Powering Data Teams Across the AI Journey

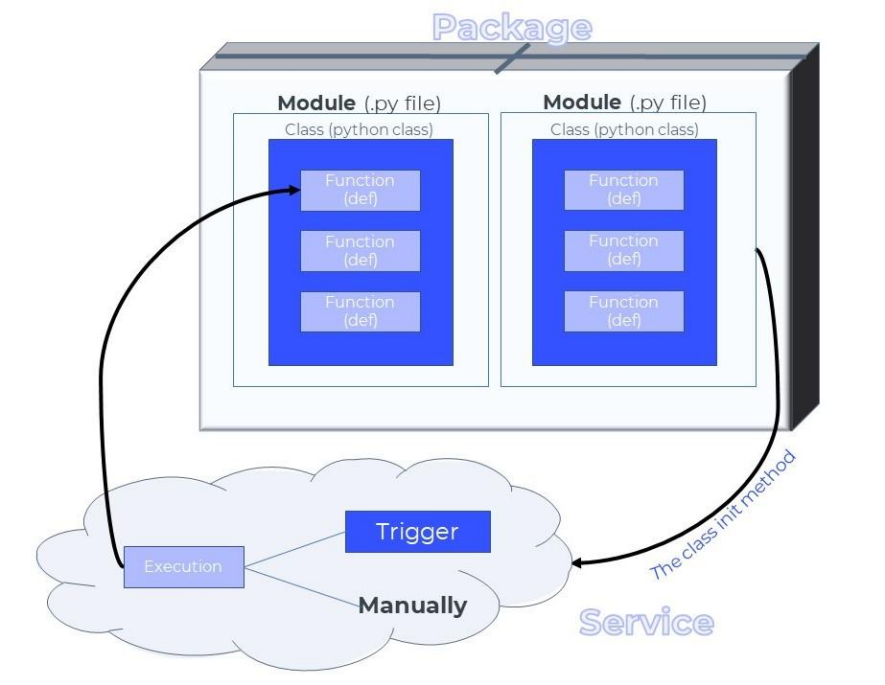
Training sessions- Session 2 FaaS & pipelines

Session 2 - Agenda

- What is FaaS?
- FaaS concepts
- Code examples
- Run +Debug in UI
- What is pipeline?
- Summary & materials
- Hands-on Task

What is FaaS?

The Dataloop Function As a Service enables users to deploy code packages and run services based on them on the Dataloop resources.
It allows flexibility and automation of processes, and allows to maximize the platform capabilities.



Entry point:

The first step to set-up your FaaS is to write your code in a python file that is going to be the **entry point** for the FaaS your building, by default we call it **main.py** .
The structure is **constant**,but you can change the:

- Function params.
- Code within run and init methods.
- Addition of other methods and classes.
- Additional imports.

Go to Code Example.

```
import os
import time
import dtlpy as dl
import logging
logger = logging.getLogger(name=__name__)
class ServiceRunner(dl.BaseServiceRunner):
    """
    Package runner class
    """
    def video_to_frames(self, item):
        """
        Splits video to frames, upload the frames and create a task for them
        :param item:
        :return:
        """
        # Extract snapshots from video and upload
        snapshots = dl.utilities.Videos.video_snapshots_generator(
            item=item,
            frame_interval=30
        )
        print('Successfully split item to frames and uploaded to dataset')
        # create an annotation task for snapshots
        task = self._create_task(
```

Module:

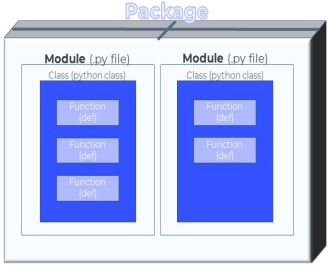
In the Dataloop FaaS, the previous entry point class is called a module and represents an **SDK object**.

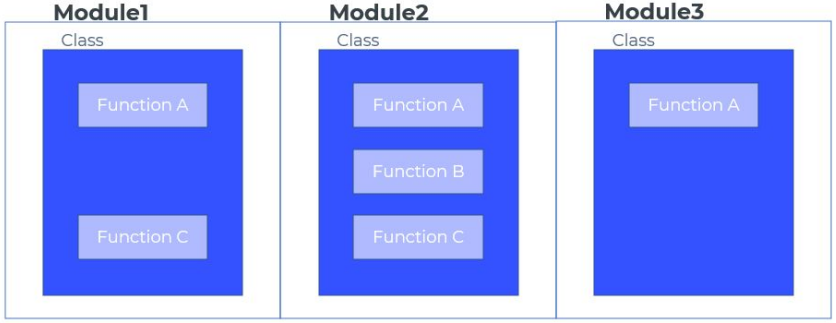
You need to define you modules in a new **.py** file and connect them :

```
module = dl.PackageModule(entry_point='main.py',
                          functions=[ dl.PackageFunction(
                                  name='video_to_frames',
                                  inputs=[
                                 dl.FunctionIO(type="Item", name="item")
                                      ],
                              description='Split video to frames and upload images'
                                          'on which a task will be created'
                          )
                         ])
```

Package:

A package is the main entity that holds all of the entities together,a static code with a schema that holds all the modules, functions, and the code-base.





The Package entity also stores the revision history of itself, allowing services to be deployed from an old revision of a Package.

The idea behind separating a package into modules is to allow users to develop many different services that share some mutual code using one codebase. This means you can translate and define the code-base or a substance of it to a service, without needing a new code-base.

Bot:

A bot is a service user that has developer role permissions within a project.

In Dataloop, bots are used to run services. Once a service has been deployed, it will log-in using the bot with which the service was created or the bot will be created automatically

Bots exist for security reasons (e.g you don't want your token to run the code on every Annotator click). If a function creates an annotation, the bot will be the creator.

More about conditions and warnings.

Create a Bot

```
project.bots.create(name='botman-name')
```

Use a specific bot for a service

```
service = package.deploy(
    service_name='my-service',
    bot='botmanemai.dataloop.ai'
)
```

Service:

A Service is a running deployment of Modules within a Package.

It specifies from which Module it should be deployed,what arguments should be passed to the Module's exported class' init method.

You can define the infrastructural configuration for it,like:

- Number of replicas-number of machines that run at the same time.
- CPU/memory requirements per replica

Once the service is ready, one may execute any available function on any input*.

Remember:
You must have a bot in order to deploy and run a service

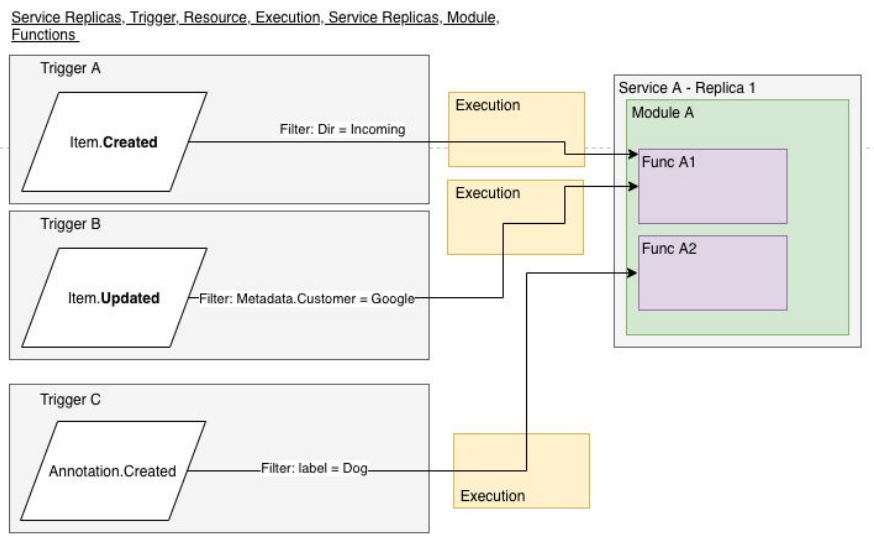
Trigger:

The Dataloop FaaS enables you to register Functions to events in the system.
Basically a Trigger triggers the FaaS to execute .

A Trigger contains:

- Project: on which it monitors events.
- Resource type:Item,Dataset,Annotation,Item_Status,Task.
- Action:on the resource-created,updated,deleted.
- DQL filter:If you want to filter the data that's going into the function.

For more information.



Technical demonstration

Now we'll do a technical demo and cover the following topics:

- Full code example of how to deploy a FaaS with all its components(you can use it as a template).
- Platform guidelines, where and how to run a FaaS and debug it.

What is a Pipeline?

Wave together humans and machines to process data in a pipeline architecture, a series of components, where each component's output is the input of the next one.

The Dataloop pipeline process allows **transitioning data** between:

- labeling tasks
- quality assurance tasks
- functions installed in the Dataloop system
- code snippets
- machine learning (ML) models

Your data can be filtered by any criteria, split, merged, and change status in the process.

Facilitate any production pipeline

Automate operations using applications and models

Preprocess data and label it

Post process data and train models of any type

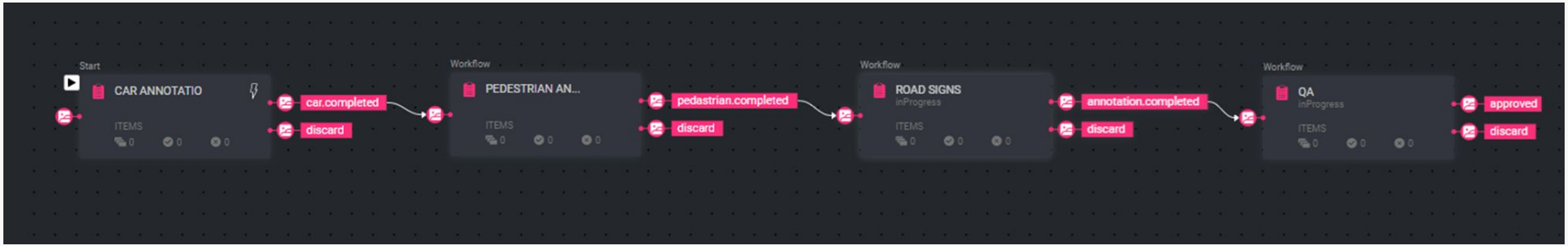
Technical demonstration

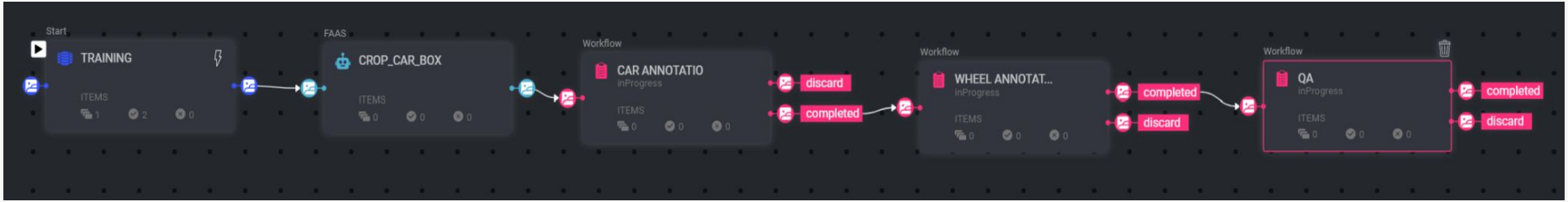
Now we'll do a technical demo and cover the following topics:

- Creating pipelines
- Drag and drop the components
- Templates
- Pipeline example
- Information panel

Pipelines examples:

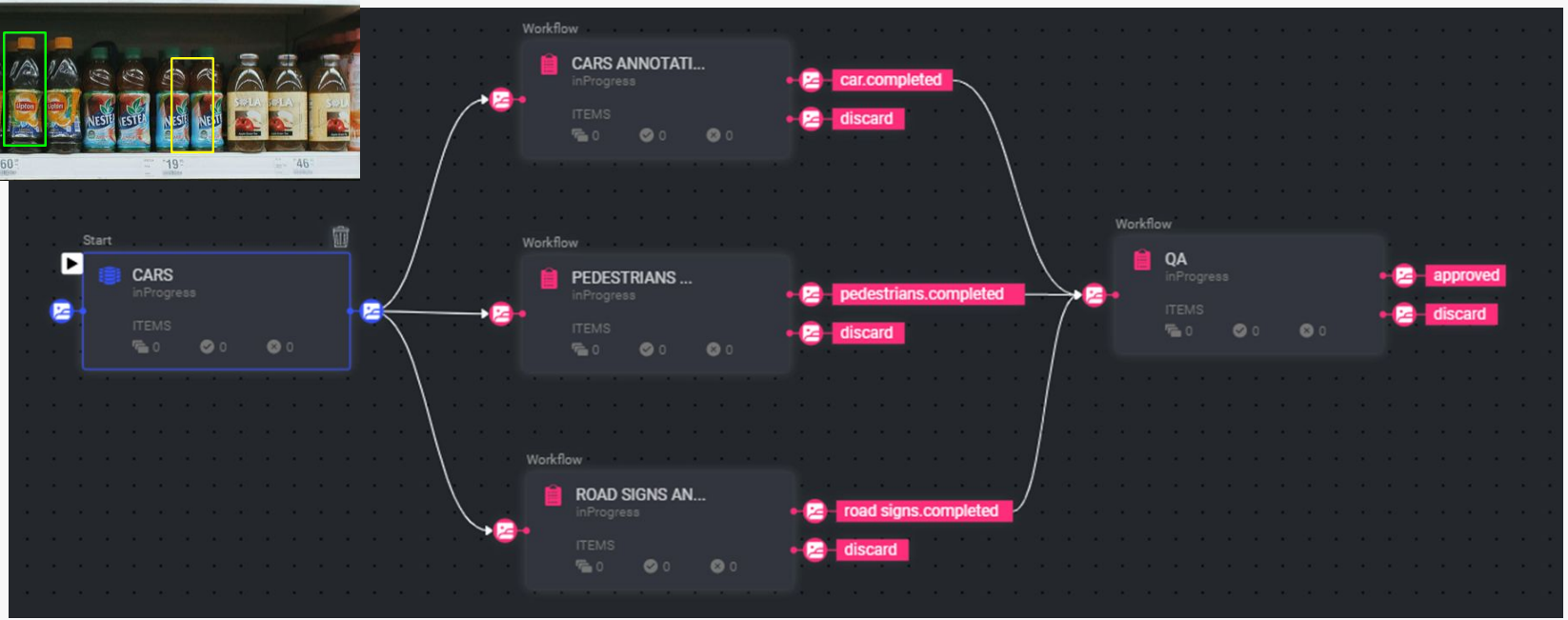
Micro-tasking





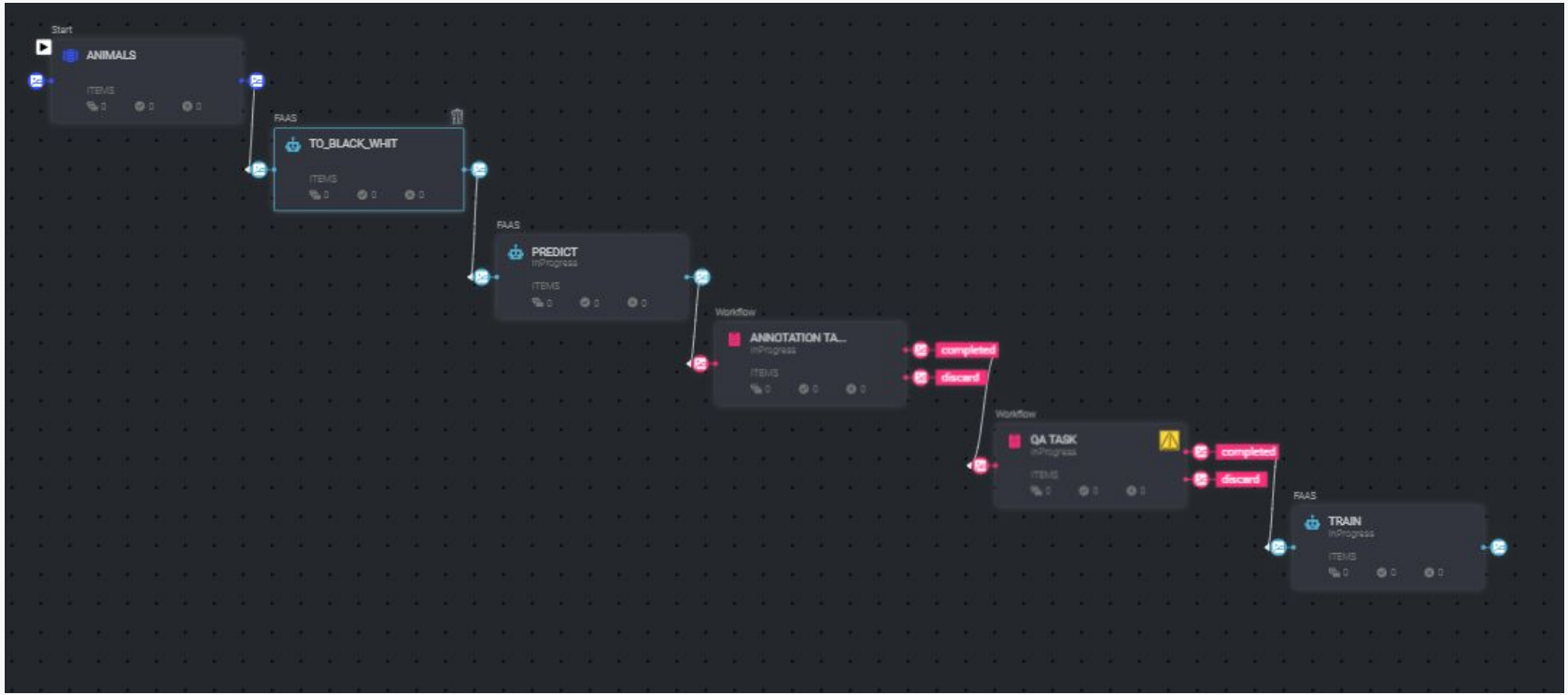
Pipelines examples:

Multi-tasking



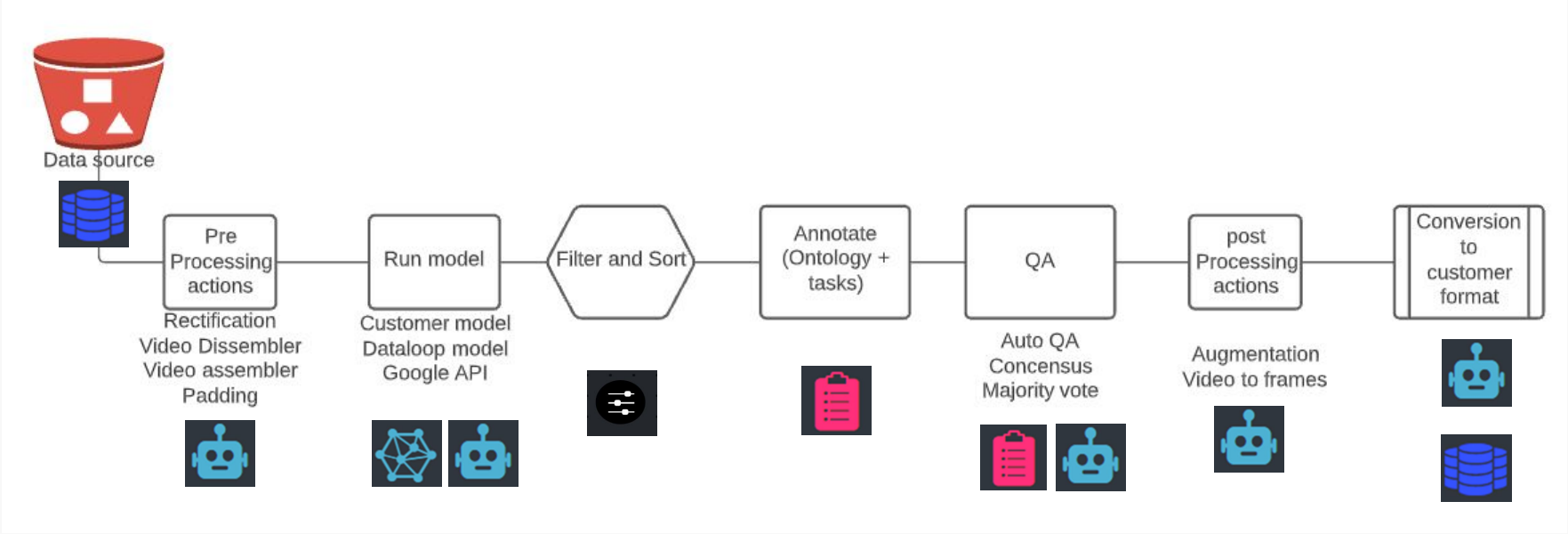
Pipelines examples:

Train your model



Pipelines examples:

Train your model-Customizability



Review

Now you know the how to build,run and debug our FaaS, later on we'll dive into more advanced FaaS capabilities .

You can visit our documentation here.
Check out the FaaS introduction .
More about Packages here.
More about Modules and entry point here .
More about Service here.
More about Triggers here.
More about Bots here.
Github link to the FaaS example template.
Check out the Pipeline overview .
More about how to create pipeline.
More about pipeline components .
More about Running and monitoring pipelines.

if you have any questions maybe you'll
find your answer here, if not feel free to ask us any time !

Task

Open the Task here.

Thank You!