

# Incident Management for Network Services

CCAMP WG, IETF116

**draft-feng-opsawg-incident-management-00**

## **Authors:**

Chong Feng(Huawei)

Tong Hu(CMCC)

Luis Contreras(Telefonica)

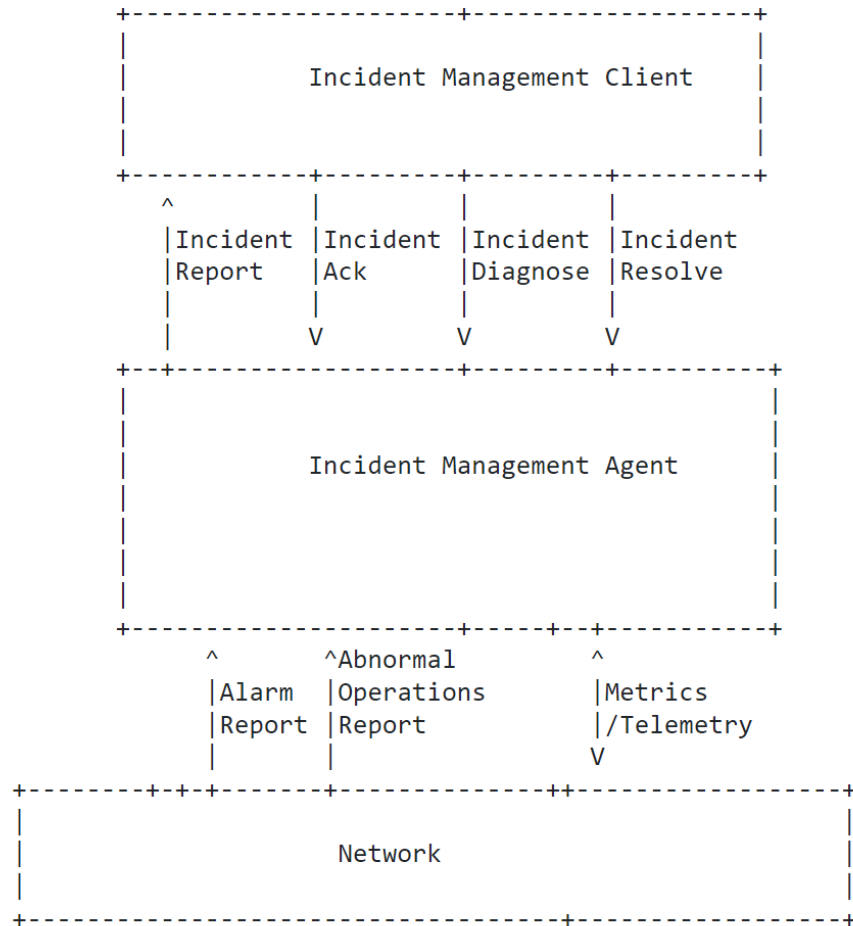
Qin Wu(Huawei)

Chaode Yu(Huawei)

# Motivation of This Draft

- **The traditional alarm management approach is not sustainable.**
  - ✓ Human experience dependency & low efficiency
  - ✓ Duplicated tickets are dispatched
  - ✓ Inaccurate root cause analysis
  
- **Incident management can help to reduce work.**
  - ✓ Root cause alarm and correlative alarms follow service & connection modeling, they can be compressed into an incident;
  - ✓ Performance data and trace log can also help for root cause analysis;
  - ✓ Technologies like AI and ML can help to deal with complicated correlation;
  - ✓ Some technology can help to fault locating;
  
- **We proposed a new incident-based management solution.**
  - ✓ Architecture of incident management;
  - ✓ Requirement between different layers;
  - ✓ YANG data model applied to the incident management agent and client;

# Incident Management Architecture



- Incident management **agent**: network analytics platform/controllers /Orchestrators;  
provides functionalities such as incident detection, report, diagnosis, resolution, querying for incident lifecycle management.
- Incident management **client**: network OSS or other business systems of operators;  
invokes the functionalities provided by incident management agent to meet the business requirements of fault management.
- The **network layer** support to report alarms & trace log & performance data as before. Smaller frequency data collection may be needed.

The main functionalities between incident management agent and client include:

- Incident report & acknowledge
- Incident diagnose
- Incident resolve

# Incident Management Yang Data Model

```
module: ietf-incident
+--ro incidents
+--ro incident* [incident-id]
  +--ro incident-id      string
  +--ro csn              uint64
  +--ro service-instance* string
  +--ro name              string
  +--ro type              enumeration
  +--ro domain            identityref
  +--ro priority          incident-priority
  +--ro status?          enumeration
  +--ro ack-status?      enumeration
  +--ro category          identityref
  +--ro tenant?          string
  +--ro detail?          string
  +--ro resolve-suggestion? string
  +--ro sources
  | ...
  +--ro root-causes
  | ...
  +--ro events
  | ...
  +--ro raise-time?      yang:date-and-time
  +--ro occur-time?     yang:date-and-time
  +--ro clear-time?     yang:date-and-time
  +--ro ack-time?       yang:date-and-time
  +--ro last-updated?   yang:date-and-time
```

```
rpcs:
  +---x incident-acknowledge
  | ...
  +---x incident-diagnose
  | ...
  +---x incident-resolve
  ...
notifications:
  +---n incident-notification
    +--ro incident-id?      string
    ...
```

- Currently we have defined the detail information of incident and how to report it through notification
- We pretend to define some RPC to support incident acknowledge, diagnosis, resolving functionalities;

# Next Step

- Confirm working group
- Define the RPC interfaces to support the whole solution
- Call for interest & joint contribution

Thank You !