



Aito Technical Service Description

This description is a draft of the aito.ai technical service description. The service and this description are bound to change in the future. For more information questions, comments or suggestions please contact us.

1. Offering maturity

Aito.ai is in pilot phase, meaning that we do not offer fixed pricing or SLAs yet, but work in co-operation with our customers to provide a best-effort service to selected customers.

Please contact us if you are interested in taking part in our pilot program.

2. Operation model

Aito.ai is a software-as-a-service company (**SaaS**), which offers a database-like solution with Machine Learning- (**ML**) and Artificial Intelligence- (**AI**) functionalities.

SaaS means that we handle all the server infrastructure, scaling, encryption certificates, storage and backups on behalf of our customers. The service can thus be seen as a turn-key service, with the customer being responsible for building the integration to the provided API, and Aito taking care of all the management and running the service.

Our service runs on cloud infrastructure provided by Amazon Web Services (**AWS**), and does not operate own hardware.

3. Solution description

Aito offers a comprehensive solution for data indexing and querying. Aito offers an API-interface (Application Programming Interface) that customers can use to upload, query and manage the data. Aito provides the API as a REST-like API, with JSON as the data-interchange protocol. Each customer gets a custom endpoint (<https://<customer>.api.aito.ai>) and needs a set of secret API-keys to access the system.

Each endpoint is private to the customer, and the API-keys are used for authentication and authorisation. Aito separates the API keys into two segments, read-only and read-write. Read-only keys can be used to query the data, whereas the read-write keys are required for adding or deleting data, or to change the index schema for the data.

The Aito API is under development. The exact version of the API is fixed for each environment, and the detailed API specification and documentation is provided separately with each environment. We strive to keep compatibility with older versions of the API, but this cannot be guaranteed during the pilot phase. We will inform our customers of such changes and help them to cope in the optimal way.

4. Technical architecture

Aito SaaS currently runs on AWS in the eu-west-1 region. The region is located in Dublin, Ireland.

4.1. Index data

Data storage is normally structured with a primary data storage, normally a relational database (RDBMS or SQL-database), and with various indices for data lookup and access. An index is normally used to improve the performance or to group the data into a format better suited for certain kinds of operations.

A primary index usually contains subsets of data and guarantees the identity and uniqueness of the data. In most cases the primary index contains only a very limited set of columns from the data, needed to fulfil the required properties.

A secondary index refers to an index created for some larger purpose. A typical example is de-normalising SQL tables to larger documents for efficient searching. Another similar example would be to use a specialised secondary index for free-text searching, a task relational database usually are not optimised for.

We currently don't recommend using Aito as the main data store. Aito is best suited as an ML-/AI-enabled secondary index e.g. for a relational database. This allows repopulating Aito at some later stage from the original data store. Aito's data model is built for the specific purpose of running statistical operations for the data. We are still optimising the format, and this might put the integrity of the indexed data in jeopardy. Our ultimate goal is to make the data format robust enough for any given purpose, but we're still working to achieve this goal.

4.2. Aito Core

The core functionality of Aito is the custom core software, implementing a database with Machine Learning- and Artificial Intelligence- functionality. The database is built for extremely fast calculation of statistical properties within the stored data. These properties can range from finding a value for mutual information between fields in the data, prediction of new field values based on the existing data set, or matching different datasets and items based on limited sets of examples. Aito also supports fast index queries for fetching and filtering data. By utilising these statistics one can achieve a deeper understanding of the properties and relations within the dataset.

Aito operates on structured and linked data. The data is stored in tables conforming to a customer specified schema, much like in a traditional RDBMS (SQL-database). Aito also supports links between entries in the tables, much like foreign keys in a RDBMS. Aito, however, is not transactional by nature. This restriction is due to the optimisation for statistical calculations rather than ultimate data integrity.

Aito differs from traditional AI-solutions in that it does not require a separate model-phase where the model creation is separated from querying the model. Instead the database operates on the entire store dataset, and can adapt very quickly to changes in existing or to addition of new data. New data is included in queries in "real-time", meaning that the query results are affected by this new data within seconds. We call this *ad-hoc modelling*.

Another distinct feature of Aito Core, compared to traditional AI-models, is the ability for interactive querying. As the system can incorporate all the stored data in any single query, it is possible to query only parts or the entire dataset with a single query. Hence new tables and data can be added at any point in time, and immediately be taken into use. This gives tremendous flexibility compared to having to update an AI model, or re-running the entire modelling phase in order to use any new data as part of the decision model.

4.3. Aito API

The technical API-documentation is available with each Aito-instance, at the path 'https://<customer>.api.aito.ai/api-docs/v3/'. The API document follows the [OpenAPI Specification](#), formerly known as Swagger. The API is still under development, but each environment contains the up-to-date documentation for the respective environment.

Aito supports querying using a REST-like API, with the data format being JSON. The API formatting and parsing is separated from the core functionality, meaning that it is possible to evolve the API independently from the internal data structures. Consequently it also enables to support backward compatibility for the API, while allowing the internal data structures to be optimised further in the future.

With the separation of query parsing and application comes the benefit of mitigating various problems and weaknesses that could lead to security breaches. JSON is the current de-facto standard for web based data interchange. Thus it is well understood and implemented in practically any language and framework. Aito uses standard and up-to-date libraries for JSON-parsing, meaning that we are effectively guarded against code and data-format bugs arising from mistakes in encoding and format.

The Aito API payload is always parsed in strict mode, so we do not accept queries which are not syntactically correct according to the query format. Invalid queries are hence dropped before they are applied on customer data.

5. Aito cloud architecture

Aito relies on AWS provided services to provide the functionality. Our aim is to provide the best possible service to our clients, so we reuse services and tools where ever these support our ultimate goal of high service quality. This helps us achieve high scalability as well as to avoid bugs and vulnerabilities, as the attack surface of our own code is as small as possible.

The external API endpoints are provided through the AWS [API Gateway](#). The domain name support is handled through [Cloudfront distributions](#), so all the Aito provided endpoints are delivered through AWS-owned IPs. API Gateway provides support for extreme scaling and customer specific usage plans. Aito can therefore offer a level of service where our SaaS and our users are close to immune to misconfigurations or malice by other customers. API Gateway also shields us against the all but very sophisticated Denial-of-service attacks by malicious third parties.

AWS API Gateway offers support for API key-based access control. The keys are checked at the edge, and requests without or with invalid keys never reach the core service.

The more detailed internal architecture details can be discussed on request. We're happy to share our experience and learnings with others.

6. Aito Service Level Agreement

Aito is in pilot phase. We don't offer fixed pricing or fixed SLAs. Please contact us to negotiate with us about your requirements. We believe we can meet your specific needs and requirements, but we don't feel comfortable in making blanket statements about the level of service.

7. Technical restrictions and considerations

There are some limits and aspects that customers need to be aware of when using our current implementation:

- The current limit of payload size is 6MB, both for inbound and outbound traffic. Headers, cookies and other parts of the message are counted into the payload, so the net limit is slightly smaller.
- Aito only support the service over https (TLS/SSL). This guarantees request integrity and end-to-end encryption, without significant cost on either server or client side. Our certificates are issued and signed by Amazon.
- Aito should currently be treated as a secondary index for your main data store. Please don't use it as the main and only store of your data.