Documenting server capabilities

› Server capabilities are often readable via Netconf/Restconf
  – Ietf-yang-library: Modules, revisions, features, deviations, datastores
  – Ietf-netconf-monitoring: Netconf-capabilities, etc.
  – Alarms supported
  – YangPush on-change notification capabilities

› Most are set/fixed at implementation time and don’t change

› Server capabilities needed in implementation-time
  – To start the NMS implementation in parallel
  – Operator buying decision may depend on the OAM capabilities,

› Standard formal YANG based form of documentation preferred
  – SW / automation needs to read capabilities
  – Makes capabilities available both via Netconf and via documentation
    › Same format on Netconf and documents
  – Encourages easy to read uniform documentation
We need to know which YANG modules and which features will be supported by the server.

Will it support my central radius based authentication?

```
{
  "instance-data": {
    "ietf-yang-library:modules-state": {
      "module": [
        {
          "name": "ietf-system",
          "revision": "2014-08-06",
          "namespace": "urn:...ietf-system",
          "feature": ["authentication", "radius-authentication"]
        }
      ]
    }
  }
}```
Preloading Data

› Vendors often define sets of default data for configuration
  – Default Access control groups
    › readOnlyUser, systemAdministrator, securityAdministrator
  – Default rule-lists, rules for the default groups

› Default monitoring routines for a node

› Document/Load these in YANG instance-data-sets
› May later be modified
Instance Data Format

› Simple – Nothing new

› XML & JSON format
› Based on GET reply formats – already defined
› Add metadata (name, revision, description)  
  – Formatted as yang-metadata
› Wrapper <instance-data> container  
  – Holds metadata  
  – Allows multiple root elements
› Follows YANG modules that define the relevant datastore parts  
  – Allow partial data sets
› May contain configuration and/or state data
XML Format

<instance-data xmlns:idai=
"urn:…ietf-yang-instance-data-annotations"
ida:instance-data-set="acme-router-modules"
ida:revision="2108-01-25"
ida:description="Modules an acme-router contains."
<modules xmlns="urn:…yang:ietf-yang-library">
  <module>
    <name>ietf-system</name>
    <revision>2014-08-06</revision>
    <namespace>urn:…ietf-system</namespace>
    <feature>authentication</feature>
    <feature>radius-authentication</feature>
  </module>
</modules>
</instance-data>
JSON Format

- RESTCONF GET reply format

- directed at the datastore:

  - `{+restconf}/data

    Proposal: Allow other root resources, specified in metadata

- No Etags or Timestamps

```json
"instance-data": {
  "@": {
    "instance-data-annotations:instance-data-set": ["acme-router-modules"],
    "instance-data-annotations:revision": "2108-01-25",
    "instance-data-annotations:description": "Defines the modules an acme-router will contains."
  },
  "ietf-yang-library:modules-state": {
    "module": [
      {
        "name": "ietf-system",
        "revision": "2014-08-06",
        "namespace": "urn:...ietf-system",
        "feature": ["authentication", "radius-authentication"]
      }
    ]
  }
}
```
Metadata Yang Module

› module ietf-yang-instance-data-annotations

› Defines metadata annotations to for YANG Instance Data sets
  - instance-data-set : name of the instance data
  - Contact
  - Organization
  - Description
  - Revision       (Semver ?)
Needed - Used

› Already used in multiple implementations
  - Based on similar principles, but slightly different formats

› Needed for YangPush
  - Proposed to be used for documenting on-change notification capabilities

› Needed for Yang-Library
  - Ietf-yang-library contains a lot of server capabilities that are usually set in implementation file, but may change
Out-of-Scope

› Which server capabilities to document
  – Separate drafts e.g. draft-lengyel-netconf-notification-capabilities
› Which yang modules to preload with data
› How instance data is loaded by the server
  – Using an instance-data-file
  – Any other way
› Life-cycle of any documented/preloaded data
  – Is it *protected* from modification, or it *may change* later
  – YangPush may supply notifications about any change