OPEN CONTRACTING DATA STANDARD
IMPLEMENTATION METHODOLOGY

THE WORLD BANK
IBRD · IDA | WORLD BANK GROUP

SEPTEMBER 2016

E-PROCUREMENT TOOLKIT — ACCELERATING E-PROCUREMENT SOLUTIONS

OPEN CONTRACTING DATA STANDARD IMPLEMENTATION METHODOLOGY

September 2016

THE WORLD BANK
IBRD • IDA | WORLD BANK GROUP

This document was prepared for Rapid e-Procurement Toolkit Project of the World Bank Group by European Dynamics, www.eurodyn.com.

EUROPEAN DYNAMICS

SAFE
Strengthening Accountability
and the Fiduciary Environment

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs FDEA
State Secretariat for Economic Affairs SECO

Table of Contents

Table of Figures

Table of Tables

List of Acronyms

Acronym	Meaning
API	Application Programming Interface
GPA	Agreement on Government Procurement
LCS	Least Cost Selection
OCDS	Open Contracting Data Standard
OCID	Open Contracting ID
OCP	Open Contracting Partnership
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WB	World Bank

1 INTRODUCTION

The Open Contracting Data Standard (OCDS) is a core product of the Open Contracting Partnership (OCP). The "1.0 Release Candidate" of a data specification was developed by the World Wide Web Foundation and the World Bank through a project supported by the Omidyar Network and the World Bank. The OCP was then transformed into a separate organization in 2015.

The objective of the OCDS is to support governments' capacity to publish contracting data in a more accessible, interoperable, and useful manner and to enable the widest possible range of stakeholders to use contracting data effectively.

According to the website of the OCDS:

> *"The Open Contracting Data Standard (OCDS) enables disclosure of data and documents at all stages of the contracting process by defining a common data model. It was created to support organisations to increase contracting transparency, and allow deeper analysis of contracting data by a wide range of users.",* available under *http://standard.open-contracting.org/latest/en/*

This document presents an overview of the OCDS as well as a high-level approach to the implementation of each element in an e-Procurement system. In order for the OCDS to be successfully incorporated into the implementation process, the OCDS-specific requirements should be explicitly defined during the system specification phase (for example, data fields, triggers, reports, visualizer). More information on this topic is available in the e-Procurement Requirements section of the World Bank's e-Procurement Toolkit.[1] If, on the other hand, the OCDS is incorporated into an existing e-Procurement implementation plan, the OCDS /schema should be examined to determine if any specific functionalities, data elements, and/or internal system modules need to be revised (such as data fields to be mapped) in order to extend the existing e-Procurement system. This option is further discussed in section 3 below.

1.1 EXPECTED BENEFITS

In adopting the OCDS into the contracting process, the expected benefits will extend equally to the government, nongovernmental organizations (NGOs), the business community, and the general public. The users included in these categories utilize open contracting data in order to:

- » achieve value for money
- » provide the potential for government to increase trust and transparency with the public
- » strengthen the transparency, accountability, and integrity of public contracting
- » access and review details of the public contracting processes of NGOs and monitor spending to ensure that citizens are getting the best outcomes
- » enable better competition in public contracting
- » monitor service delivery (from both donor- and budget-executed projects) for effectiveness
- » get the general public involved in decision making
- » reduce opportunities for fraudulent or corrupt activity by government officials

---

[1] See http://eProcurementToolkit.org.

2 STANDARD

The OCDS enables the disclosure of data and documents at all stages of the contracting process by defining a common data model. It was created to help organizations to increase contracting transparency and to allow for a deeper analysis of contracting data by a wide range of users.[2] The contracting process should contain information about the different stages in the lifespan of a contract, from the planning to the implementation and final completion/conclusion stages.

2.1 CONTRACTING PROCESS

As per the OCDS, a "unique contracting" process refers to planning, tendering, awarding, and providing information on contracts. From the procurement perspective, this means a unique contracting process for every tender.

There can be multiple awards, contracts, and implementation details as part of each open contracting process, but there should always be only one tender in a process.

Each unique contracting procedure should be assigned an Open Contracting ID (OCID). An OCID is a globally unique identifier for a contracting process. The publisher should use a consistent OCID in the contracting process in order to tie together the related tenders, awards, contracts, and implementation details. Without this globally unique identifier, users do not have a complete picture of the contracting enterprise, as they will not be able to recognize how the different procedures are connected to each other.

The use of a globally unique OCID is important when different contracting datasets are gathered together by third parties, as it enables them to confidently refer to a unique contracting process.

2.2 STRUCTURED DATA MODEL

The OCDS provides a structured data model for capturing in-depth information on all stages of the contracting process.

The current version of this data model is provided by a JSON Schema,[3] which describes field names, field definitions, and structures for the data. The compliance of data with the OCDS is assessed against this schema (Release Schema).

However, there are many instances where publishers and users will want to work with data serialized in other formats. For this reason, the current version of OCDS supports a number of secondary serializations that are based on the standardized/canonical schema.

The majority of OCDS data is held within a release, that is, an "object" that presents information about a particular event in the lifetime of a contract. Since the OCDS gathers information about the different stages in a contract's lifespan, the releases are made up of a number of blocks of data, including:

---

[2] According to http://standard.open-contracting.org/latest/en/.

[3] JSON refers to JavaScript Object Notation.

» Planning: information from the planning phase of the contracting process, which includes information related to the process of deciding what to contract for and when and how
» Tender: the activities undertaken in order to enter into a contract
» Award: information on the award of the contract (and there may be more than one award per contracting process, for instance, because the contract is split between different providers or because it is a standing offer)
» Contract: information that includes the contract creation phase, contract amendments, and contract execution process

Since the OCDS release relates to a single contracting process, it can contain information only on a specific tender, as well as information on a multiple award, the contract, and implementation. These releases should be published as part of a release package that is modeled after the Data Package protocol and includes meta-data about:

» the release(s) it contains
» the publisher
» the data licensing information

The structure of the release package is presented below:

Field Name	Description	Format
uri	The URI of this package that identifies it uniquely in the world.	uri string
publishedDate	The date that this package was published. Ideally this should be the latest date for which there is release information.	date-time string
releases	For further details, see the release-schema.json section at: http://standard.open-contracting.org/latest/en/schema/reference/	Object Array
publisher	Information to uniquely identify the publisher of this package: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#publisher	Object
license	A link to the license that applies to the data in this data package. A Public Domain Dedication or Open Definition Conformant license is strongly recommended. The canonical URI of the license should be used. Documents linked to this file may be under other license conditions.	uri string
publicationPolicy	A link to a document describing the publisher's http://ocds.open-contracting.org/standard/r/1__0__0/en/implementation/publication_patterns/" \l "publication-policy"	uri string

TABLE 1: RELEASE PACKAGE
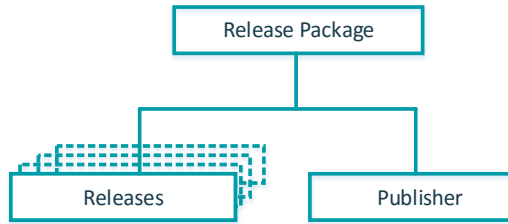
FIGURE 1: RELEASE PACKAGE

The top level of a release consists of the following fields and objects:

Field Name	Description	Format
ocid	A globally unique identifier for this Open Contracting Process, composed of a publisher prefix and an identifier for the contracting process. For more information, see the Open Contracting Identifier guidance at: http://ocds.open-contracting.org/standard/r/1__0__0/en/key_concepts/identifiers/#ocid.	string
id	A unique identifier that identifies this release. A release ID must be unique within a release package and must not contain the "#" character.	string
date	The date this information is released. It may well be the same as the parent publishedDate, but it must not be later than the publishedDate from the parent package. It is used to determine the merge order.	date-time string
tag	A value from the releaseTag codelist that identifies the nature of the release being made. Tags may be used to filter a release, or, in the future, for advanced validation when certain kinds of releases should contain certain fields.	Array
initiationType	String specifying the type of initiation process used for this contract, taken from the initiationType codelist. Currently only tender is supported. More information on the codelist is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists#initiation-type.	string
planning	Information from the planning phase of the contracting process. This includes information related to the process of deciding what to contract for and when and how. More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#planning.	Reference
tender	The activities undertaken in order to enter into a contract. More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#tender.	Reference

buyer	The entity whose budget will be used to purchase the goods. This may be different from the procuring agency that may be specified in the tender data. More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#organization.	Reference
awards	Information from the award phase of the contracting process. There may be more than one award per contracting process, e.g., because the contract is split between different providers or because it is a standing offer. More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#award.	Object Array
contracts	Information from the contract creation phase of the procurement process. More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#contract.	Object Array
language	Specifies the default language of the data using either two-digit International Standards Organization (ISO) 639-1 or extended BCP47 language tags. The use of two-letter codes from ISO 639-1 is strongly recommended.	string

TABLE 2: RELEASE SCHEMA
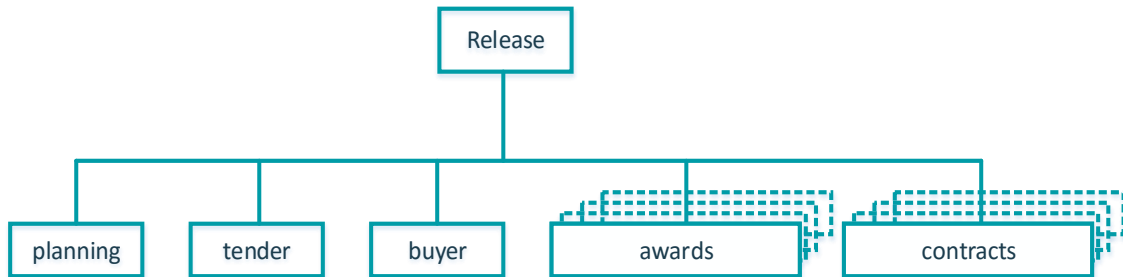
FIGURE 2: RELEASE SCHEMA

The releases are made up of a number of blocks of data, including planning, tender, award, contract, and implementation. The fields included in these blocks are presented below:

Field Name	Description	Format
budget	More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#budget.	Reference
rationale	The rationale for the procurement provided in free text. More detail can be provided in an attached document.	string
documents	A list of documents related to the planning process. More information is available at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#document	Object Array

TABLE 3: PLANNING BLOCK
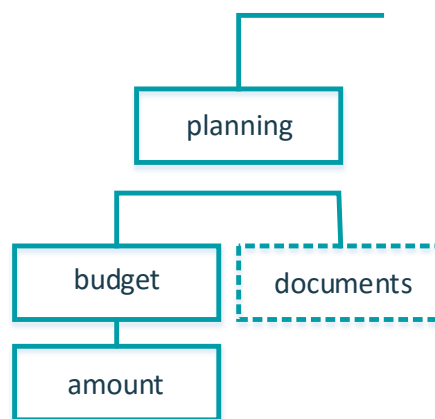
FIGURE 3: PLANNING BLOCK

Field Name	Description	Format
id	An identifier for this tender process. This may be the same as the OCID or may be drawn from an internally held identifier for this tender.	string, integer
title	Tender title.	string
description	Tender description.	string
status	The current status of the tender based on the tenderStatus codelist. More information on the codelist can be found at:	string

	http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists#tender-status.	
items	The goods and services to be purchased, broken into line items wherever possible. Items should not be duplicated, but a quantity of 2 specified instead: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#item.	Object Array
minValue	The minimum estimated value of the procurement: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#value.	Reference
value	The total upper estimated value of the procurement: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#value.	Reference
procurementMethod	Specify tendering method against the method codelist as per Agreement on Government Procurement (GPA) definitions of Open, Selective, and Limited: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists#method.	string
procurementMethodRationale	Rationale of procurement method, especially in the case of Limited tendering.	string
awardCriteria	Specifies the award criteria for the procurement, using the award criteria codelist: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists#award-criteria.	string
awardCriteriaDetails	Any detailed or further information on the award or selection criteria.	string
submissionMethod	Specifies the method by which bids must be submitted—in person, written, or electronic auction—using the submission method codelist: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists#submission-method.	Array
submissionMethodDetails	Any detailed or further information on the submission method. This may include the address, e-mail address, or online service to which bids should be submitted and any special requirements to be followed for submissions.	string

tenderPeriod	The period when the tender is open for submissions. The end date is the closing date for tender submissions: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#period.	Reference
enquiryPeriod	The period during which enquiries may be made and answered: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#period.	Reference
hasEnquiries	A Yes/No field to indicate whether enquiries were part of tender process.	boolean
eligibilityCriteria	A description of any eligibility criteria for potential suppliers.	string
awardPeriod	The date or period when an award is anticipated to be made: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#period.	Reference
numberOfTenderers		integer
tenderers	All entities that submit a tender: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#organization.	Object Array
procuringEntity	The entity managing the procurement, which may be different from the buyer who is paying for/using the items being procured: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#organization.	Reference
documents	All documents and attachments related to the tender, including any notices. More information on the codelist can be found at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists#document-type. More information on the document section can be found at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#document.	Object Array
milestones	A list of milestones associated with the tender: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#milestone.	Object Array

amendment	More information can be found at:	Reference
	http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#amendment.	

TABLE 4: TENDER BLOCK

FIGURE 4: TENDER BLOCK

Field name	Description	Format
id	The identifier for this award. It must be unique and cannot change during the Open Contracting Process it is a part of (defined by a single OCID): http://ocds.open-contracting.org/standard/r/1__0__0/en/key_concepts/identifiers.	string, integer
title	Award title.	string
description	Award description.	string
status	The current status of the award drawn from the awardStatus codelist: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists/#award-status.	string
date	The date of the contract award. This is usually the date on which a decision to award was made.	date-time string
value	The total value of the award. In the case of a framework contract, this may be the total estimated lifetime value, or maximum value, of the agreement. There may be more than one award per procurement: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#value.	Reference

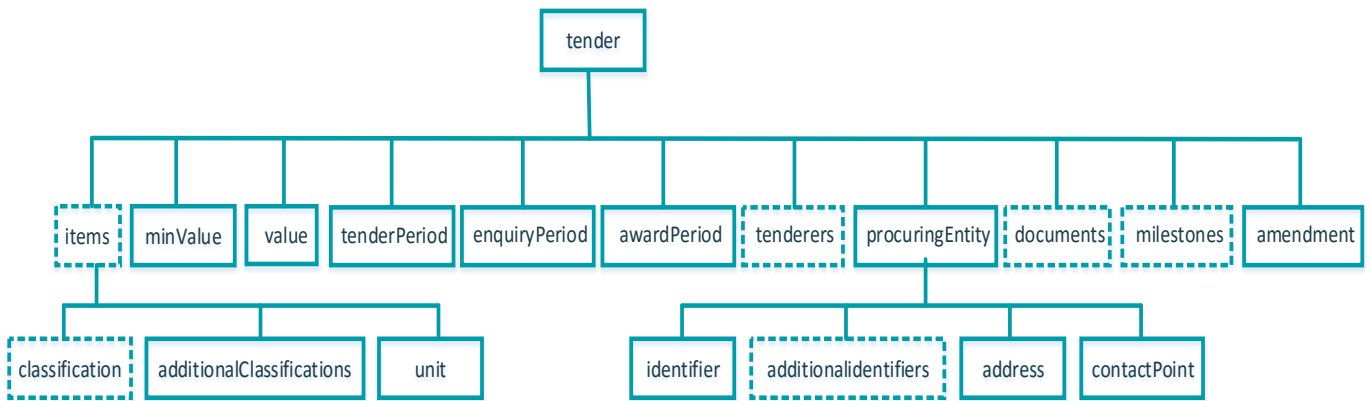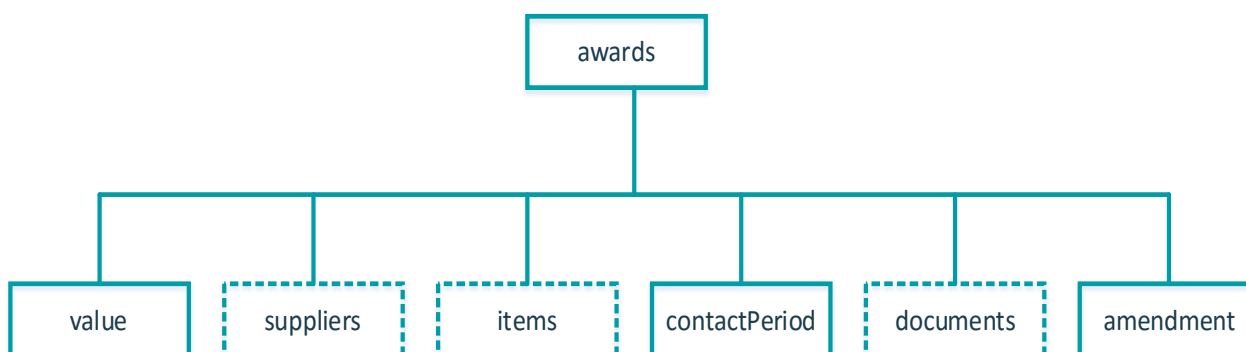| suppliers	The suppliers awarded this contract. If different suppliers have been awarded different items of value, these should be split into separate award blocks: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#organization.	Object Array
items	The goods and services awarded in this award, broken into line items wherever possible. Items should not be duplicated, but the quantity specified instead: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#item.	Object Array
contractPeriod	The period for which the contract has been awarded: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#period.	Reference
documents	All documents and attachments related to the award, including any notices: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#document.	Object Array
amendment	More information can be found at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#amendment.	Reference

TABLE 5: AWARD BLOCK

FIGURE 5: AWARD BLOCK

Field Name	Description	Format
id	The identifier for this contract. It must be unique and cannot change during its Open Contracting Process (defined by a single OCID). For more information, see the Open Contracting Identifier guidance at: http://ocds.open-contracting.org/standard/r/1__0__0/en/key_concepts/identifiers/#ocid.	string, integer
awardID	The award.id against which this contract is being issued.	string, integer
title	Contract title.	string
description	Contract description.	string
status	The current status of the contract. Drawn from the contractStatus codelist: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/codelists/#contract-status.	string
period	The start and end date for the contract: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#period.	Reference
value	The total value of this contract: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#value.	Reference
items	The goods, services, and any intangible outcomes in this contract. Note: If the items are the same as the award, do not repeat: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#item.	Object Array
dateSigned	The date the contract was signed. In the case of multiple signatures, the date of the last signature.	date-time string
documents	All documents and attachments related to the contract, including any notices: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#document.	Object Array
amendment	More information can be found at: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#amendment.	Reference
Implementation	Information related to the implementation of the contract in accordance with the obligations laid out therein: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#implementation.	Reference

TABLE 6: CONTRACT BLOCK

Field Name	Description	Format
transactions	A list of the spending transactions made against this contract: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#transaction.	Object Array
milestones	As milestones are completed, milestone completions should be documented: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#milestone.	Object Array
documents	Documents and reports that are part of the implementation phase, e.g., audit and evaluation reports. As milestones are completed, milestone completions should be documented: http://ocds.open-contracting.org/standard/r/1__0__0/en/schema/reference/#document.	Object Array

TABLE 7: IMPLEMENTATION BLOCK
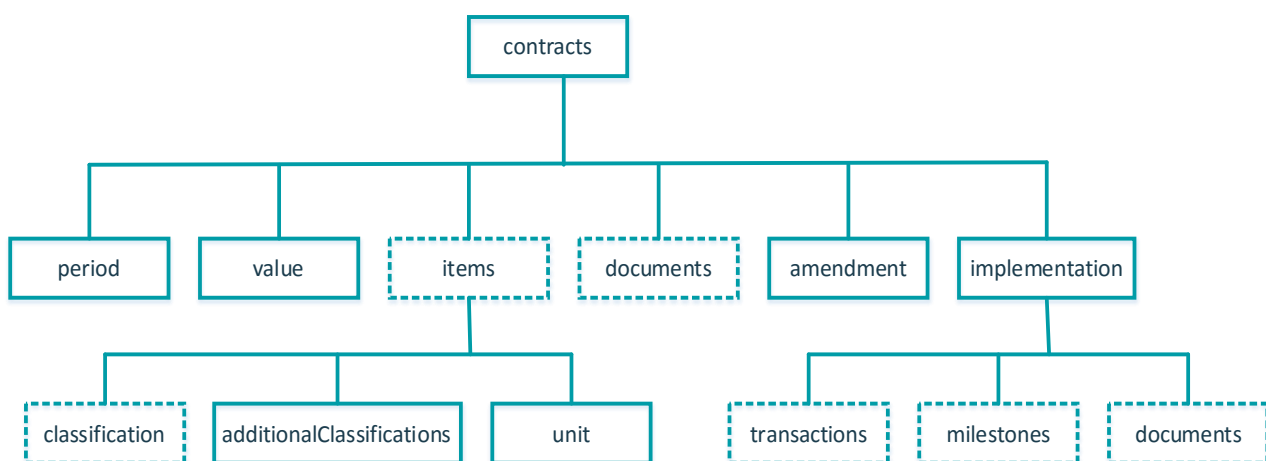
FIGURE 6: CONTRACT BLOCK

A brief field explanation regarding the release structure is available here:

http://standard.open-contracting.org/latest/en/schema/reference/#release-structure

Since there will be multiple releases in relation to a specific contracting process, a single contracting record for each OCID should be made available in order to provide a summary of all the available data about this particular contract. As per the standard (http://standard.open-contracting.org/latest/en/schema/records_reference/), a record must contain:

> » an OCID
> » an array of releases related to this contracting process, either by providing a website of the location where these releases can be found or by embedding a full copy of the release

Additionaly, records should contain:

> » compiledRelease – the latest version of all open contracting process fields, represented by the use of the release schema

Records may optionally contain:

» versionedRelease – containing the history of the data in the compiledRecord, with all known past values of any field and the release that information came from.

The versioned release format is designed to support an analysis of how data have changed between releases and will often be generated by data users rather than publishers.

2.3 RELEASE HANDLING

The full OCDS data model is based on the idea of publishing a new release every time information on a contracting process changes. This way, users can gain a full view of change over time, and third parties can understand what needs to be updated in any system that is tracking the progress of a contract. Also, the publishers should decide whether they will include all the information in each release or provide only the changes.

Practically speaking, when an action changes any of the fields (from the top or lower level) described above, a new release should be published. For example:

» When a new tender is published in the e-Procurement system, a new release with the tag "tender" is published. This release contains information regarding the initiationType, the planning, the tender, the buyer, and the language blocks.

» For the same tender, when a clarification is published, the field "hasEnquiries" included in the "tender" block should change from false to true. In this case, a new release with the tag "tenderUpdate" should be published in order to reflect the update to the existing tender release.

This model also requires publishers to pay careful attention to null values and missing fields. As per the standard, the fields that are not being used or that have no value can be excluded in their entirety from a published file. Also, when a value of a previously published field needs to be removed, this field should be explicity set to null.

2.4 USE OF CODELISTS

In order to enable the interoperability of the data, the OCDS schema references a number of codelists. There are two kinds of codelists: open and closed.

An open codelist provides suggested codes, but publishers can extend these lists with new codes on the basis of consensus with other publishers or by using a code prefixed with "x_" to indicate that it is a local "eXtension" to the codelist.

For example, the OCDS provides a list of types of documents that may be attached to tenders, awards, contracts, and milestones. However, a group of publishers may discover the need to identify another kind of document. These publishers would not need to wait for a future version of the standard to be agreed upon to add a new code to an open codelist, although they should consult with the community through the mailing list and GitHub platform and suggest that the new code be formally incorporated into the codelists.

The open codelists are presented in the following table:

Codelist	Description	Example

Item Classification Scheme	Items should be classified using existing gazetteers and codelists, such as the United Nations Standard Products and Services Code (UNSPSC). Open codelists are strongly preferred.	The items block that is used to list the line items associated with a tender, award, or contract should contain the classification for the item. In that case, the Item Classification Scheme should be used in order to identify the preferred classification list.
		
		When the UNSPSC code (80101508-Business intelligence consulting services) is used in an e-Procurement system, the classification of the item should contain the following details:
		
		"classification": {
		
		"scheme": "UNSPSC,"
		
		"description": "Business intelligence consulting services,"
		
		"id": "80101508,"
		
		"uri": http://www.unspsc.org/codeset--downloads
		
		}

Organization Identifier Scheme	The Organization Identifier Scheme currently uses the codes from the International Aid Transparency Initiative "Organisation Registration Agency" codelist.	The Organisation Registration Agency (http://iatistandard.org/201/codelists/OrganisationRegistrationAgency) contains the organization registry agencies and their codes. These can be used as the prefix for an organization identifier.
		
		For example, in an e-Procurement system, the involved buyer organization is from Zambia. The organization register prefix for Zambia Patents and Companies Registration Agency is ZM-PCR. The globally unique organization identifier can then expressed as follows:
		
		"identifier": {
		
		    "scheme": "ZM-PCR,"
		
		    "legalName": "Ministry of Finance,"
		
		    "uri": "https://www.pacra.org.zm/#/p/pacrasearch/"
		
		}

Document Type	This list provides details of the documents that publishers may wish to provide at various	In an e-Procurement system, there are many documents available in the tender describing the bidding process. In that case, the documentType of these documents should be "biddingDocuments."

	points in their contracting process.	For the documents containing the evaluation report for the tender process, the "evaluationReports" code should be used.
Award Criteria	The award criteria code list describes the basis on which contract awards will be made.	In an e-Procurement system, the evaluation mechanism for a tender could be the Least Cost Selection (LCS). In that case, the suitable award criteria code should be "lowestCost."
Submission Method	The submission method codelist is used to identify the mechanism through which a submission may be made.	In the tender process of an e-Procurement system, the suppliers are allowed to create and submit their bids using electronic means (i.e., online). In that case, the submission method code should be "electronicSubmission."

TABLE 8: OPEN CODELISTS

A closed codelist provides mandatory codes, and publishers should use only those values provided in the official list. Changes to closed codelists should take place through the governance and revision process for the schema.

Codes are case sensitive and are generally provided as English language "CamelCase." Code values should not be translated, though the OCDS team will work with publishers to provide alternative translations of code titles and definitions.

The closed codelists are presented in the following table:

Codelist	Description	Example
Release Tag	A contracting process may result in a number of releases of information over time. These should be tagged to indicate the stage of the contracting process they relate to.	The tag field is used in the release in order to identify the nature of the release being made. In case a new tender is published in an e-Procurement system, the tag field included in the release should be "tender."
Initiation Type	Contracting processes may be formed under a number of different procedures. Currently, only "tender" is supported in this codelist. Future versions of the standard may support other initiation types. The initiation type may provide information on consuming applications on the different blocks of data and releases they should expect from a contracting process.	N/A

Tender Status	The tender.status field is used to indicate the current status of a tender process.	In an e-Procurement system, the tender is opened for bid submissions. In that case, the status code should be "active."
Method	The method codelist is based upon the GPA definitions.	In an e-Procurement system, the procurement method of a tender could be "Open Bidding National," which means that all suppliers may submit a tender. In that case, the method code should be "open."
Award Status	An award as it moves through multiple states. Releases over time may update the status of an award.	In an e-Procurement system, until the award is completed, the award status should be "active."
Contract Status	Contracts can move through multiple states. Releases over time may update the status of a contract.	A contract can be in "pending" status if it is in the planning stages, i.e., the tender award has been finalized and negotiations are currently taking place with the preferred supplier. Once the contract implementation begins, its status is updated to "active." Depending on the progress, the contract can either be updated to "cancelled" or "terminated." The last code is also used to reflect contracts that have been successfully delivered and are no longer in force.
Currency	OCDS uses the ISO 3-letter currency codes maintained in http://en.wikipedia.org/wiki/ISO_4217	In an e-Procurement system, the currency of the value could be expressed in Zambia kwacha (ZMW). In that case, the ZMW code should be used.
Milestone Status	In the context of a tender block, milestones describe the key deliverables of a contract or key points during the lifetime of the contract.	Milestones are initially created in a status of "notMet" as the contract is mapped out into an implementation plan. The status of the milestones can change to either "met" or "partiallyMet" as implementation progresses.

TABLE 9: CLOSED CODELISTS

The completed list of the codelists is available at http://standard.open-contracting.org/latest/en/schema/codelists.

3 METHODOLOGY

3.1 INCEPTION

Before starting any analysis in support of the OCDS within an e-Procurement system, the publishers (that is, the governments implementing e-Procurement) may want to consider some guidance on implementation that is available on the OCDS website (http://standard.open-contracting.org/latest/en/implementation). They should then make a determination on the particular issues described in the following subsections.

3.1.1 WHAT TO PUBLISH

As per the specification of the standard, the publishers will need to decide the extent of the data that will be published and also determine their technical approach. More precisely, they will need to select the volume of data they plan to make available from among the basic, intermediate, and advanced data and documents levels that governments are able to publish along with the below five-star framework for the technical approach.

http://standard.open-contracting.org/latest/en/implementation/levels/
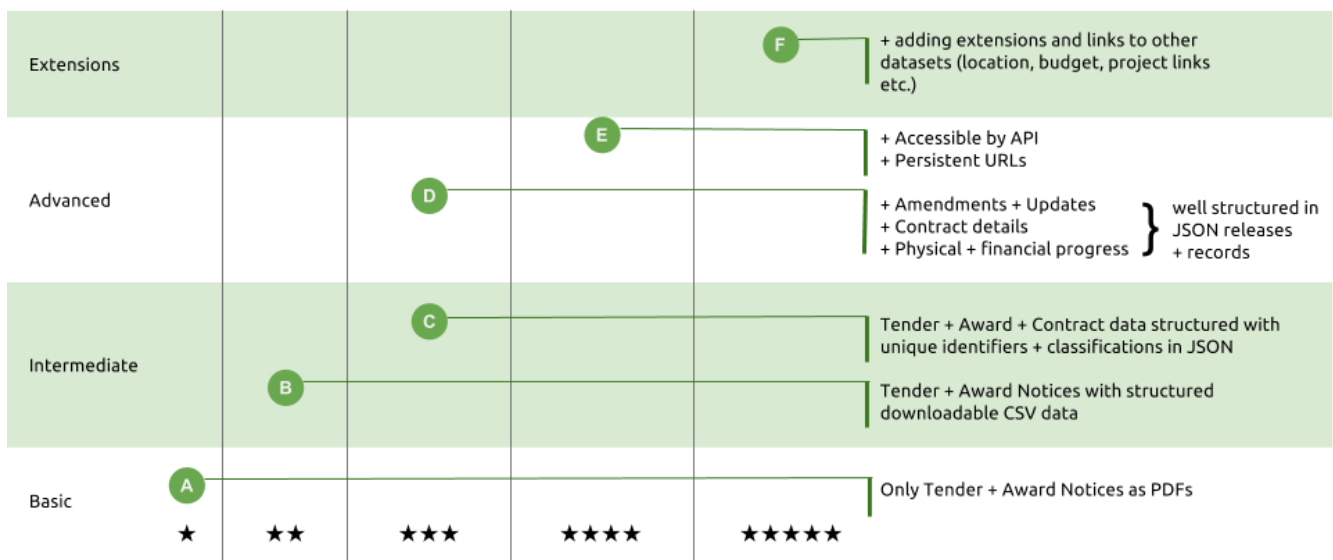
FIGURE 7: PUBLICATION LEVELS

3.1.2 LICENSING

It is important that the information is published with a license that allows for the free use of information. The guidance on the OCDS website may be consulted for further details.

http://standard.open-contracting.org/latest/en/implementation/licensing/

3.1.3 PUBLICATION POLICIES

The publishers should determine the data and documents that will be included in the releases and how they plan to classify and categorize the different items. These decisions would be made available on a public web page or in a document in the publication policy field included in the release package schema.

http://standard.open-contracting.org/latest/en/implementation/publication_policy/

3.1.4 SERIALIZATION

The current standardized/canonical version of this data model is provided by a JSON Schema that describes field names, field definitions, and structures for the data. However, there are many instances in which publishers and users will want to work with data serialized in other formats. Therefore, the publishers will need to make a decision on the serialization of their data (JSON or CSV formats).

http://standard.open-contracting.org/latest/en/implementation/serialization/

3.1.5 DATA FILES, APIS, AND DISCOVERY

Depending on the use of the OCDS data, the publishers should determine the appropriate formats in which their data will be made available.

http://standard.open-contracting.org/latest/en/implementation/hosting/

3.1.6 REGISTRATION

In order to use the globally unique identifier for their contracting procedures, the publishers are required to obtain an OCID prefix.

http://standard.open-contracting.org/latest/en/implementation/registration/

3.2 ANALYSIS

3.2.1 CONSTRUCTION OF OCID

As per the OCDS specification, each contracting process should have a consistent OCID in order to tie together the related tenders, awards, contracts, and implementation details. The publishers need to decide on the pattern that will be used in these contracting steps.

The OCID should comprise four parts with two "-" separators. The first three can generally be set as constant values in most systems:

» A prefix agency (characters 1–4), currently only OCDS
» A '-' separator
» A registered prefix (characters 6–11), a random alphanumeric string given to publishers
» A publisher namespace (characters 12–13) for allowing publishers to maintain separate series of local identifiers
» A '-' separator
» An internal identifier (characters 15 and above) drawn from publishers' existing systems

For example, the registered prefix given to the Zambia Public Procurement Authority is 23g63a. The construction of the OCID would be the following: ocds-23g63a01-10640

where 10640 is the internal identifier of a tender in an e-Procurement system.

3.2.2 FIELD MAPPING (E-PROCUREMENT SYSTEM - OCDS SCHEMA)

In order to use values in line with the OCDS and depending on the publication policy of each publisher, a mapping to the system's existing fields should be carried out. In the steps below, it is assumed that all data included in the release schema will be published.

3.2.2.1   Step1

The publisher should ensure that all of the fields required by the OCDS are available in the database of the e-Procurement modules. If any information is missing, the related fields need to be created in the database in order to facilitate the publication of records and releases with complete information. In some cases, no extra user-input is required, as the value of a field requested by OCDS can be calculated from existing information in the system, however in some cases additional user input is required. Therefore, where OCDS is being implemented as part of a e-Procurement implementation, the mapping should be carried out at the point at which forms for data collection are being developed.

For example, in e-Procurement modules, it may not be required to provide an explanation of why a selected procurement method is appropriate to a specific tender. However, in the tender block included in the release of the OCDS, a field called "procurementMethodRationale" is used for this purpose. Therefore, in this case, a new field would need to be added in the tender details of the system in order to ensure that complete release information is published, as per the specification.

3.2.2.2   Step 2

The publisher should map the fields available in the database of the e-Procurement modules with the fields defined in the structured JSON Schema of the OCDS. The mapping of the fields to the open and closed codelists is not necessary at this stage.

For example, in an e-Procurement system, the estimated value of the procurement could be stored in a field called "Estimated total contract value," and the currency of this field could be always set as ZMW. In the JSON Schema of the OCDS, the field that describes the above information is called "value," its format is referenced wherever a financial figure is used, and it is available in the tender block included in the release. As per the specification, the financial values should always be published with a currency attached.

Field Name	Description	Format
amount	Amount as a number.	number
currency	The currency in 3-letter ISO 4217 format.	string

TABLE 10: FINANCIAL VALUES

Therefore, the mapping of these fields should be:

- » amount ⇒ value of the "Estimated total contract value" field, available in the e-Procurement system
- » currency ⇒ "ZMW"

3.2.2.3   Step 3

The publisher should map the values stored in the system's database to the open and closed codelists. An example related to this mapping is described below. The tender.status field is used to indicate the current status of a tender process. The mandatory codes of this codelist are the following:

Code	Title	Description
planned	Planned	This tender has been proposed but is not yet taking place.
active	Active	A tender process is currently taking place.
cancelled	Cancelled	The tender process has been cancelled.
unsuccessful	Unsuccessful	The tender process was unsuccessful.
complete	Complete	The tender process is complete.

TABLE 11: TENDER STATUS CODES

In an e-Procurement system, the field that describes the status of the tender could hold one of the following values:

» Draft
» Bid submission
» Awaiting bid opening
» Under evaluation
» Awarded
» Established
» To be archived
» Archived
» Cancelled

Taking into consideration the potential values of the system and the description of the current status of the tender process in the codelist, each value of the tender.status field should be mapped to the related codelist value. Therefore, the mapping would be:

» Planned	⇒	Draft
» Active	⇒	Bid submission Awaiting bid opening Under evaluation
» Cancelled	⇒	Cancelled
» Complete	⇒	Awarded Established To be archived Archived

3.2.2.4 Step 4

The OCDS can be extended to support publication of additional fields of information that are important within a countries particular legal framework, or to meet local user needs. If during the field mapping you identify information that could be disclosed in this way, but for which no existing OCDS fields exist, you can consult with the OCDS helpdesk to design a schema extension for representing this information

3.2.3 WHEN TO PUBLISH AN OCDS RELEASE

After completing the field mapping between the e-Procurement system and the OCDS format, the decision must be made on the functionalities that will publish an OCDS release. At this stage, the publisher should define the logic (that is, the decision tree) for determining whether or not an action undertaken within the e-Procurement system should publish an OCDS release. This is linked to the decision on how to generate new releases. Since the full OCDS data model is based on the idea of publishing a new release every time information about a contracting process changes, the following checks should be applied.

3.2.3.1   Publication of an OCDS release

The e-Procurement system should publish the changed/added information in an OCDS release in two cases:

1. When an "add-edit-delete" action of an object may change the fields that are relevant to the OCDS requirements. Every time a user edits any of the fields linked to the OCDS JSON Schema, a new release should be published. More specifically, when it is necessary to:
    »   **Add new information.** During the tender stage of a contracting process, an OCDS field that was empty in a previous release could gain a value in the next one. For example, until the bid opening ceremony takes place, the "numberOfTenderers" and "tenderers" fields included in the tender block of the OCDS release are empty. After the successful bid opening, these fields should contain the related information. Thus, the e-Procurement system should publish the OCDS release in order to provide the aforementioned information.
    »   **Edit existing information.** During the contracting process, many fields could change their values. For example, an update may be performed in the title field of a tender. Since this field is mapped to the title included in the tender block of the OCDS release, the e-Procurement system should publish the OCDS release in order to amend the related field.
    »   **Remove existing information.** During the contracting process in an e-Procurement system, a field could be removed. For example, the "Award criteria details" field could be eliminated. In that case, the e-Procurement system should publish the OCDS release in order to define the removal.
2. When an action within the e-Procurement system may change the status of the tender/award/contract within the contracting process.
    In an e-Procurement system, the status in the tender, award, or contract blocks could change during the contracting process. For example, after the cancellation of an active tender, the tender status should be changed from active to cancelled. In that case, the e-Procurement system should publish the OCDS release in order to define the status.

3.2.3.2   No publication of the OCDS release

The e-Procurement system should not publish the information in an OCDS release when an "add-edit-delete" action of an object may change the fields that are irrelevant to the OCDS requirements.

There are many cases when a user may edit/add information to a field in the e-Procurement system that is not mapped to any of the OCDS fields. In those cases, the system should not publish an OCDS release, as none of the OCDS data has changed. For example, during the evaluation of the tender, the evaluators assign scores to the particular bids. These scores are stored in the database and constitute important information for the system. However, since this information is not included in the OCDS fields, the e-Procurement system should not publish an OCDS release in this case.

3.2.3.3   Publication of sensitive information

Although the publication of data is advertised as "open," there will be circumstances where publishers may not publish certain information. For instance, the procurement of defense equipment may not be subject to OCDS publication. The implementation of the OCDS should allow a "redacting" of the data to ensure compliance with a country's legislation with regard to the protection of confidential information.

3.2.4 SYSTEM ARCHITECTURE CONSIDERATIONS

Any enhancements made to an e-Procurement system to support the OCDS requirements should be performed in a way that does not impose risks to the service level or the quality of the service requirements to which the system must adhere. In this regard, the new functionalities incorporated into the system to satisfy the OCDS requirements should not necessitate substantial changes to its overall architectural design, impact its overall performance, or introduce any security loopholes.

On the whole, there are two core approaches that can be followed in implementing the functionalities needed to satisfy the OCDS requirements.

» One approach is for all OCDS-related functionalities to be implemented within the existing system modules. The benefit here is that absolutely no change to the overall software architecture is needed, while each e-Procurement module becomes responsible for maintaining and supporting the necessary functional services for OCDS compliance. The downside, however, is that any future updates of the OCDS would likely require changes to several e-Procurement modules, while in parallel, the synchronicity in which OCDS-compliant packages must be created can create performance issues.

» Another approach is for all OCDS-related functionalities to be implemented within a newly dedicated "OCDS module" that can be called upon when necessary by the existing e-Procurement modules. Although this approach requires some changes to the overall software architecture, it provides the benefit of centralizing all OCDS-related opeations in one module. Future updates of the OCDS can be more effectively addressed, while the addition of new e-Procurement modules can easily call on the OCDS module service in order to record the necessary data with minimal effort. Most important, however, is that communications between the core e-Procurement and the OCDS modules can run in an asynchronous manner so as not to affect overall system performance. On the other hand, the introduction of a new OCDS module requires an appropriate design to address the transactional nature of communications, while in parallel, the new OCDS module must incorporate the necessary fail-safes to eliminate potential security theats.

The exact architectural design to be followed in an e-Procurement system to support the OCDS requirements largely depends on the existing system design, the time and budget available for potential revisions to the overall system architecture, and the intention of the service provider and/or beneficiaries with regard to the future evolution of the system.

3.2.5 USUAL SOFTWARE ENGINEERING METHODOLOGY

Adjusting an e-Procurement system in order to comply with the OCDS should be viewed as a software engineering project. During the lifecycle of such projects, well-defined scientific principles, methods, and procedures should be followed. The project's progression should contain at least the following steps:

» **Scoping/analysis:** Since proper scope definition is one of the key factors of project success, a well-defined analysis should be completed in this phase (see the sections 3.2.1–3.2.3).

» **Design:** During the design phase, the user should develop a specification for the system in order to avoid risks to the service level or the quality of the service requirements (see section 3.2.4).

» **Testing:** During the testing phase, the respective test strategy should be designed to describe the testing approach of the software development cycle (see section 3.3.1).

In addition, an appropriate technique should be used in order to ensure that the final result:

» meets the original business objectives by rendering the system OCDS compliant
» does not negatively impact existing system operations/processes

Practically speaking, some sort of software engineering methodology must be followed within the context of a well-scoped project (time, budget, objectives) and a properly assigned team (client, vendor).

3.3 TESTING

3.3.1 TEST STRATEGY

Since the OCDS constitutes a data model that should be followed by the various modules of the e-Procurement system to ensure correct publication of the expected data, the tests on the software should focus primarily on the specific parameterizations/adaptations of the respective modules in accordance with OCDS specifications.

In order to ensure full functional operation within the systems, three levels of testing should be conducted: integration testing, system testing, and benefits testing.

3.3.1.1   Integration testing

During integration testing, individual software modules should be combined and tested as a group. Integration testing takes its input modules that have been tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers, as its output, the integrated system ready for the system testing.

» **Inter-module integration:** During the integration between internal modules, the tester should perform actions in the source module and ensure that the appropriate information is transferred to the target module. For example, after the finalization of awards in the tendering module, the related contracts should be created in the contracting module, including the information given in the tendering module (for example, awarded supplier, contract value, UNSPSC code, contract title, contract description, and so on). The tester should perform this testing for all possible flows described in the test plan.
» **Module-OCDS integration:** During the module-OCDS integration process, the tester should perform actions in the source module and ensure that the appropriate information is published as an OCDS release. More specifically, the tester should ensure that the field mapping between the source module and the OCDS schema is in accordance with the specification, that is, the correct mapping between the source and OCDS fields. Then the tester should ensure that the correct releases have or have not been created after each action in the source module. For example, the tester should ensure that the publication of OCDS releases is performed in line with the test plan.

3.3.1.2   System testing

During system testing, the tester should scan the complete and fully integrated system, ensuring that information is correctly transferred from one module to another and that the corresponding OCDS releases are published.

First, the tester should ensure that the exported OCDS releases are validated against the OCDS release schema. This may be performed by using a JSON Schema Validator program or/and the Validator available at http://standard.open-contracting.org/validator.

The fields that are not in line with the OCDS schema should be fixed. It may be necessary to consult with the OCDS help desk or community to resolve inquiries related to publication scenarios not covered within the existing OCDS documentation.

Next, the tester should perform appropriate actions in the source modules and ensure that the field mapping in the published OCDS releases is in accordance with the codelist specifications.

Additional importance should be given to emptying certain fields and values. As per the OCDS specifications, there may be cases when a publisher needs to remove rather than update a value that was set in a previous release. In this case, the fields should explicitly be set to null. The tester should ensure that the above specification is fully implemented.

Finally the tester should export the versioned and compiled releases involving a specific contracting process (OCID) from the OCDS system and check that the correct merging strategy has been applied in each field. The JSON validation of these releases (contained within a record package) should also be performed.

3.3.1.3   Benefits testing
In benefits testing, potentials OCDS data re-users could be invited to access, explore and provide feedback on the data produced, and the mechanisms for gaining access to it. Ultimately, the true test of OCDS implementation is whether or not different users from the private sector, public sector and civil society can work with the data to secure some or all of the outcomes described in section 1.1 above.

Benefits testing may highlight areas in need of technical adaptation, or areas in need of documentation in the publication policy of a specific implementation to help users understand how to use the data, and any limitations they should be aware of.

3.4   POTENTIAL EVOLUTION

3.4.1 PLAN FOR A VISUALISER
Although the procurement data are published and available for downloading according to the OCDS, these data are in raw format and require further processing to extract business intelligence. The data also need to be analyzed and represented in visual formats that are easy for users to understand and from which strategic or operational decisions can be made.

Data should be mapped, extracted, converted, and aggregated to provide access to OCDS data releases in JSON and CSV formats. Data captured as per the OCDS should be able to be presented at a later point and visualized from different perspectives, such as those of policy makers, development agencies, the Procurement Regulatory Authority, procuring entities, bidders, oversight agencies, civil society organizations, researchers, businesses, and so forth. It should also clearly display the contracts themselves, the procurement methods used, the contractors and their current status, trends, and comparable contracts and agreements. Web services or application programming interfaces (APIs) should be developed and be usable for producing a visualization portal and/or a procurement management information system (PMIS)—and/or user-specific dashboards for its users.

An OCDS data visualizer should implement the requirements of the aforementioned parties, enabling them to obtain access to specific business information of interest to them. Among the specific requirements, a visualizer should calculate as many of the World Bank's e-Procurement indicators[4] as possible that are related to procurement

---

[4] See the e-Procurement Indicators document in the Bank's e-Procurement Rapid Toolkit website (http:// eProcurementToolkit.org).

performance (such as award type, amount, and so on), service delivery (overlapping projects by geo-location), and anti-corruption efforts. Reports that should be able to be generated using the visualizer include:

- » Average number of bidding document purchases and number of bid submissions
- » Average number of bids per tender
- » Number and percentage of open tendering/total number of tenders
- » Time between notification of contract award and contract signature
- » Number of amendments or extensions to contract duration (per project per contract)
- » Number of days taken to complete a procurement process (per project)
- » Award statistics for a firm
- » Award statistics by entity
- » Award statistics by amount
- » Ranking by supplier country (by top-level UNSPSC/CPV categories, by project)
- » Ranking by winning bidders/suppliers (by top-level UNSPSC/CPV categories, by project)
- » Awards by geographic region
- » Locations of all suppliers (by contract value and number of contracts)
- » Locations of suppliers for particular municipalities, in other words, whether most of the contracts are awarded to companies in their locality, which may be of interest in terms of local business development
- » The depth of local markets, if company register data are available

In general, all OCDS data should be downloadable in a variety of formats, such as JSON, Excel, and CSV. Use of open source tools is preferred.

3.4.1.1 Visualization of the Bank's e-Procurement indicators

The visualizer can be an option as a means to visualize some of the Bank's e-Procurement indicators. These indicators could be addressed through the OCDS data, which contain the required information within the contracting cycle. Examples of such visualizations include:

- » In order to improve public financial management, the price reduction/price variation savings indicator can be used, calculating, for instance, the percentage of contracts awarded on the basis of lowest price. In the JSON Schema of the OCDS, the field that describes the above information is called "awardCriteria," describing the basis upon which contract awards are made, and it is available in the tender block included in the release. Based on this field, the following graph could be generated.
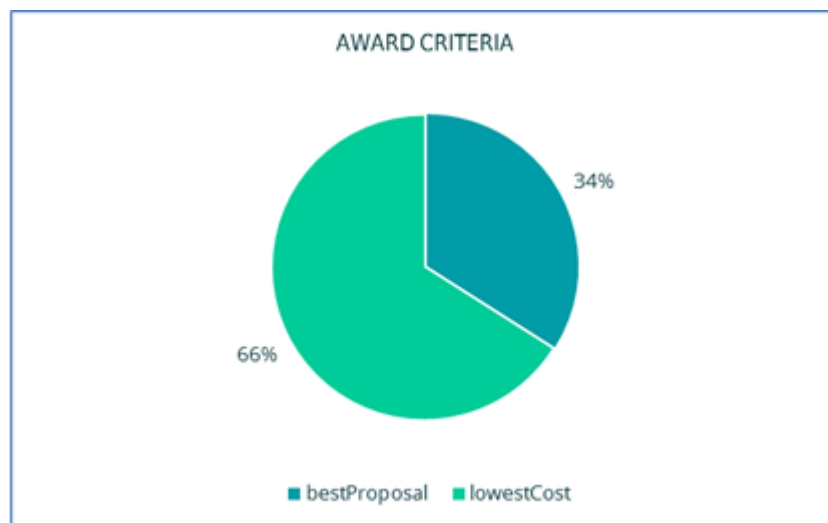

AWARD CRITERIA

34%

66%

■ bestProposal ■ lowestCost

FIGURE 8: SAMPLE CHART FOR AWARD CRITERIA INDICATOR

» In order to improve the efficiency of the procurement process, the Efficiency Improvement indicator for Economic Operators is used, calculating the average cost per procedure. The algorithm used for this indicator is the following:

$$\frac{\sum Costs\_per\_procedure}{Number\_of\_procedures}$$

In the JSON Schema of the OCDS, the field that describes the procedure is called "procurementMethod" and it is available in the tender block included in the release. The field that describes the cost is called "value" and it is available in the award block. Based on these fields, the following graph could be generated.
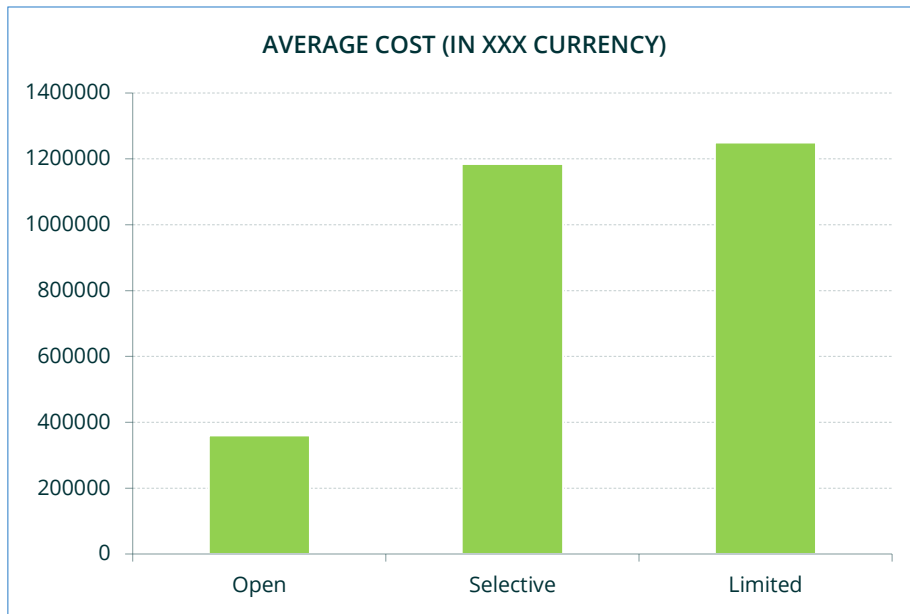

FIGURE 9: SAMPLE CHART FOR THE EFFICIENCY IMPROVEMENT FOR EOs INDICATOR

» The Efficiency Improvement indicator for Contracting Authorities can also be used to improve efficiency, calculating the average length of the pre-award stage (from contract notice to contract award notice). The algorithm used for this indicator is the following:

$$\frac{\sum Length\_of\_contracts\_pre-award\_stage}{Total\_number\_of\_contracts}$$

The dates of both notices can be found in the datePublished field, which is available in the tender block included in the release. For each Contracting Authority, the following graph could be generated.

AVERAGE DURATION OF THE PRE-AWARD STAGE (IN MONTHS)
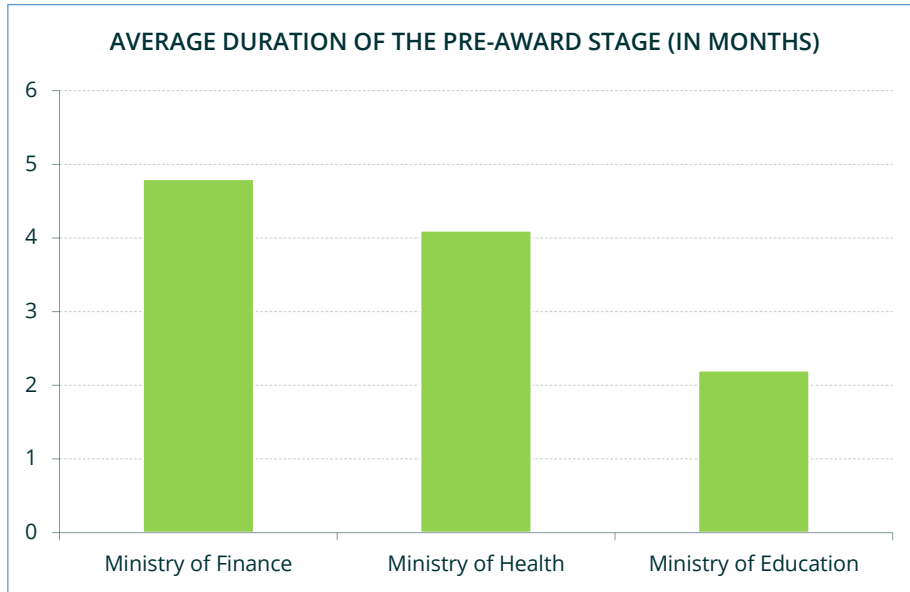
FIGURE 10: SAMPLE CHART FOR THE EFFICIENCY IMPROVEMENT FOR CAs INDICATOR

Rather than rely on third parties to analyze this data, publishers could choose to implement a visualization component to analyze the OCDS-compliant published data themselves and make the results available to interested parties. This functionality should be viewed as an extension of the existing system.

4 KEEPING UPDATED

The OCDS is updated periodically through an open governance process.

Implementers should subscribe to the standard-discuss mailing list via the link at http://standard.open-contracting.org/latest/en/support/ in order to be notified of any new releases of the standard. There is no set timeline for updating to new versions, and implementers should evaluate whether each new release requires any system changes and how these should be scheduled.

Implementers are also encouraged to engage in the governance process, suggesting any areas for further development of the standard. A 1.1 release is scheduled for early 2017.