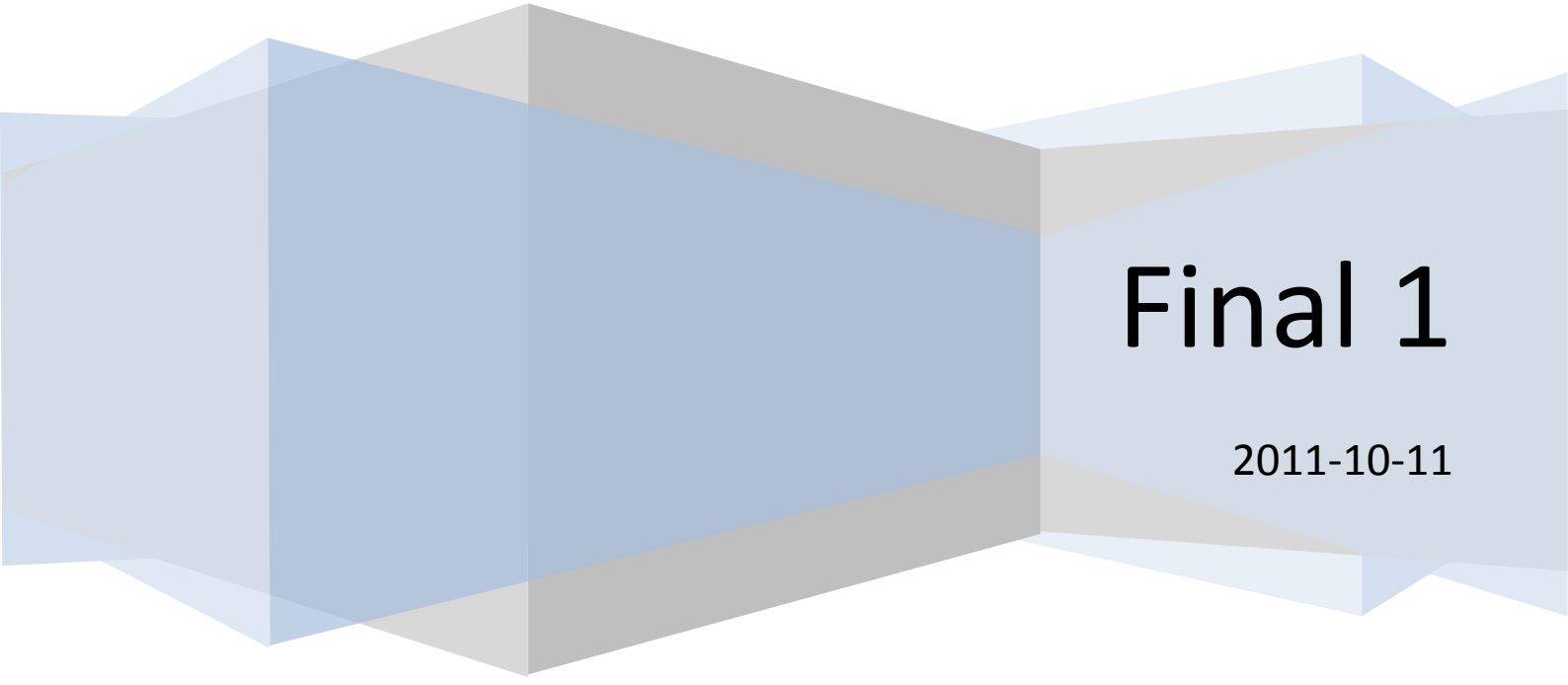
Sture ApS

Application Broker 3.0 Specification

Part 1 - External

Jakob Kruse



Final 1

2011-10-11

Contents

Contents 2

Introduction 5

Changes from version 2 6

SOAP and JSON interfaces 6

Web service based notifications 6

Case sets 6

Extended workflow for GT applications 6

ROV applications 6

More detailed site specifications 6

Geographic information 6

Draft in, preapproved out 7

Amendments 7

Compatibility 7

Less authority changes on approval 7

Minimum Feature Set 8

Applicant Client 8

Authority Client 8

Error Handling 9

The Meta System 9

SOAP Exceptions 9

Web Services 10

Login Service 10

Core Service 10

Query Service 10

Road Names Service 10

Notification Service 10

Utility Service 10
Login Service Operations 11
List of Operations 11
Operation Functionality 11
Core Service Operations 13
List of Operations 13
Operation Functionality 14
Query Service Operations 21
List of Operations 21
Operation Functionality 21
Road Names Service Operations 23
List of Operations 23
Operation Functionality 23
Notification Service Operations 24
List of Operations 24
Operation Functionality 24
Utility Service Operations 25
List of Operations 25
Operation Functionality 25
Notification Client 26
Email notifications 26
Web service notifications 26
Detail Elements 27
Case details 27
Site details 27
Cross section element details 27
Enumerations 29

Flowcharts 35

GT Application and Report 35

GT Report with Progress 36

GT Report with Completion 37

ROV Application and Report 38

GT Application and Report, Version 2 Applicant 39

GT Application and Report, Version 2 Authority 40

Introduction

Application Broker 3.0 consists of a number of web services[1] which will facilitate the submission of applications, their approval or rejection by Authorities and subsequent lifecycle as well as some supporting features.

It will provide a set of standards-based XML[2] interfaces (SOAP[3]; WSDL[4]) providing a series of *Operations*[5] which can be invoked to perform the various actions required.

It will also provide a parallel JSON[6] interface to those same operations, to facilitate greater ease of use by JavaScript[7]-based software running in web browsers.

For this reason, this specification will largely consist of a list of the web service operations which the service will provide and descriptions of the functionality – the inputs, outputs and effects – of each of these. Sections of this specification that are new or have been revised since the last revision are highlighted with a green box.

Applications may be of one of a number of types. Initially only two types – GT applications and ROV applications – will be supported, but the design of the Application Broker and its Interface should enable additional application types to be added, extending the utility of the service into new areas.

---

[1] Web Service. See: http://en.wikipedia.org/wiki/Web_service.

[2] XML: eXtensible Markup Language. See: http://www.w3schools.com/XML/xml_whatis.asp.

[3] SOAP – See: http://www.w3schools.com/soap/default.asp.

[4] WSDL: Web Service Description Language. See: http://www.w3schools.com/wsdl/wsdl_intro.asp.

[5] Operation: the unit of interaction used within a web service.

[6] JSON: JavaScript Object Notation. A lightweight data exchange format often used as an alternative to XML. See: http://www.json.org.

[7] JavaScript: a language implemented within virtually all modern web browsers. See: http://en.wikipedia.org/wiki/JavaScript. (Technically an implementation of the ECMAScript standard: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf.)

Changes from version 2

SOAP and JSON interfaces

WebGT 2.x has a partial SOAP interface (login, query and road names services only) and is otherwise built using POX ("plain old xml"). Version 3.0 will have full SOAP and JSON interfaces. The SOAP interfaces will have WSDL descriptions for easy client generation in most development environments.

Web service based notifications

WebGT 2.x is capable of notifying users about state changes by sending emails. Version 3.0 will extend this by offering both push- and pull-based web service notifications. For a push-based notification, a function in the Application Broker will post to (push) a predefined URL every time a relevant case changes state, giving details about the event. For a pull-based notification, an authority or applicant client can periodically retrieve (pull) details about state changes from the Notification Service.

Case sets

WebGT 2.x does not support the creation of case folders or case sets, although a workaround has been implemented using reference numbers. Version 3.0 will feature a proper case set entity, allowing applicants to convey information about connected applications to authorities.

Extended workflow for GT applications

The workflow supported by WebGT 2.x ends when an applicant submits a final completion message. Version 3.0 will support the approval or rejection (in case of insufficient re-establishment) of the completion by the authority, subsequent (re-) completions, warranty period specification and warranty inspections, all including comments and attachments. The extended workflow for GT applications will end when an authority submits an approved warranty inspection. Version 3 authority clients that do not wish to make use of these extensions can reply to a final completion with an approved warranty inspection, for functionality similar to version 2.

ROV applications

WebGT 2.x only supports applications for digging permissions (denoted here as "GT applications"). Version 3.0 will be prepared for any type of road related application, and will contain support for ROV ("*rådighed over vejareal*") applications, as well as the well-known GT applications. This generalized support has also led to the more general name "Application Broker".

More detailed site specifications

The site specification known from WebGT 2.x will be generalized and extended in version 3. Support for route numbers and slip road intervals will be added to the general site specification, common to all application types, in version 3. Dig methods, available at the application level in WebGT 2.x, will be moved to the site level in version 3.0, and will be available for GT applications only.

Geographic information

WebGT 2.x supports a single XY coordinate per site in each application, in the outdated System34 format. Version 3.0 will support one set of more generic geographic information – a point, line or polygon in GML format – per case and per site, allowing for integration with map services.

Draft in, preapproved out

In WebGT 2.x a newly created application is available to the authority immediately, and applications of the "report" variant (those for which the applicant has a general permission) are considered preapproved once retrieved by the authority. This has caused some problems in the past. Version 3.0 will introduce a new "draft" state for newly created applications (in which they are editable by the applicant and not available to the authority), and will remove the "preapproved" state for reports, requiring explicit approval just like "normal" applications. This allows for a more natural flow when authorities reject incorrect reports.

During the conversion process, reports in the preapproved state that are older than 8 days will be auto-approved, to provide as smooth a transition as possible.

Amendments

In WebGT 2.x once an application has been approved, it cannot be modified. Version 3 will be prepared to handle amendments to an application, starting with support for amending the end date (extending). Amendments will be possible after permission has been granted and before completion. Authorities will be able to reject (resulting in no changes to the case) or approve (resulting in the issuing of a new permission) amendment requests.

Compatibility

Although strictly not a change – version 2 was also backwards compatible with version 1 – version 3 will contain a few new features to support compatibility with version 2. It will be possible for an applicant client to query whether an authority supports ROV applications (which is optional with version 3). Version 3 will also introduce a change in application/case ids, and there will be a service to translate version 2 ids to version 3 ids.

Less authority changes on approval

WebGT 2.x allowed authorities to change most details of an application before approval. Version 3.0 is a bit stricter. The following attributes can no longer be changed by authorities: variant, date interval, character, subject, network, marking text.

Minimum Feature Set

Client systems interacting with the Application Broker 3.0 Core Service are required to implement a "minimum feature set", to ensure proper case processing. The minimum feature set for Applicant side client systems is different from the minimum feature set for Authority side client systems.

Support for the GT case type is mandatory. Clients may also optionally support the ROV case type. Some requirements below apply only to GT cases.

Applicant Client

Applicant side client systems must support:

- Creating a case, using "*createCase*".
- Sending a case, using "*sendCase*" or "*sendCaseSet*" from case state "*draft*".
- Completing a case, using "*completeCase*" from case states "*permitted*" (implies ability to receive case approval), "*restabrej*" and "*issuefound*" (implies ability to receive rejection of completion before and after warranty period).

Optional features include:

- Cancelling/deleting a case.
- Amending a case.
- Sending a progress report (temporary completion) for a case.

Authority Client

Authority side client systems must support:

- Receiving a case (with or without progress or completion details), cancellation, amendment, progress report and completion, using "*receiveCase*" from case states "*sent*", "*cancelsent*", "*amendsent*", "*progsent*" and "*compsent*".
- Approving a case, using "*approveCase*" from case state "*processing*".
- Rejecting a case, using "*rejectCase*" from case state "*processing*".
- Approving an amendment, using "*approveAmendment*" from case state "*amendrec*".
- Rejecting an amendment, using "*rejectAmendment*" from case state "*amendrec*".
- Approving a completion, using "*approveCompletion*" from case state "*completed*".

Optional features include:

- Starting a warranty period.
- Rejecting a completion.

Error Handling

The Meta System

Most of the Application Broker web service operations[8] use a common "Meta" format for returning error information. The Meta structure consists of a Boolean "Success", and a list of error codes and texts. If Success is "false", the operation has not been executed, and any part of the response outside of the Meta structure should be disregarded.

The (numeric) error codes are in the range 1 to 4 and represent the following classes of errors:

1. Server error. Caused by errors in the application logic, network or database. These should always be reported. The error text will contain technical details for Application Broker support.
2. Client error. Caused by invalid actions or data by a client system. Includes all validation errors. The error text (which is in Danish) is formatted for user display.
3. Token error. Caused by the use of an invalid Login Token. Usually rectified by obtaining a new token from the Login Service.
4. JSON error. Caused by malformed JSON sent to one of the JSON-in services.

SOAP Exceptions

Some errors, such as those caused by malformed input, are not handled by the Meta system, and will result in SOAP Exceptions. The operations of the Login Service also consistently use SOAP Exceptions to report errors.

[8] The notable exclusion being the operations of the Login Service, due to be revised in a later version.

Web Services

Login Service

The first service that a client of the Application Broker will meet is the Login Service. It handles authentication and rights management for the other services. Clients will need to obtain a token from this service to access most operations of the other services.

No changes are made to the existing Login Service, but it is migrated into the Application Broker suite.

Core Service

The Core Service handles the actual application workflow. It contains operations for creating, retrieving and updating applications, permissions, completions etc.

The Core Service is completely redesigned for version 3.0.

Query Service

Searching the broker database, retrieving lists of applications matching given criteria, is done using the Query Service. It supports the creation and execution of queries as well as the retrieval of query results.

The Query Service is augmented to support the additions and the new database format.

Road Names Service

The site specification is a central part of all applications handled by the Application Broker. To heighten the quality of site specifications the Road Names Service allows authority side clients of the broker to supply lists of valid road names. Applicant side clients of the broker can use these lists to aid users in selecting valid road names.

The Road Names Service is augmented to support more than one name per administrative street code, as well as some new attributes.

Notification Service

The Notification Service allows client systems to retrieve (pull) lists of state-changes in the broker. Alternatively clients can create a service and sign up for service-based (push) notifications from a Notification Client in the Application Broker.

The Notification Service and Notification Client are new concepts in version 3.0.

Utility Service

The Utility Service will contain various support methods for the Core Service, for instance to aid in database migration and retrieve metadata.

The Utility Service is a new concept in version 3.0.

Login Service Operations

These details are unchanged from the existing Login Service

For operation details see http://ab.vd.dk/broker/3.0/login.wso.

List of Operations

- Login
- TokenInformation
- Units
- ValidateToken
- Rights
- RightsArray
- HasRight
- VirkLogin
- VirkLinkLogin
- VirkCreateLogin
- UnitsForCVR
- UnitsForVirkToken

Operation Functionality

Login

Logs in to the service. Returns a LoginToken or a SOAP fault.

All fields are mandatory.

TokenInformation

Returns information about a LoginToken, or a SOAP fault if the LoginToken is invalid.

All fields are mandatory.

Units

Returns a list of Units for which LoginToken has right, or a SOAP fault.

All fields are mandatory.

ValidateToken

Checks whether a LoginToken is valid. Returns true or false.

All fields are mandatory.

Rights

Returns all rights for the LoginToken/UnitID pair, concatenated to a single string, or a SOAP fault.

All fields are mandatory.

RightsArray

Returns a string array of all rights for the LoginToken/UnitID pair, or a SOAP fault.

All fields are mandatory.

HasRight

Checks whether LoginToken grants Right in UnitID. Returns true or false, or a SOAP exception if LoginToken or Right is invalid.

All fields are mandatory.

VirkLogin

Logs in to the service using an SSO token. Returns a LoginToken or a SOAP fault.

All fields are mandatory.

VirkLinkLogin

Logs in to the service using both an SSO token and a native login. Links SSO and native user. Returns a LoginToken or a SOAP fault.

All fields are mandatory.

VirkCreateLogin

Creates a new user, as specified by Token, under the unit specified by UnitID, logs in the new user and returns a LoginToken, if allowed. Otherwise returns a SOAP fault.

All fields are mandatory.

UnitsForCVR

Returns a list of administrative units matching the specified CVR number.

All fields are mandatory.

UnitsForVirkToken

Returns a list of administrative units matching the CVR number of the virk.dk user specified by the token.

All fields are mandatory.

Core Service Operations

In general, operations on the Core Service will be invoked by either *Applicants* (organisations making Applications) or *Authorities* (organisations responsible for approving Applications) – these are referred to as *Actors*. Most operations will typically be invoked only by one or the other, although some may be applicable to both.

For operation details see http://ab.vd.dk/broker/3.0/core.wso.

List of Operations

General

- isAlive

Creation

- createCase

Addition

- addSite
- addContact
- addAttachment

Modification

- updateCaseSetDetail
- updateCaseDetail
- updateSiteDetail
- updateContactDetail
- updateProgressDetail
- updateCompletionDetail

Removal

- removeSite
- removeContact
- removeAttachment

Transmission

- sendCaseSet
- sendCase
- receiveCase

Cancellation

- cancelCaseSet
- cancelCase

Deletion

- deleteCase

Processing

- approveCase
- rejectCase

Amendment

- amendEndDate
- approveAmendment
- rejectAmendment

Completion

- progressCase
- completeCase
- approveCompletion
- rejectCompletion
- setWarranty

Retrieval

- getCaseSets
- getCases
- getSites
- getContacts
- getAttachments

Operation Functionality

LoginToken and any Identifier fields (CaseSetIdentifier, CaseIdentifier etc.) are mandatory in all requests that include them, with the exception of CaseSetIdentifier in the createCase request.

isAlive

Checks that the service is running.

createCase

Creates a Case of the specified type and variant.

If a non-zero CaseSetIdentifier is given, the Case is added to that Case Set. Otherwise a new Case Set is created. The Case will be created with an initial state of "*draft*".

When creating a GT report (TypeCode is "*gt*", VariantCode is "*report*"), a progress report or completion can be created simultaneously. To create a progress report, include the Case Detail named "*progress*". To create a completion, include the Case Detail named "*polluted*". Case Details are described on page 27.

Mandatory fields include TypeCode, VariantCode, AuthorityCode, StartDateTime and EndDateTime. For mandatory fields on optional Sites and Contacts, see addSite and addContact.

Returns identifiers for created entities.

This operation is only valid for applicants.

addSite
Adds a Site to an existing Case.

This operation is only valid for the applicant in Case state "*draft*" and the authority in Case state "*processing*".

addContact
Adds a Contact to an existing Case.

Mandatory fields include Addressee Type and at least one Role. Mandatory fields for any telephone numbers include TypeCode and TelephoneSubscriberIdentifier.

This operation is only valid for the applicant and only in Case state "*draft*".

addAttachment
Adds an Attachment to an existing Case.

If URI is set, then Content is ignored and file content is retrieved from URI using HTTP.

updateCaseSetDetail
Updates the attributes of an existing Case Set.

This operation is only valid for the applicant and only on Case Sets where all Case states are "*draft*".

updateCaseDetail
Updates the attributes of an existing Case.

Mandatory fields include AuthorityCode, StartDateTime and EndDateTime.

This operation is only valid for the applicant and only in Case state "*draft*".

updateSiteDetail
Updates the attributes of an existing Site.

This operation is only valid for the applicant in Case state "*draft*" and the authority in Case state "*processing*".

updateContactDetail
Updates the attributes of an existing Contact.

Mandatory fields include Addressee Type and at least one Role. Mandatory fields for any telephone numbers include TypeCode and TelephoneSubscriberIdentifier.

This operation is only valid for the applicant and only in Case state "*draft*".

updateProgressDetail
Updates the attributes of the latest progress report on an existing Case.

This operation is only valid for the applicant and only in Case state "*draft*".

updateCompletionDetail
Updates the attributes of the latest completion report on an existing Case.

This operation is only valid for the applicant and only in Case state "*draft*".

removeSite
Removes a Site from a Case.

This operation is only valid for the applicant and only in Case state "*draft*".

removeContact
Removes a Contact from a Case.

This operation is only valid for the applicant and only in Case state "*draft*".

removeAttachment
Removes an Attachment from a Case.

This operation is only valid for the applicant.

sendCaseSet
Sends all valid and unsent Cases on a previously created Case Set to the authorities.

This will change the state of all affected Cases to "*sent*".

This operation is only valid for the applicant and will only affect Cases in Case state "*draft*".

sendCase
Sends a previously created Case to the authority.

If the Case does not have at least one Site, an error will be reported. Otherwise this will change the state of the Case to "*sent*".

This operation is only valid for the applicant and only in Case state "*draft*".

receiveCase
Acknowledges the receipt of one of a number of submissions, all relating to Cases: the Sending of a Case; the Cancellation of a Case; the Amendment of a Case; the Progress of a Case or the Completion of a Case.

The Action field value must be:

- "*application*" to acknowledge receipt of an application or a report without progress or completion details
- "*application+progress*" to acknowledge receipt of a report with progress details
- "*application+completion*" to acknowledge receipt of a report with completion details
- "*cancellation*" to acknowledge receipt of a cancellation
- "*amendment*" to acknowledge receipt of an amendment
- "*progress*" to acknowledge receipt of a progress report
- "*completion*" to acknowledge receipt of a completion report

Depending on the type of submission, the state of the Case will be changed in one of the following ways:

- From “*sent*” to “*processing*”
- From “*cancelsent*” to “*cancelled*”
- From “*amendsent*” to “*amendrec*”
- From “*progsent*” to “*progrec*”
- From “*compsent*” to “*completed*” or “*warranty*” (depending on existence of warranty)

Mandatory fields include Action.

This operation is only valid for the authority and only in the Case states “*sent*”, “*cancelsent*”, “*amendsent*”, “*progsent*” or “*compsent*”.

cancelCaseSet

Cancels all Cases in a Case Set. Notifies the authorities if necessary.

Depending on the current states of the Cases, the states will be changed in one of the following ways:

- From “*draft*” to “*cancelled*”
- From “*sent*” to “*cancelled*”
- From “*processing*” to “*cancelsent*”
- From “*permitted*” to “*cancelsent*”

This operation is only valid for the applicant and only if all Cases are in one of the Case states “*draft*”, “*sent*”, “*processing*” or “*permitted*”.

cancelCase

Cancels a Case. Notifies the authority if necessary.

Depending on the current state of the Case, the state will be changed in one of the following ways:

- From “*draft*” to “*cancelled*”
- From “*sent*” to “*cancelled*”
- From “*processing*” to “*cancelsent*”
- From “*permitted*” to “*cancelsent*”

This operation is only valid for the applicant and only in the Case states “*draft*”, “*sent*”, “*processing*” or “*permitted*”.

deleteCase

Deletes a Case.

The state of the case will change to “*deleted*”.

This operation is only valid for the applicant and only in the Case states “*draft*”, “*sent*”, “*rejected*”, “*cancelled*” and “*archived*”. This operation is also valid in the Case states “*cancelsent*”, “*amendsent*”, “*amendrec*”, “*compsent*”, “*completed*” and “*warranty*” after a period of inactivity.

approveCase

Approves a Case and issues a permission document to the applicant.

Depending on the type of the Case, the state of the Case will be changed in one of the following ways:

- From "*processing*" to "*permitted*", "*progrec*" or "*completed*"

This operation is only valid for the authority and only in the Case state "*processing*".

rejectCase

Rejects a Case.

The state of the Case will change to "*rejected*".

This operation is only valid for the authority and only in the Case state "*processing*".

amendEndDate

Submits an application to amend the end date of an approved Case.

The state of the Case will change to "*amendsent*".

Mandatory fields include EndDateTime.

This operation is only valid for the applicant and only in the Case state "*permitted*".

approveAmendment

Approves an amendment application, updating the Case with the amendment data, and issues a revised permission document to the applicant.

The state of the Case will change to "*permitted*".

This operation is only valid for the authority and only in the Case state "*amendsent*".

rejectAmendment

Rejects an amendment application, discarding the amendment data.

The state of the Case will change to "*permitted*".

This operation is only valid for the authority and only in the Case state "*amendsent*".

progressCase

Sends a progress report ("temporary completion") for a Case.

The state of the Case will change to "*progsent*", unless it is "*draft*" in which case it won't change.

Mandatory fields include ProgressText.

This operation is only valid for the applicant and only in the Application states "*draft*" (report variant only), "*permitted*", "*progsent*" or "*progrec*".

completeCase

Sends a completion report for a Case.

The state of the Case will change to "*archived*" or "*compsent*" depending on whether the Case supports completion reports.

Mandatory fields include a Detail with Name "*polluted*".

This operation is only valid for the applicant and only in the Case states "*permitted*", "*progsent*", "*progrec*", "*restabrej*" or "*issuefound*".

approveCompletion

Approves completion of a Case, in effect closing the Case.

The state of the Case will change to "*archived*".

This operation is only valid for the authority and only in the Case states "*completed*" or "*warranty*".

rejectCompletion

Rejects completion of a Case.

Under normal circumstances this could occur in one of two situations:

1. If reestablishment is inadequate the recently received completion will be rejected, leading to a state of "*restabrej*".
2. If issues are found during, or just before the end of, the warranty period the completion will be rejected, leading to a state of "*issuefound*".

This operation is only valid for the authority and only in the Application states "*completed*" or "*warranty*".

setWarranty

Starts a warranty period for a Case. Implicitly this also acknowledges that reestablishment was approved. To approve reestablishment without starting a warranty period, use approveCompletion.

The state of the Case will change to "*warranty*".

Mandatory fields include WarrantyExpirationDate.

This operation is only valid for the authority and only in the Case state "*completed*".

getCaseSets

Returns all of the details for a number of Case Sets.

This operation is only valid for the applicant.

getCases

Returns all of the details for a number of Cases.

This operation is valid for both the applicant and the authority.

getSites

Returns all of the details for a number of Sites.

This operation is valid for both the applicant and the authority.

getContacts

Returns all of the details for a number of Contacts.

This operation is valid for both the applicant and the authority.

getAttachments

Returns all of the details for a number of Attachments.

This operation is valid for both the applicant and the authority.

Query Service Operations

The operations of the Query Service create and execute searches against the Case database synchronously. Consequently “startSearch”, “doSearch” and “doDiff” do not return until all results have been found.

The Query Service interface is designed with asynchronous searches in mind though. It is expected that a future version of the Query Service will execute searches asynchronously, which is why methods that return information about results also tell whether the search has completed or not.

For operation details see http://ab.vd.dk/broker/3.0/query.wso.

List of Operations

- createSearch
- dateOfFirstResult
- startSearch
- doSearch
- doDiff
- fetchResultIDs
- fetchResultSummaries
- resultCount

Operation Functionality

LoginToken is mandatory in all requests.

createSearch

Creates a query with the specified criteria and returns its ID.

Note that this does not start execution of the query.

One “*applicant*” or “*authority*” criterion is mandatory.

dateOfFirstResult

Returns the creation date and time of the first result matching the specified query.

Note that if the query contains creation date constraints, these will be ignored!

All fields are mandatory.

startSearch

Runs the specified query.

All fields are mandatory.

doSearch

Creates a query with the specified criteria, runs it and returns its ID.

One “applicant” or “authority” criterion is mandatory.

doDiff

Creates a diff query based on the specified query, runs it and returns its ID.

All fields are mandatory.

fetchResultIDs

Returns a number of results (Cases) from the specified query.

It is possible to specify the maximum number of results to fetch and the sort direction to use. The first unfetched results in the specified direction will be returned.

Results are returned in the form of Case and Case Set identifiers only.

All fields are mandatory.

fetchResultSummaries

Returns a number of results (Cases) from the specified query.

It is possible to specify the maximum number of results to fetch and the sort direction to use. The first unfetched results in the specified direction will be returned.

Results are returned in the form of Case and Case Set identifiers, a summary of the Case details and a list of valid actions.

All fields are mandatory.

resultCount

Returns the total number of results for the specified query.

All fields are mandatory.

Road Names Service Operations

These details are unchanged from the existing Road Names Service, and are only listed here for documentation purposes.

New in this version of the Road Names Service is support for a “Section” attribute, which introduces support for more than one road name per administrative street code, as well as “Status” and “Zone” attributes, providing details about the classification of the road to the applicant.

For operation details see http://ab.vd.dk/broker/3.0/roadnames.wso.

List of Operations

- StoreStreets
- QueryStreets
- QueryStreetsInitial

Operation Functionality

LoginToken is mandatory in all requests.

StoreStreets

Stores all streets in Streets under the authority specified by AuthorityCode.

Mandatory fields include AuthorityCode.

QueryStreets

Returns a list of all streets for the authority specified by AuthorityCode.

All fields are mandatory.

QueryStreetsInitial

Returns a list of all streets for the authority specified by AuthorityCode, whose names begin with the string specified by Initial.

All fields are mandatory.

Notification Service Operations

The operations of the Notification Service allow a client to retrieve and clear lists of changed cases.

For applicants, notifications are generated on[9] approveCase (application/approve), rejectCase (application/reject), approveAmendment (amendment/approve), rejectAmendment (amendment/reject), approveCompletion (completion/approve), rejectCompletion (completion/reject) and setWarranty (warranty/create).

For authorities, notifications are generated on sendCase (application/send), deleteCase from "*sent*" state (application/delete), cancelCase (cancellation/send), amendEndDate (amendment/send), progressCase (progress/send) and completeCase (completion/send).

Notifications include information on the causing transaction, the resulting state and optionally full case data.

Notifications are subscription-only. Contact Application Broker support to subscribe.

For operation details see http://ab.vd.dk/broker/3.0/notification.wso.

List of Operations

- getNotifications
- acknowledgeNotifications

Operation Functionality

getNotifications

Returns a list of notification events for the specified administrative unit.

All fields are mandatory.

acknowledgeNotifications

Acknowledges (deletes) a list of notification events.

All fields are mandatory.

[9] Parentheses show "Action" and "Transaction" pairs used in notifications.

Utility Service Operations

For operation details see http://ab.vd.dk/broker/3.0/utility.wso.

List of Operations

- translateDigIdentifier
- supportsCaseType
- getValueLists
- getAuthorityList

Operation Functionality

translateDigIdentifier

Looks up a Case based on a version 2.x DigIdentifier (applicant id and serial number) and returns the complete set of version 3.0 identifiers for the Case.

All fields are mandatory.

supportsCaseType

Returns whether the specified authority supports the specified case type.

All fields are mandatory.

getValueLists

Returns enumerations or “value lists” used in the other services. Input is a list of value list codes, as defined by the “Value List Names” enumeration.

All fields are mandatory.

getAuthorityList

Returns the list of supported authorities, their type and which case types they accept.

Notification Client

The Application Broker Notification Client sends out notifications about case changes to subscribed parties. Applicants and authorities can subscribe to email notifications and/or web service notifications.

Email notifications

To subscribe to email notifications, you register an email address with the Application Broker support team. One email address can be registered per applicant or authority. Multiple applicants or authorities can share an email address.

Web service notifications

To subscribe to web service notifications, you register an HTTP URI (an URL) with the Application Broker support team. One URI can be registered per applicant or authority. Multiple applicants or authorities can share a URI.

When the Notification Client sends a web service notification, it does so by sending an HTTP POST request to the registered URI. The POST body will contain three values:

- "Recipient" which contains the id of the applicant or authority that is being notified.
- "CaseID" which contains the id of the case in question.
- "State" which contains the code for the new state of the case.

The parameters are sent using content-type "application/x-www-form-urlencoded". For example:

Recipient=9412&CaseID=123&State=sent

The Notification Client expects the web service to send back a reply containing the string "APPLICATION BROKER NOTIFICATION RECEIVED" if the notification was received correctly. Any reply not containing this string will be interpreted as an error, and will cause the Notification Client to retry the notification later.

The Notification Client does support basic HTTP authentication. Supply a username and password to use this feature.

Detail Elements

Some operations accept or return a list of Name/Value element pairs called “Details”, used to carry type-specific information. Details exist on three levels: case, site and cross section element. Valid detail names and their value spaces are described below.

Case details

Name	Type	Value Type	Description
character	gt	Enum: Dig Character	Character of dig (DA: arbejdets art)
subject	gt	Enum: Dig Subject	Subject of dig (DA: ledningstype)
network	gt	Enum: Dig Network	Affected network part (DA: ledningsnetdel)
progress	gt (progress)	String(255)	Progress text
polluted	gt (completion)	Boolean	Polluted soil
aa/<cse>	gt (completion)	Decimal(4,2)	Affected area, one property for each Cross Section Element
character	rov	String(255)	Character of work (DA: arbejdets art)
originator	rov	String(255)	Name of Road Directorate employee that originated the task (DA: bestiller)
media relevance	rov	Boolean	Is the case relevant for the media
end	gt, rov (amendment)	Datetime	Amended end date
expiration	gt (warranty)	Date	Date of warranty expiration

Site details

Name	Type	Value Type	Description
method	gt	Enum: Dig Method	Method of digging (DA: udførelsesmetode)
marking plan	gt, rov	String(255)	Which marking plan to use
depth	gt	Decimal(4,2)	Depth in meters
width	gt	Decimal(4,2)	Width in meters
length	gt	Decimal(4,2)	Length in meters
num open lanes	rov	Integer(1)	Number of open lanes
total lanes	rov	Integer(1)	Total number of lanes
speed limit working	rov	Integer(3)	Speed limit inside working hours
speed limit nonworking	rov	Integer(3)	Speed limit outside working hours
minimum width	rov	Decimal(2,2)	Minimum lane width in meters
minimum height	rov	Decimal(2,2)	Minimum free height in meters
police force	rov	Enum: Police Force	
working hours	rov	String(50)	Textual description of working hours
1888 signage	rov	Boolean	Will an 1888 sign be shown?
direction	rov	Enum: Direction	
working length	rov	Integer(4)	Length of working area in meters
marking length	rov	Integer(4)	Length of marking area in meters

Cross section element details

Name	Type	Value Type	Description

orientation	gt	Enum: Orientation

Enumerations

Action Name (12)

Value	Meaning (in Danish)
application	Ansøgning/anmeldelse
progress	Midlertidig færdigmelding
completion	Færdigmelding
warranty	Garanti
amendment	Ændringsansøgning
cancellation	Annullering

Attachment Type (12)

Value	Meaning (in Danish)
application	Ansøgning
permission	Tilladelse
terms	Bilag til tilladelsen
userfile	Anden fil

Case State (10)

Value	Meaning (in Danish)
draft	Ansøgning/anmeldelse oprettet
sent	Ansøgning/anmeldelse sendt til myndighed
processing	Ansøgning/anmeldelse behandles af myndighed
permitted	Ansøgning/anmeldelse godkendt
rejected	Ansøgning/anmeldelse afvist
progsent	Midlertidig færdigmelding sendt til myndighed
progrec	Midlertidig færdigmelding modtaget af myndighed
compsent	Færdigmelding sendt til myndighed
completed	Færdigmelding modtaget af myndighed
cancelsent	Annullering sendt til myndighed
cancelled	Annullering modtaget af myndighed
restabrej	Retablering afvist
issuefound	Mangler fundet ved eftersyn
warranty	Under garanti
amendsent	Ansøgning om ændring sendt til myndighed
amendrec	Ansøgning om ændring behandles af myndighed
archived	Arkiveret
deleted	Slettet

Case Type (4)

Value	Meaning (in Danish)
gt	Gravesag
rov	Rådighedssag

Case Variant (12)

Value	Meaning (in Danish)
application	Ansøgning
report	Anmeldelse

Contact Type (8)

Value	Meaning (in Danish)

person	Privatperson
company	Virksomhed

Country Code (4)

Value	Meaning (in Danish)
208	Danmark
276	Tyskland
5100	Danmark
5120	Sverige
5180	Tyskland
752	Sverige
DE	Tyskland
DEU	Tyskland
DK	Danmark
DNK	Danmark
SE	Sverige
SWE	Sverige

Cross Section Element (20)

Value	Meaning (in Danish)
boundary	Vejalen
buildline	Byggelinjeareal
central reservation	Midterrabat
cycle path	Cykelsti
ditch	Grøft
dividing verge	Indre skillerabat
footpath	Fortov
outer dividing verge	Ydre skillerabat
parking area	P-areal
roadway	Kørebane
verge	Yderrabat

Direction (8)

Value	Meaning (in Danish)
along	Med kilometreringsretningen
against	Mod kilometreringsretningen
both	Begge retninger
na	Uden betydning

GT Character (14)

Value	Meaning (in Danish)
establishment	Etablering
repair	Reparation
splice	Splidsehul
cablefault	Kabelfejl
renewal	Fornyelse
disconnect	Afbrydelse
alarmfault	Alarmfejl
removal	Fjernelse
cleanup	Oprensning
sampling	Prøvehul

replacement	Udskiftning
investigation	Undersøgelse
renovation	Renovering
see-comment	Se bemærkning

GT Method (32)

Value	Meaning (in Danish)
digging	Gravning
chain-digging	Kædegravning
displacement	Jordfortrængning
cracking	Cracking
relining	Relining
drilling	Boring
displacement-digging	Jordfortrængning/gravning
horisontal-directional-drilling	Styret underboring
see-comment	Se bemærkning

GT Network (12)

Value	Meaning (in Danish)
main	Hovedledning
service	Stikledning
supply	Forsyningsledning
see-comment	Se bemærkning

GT Subject (12)

Value	Meaning (in Danish)
water	Vand
sewer	Kloak
power	El
gas	Gas
comm	Tele
antenna	Antenne
heating	Fjernvarme
fiber	Fiber
see-comment	Se bemærkning

Orientation (6)

Value	Meaning (in Danish)
along	Langsgående
across	Tværgående

Phone Type (14)

Value	Meaning (in Danish)
phone-day	Telefon dag
phone-evening	Telefon aften
phone-24hr	Døgn-telefon
mobile	Mobil
fax	Telefax

Planning Zone (6)

Value	Meaning (in Danish)
urban	Byzone
rural	Landzone

Police Force (18)

Value	Meaning (in Danish)
koebenhavn	København
vestegn	Københavns Vestegn
nordsjaelland	Nordsjælland
midtvestsjaelland	Midt- og Vestsjælland
sydsjaelland	Sydsjælland og Lolland-Falster
fyn	Fyn
soenderjylland	Syd- og Sønderjylland
sydoestjylland	Sydøstjylland
oestjylland	Østjylland
vestjylland	Midt- og Vestjylland
nordjylland	Nordjylland
bornholm	Bornholm

Result Status (1)

Value	Meaning (in Danish)
+	Tilføjet
-	Fjernet
*****	Opdateret

Road Status (22)

Value	Meaning (in Danish)
private path	Privat sti
private communal path	Privat fællessti
municipality path	Kommunal sti
state path	Statslig sti
private road	Privat vej (1 ejer)
private communal road	Privat fællesvej
ordinary road	Almen vej
municipality road	Kommunevej
state road	Statsvej
public area	Offentligt areal

Role (10)

Value	Meaning (in Danish)
applicant	Ansøger / anmelder
authority	Vejmyndighed
contractor	Entreprenør
marker	Afmærkningsansvarlig
proprietor	Ledningsejer

Side (6)

Value	Meaning (in Danish)

left	Venstre
middle	Midt
right	Højre

Source (10)

Value	Meaning (in Danish)
applicant	Ansøger / anmelder
authority	Vejmyndighed

Transaction (8)

Value	Meaning (in Danish)
create	Opret
send	Send
receive	Kvittér
approve	Godkend
reject	Afvis
archive	Arkivér (kun version 2)
delete	Slet

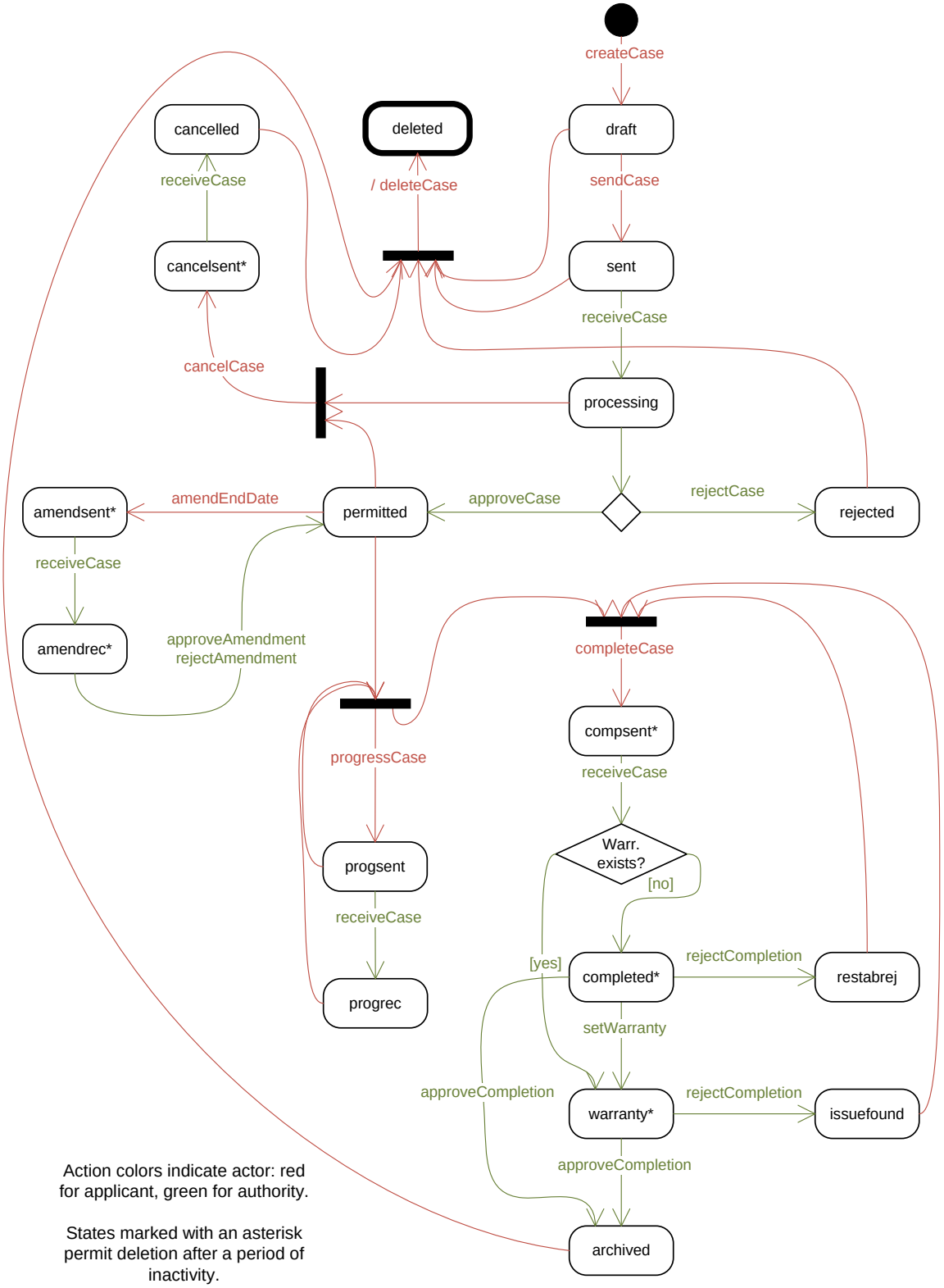
Value List Name (22)

Value	Meaning
action-name	Action Name
attachment-type	Attachment Type
case-detail-name	Case Detail Name
case-state	Case State
case-type	Case Type
case-variant	Case Variant
change-request-type	Change Request Type
contact-type	Contact Type
country-code	Country Code
criteria-name	Criteria Name
cross-section-element	Cross Section Element
cse-detail-name	CSE Detail name
direction	Direction
gt-character	GT Character
gt-method	GT Method
gt-network	GT Network
gt-subject	GT Subject
notify-type	Notify Type
orientation	Orientation
phone-type	Phone Type
planning-zone	Planning Zone
police-force	Police Force
query-state	Query State
result-status	Result Status
right	Right
road-status	Road Status
role	Role
side	Side
site-detail-name	Site Detail Name
source	Source

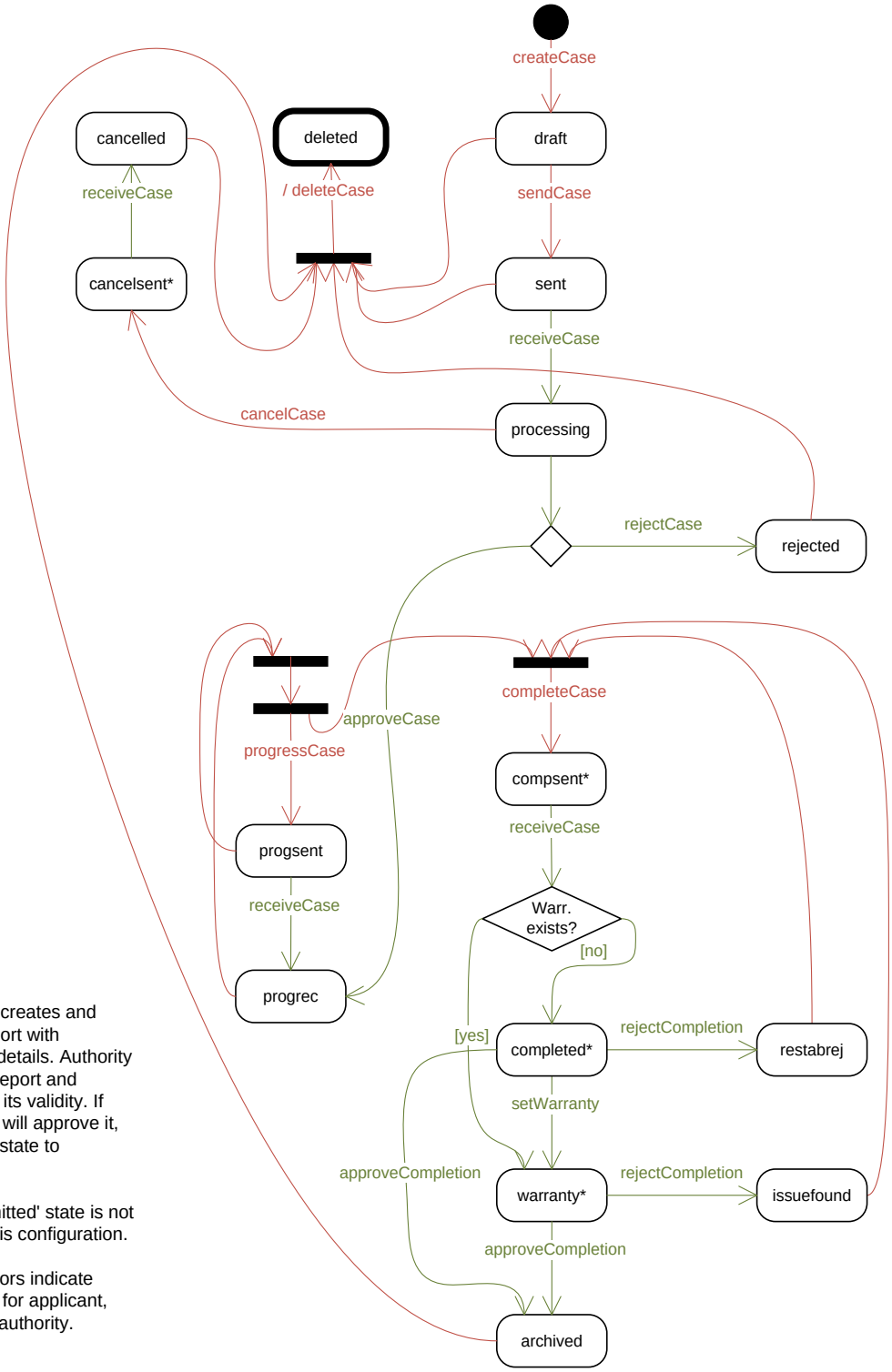
status-code	Status Code
transaction	Transaction
unit-type	Unit Type
value-list-name	Value List Name

Flowcharts

GT Application and Report



GT Report with Progress

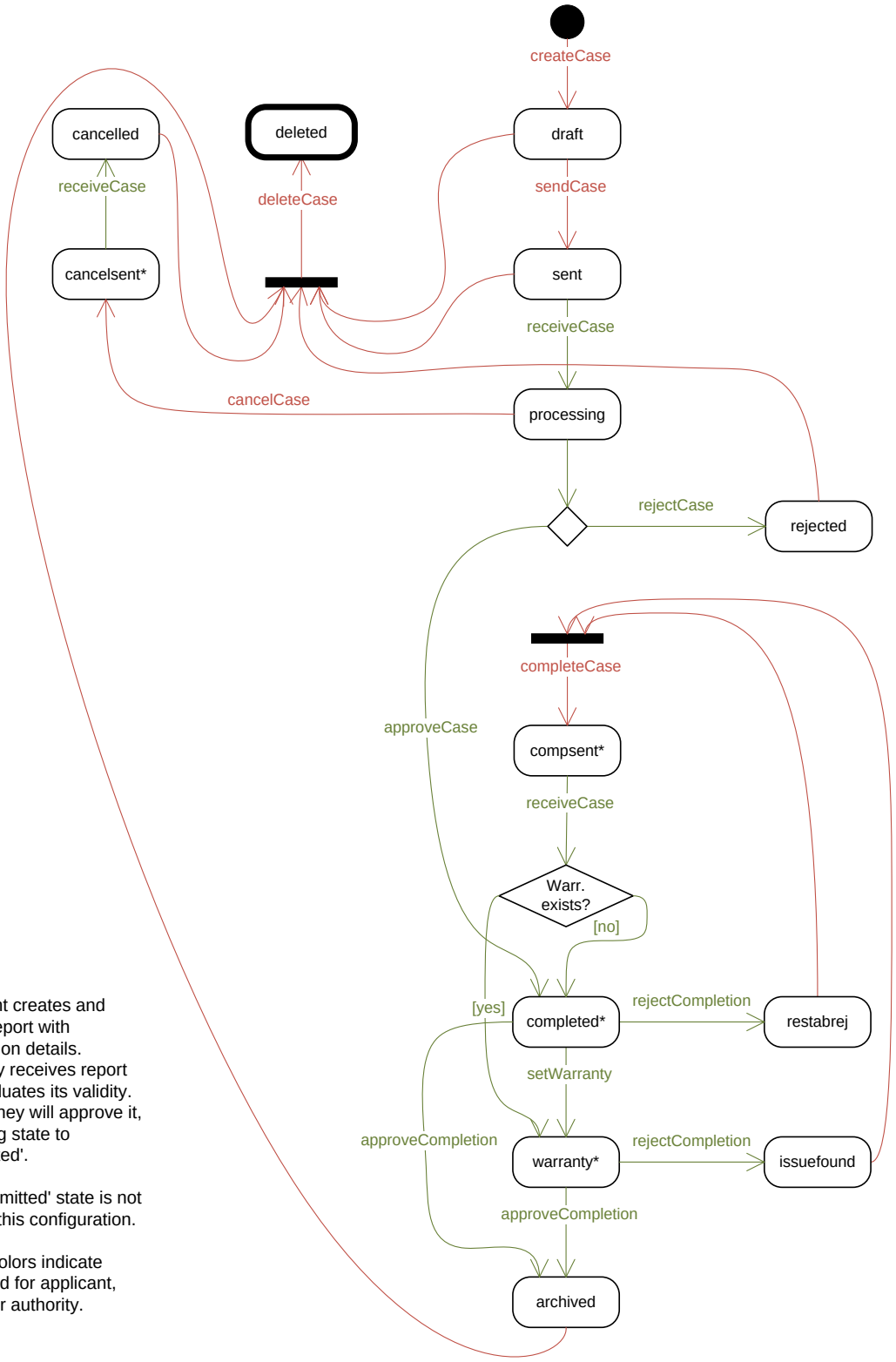


Applicant creates and sends report with progress details. Authority receives report and evaluates its validity. If valid they will approve it, changing state to 'progrec'.

The 'permitted' state is not used in this configuration.

Action colors indicate actor: red for applicant, green for authority.

GT Report with Completion

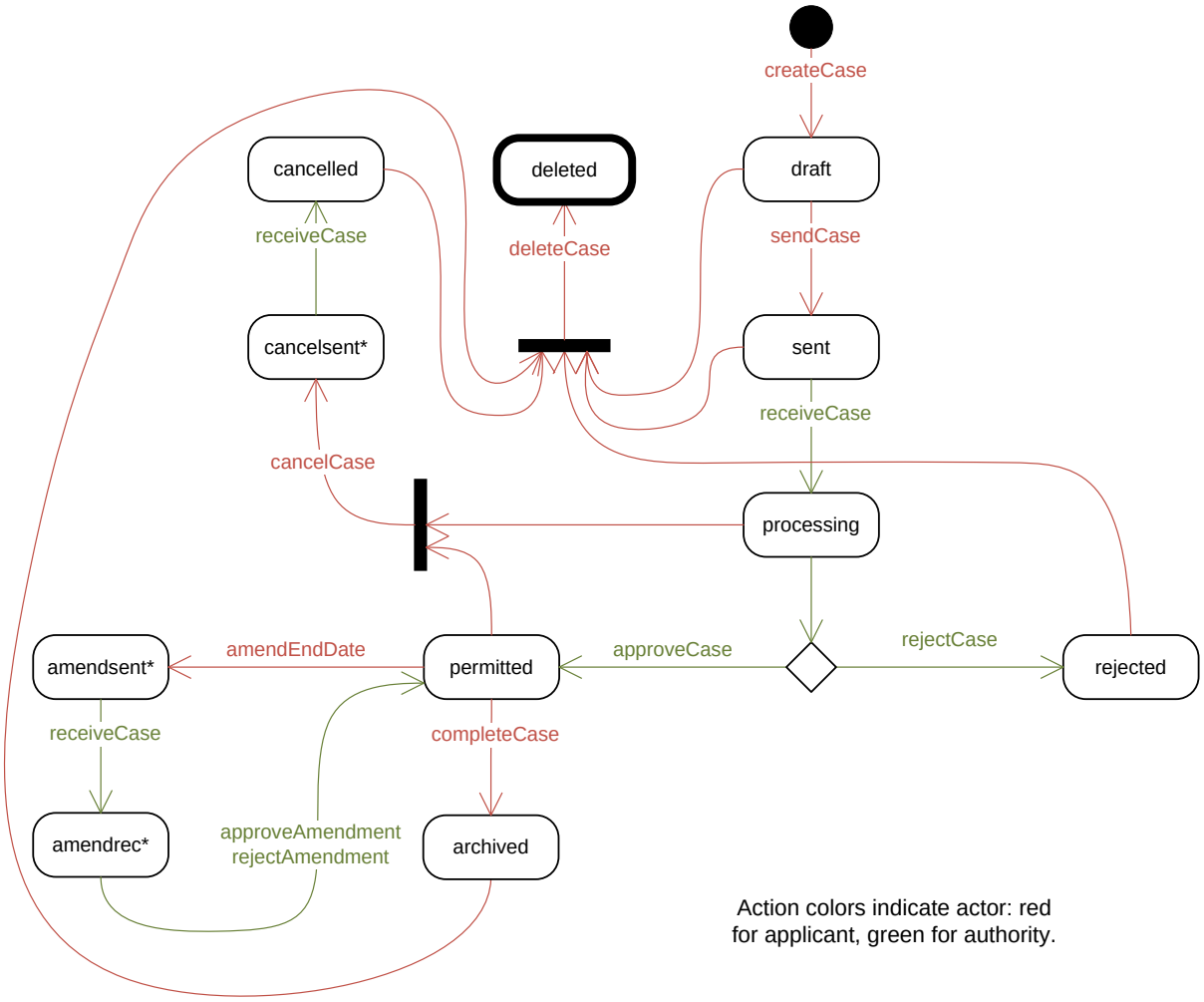


Applicant creates and sends report with completion details. Authority receives report and evaluates its validity. If valid they will approve it, changing state to 'completed'.

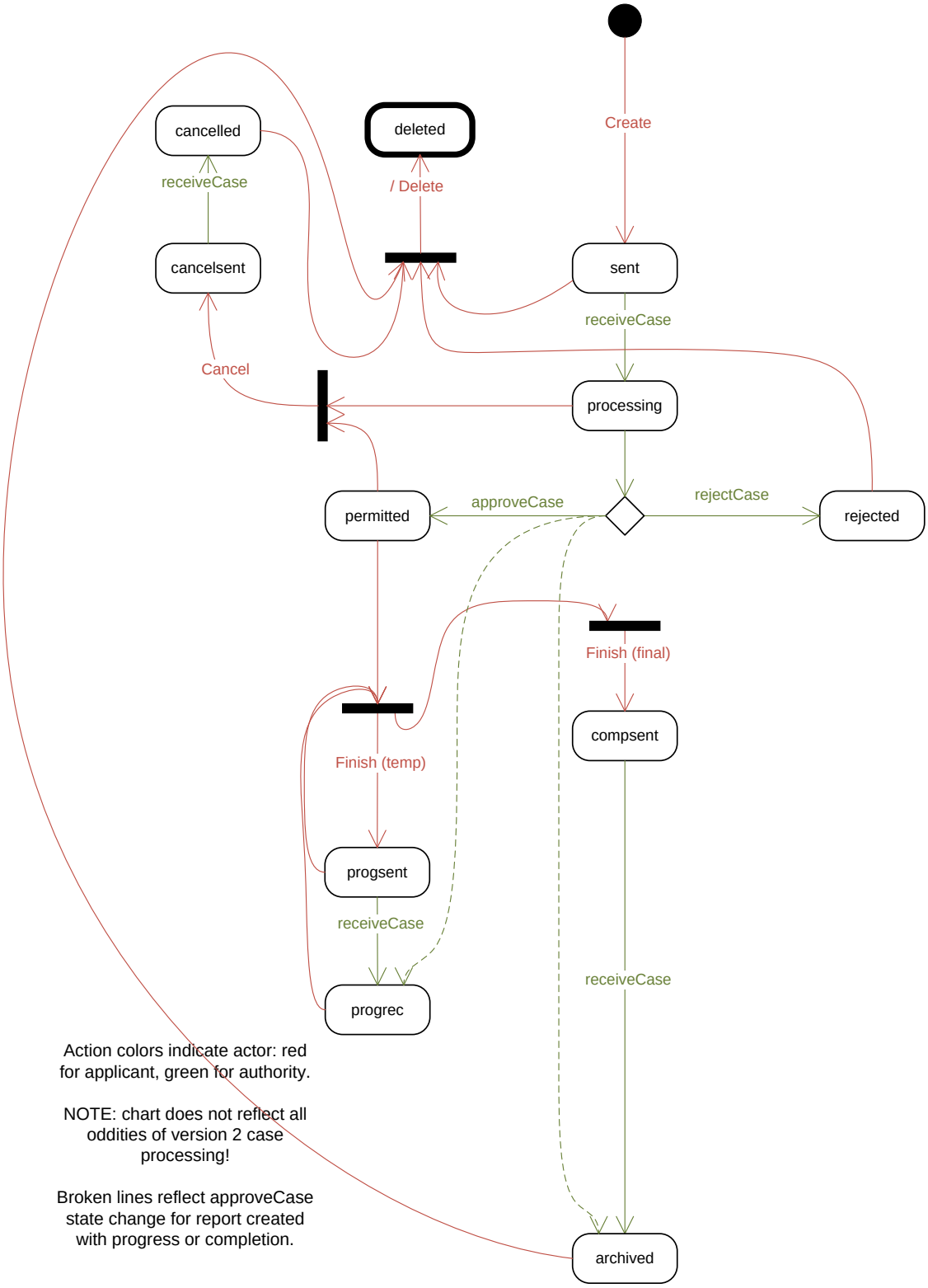
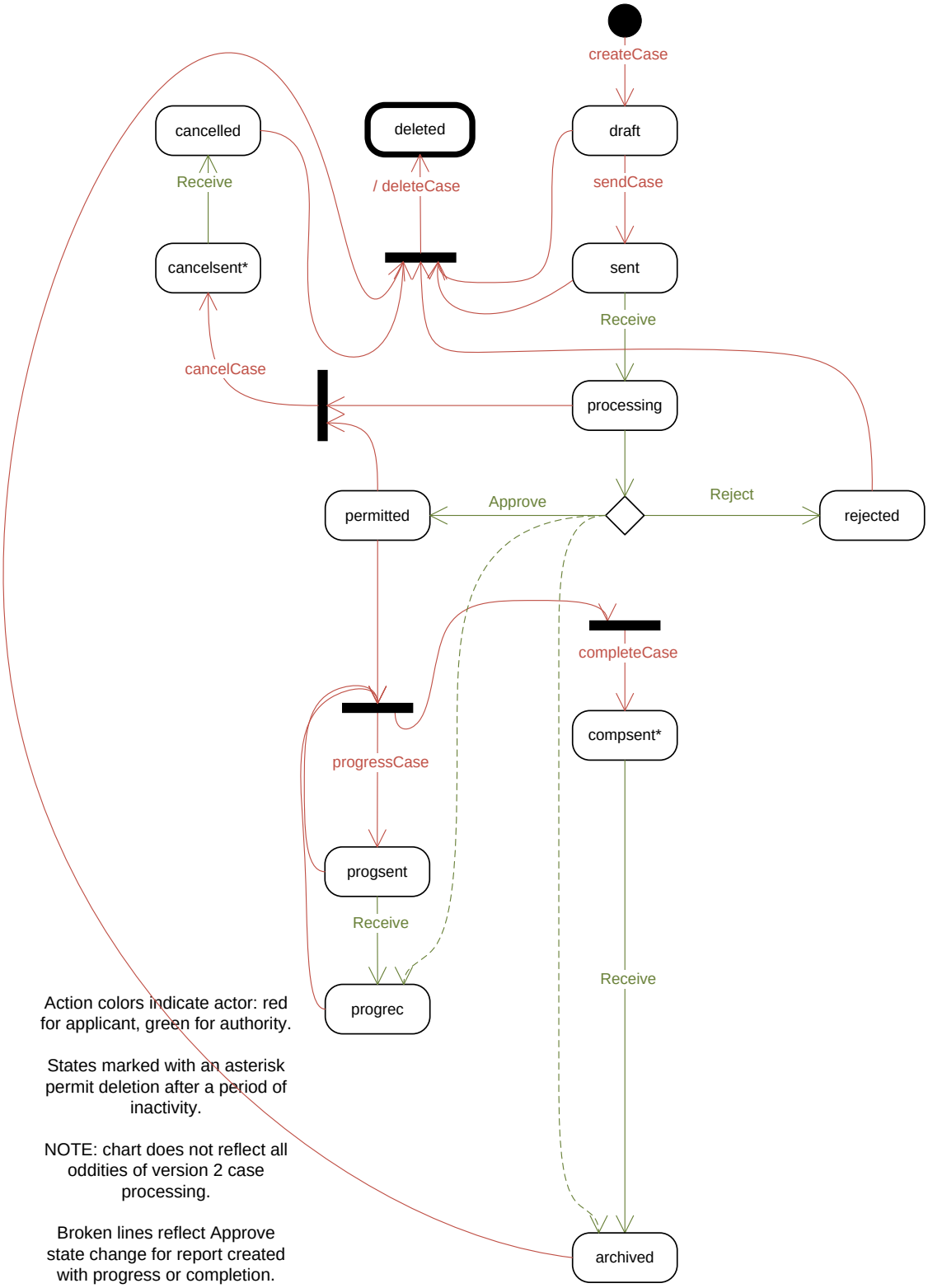
The 'permitted' state is not used in this configuration.

Action colors indicate actor: red for applicant, green for authority.

ROV Application and Report



GT Application and Report, Version 2 Applicant



GT Application and Report, Version 2 Authority