Trifork Public
Åboulevarden 13
8000 Aarhus C
info@trifork.com
87 32 87 87

ATAH

6.16.1.1 Implementation guideline v. 4.1.1

TRIFORK.

Table of contents

1 Device description

This document contains guidelines for implementation of ATAH v. 4.1.z (x.y.z) in a client system.
For information about ATAH, please see the instructions for use on wiki.kliniskbeslutningsstøtte.dk.

2 Guidelines for implementation

This document is *guidance* only

The *visual* representation of decision support messages can be customised

The *content* of decision support messages shall not be modified

2.1 Prerequisites

2.1.1 Client system approval

New client systems who have not implemented ATAH before shall be approved by Region Nord prior to use. Please contact Region Nord via e-mail beslutningsstotte@rn.dk

2.1.2 Required resources

- Provided by the client system
    - E-mail for authentication
- Provided by the manufacturer, Trifork Digital Health A/S
    - API documentation site
        - API specification
    - URL to *test* environment
        - /authorize
        - /evaluate
        - /evaluateDrugMedications
    - URL to *production* environment
        - /authorize
        - /evaluate
        - /evaluateDrugMedications
- Password for authenticated user
    - Generated and sent to client system e-mail
- Valid token

- Unique per environment
- Expires every 2 years

2.1.3 Client system requirements

- Communication with a REST API
  - Protection via SDN (Sundhedsdatanettet)
  - Routing available via DGWS (Den Gode Web Service)
- HTTP client sending HTTP requests and receiving HTTP responses
  - JSON for REST
  - XML for DGWS
- Patient specific data
  - Birth date, NB: Not social security number (CPR-number) ⚠️
  - Gender
  - Pregnancy status (true/false)
  - Breastfeeding status (true/false)
  - Diagnoses
  - Cave (allergies)
  - Measurements
  - Drugs
    - Existing drugs (medication list)
    - New prescriptions
- Storage and processing capacity to handle requests/responses

2.1.4 Network requirements

- Secure- and reliable network connection
  - Communication via SDN (Sundhedsdatanettet) is recommended

2.1.5 Security Requirements

- Proper SLS/TLS configuration

SSL/TLS encryption certificates ensures secure communication between the client and ATAH.

2.2 Authorisation

Communication with the decision support API requires authorisation with a *bearer token*.

A token is,

- Unique per authenticated user
- Unique per environment (test and production)
- Valid for two years

ℹ Please store the *token* securely in the client system.

A valid token is required for all requests to ATAH.
To stay authorised, please request a new token *before* the current token expires.

2.2.1 How to request a token

- Download an API tool such as Postman
- Open Postman, create a collection with a POST request
- Insert the authorization url, NB: remember to replace “URL” with the correct URL

```
https://URL/v4/authorize
```

- Add header in “Headers”

Key	Value
Content-Type	application/json

- Add JSON request in “Body”

```
{
  "email": "email@email.dk",
  "password": "password"
}
```

- Click “Send”
- Copy- and save the generated token. Please note the date- and time of expiration.

2.3 RESTful API

A REST API (RESTful API) can handle operations (such as GET, POST, PUT and DELETE) in a stateless manner. When stateless, all requests are individually processed and have no connection. For this reason, all request to ATAH must contain all the required- and available patient-specific data.

When communicating directly with the REST API, requests/responses are written in JSON. ATAH only supports POST operations.

The REST specification is available on http://docs.exttest.kliniskbeslutningsstøtte.dk/v4.

2.3.1 Routing via DGWS

ATAH supports routing via DGWS by translating SOAP-based requests (XML) into REST-based requests (JSON), and REST-based responses (JSON) into SOAP-based responses (XML).

DGWS (Den Gode Web Service) is a Danish standard primarily used in healthcare for secure communication between systems. It ensures secure exchange of data by providing authentication, authorisation, integrity, and confidentiality in

web services, particularly for communication between healthcare providers and systems like EHRs (Electronic Health Records).

DGWS is based on SOAP (Simple Object Access Protocol), and it provides a security layer by using SAML (Security Assertion Markup Language) tokens for authorisation and identity verification. It is commonly used in Denmark for systems that need to comply with healthcare data security standards, ensuring that the communication between systems is both encrypted and authorised.

The GDWS specification is available on http://docs.exttest.kliniskbeslutningsstøtte.dk/v4.

2.4 Request headers

ATAH has various request headers that are either mandatory or optional.

- systemownername
    - Manufacturer of client system
- systemname
    - Client system name
- orgresponsiblename
    - Region / area
- orgusingname
    - Clinic / department / general practice
- messageId
    - A unique message id for traceability
    - Generated by the client system
- orgUsingId
    - A number that matches the OrgUsingIdType
- orgUsingIdType
    - medcom:ynumber
    - medcom:pnumber
    - medcom:skscode
    - medcom:cvrnumber
    - medcom:communalnumber
    - medcom:sornumber
    - medcom:locationnumber
- systemversion
    - Client system software version
- useDecisionSupportModulesHeader
    - List of enabled modules
    - If empty, default list is used
    - Default list contains all modules (R1-R12) *except* R8
- feedbackLinkWeb
    - Generates a link per message to the feedback web client
    - Default is true
- feedbackLinkService
    - Generates a link per message to the feedback service
    - Default is true
- r1-grouping
    - Grouping messages in R1

    - When more than two messages are provided for the same drug
    - Default is true
- r5-grouping
    - Grouping messages in R5
    - When more than two messages are provided for the same drug
    - Default is true
- r5-enableCombinationsWarnings
    - Enables messages of type R5-3
    - Default is false
- ValidationContextRich
    - Enables context-specific texts in validation messages
    - Default is false

2.4.1 Mandatory headers

The headers must be base64 encoded.

Header	Example value
x-request-bst-systemownername	CompuGroup Medical
x-request-bst-systemname	XMO
x-request-bst-orgresponsiblename	Region Nordjylland
x-request-bst-orgusingname	Mariager Lægeklinik I/S

2.4.2 Optional headers

The headers: systemversion, orgUsingId and orgUsingIdType must be base64 encoded.

Header	Example value
x-request-bst-systemversion	1.2.3
x-request-bst-orgusingid*	67911
x-request-bst-orgusingidtype*	medcom:ynumber
x-request-atah-useDecisionSupportModulesHeader	R1,R2,R4,R5,R7,R8,R10
x-request-atah-feedbacklinkweb	true
x-request-atah-feedbacklinkservice	true
x-request-atah-r1-grouping	true
x-request-atah-r5-grouping	true

Header	Example value
x-request-atah-r5-enablecombinationswarnings	true
x-request-atah-validationcontextrich	true
x-request-atah-messageid*	ABCD12345678

*Highly recommended to set, even if optional.
Setting these headers enables the manufacturer to analyze usage data in more detail.

2.5 HTTP request

Please use the *test* environment 'exttest' before the production environment

When sending a request, ensure you have

- A valid token to the specific environment (see section 2.2)
- Headers set, both mandatory and optional (as applicable)
- A valid request body (JSON or XML depending on routing)
    - Schemas are available on https://docs.exttest.kliniskbeslutningsstøtte.dk

Furthermore, it is important to consider *when* to send a request

- For a patient's medication list (existing medications)
    - Trigger when opening the list, or
    - Trigger when the user clicks a *button* to request decision support
    - Use endpoint + /evaluate for this
- For new medication (prescription)
    - Trigger continuously as the prescription changes (input data changes)
    - Use endpoint + /evaluateDrugMedications for this

Important considerations when implementing the decision support API in a client system:

It is important to consider triggers appropriate to the user's workflow, so that decision support is received at point-of-care, and is always relevant to the current input

2.5.1 Input validation

ATAH validates the input data to ensure requests are valid.

2.5.1.1 Validation types

- Hard error
    - The request is invalid
    - The

- Error
    - The request is valid but illogical
    - The invalid data is filtered out
- Omission
    - The request is valid but conflicts with rules in decision support rules
    - The invalid data is filtered out

Validation messages are sent by ATAH to the client system

Validation messages may be displayed in the client system's user interface, however, as validations are *technical* these may also be logged for developer's of the client system

2.6 HTTP response

When a request is sent to the endpoint with

- /evaluate
- /evaluateDrugMedications

with a POST operation, and the operation is *successful*,

- HTTP code 200
    - Successful response with decision support message(s),
- HTTP code 204
    - Successful response with no decision support message (no content)

If the request is unsuccessful, a number of error codes are possible. A code 4xx (e.g. 400) is a client-side error, and a code 5xx (e.g. 500) is a server-side error. All success- and error codes are available on page https://restfulapi.net/http-status-codes/ .

An example of a JSON response (RESTful API) and an XML response (DGWS) can be found on the API documentation site.

2.6.1 Handling a response

After receiving the PatientResponse from the ATAH API, the client system need to process the response and handle decision support messages appropriately. Follow the steps below to ensure logic in the client system manages the response effectively.

Parse the response

Once a response is received, the client system should parse the *PatientResponse* object:

- Extract the *warnings* array which contains the decision support message(s) that shall be displayed to the user.

Evaluate the warning, visualise based on this evaluation

Decision support messages in the warnings array contain a number of attributes.
These are:

- warningId
    - corresponds to the decision support module and warning type of that module
- warningType
    - The decision support module, e.g. WarningTypeR3

- warningSubType
    - The warning subtype of that module, e.g 1 (warning R3-1)
- warningText:
    - Contains a shortText, and a reasonText
        - shortText:
            - The *summary* / *title* of the message
            - Should be displayed clearly
        - reasonText:
            - The reasonText contains an actionText and a backgroundText
                - actionText
                    - Detailed decision support - reason why the message is received, e.g. the maximum dose is exceeded.
                    - backgroundText: Information about patient data that contributes to the calculation, e.g. age or measurements.
- severity*
    - Severity of the message
    - Responses are divided into different categories based on severity.
    - Contraindications (red) are the most important messages
- links**
    - Link to additional information on http://pro.medicin.dk
    - Link to scientific literature (provided by http://pro.medicin.dk)
- references
    - Scientific literature used to evidence-base the specific decision support message (where available)
- sources
    - Patient specific sources (birthdate, measurements)
- meta
    - Information about datasets, versions
- feedbackLinkWeb
    - Link to a web client where users can provide feedback to a *specific* decision support message or report issues
- feedbackLinkService
    - Link to a service for quick feedback of a *specific* decision support message
- docsURL
    - Link to the API documentation site

*The available severities and sub-severities can be seen on the API documentation site
**Additional link(s) are available in the *response headers*, see 2.6.2 Response headers.

2.6.1.1 Visual representation of response

The client system is responsible for defining the visual presentation of decision support messages. However, the following national guidelines should be taken into consideration. The guideline for visual representation is documented by Region Nord, who manages ATAH on behalf of the Danish regions.

(MUST)*	• Contraindications of enabled modules must be shown
	• Contraindications must be coloured red
	• Cautions must be coloured yellow / orange
	• The following use cases must be implemented for the physician's workflow
	  • Show "warnings" (messages) in the patient's list of current medications
	  • Continuous showing "warnings" (messages) in the prescription process
	  • Showing "warnings" decision support when an existing prescription is changed
	• When a list of messages are shown, contraindications must show before cautions etc.
	• IF users can change settings; An option to *reset* settings must be available
	• The patient's medication list must clearly indicate the number of decision support "warnings"

MUST NOT	• The decision support system must not be deactivated fully • Redundant (duplicate) messages must not be shown during the above workflows
SHOULD	• An icon of appropriate color and symbol should be shown for "warnings" in the medication list • The symbol for a contraindications should be a triangle and the symbol for a caution should be a circle • Cautions should always be enabled • The shortText of the decision support message should be presented clearly to the user
MAY	• Decision support modules and/or message severities may be enabled / disabled on a *user level* • Severities *below* contraindications may be disabled • During prescription, solely contraindications and cautions (optional) may be chosen to show • Module R8 is disabled per default, but may be enabled when the user wish to use the module.

*Strongly recommended. Even though considered a *guideline*, please justify any deviations from the above listed guidelines for implementation (visual representation).

2.6.2 Response headers

A number of response headers are provided with each response, including:

- Request id
- Unique device identification (UDI)
- Link to documentation site, including
  - Product *label* (for CE-marking)
  - Instructions for use, electronically in PDF format
- Used decision support module(s)

Please visualise headers to users to enable access to the device label- and instructions for use.

Response header	Value (example) + description
x-request-id	Example: 56a00ece-d7f9-4078-b494-dbf5844524b5 The unique id of the request, in which a *response* (decision support message) is provided to. Can be used for investigating errors.
x-response-atah-udi	Example: (0)5704002040044(8012)4.1.1 Unique Device Identification (UDI) of the specific ATAH version used for decision support. The number is provided by the manufacturer, Trifork Digital Health A/S, and (8012) means that it is a standalone software product, and the software version is 4.1.1.
x-response-atah-docs-url	Example: https://docs.develop.xn--kliniskbeslutningssttte-4mc.dk/ A direct link for the API documentation site, which contains information about ATAH, decision support modules, contact information etc. Most importantly, users can identify the product via the product label and access the instructions for use (IFU) via the link available in the label.
x-response-atah-useddecisionsupportmodulesheader	Example: "R1" Shows the module(s) requested with a request header, the corresponding "x-request-atah-useDecisionSupportModulesHeader", see section 2.4 Request headers.

2.7 Decision support modules

The available decision support modules are,

- Interactions (R1)
- Cross reactivity (R2)
- Maximum dose (R3)
- Reduced kidney function (R4)
- Double prescription (R5)
- Children (R6)
- Decision support based on diagnosis (R7)
- Indications (R8)
- Pregnancy (R9)
- Lactation (R10)
- Antibiotics (R11)
- Monitoring (R12)

R refers to *rule* as ATAH consist of rule-based decision support modules

By default, all modules, except module R8, are enabled. If only specific modules shall be requested, this shall be done by providing a comma-separated list of modules in the header: *x-request-atah-useDecisionSupportModulesHeader*.

For a description of the modules, please see the instructions for use on webpage wiki.kliniskbeslutningsstøtte.dk.

2.7.1 Patient data in requests

The syntax for patient data inputs are available on the API documentation site.

A *request* must consist of the necessary patient data to evaluate decision support.

- Gender
- Birthdate
- Cave
- Diagnoses
- Measurements
- Pregnancy status
- Lactation status
- Current medications
- Any new medications
- Current unregistered medications
- Any new unregistered medications

Gender	Gender must be either "female" or "male"
Birthdate	A birthdate must be provided in simple date format YYYY-MM-DD. This must not contain a CPR-number.
Cave	Cave must have the MedicineAllergy flag, see the API documentation for more information. A patient's Cave (allergies) must be provided with either drug id or ATC-code depending on the Cave registration. A patient can have multiple Caves registered with ATC-codes and/or drug ids, separated with a comma. Drug ids are available from the Danish Medicines Agency (Lægemiddelstyrelsen) and thus not provided by the system. NB: A unique local reference (localref) shall be generated
Diagnoses	A patient's diagnoses can be provided with either ICD-10, ICPC-2 or ICPC-2E diagnosis codes. If providing the diagnosis with ICPC-2E, the ICD-10 code will be favoured over the ICPC-2 code. The date format for diagnosis is YYYY-MM-DDTHH:mm:ss.SSSZ (ISO 8601 standard, with milliseconds) NB: A unique local reference (localref) shall be generated
Measurements	All measurements with an NPU code (https://npu-terminology.org/npu-database/) are valid when sent with a request. However, only specific measurements are useable in the decision support system. The codes / measurements used by the system are; Height, weight, GFR, Sodium, Potassium and Hemoglobin. Weight shall be given in *kilos* (e.g. 70.5) and height shall be given in *meters* (e.g. 1.7). Any value with decimals shall be separated with a period "." The operatores: <, >, <=, >=, = are accepted for all measurements. See syntax on the API documentation site. NB: A unique local reference (localref) shall be generated
Pregnancy status	A patient's pregnancy status shall always be provided for a request to be valid. Default is *false*. Please note, if the patient is younger than 8 years old or older than 80 years old, the system's validation filter will provide an *omission* if the patient's pregnancy status is set to true.
Breastfeeding (lactation) status	A patient's lactation status shall always be provided for a request to be valid. Default is *false*. Please note, if the patient is younger than 8 years old or older than 80 years old, the system's validation filter will provide an *omission* if the patient's lactation status is set to true.

Current medications (in medication list)	A patient's current medications (drugs in the medication list) shall be provided with drugMedications[]. "drugMedications[]" shall always be included for requests to be valid, both for conducting medication reviews, but also new prescriptions. This means that requests for both /evaluate and for /evaluateDrugMedications shall include "drugMedications[]". *See below.*

drugMedication[]	A drugMedication consist of the following;
	• A drug id
	• A prescription startDate and prescription endDate
	• A paused period (leave empty for no paused period)
	• An indication with code and text
	• A dosage that consist of
	  • Iteration interval, 1 means daily and 7 means weekly (one per seven days)
	  • A dosage startDate and dosage endDate
	  • DosageUnitType, NB: See section 4.6.1.1 DosageUnitType
	  • PN, true/false - when true, the drugMedication is excluded from decision support in module 5 - Double prescription.
	  • DosageDays, the days in a week where the drug shall be taken, i.e. 7 means sunday.
	  • Doses for that specific day
	  • Quantity of the medication, NB: be aware of the dosageUnitType, See section 4.6.1.1 DosageUnitType.
	  • TimeOfDay (morning, noon, evening, night)
	  • A dosage startDate and dosage endDate
	  • PN true/false. Module R5 will filter out PN medications.
	• Route of administration (ROA)
	• Unique local reference, one per "drugMedications"
	  • Paused periods, dosages and local reference

New medications (prescription)	New medications (new prescriptions or re-issue of prescriptions) shall be provided with drugMedicationsToAdd[]. "drugMedicationsToAdd[]" shall only be included for prescriptions, i.e. /evaluateDrugMedications. For input in a drug prescription, please see the drugMedications[] above.

Current unregistered medications (in medication list)	Unregistered drugs, also known by magistral medicines, are drugs that are not on the Danish market, but approved for use for the specific patient and no other, approved, treatment is available. Such medicines have no drug id. The decision support system is still able to provide decision support based on the ATC-code of the drug. Modules R2 - Cross-reactivity and R5 - Double prescription can provide decision support with the ATC-code. Requests to /evaluate (conduction of a medication review) and / evaluateDrugMedications (prescription) must have a customDrugs[]. The list can be left empty where there is no custom drugs in the patient's medication list. For the system to convert the drug to a customDrug, the customDrugInformation field must be filled out. See the API documentation for more information. A customDrug consist of the following; • An identifier • Name, used in decision support messages • Form with type "CODE" and a code; "TAB" for tablets • Strength with type "CODE" and a unitCode, for example "MG" for milligrams, value; amount of mg • A startDate and endDate • A paused period, if the custom drug prescription is paused • Route of administration (ROA) • PN, if PN the custom drug will be excluded from decision support in module 5 - Double prescription • ATC-code, necessary for providing decision support in modules 2 and 5, which can base decision support on ATC-code • A unique local reference per custom drug Please note; when a drug id provided in drugMedications[] is unknown, the drug will be converted to a customDrug and the user will still receive decision support if modules 2 and 5 are enabled.
Any new unregistered medications (prescription)	Requests to /evaluateDrugMedications (prescription) must have a customDrugsToAdd[]. The structure is identical to customDrugs[] above. The list can be left empty where no custom drug is prescribed. Please note; when a drug id provided in drugMedications[] is unknown, the drug will be converted to a customDrug and the user will still receive decision support if modules 2 and 5 are enabled. For the system to convert the drug to a customDrug, the customDrugInformation field must be filled out. See the API documentation for more information.

2.7.1.1 DosageUnitType

The DosageUnitType consists of one of the following;

- DosageUnitCountType
  - type
- DosageUnitActiveSubstanceType:
  - type
  - unitText
- DosageUnitAmountProductType:
  - type
  - unitText

Please refer to the API documentation site for further specification of the schema, DosageUnitType.

DosageUnitType conversion service

A conversion service that converts drug units to ATAH specific dosage unit types is provided with the device. Please visit the API documentation site for more information.

2.7.2 Enabling/disabling decision support modules

It's possible to configure which modules to enable or disable. To configure which modules to enable, the client system should:

- Include a comma-separated list of modules (R1,R2...) in the *header;*
  - 'xrequest-atah-usedecisionsupportmodulesheader'
  - Please refer to section *2.4) Request headers*
- Only the specified modules will use the request data and return a response
- If no comma-separated list is provided, all modules except module R8 will be enabled
- Enabling/disabling may be with client system level, system administrator level or user level
- If user level; the user interface of the client system shall include options to enable/disable modules
- If message severities can be enabled/disabled, please note that *contraindications* may not be disabled

It is recommended to enable all modules. Consequences of disabling modules are specified in section 2.7.3 below.

2.7.3 Consequences of Disabling Modules

Consequences of disabling decision support modules;

- R1: Potential drug-drug interactions will not be detected
- R2: Patient allergies, in relation to prescriptions, will not be detected
- R3: Potential risks of *overdose* will not be detected
- R4: Contraindications/cautions, due to a patient's GFR, will not be detected
- R5: Duplicate prescriptions, or inappropriate combinations, will not be detected
- R6: Medications inappropriate for children will not be detected
- R7: Medications inappropriate for a specific diagnosis will not detected
- R8: Medications without an indication will not be detected
- R9: Medications inappropriate to pregnant women will not be detected
- R10: Medications inappropriate to lactating (breastfeeding) women will not be detected
- R11: Antibiotic guidance will not be provided
- R12: Monitoring relevant to reduced kidney function, children or a specific diagnosis will not be detected

3 Test before production

Please use the URL provided for the *exttest* environment before production. This ensures that the implementation, and any issues that may occur, will not affect any other users using the production environment.

4 Contact Information

For further assistance or inquiries related to the API integration, please contact:

- Support E-mail: atah-prodsupport@trifork.com

More information can be found on page: https://wiki.kliniskbeslutningsstøtte.dk

5 Decommissioning

In case of decomissioning (removing the product from the market), client systems will be contacted directly by Region North, e.g. when a specific major version will no longer be supported.

Two possible ways of decomissioning:

- The environment of a major version is shut down, which
    - Will result in error messages in the client system
    - Should not be visible to users, however messages will no longer be provided *or*
- The client system removes the version by removing / changing the URL
    - Remove implementation if no version shall be used
    - Change URL (API endpoint) to new major version
        - NB: Use implementation guideline in case of new major version, as the request/response structure can impact implementation

When a new major version is to be implemented, and and old version is to be decommissioned, information hereof will be provided beforehand

Two major versions are supported in production. Previous major versions will not be decommissioned until all client systems have implemented the new major version