



Frameworkx Specification

Trouble Ticket API REST Specification

TMF621
Release 14.5.1
June 2015

| | |
|-----------------------------------------------|--------------------------|
| Latest Update: Frameworkx Release 14.5 | TM Forum Approved |
| Version 1.3.5 | IPR Mode: RAND |

NOTICE

Copyright © TM Forum 2015. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website, but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

| | |
|--------------------------------------------------------------------------------|----|
| NOTICE..... | 2 |
| Table of Contents..... | 4 |
| List of Tables..... | 5 |
| Introduction | 6 |
| SAMPLE USE CASES..... | 7 |
| RESOURCE MODEL..... | 8 |
| Managed Entity and Task Resource Models | 8 |
| UML model..... | 13 |
| Notification Resource Models | 14 |
| Ticket Status Change Notification | 15 |
| Ticket Change Notification..... | 17 |
| Ticket Clearance Request Notification | 18 |
| Information Required Notification | 20 |
| API OPERATION for tickets..... | 23 |
| GET /API/troubleTicket/{ID}}/{?fields=attributes}&{filtering expression} | 24 |
| PUT API/troubleTicket/{ID} | 26 |
| PATCH API/troubleTicket/{ID} | 30 |
| POST API/ticket | 33 |
| API NOTIFICATION FOR TICKET | 36 |
| REGISTER LISTENER POST /hub | 36 |
| UNREGISTER LISTENER DELETE hub/{id} | 36 |
| publish {EventTYPE} POST /listener | 37 |
| Appendix A. : Alignment with NBN Co. specifications | 38 |
| Release History..... | 41 |
| Acknowledgements..... | 42 |
| Contributors to Document..... | 42 |

LIST OF TABLES

N/A

INTRODUCTION

The following document is the specification of the REST API for the ticket resource. It includes the model definition as well as all available operations. Possible actions are creating and retrieving a Trouble Ticket, updating the whole Trouble Ticket or only do a patch update. Furthermore the HTTP GET allows filtering.

The Trouble ticketing API provides a standardized client interface to Trouble Ticket Management Systems for creating, tracking and managing trouble tickets among partners as a result of an issue or problem identified by a customer or another system. Examples of Trouble Ticket API originators (clients) include CRM applications, network management or fault management systems, or other trouble ticket management systems (e.g. B2B).

The API supports the ability to send requests to create a new trouble ticket specifying the nature and severity of the trouble as well as all necessary related information. The API also includes mechanisms to search for and update existing trouble tickets. Notifications are defined to provide information when a ticket has been updated, including status changes. A basic set of states of a trouble ticket has been specified to handle ticket lifecycle management.

Trouble ticketing API manages Trouble ticket resource:

- A trouble ticket represents a record used for reporting and managing the resolution of resource problems
- Main trouble ticket attributes are its description, severity, type, related dates (creation, target resolution, resolution), state, sub state and related information (change reason and change date), related parties (originator, owner, reviser, etc.) and notes

Trouble ticketing API performs the following operations on trouble ticket

- Retrieval of a trouble ticket or a collection of trouble ticket depending on filter criteria
- Full update of a trouble ticket
- Partial update of a trouble ticket
- Creation of a trouble ticket
- Notification of events on Trouble Ticket:
 - o Ticket state change
 - o Ticket change
 - o Ticket clearance request
 - o Information required

SAMPLE USE CASES

Reader will find examples of use cases using Trouble Ticket API in “Open Digital Business Scenarios and Use Cases” document

RESOURCE MODEL

Managed Entity and Task Resource Models

Example of the JSON representation of a ticket:

```
{
  "id": "1",
  "correlationId": "TT53482",
  "description": "Customer complaint over last invoice.",
  "severity": "Urgent",
  "type": "Bills, charges or payment",
  "creationDate": "2013-07-23T08:16:39.0Z",
  "targetResolutionDate": "2013-07-30T10:20:01.0Z",
  "status": "In Progress",
  "subStatus": "Held",
  "statusChangeReason": "Waiting for invoicing expert.",
  "statusChangeDate": "2013-07-28T08:55:12.0Z",
  "relatedParty": [
    {
      "href": "/customer/1234",
      "role": "Originator"
    },
    {
      "href": "/operator/1234",
      "role": "Owner"
    },
    {
      "href": "Roger Collins",
      "role": "Reviser"
    }
  ],
  "relatedObject": [
    {
      "involvement": "Disputed",
      "reference": "/invoice/1234"
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/5678"
    }
  ],
  "note": [
    {
      "date": "2013-07-24T09:55:30.0Z",
      "author": "Arthur Evans",
      "text": "Already called the expert"
    },
    {
      "date": "2013-07-25T08:55:12.0Z",
      "author": "Arthur Evans",
      "text": "Informed the originator"
    }
  ]
}
```



```
}

```

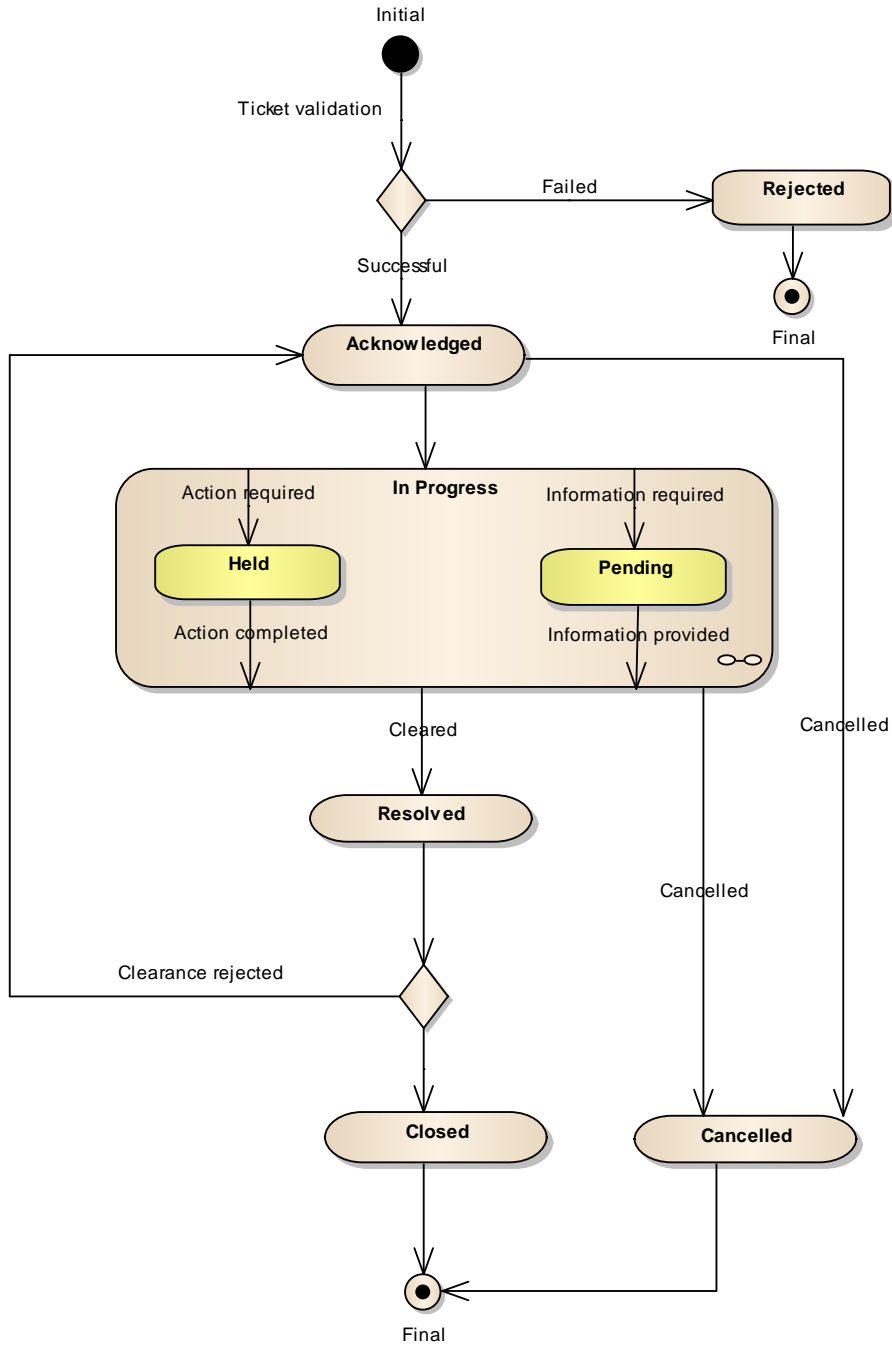
Product Order attributes description:

| Field | Description |
|-----------------------------|-----------------------------------------------------------------------------|
| id | Unique identifier of the trouble ticket |
| correlationId | Additional identifier coming from an external system |
| description | Description of the trouble |
| severity | The severity of the trouble. It can be for example : minor, major, critical |
| type | Type of trouble ticket |
| creationDate | The date on which the trouble was discovered |
| targetResolutionDate | Foreseen trouble resolution date |
| status | The current status of the Trouble Ticket |
| subStatus | The current sub status of the Trouble Ticket |
| statusChangeReason | The reason of state change |
| statusChangeDate | The date of state change |
| resolutionDate | The date on which the service was brought back to its original condition |
| relatedParty | Party playing a role within trouble ticket |
| role | Role of the party |
| reference | Identifier of the party |
| relatedObject | Objects linked with trouble ticket |
| involvement | |
| reference | Identifier of the object |
| note | Extra-information about the trouble ticket |
| date | Date of the note |
| author | Author of the note |
| text | Text of the note |

Trouble Ticket States

Following the available status values for a ticket are listed. The state graphic gives an overview of the allowed status changes.

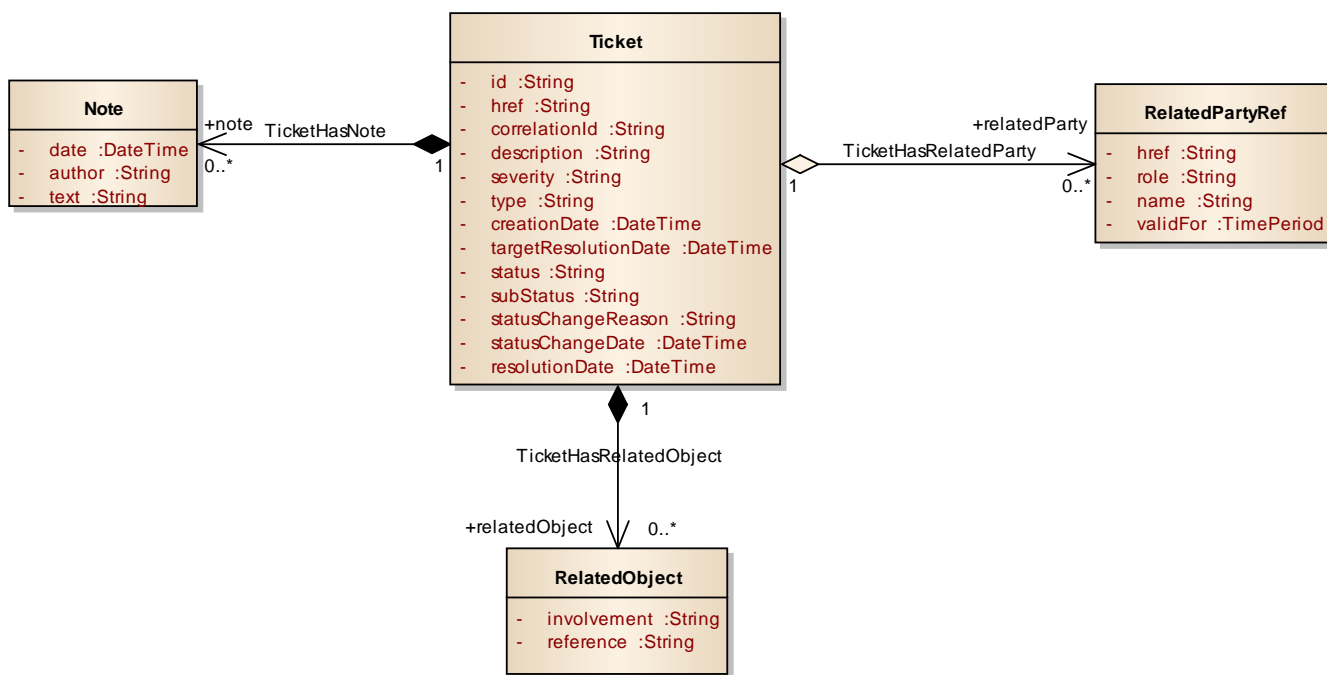
- Submitted
- Rejected
- Acknowledged
- In Progress
 - Held
 - Pending
- Resolved
- Closed
- Cancelled



| State | Description |
|-----------|------------------------------------------------------------------------|
| Submitted | The initial state of a Trouble Ticket when created by a TT originator. |

| | |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Acknowledged | The Trouble Ticket was accepted and allocated a unique Trouble Ticket ID by the TT handler. |
| In Progress | The Trouble Ticket was validated by the TT handler and is being processed. |
| Resolved | The Fault indicated in the Trouble Ticket was corrected by the TT handler and acknowledgement is awaited from its originator. |
| Closed | The Trouble Ticket’s originator has acknowledged the ‘Resolved’ state of the TT, or the timeframe for acknowledgement has passed without response from TT originator. |
| Rejected | The Trouble Ticket was rejected because it: <ul style="list-style-type: none"> • is not submitted • provides invalid information • fails to meet the Business Rules in respect of the Product which originator is raising a Trouble Ticket against • is otherwise defective |
| In Progress - Pending | TT handler is awaiting further confirmation on details of a Fault from originator before it can progress the Fault. An example is where Appointment information is required. |
| In Progress - Held | TT handler is confirming further details internally before completing a Trouble Ticket. An example is where TT handler for network infrastructure spare parts to progress with the Fault rectification. |

UML model

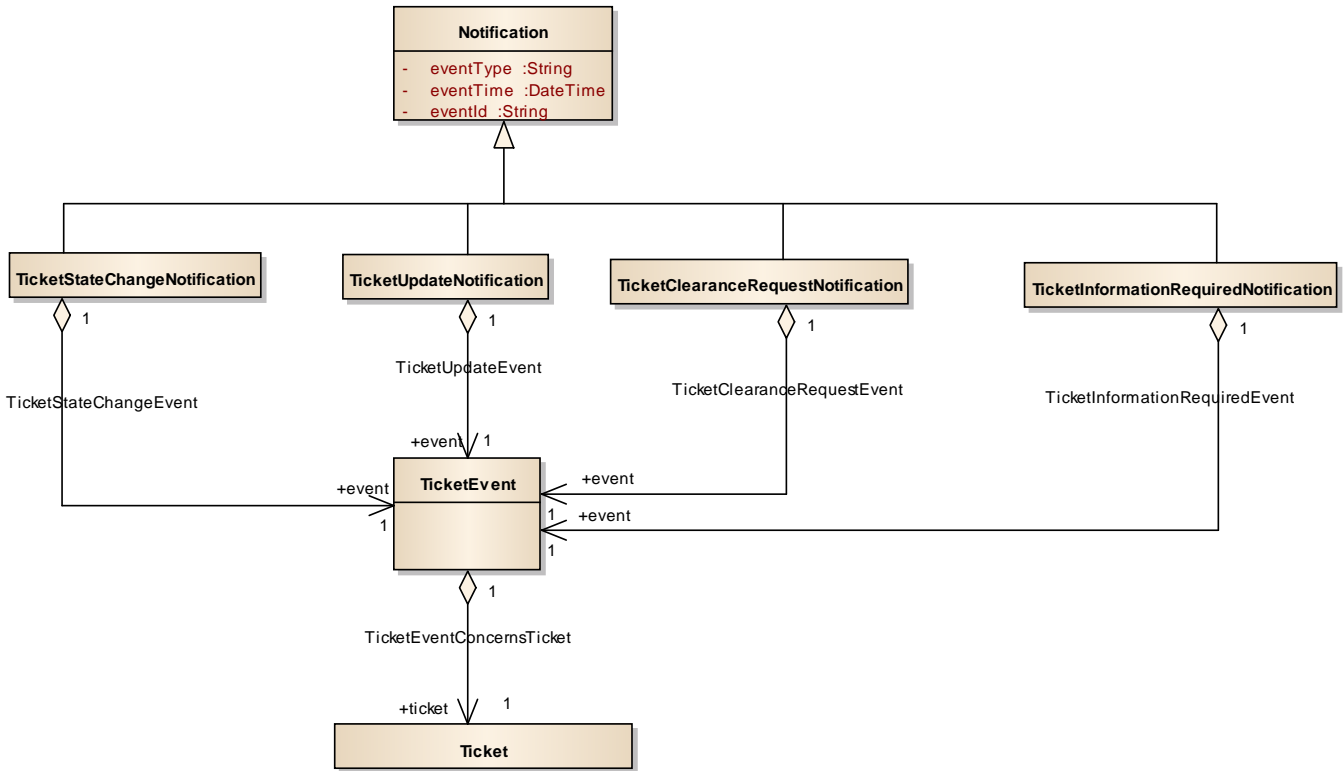


Notification Resource Models

Following the available event types for ticketing:

| TMF REST | |
|----------------------------------------|----------------------------------|
| POST ticket | body : ticket |
| GET ticket | {spec attr } |
| HTTP response to GET | body : ticket |
| TicketChangedNotification | body : ticket |
| TicketStatusChangedNotification | body : ticket |
| TicketStatusChangedNotification | body : ticket |
| TicketStatusChangedNotification | body : ticket |
| TicketClearanceRequestNotification | body : ticket |
| PATCH on status ('resolved'->'closed') | body : ticket.status |
| TicketStatusChangedNotification | body : ticket |
| TicketStatusChangedNotification | body : ticket |
| PATCH ticket | body : ticket attributes |
| HTTP response to PATCH | body : ticket |
| InformationRequiredNotification | body : free text + { spec attr } |

UML Model:



TICKET STATUS CHANGE NOTIFICATION

Decision was made to accept customer complaint and withdraw disputed charges.

```

{
  "eventType": "TicketStatusChangeNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "92445",
  "event": {
    "ticket": {
      "id": "1",
      "correlationId": "TT53482",
      "description": "Customer complaint over last invoice.",
      "severity": "Urgent",
      "type": "Bills, charges or payment",
      "creationDate": "2013-07-23T08:16:39.0Z",
      "targetResolutionDate": "2013-07-30T10:20:01.0Z",
    }
  }
}

```

```
"status": "Resolved",
"subStatus": "",
"statusChangeReason": "Waiting for invoicing expert.",
"statusChangeDate": "2013-07-25T08:55:12.0Z",
"relatedParty": [
  {
    "href": "/customer/1234",
    "role": "Originator"
  },
  {
    "href": "/operator/1234",
    "role": "Owner"
  },
  {
    "href": "Roger Collins",
    "role": "Reviser"
  }
],
"relatedObject": [
  {
    "involvement": "Disputed",
    "reference": "/invoice/1234"
  },
  {
    "involvement": "Adjusted",
    "reference": "/invoice/5678"
  }
],
"note": [
  {
    "date": "2013-07-24T09:55:30.0Z",
    "author": "Arthur Evans",
    "text": "Already called the expert"
  },
  {
    "date": "2013-07-25T08:55:12.0Z",
    "author": "Arthur Evans",
```



```

        "text": "Informed the originator"
      }
    ]
  }
}

```

TICKET CHANGE NOTIFICATION

Call with customer clarifies that dispute relates to another invoice than initially understood.

```

{
  "eventType": "TicketChangeNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "8976",
  "event": {
    "ticket": {
      "id": "1",
      "correlationId": "TT53482",
      "description": "Customer complaint over last invoice.",
      "severity": "Urgent",
      "type": "Bills, charges or payment",
      "creationDate": "2013-07-23T08:16:39.0Z",
      "targetResolutionDate": "2013-07-30T10:20:01.0Z",
      "status": "Resolved",
      "subStatus": "",
      "statusChangeReason": "Wrong invoice.",
      "statusChangeDate": "2013-07-25T13:35:12.0Z",
      "relatedParty": [
        {
          "href": "/customer/1234",
          "role": "Originator"
        },
        {
          "href": "/operator/1234",

```

```
        "role": "Owner"
      },
      {
        "href": "Roger Collins",
        "role": "Reviser"
      }
    ],
    "relatedObject": [
      {
        "involvement": "Disputed",
        "reference": "/invoice/1234"
      },
      {
        "involvement": "Adjusted",
        "reference": "/invoice/4518"
      }
    ],
    "note": [
      {
        "date": "2013-07-24T09:55:30.0Z",
        "author": "Arthur Evans",
        "text": "Already called the expert"
      },
      {
        "date": "2013-07-25T08:55:12.0Z",
        "author": "Arthur Evans",
        "text": "Informed the originator"
      }
    ]
  }
}
```

TICKET CLEARANCE REQUEST NOTIFICATION

As a result of disputed charges withdrawal, ticket is assumed to be resolved. Notification asks Ticket requester to confirm resolution (by setting status to "Closed").

```
{
  "eventType": "TicketClearanceRequestNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "8976",
  "event": {
    "ticket": {
      "id": "1",
      "correlationId": "TT53482",
      "description": "Customer complaint over last invoice.",
      "severity": "Urgent",
      "type": "Bills, charges or payment",
      "creationDate": "2013-07-23T08:16:39.0Z",
      "targetResolutionDate": "2013-07-30T10:20:01.0Z",
      "status": "Resolved",
      "subStatus": "",
      "statusChangeReason": "Customer dispute resolved.",
      "statusChangeDate": "2013-07-25T13:35:12.0Z",
      "relatedParty": [
        {
          "href": "/customer/1234",
          "role": "Originator"
        },
        {
          "href": "/operator/1234",
          "role": "Owner"
        },
        {
          "href": "Roger Collins",
          "role": "Reviser"
        }
      ],
      "relatedObject": [
        {
          "involvement": "Disputed",
          "reference": "/invoice/1234"
        }
      ]
    }
  }
}
```

```
    },
    {
      "involvement": "Adjusted",
      "reference": "/invoice/4518"
    }
  ],
  "note": [
    {
      "date": "2013-07-24T09:55:30.0Z",
      "author": "Arthur Evans",
      "text": "Already called the expert"
    },
    {
      "date": "2013-07-25T08:55:12.0Z",
      "author": "Arthur Evans",
      "text": "Informed the originator"
    }
  ]
}
}
```

INFORMATION REQUIRED NOTIFICATION

Call with customer clarifies that dispute relates to another invoice than initially understood.

```
{
  "eventType": "InformationRequiredNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "8976",
  "event": {
    "ticket": {
      "id": "1",
      "correlationId": "TT53482",
      "description": "Customer complaint over last invoice."
    }
  }
}
```

```
"severity": "Urgent",
"type": "Bills, charges or payment",
"creationDate": "2013-07-23T08:16:39.0Z",
"targetResolutionDate": "2013-07-30T10:20:01.0Z",
"status": "In Progress",
"subStatus": "Pending",
"statusChangeReason": "Wrong invoice reference number.",
"statusChangeDate": "2013-07-24T13:35:12.0Z",
"relatedParty": [
  {
    "href": "/customer/1234",
    "role": "Originator"
  },
  {
    "href": "/operator/1234",
    "role": "Owner"
  },
  {
    "href": "Roger Collins",
    "role": "Reviser"
  }
],
"relatedObject": [
  {
    "involvement": "Disputed",
    "reference": "/invoice/1234"
  },
  {
    "involvement": "Adjusted",
    "reference": "/invoice/xxxx"
  }
],
"note": [
  {
    "date": "2013-07-24T09:55:30.0Z",
    "author": "Arthur Evans",
    "text": "Already called the expert"
```

```
    },
    {
      "date": "2013-07-25T08:55:12.0Z",
      "author": "Arthur Evans",
      "text": "Informed the originator"
    },
    {
      "date": "2013-07-25T09:35:12.0Z",
      "author": "Arthur Evans",
      "text": "Need reference of the right invoice"
    }
  ]
}
}
```

API OPERATION FOR TICKETS

| Operation on Entities | Uniform API Operation | Description |
|--------------------------------|-----------------------|---------------------------------------------------------------------------------|
| Query Entities | GET Resource | GET must be used to retrieve a representation of a resource. |
| Create Entity | POST Resource | POST must be used to create a new resource |
| Partial Update of an Entity | PATCH Resource | PATCH must be used to partially update a resource |
| Complete Update of an Entity | PUT Resource | PUT must be used to completely update a resource identified by its resource URI |
| Remove an Entity | DELETE Resource | DELETE must be used to remove a resource |
| Execute an Action on an Entity | POST on TASK Resource | POST must be used to execute Task Resources |
| Other Request Methods | POST on TASK Resource | GET and POST must not be used to tunnel other request methods. |

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

GET /API/troubleTicket/{ID}}/{?fields=attributes}&{filtering expression}

This Uniform Contract operation is used to retrieve the representation of a ticket.

Note that collections can be retrieved via GET /API/TROUBLETICKET/ with no {ID}

Filtering is allowed on all attributes. See example below.

Attribute selection is possible for all attributes. See example below.

| |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| REQUEST |
| <pre>GET /api/troubleTicket Accept: application/json</pre> |
| RESPONSE |
| <pre>200 Content-Type: application/json [{ "id": "1", "correlationId": "TT53482", "description": "Customer complaint over last invoice.", "severity": "Urgent", "type": "Bills, charges or payment", "creationDate": "2013-07-23T08:16:39.0Z", "targetResolutionDate": "2013-07-30T10:20:01.0Z", "status": "In Progress", "subStatus": "Held", "statusChangeReason": "Waiting for invoicing expert.", "statusChangeDate": "2013-07-24T08:55:12.0Z", "relatedParty": [{ "href": "/customer/1234", "role": "Originator" }, { "href": "/operator/1234", "role": "Owner" }] }</pre> |


```
        "href": "Roger Collins",
        "role": "Reviser"
    }
],
"relatedObject": [
    {
        "involvement": "Disputed",
        "reference": "/invoice/1234"
    },
    {
        "involvement": "Adjusted",
        "reference": "/invoice/5678"
    }
],
"note": [
    {
        "date": "2013-07-24T09:55:30.0Z",
        "author": "Arthur Evans",
        "text": "Already called the expert"
    },
    {
        "date": "2013-07-25T08:55:12.0Z",
        "author": "Arthur Evans",
        "text": "Informed the originator"
    }
]
},
{
    "id": "2",
    "correlationId": "",
    "description": "Customer asks for information about upgrading products.",
    "severity": "Low",
    "type": "Products and Services",
    "creationDate": "2013-07-25T08:16:00.0Z",
    "targetResolutionDate": "2013-07-28T00:00:00.0Z",
    "status": "",
    "subStatus": "",
```

```
"statusChangeReason": "",
"statusChangeDate": "2013-07-25T08:16:00.0Z",
"resolutionDate": "",
"relatedParty": [
  {
    "href": "/customer/4567",
    "role": "Originator"
  }
],
"relatedObject": [
],
"note": [
]
}
]
```

Retrieving all tickets – returns an array/ a list of tickets:

- GET /api/ticket

Retrieving all tickets which were closed after 2013-05-01:

- GET /api/ticket?status=Close&statusResolutionDate.gt=2013-05-01

Retrieve ticket with specified ID – only one ticket is returned:

- GET /api/ticket/1

Retrieve ticket with ID but only the attributes status and statusChangeReason in the response, separator for attributes is comma:

- GET /api/ticket/1/status,statusChangeReason

PUT API/TROUBLETICKET/{ID}

Description :

- This Uniform Contract operation is used to completely update the representation of a ticket.
- Resource represents a managed entity.

Behavior :

- Returns HTTP/1.1 status code 201 if the request was successful.
- Returns HTTP/1.1 status code 400 (Bad request) if content is invalid (missing required attributes, ...).

Updating the whole ticket – if you try to change the ticket ID itself an exception is returned. All fields with different values will be changed. If the request contains the same values like the current ticket representation, nothing is changed. If an element is empty in the request, the value of the element will be deleted. If it is a required element, an exception is returned.

| REQUEST |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PUT API/ troubleTicket/1 Content-type: application/json |
| <pre>{ "id": "1", "correlationId": "TT53482", "description": "Customer complaint over last invoice.", "severity": "Urgent", "type": "Bills, charges or payment", "creationDate": "2013-07-23T08:16:39.0Z", "targetResolutionDate": "2013-07-30T10:20:01.0Z", "status": "In Progress", "subStatus": "Held", "statusChangeReason": "Waiting for invoicing expert.", "statusChangeDate": "2013-07-24T08:55:12.0Z", "relatedParty": [{ "href": "/customer/1234", "role": "Originator" }, { "href": "/operator/1234", "role": "Owner" }, { "href": "Roger Collins", "role": "Reviser" }] },</pre> |

```
"relatedObject": [  
  {  
    "involvement": "Disputed",  
    "reference": "/invoice/1234"  
  },  
  {  
    "involvement": "Adjusted",  
    "reference": "/invoice/5678"  
  }  
],  
"note": [  
  {  
    "date": "2013-07-24T09:55:30.0Z",  
    "author": "Arthur Evans",  
    "text": "Already called the expert"  
  },  
  {  
    "date": "2013-07-25T08:55:12.0Z",  
    "author": "Arthur Evans",  
    "text": "Informed the originator"  
  }  
]  
}
```

RESPONSE

201

Content-Type: application/json

```
{  
  "id": "1",  
  "correlationId": "TT53482",  
  "description": "Customer complaint over last invoice.",  
  "severity": "Urgent",  
  "type": "Bills, charges or payment",  
  "creationDate": "2013-07-23T08:16:39.0Z",  
  "targetResolutionDate": "2013-07-30T10:20:01.0Z",  
  "status": "In Progress",  
  "subStatus": "Held",  
}
```

```
"statusChangeReason": "Waiting for invoicing expert.",
"statusChangeDate": "2013-07-24T08:55:12.0Z",
"relatedParty": [
  {
    "href": "/customer/1234",
    "role": "Originator"
  },
  {
    "href": "/operator/1234",
    "role": "Owner"
  },
  {
    "href": "Roger Collins",
    "role": "Reviser"
  }
],
"relatedObject": [
  {
    "involvement": "Disputed",
    "reference": "/invoice/1234"
  },
  {
    "involvement": "Adjusted",
    "reference": "/invoice/5678"
  }
],
"note": [
  {
    "date": "2013-07-24T09:55:30.0Z",
    "author": "Arthur Evans",
    "text": "Already called the expert"
  },
  {
    "date": "2013-07-25T08:55:12.0Z",
    "author": "Arthur Evans",
    "text": "Informed the originator"
  }
]
```

```

    ]
}
    
```

Example see TMF REST Design Guidelines.

PATCH API/TROUBLETICKET/{ID}

Description :

- This Uniform Contract operation is used to partially update the representation of a ticket.
- Resource represents a managed entity.

Behavior :

Patching of status is only allowed to transition :

- from “resolved” to “closed”,
- from “created”, “acknowledged” or “in progress” to “cancelled”

Patching of relatedParties[], relatedObjects[] or notes[] replaces previous value set.

| Attribute name | Patchable | Rule |
|----------------------|-----------|------|
| id | N | |
| correlationId | N | |
| description | Y | |
| severity | Y | |
| type | Y | |
| creationDate | N | |
| targetResolutionDate | Y | |
| status | Y | |
| subStatus | Y | |

| | | |
|--------------------|---|--|
| statusChangeReason | Y | |
| statusChangeDate | Y | |
| resolutionDate | Y | |
| relatedParty | Y | |
| relatedObject | Y | |
| note | Y | |

Further document any rules that must be implemented when patching attributes.

| Rule name | Rule/Pre Condition/Side Effects/Post Conditions |
|-----------------|-------------------------------------------------|
| To be completed | To be completed |

| |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>REQUEST</p> <p>PATCH API/ troubleTicket /{ID} Content-type: application/json</p> <pre>{ "status": "Close", "statusChangeReason": "Dispute has been resolved in favor of the customer." }</pre> |
| <p>RESPONSE</p> <p>201 Content-Type: application/json</p> <pre>{ "id": "1", "correlationId": "TT53482", "description": "Customer complaint over last invoice.", "severity": "Urgent", "type": "Bills, charges or payment", "creationDate": "2013-07-23T08:16:39.0Z", }</pre> |

```
"targetResolutionDate": "2013-07-30T10:20:01.0Z",
"status": "Closed",
"subStatus": "",
"statusChangeReason": "Dispute has been resolved in favor of the customer.",
"statusChangeDate": "2013-07-24T08:55:12.0Z",
"relatedParty": [
  {
    "href": "/customer/1234",
    "role": "Originator"
  },
  {
    "href": "/operator/1234",
    "role": "Owner"
  },
  {
    "href": "Roger Collins",
    "role": "Reviser"
  }
],
"relatedObject": [
  {
    "involvement": "Disputed",
    "reference": "/invoice/1234"
  },
  {
    "involvement": "Adjusted",
    "reference": "/invoice/5678"
  }
],
"note": [
  {
    "date": "2013-07-24T09:55:30.0Z",
    "author": "Arthur Evans",
    "text": "Already called the expert"
  },
  {
    "date": "2013-07-25T08:55:12.0Z",
```



```

        "author": "Arthur Evans",
        "text": "Informed the originator"
    }
]
}
    
```

POST API/TICKET

Description :

- This Uniform Contract operation is used to create a ticket.
- Resource represents a managed entity.
- Mandatory attributes that must be provided when you create the ticket :

Description, severity, type

Behavior :

- Returns HTTP/1.1 status code 201 if the request was successful.
- Returns HTTP/1.1 status code 400 (Bad request) if content is invalid (missing required attributes, ...).

| Attribute name | Mandatory | Default | Rule |
|----------------------|-----------|-----------------|---------------------------|
| Id | N | | Initialized by TT handler |
| correlationId | N | | |
| description | Y | | |
| severity | Y | | |
| type | Y | | |
| creationDate | N | Date of the day | Initialized by TT handler |
| targetResolutionDate | N | | |
| status | N | Created | Initialized by TT handler |

| | | | |
|--------------------|---|-----------------|--|
| subStatus | N | | |
| statusChangeReason | N | | |
| statusChangeDate | N | Date of the day | |
| resolutionDate | N | | |
| relatedParty | N | | |
| relatedObject | N | | |
| Note | N | | |

Further specify any rules on the creation of the entity

| Rule name | Rule |
|-----------------|-----------------|
| To be completed | To be completed |

The requester should not provide an id – it is generated automatically by the TT handler. A correlationId can be provided, but is not mandatory.

Create a ticket only with mandatory attributes:

| |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| REQUEST |
| POST API/ troubleTicket Content-type: application/json <pre>{ "description": "Customer complaint over last invoice.", "severity": "Urgent", "type": "Bills, charges or payment" }</pre> |
| RESPONSE |
| 201 Content-Type: application/json <pre>{</pre> |

```
"id": "1",
"correlationId": "TT53482",
"description": "Customer complaint over last invoice.",
"severity": "Urgent",
"type": "Bills, charges or payment",
"creationDate": "2013-07-23T08:16:39.0Z",
"targetResolutionDate": "2013-07-30T10:20:01.0Z",
"status": "Created",
"subStatus": "",
"statusChangeReason": "",
"statusChangeDate": "2013-07-24T08:55:12.0Z",
"resolutionDate": "",
"relatedParty": [
],
"relatedObject": [
],
"note": [
]
}
```

API NOTIFICATION FOR TICKET

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

REGISTER LISTENER POST /HUB

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

| |
|------------------------------------------------------------------------------------------------------------------------------------------------------|
| REQUEST |
| POST /api/hub Accept: application/json <pre>{ "callback": "http://in.listener.com" }</pre> |
| RESPONSE |
| 201 Content-Type: application/json Location: /api/hub/42 <pre>{ "id": "42", "callback": "http://in.listener.com", "query": null }</pre> |

UNREGISTER LISTENER DELETE HUB/{ID}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

| |
|--------------------------------------------------|
| REQUEST |
| DELETE /api/hub/{id} Accept: application/json |
| RESPONSE |
| 204 |

PUBLISH {EVENTTYPE} POST /LISTENER

Description :

Provide the Event description

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

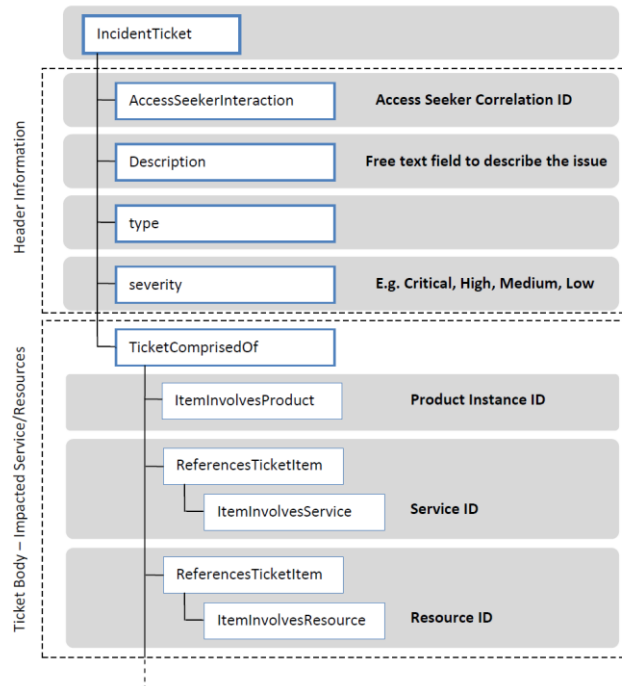
| |
|------------------------------------------------------------------------------------------------------------------------------------------------------|
| REQUEST |
| POST /client/listener Accept: application/json |
| <pre>{ "eventType": "EventType", "eventTime": "2014-09-27T05:46:25.0Z", "eventId": "1562231", "event": { EVENT BODY } }</pre> |
| RESPONSE |
| 201 Content-Type: application/json |

Example see TMF REST Design Guidelines.

APPENDIX A. : ALIGNMENT WITH NBN CO. SPECIFICATIONS

The following diagram From NBN Co shows the generic, high level structure for a Trouble Ticket submission message.

Unlike NBNCo, **TMF Ticket is atomic, that is it does NOT contain TicketItems.**



| NBNCo | TM Forum | Comments |
|-------------------------------------------|--------------------------------|---------------------------------|
| AccessSeekerInteraction | <i>correlationId</i> | |
| Description | <i>description</i> | |
| Type | <i>type</i> | |
| Severity | <i>severity</i> | |
| interactionDate | <i>creationDate</i> | |
| interactionDateComplete | not supported | use <i>statusChangeDate</i> |
| plannedCompletionDate | <i>targetResolutionDate</i> | |
| interactionStatus | <i>status</i> | |
| interactionSubStatus | <i>substatus</i> | |
| | <i>statusChangeReason</i> | |
| | <i>statusChangeDate</i> | |
| resolvedDate | <i>resolutionDate</i> | |
| | <i>Ticket.relatedParties[]</i> | Refers to end-user, CSR, ... |
| TicketItem.InvolvesProduct/Service | <i>Ticket.relatedObjects[]</i> | Product, Service, Resource, ... |

The following tables list the supported/non supported processes and touchpoints as defined in

NBNCo - B2B Interaction Business Processes – Technical Specification, 02/01/2013

| TT-BP001 : Assurance Ticket Process | | supported |
|--------------------------------------------|---------------------------------|---------------------------------------------|
| PH-TP001 | requestTroubleTicketCreate | POST ticket |
| PH-TP002 | queryTroubleTicketDetails | GET ticket |
| PH-TP002.1 | responseTroubleTicketDetails | HTTP response to GET |
| PH-TP004 | notifyKeepCustomerInformed | TicketChangedNotification |
| PH-TP005 | notifyTroubleTicketAcknowledged | TicketStatusChangedNotification |
| PH-TP006 | notifyTroubleTicketAccepted | TicketStatusChangedNotification |
| PH-TP007 | notifyTroubleTicketRejected | TicketStatusChangedNotification |
| PH-TP030 | requestTroubleTicketClearance | TicketClearanceRequestNotification |
| PH-TP030.1 | responseTroubleTicketClearance | PATCH ticket.status ('resolved'-'>'closed') |
| PH-TP014 | notifyTroubleTicketResolved | TicketStatusChangedNotification |
| PH-TP020 | notifyInformationRequired | InformationRequiredNotification |
| PH-TP022 | notifyTroubleTicketClosed | TicketStatusChangedNotification |

| TT-BP003 : Query Trouble Ticket History or Details | | supported (*) |
|-----------------------------------------------------------|------------------------------|----------------------|
| PH-TP002 | queryTroubleTicketDetails | GET ticket |
| PH-TP002.1 | responseTroubleTicketDetails | HTTP response to GET |

(*) history not supported

| TT-BP005 : TroubleTicketAmendment | | supported |
|------------------------------------------|----------------------------|------------------|
| PH-TP011 | requestTroubleTicketAmend | PATCH ticket |
| PH-TP011.1 | responseTroubleTicketAmend | HTTP response |

| TT-BP006 : Trouble ticket Jeopardy | | not supported |
|-------------------------------------------|------------------------|----------------------|
| PH-TP004 | notifyCustomerJeopardy | not supported |

| TT-BP007 : Planned changed / hazard | | not supported |
|--------------------------------------------|----------------------------|----------------------|
| CM-TP001 | notifyPlannedChange | not supported |
| CM-TP004 | notifyKeepCustomerInformed | not supported |

| TT-BP008 : Notify Network Fault | | not supported |
|----------------------------------------|----------------------------|----------------------|
| PH-TP004 | notifyTroubleTicketCreated | not supported |

| TT-BP009 : RequestMoreTime | | not supported |
|-----------------------------------|-----------------------------------|---------------------------------|
| PH-TP020 | notifyInformationRequired | InformationRequiredNotification |
| PH-TP004 | notifyKeepCustomerInformed | TicketChangedNotification |
| PH-TP025 | requestMoreTime | not supported |
| PH-TP025.1 | responseMoreTime | not supported |
| PH-TP026 | notifyInformationRequiredReminder | not supported |
| PH-TP022 | notifyTroubleTicketClosed | TicketStatusChangedNotification |

| TT-BP010 : QueryTroubleTicketAttachment | | not supported |
|------------------------------------------------|---------------------------------|----------------------|
| PH-TP029 | queryTroubleTicketAttachment | not supported |
| PH-TP029.1 | responseTroubleTicketAttachment | not supported |

RELEASE HISTORY

| Version Number | Date | Release led by: | Description |
|----------------|-------------|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Version 1.0 | 04/15/2013 | Pierre Gauthier TM Forum pgauthier@tmforum.org | First Release of Draft Version of the Document. |
| Version 1.1 | 07/25/2013 | Christian Traxler Infonova christian.traxler@infonova.com | Spec Jam workshop results |
| Version 1.2 | 07/26/2013 | Jérôme Hannebelle Orange jerome.hannebelle@orange.com | Modifications post-spec Jam, with the addition of NBNC Co support appendix. |
| Version 1.3 | 2 Oct 2013 | Tina O'Sullivan (TM Forum) | Minor modifications |
| Version 1.3.3 | 11 Oct 2013 | Alicja Kawecki (TM Forum) | Minor fixes to cover and footer |
| Version 1.3.4 | Sept 2014 | J-L Tymen, T Ben Meriem, J Hannebelle | Re-aligned with API new template |
| Version 1.3.5 | Jun 2015 | Alicja Kawecki | Updated cover, footer and Notice to reflect TM Forum Approved status |

ACKNOWLEDGEMENTS

| Release Number | Date | Contributor | Company | Email |
|----------------|----------------------------|-------------------|----------|------------------------------------------------------------------------------------|
| 1.0 | 27 th July 2013 | Christian Traxler | Infonova | Christian.Traxler@infonova.com |
| | | Jean Luc Tymen | Orange | jeanluc.tymen@orange.com |
| | | Andreas Polz | Infonova | andreas.polz@infonova.com |
| | | John Storrie | O2 | John.Storrie@o2.com |
| | | Jerome Hannebelle | Orange | jerome.hannebelle@orange.com |
| | | Pierre Gauthier | TM Forum | pgauthier@tmforum.org |
| | | | | |

CONTRIBUTORS TO DOCUMENT

| | |
|-------------------|------------|
| Veronique Mauneau | Orange |
| Jean-Luc Tymen | Orange |
| Pierre Gauthier | TM Forum |
| John Morey | Ciena |
| Cliff C Faurer | AMKB Cloud |