

Trusted Contract Observation

Service Description

Service ID: *"trusted-contract-observation"*

Abstract

This document describes a service useful for getting copies of negotiation sessions, as managed by the *Trusted Contract Negotiation* service.



ARTEMIS Innovation Pilot Project: Arrowhead
THEME [SP1-JTI-ARTEMIS-2012-AIPP4 SP1-JTI-ARTEMIS-2012-AIPP6]
[Production and Energy System Automation Intelligent-Built environment and urban infrastructure for sustainable and friendly cities]



ARROWHEAD

Document title
Trusted Contract Observation
Date
2020-06-05

Version
0.2
Status
DRAFT
Page
2 (8)

Contents

1 Overview	3
2 Service Interface	4
2.1 function GetNegotiationByNamesAndId	4
3 Information Model	5
3.1 struct Argument	5
3.2 struct Contract	5
3.3 struct Offer	6
3.4 struct Negotiation	6
3.5 struct NegotiationRequest	6
3.6 enum NegotiationStatus	6
3.7 Primitives	7
4 Revision History	8
4.1 Amendments	8
4.2 Quality Assurance	8



ARROWHEAD

Document title
Trusted Contract Observation
Date
2020-06-05

Version
0.2
Status
DRAFT
Page
3 (8)

1 Overview

This document describes the Trusted Contract Observation (TrCoOb) service, which allows for arbitrary Eclipse Arrowhead systems to obtain copies of negotiation sessions, as they are maintained by systems provided by the Trusted Contract Negotiation (TrCoNe) service.

Readers of this document are assumed to be familiar with the TrCoNe and Contract Negotiation services. For more information about those services, please refer to their respective service description documents.

The rest of this document is organized as follows. In Section 2, we describe the abstract message function provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned function.



ARROWHEAD

Document title
Trusted Contract Observation
Date
2020-06-05

Version
0.2
Status
DRAFT
Page
4 (8)

2 Service Interface

This section describes the one function that must be exposed by TrCoOb services. In particular, the below subsection names an abstract function, an input type and an output type, in that order. The input type is named inside parentheses, while the output type is preceded by a colon. All abstract data types named in this section are defined in Section 3.

2.1 function **GetNegotiationByNamesAndId** (**NegotiationRequest**) : **Negotiation**

Called to acquire the candidate of some identified negotiation session, as well as some additional details.

3 Information Model

Here, all data objects that can be part of TrCoOb service calls are listed in alphabetic order. Note that each subsection, which describes one type of object, begins either of the *struct* or *enum* keywords, which are used to denote a collection of named fields, each with its own data type, or an enumeration over a set of variants, respectively. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.7, which are used to represent things like hashes and identifiers.

An overview of the data object types is illustrated in Figure 1.

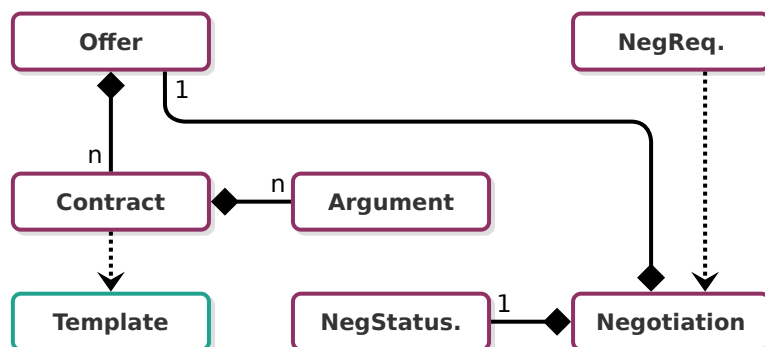


Figure 1: Information model as a UML class diagram. Black rhombuses signify data ownership, solid lines with triangle arrows are used to signify that the type pointed from is a variant of the target, while dotted lines with arrows denote references. The type closest to each rhombus is the one owning objects of the connected type. Numbers are used to specify how many objects are owned, where n is to be regarded as an arbitrary positive number or zero. Note that *Template* is only indirectly part of the information model, but is included for the sake of completeness.

3.1 struct **Argument**

A concrete **Contract** parameter value. In other words, this data structure identifies a variable parameter in a template and provides a concrete parameter value. The type of the *Value* field should, in a non-naive implementation, be dynamic enough to be able to take on at least a few concrete types, such as integer, float, string, list, associative array, and so on. Conceptually, the parameter specification in the template that matches the *Name* field could contain additional type information, such as range restrictions, required list lengths, or regular expressions.

Field	Type	Description
name	Name	Human and machine-readable name of parameter.
value	Custom	Concrete value.

3.2 struct **Contract**

A parameterized reference to a *template*, representing certain rights and obligations accepted by two parties. In other words, a contract derives its meaning from both the template it refers to and the arguments it contains. Note that contract objects only become binding if they refer correctly to legally valid templates and become part of **Acceptances**.

Field	Type	Description
templateName	Name	Name of invoked <i>template</i> .
arguments	List< Argument >	Arguments matching <i>parameters</i> defined in <i>template</i> .

3.3 struct Offer

A concrete offer of **Contracts** intended for a specific receiver. Note that in contrast to the CoNe service, Names are used instead of hashes to represent the offeror and receiver.

Field	Type	Description
offerorName	Name	Offeror's common name.
receiverName	Name	Receiver's common name.
validAfter	DateTime	The time after which the offer may be accepted.
validUntil	DateTime	The time after which the offer may no longer be accepted.
contracts	List< Contract >	Offered contracts.
offeredAt	DateTime	The time at which this offer was made.

3.4 struct Negotiation

Data extracted from a negotiation session.

Field	Type	Description
negotiationId	RandomID	Negotiation session identifier.
offer	Offer	Last offer to become part of the session.
status	NegotiationStatus	Enumerates the state of the negotiation session.

3.5 struct NegotiationRequest

Identifies a requested **Negotiation**.

Field	Type	Description
negotiationId	RandomID	Negotiation session identifier.
name1	Name	Common name of one party part of the desired negotiation session.
name2	Name	Common name of another party part of the desired negotiation session.

3.6 enum NegotiationStatus

Identifies the status of a **Negotiation**.

Variant	Description
OFFERING	The last session offer can currently be replaced, accepted or rejected.
ACCEPTED	The last session offer has been accepted and the session closed.
REJECTED	The last session offer has been rejected and the session closed.
EXPIRED	The negotiation session has expired and is closed.

3.7 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
Custom	Any suitable type chosen by the implementor of the service.
DateTime	Pinpoints a specific moment in time.
List<A>	An <i>array</i> of a known number of items, each having type A.
Name	A string identifier that is intended to be both human and machine-readable.
RandomID	An identifier produced using a cryptographically secure random function.



ARROWHEAD

Document title
Trusted Contract Observation
Date
2020-06-05

Version
0.2
Status
DRAFT
Page
8 (8)

4 Revision History

4.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1				

4.2 Quality Assurance

No.	Date	Version	Approved by
1			