Typical approach to AI/ML deployment

Amazon/Azure/Google/Watson APIs
Other APIs
Python/R libraries

Data Scientists
Solution Architects
Developers
Data Normalisation
Data Lakes
Model Building
Model Pipelining
Production Integration

Unique bespoke system

Enterprise

Our solution to AI/ML deployment and Data Science

Amazon/Azure/Google/Watson APIs
Other APIs
Python/R libraries

AICHOO
AI OS

Enterprise

AICHOO OS provides the easiest, quickest, lowest-risk implementation for any enterprise machine learning application, with minimal need for specialist staff such as data scientists.

Benefits of Aichoo OS

Intelligently automates key processes present in the data science activities.

Automatically evolves ML pipeline using library of universal Agents achieving true self-learning capability.

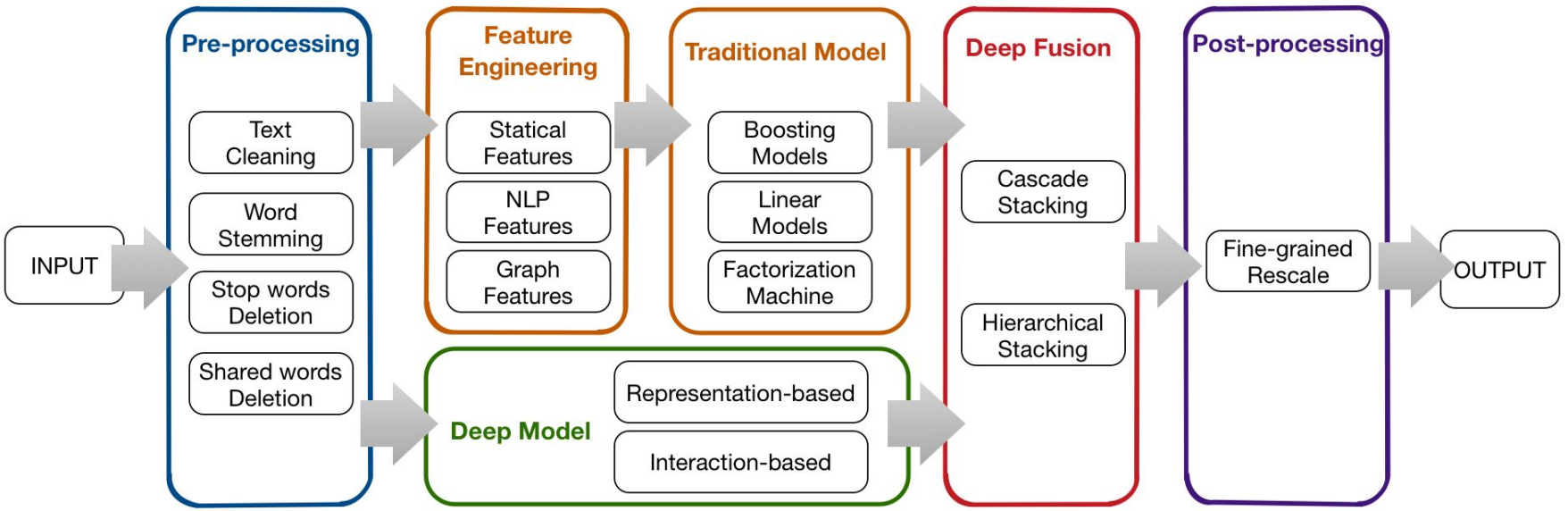
Data science project becomes a structured and re-useable process which can reach tangible results within weeks.

Data scientists can concentrate their brain power on developing portable AICHOO Agents, which can be re-used on other projects and offered for sale to others via marketplace.

Once the marketplace has enough Agents which can be activated on the AICHOO instance, companies most likely require no data scientists for typical projects as all procedures can be configured and executed by skilled IT personnel or even business line staff.

From a business perspective AICHOO provides the missing framework that all clients are implicitly expecting because they assume that this is what AI tools provide.

Typical ML Pipeline



Source: https://github.com/HouJP/kaggle-quora-question-pairs/blob/master/img/flowchart.png

- Fixed architecture
- Pipeline hard-coded in e.g., Python
- Difficult to integrate different steps into coherent process
- Difficult to parallelise computations
- Difficult to optimise end-to-end
- Difficult to change and re-use
- Not obvious how to move to live production in an enterprise environment
- Impossible to keep up-to-date with changes in original source data or environment

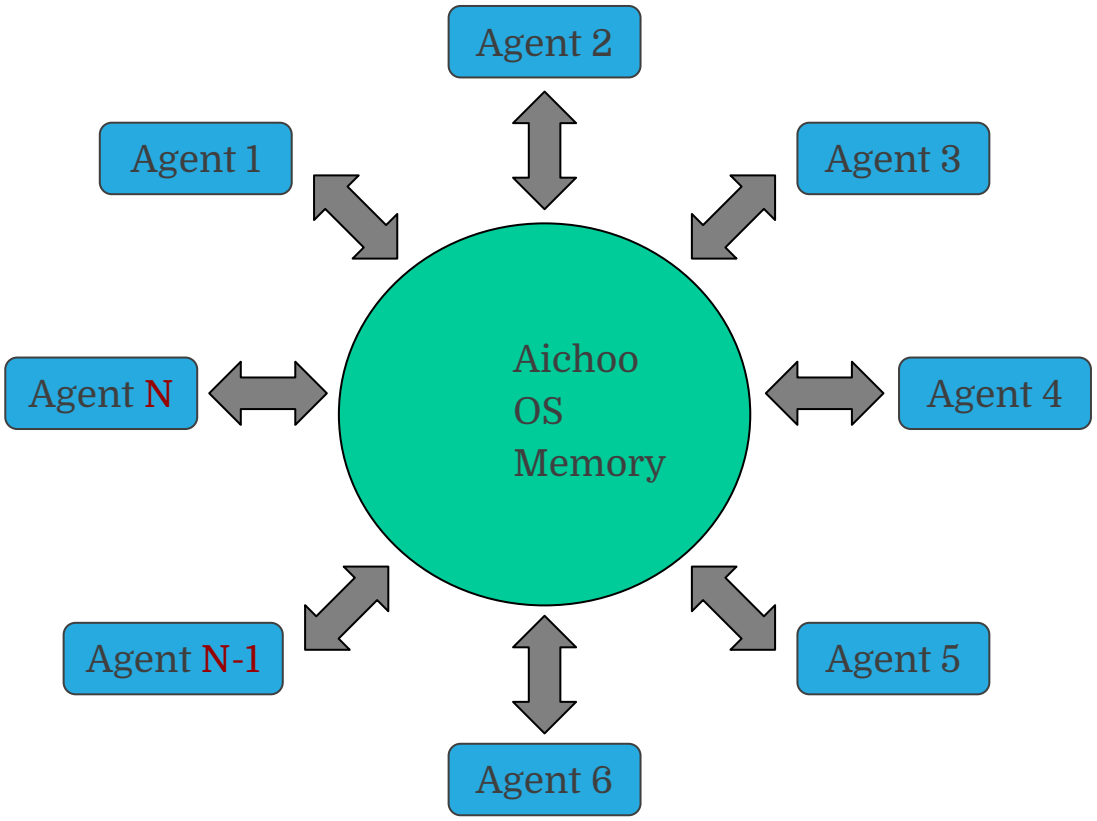
Aichoo ML Pipeline

Aichoo OS Memory (column based database)

	question1	azure_ta_key_phrases_question1_1_lang	azure_ta_key_phrases_question1_1_lang_sc	azure_ta_key_phrases_question1_1_0	azure_ta_key_phrases_question1_1_1	azure_ta_key_phrases_question1_1_2	azure_ta_key_phrases_question1_1_3	azure_ta_key_phrases_question1_1_4	azure_ta_key_phrases_question1_1_5	azure_ta_key_phrases_question1_1_6	...	azure_ta_key_phrases_question1_1_10	azure_ta_key_phrases_question1_1_11	azure_ta_key_phrases_question1_1_12	azure_ta_key_phrases_question1_1_13	azure_ta_key_phrases_question1_1_14	azure_ta_key_phrases_question1_1_15	azure_ta_key_phrases_question1_1_16	azure_ta_key_phrases_question1_1_17	azure_ta_key_phrases_question1_1_18	azure_ta_key_phrases_question1_1_19
0	What is the step by step guide to invest in sh...	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	What is the story of Kohinoor (Koh-i-Noor) Dia...	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0
2	How can I increase the speed of my internet co...	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	Why am I mentally very lonely? How can I solve...	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	Which one dissolve in water quikly sugar, salt...	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
5	Astrology: I am a Capricorn Sun Cap moon and o...	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
6	Should I buy	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
7	How can I be a good geologist ?	en	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8	When do you use シ instead of し?	en	0.75	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

- Universal architecture
- Any possible pipeline
- Transparent and flexible
- Agents are multi-purpose
- Fits enterprise environment
- Easy to optimise and adapt

Aichoo ML Pipeline



Heliocentric Architecture

- Infinitely expandable
- Unrestricted connectivity between Agents
- Infinite complexity
- Easy to change configuration and optimise to objective
- Parallelised by design
- Super deep network of Agents

Aichoo OS Agent

- Any self-contained procedure that reads columns from Aichoo OS Memory and adds new columns to the Memory as its outcome
- Agents executed on Jupyter Notebooks server
- Agents can be created in any of the 40 languages supported by Jupyter e.g., Python, R, Julia, Scala etc.
- Agent is capable of receiving any type of column as source
- Agent has parameters that affect its procedure and outcome
- Source columns are given to Agent's instance by Aichoo OS Kernel
- Multiple instances of any Agent with different parameters can be created and executed in parallel by the Kernel

- Agent is implemented as a class with two required methods e.g., in Python:

```
class cls_agent_{id}:
  def run(self, mode):

  def apply(self, df_add):

agent_{id} = cls_agent_{id}()
```

- Agent can call any external API (e.g., Azure/Watson/Google/AWS) to perform its function

Aichoo OS Agent Types

Simple Agent

Typically receives data from other sources, transforms or converts columns into other formats. It usually has fixed parameters that do not need optimising.

Evolving Agent

Complex procedure that takes subset of available columns in Memory, builds a model and creates a simulation of the input data based on its model. It is called Evolving because OS Kernel implements an evolutionary algorithm to hyper-tune all such Agents' parameters. Evolving Agent literally evolves and self-improves with time.

Each type of Simple and Evolving Agents copies itself as many times as allowed by computational/hardware resources, creating a multitude of agent instances. Consequently each instance of Simple and Evolving Agents continually expands Aichoo OS Memory with its output.

Target Agent

It is essentially the same as Evolving Agent but it is always configured to construct the final objective function needed for the decision making target specified by user.

Output Agent

Procedure that assembles the final pipeline of Simple, Evolving and Target agents needed to produce the decision on new incoming data. It creates a complete executable code which is loaded into memory of the computing cluster with all required pre-loading of data and models. It provisions such code as REST API and handles API requests to apply the final model pipeline on new incoming data.

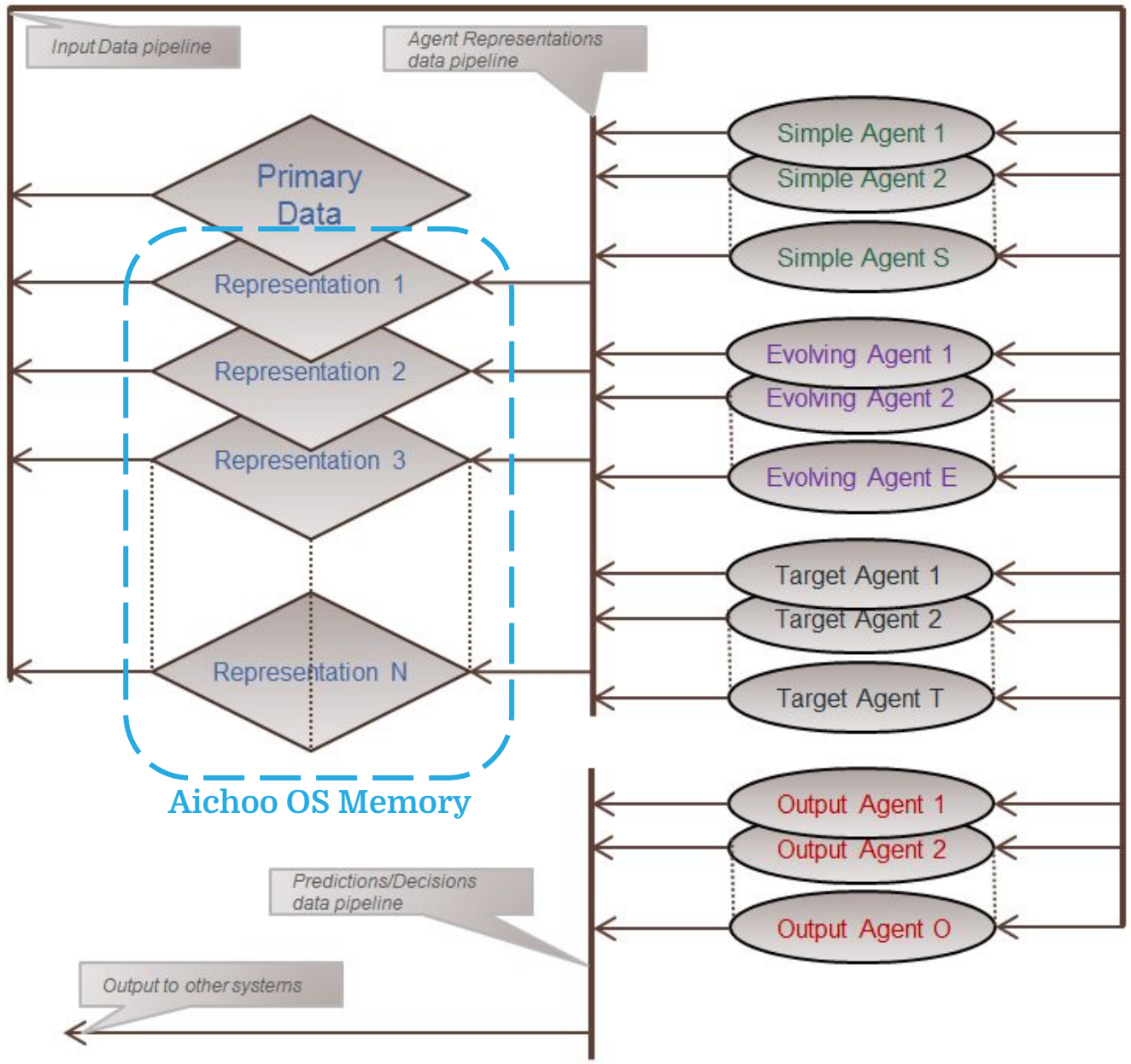
Report Agent

It is similar to Output Agent but its purpose is to produce reports based on all data available in Aichoo OS Agents and Memory.

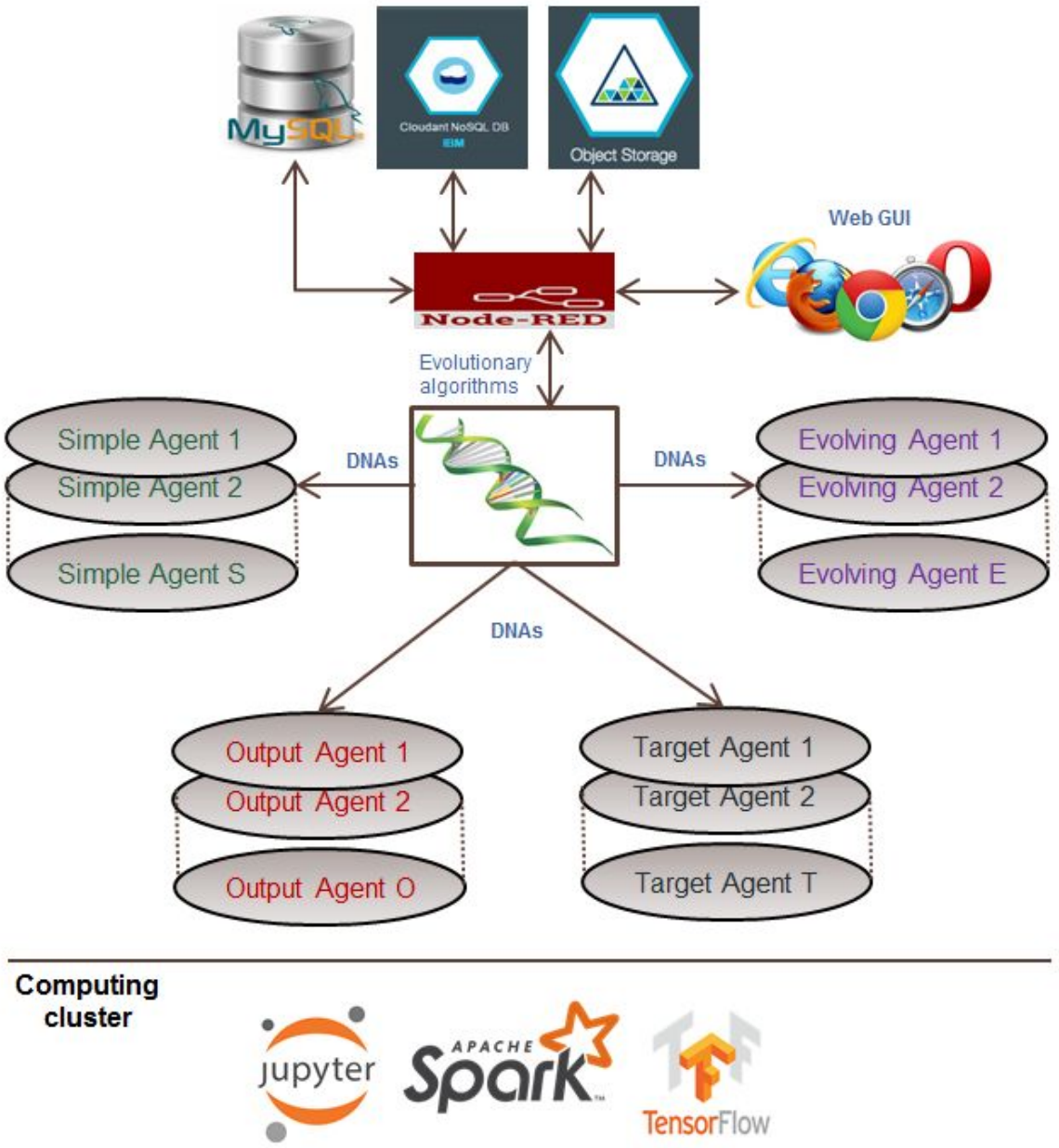
Aichoo OS Agent Library

- Agent's code is fully transparent
- Libraries are kept in GitHub and Aichoo GUI integrates with GitHub
- Aichoo maintains a public library of Agents
- Public library already has hundreds of Agents implementing most common data processing and ML procedures
- Public agents already created for integration with Watson/Azure/AWS/Google APIs
- Users can fork public library and continue developing on their private GitHub repos

Aichoo OS Data Flow Architecture



Aichoo OS Kernel Components Architecture



IBM Watson

Microsoft Azure

Google Cloud

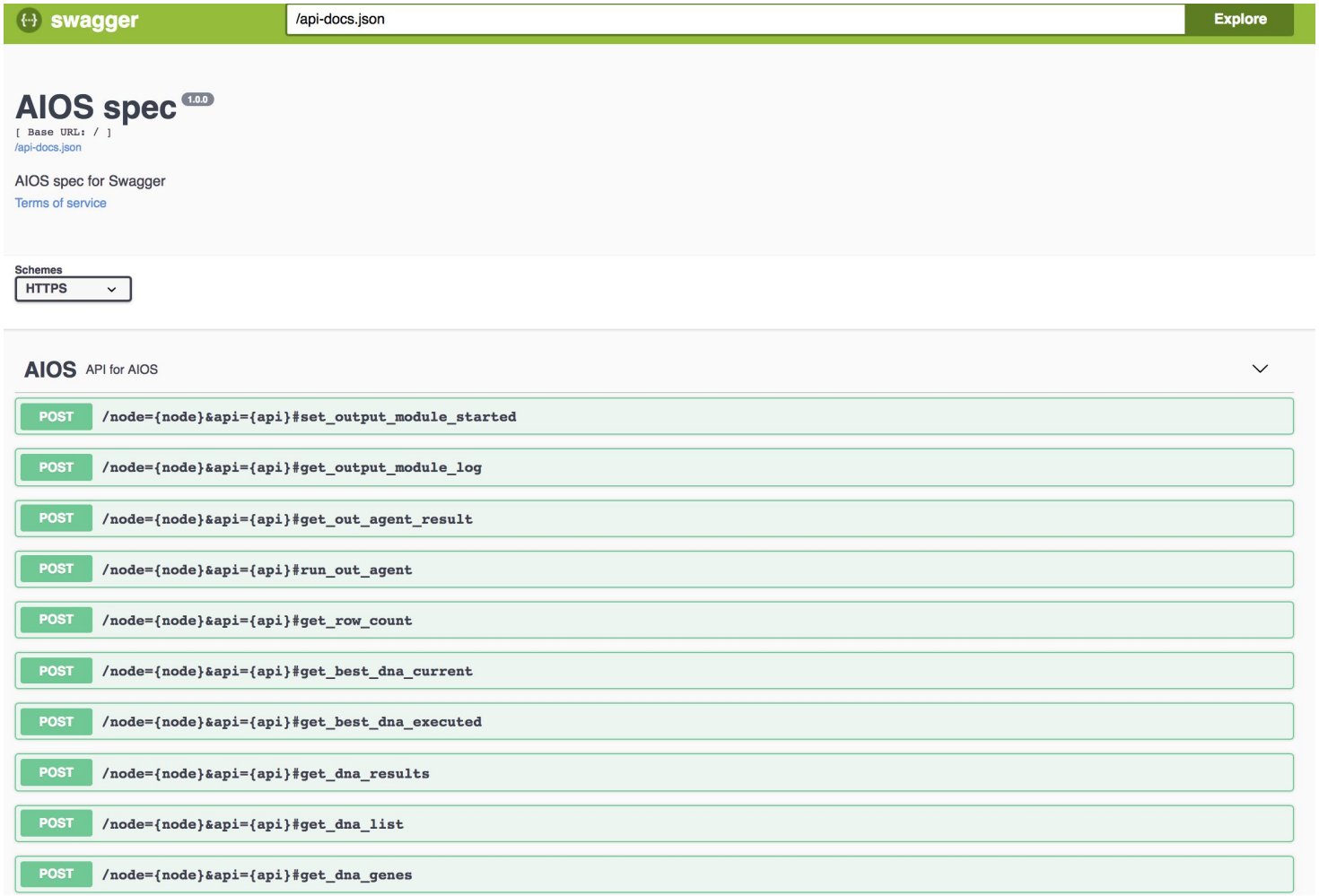
aws

- Kernel implements self-learning evolutionary process
- Creates most suitable pipeline of agents needed to achieve the objective
- Agents loaded from library in GitHub

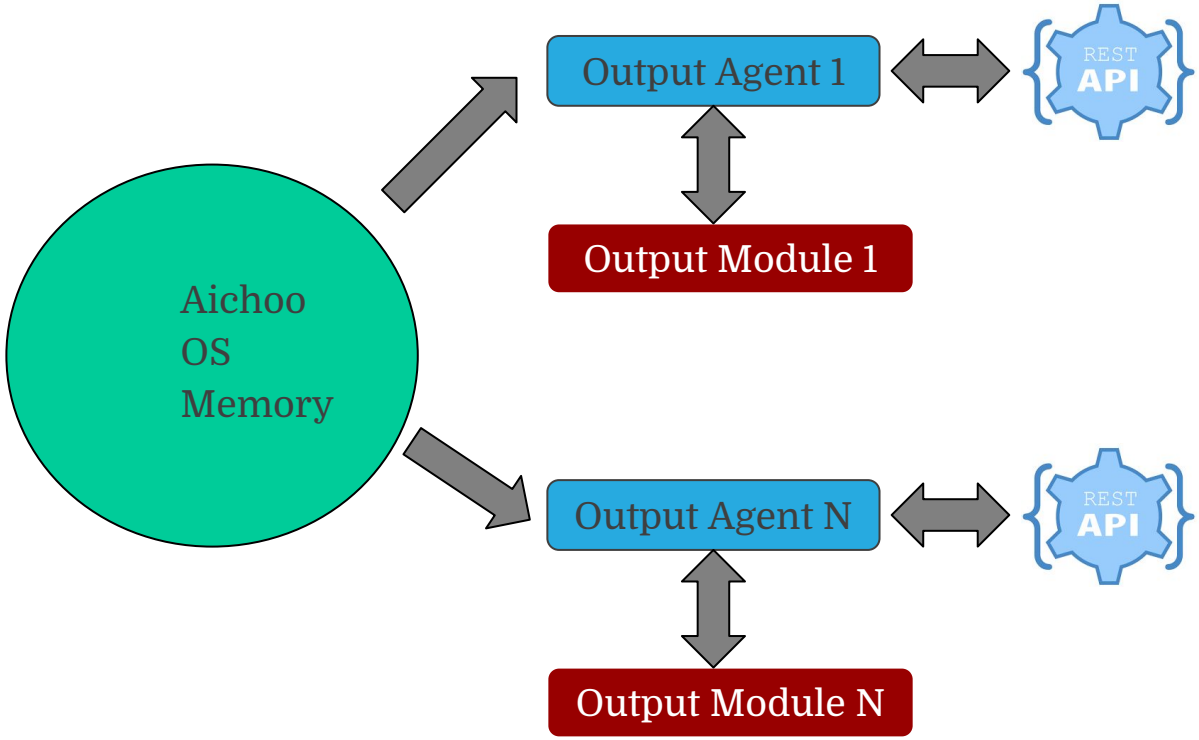
Aichoo OS API

Output Agent

Procedure that assembles the final pipeline of Simple, Evolving and Target agents needed to produce the decision on new incoming data. It creates a complete executable code which is loaded into memory of the computing cluster with all required pre-loading of data and models. It provisions such code as REST API and handles API requests to apply the final model pipeline on new incoming data.



Aichoo OS API



- For each pipeline that needs to go to production user would create Output Agent
- Output Agent assembles the final pipeline only of the agents needed to produce the decision on new incoming data
- It creates an instance of Output Module which is a segregated Jupyter Kernel into which all agents code and models are pre-loaded
- Output Agent provisions such pre-loaded segregated code as secure REST API and handles API requests to apply the pipeline on new incoming requests