



railML.org

News from Common

Thomas Nygreen, common coordinator, railML.org

Agenda

A: railML 3.x:

1. Organizational units remodelled
2. Handling changes between minor versions – stability or flexibility?
3. Type designators

B: railML 2.5:

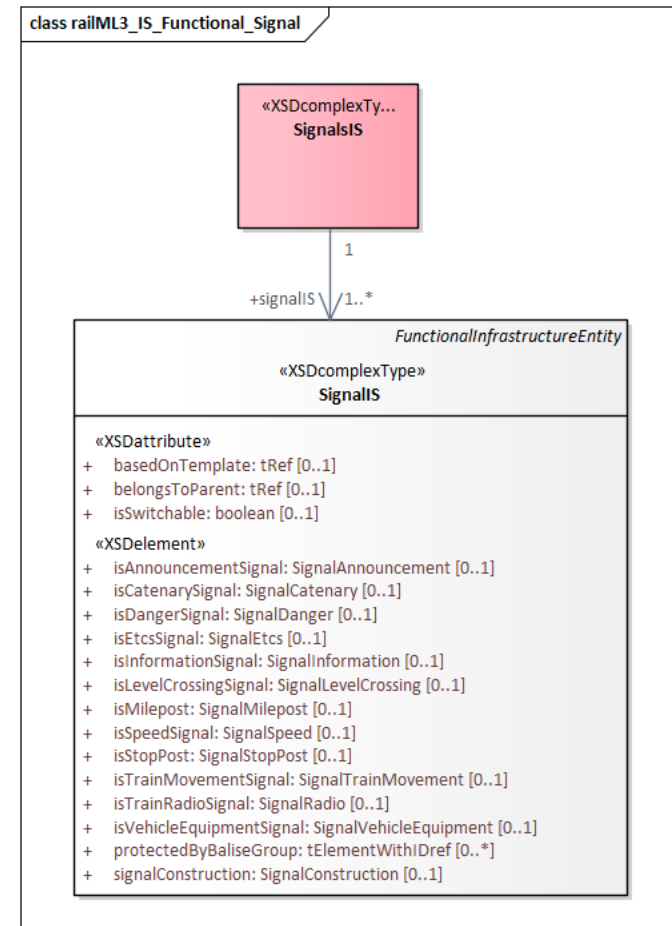
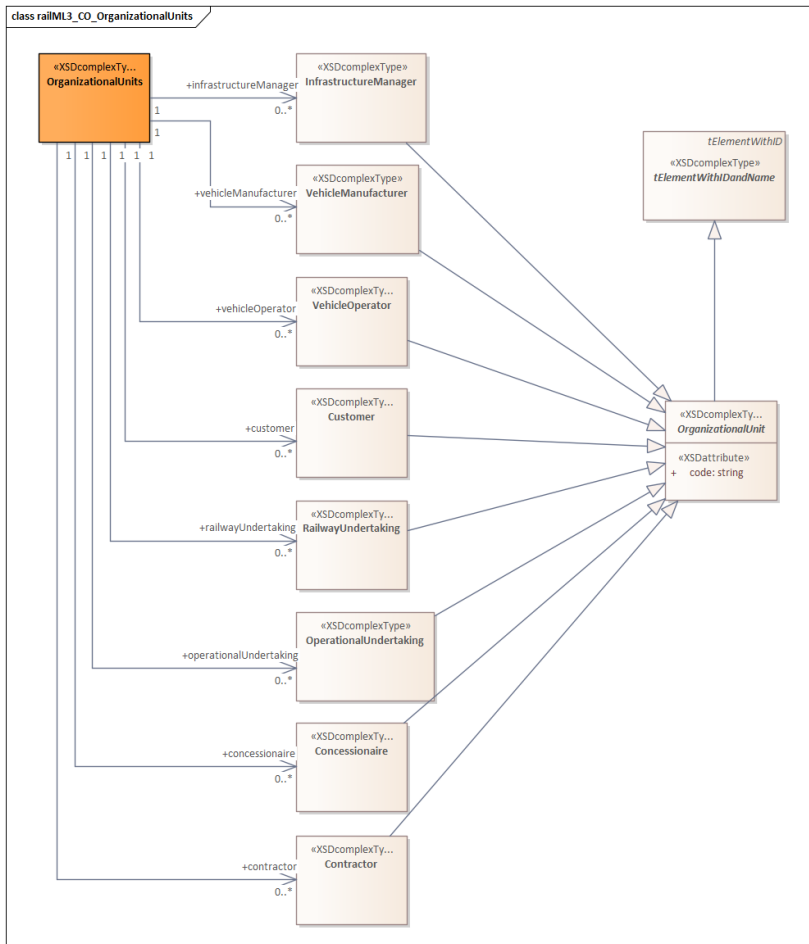
4. <designator> for organizational units

C: Common:

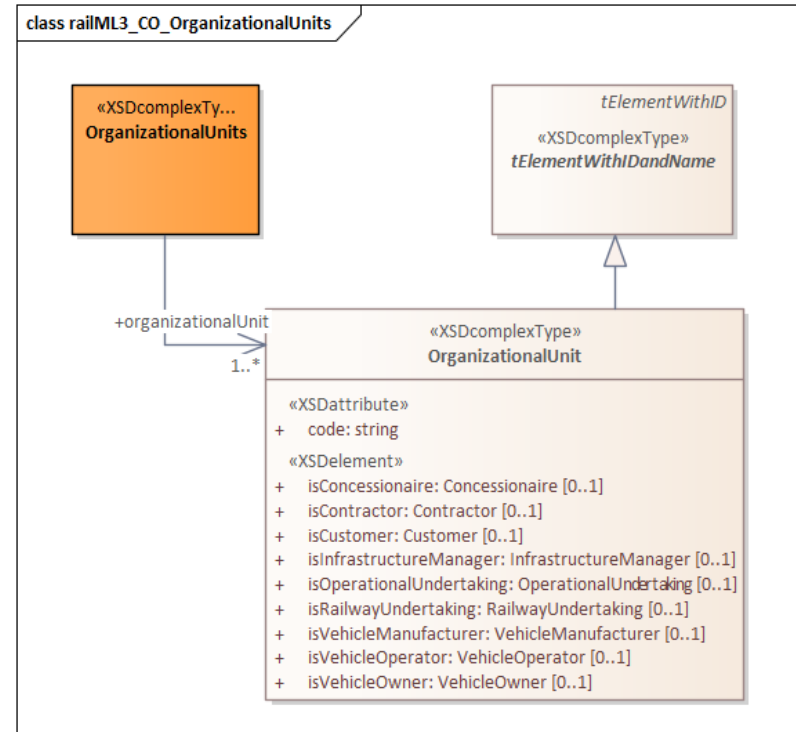
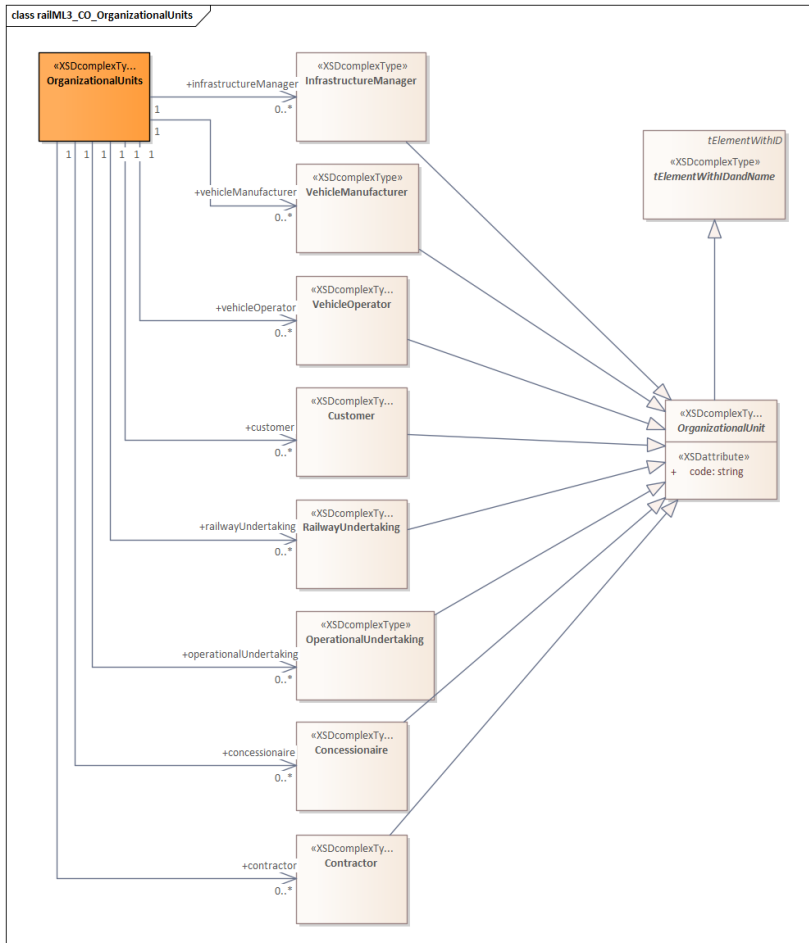
5. New organizational unit for vehicle owners
6. Metadata for revision management
7. Metadata for project descriptions

Part A: railML 3.x

1: Organizational units remodelled



1: Organizational units remodelled



2: Handling changes between minor versions in railML 3.x

- Compatibility between minor versions is an important principle in railML 2.x
 - New elements and attributes can be introduced in minor versions
 - Elements and attributes can be deprecated in minor versions
 - No elements or attributes are removed in minor versions
 - Documentation, interpretations, semantic constraints and code list elements can be introduced or improved continuously
- Compatibility restricts flexibility for improvements
 - Cannot correct errors (e.g. rename <brigde> to <bridge>)
 - Cannot make required elements or attributes optional
 - Limits improved modelling of existing concepts
 - Free to add new concepts

2: Handling changes – options

Options proposed by the coordinators:

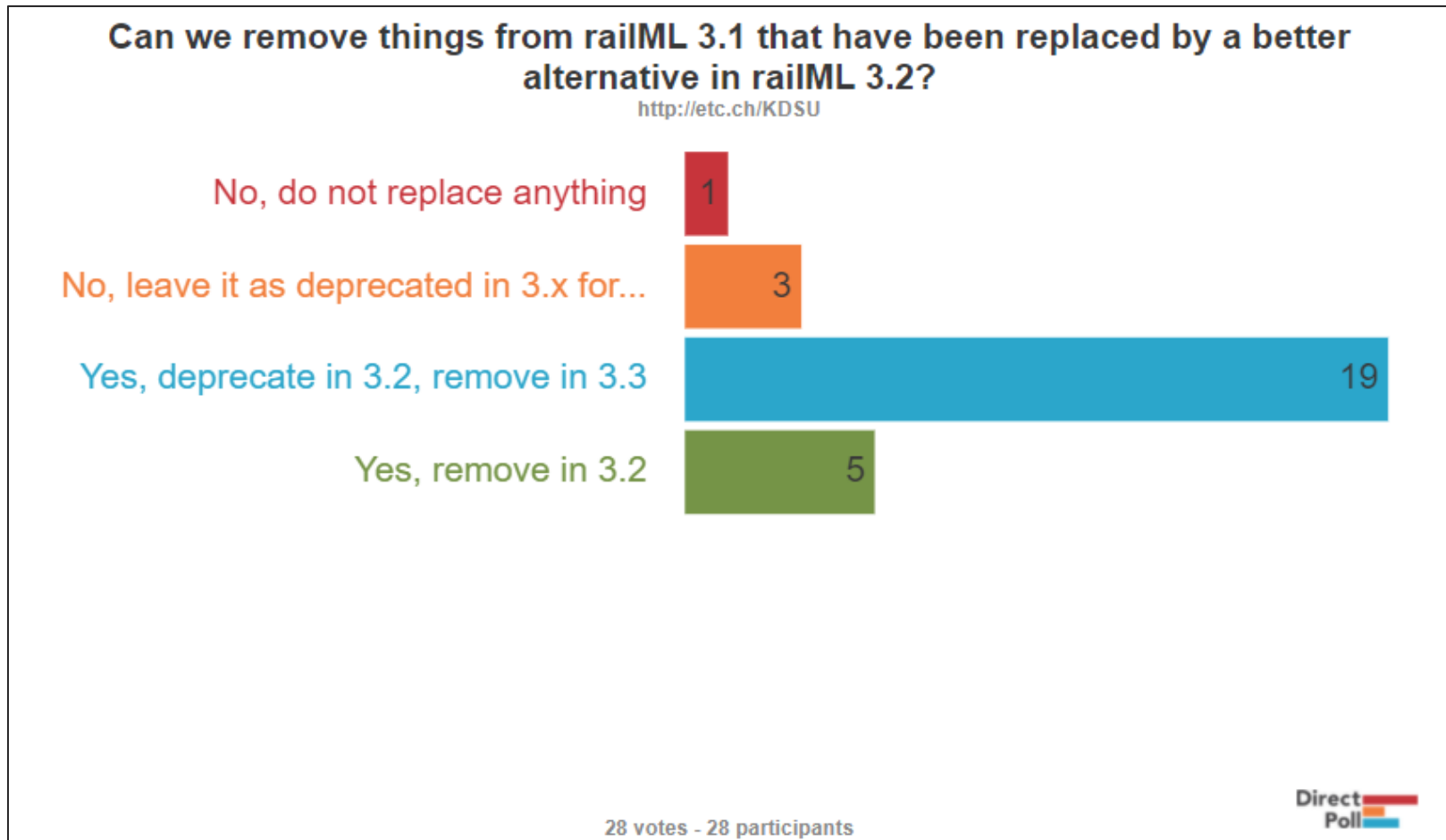
1. Follow the path of railML 2.x in railML 3.x: downward compatibility will be guaranteed. Changes may lead to elements or attributes being deprecated, but not removed.
2. Changes may lead to elements or attributes being deprecated. Elements and attributes that are deprecated in one minor version will be removed in the following minor version. Compatibility is only guaranteed between one minor version and the next.
3. Changes may lead to elements or attributes being removed in a new minor version, without first being deprecated. No compatibility guaranteed.



- Poll: go to <http://etc.ch/KDSU> (or use link in chat)



2: Handling changes – poll results



3: Type designators

- `<designator>` was first introduced as a more flexible way to provide the `@code` for an `<ocp>`
- In railML 3.x `<designator>` is applied on most elements, as an external key, using a specified register
- Question: should `<designator>` also be allowed for identifying categories within element types?
 - *Example: In railML 2.x there is an attribute `@ruleCode` of `<signal>`*
 - *In railML 3.x, can we use `<designator>` for this with the signal book as register?*
 - *Or should there be a separate element, equivalent to `@ruleCode`?*
- Poll: go to <http://etc.ch/KDSU> (same link as before)



3: Type designators – Example

A stopPost of the hypothetical type 7d in railML 2.x:

```
<stopPost id="sig_94" name="Stop post 200m"
pos="..." code="Object-12345" ruleCode="7d"/>
```

The same in railML 3.x?

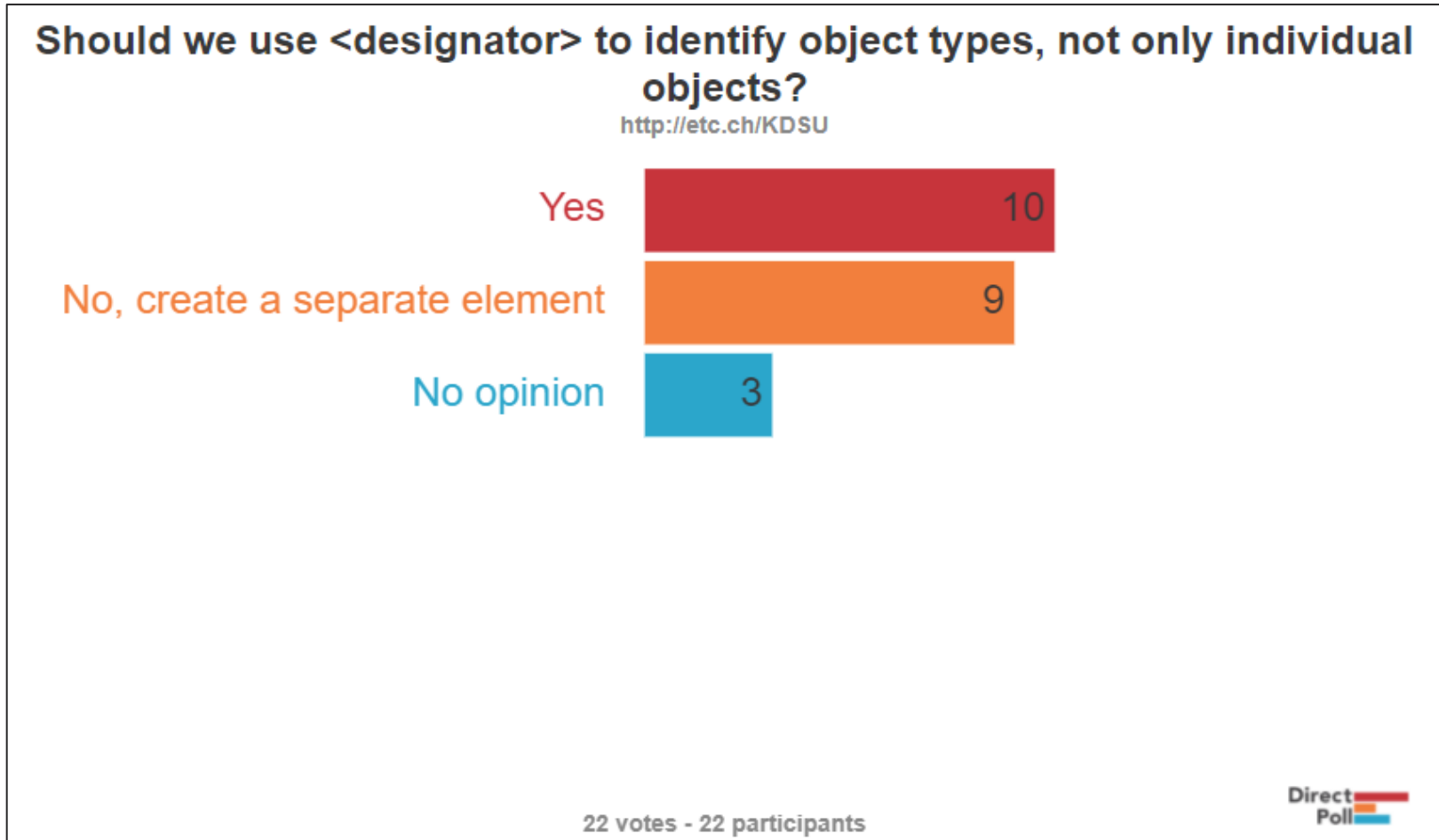
```
<signalIS id="sig03">
  <name name="Stop post 200m"/>
  <spotLocation ... />
  <designator
    register="AssetDB"
    entry="Object-12345"/>
  <designator
    register="RuleBook"
    entry="7d"/>
  <isStopPost
    refersToStoppingPlace="stp27"/>
</signalIS>
```

Extend use?

```
<signalIS id="sig03">
  <name name="Stop post 200m"/>
  <spotLocation ... />
  <designator
    register="AssetDB"
    entry="Object-12345"/>
  <typeDesignator
    register="RuleBook"
    entry="7d"/>
  <isStopPost
    refersToStoppingPlace="stp27"/>
</signalIS>
```

New element?

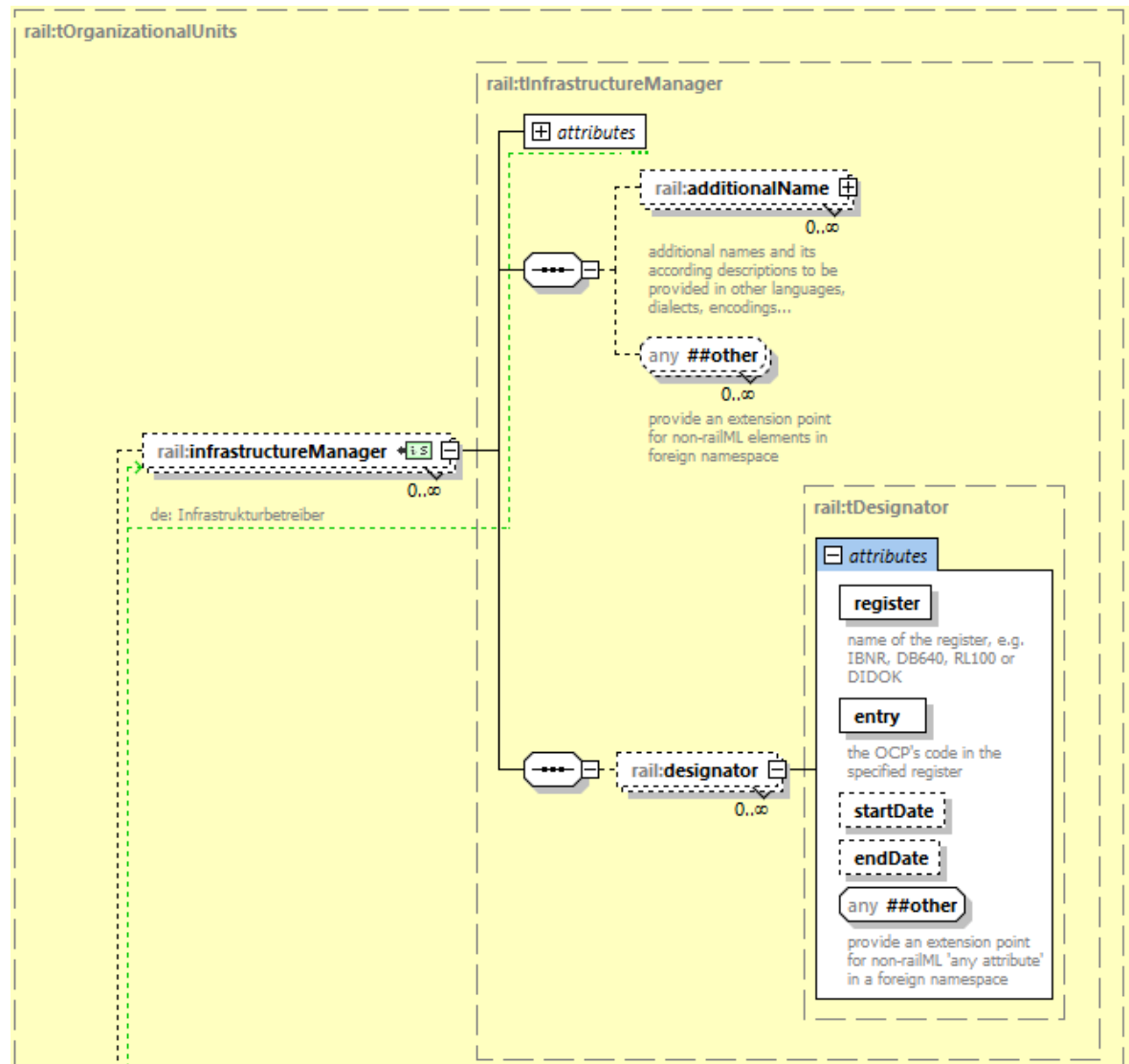
3: Type designators – poll results



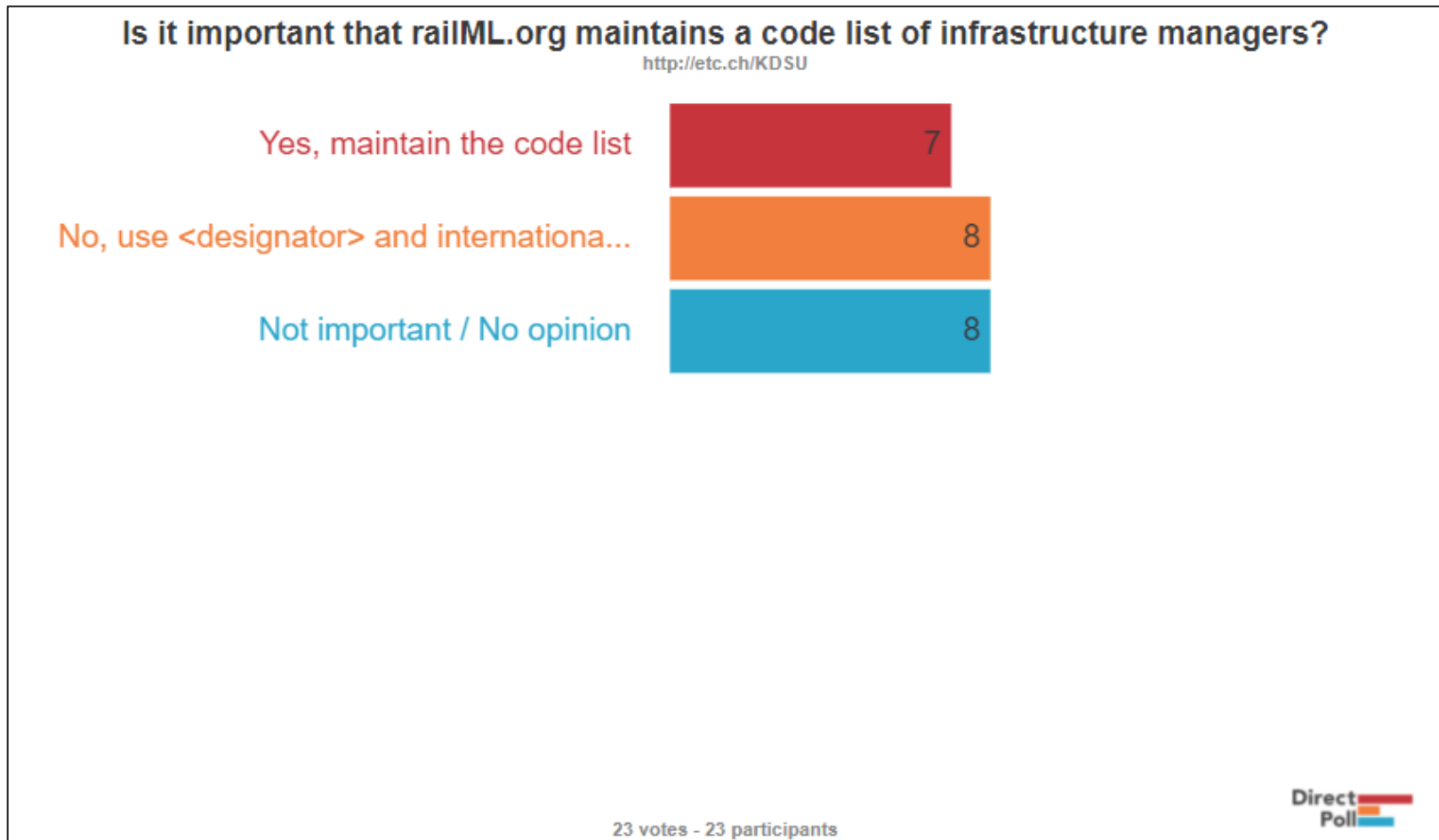
Part B: railML 2.5

4: <designator> for organizational units

- **Question:** Keep infrastructureManagers.xml or not?
- **Poll:** <http://etc.ch/KDSU> (same link as before)



3: <designator> for organizational units – poll results



Part C: Common

5: New organizational unit for vehicle owners

- There are already roles for vehicle manufacturers and vehicle operators, but no role for companies only owning vehicles that are provided to operators.
- The door is open for suggestions on further improvement of organizational unit definitions and categorization.
- <https://www.railml.org/forum/index.php?t=msg&th=706&goto=2325>

Requests for extended metadata

6. The ETCS Track Net workgroup has proposed four new metadata fields for revision management in railML 3.2:
 - status: draft/verified/released
 - fileDocumentId: e.g. "ID123467890-LineX-Station1/Station2-XYZrailways"
 - fileVersion: e.g. 00.00, ..., 99.99
 - md5checksum: checksum for file contents
 - Forum: <https://www.railml.org/forum/index.php?t=msg&th=726&start=0>
7. Norway has asked for project descriptions in railML 2.5
 - The <project> element describes the purpose of the infrastructure objects which are part of a model. One <infrastructure> can have multiple projects (<project>).
 - For more details see document “railML2.4nor Infrastructure Documentation” (<https://www.jernbanedirektoratet.no/railML>), version 1.3, 03.07.2020, point 4.13.
 - Trac: <https://trac.railml.org/ticket/390>



Thank you for your attention!



Jernbane-
direktoratet

www.railml.org

Thomas Nygreen, Norwegian Railway Directorate
Common coordinator, railML.org



coord@common.railml.org



+47 47 47 50 88