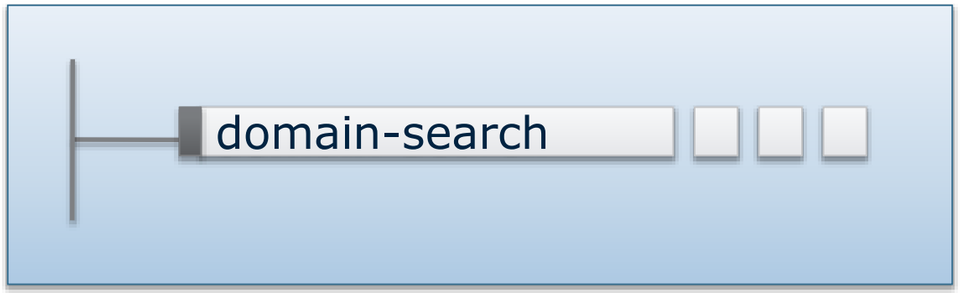
Leaf-list Statement

Holds multiple values of a particular type

Has no children




```
leaf-list domain-search {
  type string;
  ordered-by user;
  description "List of domain names to search";
}
```

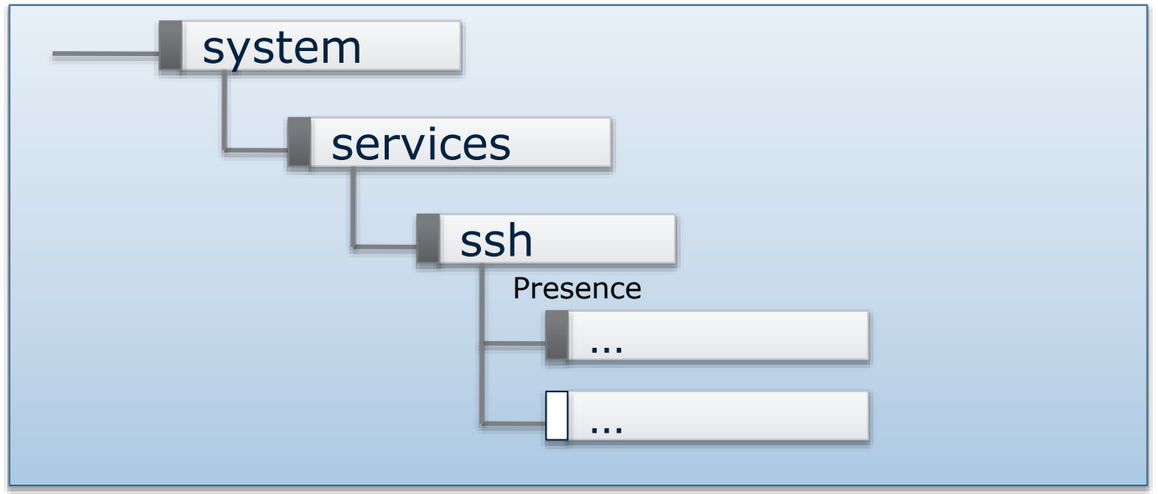
NETCONF operations to insert first, last, before, after

NETCONF XML:

```
<domain-search>high.example.com</domain-search>
<domain-search>low.example.com</domain-search>
```

Container Statement

Groups related leafs and containers



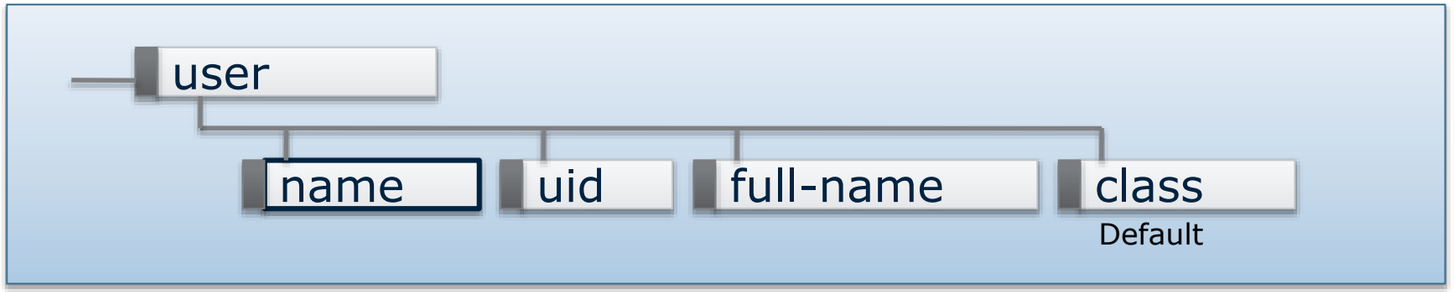

```
container system {
  container services {
    container ssh {
      presence "Enables SSH";
      description "SSH service specific configuration";
      // more leafs, containers and other things here...
    }
  }
}
```

Presence containers explicitly created/deleted by NETCONF client.
They also represent config “themselves”.
“Normal” containers have no meaning, just organization of data.

NETCONF XML:

```
<system>
   <services>
      <ssh>
      </ssh>
   </services>
</system>
```

List Statement




```
list user {
  key name;
  leaf name {
    type string;
  }
  leaf uid {
    type uint32;
  }
  leaf full-name {
    type string;
  }
  leaf class {
    type string;
    default viewer;
  }
}
```

NETCONF XML:

```
<user>
  <name>glocks</name>
  …
</user>
<user>
  <name>snowey</name>
  …
</user>
```

Non-config lists can skip key
Given at create!

NETCONF operations to insert first, last, before, after

Putting things together

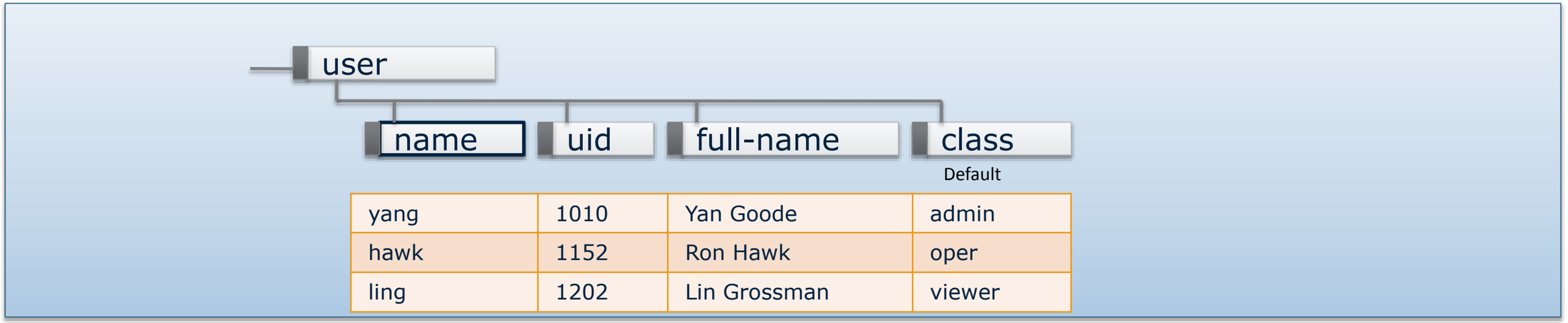
```
module acme-system {
        namespace "http://acme.example.com/system";
        prefix "acme";

        organization "ACME Inc.";
        contact "joe@acme.example.com";
        description
            "The module for entities implementing the

        revision 2007-06-09 {
            description "Initial revision.";
        }
```

```
container system {
    leaf host-name {
        type string;
        description "Hostname for this system";
    }

    leaf-list domain-search {
        type string;
        description "List of domain names to search";
    }

    container login {
        leaf message {
            type string;
            description
                "Message given at start of login session";
        }

        list user {
            key "name";
            leaf name {
                type string;
            }
            leaf full-name {
                type string;
            }
            leaf class {
                type string;
            }
        }
    }
}
```

Attributes for list and leaf-list

max-elements	Max number of elements in list. If max-elements is not specified, there is no upper limit, i.e. “unbounded”
min-elements	Min number of elements in list. If min-elements is not specified, there is no lower limit, i.e. 0
ordered-by	List entries are sorted by “system” or “user”. System means elements are sorted in a natural order (numerically, alphabetically, etc). User means the order the operator entered them in is preserved. “ordered-by user” is meaningful when the order among the elements have significance, e.g. DNS server search order or firewall rules.

Keys



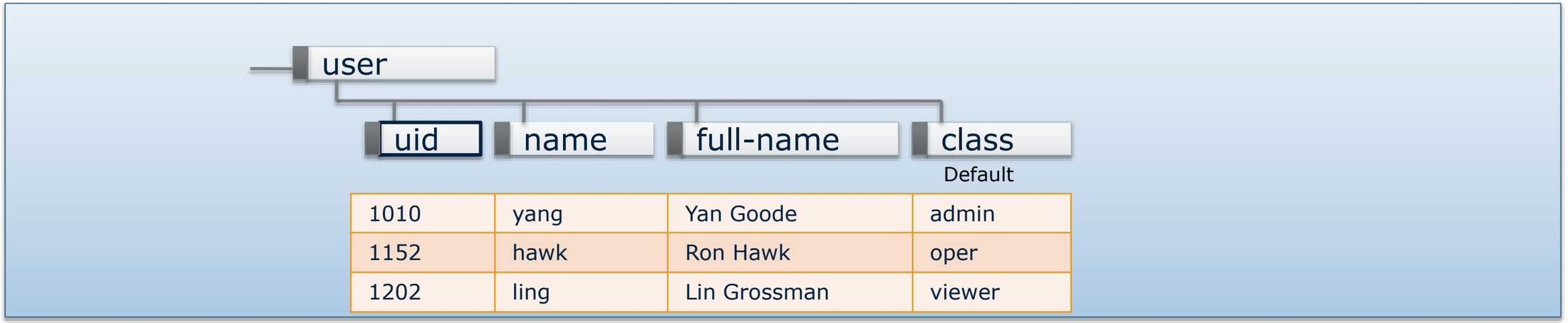
yang	1010	Yan Goode	admin
hawk	1152	Ron Hawk	oper
ling	1202	Lin Grossman	viewer

The key field is used to specify which row we're talking about.

No two rows can have same key value

```
/user[name='yang']/name = yang
/user[name='yang']/uid  = 1010
/user[name='yang']/class = admin

/user[name='ling']/class = viewer
```

Keys

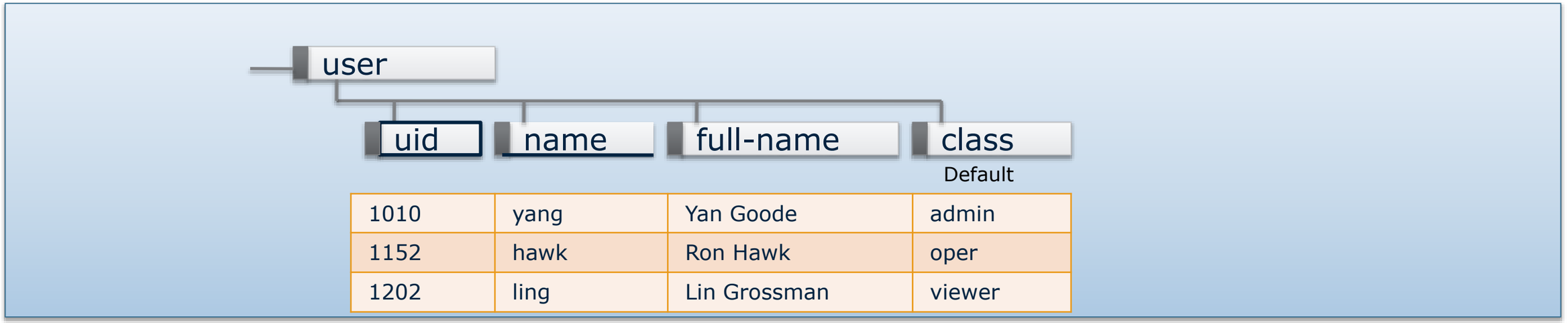


1010	yang	Yan Goode	admin
1152	hawk	Ron Hawk	oper
1202	ling	Lin Grossman	viewer

If we want, we could select the uid to be key instead.

```
/user[uid='1010']/name = yang
/user[uid='1010']/uid  = 1010
/user[uid='1010']/class = admin

/user[uid='1202']/class = viewer
```

Unique Statement



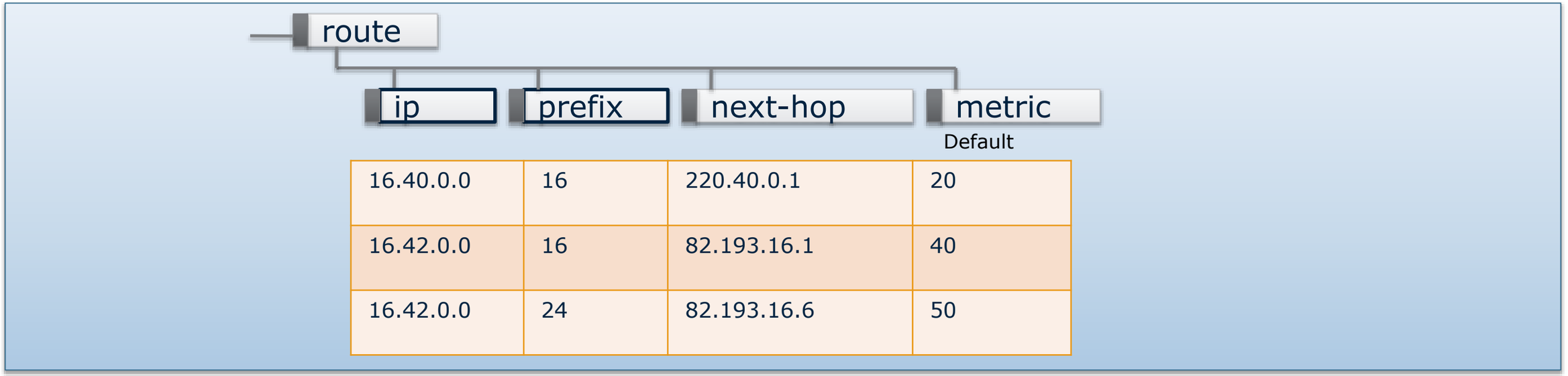
Non- key fields can also be declared unique.

Multiple fields can be declared unique separately or in combination

```
list user {
  key uid;
  unique name;
…
```

No two rows above can have same uid, nor name

Multiple keys



16.40.0.0	16	220.40.0.1	20
16.42.0.0	16	82.193.16.1	40
16.42.0.0	24	82.193.16.6	50

Multiple key fields are needed when a single key field isn't unique.

Key fields must be a unique combination

```
list route {
  key "ip prefix";
  …

/route[ip='16.40.0.0'][prefix='16']/next-hop
 = 220.40.0.1
```

Key order significant

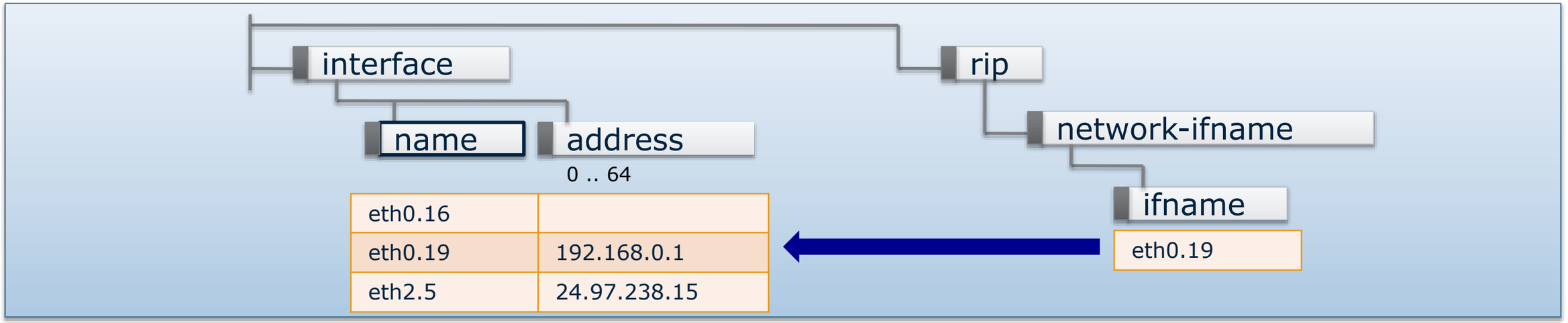
Leafrefs

To make an element reference one of the rows in a list, set the element type to leafref

For lists with multiple keys, the #leafrefs must match #keys in list

- A valid leafref can never be null/empty
  - But the parent leaf can be optional
- A valid leafref can never point to a row that has been deleted or renamed
- System checks validity of leafrefs automatically

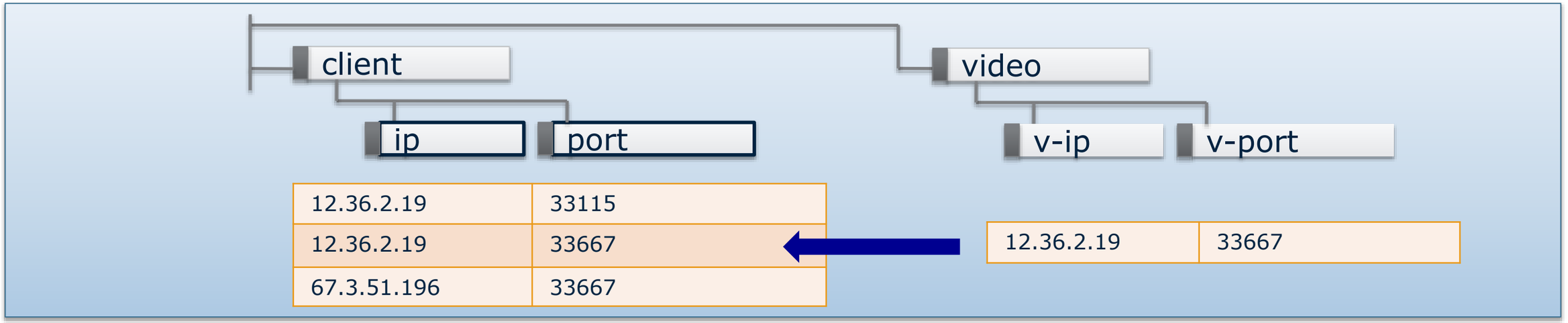
Leafref



Here, the RIP routing subsystem has a list of leafrefs pointing out existing interfaces

```
container rip {
  list network-ifname {
    key ifname;

    leaf ifname {
      type leafref {
        path "/interface/name";
      }
    }
```

Multiple Key Leafref




```
container video {
  leaf v-ip {
    type leafref {
      path "/client/ip";
    }
  }
  leaf v-port {
    type leafref {
      path "/client[ip=current()/../v-ip]/port";
    }
  }
```

Deref() XPATH Operator

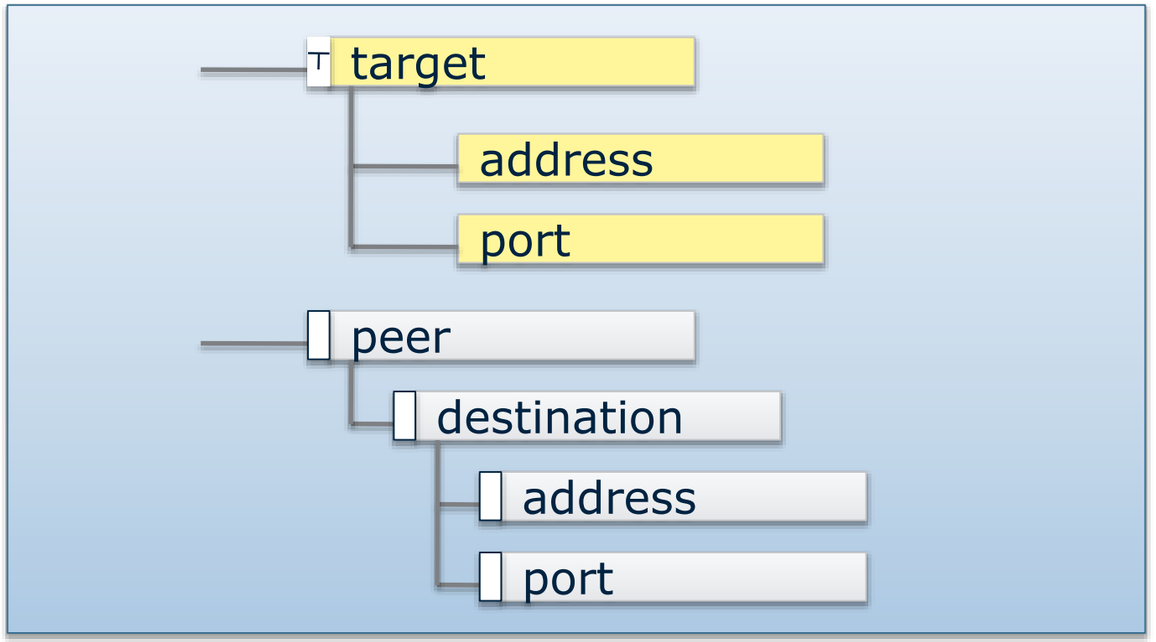
```
container video {
  leaf v-ip {
    type leafref {
      path "/client/ip";
    }
  }
  leaf v-port {
    type leafref {
      path "/client
[ip=current()/../v-ip]/port";
    }
  }
  leaf v-stream {
    type leafref {
      path "/client
[ip=current()/../v-ip]
[port=current()/../v-port]
/stream";
    }
  }
}
```

```
container video-deref {
  leaf v-ip {
    type leafref {
      path "/client/ip";
    }
  }
  leaf v-port {
    type leafref {
      path "deref(../v-ip)
/../port”;
    }
  }
  leaf v-stream {
    type leafref {
      path "deref(../v-port)
/../stream";
    }
  }
}
```

Grouping Statement

Think of macro expansions

```
grouping target {
  leaf address {
    type inet:ip-address;
    description "Target IP";
  }
  leaf port {
    type inet:port-number;
    description
      "Target port number";
  }
}
container peer {
  container destination {
    uses target;
  }
}
```



Groupings can be refined when used

Grouping Statement with Refine

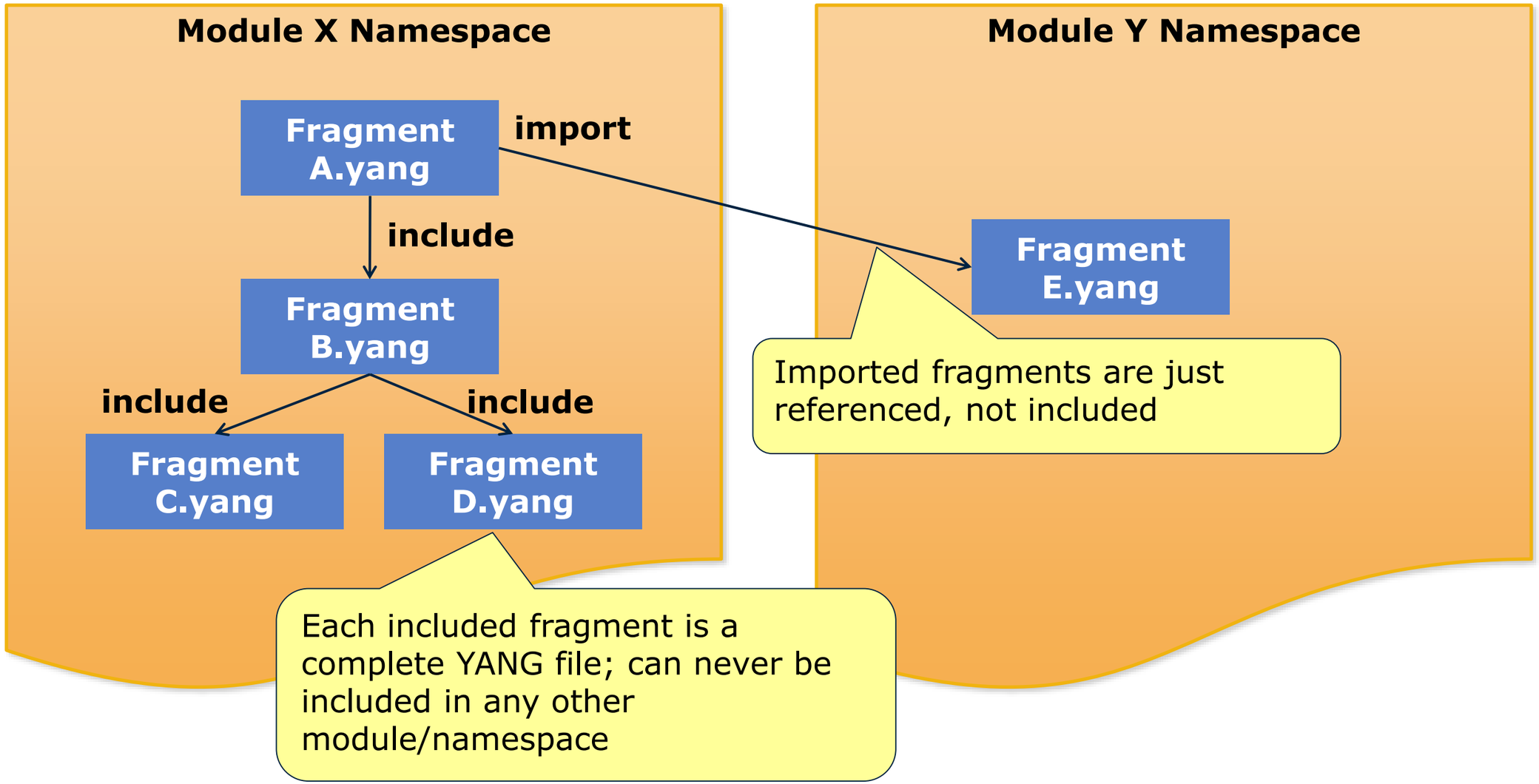
Groupings may be refined when used

```
grouping target {
  leaf address {
    type inet:ip-address;
    description "Target IP";
  }
  leaf port {
    type inet:port-number;
    description
      "Target port number";
  }
}
```

```
container servers {
  container http {
    uses target {
      refine port {
        default 80;
      }
    }
  }
}
```

Import and Include

Imports & Includes



Submodules

```
module acme-module {
  namespace “…”;
  prefix acme;

  import ”ietf-yang-types" {
    prefix yang;
  }
  include "acme-system";
```

Each submodule belongs to one specific main module

```
submodule acme-system {
  belongs-to acme-module {
    prefix acme;
  }

  import ”ietf-yang-types" {
    prefix yang;
  }

  container system {
    …
  }
}
```

Attention: The submodule cannot reference definitions in main module

YANG Types

YANG Base Types

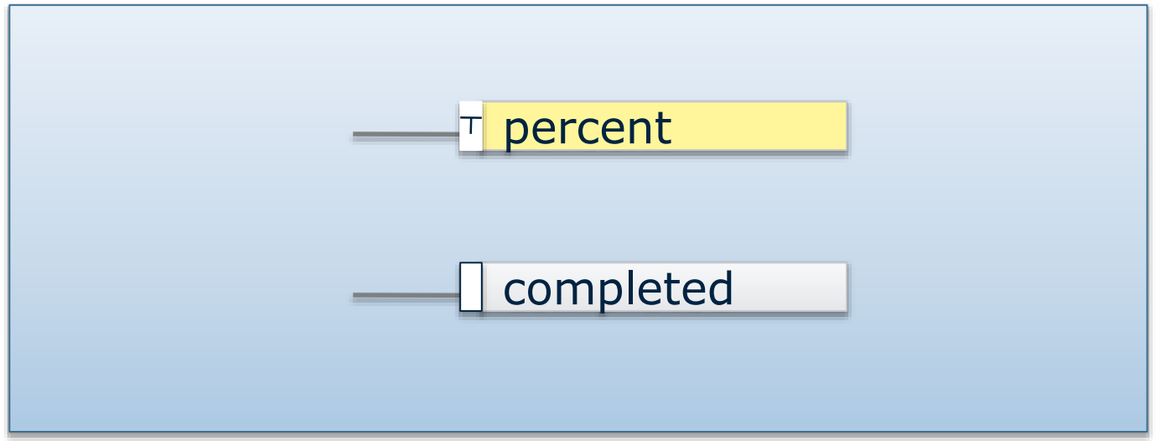
- Most YANG elements have a data type
- Type may be a base type or derived type
  - Derived types may be simple typedefs or groupings (structures)
  - There are 20+ base types to start with
- And more in modules including:
  - ietf-netmod-yang-types, RFC 6021
  - iana-if-type, RFC 7225

Type Name	Meaning
`int8/16/32/64`	Integer
`uint8/16/32/64`	Unsigned integer
`decimal64`	Non-integer
`string`	Unicode string
`enumeration`	Set of alternatives
`boolean`	True or false
`bits`	Boolean array
`binary`	Binary Blob
`leafref`	Reference “pointer”
`identityref`	Unique identity
`empty`	No value, void
	…and more

Typedef Statement

Defines a new simple type

```
typedef percent {
  type uint16 {
    range "0 .. 100";
  }
  description "Percentage";
}

leaf completed {
  type percent;
}
```



Can be scoped

Type Restrictions

Integers

```
typedef my-base-int32-type {
  type int32 {
    range "1..4 | 10..20";
  }
}

typedef derived-int32 {
  type my-base-int32-type {
    range "11..max"; // 11..20
  }
}
```

Strings

```
typedef my-base-str-type {
  type string {
    length "1..255";
  }
}

typedef derived-str {
  type my-base-str-type {
    length "11 | 42..max";
   pattern "[0-9a-fA-F]*";
  }
}
```

Union Statement

A value that represents one of its member types

```
typedef threshold {
  description ”Threshold value in percent";
  type union {
    type uint16 {
      range "0 .. 100";
    }
    type enumeration {
      enum disabled {
        description "No threshold";
      }
    }
  }
}
```

Common YANG Types

- Commonly used YANG types defined in RFC 6021
- Use

```
import "ietf-yang-types" {
  prefix yang;
}
```

to reference these types as e.g.

```
type yang:counter64;
```

counter32/64	ipv4-address
gauge32/64	ipv6-address
object-identifier	ip-prefix
date-and-time	ipv4-prefix
timeticks	ipv6-prefix
timestamp	domain-name
phys-address	uri
ip-version	mac-address
flow-label	bridgeid
port-number	vlanid
ip-address	… and more

ⓘ www.rfc-editor.org/rfc/rfc6021.txt

YANG RPCs and Notifications

RPC Statement

Administrative actions with input and output parameters

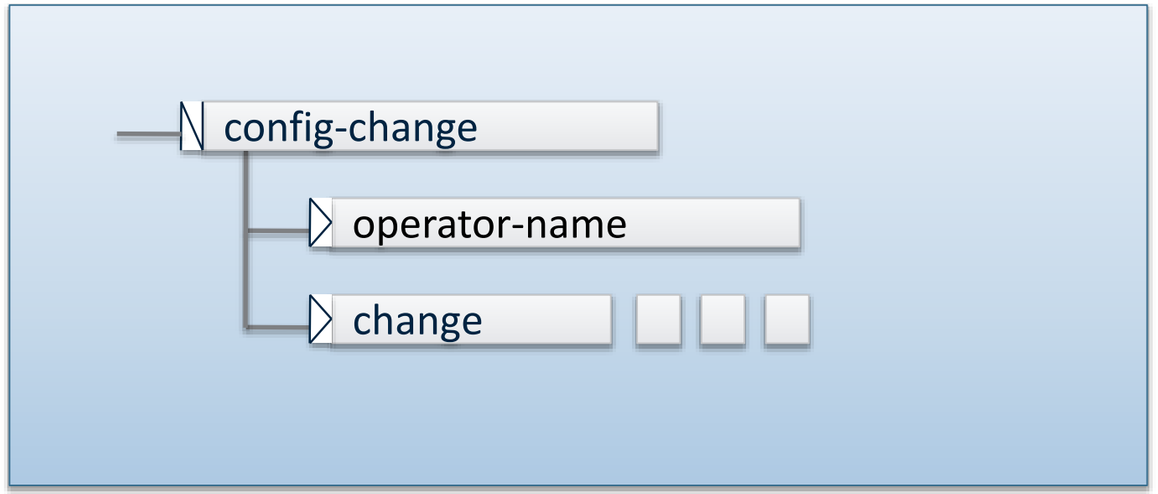
…and side effects

```
rpc activate-software-image {
  input {
    leaf image {
      type binary;
    }
  }
  output {
    leaf status {
      type string;
    }
  }
}
```

Notification Statement

Notification with output parameters

```
notification config-change {
  description
    "The configuration changed";
  leaf operator-name {
    type string;
  }
  leaf-list change {
    type instance-identifier;
  }
}
```



Instance-identifier values

```
<change>/ex:system/ex:services/ex:ssh/ex:port</change>
<change>/ex:system/ex:user[ex:name='fred']/ex:type</change>
<change>/ex:system/ex:server[ex:ip='192.0.2.1'][ex:port='80']</change>
```

Advanced YANG Statements

Must Statement

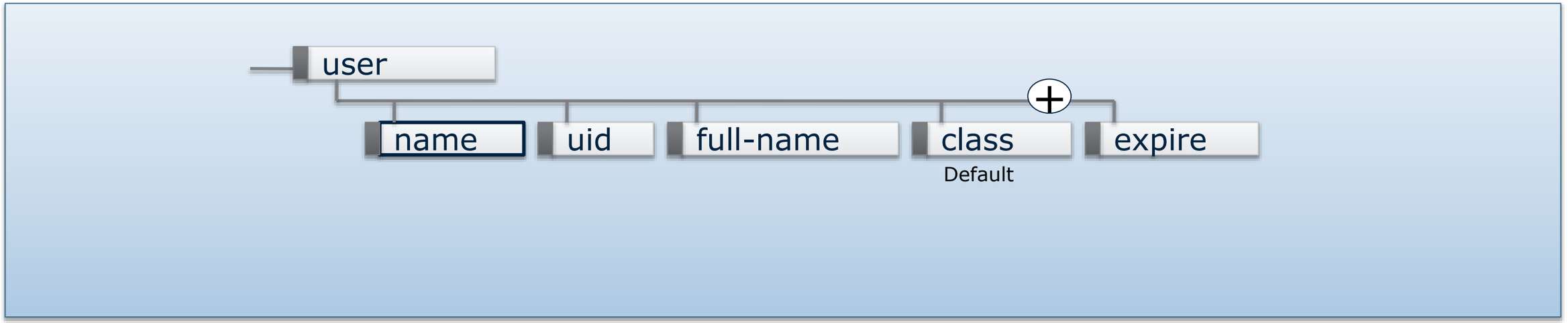
Restricts valid values by Xpath 1.0 expression

```
container timeout {
  leaf access-timeout {
    description "Maximum time without server response";
    units seconds;
    mandatory true;
    type uint32;
  }
  leaf retry-timer {
    description "Period to retry operation";
    units seconds;
    type uint32;
    must ”current() < ../access-timeout" {
      error-app-tag retry-timer-invalid;
      error-message "The retry timer must be "
                  + "less than the access timeout";
    }
  }
}
```

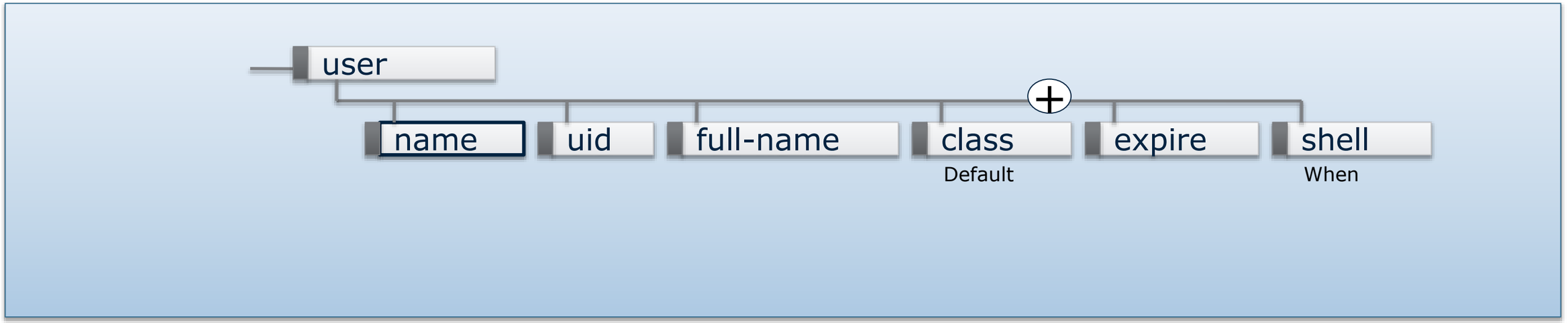
Must Statement

```
leaf max-weight {
  type uint32 {
    range "0..1000";
  }
  default 100;

  must "sum(/sys:sys/interface[enabled = 'true']/weight)
        < current()" {

    error-message "The total weight exceeds the configured
                   max weight";
  }
}
```

Augment Statement



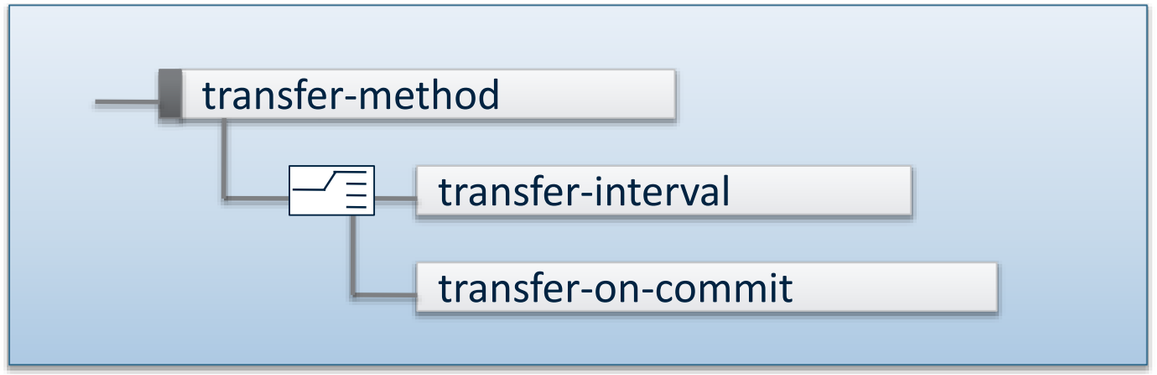

```
augment  /sys:system/sys:user {
  leaf  expire {
    type  yang:date-and-time;
  }
}
```




```
augment /sys:system/sys:user {
  when "sys:class = 'wheel'";
  leaf shell {
    type string;
  }
}
```

Choice Statement

Choice allows one of several alternatives




```
choice transfer-method {
  leaf transfer-interval {
    description "Frequency at which file transfer happens";
    type uint16 {
      range "15 .. 2880";
    }
    units minutes;
  }
  leaf transfer-on-commit {
    description "Transfer after each commit";
    type empty;
  }
}
```

Choice Statement

Each alternative may consist of multiple definitions

- Either as a named or anonymous group

```
choice counters {
  case four-counters {
    leaf threshold {…}
    leaf ignore-count {…}
    leaf ignore-time {…}
    leaf reset-time {…}
  }
  container warning-only {
    …
  }
  default four-counters;
}
```

- Only in schema tree
- Not in the data tree or NETCONF
- Device handles deletion of “other” case when case is created.

Identity Statement

Identities for modeling families of related enumeration constants

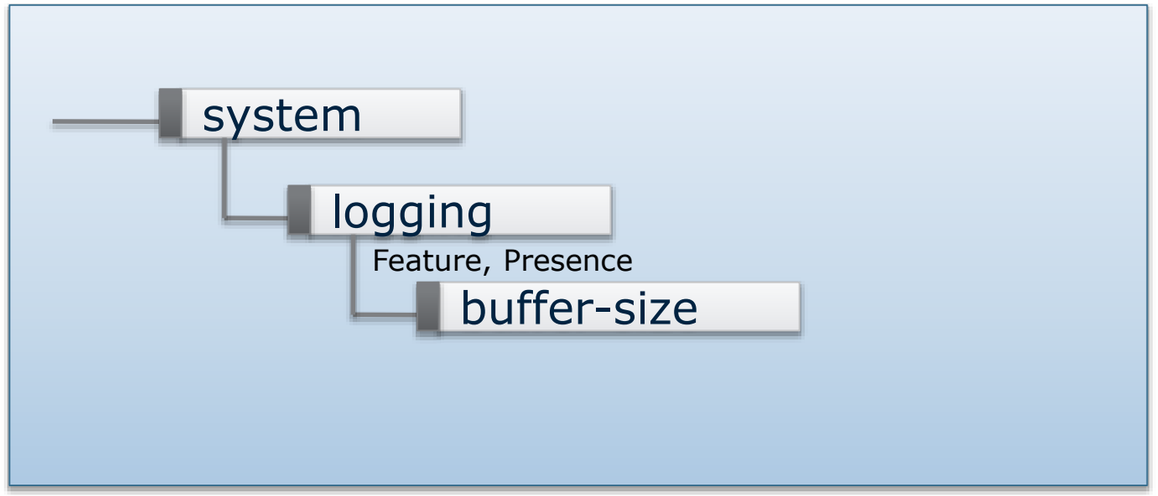
```
module phys-if {
…
  identity ethernet {
    description
  }

  identity eth-1G {
    base ethernet;
  }

  identity eth-10G {
    base ethernet;
  }
```

```
module newer {
…
  identity eth-40G {
    base phys-if:ethernet;
  }
  identity eth-100G {
    base phys-if:ethernet;
  }
…
  leaf eth-type {
    type identityref {
      base ”phys-if:ethernet";
    }
  }
```

Feature Statement

Mark data as conditional

```
feature has-local-disk {
  description
    "System has a local file
     system that can be used
     for storing log files";
}

container system {
  container logging {
    if-feature has-local-disk;
    presence "Logging enabled";
    leaf buffer-size {
      type filesize;
    }
  }
}
```



The features supported by a system are meant to stay relatively constant. Adding a feature is comparable to a hardware upgrade.

Deviations

Systems should conform to standard YANG Modules

- If an implementation can’t, then that can be properly declared
- It’s a secret, but:

```
deviation /base:system/base:user/base:type {
  deviate add {
    default "admin"; // new users are 'admin' by default
  }
}
deviation /base:system/base:name-server {
  deviate replace {
    max-elements 3;
  }
}
```

IETF Activities

IETF Chartered Activities

NETCONF Working Group

- RESTCONF
- NETCONF Call Home
- Advance NETCONF over TLS

Maybe:

- DHCPv6 option for server discovery
- Efficiency extensions
- Schema conformance
- Timed operations

NETMOD Working Group

- YANG 1.1
- SNMP Configuration
- Model for Routing Management
- Guidelines for Authors and Reviewers

Now:

- More protocols (OSPF, BGP, etc)
- Topologies
- Services
- **Come help!**
  - **Join the YANG-Doctors**
  - **Sundays before IETFs**
  - **https://github.com/YangModels**

NETCONF RFC Overview

- RFC 6241 Base NETCONF Protocol
- RFC 6242, 5539 SSH and TLS Transport Mappings
- RFC 5277 Notifications
- RFC 5717 Partial Lock
- RFC 6243 With Defaults
- RFC 6470 Base Notifications
- RFC 6536 NETCONF Access Control Model

ⓘ https://datatracker.ietf.org/wg/netconf/charter/

ⓘ www.rfc-editor.org/rfc/rfcXXXX.txt

YANG RFC Overview

- RFC 6020 YANG Base Specification
- RFC ~~6021~~ 6991 Common YANG Types
- RFC 6087 Guidelines for YANG Authors and Reviewers
- RFC 6110 Mapping and Validating YANG using DSDL
- RFC 6244 NETCONF and YANG Architectural Overview
- RFC 6643 Translation of SMIv2 MIBs to YANG
- RFC 7223 Interface Management
- RFC 7224 IANA Interface Type Module

ⓘhttps://datatracker.ietf.org/wg/netmod/charter/

ⓘhttps://www.ietf.org/iesg/directorate/yang-doctors.html

ⓘhttp://www.yang-central.org/

Thank You! Questions? && <kill-session/>