

P802.1Qdy - RSTP/MSTP YANG – Reusability across SDOs
Murugan Balraj (Nokia)
(murugan.balraj@nokia.com)

Introduction

This presentation

- Highlights a problem in the proposed YANG model for RSTP, MSTP as part of P802.1Qdy that would prevent its reuse in other SDOs like BBF.
- Presents a possible solution that would allow the YANG model to be reused by other SDOs, but at the same time preserve the overall structure and functionality of the YANG from IEEE 802.1 perspective.
- Highlights some inconsistencies in the data that could arise and possible solution.

The problem (RSTP)

1 48.6 YANG modules^{6 7 8}

2 *Insert 48.6.26 after 48.6.25 (inserted by IEEE Std 802.1Qdx-2024) as follows::*

3 48.6.26 The ieee802-dot1q-rstp YANG module

```
4 module ieee802-dot1q-rstp {
5   yang-version 1.1;
6   namespace "urn:ieee:std:802.1Q:yang:ieee802-dot1q-rstp";
7   prefix rstp;
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24 augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
25   description
26     "Augment Bridge with RSTP configuration.";
27   reference
28     "13.24, 13.25, and 13.26 of IEEE Std 802.1Q.";
29   container rstp {
30     presence "The presence of this container indicates that RSTP is supported";
31   }
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

The "augments" clause present in the YANG module cannot be overridden and this makes it impossible for other SDOs to use this YANG module since they may not have the notion of Bridge / Component

The problem (MSTP)

2 48.6.27 The ieee802-dot1q-mstp YANG module

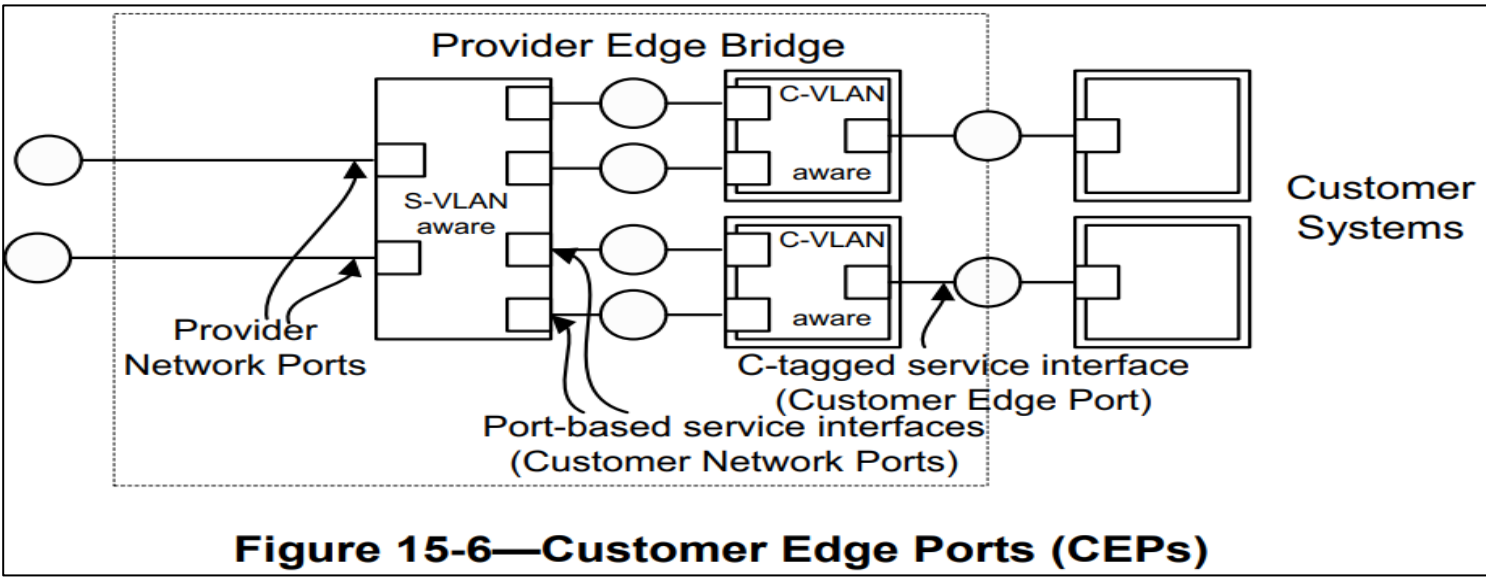
```
3 module ieee802-dot1q-mstp {
4   yang-version 1.1;
5   namespace "urn:ieee:std:802.1Q:yang:ieee802-dot1q-mstp";
6   prefix mstp;

55 }
56
57 augment "/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst" {
58   when "../rstp:rstp";
59
60   description
61     "Augment RSTP-capable Bridge component with MSTP configuration and
62     management.";

59 }
60 }
61 augment "/if:interfaces/if:interface/dot1q:bridge-port" {
62   when "rstp:rstp";
63   description
64     "Augment RSTP Bridge Port with MSTP configuration";
```

The “augments” clause present in the YANG module cannot be overridden and this makes it impossible for other SDOs to use this YANG module since they may not have the notion of Bridge / Component

IEEE 802.1 Provider Edge Bridge

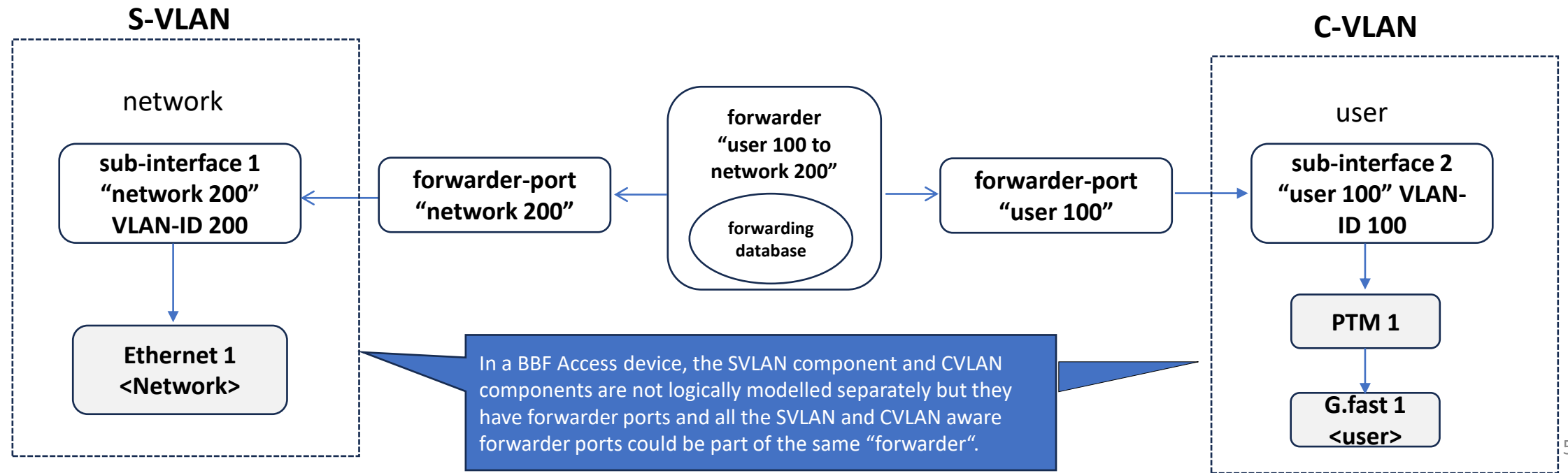


Reference :
 IEEE 802.1Q-2022, Section "15.4 C-tagged service interface", Figure 15-6 – Customer Edge Ports (CEP)

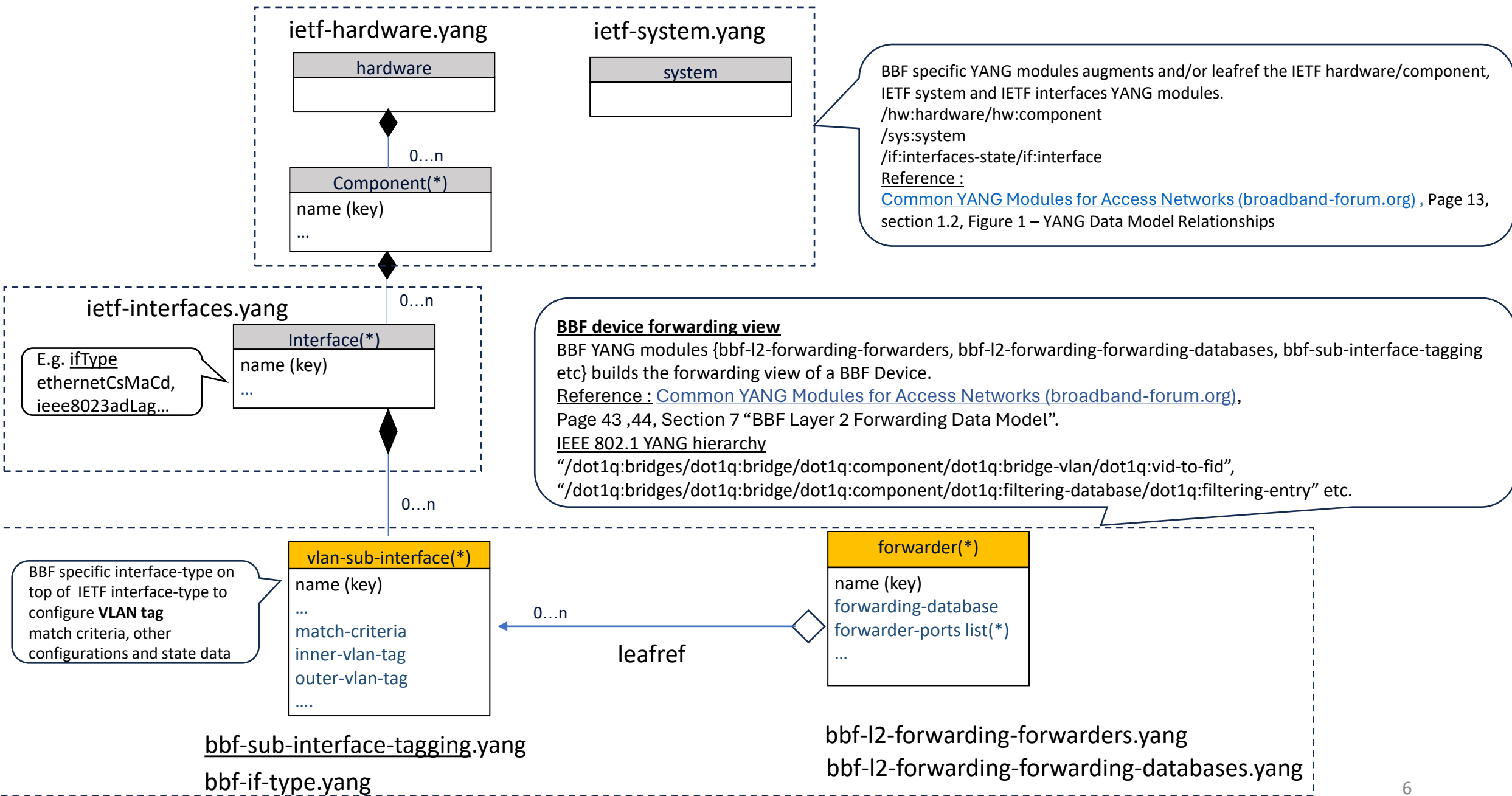
BBF Layer-2 Access device possibly like the IEEE 802.1 Provider Edge Bridge

Figure 15-6—Customer Edge Ports (CEPs)

BBF Layer-2 Access device



BBF Layer-2 Access device YANG view (hardware/component/interface/L2-forwarding)



Proven strategy for reusability across SDOs

The YANG module that other SDOs like BBF would use in their devices has to be free of “augment” clauses and references to Bridge / Component / Bridge Port.

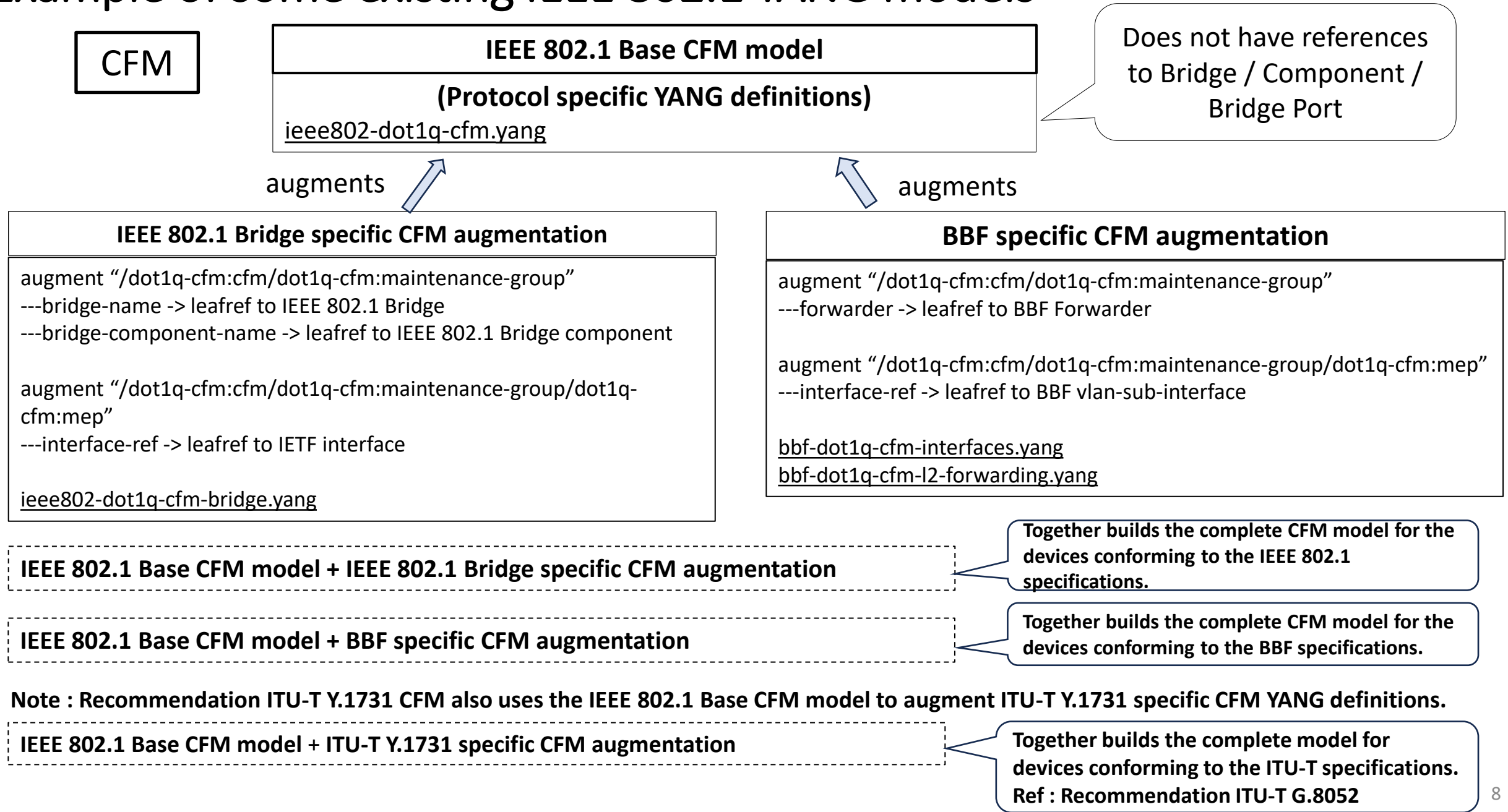
- Split the YANG module into two modules. The Base YANG module would contain the protocol specific YANG definitions and an IEEE 802.1 specific module would augment this base module and contain the “augment” clauses.
- Other SDOs like BBF could then use just the Base YANG module.

Following are some examples of existing IEEE 802.1 YANG models that are reusable

(Path : <https://github.com/YangModels/yang/blob/main/standard/ieee/published/802.1>)

Domain	Common reusable YANG These base modules are reusable in other SDOs	IEEE 802.1 Bridge specific YANG These IEEE 802.1 specific modules augment the base module by adding the “augment” and/or leafref clauses that link the base module to Bridge/Component/Bridge Port
CFM	ieee802-dot1q-cfm.yang	ieee802-dot1q-cfm-bridge.yang
Scheduler	ieee802-dot1q-sched.yang	ieee802-dot1q-sched-bridge.yang
PSFP	ieee802-dot1q-psfp.yang	ieee802-dot1q-psfp-bridge.yang
Stream Filters and Gates	ieee802-dot1q-stream-filters-gates.yang	ieee802-dot1q-stream-filters-gates-bridge.yang
Preemption	ieee802-dot1q-preemption.yang	ieee802-dot1q-preemption-bridge.yang
Congestion-isolation	ieee802-dot1q-congestion-isolation.yang	ieee802-dot1q-congestion-isolation-bridge.yang

Example of some existing IEEE 802.1 YANG models



Example of some existing IEEE 802.1 YANG Models

PSFP

The IEEE 802.1 Bridge specific YANG module has a “uses” statement that augments the “Component” with the “grouping” from the base module

PSFP common reusable YANG definitions

ieee802-dot1q-psfp.yang

Does not have references to Bridge / Component / Bridge Port

The base YANG module uses the “grouping” statement

uses

uses

IEEE 802.1 Bridge specific augmentation

ieee802-dot1q-psfp-bridge.yang

```
augment "/dot1q:bridges/dot1q:bridge/dot1q:component {  
  uses psfp:psfp-parameters  
}
```

Other SDO specific augmentation

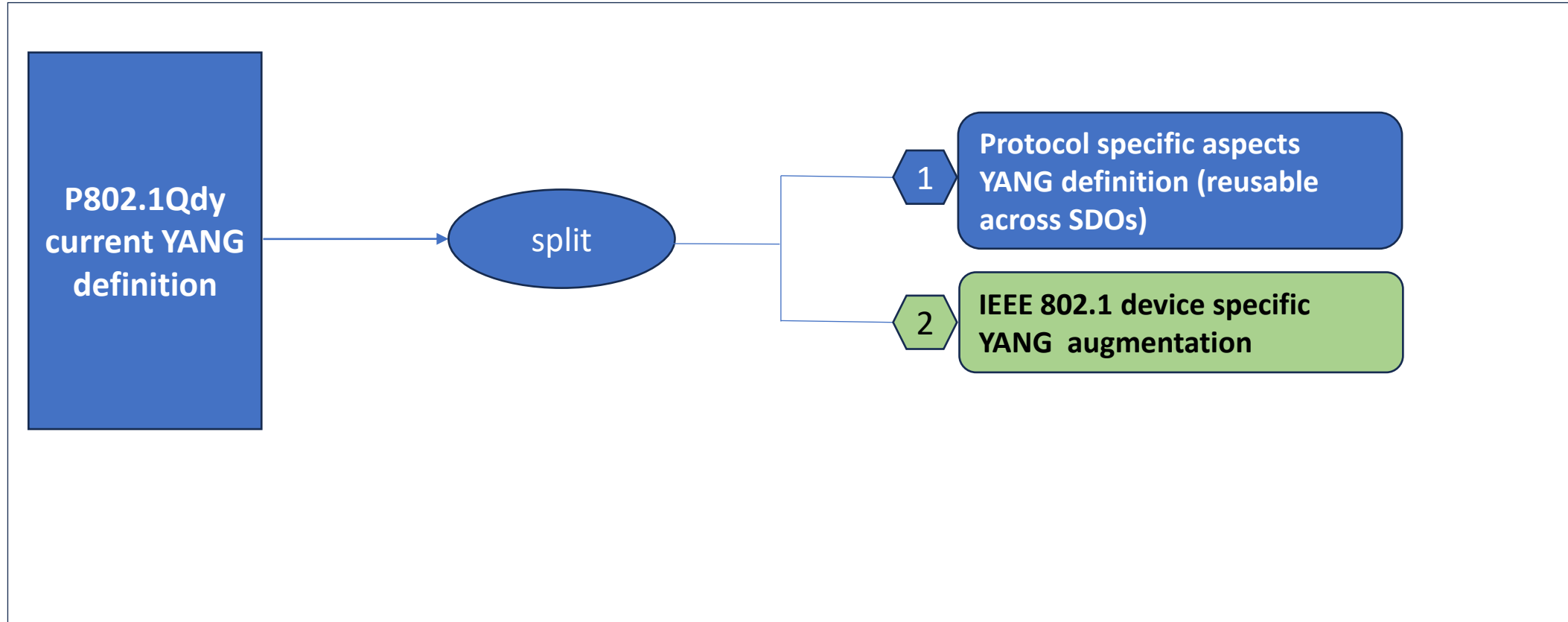
Other SDOs like BBF could also utilize the “uses” statement to include the “grouping”

Other YANG models “**Scheduler, Stream Filters and Gates, Preemption and Congestion-isolation**” also follow the same approach.

The Base module has a “**grouping**” statement and the SDO specific module achieves the augmentation with the “**uses**” statement.

P802.1Qdy decoupling proposal – overall strategy

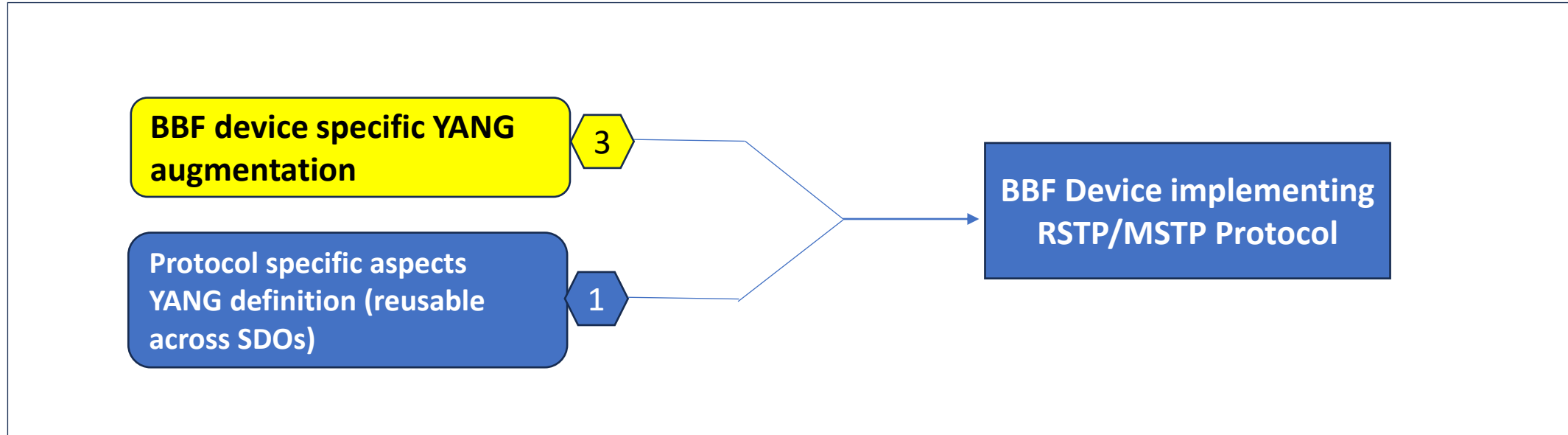
Implementation in a device conforming to IEEE 802.1 specifications



Device conforming to IEEE 802.1 specifications would implement the YANG model of : **1** & **2**

P802.1Qdy decoupling proposal – overall strategy

Implementation in other SDOs



Device conforming to BBF specifications would implement the YANG model of :  & 

P802.1Qdy decoupling proposal – more details

MSTP

IEEE 802.1Qdy current YANG definition

IEEE 802.1 device specific YANG module which uses Base YANG module

Can be reused in other SDOs like BBF

split

IEEE 802.1 device specific augments YANG definition

Base YANG module – MSTP protocol specific YANG definitions as YANG grouping

Per component MSTP augment

Per component MSTP YANG data

YANG module "ieee802-dot1q-mstp-bridge.yang" :

YANG module "ieee802-dot1q-mstp.yang" :

- ❖ augment "/dotq:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst":
 - uses mstp-per-component-protocol-specific-data

uses

- ❖ YANG "Grouping" (mstp-per-component-protocol-specific-data):
 - MSTP YANG configuration/state data which are per component specific.

Per interface MSTP augment

Per interface MSTP YANG data

YANG module "ieee802-dot1q-mstp-bridge.yang" :

YANG module "ieee802-dot1q-mstp.yang" :

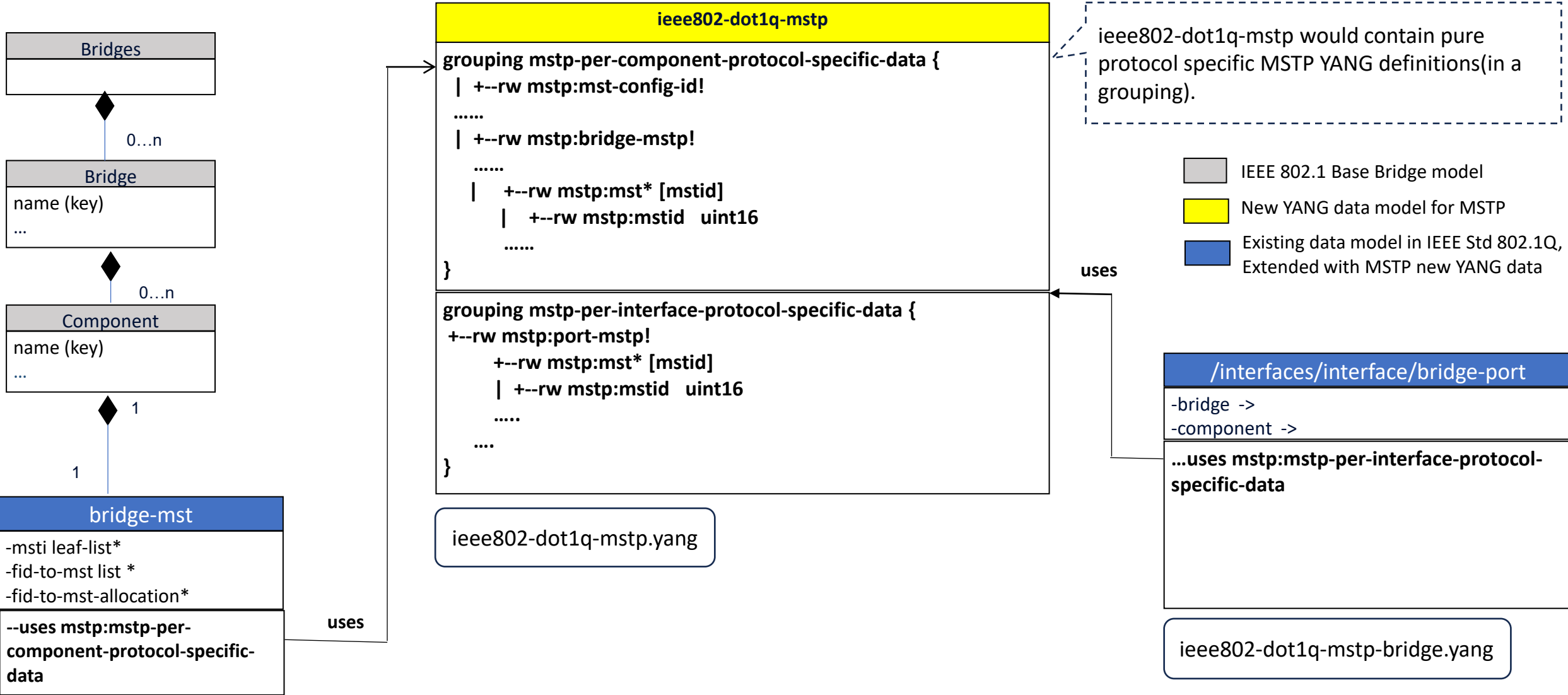
- ❖ augment "/if:interfaces/if:interface/dot1q:bridge-port":
 - uses mstp-per-interface-protocol-specific-data

uses

- ❖ YANG "Grouping" (mstp-per-interface-protocol-specific-data):
 - MSTP YANG configuration/state data which are per interface specific.

P802.1Qdy decoupling proposal – MSTP YANG

IEEE 802.1 UML-like view



Note : Only the **YANG** is restructured considering reusability. The resultant **YANG** tree/functionality is the same as that on page 20 of [802-1Qdy-d2-0.pdf](#)

P802.1Qdy decoupling proposal – MSTP YANG

Resultant YANG tree (IEEE 802.1 Bridge component)

```
module: ieee802-dot1q-mstp-bridge
```

```
augment /dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst:
```

```
  +---rw mst-config-id!  
  | +---rw format-selector?   int32  
  | +---rw configuration-name? string  
  | +---rw revision-level?   uint32  
  | +---ro configuration-digest? binary  
  +---rw bridge-mstp!  
    +---rw max-hops?          int32  
    +---ro ist-internal-root-path-cost? uint32  
    +---rw mst* [mstid]  
      +---rw mstid           uint16  
      +---rw port-id-priority? dot1q-types:priority-type  
      +---ro internal-root-path-cost? uint32  
      +---ro root-port-number? dot1q-types:port-number-type
```

The config / state parameters augmented using ieee802-dot1q-mstp.yang grouping.

P802.1Qdy decoupling proposal – MSTP YANG

Resultant YANG tree (IEEE 802.1 Bridge Port)

```
module: ieee802-dot1q-mstp-bridge
```

```
augment /if:interfaces/if:interface/dot1q:bridge-port:
```

```
  +--rw port-mstp!
```

```
    +--rw mst* [mstid]
```

```
      | +--rw mstid
```

```
      | +--ro msti-port-state?      enumeration
```

```
      | +--ro msti-port-role?      enumeration
```

```
      | +--rw msti-bridge-id-priority?  dot1q-types:priority-type
```

```
      | +--rw msti-internal-port-path-cost? uint32
```

```
      | +--ro msti-regional-root-id?    uint32
```

```
  +--ro msti-internal-root-path-cost?  uint32
```

```
  +--ro msti-designated-bridge-id?    uint32
```

```
  +--ro msti-designated-port-id?     uint32
```

The config / state parameters augmented using ieee802-dot1q-mstp.yang grouping.

P802.1Qdy decoupling proposal – more details

RSTP

IEEE 802.1Qdy current YANG definition

split

IEEE 802.1 device specific YANG module which uses Base YANG module

Can be reused in other SDOs like BBF

IEEE 802.1 device specific augments YANG definition

Per component RSTP augment

YANG module "ieee802-dot1q-rstp-bridge.yang" :

- ❖ augment "/dotq:bridges/dot1q:bridge/dot1q:component" :
 - **uses rstp-per-component-protocol-specific-data**

uses

Per interface RSTP augment

YANG module "ieee802-dot1q-rstp-bridge.yang" :

- ❖ augment "/if:interfaces/if:interface/dot1q:bridge-port" :
 - **uses rstp-per-interface-protocol-specific-data**

uses

Base YANG module – RSTP protocol specific YANG definitions as YANG grouping

Per component RSTP YANG data

YANG module "ieee802-dot1q-rstp.yang" :

- ❖ **YANG "Grouping" (rstp-per-component-protocol-specific-data) :**
 - RSTP YANG configuration/state data which are per component specific.

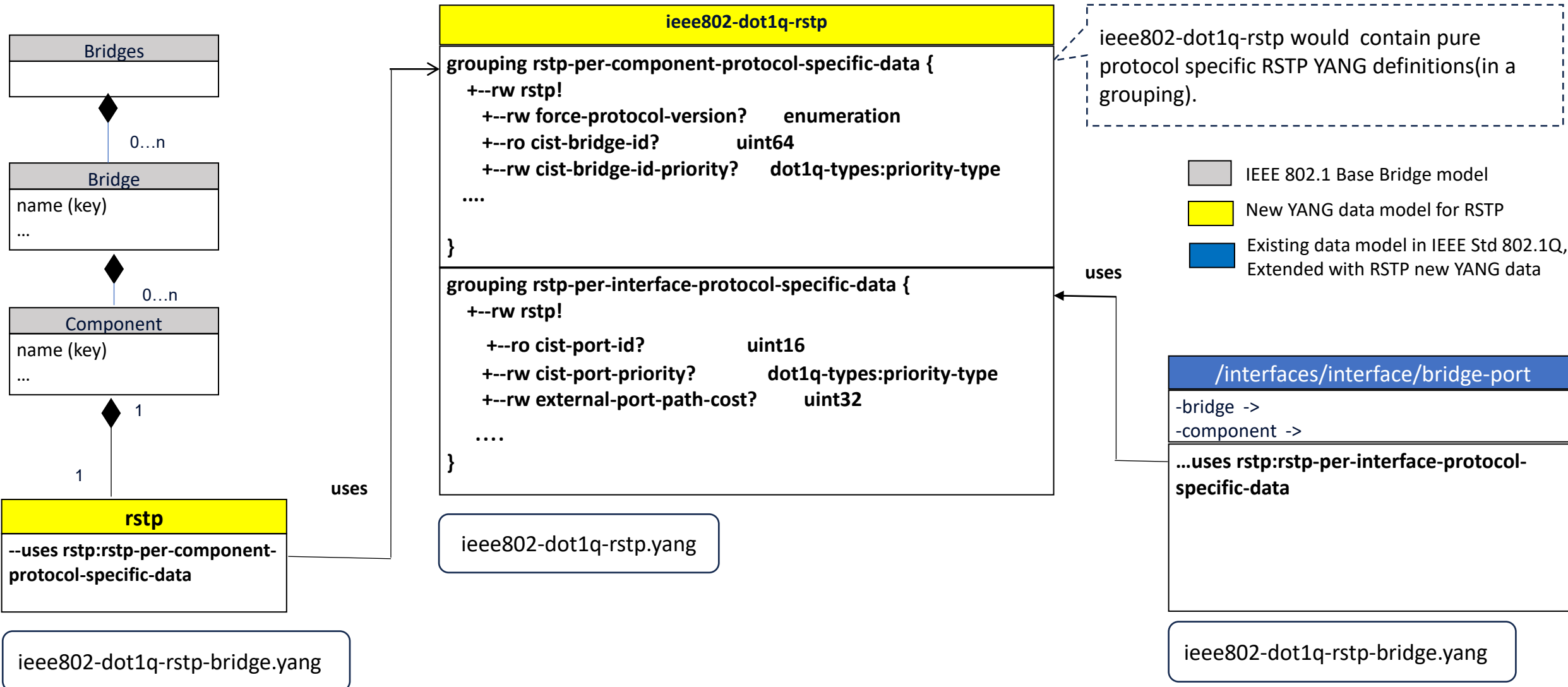
Per interface RSTP YANG data

YANG module "ieee802-dot1q-rstp.yang" :

- ❖ **YANG "Grouping" (rstp-per-interface-protocol-specific-data) :**
 - RSTP YANG configuration/state data which are per interface specific.

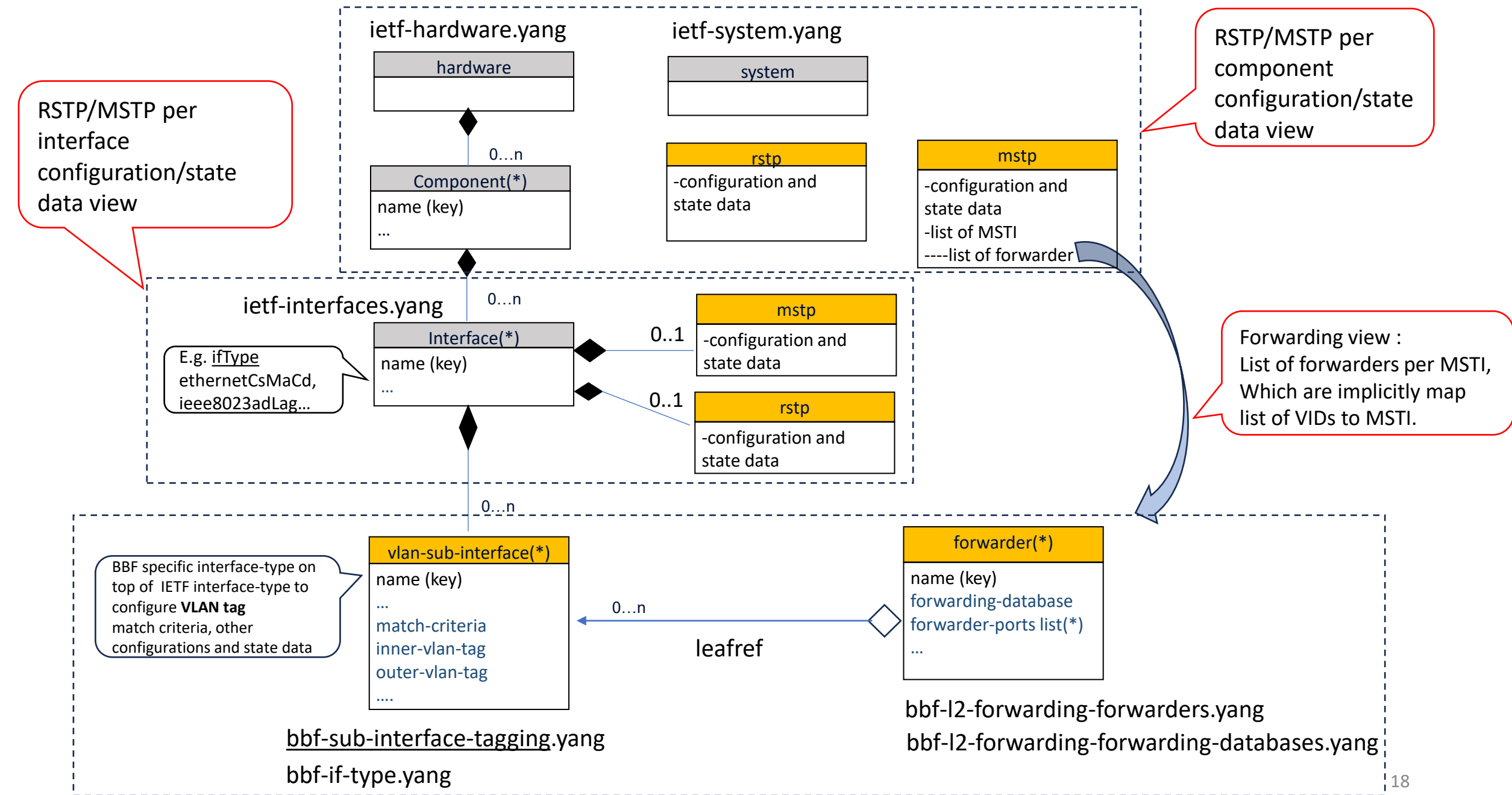
P802.1Qdy decoupling proposal – RSTP YANG

IEEE 802.1 UML-like view



Note : Only the YANG is restructured considering reusability. The resultant YANG tree/functionality is the same as that on page 19 of [802-1Qdy-d2-0.pdf](#).

BBF Layer-2 Access device : RSTP/MSTP Implementation - possible UML-like view

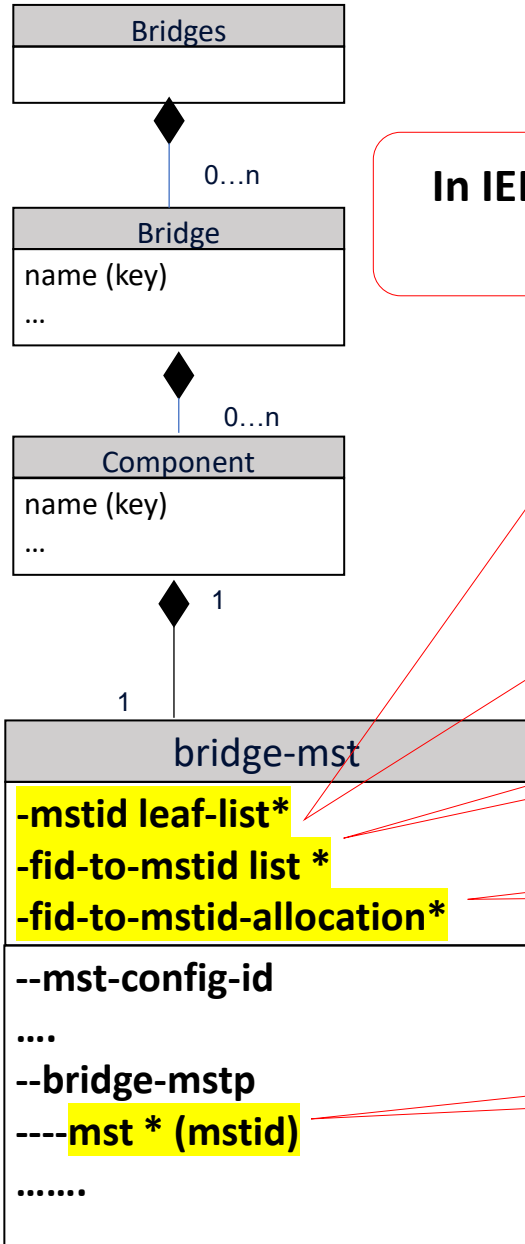


Some observations on configuration/data consistency

- By using the YANG definitions for MSTP specified/proposed across IEEE P802.1Q-2022-REV and P802.1Qdy, there exists possibilities of the resultant configuration being inconsistent.
- Value(s) for one (set of) YANG object(s) could differ from another (set of) YANG object(s) that are otherwise supposed to have the same value(s).
- Even if the values are same/aligned between the different YANG objects, it would still mean that there are redundant objects in the YANG tree , i.e. two or more XPATHS pointing to essentially the same configuration.
- Details are explained in subsequent slides.

Some observations on configuration/data consistency

Possible inconsistencies



In IEEE P802.1Q-2022-REV , a container “bridge-mst” is defined with an object “mstid” which is a leaflist (/bridges/bridge/component/bridge-mst/mstid)

In IEEE P802.1Q-2022-REV, a container “bridge-mst” is defined which has a list “fid-to-mstid” that also has an object mstid /bridges/bridge/component/bridge-mst/fid-to-mstid/mstid

In IEEE P802.1Q-2022-REV, a container “bridge-mst” is defined which has a list “fid-to-mstid-allocation” that also has an object mstid /bridges/bridge/component/bridge-mst/fid-to-mstid-allocation/mstid

In P802.1Qdy, another list for the mst instance is introduced (/bridges/bridge/component/bridge-mst/bridge-mstp/mst)

Since all the above objects refer to the same MST Instance ID, there could be a different set of values in each of these lists

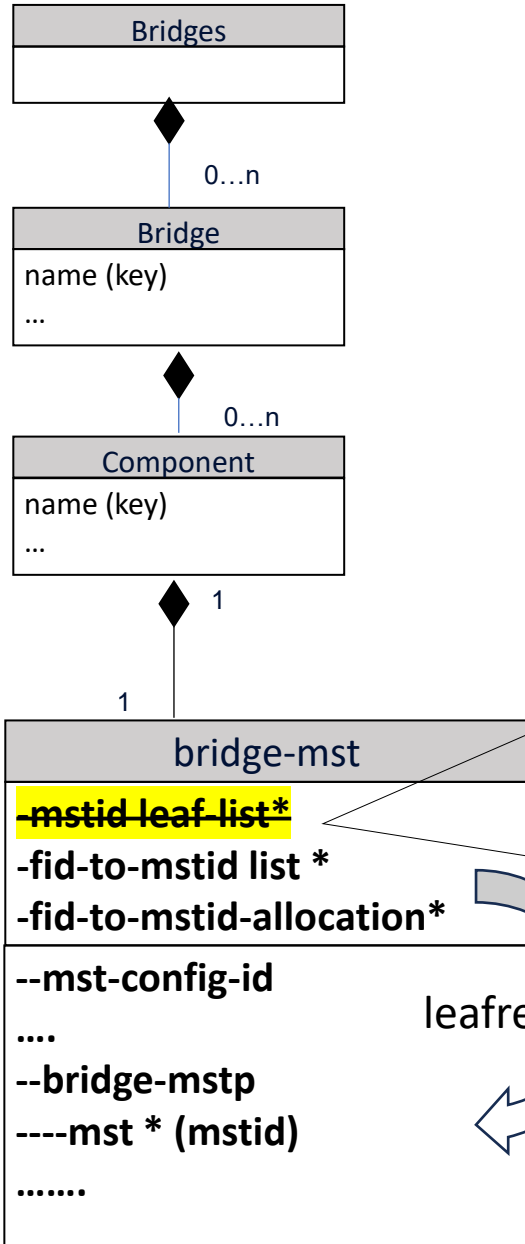
Some observations on configuration/data consistency

Possible inconsistencies

XPATHS	Example configured values (mstid)
<code>(leaf-list) /bridges/bridge/component/bridge-mst/mstid</code>	{100, 101, 102, 103}
<code>(list) /bridges/bridge/component/bridge-mst/fid-to-mstid {fid, mstid}</code>	{1, 100}, {2, 109}, {3, 103}
<code>(list) /bridges/bridge/component/bridge-mst/fid-to-mstid-allocation {fids, mstid}</code>	{500, 110}, {501, 100}, {502, 111}
<code>(list) /bridges/bridge/component/bridge-mst/bridge-mstp/mst {mstid, config/state parameters}</code>	{100, 200}
Highlighted in RED are misconfigurations w.r.t <code>/bridges/bridge/component/bridge-mst/mstid</code> leaf-list configuration	

Some observations on configuration/data consistency

Possible solution



Avoiding inconsistent configuration and/or redundant data may require modifying the YANG objects specified in IEEE Std 802.1Q-2022 under “/bridges/bridge/component/bridge-mst”.

- Would it be acceptable to remove the “mstid” leaf-list from IEEE Std 802.1Q ?
- The list /bridges/bridge/component/bridge-mst/bridge-mstp/mst proposed as part of P802.1Qdy should be sufficient to configure the MST instances. This list would be the main list/configuration for the set of MST instances in each Bridge component.
- If there is a need to refer/specify the “mstid” anywhere else in the YANG tree, a “leafref” type could be used.
- E.g. In the two lists “fid-to-mstid” and “fid-to-mstid-allocation”, the definition of “leaf mstid” could be changed from “type dot1qtypes:mstid-type” to “type leafref”.
- Also, these two lists would need to be moved to the newly proposed IEEE 802.1 specific YANG module “ieee802-dot1q-mstp-bridge” (since these would need to be augmented to Bridge-component).

Some observations on configuration/data consistency

Possible solution

Module ieee802-dot1q-bridge

```
leaf-list mstid {  
  type dot1qtypes:mstid type;  
  description  
    "The list of MSTID values that are currently supported by the  
    Bridge";  
}
```

Remove the "mstid" leaf-list
from IEEE Std 802.1Q

Some observations on configuration/data consistency

Possible solution

Module ieee802-dot1q-bridge

```
list fid-to-mstid {
  key "fid";
  description
    "The FID to MSTID allocation table.";
  reference
    "12.12.2 of IEEE Std 802.1Q";
  leaf fid {
    type uint32;
    description
      "The Filtering Database identifier.";
    reference
      "12.12.2 of IEEE Std 802.1Q";
  }
  leaf mstid {
    type dot1qtypes:mstid-type;
    description
      "The MSTID to which the FID is to be allocated.";
    reference
      "12.12.2 of IEEE Std 802.1Q";
  }
}
```

Change the type of "mstid"
in list "fid-to-mstid"

Module ieee802-dot1q-bridge

```
list fid-to-mstid {
  key "fid";
  description
    "The FID to MSTID allocation table.";
  reference
    "12.12.2 of IEEE Std 802.1Q";
  leaf fid {
    type uint32;
    description
      "The Filtering Database identifier.";
    reference
      "12.12.2 of IEEE Std 802.1Q";
  }
  leaf mstid {
    type leafref { path '../dot1q-mstp:bridge-
      mstp/dot1q-mstp:mst/dot1q-mstp:mstid'; }
    description
      "The MSTID to which the FID is to be allocated.";
    reference
      "12.12.2 of IEEE Std 802.1Q";
  }
}
```


Some observations on Configuration/Data consistency

Possible Solution

Module ieee802-dot1q-bridge

```
list fid-to-mstid-allocation {  
  key "fids";  
  description  
    "The FID to MSTID allocation table";  
  leaf fids {  
    type dot1qtypes:vid-range-type;  
    description  
      "Range of FIDs.";  
    reference  
      "12.12.2 of IEEE Std 802.1Q";  
  }  
  leaf mstid {  
    type dot1qtypes:mstid-type;  
    description  
      "The MSTID to which the FID is allocated.";  
    reference  
      "12.12.2 of IEEE Std 802.1Q";  
  }  
}
```

Change the type of "mstid"
in list "fid-to-mstid-
allocation"

Module ieee802-dot1q-bridge

```
list fid-to-mstid-allocation {  
  key "fids";  
  description  
    "The FID to MSTID allocation table";  
  leaf fids {  
    type dot1qtypes:vid-range-type;  
    description  
      "Range of FIDs.";  
    reference  
      "12.12.2 of IEEE Std 802.1Q";  
  }  
  leaf mstid {  
    type leafref { path '.././dot1q-mstp:bridge-mstp/dot1q-  
      mstp:mst/dot1q-mstp:mstid'; }  
    description  
      "The MSTID to which the FID is allocated.";  
    reference  
      "12.12.2 of IEEE Std 802.1Q";  
  }  
}
```

Example of an existing IEEE 802.1 CFM YANG approach preventing configuration/data inconsistency

Module `ieee802-dot1q-cfm`

```
container cfm {  
  ....  
  list maintenance-domain {  
    key "md-id";  
    .....  
    list maintenance-association {  
      key "ma-id";  
      ...  
      list maintenance-association-mep {  
        key mep-id;  
        .....  
      }  
    }  
  }  
  list maintenance-group {  
    key "maintenance-group-id";  
    ....  
    leaf md-id {  
      type leafref { path '/cfm/maintenance-domain/md-id';  
    }  
  }  
  leaf ma-id {  
    type leafref {  
      path '/cfm'  
        + '/maintenance-domain[md-id = current()/../md-id]/maintenance-association/ma-id';  
    }  
  }  
  list mep {  
    key "mep-id";  
    leaf mep-id {  
      type leafref {  
        path '/cfm/maintenance-domain[md-id = current()/../md-id]'  
          + '/maintenance-association[ma-id = current()/../ma-id]/maintenance-association-mep/mep-id';  
      }  
    }  
  }  
  ....  
}
```

Ref : [ieee802-dot1q-cfm.yang](#)

The CFM list (`/cfm/maintenance-group/mep/mep-id`) for the local MEP configuration refers the main list (`/cfm/maintenance-domain/maintenance-association/maintenance-association-mep/mep-id`) using **leafref** YANG type.

By using YANG leafref type approach, configuration/data inconsistency problem avoided between the main list and the local MEP configuration list.

Similarly,

- the “md-id” leaf configuration in the “maintenance-group” also defined with **leafref** YANG type to the main list “`/cfm/maintenance-domain/md-id`”.
- the “ma-id” leaf configuration in the “maintenance-group” also defined with **leafref** YANG type to the main list “`/cfm/maintenance-domain/maintenance-association/ma-id`”.

leafref

Some observations on configuration/data consistency

Possible solution

- If the lists “fid-to-mstid” and “fid-to-mstid-allocation” are updated, then these need to be placed in the IEEE 802.1 specific YANG module that has been proposed “ieee802-dot1q-mstp-bridge”, since these are linked to Bridge-component that may not be used by other SDOs.
- We could also decide to not disturb the existing objects “leaf-list mstid”, list “fid-to-mstid” and list “fid-to-mstid-allocation” and leave it to equipment vendors or even service providers / operators to ensure that an actual device configuration does not have inconsistencies.
- But it may be preferable to have the consistency checks inherent in the YANG model.

P802.1Qdy – RSTP/MSTP resultant YANG files

YANG modules definition and tree

File name :

“dy-balraj-reusability-across-sdos-yang-0624-v01.zip”

Scope:

- **De-coupling proposal for reusability**

Includes :

ieee802-dot1q-mstp.yang => MSTP Base module
ieee802-dot1q-mstp-bridge.yang => MSTP IEEE 802.1 specific module
ieee802-dot1q-mstp-bridge.tree => MSTP IEEE 802.1 specific module YANG tree
ieee802-dot1q-rstp.yang => RSTP Base module
ieee802-dot1q-rstp-bridge.yang => RSTP IEEE 802.1 specific module
ieee802-dot1q-rstp-bridge.tree => RSTP IEEE 802.1 specific module YANG tree
ieee802-dot1q-bridge.tree => Resultant IEEE 802.1 Bridge Base YANG tree

No change proposed for the IEEE Base YANG file “[ieee802-dot1q-bridge.yang](#)”. The “bridge-mst” container definition considered as such.

YANG modules definition and tree

File name :

“dy-balraj-reusability-across-sdos-yang-without-inconsistencies-0624.v01.zip”

Scope:

- **De-coupling proposal for reusability**
- **Solution for avoiding configuration inconsistency**

Includes :

ieee802-dot1q-mstp.yang => MSTP Base module
ieee802-dot1q-mstp-bridge.yang => MSTP IEEE 802.1 specific module
ieee802-dot1q-mstp-bridge.tree => MSTP IEEE 802.1 specific module YANG tree
ieee802-dot1q-rstp.yang => RSTP Base module
ieee802-dot1q-rstp-bridge.yang => RSTP IEEE 802.1 specific module
ieee802-dot1q-rstp-bridge.tree => RSTP IEEE 802.1 specific module YANG tree
ieee802-dot1q-bridge.yang => IEEE 802.1 Bridge Base model.
Container “bridge-mst” removed from this module and redefined in ieee802-dot1q-mstp-bridge.yang module.
ieee802-dot1q-bridge.tree => Resultant IEEE 802.1 Bridge Base YANG tree

Backup slides

P802.1Qdy - existing YANG view

component (name)		
string	name;	// r-w
...		
rstp		
enum	force-protocol-version;	// r-w
uint64	cist-bridge-id;	// r
priority-type	cist-bridge-id-priority;	// r-w
uint64	cist-root-id;	// r
uint32	external-root-path-cost;	// r
port-number-type	cist-root-port-number;	// r
uint8	max-age;	// r
tv-secs-type	hello-time;	// r
uint8	forward-delay;	// r
uint8	bridge-max-age;	// r-w
uint8	bridge-hello-time;	// r
uint8	bridge-forward-delay;	// r-w
int32	tx-hold-count;	// r-w
int32	migrate-time;	// r
uint32	time-since-topology-change;	// r
counter64	topology-change-count;	// r



These RSTP protocol specific configuration and state data directly augment "/bridges/bridge/component".

A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.

RSTP

bridge-port		
leafref	bridge-name;	// r-w
leafref	component-name;	// r-w
...		
rstp		
enum	cist-port-state;	// r
enum	cist-port-role;	// r
bool	restricted-role;	// r
bool	restricted-tcn;	// r
uint16	cist-port-id;	// r
priority-type	cist-port-priority;	// r
int32	external-port-path-cost;	// r-w
uint32	cist-root-id;	// r
int32	cist-external-path-cost;	// r
uint32	designated-bridge-id;	// r
binary	designated-port-id;	// r
bool	port-protocol-migration-check;	//rw
bool	admin-edge-port;	// r-w
bool	oper-edge-port;	// r
bool	auto-edge-port;	// r-w
bool	auto-isolate-port;	// r
bool	isolate-port;	// r



These RSTP protocol specific interface configuration and state data directly augment "/interface/interface/bridge-port".

A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.

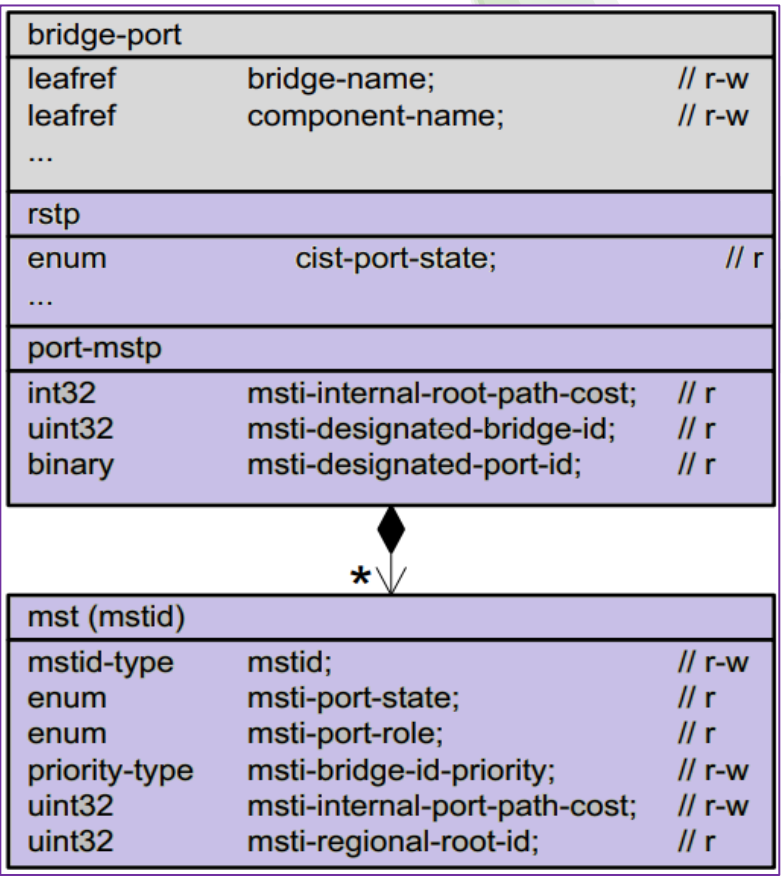
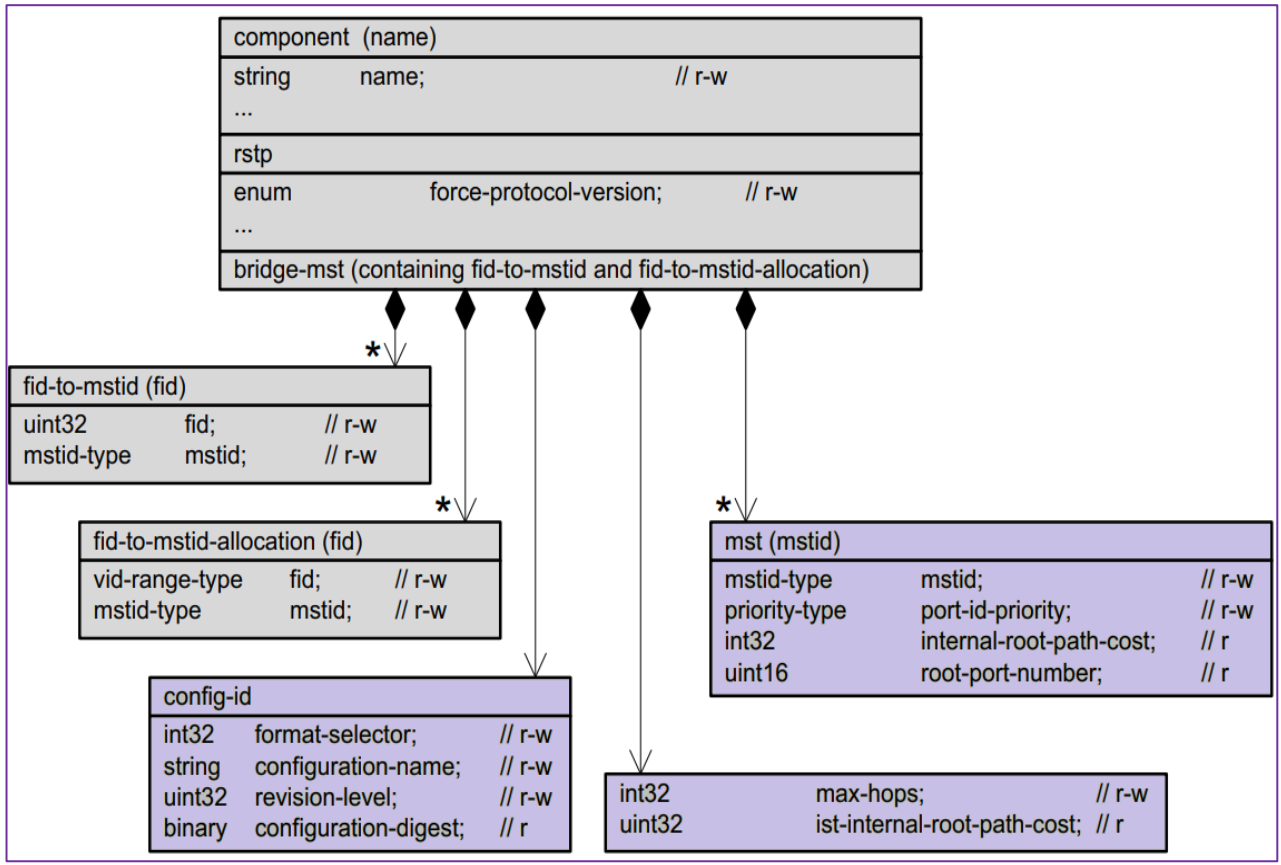
Ref : [802-1Qdy-d2-0.pdf](#),
Page 19

VLAN Bridge component and port nodes + RSTP nodes

Objects added or augmented by this model

P802.1Qdy - existing YANG view

MSTP



Ref : 802-1Qdy-d2-0.pdf, Page 20

VLAN Bridge component and port nodes + RSTP nodes

Objects added or augmented by this model

These MSTP protocol specific configuration and state data directly augment `"/bridges/bridge/component"`.
A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.

These MSTP protocol specific interface configuration and state data directly augment `"/interface/interface/bridge-port"`.
A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.

P802.1Qdy decoupling proposal – more details

ieee802-dot1q-mstp.yang

YANG “Grouping” - Group MSTP Protocol specific configurations and state data.

A grouping for MSTP protocol configuration and state data parameters which are component specific.

A grouping for MSTP protocol configuration and state data parameters which are interface specific.

ieee802-dot1q-rstp.yang

YANG “Grouping” - Group RSTP Protocol specific configurations and state data.

A grouping for RSTP protocol configuration and state data parameters which are component specific.

A grouping for RSTP protocol configuration and state data parameters which are interface specific.

ieee802-dot1q-rstp-bridge.yang

Augments IEEE 802.1 Bridge Component/Bridge-port with RSTP YANG definitions from ieee802-dot1q-rstp.yang.

ieee802-dot1q-mstp-bridge.yang

Augments IEEE 802.1 Bridge Component/Bridge-port with MSTP YANG definitions from ieee802-dot1q-mstp.yang.

P802.1Qdy decoupling proposal – MSTP YANG

Base YANG module view

```
module ieee802-dot1q-mstp {  
  .....  
  .....  
  grouping mstp-per-component-protocol-specific-data {  
  
    description  
      "Grouping for MSTP configuration and state data";  
    .....  
    .....  
  
  }// End of grouping mstp-per-component-protocol-specific-data  
  
  grouping mstp-per-interface-protocol-specific-data {  
  
    description  
      "Grouping for MSTP configuration and state data augment under the interface";  
    .....  
    .....  
  
  }// End of grouping mstp-per-interface-protocol-specific-data  
  
}
```

P802.1Qdy decoupling proposal – MSTP YANG

Augmenting Bridge component/Bridge port with MSTP YANG objects

```
module ieee802-dot1q-mstp-bridge {  
  
  augment "/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst" {  
    when "../dot1q-rstp:rstp";  
    description  
      "Augment RSTP-capable Bridge component with MSTP configuration and  
      management.";  
    reference  
      "13.24, 13.25, and 13.26 of IEEE Std 802.1Q.";  
    uses mstp:mstp-per-component-protocol-specific-data;  
  }  
  
  augment "/if:interfaces/if:interface/dot1q:bridge-port" {  
    when "dot1q-rstp:rstp";  
    description  
      "Augment RSTP Bridge Port with MSTP configuration";  
    reference  
      "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";  
    uses mstp:mstp-per-interface-protocol-specific-data;  
  }  
}
```

P802.1Qdy decoupling proposal – RSTP YANG

Resultant YANG tree (IEEE 802.1 Bridge component)

```
module: ieee802-dot1q-rstp-bridge
```

```
augment /dot1q:bridges/dot1q:bridge/dot1q:component:  
  +--rw rstp!  
    +--rw force-protocol-version?  enumeration  
    +--ro cist-bridge-id?          uint64  
    +--rw cist-bridge-id-priority? dot1q-types:priority-type  
    +--ro cist-root-id?           uint64  
    +--ro external-root-path-cost? uint32  
    +--ro cist-root-port-number?  dot1q-types:port-number-type  
    +--ro max-age?                uint8  
    +--ro hello-time?             rt-types:timer-value-seconds16  
    +--ro forward-delay?         uint8  
    +--rw bridge-max-age?        uint8  
    +--ro bridge-hello-time?     uint8  
    +--rw bridge-forward-delay?  uint8  
    +--rw tx-hold-count?         int32  
    +--ro migrate-time?          int32  
    +--ro time-since-topology-change? uint32  
    +--ro topology-change-count? yang:counter64
```

The config / state parameters augmented from ieee802-dot1q-rstp.yang grouping.

P802.1Qdy decoupling proposal – RSTP YANG

Resultant YANG tree (IEEE 802.1 Bridge Port)

module: ieee802-dot1q-rstp-bridge

augment /if:interfaces/if:interface/dot1q:bridge-port:

+--rw rstp!

+--ro cist-port-state?	enumeration
+--ro cist-port-role?	enumeration
+--ro restricted-role?	boolean
+--ro restricted-tcn?	boolean
+--ro cist-port-id?	uint16
+--rw cist-port-priority?	dot1q-types:priority-type
+--rw external-port-path-cost?	uint32
+--ro cist-root-id?	uint32
+--ro cist-external-path-cost?	uint32
+--ro designated-bridge-id?	uint32
+--ro designated-port-id?	binary
+--rw port-protocol-migration-check?	boolean
+--rw admin-edge-port?	boolean
+--ro oper-edge-port?	boolean
+--rw auto-edge-port?	boolean
+--rw auto-isolate-port?	boolean
+--ro isolate-port?	boolean

The config / state parameters augmented from ieee802-dot1q-rstp.yang grouping.

P802.1Qdy decoupling proposal – RSTP YANG

Base YANG module view

```
module ieee802-dot1q-rstp {  
  ....  
  ....  
  grouping rstp-per-component-protocol-specific-data {  
  
    description  
      "Grouping for RSTP configuration and state data";  
    .....  
    .....  
  
  } // End of grouping rstp-per-component-protocol-specific-data  
  
  grouping rstp-per-interface-protocol-specific-data {  
  
    description  
      "Grouping for RSTP configuration and state data augment under the interface";  
    .....  
    .....  
  
  } // End of grouping rstp-per-interface-protocol-specific-data  
  
}
```

P802.1Qdy decoupling proposal – RSTP YANG

Augmenting Bridge component/Bridge port with RSTP YANG objects

```
module ieee802-dot1q-rstp-bridge {  
  
  augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {  
    description  
      "Augment RSTP configuration and state data.";  
    reference  
      "13.24, 13.25, and 13.26 of IEEE Std 802.1Q.";  
    uses rstp:rstp-per-component-protocol-specific-data;  
  }  
  
  augment "/if:interfaces/if:interface/dot1q:bridge-port" {  
    description  
      "Augment Bridge Port with RSTP configuration";  
    reference  
      "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";  
    uses rstp:rstp-per-interface-protocol-specific-data;  
  }  
}
```