Technical options for OCDS implementation

Architecture and publication patterns
Architecture
Direct publication from live systems

Live System

Datastore (non-OCDS) → API → OCDS Conversion

→ users

→ 3rd party systems

OCDS data
non-OCDS data
API call / DB query
Separate OCDS datastore - pull and convert

Live System

Datastore (non-OCDS)

Middleware

OCDS Conversion

Datastore (OCDS)

API

3rd party systems

users

OCDS data

non-OCDS data

API call / DB query
Separate OCDS datastore - pull and convert

Live System

Datastore (non-OCDS)

pull

Middleware

OCDS Conversion

Datastore (OCDS)

API

users

3rd party systems

OCDS data

non-OCDS data

API call / DB query
Separate OCDS datastore - convert and push

Live System

Datastore (non-OCDS)

OCDS Conversion

Middleware

Datastore (OCDS)

API

users

3rd party systems

OCDS data
non-OCDS data
API call / DB query
Separate OCDS datastore - manual import

Live System
Datastore (non-OCDS)

Middleware
OCDS Conversion
Datastore (OCDS)

API

user

3rd party system

Flat file export
Publication patterns
**Documents and data**

**Documents**
- Direct URLs
- Long-term archival

**JSON**
- Individual releases
- API
- Bulk downloads

**CSV / Excel**
- Bulk download
- Segmented data: month, year, department etc.
Skills and user needs

It’s important to provide data in structured JSON format for tool developers and advanced users. But more users have the skills to work with flattened (tabular) data, so providing CSVs and spreadsheets is important too.
Serialization

Structured (JSON)

```json
"milestones": [
  {
    "id": "1.0",
    "type": "preProcurement",
    "title": "Public consultation",
    "description": "A public consultation will be held at the offices of the Ministry of Communications on 1st December 2015.",
    "status": "scheduled",
    "dueDate": "2015-12-01T00:00:00:00:00"
  },
  {
    "id": "2.0",
    "type": "approval",
    "title": "Strategic outline case approval",
    "description": "Ministry of finance approved the strategic outline business case for the project",
    "status": "met",
    "dueDate": "2015-01-01T00:00:00:00:00",
    "dateSet": "2015-01-01T00:00:00:00:00",
    "dateModified": "2015-01-01T00:00:00:00:00",
    "documents": []
  }
]
```

Flattened (CSV/Excel)
## Spreadsheets

### Simple data can be presented in a single sheet

<table>
<thead>
<tr>
<th>ocid</th>
<th>id</th>
<th>tender/id</th>
<th>tender/value/amount</th>
<th>contracts/0/id</th>
<th>contracts/0/value/amount</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Complex data can be presented in a multiple sheets

<table>
<thead>
<tr>
<th>ocid</th>
<th>id</th>
<th>tender/id</th>
<th>tender/value/amount</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

[Open Contracting Data Standard]
Individual files are important for developers and tools:

- Persistent URIs for individual releases and records

Bulk files make your data accessible to a wider range of users. Consider:

- Periodic generation and segmentation, by date, release ID etc.
- Dynamic generation: streaming the latest database contents
APIs

- Individual releases and records at **stable URLs**
- All releases and records ordered by date
- **Search** endpoints
- Aggregation
- Don’t forget **documents**

**resource:**

API specification

informed by user needs
Change history
Change history: releases and records

**Releases:**
Updates on individual events in a process, e.g:

- Tender notice
- Contract award
- Contract amendment

**Records:**
Current state of the whole process.
+ Links to individual releases

Open Contracting Data Standard
We recommend publishing a **new release** of data whenever anything changes in the contracting process.

**Worked example**

15th Jan
Tender issued

28th Feb
Award announced

15th March
Contract signed

**Option 1:** Publish **backward looking** data after contract is signed;

**Option 2:** Publish a single **record**, updating every time things change;

**Option 3:** Publish **individual releases** as each event happens, along with an OCDS record to compile a snapshot of the whole process.