

# Augment Statements

Scott Mansfield

Ericsson

YANGsters

# Relative vs. Absolute XPath Statements

- Augmentation allows one YANG module to be extended by another YANG module
- Care must be taken how the XPath is written to ensure the augmentation is working properly
- For example, we use presence containers to indicate if a certain capability is enabled
- Depending on how the XPath is written, it is possible to be too permissive in how the presence container is recognized

# Minerals Example

- For example:
- I have a YANG module for my minerals database
- I have an augmentation that adds gem information when the mineral is a gem.

```
module: minerals
  +--rw mineral-db
    +--rw mineral* [mineral-name]
      +--rw mineral-name          string
      +--rw physical-properties?  string
      +--rw hardness?             decimal64
      +--rw ma:gems!
        +--rw ma:gem-name-absolute?  string
        +--rw ma:gem-name-absolute-current? string
        +--rw ma:gem-name-relative?  string
```

# The three examples of When statements

- Absolute
  - when `"/m:mineral-db/m:mineral/ma:gems"`
  - This will match if ANY of the instances have the gems presence container set.
  - Not what is wanted in most cases.
- Absolute with `current()`
  - when `"/m:mineral-db/m:mineral[m:mineral-name=current()/m:mineral-name]/ma:gems"`
  - This will only match if the instance of mineral has the gems presence container set.
  - This is the behavior desired.
- Relative
  - when `"ma:gems"`
  - This works the same as using absolute with `current()`
  - In some cases (this one in particular) it is easier to read than using the `current()` function.