



# ATDIS-1.0.2 Application Tracking

---

## *Data Interchange Specification*

Version	1.0.2 (145)
Status	Published
Printed	21/03/2014 10:15 AM
Reference	ATDIS-1.0.2 Application Tracking Data Interchange Specification (v1.0.2).doc
Prepared by	Carpadium Pty Ltd <a href="http://www.carpadium.com">www.carpadium.com</a>

## Table of Contents

Table of Contents .....	2
Figures.....	5
<b>1 Context.....</b>	<b>6</b>
1.1 Background.....	6
1.1.1 Planning reform.....	6
1.1.2 Application tracking.....	6
1.2 Revisions.....	6
1.3 Definitions.....	6
1.4 About this document .....	7
1.5 Specification implementation.....	7
1.5.1 Steps.....	7
1.5.2 Feedback.....	7
<b>2 About the specification .....</b>	<b>8</b>
2.1 Introduction .....	8
2.2 Examples.....	8
2.2.1 Scenario 1: Get a list of all active development applications from a local government area .....	8
2.2.2 Scenario 2: Get a list of active development applications within a specific postcode.....	10
2.2.3 Scenario 3: Get a list of active development applications within a specific suburb .....	10
2.2.4 Scenario 4: Get a list of active development applications within a specific street.....	11
2.2.5 Scenario 5: Combining search queries.....	11
2.3 Goals and objectives .....	12
2.3.1 What the ATDIS interchange specification is.....	12
2.3.2 What the ATDIS interchange specification is not.....	12
2.4 Approach .....	12
2.5 Success criteria .....	12
2.6 Complying with the specification.....	13
2.7 Considerations .....	13
<b>3 Conceptual architecture.....</b>	<b>14</b>
3.1 Roles and responsibilities .....	15
<b>4 ATDIS-1.0.2 Application Tracking Data Interchange Specification .....</b>	<b>16</b>
4.1 Components of the specification .....	16
4.2 Feed.....	17
4.3 Schema.....	18
4.3.1 Info [Mandatory].....	19
4.3.2 Reference [Mandatory].....	20
4.3.3 Locations [Mandatory].....	21
4.3.4 Events [Mandatory] .....	22
4.3.5 Documents [Mandatory].....	22
4.3.6 People [Optional] .....	22
4.3.7 Extended [Optional].....	23
4.3.8 Date Formats [L1] .....	23
4.4 Other Technical Considerations .....	23

4.4.1	Encoding .....	23
<b>5</b>	<b>Use Cases.....</b>	<b>24</b>
5.1	Operational Use Cases.....	24
5.1.1	List all development applications for [Council].....	25
5.1.2	List all development applications for [Council] for [Period].....	25
5.1.3	List all development applications for [Council] in [Address].....	25
5.1.4	Get more information about development application [Id].....	26
5.1.5	List all development applications .....	26
5.1.6	Notes on paging.....	26
5.2	Specification Use Cases.....	27
5.2.1	Create ATDIS specification .....	27
5.2.2	Manage ATDIS specification lifecycle .....	27
5.2.3	Comply with ATDIS specification.....	27
5.2.4	Produce compliant tracking data.....	27
5.2.5	Implement ATDIS specification.....	27
5.2.6	Deploy ATDIS-compliant software.....	28
<b>6</b>	<b>Distribution Channels .....</b>	<b>29</b>
6.1	REST/JSON [Mandatory] .....	29
6.2	REST/XML [Optional] .....	29
6.3	Web standards and accessibility [Mandatory].....	29
6.4	Paging.....	30
<b>7</b>	<b>Compliance.....</b>	<b>32</b>
7.1	Compliance requirements.....	32
7.1.1	General.....	32
7.1.2	Schema.....	32
7.1.3	Use cases .....	32
7.1.4	Distribution channels.....	33
7.2	Compliance and certification process .....	33
7.3	Obtaining certification.....	33
7.4	Using "ATDIS-1.0.2 Data Interchange Specification Compliant" in Vendor software marketing ..	34
<b>8</b>	<b>Lifecycle.....</b>	<b>35</b>
8.1	Version history.....	35
8.2	Changed in this version .....	35
8.2.1	Data items .....	35
8.2.2	Use cases .....	35
8.2.3	Distribution channels.....	35
8.3	Deprecated in this version .....	35
8.3.1	Data items .....	35
8.3.2	Use cases .....	35
8.3.3	Distribution channels.....	35
8.4	Planned for deprecation in the next version.....	35
8.4.1	Data items .....	35
8.4.2	Use cases .....	35
8.4.3	Distribution channels.....	35
8.5	Planned for addition in the next release.....	36
8.5.1	Data items .....	36
8.5.2	Use cases .....	36

8.5.3	Distribution channels.....	36
8.6	Notes and clarifications.....	36
<b>9</b>	<b>Appendix 1: Future possibilities .....</b>	<b>40</b>
9.1	Subscription and publication of ATDIS events.....	40
9.2	Extended/enhanced attribute publication .....	40
<b>10</b>	<b>Appendix 2: Examples of compliant ATDIS-1.0.2 requests and responses .....</b>	<b>41</b>
10.1	Obtain a list of all applications .....	41
10.2	Obtain a list of all applications for a set of postcodes .....	42
<b>11</b>	<b>Appendix 3: Carpadium Pty Ltd .....</b>	<b>44</b>
11.1	About Carpadium .....	44
11.2	Carpadium's role in the specification process.....	44

## Figures

<i>Figure 1: Conceptual architecture .....</i>	<i>14</i>
<i>Figure 2: Components of the specification.....</i>	<i>16</i>
<i>Figure 3: Feed web address format.....</i>	<i>17</i>
<i>Figure 4: Feed structure.....</i>	<i>18</i>
<i>Figure 5: Operational use cases.....</i>	<i>24</i>
<i>Figure 6: Specification use cases.....</i>	<i>27</i>

# 1 Context

## 1.1 Background

### 1.1.1 Planning reform

In April 2013, the NSW State Government released the [White Paper: A New Planning System for NSW](#) along with supporting draft legislation. Chapter 4 of the White Paper: Community Participation, informs the community of the direction for ePlanning in NSW. One of the ePlanning deliverables under the White Paper is making Application Tracking universally available to the community online.

Legislation for the planning reforms is now before the NSW State Parliament and successful passage is expected during the 2014 calendar year.

### 1.1.2 Application tracking

The main objective of the Application Tracking project is to help every Council in NSW work with a software vendor of their choice to implement an electronic application tracking tool that can extract application information from their electronic management systems in a machine readable format for people across NSW to access online. Making this information universally and easily available online requires a common standard for recording and extracting application tracking information from council electronic management systems.

The purpose of this document (version 1.0.2) is to provide vendors of application tracking software with an Application Tracking Data Interchange Specification to facilitate a consistent approach to application tracking across all Councils in NSW.

Specific goals for this project are:

1. Implement a common machine-readable standard for data produced by application tracking tools.
2. Adopt a minimum set of fields that a vendor's product must be able to produce.
3. Adopt a best-practice set of fields that a Council should be aiming for.
4. Make it easy for other parties to consume the data produced by the tools deployed under (1, 2 and 3).

## 1.2 Revisions

Ver	Date	Who	Notes
1.0.2	2014-03-21	Matthew Sinclair, Carpadium Pty Ltd	Published version (this document)
1.0.1	2014-03-04	Matthew Sinclair, Carpadium Pty Ltd	Updates based on feedback from stakeholders (final draft)
1.0.0	2014-01-21	Matthew Sinclair, Carpadium Pty Ltd	Initial Version (working draft)

## 1.3 Definitions

- Agency: [Planning and Infrastructure](#)
- AT: application tracking
- DIS: data interchange specification
- Specification: this document
- Specifying Authority: Planning and Infrastructure acting to create and manage the ATDIS-1.0.2 data interchange specification.
- Complying Authority: the Local Government Area or Municipal Council that processes development applications and makes them available in compliance with the ATDIS-1.0.2 data interchange specification.

- LGA: Local Government Area

When describing a component for compliance, the Specification will make use of one or more of the following key words: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL". The key words are to be interpreted as described in [RFC 2119](#).

## 1.4 About this document

The first part of this document presents a narrative explanation of the Specification and its purpose. The second part of the document describes the specification in detail using formal language suitable for use in compliance verification.

## 1.5 Specification implementation

### 1.5.1 Steps

1. Prepare draft Specification [Carpadium]
2. Issue draft Specification to Vendors for review and feedback [Agency]
3. Provide feedback on draft Specification [Vendors]
4. Collate and integrate feedback into Final Draft Specification [Carpadium]
5. Issue Final Draft Specification to Vendors for final review and feedback [Agency]
6. Publish ATDIS-1.0.2 Specification for pilot implementation [this document]
7. Update application tracking software to comply with this Specification [Vendors]
8. Implement this application tracking Specification [Councils, Vendors]

### 1.5.2 Feedback

Please address any feedback on this version of the Specification to [info@carpadium.com](mailto:info@carpadium.com) with the subject "ATDIS-1.0.2 Data Interchange Specification" in the subject line.

## 2 About the specification

### 2.1 Introduction

The Agency commissioned Carpadium to prepare this Application Tracking Data Interchange Specification with the aim of giving residents of NSW the ability to track the status of development applications online in a consistent, timely and complete manner, with a view to enabling a bird's eye view of development applications across the State of NSW.

This specification is concerned with providing online access to publically available application tracking *data* in accordance with this specification. It is not a specification for application software nor is it an attempt by The Agency to prescribe or proscribe any application functionality or seek in any way to control or specify the *software functionality* of application tracking systems offered by Vendors.

This version of the Specification [1.0.2, revision 145] has now been through a review cycle within The Agency and with Vendors. It is now being published as a first release for implementation.

### 2.2 Examples

The following examples describe some of the simplest and most common interactions that the specification hopes to foster. In each case, the important details relate to the *business scenario* triggering the event, and the *data* captured and interchanged in that process.

The examples assume familiarity with local government development planning tracking processes, as well as standard Internet protocols.

#### 2.2.1 Scenario 1: Get a list of all active development applications from a local government area

The simplest scenario that the specification seeks to guide is the process of obtaining a list of currently active development applications from a specific local government area (LGA).

To satisfy this business scenario, a Resident visits the public website of the LGA or an alternative 3<sup>rd</sup> party service provider, and clicks on a link to present the development tracking list. Within the LGA's software system a query executes to generate a list of currently active development applications, which the vendor's software system formats for display in the Resident's browser.

It is important to stress that the ATDIS-1.0.2 Specification does not seek to control *how* the Vendor's software stores or processes development planning information, the business work flows used to create it, or even how the information is visually styled and displayed on the screen. However what the Specification does concern itself with is the *data items* that make up the information displayed.

For example, the following<sup>1</sup> presents an indication of the types of data items and their naming conventions for a single development application:

```
{
  "response": [
    {
      "application": {
        "info": {
          "dat_id": "DA2013-0381",
          "last_modified_date": "2013-04-20T02:01:07Z",
          "description": "New pool plus deck",
          "authority": "Example Council Shire Council",
          "lodgement_date": "2013-04-20T02:01:07Z",
          "determination_date": "2013-06-20T02:01:07Z",
          "notification_start_date": "2013-04-20T02:01:07Z",
          "notification_end_date": "2013-05-20T02:01:07Z",
          "status": "OPEN"
        },
        "reference": {
          "more_info_url": "http://www.exemplecouncil.nsw.gov.au/atdis/1.0/applications/DA2013-0381"
        },
        "locations": {
          [
            {
              "address": "123 Fourfivesix Street Neutral Bay NSW 2089",
              "land_title_ref": {
                "lot": "10",
                "section": "ABC",
                "dpsp_id": "DP2013-0381"
              }
            }
          ]
        }
      },
      {
        "application": {
          ...
        }
      }
    ],
    "count": 2,
    "pagination": {
      "previous": null,
      "next": null,
      "current": 1,
      "per_page": 25,
      "count": 100,
      "pages": 1
    }
  ]
}
```

The way in which this data is rendered in a browser from the LGA's web site is entirely up to the LGA and their software vendor, but the hope is that an ATDIS-1.0.2-compliant system would be able to provide the above data items as a minimum set.

Importantly, the Specification seeks to provide for *programmatic* access to this kind of data for consumption by other systems, residents and applicants, as well as The Agency and content aggregators.

---

<sup>1</sup> Note: Responses must be pagination-aware, please see 6.4 for more details.

The preferred mode of system-to-system integration is via standard, open web protocols. For example, an HTTP request for a URL of the following form could yield the above application data:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json  
RES: 200 OK
```

The exact structure of the URL is defined later in the specification, but the important point to note here is that the URL represents an address to which a remote system can make a standard HTTP(s) GET request, with the expectation that a list of development applications will be returned in a machine readable format (in this case JSON, but XML would be equally appropriate). The basic interaction described here is the same for all use cases indicated within the Specification.

In summary, a remote system makes a request using standard HTTP protocols to an address in a known format, and the system returns a machine-readable data set for consumption by the remote system.

### 2.2.2 Scenario 2: Get a list of active development applications within a specific postcode

A simple refinement of Scenario 1 is to qualify the request so that the LGA system returns a subset of the available data. For example, a Resident might wish to know about all development applications within a specific postcode:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?postcode=2089  
RES: 200 OK
```

The above request contains a *query string* ("postcode=2089") that informs the LGAs system that instead of returning all development applications, that it should only return applications for the postcode 2089. The format of the returned data would be identical to that in Scenario #1, with the expectation that only those applications in the specified postcode would be returned.

A client system could also make a request for multiple postcodes, for example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?postcode=2089,2090  
RES: 200 OK
```

This is useful for when a local government area occupies multiple postcodes, or for when a resident might be interested in applications at the boundary of more than one postcode.

### 2.2.3 Scenario 3: Get a list of active development applications within a specific suburb

Client systems should be able to request a list of applications within a specific suburb. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?suburb=Mosman  
RES: 200 OK
```

The above request contains a *query string* ("suburb=Mosman") that informs the LGAs system that instead of returning all development applications, that it should only return applications for where the suburb indicated in the application address is equal to "Mosman". The format of the returned data would be identical to that in Scenario #1, with the expectation that only those applications in the specified suburb would be returned.

A client system could also make a request for multiple suburbs, for example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?suburb=Mosman,Manly
RES: 200 OK
```

As with Scenario #2, this form is useful for when a local government area occupies multiple suburbs, or when a resident might be interested in applications at the boundary of more than one suburb.

#### 2.2.4 Scenario 4: Get a list of active development applications within a specific street

Client systems should be able to request a list of applications where the address matches a specific street name. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?street=Military
RES: 200 OK
```

The above request contains a *query string* ("street=Military") that informs the LGAs system that instead of returning all development applications, that it should only return applications for where the street name indicated in the application address contains the value "Military". The format of the returned data would be identical to that in Scenario #1, with the expectation that only those applications where the specified street name occurs would be returned.

A client system could also make a request for multiple streets, for example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?street=Military,Cardinal
RES: 200 OK
```

As with Scenario #2, this form is useful for when a local government area occupies multiple suburbs, or when a resident might be interested in applications in a number of nearby streets.

Note that the street form of the search should look for the supplied street name in *any part* of the street address of the application.

#### 2.2.5 Scenario 5: Combining search queries

Client systems should be able to combine street, suburb and postcode searches in a single query. LGA systems should return the subset of applications that contain addresses matching the search criteria. For example:

```
REQ: GET
http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?street=Military,Cardinal&postcode=2088
RES: 200 OK
```

The above query should return applications where the street name of the application is either "Military" or "Cardinal" and the postcode is "2088".

## 2.3 Goals and objectives

In creating the Specification, The Agency seeks to:

1. Create an interchange format for the data associated with development applications that allows publicly available information to be published in a consistent fashion to foster integration and aggregation
2. Give the residents of NSW confidence that their Councils are collecting and publishing relevant application tracking data in a consistent, timely and complete manner
3. Facilitate creation of a bird's eye view of development applications across NSW

### 2.3.1 What the ATDIS interchange specification is

- It is a specification that codifies the types of data that Council has to capture and make available for residents
- It is a specification for a data interchange format for application tracking
- It is designed to make publishing application tracking data simple and easy for system-to-system integration

### 2.3.2 What the ATDIS interchange specification is not

- It is not a specification for a software application
- It is not a specification for application functionality
- It is not a solicitation for provision of software or services

## 2.4 Approach

The Agency has sought to collaboratively develop the Specification with interested parties including councils and vendors of software systems that offer tracking capabilities.

Specifically, The Agency hopes to:

1. Establish a State-wide data interchange standard for the online publication of data associated with development applications
2. Work with vendors of municipal software systems to deliver application tracking tools for Councils that are interoperable according to Planning & Infrastructure standards

## 2.5 Success criteria

To be successful, this specification must:

1. Achieve industry wide consensus from all stakeholders on the Specification, including:
  - Schema: the format for publicly available mandatory, optional and extended data items published by application tracking systems
  - Use Cases: an agreed minimum set of expected uses for published application tracking data
  - Channels: an agreed minimum set of channels and data syntax models over which application tracking data can be published
2. Achieve sign-off of version 1.0 of the Specification from The Agency

## 2.6 Complying with the specification

In order to comply with the Specification, Complying Authorities must work with ATDIS Vendors and their own IT infrastructures to publish information about development applications so that Applicants and other Residents within NSW can access that information online through browser or mobile devices.

The Agency acknowledges that there are different levels of technical capability within councils and varying degrees of maturity of the software systems employed to manage application tracking.

This version of the specification captures the basic set of data items necessary for meaningful display of application tracking data. The specification captures the baseline set of data items that all Councils with application tracking software should be able to meet, and defines an interchange format that is both simple and easy for Residents and Applicants to consume, and suitable for Aggregators to work with. The Agency believes that this specification captures *best practice* for the management of online application tracking data.

To comply with the Specification, a Complying Authority's application tracking system must be able to provide an online *feed* of the development applications processed by the Complying Authority that:

1. Contains each of the *mandatory* data items,
2. Supports consumption according to the *mandatory* use cases, and
3. Presents data across each of the *mandatory* distribution channels,

as defined in the ATDIS-1.0.2 version of the Specification.

In the context of compliance, a *feed* is a public data set that is available for consumption by a person using a desktop or mobile browser and Internet protocols such as IP and HTTP(s), or by a system capable of interacting with the feed using the same Internet protocols.

Unless otherwise specified as part of a query, application data returned from a compliant system should sort the results in order of *most recent applications first*, using the application date as the sort field.

## 2.7 Considerations

Based on feedback from stakeholders, the following list of considerations is relevant for Vendors and Councils wishing to develop compliant systems:

1. *Authentication*: The Specification introduces no authentication requirements and makes the assumption that compliant data is available via a public, unauthenticated link.
2. *Throttling*: Vendors and councils are free to implement throttling against a *reasonable use* policy. The definition of *reasonable use* is beyond the scope of the specification, and left to individual LGAs to define a suitable policy appropriate for their scale and demand.

### 3 Conceptual architecture

Figure 1 presents the conceptual architecture for the Specification:

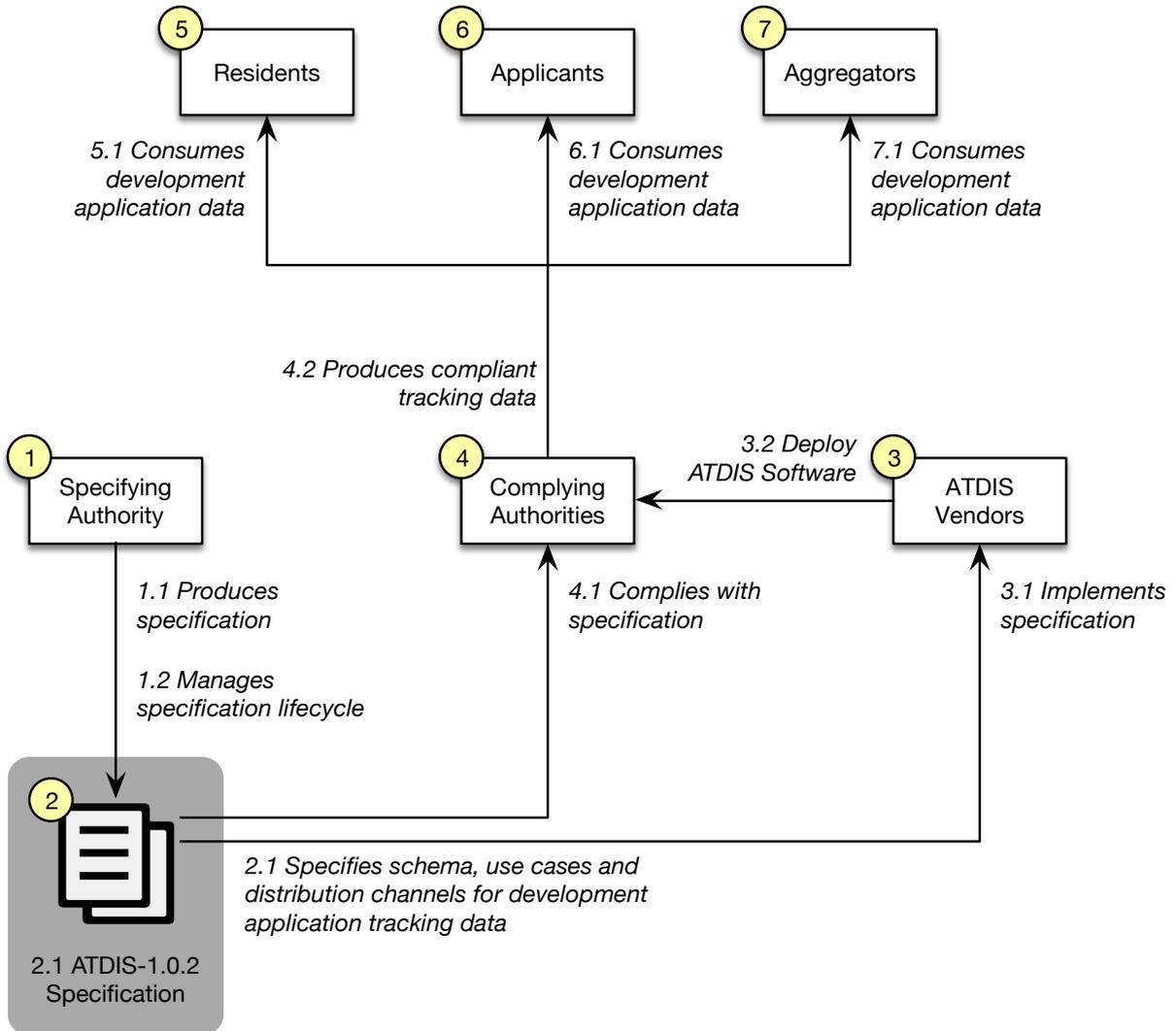


Figure 1: Conceptual architecture

The conceptual architecture defines the high-level entities and processes that cooperate within the specification. This Specification document is principally concerned with defining the *schema*, *use cases* and *distribution channels* for the ATDIS-1.0.2 data interchange format.

Entities and activities are captured in the conceptual architecture to provide context.

### 3.1 Roles and responsibilities

This specification attempts to coordinate the *publicly available data* captured and produced by application tracking systems, and in doing so, necessitates defining responsibilities for a number of roles within the ecosystem

The Specification defines the following *roles*:

- **Specifying Authorities:** Planning & Infrastructure acts to create the Specification for application tracking data, and manage the lifecycle of change to the specification over time.
- **Complying Authorities:** Councils work to comply with the publication standard codified in the Specification and use ATDIS Vendor software to produce compliant tracking data.
- **ATDIS Vendors:** implement the specification in their municipal systems and deploy to NSW Councils.
- **Residents:** consume application tracking data for other applicants via online and/or mobile channels.
- **Applicants:** consume application tracking data for their own applications and the applications of other residents via online and/or mobile channels.
- **Aggregators:** consume application tracking data and aggregate according to market requirements.

The Specification captures a number of *responsibilities* that indicate how each role participates in the inputs and outputs of the ATDIS-1.0.2 process:

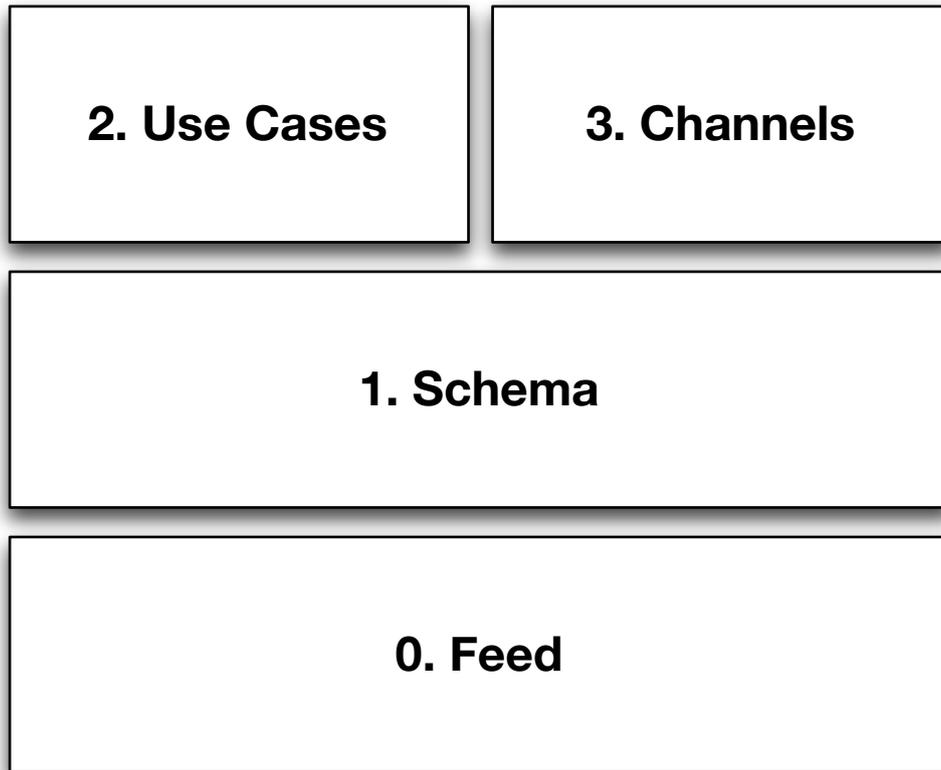
1. Specifying Authority
  - a. Produces specification
  - b. Manages specification lifecycle
2. ATDIS-1.0.2 Specification
  - a. Specifies schema, use cases and distribution channels for application tracking data
3. ATDIS Vendor
  - a. Implements Specification
4. Complying Authority
  - a. Complies with Specification
5. Resident
  - a. Consumes application tracking data
6. Applicant
  - a. Consumes application tracking data
7. Aggregator
  - a. Consumes application tracking data

Note: It is well understood that participants in this ecosystem have other roles and perform a variety of other responsibilities that are beyond the scope of this Specification. These roles and responsibilities are described for clarity within this Specification, and are not meant to constrain the operation of those entities in activities beyond the scope of the Data Interchange Specification.

## 4 ATDIS-1.0.2 Application Tracking Data Interchange Specification

### 4.1 Components of the specification

Figure 2 presents the four high-level components of the Specification:



**Figure 2: Components of the specification**

- **Feed:** defines a publicly available source of application tracking data. For example, each complying Council would provide a single feed for application tracking data.
- **Schema:** defines the data elements that make up individual application tracking records. Examples include "date", "location" and "reference".
- **Use cases:** defines the agreed use cases for which ATDIS data can participate. Examples include "enquiries" and "subscriptions".
- **Channels:** defines the channels over which ATDIS data can be delivered. Examples include RSS, REST/[JSON, XML] and Browser.

## 4.2 Feed

Any compliant source of application tracking data is referred to as a *feed*. A feed is defined by a standard web address of the form:

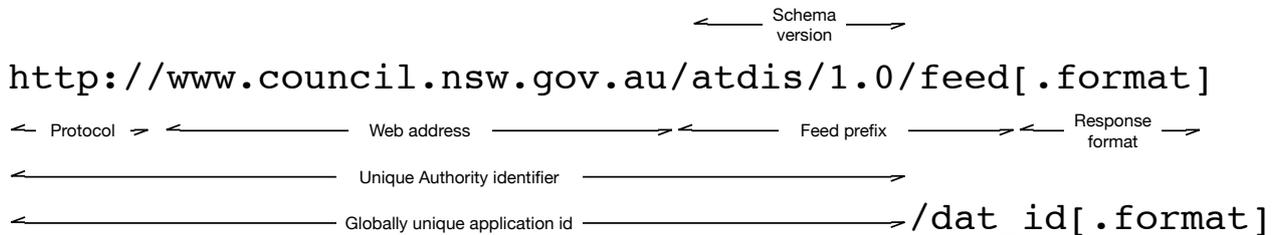


Figure 3: Feed web address format

### Where

- **Protocol:** assume HTTP, but HTTPS can be used by a Council if required
- **Web address:** the fully qualified web host name for the Complying Authority. By convention, this should be the same as the URL used to access the Council's publicly available web site.
- **Feed prefix:** component of the URI that indicates a complying feed.
- **Schema version:** component of the feed prefix that indicates the version of the schema offered by the feed.
- **Unique Authority identifier:** The UAI uniquely identifies an authority for applications published by an ATDIS-compliant authority. This allows for applications within one LGA to be referenced uniquely by documents within another authority without the need to ensure that `dat_id` values are unique across all participants.
- **Globally Unique DAT Id:** When combined with the UAI, the `dat_id` for an application creates a globally unique identifier for an application that is portable between LGA jurisdictions.

### Notes

- There are a number of options for how the schema version number might be encoded into the feed URL. The above format (where the version is encoded into the URI) is the simplest mechanism, but it is also possible to encode version number with an HTTP header variable, with a query string parameter, or by using content negotiation. There is a vibrant and passionate debate about the best way to version URLs for access to resources that shows no prospect of being settled soon. Therefore, for the purposes of ATDIS-1.0.2, a simple URI-encoded version string is proposed. If content negotiation becomes the preferred standard, then this it is relatively easy to change the prefix in a future release.
- Given a version of the specification of the form *MAJOR.minor.patch*, it is assumed that all *patch* versions of the same major/minor version remain semantically and syntactically compatible with each other. Minor may introduce new items but will not remove any existing items. Major versions may make changes to the schema that are not backwards compatible.

### 4.3 Schema

The *schema* defines the specific data elements that are published in a compliant ATDIS-1.0.2 *feed*. The schema has a defined structure, broken down into one or more *application* records, which are further decomposed into *blocks* and *fields*.

To comply with the Specification, a feed must support the basic conceptual structure of a **list** of application records, within which the feed will provide blocks and fields for each application record.

Record types can be **mandatory** or **optional**. Within a record, fields can be **mandatory** or **optional**.

A feed *must* include all **mandatory** record types. A feed *may* include any of the **optional** record types.

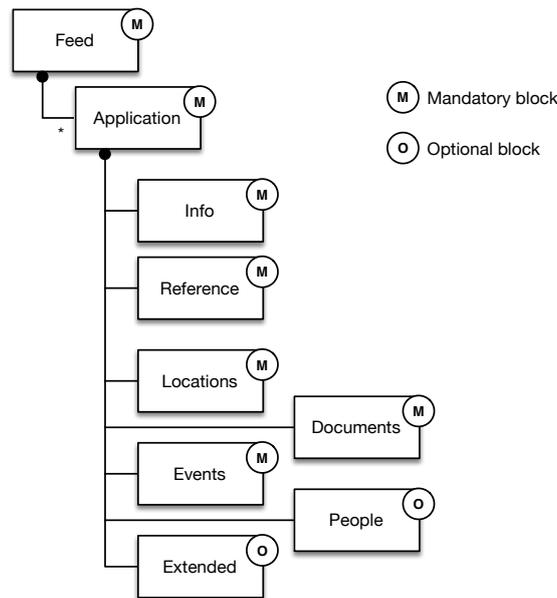
For each mandatory record type, the feed *must* include all **mandatory** fields, and *may* include any of the **optional** fields.

If a feed chooses to include a record type then it must include all mandatory fields for that record type. A feed can chose whether or not to include any optional fields within a block.

If a field is empty, then it should be populated with the appropriate JSON null value. For example:

```
{"field": null}
```

Figure 4 describes the conceptual structure of a complying feed:



**Figure 4: Feed structure**

There are *seven* record types defined within the Specification:

1. Info
2. Reference
3. Locations
4. Events
5. Documents
6. People
7. Extended

The following sections describe the semantics of each of the seven record types that make up the Specification. For each record type, the Specification indicates whether the record type is mandatory or optional, and within each record type, which fields are mandatory or optional.

#### 4.3.1 Info [Mandatory]

The *info* record is the top-level block that contains unique identifying information about a single development application. The info record must contain the following fields:

M/O	Field	Notes
M	dat_id	An id that uniquely identifies the application within this authority
M	development_type	Relevant category requested by Planning and Infrastructure for the Local Development Performance Monitoring program
M	application_type	The Agency needs to collect statistics on types of application so that it can assess the effectiveness of different assessment pathways. Valid types are: DA, CDC, S96, Review, Appeal, Other.
M	last_modified_date	The date of the last change to this record
M	description	A short, human readable description of the application
M	authority	Identifying information for this Authority
M	ref	The unique Authority identifier (UAI) for this authority
M	name	Human readable Authority name
M	lodgement_date	Date the application was lodged with the Authority
M	determination_date	Date that the application was determined at the Authority
M	determination_type	One of the determination types specified below
M	status	Current status of the application with the Authority
O	notification_start_date	Start date for notification period of application
O	notification_end_date	End date for notification period of application
O	officer	Name of responsible officer from the Authority
O	estimated_cost	Estimated cost of the work described by the application
O	related_apps	List of fully qualified authority/ref/dat_id for related applications

#### Notes

- If the field `dat_id` contains non-URL characters in its raw form, then it must be URL-encoded so that it can form part of a valid URL.
- The value `development_type` should be the relevant category requested by Planning and Infrastructure for the Local Development Performance Monitoring program. ATDIS seeks to reduce Council manual work effort spent each year extracting, collating and reporting DA data for the annual Local Development Performance Monitoring report prepared by Planning and Infrastructure. In this context, `development_type` is a key piece of data needed for this report.
- The field `last_modified_date` captures the date at which the authority last modified the record. This allows consuming systems to know if an application changes over time. The `last_modified_date` field in the info record should capture the date that the application was last changed in the underlying source system. Vendors are responsible for determining which internal date in their system is the relevant field to use to populate this value. Conceptually, the `last_modified_date` field is the business concept that represents the date/time that the application was changed.
- Dates can be specified with or without time information. For example, the following are both valid date values (see also 4.3.8):  

```
"determination_date": "2013-06-20T02:01:07Z"
```

```
"determination_date": "2013-06-20"
```
- If the application does not have a determination date (yet, or at all), the value "null" should be specified for the `determination_date` field.
- The `notification_start_date` and `notification_end_date` are optional fields. Vendors are free to populate this value if it is available in the source system.

- The contents of `determination_type` and should be one of the values:
  - Pending
  - Refused by Council
  - Refused under delegation
  - Withdrawn
  - Approved by Council
  - Approved under delegation
  - Rejected
- If the application does not have a notification period at all, the value “null” should be specified for the `notification_start_date` and `notification_end_date` fields.
- The contents of `estimated_cost` and should be a formatted currency string in Australian Dollars, of the form “\$300,000”.
- If an application is related to other applications either within this LGA or in another LGA, then `related_apps` may contain a list of references to other applications. The reference should include the `authority/ref` value for that LGA as well as the `dat_id` in each case. The reference should be a URI that includes the protocol and the globally unique application id followed by the `dat_id` and `.json`. The specification does not prescribe the *type* of the relationship, other than to say that if this field is included, consuming systems should assume that the other applications are related to this one in some form. The intent of this field is to capture the relationship between applications with different application numbers.
- If a consuming system accesses a specific record using the globally unique identifier, the System should return data for just the single application record that was asked, and no others.
- Any globally unique id used in `related_apps` should also include the format `(.json)` in the URL.

#### 4.3.2 Reference [Mandatory]

The *reference* record contains links to the original source material for the application at the Complying Authority. It can optionally contain a link for where Residents can comment on the application at the Complying Authority. In both cases, the links to URLs captured in the *reference* block would take the user to another location on the Complying Authority's web site.

The content behind the `more_info_url` and `comments_url` values must link to an endpoint that directly shows the information for the specific development application. The underlying content must be directly accessible without authentication, cookies or other limiting requirements for the consuming system. The data and behaviour of the pages at these links is beyond the scope of the ATDIS-1.0.2 Specification.

The *reference* record must contain the following fields:

M/O	Field	Notes
M	<code>more_info_url</code>	web address for more information about this application
O	<code>comments_url</code>	web address for commenting on this application

### 4.3.3 Locations [Mandatory]

The *locations* record is designed to provide consumers with information about the geographic locations of the development application. Because not all Authorities have access to geographic information for their development applications, parts of this record are optional. The feed must include the address and `land_title_ref` attributes within each location record.

It is mandatory that feeds provide the location record with the address and `land_title_ref` fields, and it is anticipated that future versions of the Specification will promote the `coordinates` field into mandatory compliance.

Because some applications cover multiple land parcels, the location record is comprised of a list of location blocks under a single `locations` list field.

M/O	Field	Notes
M	<code>locations</code>	Composite record containing list of locations related to this app
M	<code>address</code>	Human readable street address for the application
M	<code>street</code>	Street name for land parcel
M	<code>suburb</code>	Suburb name for land parcel
M	<code>postcode</code>	Postcode for land parcel
M	<code>state</code>	State for land parcel
M	<code>land_title_ref</code>	Composite record containing Land Title Office references
M	<code>torrens   other</code>	Type of land title reference: torrens   other
M	<code>lot</code>	Lot number
M	<code>section</code>	Section number, or "null"
M	<code>dpsp_id</code>	DP/SP identifier
O	<code>geometry</code>	Composite record containing geographic coordinates (GeoJSON format)

#### Notes

- The `locations` record must contain at least one `location` field, but can contain more than one.
- The contents of `land_title_ref` can be either `torrens` or `other`. In the case of `torrens`, the field must include a specification of `lot`, `section` and `dpsp_id`. In the case of `other` the specification can include whatever field values are appropriate for that type.
- In each case the values for `lot`, `section` and `dpsp_id` should all be captured as strings.
- When used, GPS coordinates must make use of the WGS84 datum as per "*Earth Gravitational Model 2008 (EGM2008)*". See: <http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/index.html> for details.
- The format for geographic coordinates must use the GeoJSON specification as per: <http://geojson.org/geojson-spec.html#geojson-objects>
- The simplest compliant version of a `geometry` instance would be a *Point* of the form:

```
{ "type": "Point", "coordinates": [100.0, 0.0] }
```

Point coordinates are in x, y order (easting, northing for projected coordinates, longitude, latitude for geographic coordinates). For more advanced geometric types, please refer to the GeoJSON specification.

- As `geometry` is optional for the Location block in ATDIS-1.0.2, there is no requirement to specify the kinds of geometric data for an application. However, if a feed is supplying geometric data, then it must be formatted according to the GeoJSON specification.
- In general, the `dpsp_id` will start with either the letters `SP` or `DP`, and be followed by a number, but there will be cases where this does not apply.
- A title reference does not always include a section number. If the `section` value in `land_title_ref` is not available, then it should be populated with the value "null".

- When referencing a subdivision, Vendors and Councils should use the original parcel street numbers and/or lot references. In the case of a subdivision, then the pre- subdivision data should be used.

#### 4.3.4 Events [Mandatory]

The *events* record contains a list of events that have occurred against the application since lodgement. The *events* record must include a list of events, where each event has an *id*, *date* and *description*. Inclusion of an *event\_type* and *status* is optional.

M/O	Field	Notes
M	events	Composite record containing a list of events for the application
M	id	Internal id of event raised against application
M	timestamp	Date+time the event occurred
M	description	Short, human readable description of the event
O	event_type	Authority-specific event type
O	status	Authority-specific event status code

#### Notes

- This version of the specification does not define a list of valid *event\_types*, so the contents of this field can be Vendor or Council specific.
- The value for *id* should be a *string*. The value should be unique from the perspective of the authority such that it can be used as a reference for the underlying event.
- The system should produce events that are ordered by [*timestamp*, *id*]. In systems where all events generated on the same day are given the same *timestamp*, then those systems should order the *ids* so that they are non-decreasing. Consuming systems should assume that ordering the events in terms of [*timestamp*, *id*] places them into the sequential order in which they occurred.
- In this context, the *events* record means any kind of activity that can occur against an application. For example, "tasks" and "workflows" would be synonyms for "events". The specification does not dictate the type of event, but rather, that if an event of any type is captured, then this is the format in which it needs to be articulated.

#### 4.3.5 Documents [Mandatory]

The *documents* record contains a list of references (as links) to documents that relate to the application. The *documents* record must include a list of documents, where each document has a *ref*, *title* and a *link* (in the form of a URL) to the location where the document can be downloaded.

M/O	Field	Notes
M	documents	Composite record containing a list of documents for the application
M	ref	Internal reference of document for this application
M	title	Short, human readable title of the document
M	document_url	URL to location of document for download

#### Notes

- If there are no documents available for a particular application then the record should be included, but with no document records specified.

#### 4.3.6 People [Optional]

The *people* record contains a list of people that relate to the application. As an optional record type, this block can be populated as required by an LGA. However, if it the people record is included at all, then it is assumed to

include a list of people, where each person has a name and a role indicating their responsibility with the application. Optionally, the person record can include contact information.

M/O	Field	Notes
M	people	Composite record containing a list of people related to the app
M	name	Name of person related to the application
M	role	Authority-specific role type for the person with respect to the app
O	contact	Authority-specific contact information for the related person

#### 4.3.7 Extended [Optional]

The *extended* record is specifically designed to allow ATDIS-1.0.2 compliant feeds to include any additional information that might be relevant to an application. The contents of the extended record are not specified, and left up to the ATDIS Vendor and Complying Authority to populate as they see fit. There are no mandatory or optional fields for the extended record. Consuming systems should expect its existence, but are not required to do anything with data contained within it.

M/O	Field	Notes
		Not specified

#### 4.3.8 Date Formats [L1]

Any field within the schema of the form \*\_date or timestamp should be treated as an ISO-8601 date format. See: <http://www.w3.org/TR/NOTE-datetime> for details.

### 4.4 Other Technical Considerations

#### 4.4.1 Encoding

All data in ATDIS requests should be encoded using UTF-8, and all data returned by ATDIS-compatible systems should be returned in UTF-8 encoding. See <http://en.wikipedia.org/wiki/UTF-8> for details.

## 5 Use Cases

The specification defines a number of use cases to assist with scoping. The pre-draft version considers two kinds of use cases:

- Operational use cases
- Specification use cases

Operational use cases describe the scope of use of systems that *comply with* the specification.

Specification use cases describe of the specification process itself. It is likely that the specification use cases will be deprecated and removed before release of the *final* version of the ATDIS-1.0.2 specification.

### 5.1 Operational Use Cases

Figure 5 presents the top-level *operational* use cases for ATDIS-1.0.2:

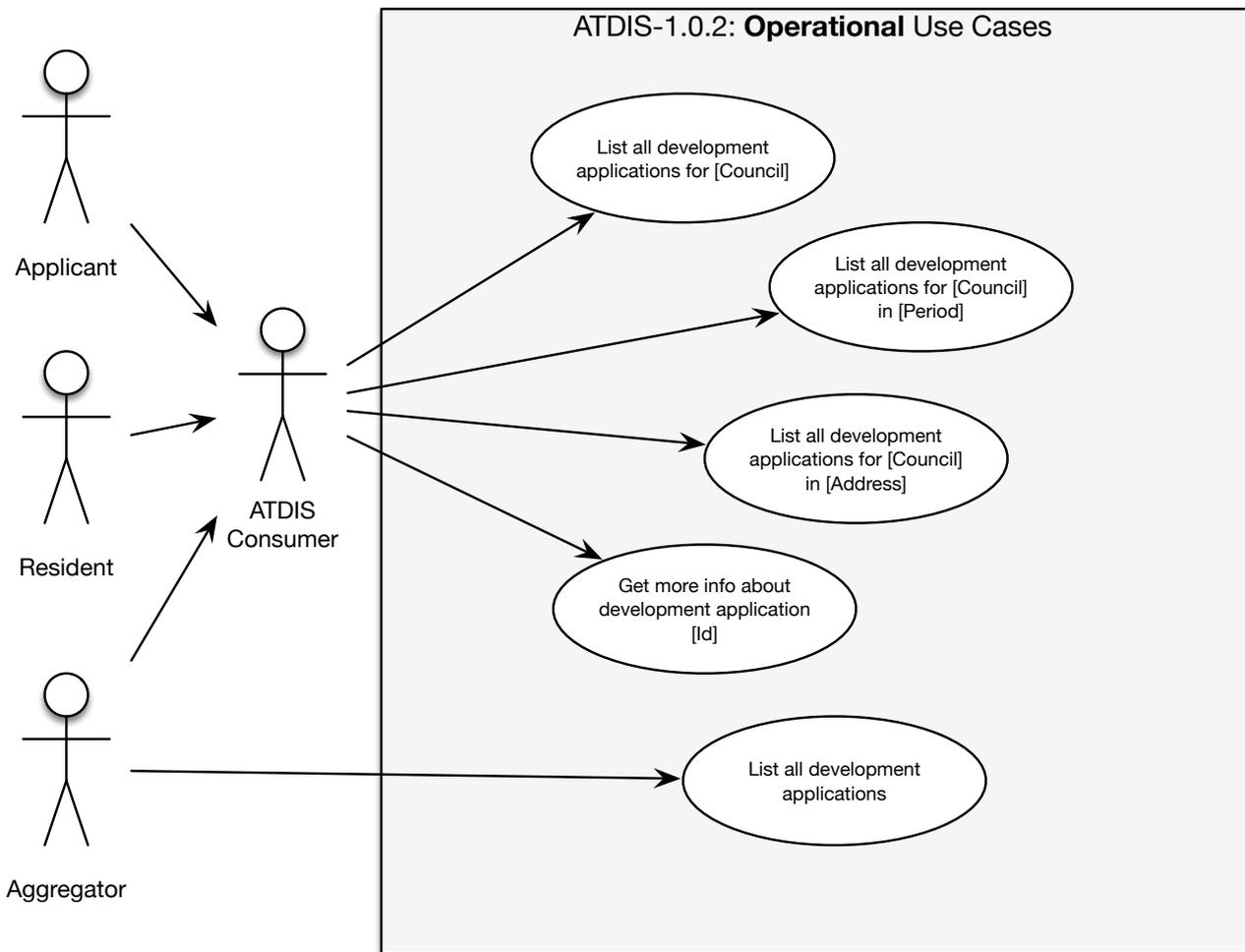


Figure 5: Operational use cases

### 5.1.1 List all development applications for [Council]

Allows an ATDIS consumer to return all development applications for the specified Council. In this context, the specified Council would be the LGA that is hosting Vendor software capable of exposing application tracking data. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json
RES: 200 OK
```

In this case, [Council] is specified by virtue of accessing the feed published by the Council at their web address ([www.examplecouncil.nsw.gov.au](http://www.examplecouncil.nsw.gov.au)).

### 5.1.2 List all development applications for [Council] for [Period]

This is a refinement of 5.1.1 that allows for specification of a period in which to select development applications. The results returned should only be for development applications within the period.

By default, the feed must support querying of the `lodgement_date` field (see 4.3.1). For example:

```
REQ: GET
http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?lodgement_date_start=YYYY-MM-DD&lodgement_date_end=YYYY-MM-DD
RES: 200 OK
```

The feed must also support querying by `last_modified_date` field (see 4.3.1). For example:

```
REQ: GET
http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?last_modified_date_start=YYYY-MM-DD&last_modified_date_end=YYYY-MM-DD
RES: 200 OK
```

If the `*_date_end` field is not specified in the query string, the feed should assume that the request is for the single date specified in the `last_modified_date_start` or `lodgement_date_start` fields.

### 5.1.3 List all development applications for [Council] in [Address]

This is a refinement of 5.1.1 that allows for a query to filter out only those development applications that occur within a specified address. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?postcode=2059
RES: 200 OK
```

To query across a number of postcodes, the consumer should separate multiple postcode values with a “,”:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?postcode=2059,2060
RES: 200 OK
```

To query across compound address parameters, include a value (or values) for street, suburb or postcode. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?street=Military,Cardinal&postcode=2088
RES: 200 OK
```

#### 5.1.4 Get more information about development application [Id]

Given information about a specific development application, the consuming system should be able to *drill into* a link to find more information. For example, if a Resident is accessing a Council's web site via a browser, clicking a *more info* link would provide detailed information about the specific development application.

#### 5.1.5 List all development applications

This is a use case for *future* consideration. The intent here is to allow a consumer to access *aggregated* application tracking data, which might come from a variety of LGAs across the state.

#### 5.1.6 Notes on paging

If a requester specifies a page number to a request, then the feed must return just that page of data, as opposed to the entire set of items for the basic request. Requests should be parameterised using the "page=n" qualifier in the query string. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?page=2
RES: 200 OK
```

In this case the feed should return the *second* page of data.

If the requester wishes to control the page size in items, it can be specified via the "count=k" qualifier in the query string. For example:

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?page=2&count=20
RES: 200 OK
```

It is valid to specify both page number and the number of items per page in the same query string.

## 5.2 Specification Use Cases

Figure 6 describes the specification use cases for ATDIS-1.0.2:

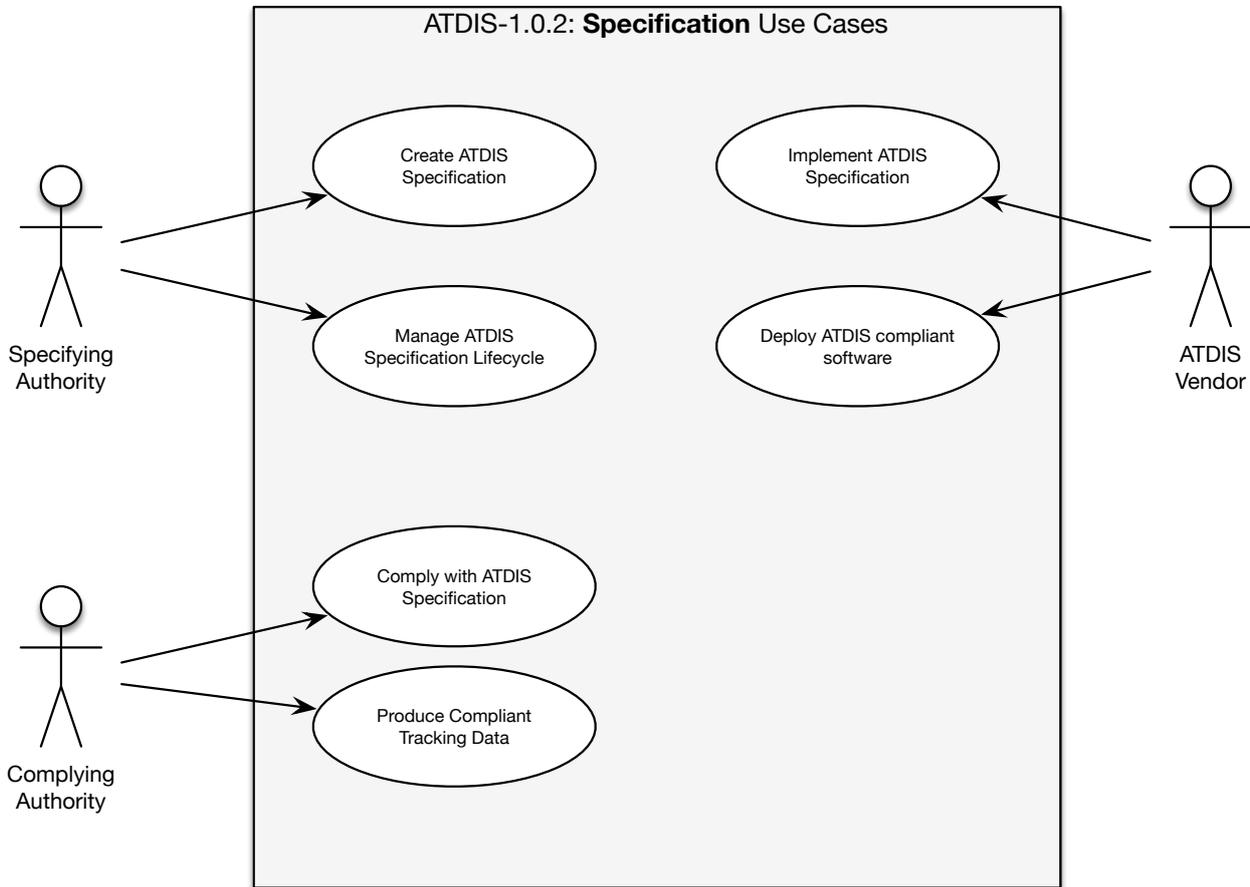


Figure 6: Specification use cases

### 5.2.1 Create ATDIS specification

This is the process by which the Specifying authority creates the specification. As of the pre-draft version of the document, the specification is a collaborative effort between The Agency and Vendors of application tracking software.

### 5.2.2 Manage ATDIS specification lifecycle

The Agency will manage publication of the specification and changes to it over time.

### 5.2.3 Comply with ATDIS specification

Complying Authorities (LGAs) collect application tracking data according to their preferred business processes and publish it using Vendor software according to the ATDIS-1.0.2 specification.

### 5.2.4 Produce compliant tracking data

LGAs produce tracking data by virtue of their daily activities operating as local Councils.

### 5.2.5 Implement ATDIS specification

Vendors use this specification to produce compliant software.



#### 5.2.6 Deploy ATDIS-compliant software

Vendors work with LGAs to deploy compliant software into their Council's IT infrastructure.

## 6 Distribution Channels

### 6.1 REST/JSON [Mandatory]

It is mandatory for compliance to produce a REST/JSON style feed of application tracking data.

### 6.2 REST/XML [Optional]

Vendors can *optionally* produce an XML formatted version of their feed.

Transformation between REST/JSON and REST/XML is technically straightforward.

### 6.3 Web standards and accessibility [Mandatory]

All browser-distributed feed data must be compatible with W3C standards for HTML5 and CSS3.

It is also a mandatory requirement for compliance with ATDIS-1.0.2 that systems conform to the Australian Government's **Web Content Accessibility Guidelines (WCAG) version 2.0** (WCAG2.0). Full details of these accessibility guidelines can be found here:

<http://webguide.gov.au/accessibility-usability/accessibility/>

## 6.4 Paging

In some circumstances, the data for a particular ATDIS query might be larger than is sensible to return in a single response. In this case, the feed can use *paging* to indicate to the client that the server has more data available. For example, a paged response to a request for all items within a postcode might look like this:

```
{
  "response": [
    {
      "application": {
        "info": {
          "dat_id": "DA2013-0381",
          "last_modified_date": "2013-04-20T02:01:07Z",
          "description": "New pool plus deck",
          "authority": "Example Council Shire Council",
          "lodgement_date": "2013-04-20T02:01:07Z",
          "determination_date": "2013-06-20T02:01:07Z",
          "notification_start_date": "2013-04-20T02:01:07Z",
          "notification_end_date": "2013-05-20T02:01:07Z",
          "status": "OPEN"
        },
        "reference": {
          "more_info_url": "http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications/DA2013-0381"
        },
        "locations": {
          [
            {
              "address": "123 Fourfivesix Street Neutral Bay NSW 2089",
              "land_title_ref": {
                "lot": "10",
                "section": "ABC",
                "dpsp_id": "DP2013-0381"
              }
            }
          ]
        }
      },
      ...
    },
    {
      "application": {
        ...
      }
    }
  ],
  "count": 2,
  "pagination": {
    "previous": null,
    "next": null,
    "current": 1,
    "per_page": 25,
    "count": 100,
    "pages": 1
  }
}
```

In this format, the returned data is split up into three sections:

- **response:** contains the raw data of the response, paged according to the pagination block
- **count:** the number of items returned in this response
- **pagination:** information about how the returned data was paged relative to the full set of data available for that query. Within the pagination block, the following items indicate *how* the data was paged:

- previous: the page number of the immediately preceding page, or null if this is the first page
- next: the page number of the immediately following page, or null if this is the last page
- current: the page number of the current page
- per\_page: the number of paged items returned per page
- count: the total number of items available from the underlying data, unpaged
- pages: the total number of pages available from the underlying data

## Notes

- Paged data is at the discretion of the feed producer.
- In all cases, the response block must be provided by the feed producer, regardless of whether or not the feed is producing paged data.
- In a feed cannot handle pagination, then it can leave out the count and pagination blocks

Because back-end systems page data in a variety of different ways, specification of paging has been set at L2 for the ATDIS-1.0.2 version of the specification. However, Vendors should be aware that the paging specification would be promoted to L1 for the next point release, and that all responses should include the response block when returning data to the consuming system.

## 7 Compliance

### 7.1 Compliance requirements

The following items represent the canonical set of requirements for ATDIS-1.0.2 compliant feed of application tracking data. These requirements will be used during testing as the high-level criteria for compliance.

In this context “The System” refers to a Vendor application tracking system installed at a Complying Authority such as a Council.

#### 7.1.1 General

1. The System must provide an Internet-accessible *feed* to application tracking data
2. The feed must be openly available to residents of NSW using standard Internet protocols such as HTTP or HTTPS.
3. The feed must be available in a browser-rendered format.
4. The feed must be available in machine-readable format.

#### 7.1.2 Schema

5. The feed must provide a list of application tracking records.
6. The feed must support the data interchange schema as defined in section 4.3.
7. The feed must support the following mandatory record types:
  - a. Info
  - b. Reference
  - c. Location
  - d. Events
  - e. Documents
8. The feed may support the following optional record types:
  - a. People
  - b. Extended
9. The System must populate the *mandatory* fields for the following record types:
  - a. 4.3.1 Info
  - b. 4.3.2 Reference
  - c. 4.3.3 Location
  - d. 4.3.4 Events
  - e. 4.3.5 Documents
  - f. 4.3.6 People
10. The System may populate the *optional* fields for the following record types:
  - a. 4.3.1 Info
  - b. 4.3.2 Reference
  - c. 4.3.3 Location
  - d. 4.3.4 Events
  - e. 4.3.5 Documents
  - f. 4.3.6 People
  - g. 4.3.7 Extended

#### 7.1.3 Use cases

11. The System must support the following operational use cases:
  - a. 5.1.1 List all development applications for [Council]
  - b. 5.1.2 List all development applications for [Council] in [Period]
  - c. 5.1.3 List all development applications for [Council] in [Address]
  - d. 5.1.4 Get more info about development application [Id]
12. The System may support the following operational use cases:

- a. 5.1.5 List all development applications
- 13. The System must support the following specification use cases:
  - a. 5.2.3 Comply with ATDIS specification
  - b. 5.2.4 Produce compliant tracking data
  - c. 5.2.5 Implement ATDIS specification
  - d. 5.2.6 Deploy ATDIS-compliant software

Note, the following specification use cases are included in the requirements, but Vendors are not required to implement them in the System:

- e. 5.2.1 Create ATDIS specification
- f. 5.2.2 Manage ATDIS specification lifecycle

#### 7.1.4 Distribution channels

- 14. The System must provide a machine-readable feed in REST/JSON format.
- 15. The System may provide a machine-readable feed in REST/XML format.
- 16. The System must provide a feed available to Internet browser connections.
- 17. The browser feed must conform to the NSW Government's Web Content Accessibility Guidelines (WCAG) version 2.0.
- 18. The System may provide an RSS-style feed. If the System provides an RSS feed, then data must be provided according to the ATDIS schema.
- 19. The system should allow clients to consume data in *pages*. If a system offers paging, it must indicate paged data by populating the count and pagination properties of the response.

## 7.2 Compliance and certification process

In conjunction with this Specification, The Agency will define test harness platform suitable for verifying that a particular application tracking system complies with the Specification. The line items in section 7.1 describe the individual requirement items that the test harness will verify.

At a mutually agreeable time between The Agency, LGA and Vendor, an execution run will be organised for the implementation. The goal of the certification process is to validate that the vendor system produces a feed that complies with the mandatory requirements of the Specification, and if the vendor system is capable of providing the optional/best practice data, that it is also correctly provided.

This test execution run will involve:

- 1. A test script will be run against the vendor system running at the LGA
- 2. The script will access the feed via a URL specified by the Complying Authority
- 3. The script will attempt to query data from the feed according to the use cases specified in 5.1
- 4. For each use case, the script will:
  - a. Validate that all *mandatory* fields defined in 4.3 are present
  - b. Validate that when any *optional* field is provided that it complies with the field formats defined in 4.3

Additionally, a manual inspection test will be performed that will:

- 1. Validate by manual review that the LGA's implementation system provides data to supported browsers and according to the data items defined in 4.3

## 7.3 Obtaining certification

The Agency is interested in ensuring that LGA implementations of application tracking systems comply with this Specification. Optionally, Vendors may wish to obtain Vendor certification of a test or staging installation of their system using sample data.



#### **7.4 Using "ATDIS-1.0.2 Data Interchange Specification Compliant" in Vendor software marketing**

Vendors may make use of the certification process in their marketing following successful completion of a compliance test execution.

The Agency will notify the LGA in writing upon meeting the requirements of the Specification.

## 8 Lifecycle

### 8.1 Version history

This is: **ATDIS-1.0.2 Data Interchange Specification [version: 1.0.2, revision: 145]**.

This section captures information about how The Agency plans to modify the specification over time. It is designed to give Vendors and LGAs information about possible changes in upcoming versions. As this is the first published version, there are no changes or deprecations for this version from the previous version.

### 8.2 Changed in this version

#### 8.2.1 Data items

- None identified

#### 8.2.2 Use cases

- None identified

#### 8.2.3 Distribution channels

- None identified

### 8.3 Deprecated in this version

#### 8.3.1 Data items

- None identified

#### 8.3.2 Use cases

- None identified

#### 8.3.3 Distribution channels

- None identified

### 8.4 Planned for deprecation in the next version

#### 8.4.1 Data items

- None identified

#### 8.4.2 Use cases

- Specification use cases (all)

#### 8.4.3 Distribution channels

- RSS

## 8.5 Planned for addition in the next release

### 8.5.1 Data items

- None identified

### 8.5.2 Use cases

- None identified

### 8.5.3 Distribution channels

- None identified

## 8.6 Notes and clarifications

The following notes present specific responses to anonymised feedback items captured during the review period:

Ref	Feedback	Response
General	<i>Use of "none" for empty date fields</i>	The specification has been updated throughout to use the JSON-standard of: <pre>{"element": null}</pre> to represent an empty value for an attribute. This change affects the following attributes: info/determination_date (mandatory) info/notification_start_date (optional) info/notification_end_date (optional) locations/section (mandatory)
General	<i>Security</i>	The Agency will provide guidance on security and copyright issues for documents and other material produced under the specification in communications external to the ATDIS specification.
General	<i>Schema / version changes</i>	The Agency expects one or two iterations of this specification based on the outcome of development of the Test Harness (7.2, page 33). The Agency expects little change to the schema once the initial version is bedded down. Any changes would be minimal, and based on feedback and Test Harness implementation. The Agency is likely to review the specification annually. Please see section 8 "Lifecycle" (page 35) for details on version-specific changes to the specification.
General	<i>Technology</i>	Compliance with different browsers, desktop and mobile operating systems has been intentionally omitted from the specification. For details on The Agency's view on browser standards, please see: <a href="http://webguide.gov.au/accessibility-usability/accessibility/">http://webguide.gov.au/accessibility-usability/accessibility/</a>
General	<i>Technology</i>	The design of an appropriate technical architecture necessary to support provision of an ATDIS feed (such as Web Servers and DMZ infrastructure) is the responsibility of the Council in conjunction with its software Vendor.

Ref	Feedback	Response
General	<i>Document management</i>	<p>The ATDIS specification concentrates on the data that moves <i>between</i> systems, not how the underlying data is persisted or surfaced for consumption.</p> <p>Decisions about content management system implementation remain within the purview of Councils and their software vendors.</p>
2.2.4 Page 11	<i>Use of wildcards for address search</i>	<p>After considering this feedback, the review team decided to retain use of wildcards for address searches. There are two reasons for this decision.</p> <p>Firstly, wildcards, and partial searching in general, is easier for end-users to work with because use of tools like Google Maps has accustomed them to typing parts of an address, and relying on the underlying to match the search terms accordingly.</p> <p>Secondly, there are a number of ways to break up an address into its constituent parts, and different systems will make system-dependent decisions about the correct way to do this. Any set of elements can be easily combined into a single text string, but it is non-trivial to take an arbitrary text string and break it up into its constituent parts.</p>
4.2 Page 17	<i>Use of .json extension for URIs</i>	<p>JSON is the modern approach for computer-to-computer interactions and is broadly accepted as the standard for data interchange from Internet-hosted APIs.</p> <p>Specifying the content type as a dot (".") extension is also the common approach for consuming systems to specify the format of a response to a requested URI.</p> <p>Please note that this version of the ATDIS specification allows for content providers to use XML <i>in addition to</i> (but not instead of) JSON for content if that is deemed necessary by a vendor.</p>
4.3 Page 18	<i>Specification of "documents" as mandatory</i>	<p>Documents relating to applications exist regardless of whether or not they are in digital form, and are a mandatory requirement as per current legislation.</p> <p>The Agency will work with stakeholders to clarify any issues relating to the copyright of documents published under the ATDIS specification.</p> <p>After review, the review team has decided that "documents" will remain a mandatory part of the specification.</p>
4.3.1 Page 19	<i>"related_apps" in "info" record</i>	<p>The <code>related_apps</code> field in the <code>info</code> record is designed to capture generic relationship <i>between</i> applications without attempting to specify the <i>type</i> of that relationship, or what it might mean.</p> <p>The basic intent of this field is to capture the relationship between applications with different application numbers.</p>

Ref	Feedback	Response
4.3.1 Page 19	<i>"last_modified_date" in "info" record</i>	The <code>last_modified_date</code> field in the <code>info</code> record should capture the date that the application was last changed in the underlying source system. Vendors are responsible for determining which internal date in their system is the relevant field to use to populate this value.  Conceptually, the <code>last_modified_date</code> field is the business concept that represents the date/time that the application was changed.
4.3.2 Page 20	<i>"more_info_url" in "reference" record</i>	The <code>more_info_url</code> field is designed to provide a simple URL to any further information available from the Council in relation to the application. The specification does not mandate the content available at this URL.
4.3.1 Page 19	<i>Application dates</i>	The <code>notification_start_date</code> and <code>notification_end_date</code> are optional fields. Vendors are free to populate this value if it is available in the source system.
4.3.1 Page 19	<i>"determination_type"</i>	The review team believes that it is important to have a normative set of determination types so that the data from different Councils is interchangeable.  The recommended set of values for <code>determination_type</code> is: <ul style="list-style-type: none"> <li>• Pending</li> <li>• Refused by Council</li> <li>• Refused under delegation</li> <li>• Withdrawn</li> <li>• Approved by Council</li> <li>• Approved under delegation</li> <li>• Rejected</li> </ul>
4.3.3 Page 21	<i>Lot / plan numbers</i>	The specification requires that lot / plan numbers be presented in a consistent and interchangeable format.  The review team has decided that the format for lot / plan numbers should be as currently described.
	<i>Subdivisions</i>	Vendors and Councils should use the original parcel street numbers and/or lot references. In the case of a subdivision, then the pre-subdivision data should be used.
4.3.4 Page 22	<i>"events" record</i>	In this context, the events record means any kind of activity that can occur against an application. "Tasks" and "workflows" would be synonyms for "events".  The specification does not dictate the type of event, but rather, that if an event of any type is captured, then this is the format in which it needs to be articulated.  The <code>last_modified_date_start</code> field is used for to specify a date range when searching.
4.3.4 Page 22	<i>"title_reference=none"</i>	See response to "Use of "none" for empty date fields" above

Ref	Feedback	Response
<p>5.1.6 Page 26</p>	<p><i>"page=2"</i></p>	<p>The vendor should provide pagination information as part of the response. As indicated in section 6.4 (page 30), the pagination block contains information about the number of items in the current page, the number of items per page and the total number of pages. It also includes an indication as the previous and next pages.</p> <pre data-bbox="687 568 1422 831"> "pagination": {   "previous": null,   "next": null,   "current": 1,   "per_page": 25,   "count": 100,   "pages": 1 } </pre> <p>This information is sufficient for any consuming system to work out how to a) present the information, and b) navigate from page to page.</p> <p>It is up to the Vendor system to determine page size, and if a response requires paging because of its size.</p> <p>As this is a standard pattern for JSON-based Internet-hosted APIs, the review team has decided not to make any changes.</p>

## 9 Appendix 1: Future possibilities

Although this section does not form part of the compliance process, it is worth presaging possible future ideas for uses of the data items and feeds imagined by the ATDIS-1.0.2 specification.

### 9.1 Subscription and publication of ATDIS events

The ability to *subscribe* for alerts on changes in state of development applications would make it very easy for Residents to be informed about applications taking place in their local areas.

### 9.2 Extended/enhanced attribute publication

Vendors of application tracking software have different capabilities that will make it easier to comply with the specification. This presents an opportunity to provide extensions to the basic capabilities by adding innovative functionality.

The Agency, in stewarding this process, hopes to influence vendors to add more and more capability over time into the core of the specification by pushing what it believes to be *best practice*. This will mean that what is optional in one release may be promoted to a mandatory part of the specification in a subsequent release.

## 10 Appendix 2: Examples of compliant ATDIS-1.0.2 requests and responses

### 10.1 Obtain a list of all applications

#### Request

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json
RES: 200 OK
```

#### Response

```
{
  "response": [
    {
      "application": {
        "info": {
          "dat_id": "DA2013-0381",
          "last_modified_date": "2013-04-20T02:01:07Z",
          "description": "New pool plus deck",
          "authority": "Example Council Shire Council",
          "lodgement_date": "2013-04-20T02:01:07Z",
          "determination_date": "2013-06-20T02:01:07Z",
          "notification_start_date": "2013-04-20T02:01:07Z",
          "notification_end_date": "2013-05-20T02:01:07Z",
          "status": "OPEN"
        },
        "reference": {
          "more_info_url": "http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications/DA2013-0381"
        },
        "locations": {
          [
            {
              "address": "123 Fourfivesix Street Neutral Bay NSW 2089",
              "land_title_ref": {
                "lot": "10",
                "section": "ABC",
                "dpsp_id": "DP2013-0381"
              }
            }
          ]
        }
      }
    },
    {
      "application": {
        ...
      }
    }
  ],
  "count": 2,
  "pagination": {
    "previous": null,
    "next": null,
    "current": 1,
    "per_page": 25,
    "count": 100,
  }
}
```

```
"pages": 1
}
}
```

## 10.2 Obtain a list of all applications for a set of postcodes

### Request

```
REQ: GET http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications.json?postcode=2089,2090
RES: 200 OK
```

### Response

```
{
  "response": [
    {
      "application": {
        "info": {
          "dat_id": "DA2013-0381",
          "last_modified_date": "2013-04-20T02:01:07Z",
          "description": "New pool plus deck",
          "authority": "Example Council Shire Council",
          "lodgement_date": "2013-04-20T02:01:07Z",
          "determination_date": "2013-06-20T02:01:07Z",
          "notification_start_date": "2013-04-20T02:01:07Z",
          "notification_end_date": "2013-05-20T02:01:07Z",
          "status": "OPEN"
        },
        "reference": {
          "more_info_url": "http://www.examplecouncil.nsw.gov.au/atdis/1.0/applications/DA2013-0381"
        },
        "locations": {
          [
            {
              "address": "123 Fourfivesix Street Neutral Bay NSW 2089",
              "land_title_ref": {
                "lot": "10",
                "section": "ABC",
                "dpsp_id": "DP2013-0381"
              }
            }
          ]
        }
      }
    },
    {
      "application": {
        ...
      }
    }
  ],
  "count": 2,
  "pagination": {
    "previous": null,
    "next": null,
    "current": 1,
    "per_page": 25,
    "count": 100,
    "pages": 1
  }
}
```



## 11 Appendix 3: Carpadium Pty Ltd

### 11.1 About Carpadium

[Carpadium](#) is a boutique Sydney-based consulting firm specialising in technology strategy, enterprise architecture and program management. Our Consulting Directors have significant experience delivering complex programs in financial services for online and mobile, as well as unique product development experience building enterprise-class software-as-a-service offerings in security, electronic presentment and payments.

- [Matthew Sinclair](#), *Director* – Matthew is a technology strategy specialist with almost 20 years experience in software engineering, consulting and product development. Matthew has delivered highly complex technology systems in the telecommunications, financial services and payments sectors. Matthew has been involved in the delivery of projects ranging from network analysers, to flight simulators and more recently high-performance mission-critical EFT payments transaction switching and Tier-1 online/mobile banking. [@matthewsinclair](#)
- [Andrew McDonald](#), *Director* – Andrew has more 20 years of senior program delivery experience in the banking and financial services sector. In the last 5 years, Andrew has focused on a number of large scale programs across Australia’s banking community, including AML, process redesign, online banking and identity / security management. Working at the heart of Australia's tier-1 retail online banking environment, Andrew understands and has experienced the issues in delivering business critical programs in a competitive and increasingly frugal project environment.

### 11.2 Carpadium's role in the specification process

The NSW Planning & Infrastructure engaged Carpadium to coordinate drafting of the ATDIS-1.0.2 specification.

As an independent organisation with no interest in the commercial software market for application tracking software, Carpadium’s role is to prepare the technical components of the specification, and to work with interested vendors to incorporate their requirements and feedback into the final specification.

We understand the commercial sensitivities given the competitive landscape and as such, we can act as an independent third party. Unless specifically instructed by the vendor or The Agency, all communications with Carpadium will be in the strictest confidence.