

UFTP messages for flex trading with Capacity Limit Contracts at GOPACS

GOPACS supports flex trading with Capacity Limit Contracts (*capaciteitsbeperkende contracten* or *lange termijncontracten*) through the exchange of Shapeshifter UFTP messages.

The Trading Company has to build an interface to send messages through API to the Grid operator and an endpoint to receive messages through API from the Grid operator.

The implementation must be compliant to the UFTP-specification:

<https://github.com/shapeshifter/shapeshifter-specification>. Currently GOPACS supports version 3.0.0 of the specification with some restrictions which are described separately in this document.

To aid in building a compliant implementation GOPACS has built a open source library.

- GOPACS library <https://github.com/shapeshifter/shapeshifter-library> (when GOPACS will release the 1.0 version, use the Main version)
- Main library <https://github.com/shapeshifter/shapeshifter-library>

When implementing the protocol, make sure the implementation is compliant to the GOPACS specifications as described in this document. All CLC traffic must be uniform in order to make sure that every participant is able to communicate with each other.

GOPACS uses (in this documentation) the term Trading Company for every participant that has a contract with the Grid Company. In the UFTP specification the term AGR (aggregator) is used, and in some Capacity Limit Contracts the term CSP is used.

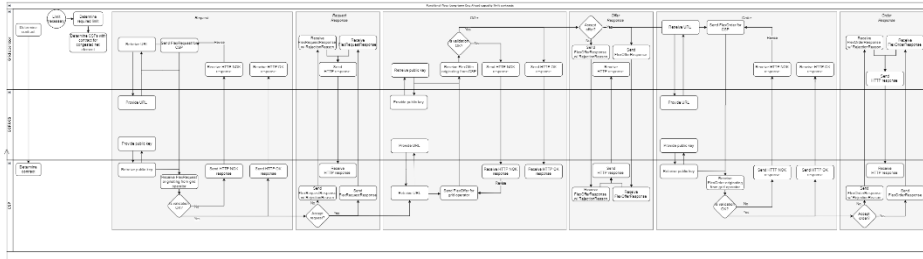
Table of contents

- [Usage UFTP protocol for CLC](#)
- [GOPACS environments](#)
- [Participant API \('Address book'\)](#)
 - [Response statuses](#)
- [GOPACS on behalf of a Grid Company or Trading Company](#)
- [SignedMessage](#)
 - [Response statuses](#)
- [FlexRequest](#)
 - [FlexRequestResponse](#)
- [FlexOffer](#)
- [FlexOrder](#)
 - [FlexOrderResponse](#)
- [Other message types](#)
- [GOPACS implementation specifics](#)

Usage UFTP protocol for CLC

Overview of the GOPACS and UFTP message exchange:

The part where the URL and the public key is retrieved, is GOPACS/CLC specific.



GOPACS environments

Environment	TST (nightly cleanup)	ACC (suggested for testing)	PRD
GOPACS UI	https://test.idcons.nl	https://acc.idcons.nl	https://idcons.nl
UFTP endpoint	https://capacity-limit-endpoint.tst.gopacs-services.eu/shapeshifter/api/v3/message	https://capacity-limit-endpoint.acc.gopacs-services.eu/shapeshifter/api/v3/message	https://capacity-limit-endpoint.gopacs-services.eu/shapeshifter/api/v3/message
Participant API endpoint ('Address book')	https://capacity-limit-contracts.tst.gopacs-services.eu/participant/{role}/{domain}/information	https://capacity-limit-contracts.acc.gopacs-services.eu/public-api/1.0/participant/{role}/{domain}	https://capacity-limit-contracts.gopacs-services.eu/public-api/1.0/participant/{role}/{domain}
API documentation	https://capacity-limit-contracts.tst.gopacs-services.eu/swagger-ui/index.html Address book	https://capacity-limit-contracts.acc.gopacs-services.eu/swagger-ui/index.html Address book	https://capacity-limit-contracts.gopacs-services.eu/swagger-ui/index.html Address book
	https://capacity-limit-endpoint.tst.gopacs-services.eu/swagger-ui/index.html UFTP endpoint	https://capacity-limit-endpoint.acc.gopacs-services.eu/swagger-ui/index.html UFTP endpoint	https://capacity-limit-endpoint.gopacs-services.eu/swagger-ui/index.html UFTP endpoint
UFTP Public Key (used for all outgoing messages from UI)	y6BlELL9Kria8Nm9zRJyJhaocTt7dtG2taU5szXysg=	VFHpQ4B71gOKrVJAG+HK1zQctr1J3zjkK4BYGK79E+c=	pI/s07d9fvoaty05gVLZKLfdqQmdkaMAIgcConmcl76U=

Participant API ('Address book')

DNS discovery as described in the UFTP specification is currently not supported by GOPACS.

GOPACS provides an alternative API for discovery of UFTP participant information: the Participant API.

Example Participant API request:

```
GET /public-api/1.0/participant/DSO/example.com HTTP/1.1
Accept: application/json
```

Example Participant API response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "domain": "example.com",
  "publicKey": "VFHpQ4B71gOKrVJAG+HK1zQctr1J3zjkK4BYGK79E+c=",
  "endpoint": "https://uftp.example.com/shapeshifter/v3/message"
}
```

Response statuses

HTTP status	Possible cause
200	
404	Participant not found for role + domain.

GOPACS on behalf of a Grid Company or Trading Company

If a Grid Company or Trading Company doesn't have their own UFTP API, they can use the GOPACS platform (<https://idcons.nl/>) to send and receive messages on their behalf. In this case, the UFTP messages are delivered to a GOPACS endpoint and GOPACS responds on behalf of that participant.

If GOPACS sends a message on behalf of a participant, the SenderDomain is always the UFTP domain of the actual (original) Grid Company or Trading Company, not GOPACS.

SignedMessage

Example of a SignedMessage HTTP request (some headers omitted for clarity):

```
POST /shapeshifter/api/v3/message HTTP/1.1
Content-Type: text/xml
```

```
<SignedMessage
  SenderDomain="dso.nl"
  SenderRole="DSO"
  Body="..." />
```

Attribute	GOPACS expectation
-----------	--------------------

SenderDomain	<p>Grid Company must log in to the GOPACS UI and configure UFTP domain, public key and endpoint in GOPACS, prior to receiving or sending messages.</p> <p>When GOPACS is sending messages on behalf of Grid Company or Trading Company, then the SenderDomain is equal to the domain of actual UFTP participant.</p>
SenderRole	<p>Must be <code>DSO</code> for a Grid Company.</p> <p>Must be <code>AGR</code> for a Trading Company.</p> <p>Other values are not supported.</p> <p>GOPACS is not considered a 'party' in the UFTP exchange and therefore does not have a 'role'.</p> <p>When GOPACS is sending messages on behalf of Grid Company or Trading Company, then the SenderRole is equal to the role of actual UFTP participant.</p>
Body	<p>Base64 encoded payload that will be decrypted using <code>crypto_sign_open</code> (see https://libsodium.gitbook.io/doc/public-key_cryptography/public-key_signatures)</p> <p>GOPACS uses Lazysodium which is a Java wrapper over the Libsodium library.</p> <p>GOPACS uses the SenderDomain and SenderRole to lookup the public key that is used for unsealing the message.</p>

Response statuses

HTTP status	Possible cause
200	Message was unsealed and XSD valid and will be processed asynchronously.
400	<p>Wrong content type</p> <p>Technical XSD validation error (no functional validations yet)</p> <p>Error during XML deserialization</p>
401	<p>Public key of sender not found or incorrect.</p> <p>Message could not be unsealed with public key of sender.</p>
5xx	Unexpected (temporary) error on GOPACS side.

FlexRequest

Sent by the Grid Company to the Trading Company.

Example FlexRequest:

```
<FlexRequest
  Version="3.0.0"
  SenderDomain="dso.nl"
  RecipientDomain="agr.nl"
  TimeStamp="2021-10-29T06:54:26.861Z"
  MessageID="d3ae4836-55b1-4084-b54e-34107b22648c"
  ConversationID="48cdc3d2-56c0-436c-8d5a-6f6cc3dc538d"
  ISP-Duration="PT15M"
  TimeZone="Europe/Amsterdam"
  Period="2021-10-30"
  ContractID="A-AA-A-12345"
  CongestionPoint="ean.265987182507322951"
  Revision="1"
  ExpirationDateTime="2021-10-29T22:15:00.0000Z">
  <ISP Disposition="Requested" MinPower="0" MaxPower="50000000" Start="48"
Duration="1"/>
  <ISP Disposition="Requested" MinPower="0" MaxPower="50000000" Start="49"
Duration="1"/>
  <ISP Disposition="Requested" MinPower="0" MaxPower="50000000" Start="50"
Duration="1"/>
  <ISP Disposition="Requested" MinPower="0" MaxPower="50000000" Start="51"
Duration="1"/>
</FlexRequest>
```

All the validations from the UFTP specification apply. On top of that GOPACS has some additional restrictions with respect to the usage of UFTP message for CLC:

Attribute/Element	GOPACS additional restrictions
Version	Must be 3.0.0 (currently).
SenderDomain	Grid Company must log in to the GOPACS UI and configure UFTP domain, public key and endpoint in GOPACS, prior to receiving or sending messages.
RecipientDomain	Trading Company must log in to the GOPACS UI and configure UFTP domain, public key and endpoint in GOPACS, prior to receiving or sending messages.

TimeStamp	<p>Parsing supports different offsets and handles accordingly. The offset can be either in “+HH:mm:ss” format or “Z”.</p> <p>GOPACS ignores the milliseconds part when parsing.</p> <p>GOPACS always sends either a UTC timestamp (no offset and “Z” suffix) or a timestamp in the Europe/Amsterdam timezone (offset +01:00 or +02:00 depending on DST).</p> <p>The milliseconds part can be between 0 and 9 digits where the omitted digits are implied to be zero.</p>
ISP-Duration	Required. Must be PT15M
TimeZone	Required. Must be Europe/Amsterdam
Period	<p>Required. ⚠️ The day of congestion. Format: YYYY-MM-DD always interpreted in Europe/Amsterdam time zone. <u>Any offset is ignored.</u></p> <p>The message must be sent before 12:00:00 the day before Period.</p> <p>Examples:</p> <ul style="list-style-type: none"> • If the message is received before 12:00:00, then the Period may be tomorrow or later. • If the message is received after 12:00:00, then the Period must be the day after tomorrow or later.
ExpirationDateTime	⚠️ The expiration date time must be no later than 12:00:00 the day before the day of congestion (Period).
ContractID	<p>Functional required. Typical format: A-AA-A-12345</p> <p>Trading Company must log in to the GOPACS UI and register the CLC contract, prior to receiving or sending messages.</p>
CongestionPoint	<p>ean. [0-9]{18}</p> <p>Must be a known EAN of a preregistered CLC contract in GOPACS.</p> <p>Does <u>not</u> have to be known as Grid Connection in GOPACS.</p>
ServiceType	Optional. Ignored.
ISP	⚠️ GOPACS currently only expects the ISPs that are to be limited under the contract.

<p>ISP.Start</p> <p>ISP.Duration</p>	<p>First ISP of the day is 1. 00:00:00 (inclusive) until 00:15:00 (exclusive)</p> <p>Second ISP of the day is 2. 00:15:00 (inclusive) until 00:30:00 (exclusive)</p> <p>Last ISP of the day is 96 or 100 or 92. 23:45:00 (inclusive) until 00:00:00 the next day (exclusive)</p> <p>ISPs with respect to Daylight Saving Time (DST), assuming Europe/Amsterdam and a 15 minute ISP duration:</p> <ul style="list-style-type: none"> • On the last Sunday of March when the clock goes from CET (standard) to CEST (summer), the number of ISPs will be 92: ISP 1: 00:00-00:15 ISP 2: 00:15-00:30 ISP 3: 00:30-00:45 ISP 4: 00:45-01:00 ISP 5: 01:00-01:15 ISP 6: 01:15-01:30 ISP 7: 01:30-01:45 ISP 8: 01:45-03:00 ISP 9: 03:00-03:15 etc. ISP 92: 23:45-00:00 • On the last Sunday of October when the clock goes from CEST (summer) to CET (standard), the number of ISPs will be 100. ISP 1: 00:00-00:15 ISP 2: 00:15-00:30 ISP 3: 00:30-00:45 ISP 4: 00:45-01:00 ISP 5: 01:00-01:15 ISP 6: 01:15-01:30 ISP 7: 01:30-01:45 ISP 8: 01:45-02:00 ISP 9: 02:00-02:15 ISP 10: 02:15-02:30 ISP 11: 02:30-02:45 ISP 12: 02:45-02:00 ISP 13: 02:00-02:15 etc. ISP 100: 23:45-00:00 • On any other day, the number of ISPs will be 96.
--------------------------------------	---

ISP.Disposition	<p>Must be Requested</p> <p>GOPACS expects only the ISPs that are to be limited.</p>
ISP.MinPower ISP.MaxPower	<p>⚠ GOPACS currently expects power in Watts as per the Shapeshifter specification 3.0.0.</p> <p>Must be 0 (see possibilities below), or ≤ -100000 or ≥ 100000 (at least ± 0.1 MW).</p> <p>Must be a multiple of 100000 (specify power in increments of 0.1 MW).</p> <p>A positive power number means consumption. And negative power number means production.</p> <p>⚠ Only supported possibilities at GOPACS:</p> <ul style="list-style-type: none"> • $\text{MinPower}=0$ and $\text{MaxPower} \geq 100000$: means Trading Company is requested to limit power consumption to MaxPower • $\text{MinPower} \leq -100000$ and $\text{MaxPower}=0$: means Trading Company is requested to limit power production to $-\text{MinPower}$

FlexRequestResponse

Example FlexRequestResponse:

```
<FlexRequestResponse
  Version="3.0.0"
  SenderDomain="agr.nl"
  RecipientDomain="dso.nl"
  Timestamp="2021-10-29T06:54:36.4437962Z"
  MessageID="7f0f4e68-f842-4b92-911e-b26f85525067"
  ConversationID="48cdc3d2-56c0-436c-8d5a-6f6cc3dc538d"
  Result="Accepted"
  FlexRequestMessageID="d3ae4836-55b1-4084-b54e-34107b22648c"/>
```

Or when it is rejected:

```
<FlexRequestResponse
  Version="3.0.0"
  SenderDomain="aggregator.org"
  RecipientDomain="uftp.dso.nl"
  Timestamp="2021-10-29T06:54:36.4437962Z"
  MessageID="7f0f4e68-f842-4b92-911e-b26f85525067"
  ConversationID="48cdc3d2-56c0-436c-8d5a-6f6cc3dc538d"
  Result="Rejected"
  RejectionReason="Reference Period mismatch"
  FlexRequestMessageID="d3ae4836-55b1-4084-b54e-34107b22648c"/>
```


FlexOffer

Sent by the Trading Company to the Grid Company as answer to a FlexRequest.

Example FlexOffer:

```
<FlexOffer
  SenderDomain="agr.nl"
  RecipientDomain="dso.nl"
  TimeStamp="2021-10-29T06:54:36.8868538Z"
  MessageID="338ed243-5517-4400-962e-2b7b812c468c"
  ConversationID="48cdc3d2-56c0-436c-8d5a-6f6cc3dc538d"
  ISP-Duration="PT15M"
  TimeZone="Europe/Amsterdam"
  Period="2021-10-30"
  CongestionPoint="ean.265987182507322951"
  ExpirationDateTime="2021-10-29T10:30:00Z"
  FlexRequestMessageID="d3ae4836-55b1-4084-b54e-34107b22648c"
  ContractID="A-AA-A-12345"
  BaselineReference=""
  Currency="EUR">
  <OfferOption OptionReference="ba40a5f8-849b-4fe6-958f-e628a1653558"
    Price="0">
    <ISP Power="50000000" Start="58"/>
    <ISP Power="50000000" Start="59"/>
    <ISP Power="50000000" Start="60"/>
    <ISP Power="50000000" Start="61"/>
  </OfferOption>
</FlexOffer>
```

⚠️ Unsolicited FlexOffer messages are rejected by GOPACS. There must always be a preceding FlexRequest.

⚠️ At most 1 FlexOffer message may be sent as part of a conversation. All successive FlexOffer messages will be rejected by GOPACS.

Attribute/Element	
D-PrognosisMessageID	Ignored.
BaselineReference	Ignored.
Currency	Must be EUR.
Price	Ignored.
OfferOption	⚠️ Exactly 1 OfferOption element is expected.
OfferOption.MinActivationFactor	Optional. Ignored.

ISP.Power	<p> Same rules apply as ISPs in FlexRequest.</p> <p>It is allowed to send an offer on a subset of the requested ISPs.</p>
-----------	---

Other attributes like Period, CongestionPoint, ContractID, etc. must be equal to the original FlexRequest.

FlexOrder

Example FlexOrder:





```
<FlexOrder
  Version="3.0.0"
  SenderDomain="dso.nl"
  RecipientDomain="agr.nl"
  Timestamp="2021-10-29T06:55:36.518Z"
  MessageID="dc0f19c4-3835-4753-8f0c-0319d6642fbb"
  ConversationID="48cdc3d2-56c0-436c-8d5a-6f6cc3dc538d"
  ISP-Duration="PT15M"
  TimeZone="Europe/Amsterdam"
  Period="2021-10-30"
  CongestionPoint="ean.265987182507322951"
  FlexOfferMessageID="338ed243-5517-4400-962e-2b7b812c468c"
  ContractID="A-AA-A-12345"
  Price="0"
  Currency="EUR"
  OrderReference="None">
  <ISP Power="50000000" Start="1" Duration="1"/>
  <ISP Power="50000000" Start="2" Duration="1"/>
  <ISP Power="50000000" Start="3" Duration="1"/>
  <ISP Power="50000000" Start="4" Duration="1"/>
</FlexOrder>
```

Attribute/Element	
Currency	Must be EUR.
Price	Ignored.
OrderReference	GOPACS always generates a unique GUID as order reference.
ISP	Currently GOPACS orders exactly what was offered (if on behalf of a Grid Company) - including ISPs and min activation factor.

FlexOrderResponse


An “Accepted” response from the Trading Company means that there is a binding agreement.


Other message types


-  FlexOfferRevocation not supported yet
-  FlexRequest revisions not supported yet
-  Test messages are not supported yet
-  Other message types are not supported yet


GOPACS implementation specifics

 A duplicate MessageID is immediately responded to with a 400 Bad Request and **not** a 200 OK followed by “Rejected” response as described in the specification!

 After a 200 OK is returned, a received message is immediately processed by GOPACS. An accepted or rejected response is sent back almost instantaneously.

 The user receives realtime email notifications when a FlexRequest, FlexOffer or FlexOrder is received, rejected or failed to deliver.

 An outgoing UFTP message is retried every 3 minutes for a maximum of 5 tries. After that, the user and GOPACS DevOps team are notified of a failure to deliver. Specifically on a 400 Bad Request, a message is not retried.

 Typically there will be at most 15 mins between FlexRequest and FlexOrder.

 TODO: Handling of daylight savings with respect to ISPs.