

# FSC - External Contract Reference

## 1.0.0



Logius Standard

Consultation version January 19, 2026

**This version:**

<https://gitdocumentatie.logius.nl/publicatie/fsc/ext/1.0.0/>

**Latest published version:**

<https://gitdocumentatie.logius.nl/publicatie/fsc/ext/>

**Latest editor's draft:**

<https://logius-standaarden.github.io/fsc-external-contract/>

**Previous version:**

<https://gitdocumentatie.logius.nl/publicatie/fsc/ext//>

**Editor:**

Logius ([Logius](#))

**Authors:**

Ronald Koster (PhillyShell), [Email](#)

Niels Dequeker (ND Software), [Email](#)

Lonneke Dikmans (RINIS), [Email](#)

**Participate:**

[GitHub Logius-standaarden/fsc-external-contract](#)

[File an issue](#)

[Commit history](#)

[Pull requests](#)

This document is also available in these non-normative format: [PDF](#)



This document is licensed under

[Creative Commons Attribution 4.0 International Public License](#)

---

## Status of This Document

This is a proposed recommendation approved by TO. Comments regarding this document may be sent to [api@logius.nl](mailto:api@logius.nl)

# Table of Contents

## **Status of This Document**

## **Conformance**

## **Abstract**

## **1. Introduction**

- 1.1 Purpose
- 1.2 Overall Operation of External Contract Reference
- 1.3 Caveat
- 1.4 Terminology

## **2. Architecture**

- 2.1 Trust and Validation Model
  - 2.1.1 Trust Relationships
  - 2.1.2 Validation Requirements
- 2.2 Service Publication Delegation
  - 2.2.1 Publication Delegation Flow
  - 2.2.2 Publication Architecture Implications
- 2.3 Service Connection Delegation
  - 2.3.1 Connection Delegation Flow
- 2.4 Connection Architecture Implications
- 2.5 Property Lifecycle
- 2.6 Integration with FSC Core

## **3. Specification**

- 3.1 Properties for an External Contract Reference
  - 3.1.1 ServiceConnectionGrant
  - 3.1.2 ServicePublicationGrant
- 3.2 Transaction log
- 3.3 Access token
- 3.4 Inway
- 3.5 Outway

## **4. List of Figures**

## **A. References**

- A.1 Normative references

## § Conformance

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words *MUST* and *SHOULD* in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

## Abstract

External Contract Reference is an extension on the Federated Service Connectivity (FSC) standard [FSC Core](#). It can only be used in addition to the FSC Core specification. The purpose of the External Contract Reference extension is to allow delegation through contracts that are not defined by FSC. The External Contract Reference extension contains descriptions of the Properties that can be added to the ServiceConnectionGrant and the ServicePublicationGrant, and how these properties can be used by the FSC components.

## § 1. Introduction

The External Contract Reference specification is an extension of the Federated Service Connectivity (FSC) Core specification. This extension enables delegation of a Service publication and Service consumption through references to external contracts (contracts not defined by FSC).

### § 1.1 Purpose

In delegation as defined by FSC Core, the *delegator* organization *MUST* use a FSC Manager to create and sign Contracts. This extension addresses scenarios where the delegator lacks the technical capability to operate FSC infrastructure, allowing a *delegatee* (API provider or consumer) to act based on an external contract.

## § 1.2 Overall Operation of External Contract Reference

An API provider creates a Contract using a ServicePublicationGrant. In this Contract the API provider adds a reference to an existing (paper) contract using Properties. The Contract is received by the Peer acting as Directory. The Peer validates the reference manually or automatically, outside FSC with a custom component. The validation itself is out of scope for this extension. The Contract is accepted and the Service is published on behalf of the organization specified in the reference.

An API consumer creates a Contract with a ServiceConnectionGrant. In this Contract the API consumer adds a reference to an existing (paper) contract using Properties. The Contract is received by the API provider. The API provider validates the reference manually or automatically, outside FSC with a custom component. After validation the Contract is accepted and the API consumer can consume the Service on behalf of the organization specified in the reference.

## § 1.3 Caveat

This extension is developed as a temporary solution because external contracts defined in this extension cannot be validated by FSC components. This breaks the trust. In the future, this extension will be replaced by an extension that allows Delegators to sign contracts in a cryptographically secure manner without having to use an FSC Manager.

## § 1.4 Terminology

This section lists terms (#header) and abbreviations that are used in this document. This document assumes that the reader is familiar with the Terminology of FSC Core.

*contract:*

contract without a capital C is a contract that is not defined by FSC.

## § 2. Architecture

This chapter describes the architecture of an External Contract Reference.

## § 2.1 Trust and Validation Model

The key architectural principle is **trust transfer**: the delegatee asserts authority to act on behalf of the delegator, backed by an external contract reference. The receiving party (Directory or API provider) performs validation of this assertion before accepting the Contract.

### § 2.1.1 Trust Relationships

The External Contract Reference introduces a three-party trust model:

1. **Delegator** - The organization on whose behalf services are published or consumed
2. **Delegatee** - The organization (API provider or consumer) acting on behalf of the delegator
3. **Validator** - The organization (Directory or API provider) that validates the External Contract Reference

Trust is established through the external contract, which exists outside the FSC framework. The *external contract reference* serves as a pointer to evidence of this trust relationship.

### § 2.1.2 Validation Requirements

When a Contract containing an *External Contract Reference* is received, the receiving organization:

1. *MUST* verify that the external contract exists and is accessible via the provided reference
2. *MUST* verify that the external contract grants authority to the delegatee to act on behalf of the delegator
3. *MUST* verify that the delegator identity (delegator\_id and delegator\_name) matches the external contract
4. *SHOULD* verify that the scope of delegation in the external contract covers the specific Service being published or consumed

The validation process itself is **out of scope** for this extension and *MUST* be performed manually or automatically by a component outside of FSC.

If validation fails, the Contract *MUST* be rejected. If validation succeeds, the Contract follows the FSC Core Contract lifecycle rules.

## § 2.2 Service Publication Delegation

### § 2.2.1 Publication Delegation Flow

When an API provider publishes a Service on behalf of a delegator organization, the following architectural flow is executed:

Figure 1: Service Publication Delegation Flow

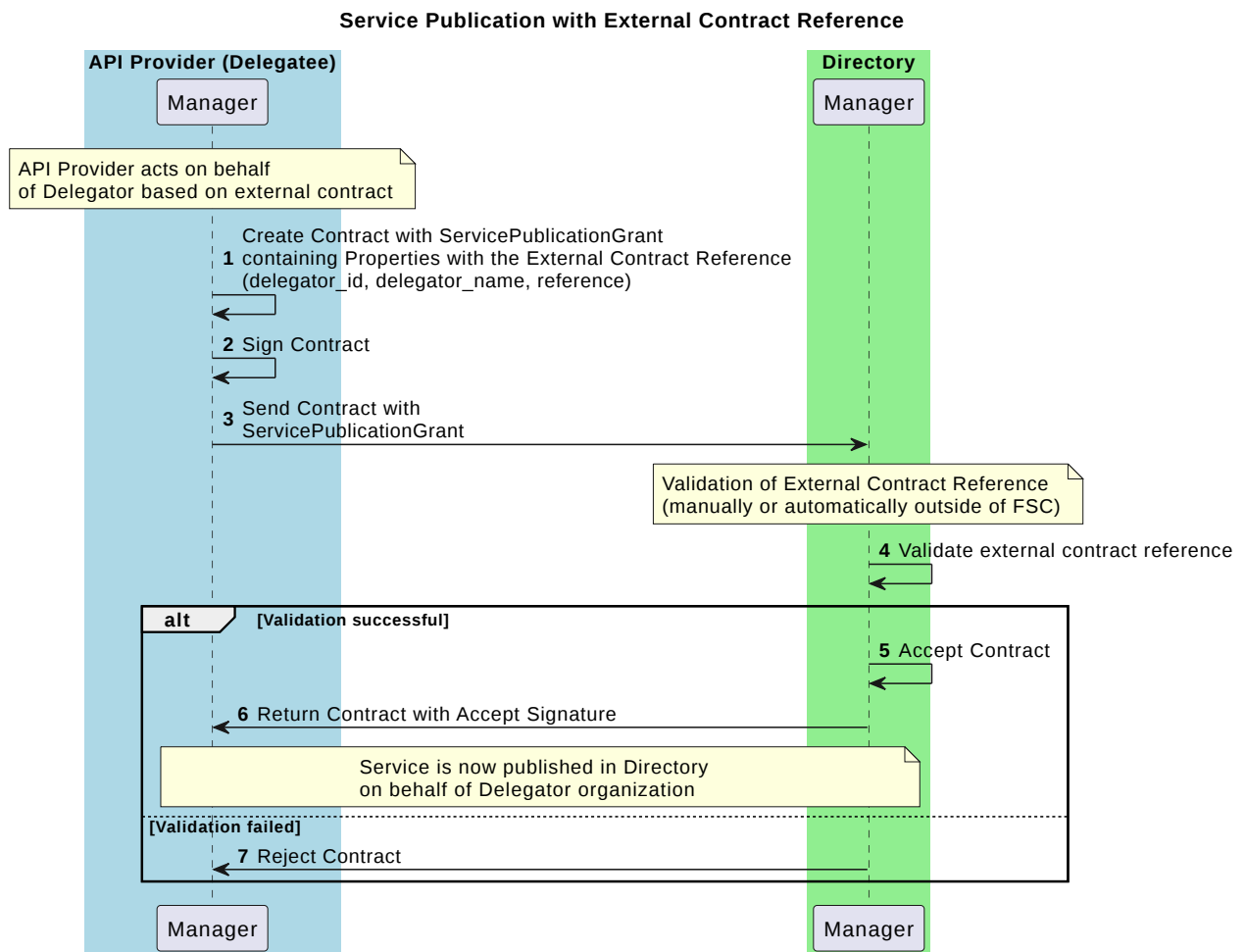


Figure 1 Figure 1: Service Publication Delegation Flow

#### Step-by-step process:

1. **Grant Creation:** The Manager of the API provider creates a `ServicePublicationGrant` containing the property `delegation.publication.external_contract_reference`. This object contains three fields:
  - `reference` - URL or identifier of the external contract

- `delegator_id` - Identity of the delegator organization
  - `delegator_name` - Name of the delegator organization
2. **Contract Transmission:** The Manager of the API provider creates a Contract containing the `ServicePublicationGrant` and transmits it to the Manager of the Directory.
  3. **Validation:** The Directory performs validation of the External Contract Reference according to the validation requirements in Section 2.2
  4. **Contract Acceptance or Rejection:**
    - If validation succeeds: The Directory accepts the Contract
    - If validation fails: The Directory rejects the Contract
  5. **Contract Completion:** Upon receiving the signed Contract, the API provider signs it, making the Contract valid according to FSC Core rules
  6. **Service Publication:** The Service is now published in the Directory on behalf of the delegator organization

## § 2.2.2 Publication Architecture Implications

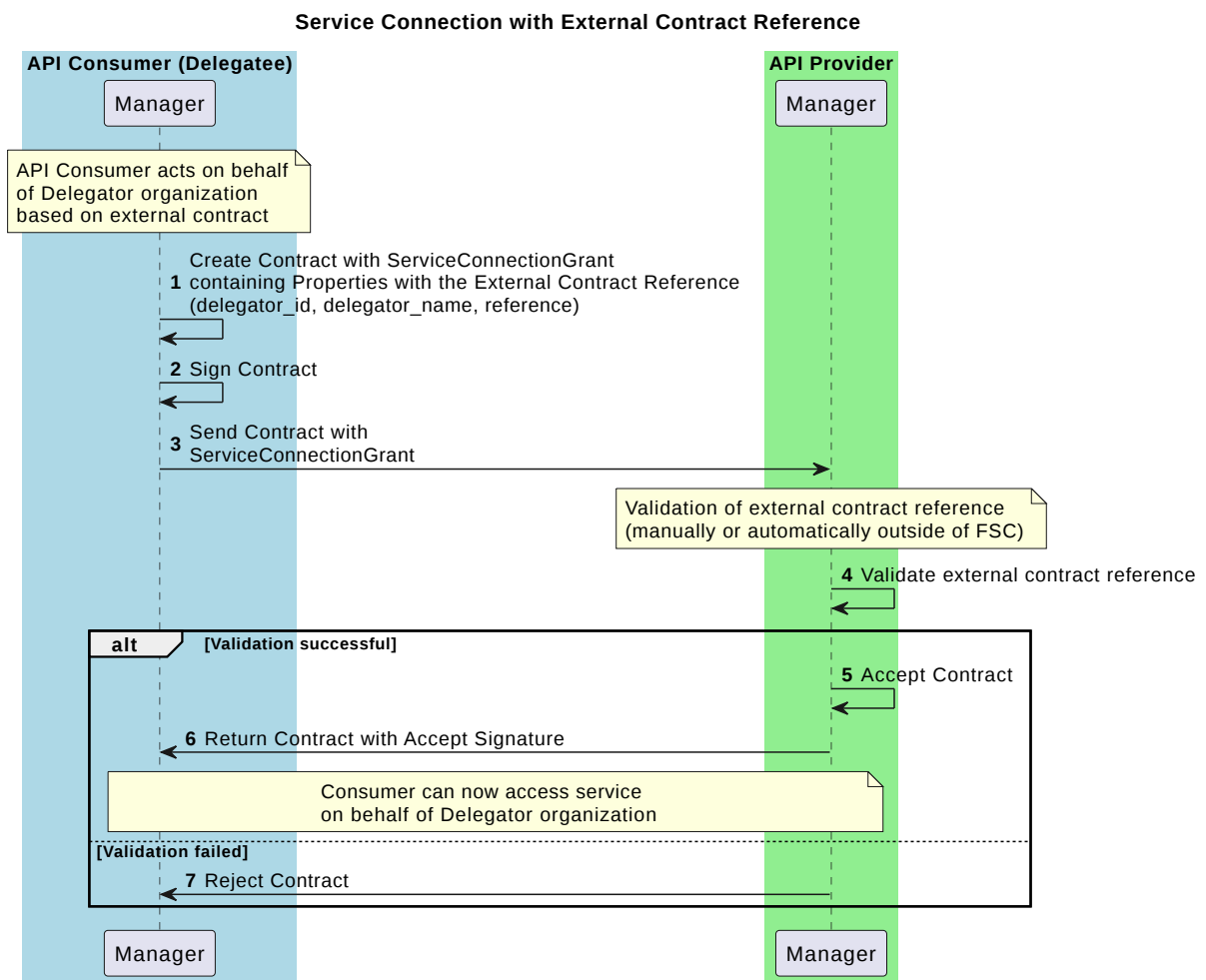
The Directory *MUST*:

- Include the content of `delegation.publication.external_contract_reference` in the Service discovery responses to indicate the true Service owner
- Maintain audit trails of validation decisions for external contract references

## § 2.3 Service Connection Delegation

### § 2.3.1 Connection Delegation Flow

When an API consumer consumes a Service on behalf of an organization, the following architectural flow is executed:



*Figure 2 Figure 2: Service Connection Delegation Flow*

### Step-by-step process:

- 1. Grant Creation:** The Manager of the API consumer creates a `ServiceConnectionGrant` containing the Property `delegation.connection.external_contract_reference`. This object contains three fields:
  - `reference` - URL or identifier of the external contract
  - `delegator_id` - Identity of the delegator organization
  - `delegator_name` - Name of the delegator organization
- 2. Contract Transmission:** The Manager of the API consumer creates a Contract containing the `ServiceConnectionGrant` and transmits it to the Manager of the API provider.
- 3. Validation:** The API provider performs validation of the External Contract Reference according to the validation requirements in Section 2.2
- 4. Contract Acceptance or Rejection:**



- If validation succeeds: The API provider signs the Contract and returns it to the API consumer
  - If validation fails: The API provider rejects the Contract
5. **Contract Completion:** Upon receiving the signed Contract, the API consumer signs it, making the Contract valid according to FSC Core rules
  6. **Service Access:** The API consumer can now access the Service on behalf of the delegator organization

## § 2.4 Connection Architecture Implications

API providers can use the `delegation.connection.external_contract_reference` added in the prp of the access token to:

- perform additional authorization
- offer functionality based on the identity of the delegator

## § 2.5 Property Lifecycle

The architectural flow of external contract reference properties through the system:

1. **Creation:** Properties defined in `ServicePublicationGrant` or `ServiceConnectionGrant`
2. **Contract Storage:** Properties stored with the Contract in the Manager
3. **Access Token Generation** (for service connections only): Properties included in prp claim when API consumer requests access
4. **Service Request:** Access token with properties sent to Inway of the API provider
5. **Transaction Logging:** Properties extracted and logged in transaction log as `additional_data`

This end-to-end propagation ensures complete traceability and auditability of delegated service access.

## § 2.6 Integration with FSC Core

This extension integrates with FSC Core as follows:

- **Section 3.2 (Contract Management):** Adds optional properties to `ServicePublicationGrant` and `ServiceConnectionGrant`
- **Section 3.6 (Delegation):** Provides an alternative delegation mechanism for organizations that cannot operate FSC Managers
- **Section 3.8 (Authorization):** External Contract Reference properties in access tokens enable fine-grained authorization decisions
- **Transaction Log Extension:** Defines how External Contract References are logged for auditability

## § 3. Specification

### § 3.1 Properties for an External Contract Reference

The External Contract Reference can be added to the `ServiceConnectionGrant` and `ServicePublicationGrant` using Properties. The Grants and Properties are both defined in [FSC Core](#).

#### § 3.1.1 ServiceConnectionGrant

- *delegation.connection.external\_contract\_reference(object)*: The object that contains the delegation information
- *reference(string)*:  
The reference to the contract. Must be a string. Example shown uses a URL, but any string value is valid
- *delegator\_id(string)*: The ID of the Delegator that has delegated the authorisation to publish the Service
- *delegator\_name(string)*: The name of the Delegator that has delegated the authorisation to publish the Service

All properties are required when using the `delegation.connection.external_contract_reference` object.

Example:

```
{
  "delegation.connection.external_contract_reference": {
    "reference": "https://linktocontract.nl/1",
    "delegator_id": "12345678901234567890",
    "delegator_name": "Example organisation"
  }
}
```

### § 3.1.2 ServicePublicationGrant

- *delegation.publication.external\_contract\_reference(object)*: The object that contains the delegation information
- *reference(string)*:  
The reference to the contract. Must be a string. Example shown uses a URL, but any string value is valid
- *delegator\_id(string)*: The ID of the Delegator that has delegated the authorisation to publish the Service
- *delegator\_name(string)*: The name of the Delegator that has delegated the authorisation to publish the Service

All properties are required when using the `delegation.publication.external_contract_reference` object.

Example:

```
{
  "delegation.publication.external_contract_reference": {
    "reference": "https://linktocontract.com/1",
    "delegator_id": "12345678901234567890",
    "delegator_name": "Example organisation"
  }
}
```

## § 3.2 Transaction log

In Groups that use the [Transaction log extension](#) the External Contract Reference properties **MUST** be stored as `additional_data` in the transaction log.

Example Transaction Log record:

```
{
  "transaction_id": "019a7dea-9048-7c4d-9e66-4d3f75a18f58",
  "direction": "DIRECTION_INCOMING",
  "grant_hash": "$1$4$+PQI7we01qIfEwq405UioLKzjGBgRva6F5+bUfDlKxUjcY5yX",
  "source": {
    "type": "SOURCE_TYPE_SOURCE",
    "outway_peer_id": "12345678901234567891"
  },
  "destination": {
    "type": "DESTINATION_TYPE_DESTINATION",
    "service_peer_id": "12345678901234567891"
  },
  "service_name": "random_service_name",
  "additional_data": {
    "delegation.connection.external_contract_reference": {
      "reference": "https://linktocontract.com/1",
      "delegator_id": "12345678901234567890",
      "delegator_name": "Example organization"
    }
  }
  "created_at": 1672527600
}
```

## § 3.3 Access token

The External Contract Reference properties are added to the Access Token. FSC Core defines that properties added to a Grant **MUST** be added the access token as the `prp` claim.

Example access token bases on a ServiceConnectionGrant with an External Contract Reference.

```
{
  "gid": "fsc.group.example.id",
  "gth": "$1$3$n4kellB6p2-VjrriscICkH3P5flgMmG6rX7MFIF3cU2Y20VysoU6W903",
  "sub": "12345678901234567890",
  "iss": "12345678901234567891",
  "svc": "test",
  "aud": "https://inway.organization-b.open-fsc.localhost:443",
  "exp": 1493726400,
  "nbf": 1493722800,
  "cnf": {
    "x5t#S256": "06QekfpQ2IkYWhXyZqz3T1llvPEqDYY00UyETSr7QdU"
  },
  "prp": {
    "delegation.connection.external_contract_reference": {
      "reference": "https://linktocontract.com/1",
      "delegator_id": "12345678901234567893",
      "delegator_name": "Example organization"
    }
  }
}
```

### § 3.4 Inway

The Inway **MUST** include External Contract Reference properties in the Transaction Log records as `additional_data` when the Logging extension is used.

### § 3.5 Outway

The Outway **MUST** include External Contract Reference properties in the Transaction Log records as `additional_data` when the Logging extension is used.

## § 4. List of Figures

[Figure 1 Figure 1: Service Publication Delegation Flow](#)

[Figure 2 Figure 2: Service Connection Delegation Flow](#)

## § A. References

### § A.1 Normative references

#### [RFC2119]

*Key words for use in RFCs to Indicate Requirement Levels*. S. Bradner. IETF. March 1997.  
Best Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc2119>

#### [RFC8174]

*Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. B. Leiba. IETF. May 2017.  
Best Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc8174>