterion: <ATSelectionCriterion at crit__Subject_ATSelectionCriterion>
```

Figure: Running migrator by hand

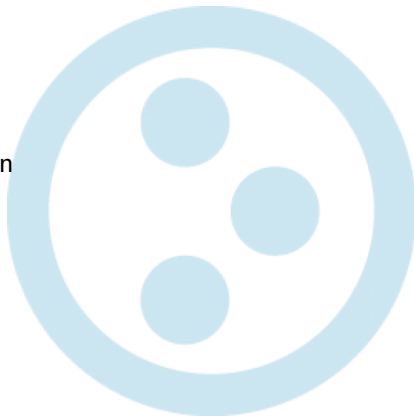
# Where we are

- 1 Knoledgements
- 2 Use cases
- 3 The challenge
- 4 Our way
- 5 Live coding
- 6 **Details on how we did things**



# General Backend

- ATTopics → Collection



# General Backend

- ATTopics → Collection
- RichText → Volto Tiles



# General Backend

- ATTopics → Collection
- RichText → Volto Tiles
- Portlets

- What we Polish to enter in Volto land:
  - Use Collective Folderish Types



# The Volto part

- What we Polish to enter in Volto land:
  - Use Collective Folderish Types
  - Deal with default pages



# The Volto part

- What we Polish to enter in Volto land:
  - [Use Collective Folderish Types](#)
  - Deal with default pages
  - Convert RichText HTML to Volto DraftJS (node utility)

# The Volto part

- What we Polish to enter in Volto land:
  - [Use Collective Folderish Types](#)
  - Deal with default pages
  - Convert RichText HTML to Volto DraftJS (node utility)
  - Easily point to old website when content not imported

# The Volto part

- What we Polish to enter in Volto land:
  - [Use Collective Folderish Types](#)
  - Deal with default pages
  - Convert RichText HTML to Volto DraftJS (node utility)
  - Easily point to old website when content not imported
  - Fix URLs (planned resolveuid)

# The Volto part

- What we Polish to enter in Volto land:
  - [Use Collective Folderish Types](#)
  - Deal with default pages
  - Convert RichText HTML to Volto DraftJS (node utility)
  - Easily point to old website when content not imported
  - Fix URLs (planned resolveuid)
  - Simple Folders → Document with Collection Block (planned)

# The Volto part

- What we Polish to enter in Volto land:
  - [Use Collective Folderish Types](#)
  - Deal with default pages
  - Convert RichText HTML to Volto DraftJS (node utility)
  - Easily point to old website when content not imported
  - Fix URLs (planned resolveuid)
  - Simple Folders → Document with Collection Block (planned)
  - Simple Collection → Document with Collection Block (planned)