Introduction

The contract concluded after the tender procedure must comply with the template defined by the Ministry of Finance of Ukraine, created, signed, processed and published online.

Responsibilities arising in the process of creation, completion, processing and publication of the contract in the tender are:

> The Ministry of Finance of Ukraine in the E-Contracting system describes and publishes contract templates that can be used in tendering procedures.
> ProZorro CDB receives the template description from the E-Contracting system and shares the description of the published templates to all marketplaces.

The marketplaces allow the Contracting Authority (hereinafter CA) to use the contract template when posting the tender, to create the instances of the contractProforma and later final contract for that tender and to fill it with the details of the created tender and CA.

Business process and workflow for e-Contracting developed by Quintagroup with cooperation with ProZorro

For the initial phase of the project, it is proposed to develop the functionality described below for use in electronic catalogues *(Direct order and price quotation procedures),* with further extension to all existing procedures in the ProZorro system.

To implement e-Contracting in ProZorro, the following API changes, functionality enhancements, and software development are proposed:

templateRegistryBot (trBot) and rendererBot (rBot) .

In order to implement the use of templates from the E-contracting system, it is envisaged to develop a **templateRegistryBot**, which will keep track of selected template and provide centralized communication with the template registry.

For rendering the document in a human-readable form, **rendererBot** development is envisaged, which will generate a PDF file from the template and JSON data for it.

Workflow

1. Register a draft **contractProforma** with a template ID to use.

2. **trBot** extracts the **Template** from the E-Contract Template Registry and uploads the document as a **contractTemplate**, linking to the draft **contractProforma**.
3. **trBot** extracts / generates a schema for / from the template and loads the **contractProforma** document as a **contractSchema**.
4. **trBot** loads the recommended **contractForm** document associated with the draft **contractProforma**
5. The marketplaces extract **contractSchema** and **contractForm**, render, receive and validate.
6. The marketplaces download **contractData** associated with the draft **contractProforma**
7. **rBot** generates **contractProforma** with **contractTemplate** + **contractData**, and loads as a new version of the **contractProforma** document.

Changes to the Tender object

The changes will affect the Tender object in **status: active.tendering** and **active.enquiries**, namely creating new document types and changing the behaviour of other fields.

Table 8 Changes to the Tender object

Component	Description	Type
documents: **type:** ● **contractTemplate**	Template from E-Contracting system	Word doc
documents: **type:** ● **contractSchema**	The data model for the template	JSON Schema

documents: **type:** ● **contractForm**	Form for receiving data on the schema, with official recommendations for filling	JSON Form
documents: **type:** ● **contractData**	Customer information on the schema from the corresponding **contractSchema**	JSON
documents: **type:** ● **contractProforma**	Generated contract document with already filled in CA details	PDF
documents: **documentOf: document**	New reference type (linked document).	
documents: **relatedItem: {id}**	Document ID of the documents collection. (Latest version)	

Changes to the Award object

The changes will affect the **Award** object in **status: pending** / **active**

Table 9 Changes to the Award object

Component	Description	Type

documents: **type:** ● **contractData**	Result / EO information	JSON
documentOf : document	New reference type (linked document).	

Changes to the Bid object

Table 10 Changes to the Bid object

Component	Description	Type
documents : **type :** ● **contractData**	EO information	JSON
documentOf : document	New reference type (linked document).	

Changes to the Contract object

The changes will affect the **Contract** object in **status : pending**

Table 11 Changes to the Contract object

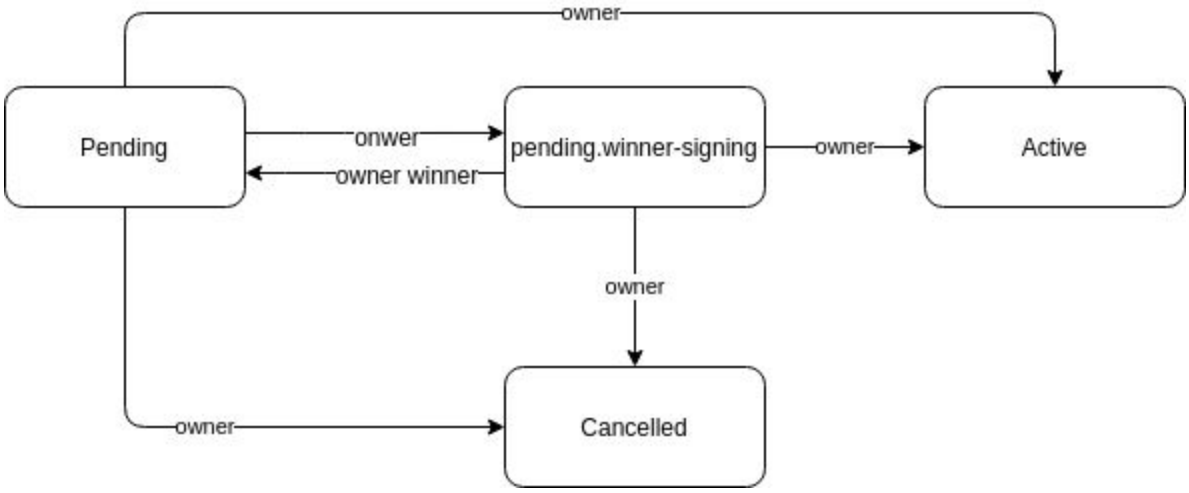
Component	Description	Type

documents : **type :** ● **contractData**	Consolidation of information from **Tender, Award, Bid** objects into a single JSON file	JSON
documents : **type :** ● **contract**	Generated final contract document with full JSON data set.	PDF

Process of signing the Contract

The contract signing process will take place in the Contract object during **status: pending** .

Figure 4 Signing process workflow



Status: Pending. Initial status after contract was awarded. Under this state users can perform following activities:

1. Sign the contract by CA (*owner*).
2. Send contract to EO (*owner winner*) for signing first, switch to status **pending.winner-signing** by CA**.**
3. Activate contract, switch to status **active** (*in case if EO already signed*) by CA.

4. Cancel contract, switch to status **cancelled** by CA.

Status: pending.winner-signing. Interim status for use when EO must to sign a contract first. Under this state users can perform following activities:

1. Sign the contract by EO.
2. Send signed contract to CA, switch to status **pending** by EO.
3. Activate contract, switch to status **active** (*in case if EO already signed*) by CA.
4. Cancel contract, switch to status **cancelled** by CA.

Status: Active. Indicates that the contract signed by CA and EO is published.

Status: Cancelled. Contract final status for cases when contract can't be signed. Initiate return to Qualification phase in procedure.

System support different workflows of contract signing, such as:

1. The generated contract is sent to the EO for signature by CA. After that it is signed by the CA and with all signatures is transferred into status: **active**.
2. The generated contract is signed by the CA. After that, it is sent to the EO for signature. With all signatures the CA switches into status: **active**.
3. The generated contract is sent to the EO for signature. If the contract is not signed within the expected period of time or if the contract is canceled, the CA may cancel the contract by transferring the contract to status: **cancelled**.

Once the contract is active, an **ASIC (Associated Signature Container)** contract container is generated and sent to the contract registry.

Contract container requirements can instruct e-signature to be added with one of the following options:

Table 12 ASIC (Associated Signature Container)

.p7s container with the Contract PDF embedded	.p7s signature linked to the Contract PDF file

document: **type:** contract Versions: **author:** contractBot **author:** winner **author:** owner **author:** owner	**document:** **type:** contractSignature **author:** winner **documentOf**: document **relatedItem:** {id}