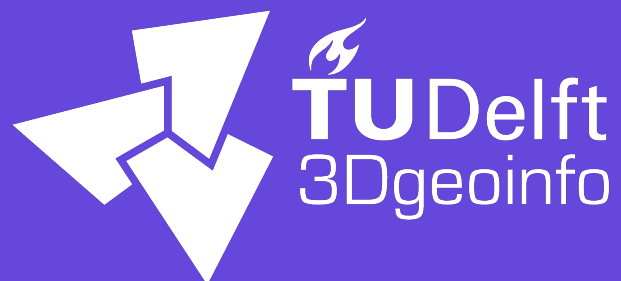


Stop copying your files: versioning of 3D city models

**Stelios Vitalis, Anna Labetski, Ken Arroyo Ohori,
Hugo Ledoux, Jantien Stoter**

Amsterdam
03-12-2019



3D city models are static

Most datasets:

- are released as “read-only”
- remain intact during their life-time
- they become outdated and then replaced with a newer model

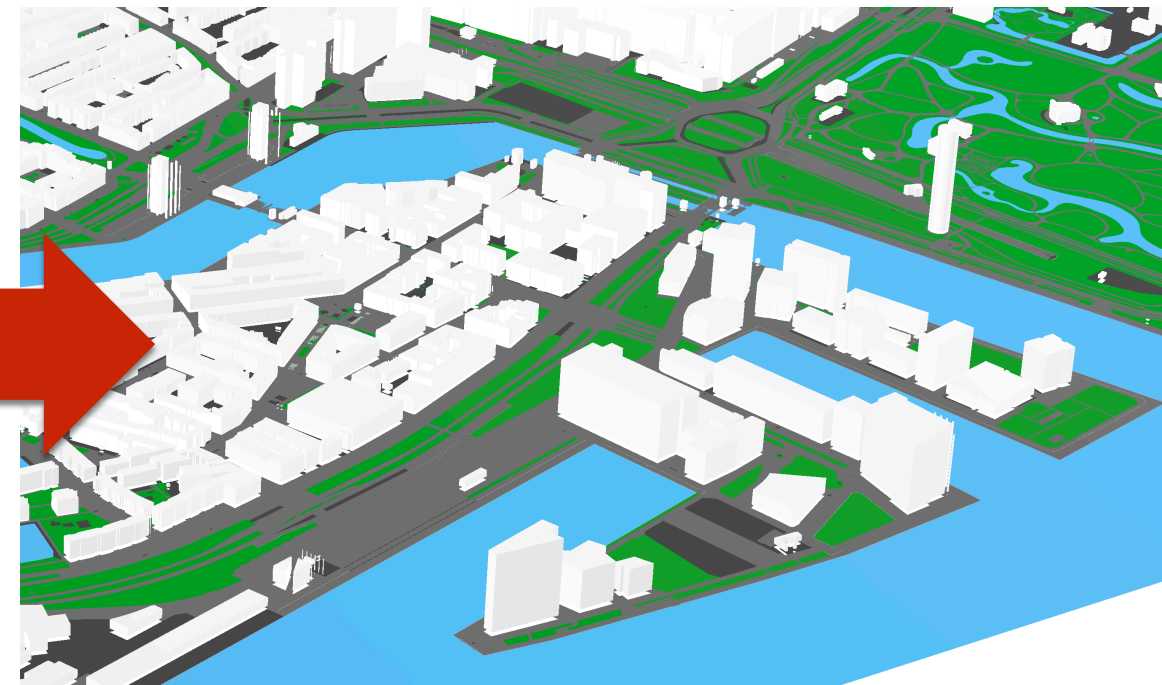
Why we need versioning?

- Keep track of changes
- Allow collaboration
- Allow experimentation with data
- Embrace small updates and distribution

3D city models



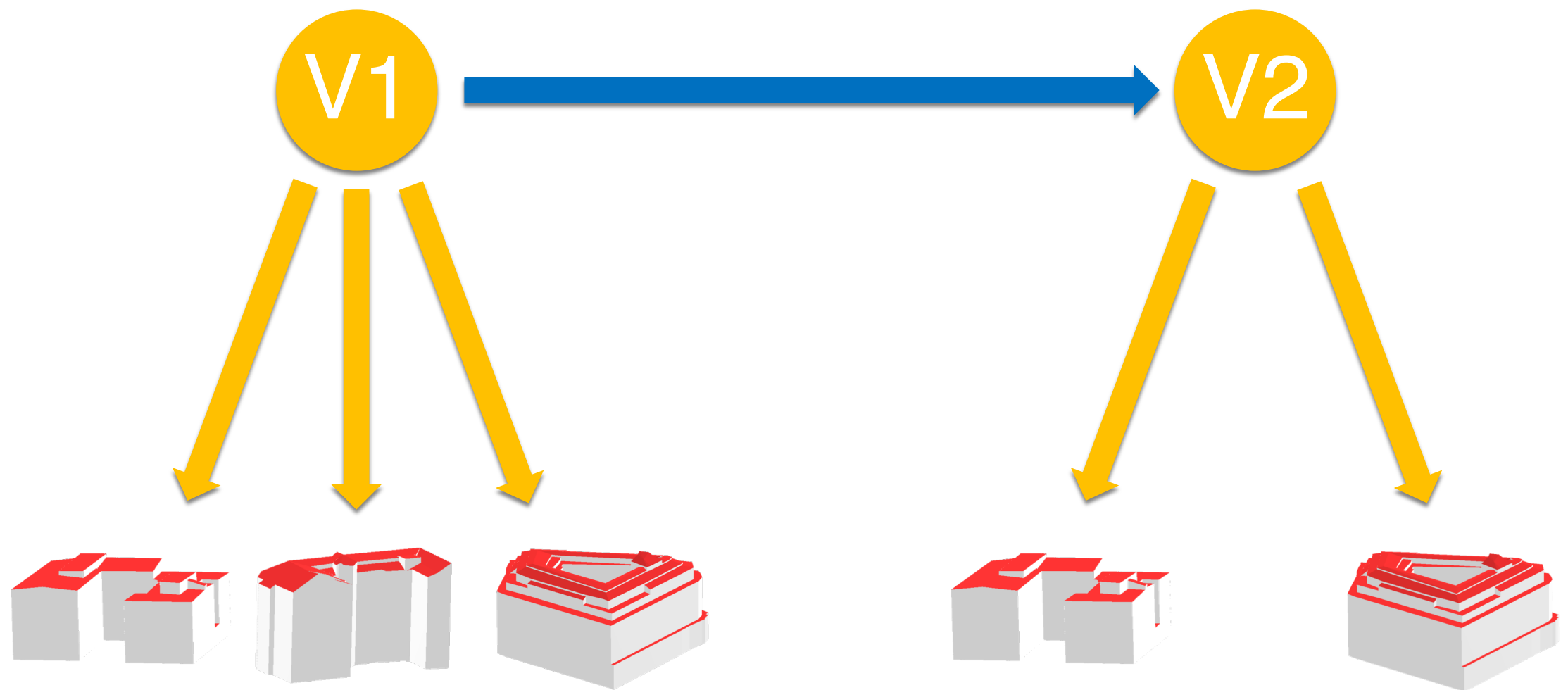
Modelling



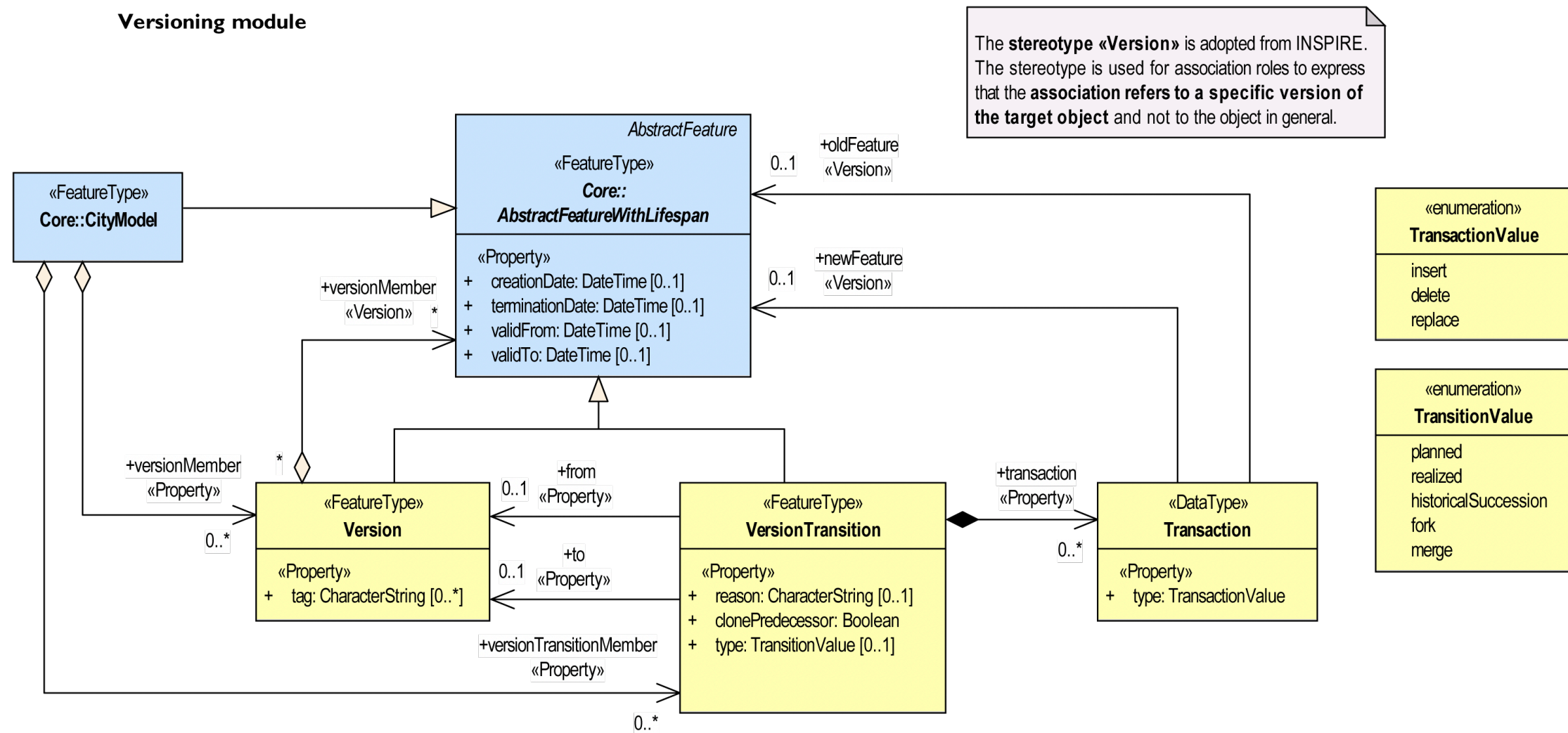
Reality

3D city model

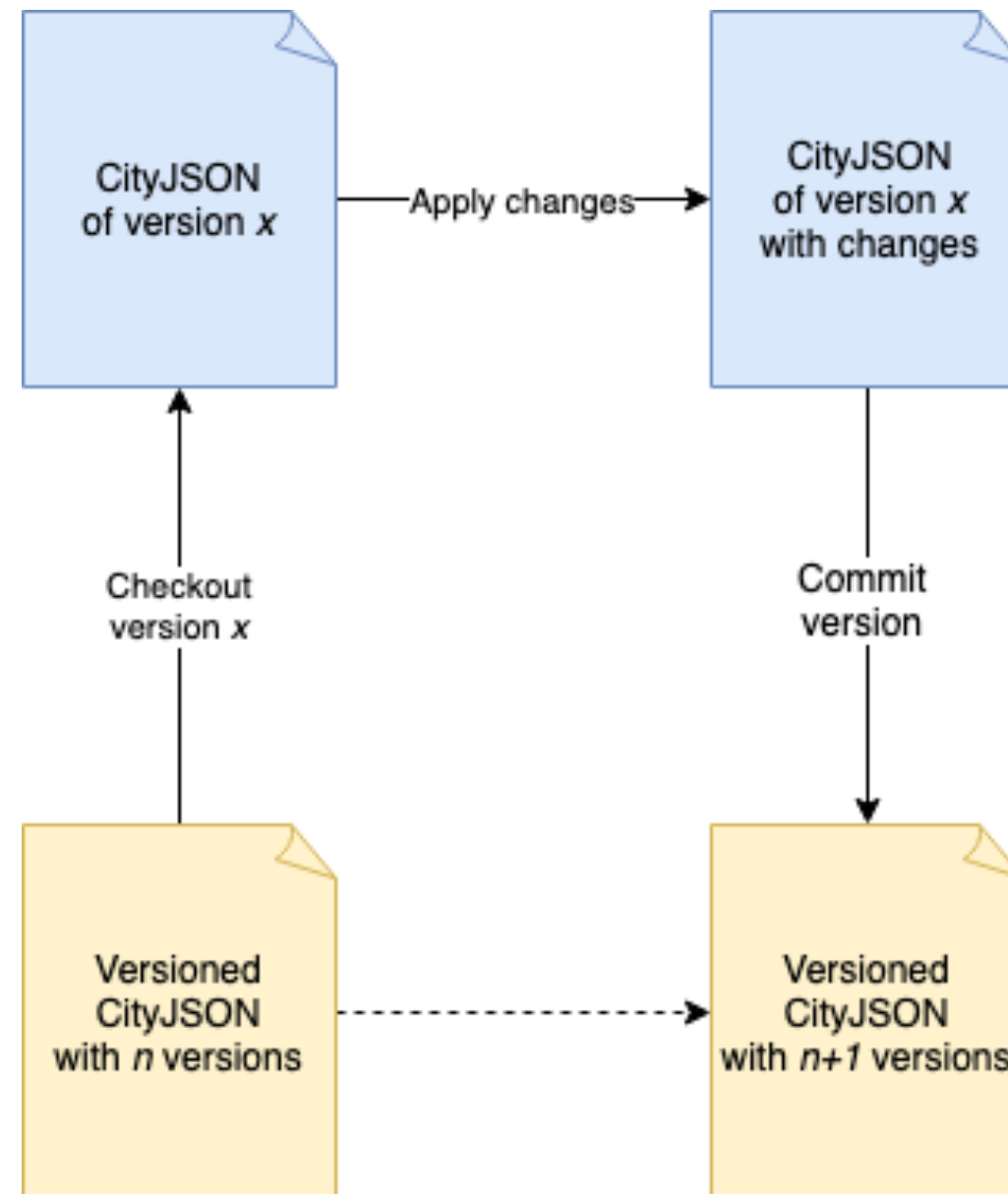
Versioning



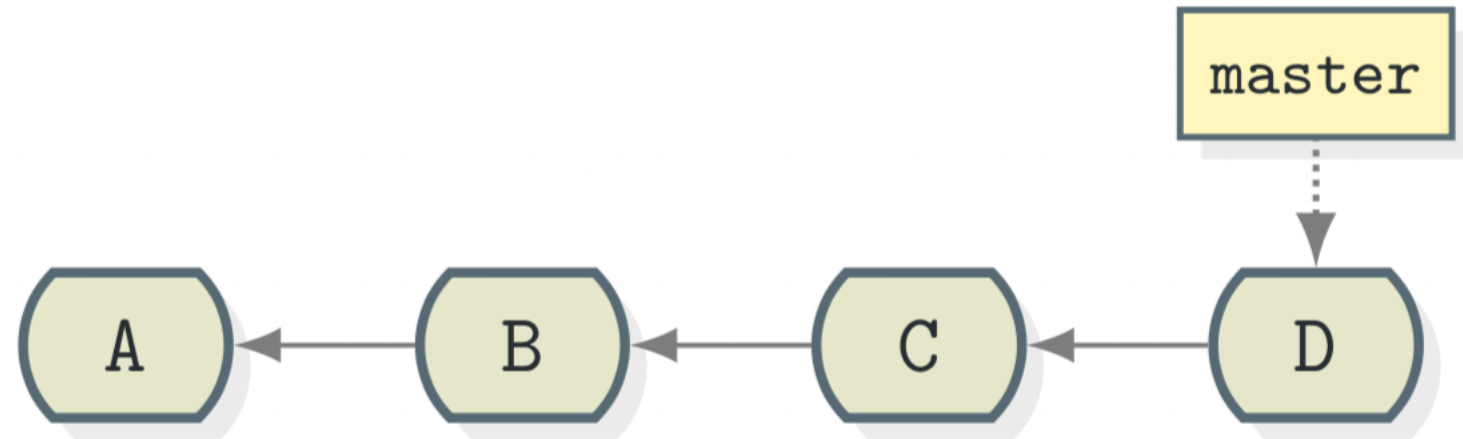
Versioning in CityGML 3.0



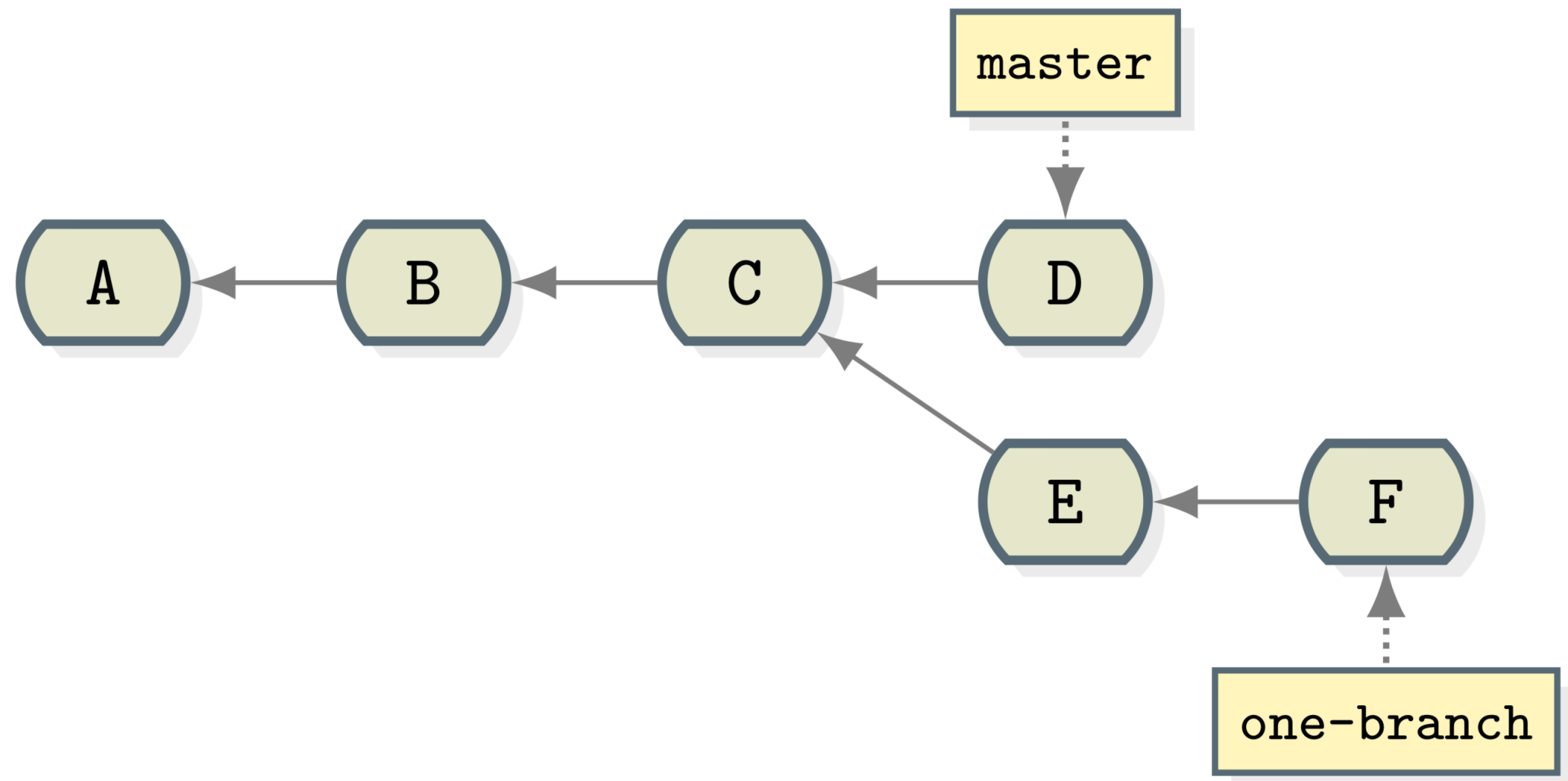
Versioning workflow



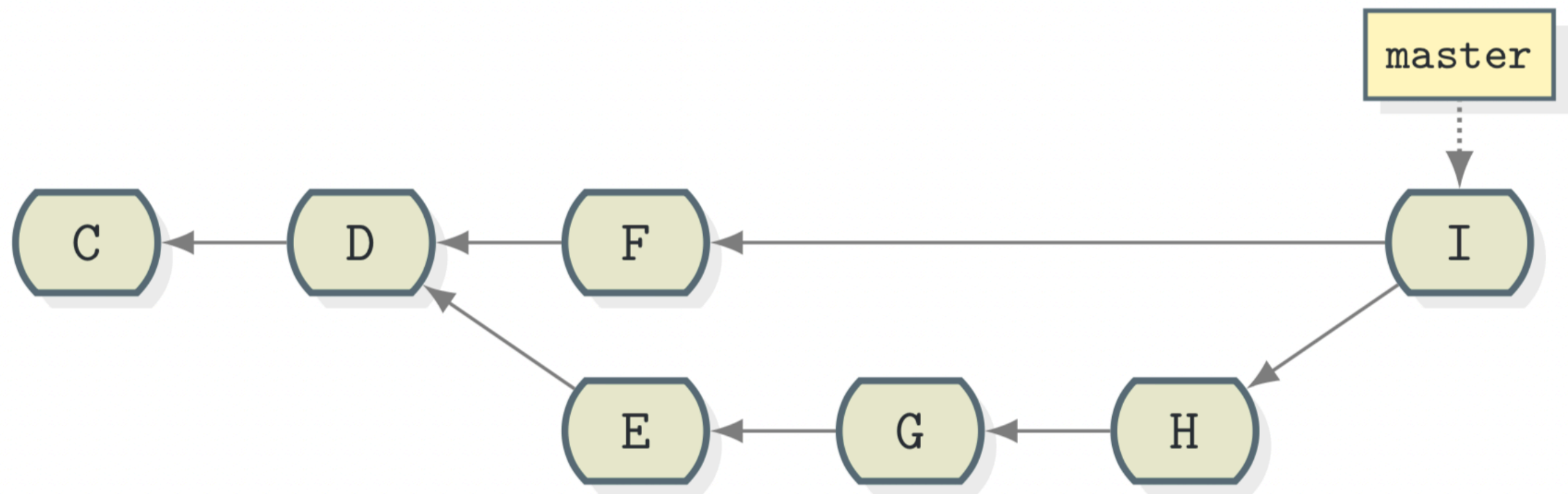
Version history is a DAG



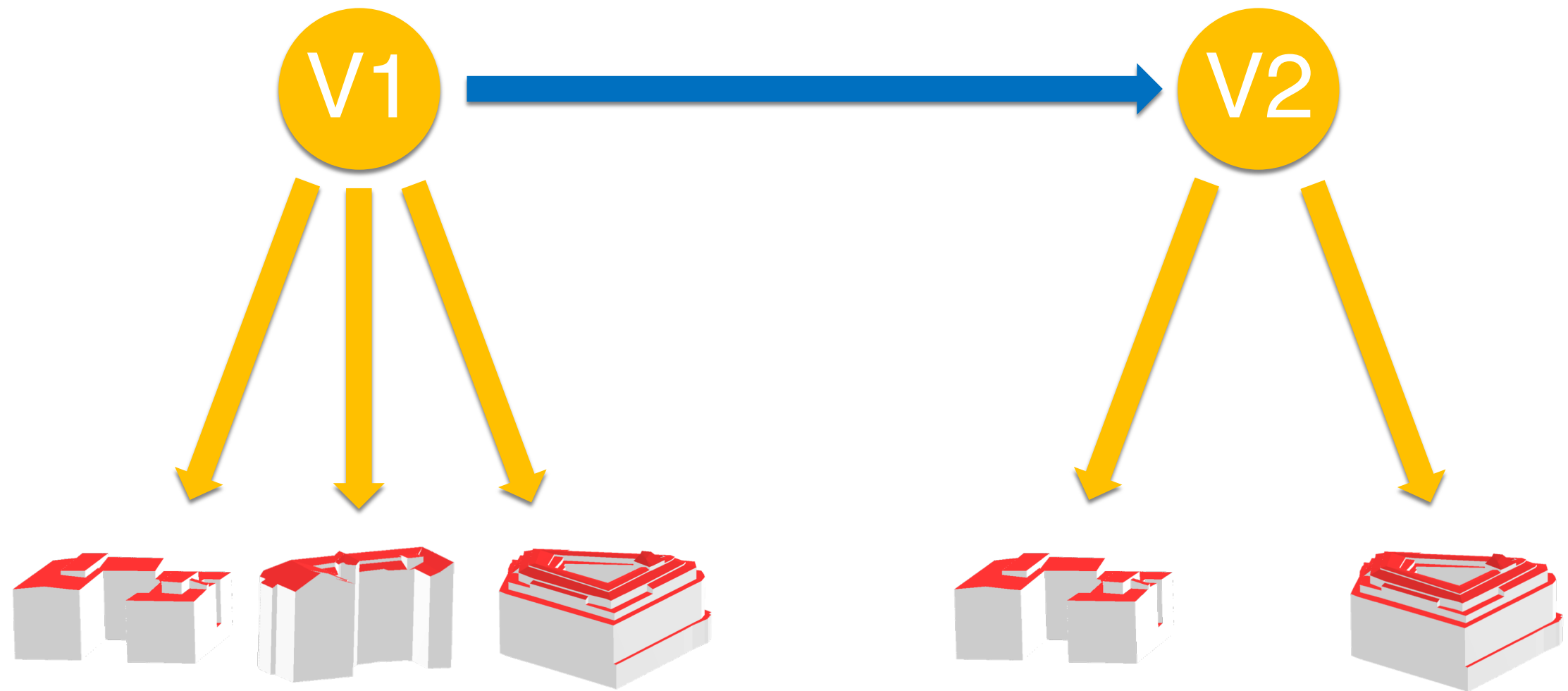
Version history is a DAG



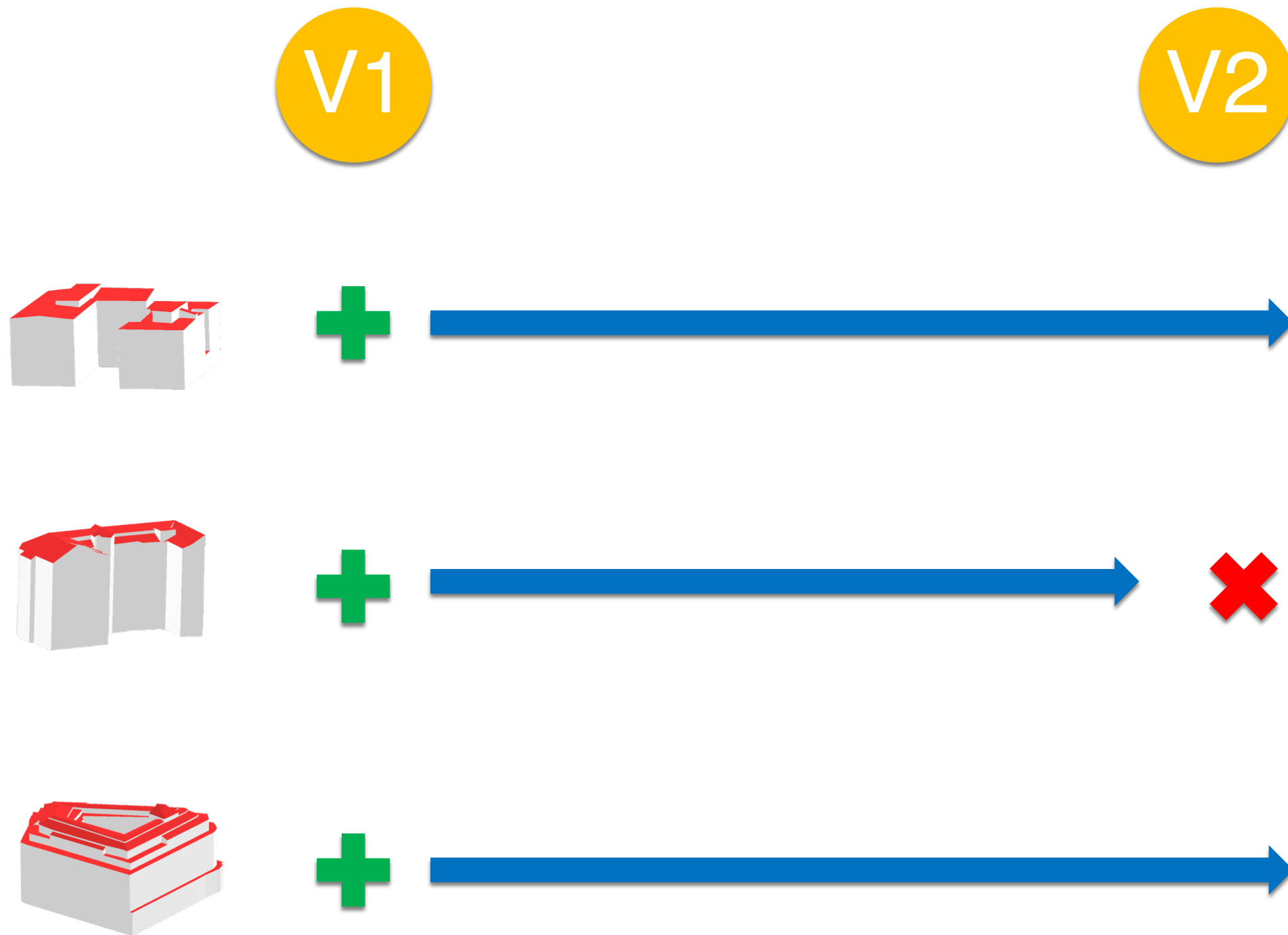
Branches



The two layers: 1) data structure



The two layers: 2) software



We have a paper!

ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W8,
123–130, 2019

<https://doi.org/10.5194/isprs-annals-IV-4-W8-123-2019>

© Author(s) 2019. This work is distributed under
the Creative Commons Attribution 4.0 License.



Volume IV-4/W8

23 Sep 2019

A DATA STRUCTURE TO INCORPORATE VERSIONING IN 3D CITY MODELS

S. Vitalis, A. Labetski, K. Arroyo Otori, H. Ledoux, and J. Stoter

3D Geoinformation Group, Delft University of Technology, the Netherlands

Keywords: Versioning, 3D City Models, CityJSON, CityGML, Git

Abstract. A 3D city model should be constantly updated with new versions, either to reflect the changes in its real-world counterpart, or to improve and correct parts of the model. However, the current standards for 3D city models do not support versioning, and existing version control systems do not work well with 3D city models. In this paper, we propose an approach to support versioning of 3D city models based on CityJSON and the concepts behind the Git version control system, including distributed and non-linear workflows. We demonstrate the benefits of our approach in two examples and in our software prototype, which is able to extract a given version of a 3D city model and to display its history.

[Conference paper](#) (PDF, 1067 KB)

How to cite: Vitalis, S., Labetski, A., Arroyo Otori, K., Ledoux, H., and Stoter, J.: A DATA STRUCTURE TO INCORPORATE VERSIONING IN 3D CITY MODELS, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W8, 123–130, <https://doi.org/10.5194/isprs-annals-IV-4-W8-123-2019>, 2019.

[BibTeX](#) [EndNote](#) [Reference Manager](#) [XML](#)

And we have software!

```
Opening buildingBeforeAndAfter.json...
Found 3 versions.

version v30 (master) (tag: release-2019)
Author: J.R.R. Tolkien
Date: 2019-03-04T18:34:12.24Z
Message:

Remove building01

This is what changed in this version:
- building01-02

version v29
Author: J.R.R. Tolkien
Date: 2019-02-04T11:00:17.58Z
Message:

Change ownership and structure of building01

This is what changed in this version:
+ building01-02
- building01-01

version v28
Author: J.R.R. Tolkien
Date: 2019-01-02T13:20:21.50Z
Message:

Add Baggins' new building

This is what changed in this version:
+ building01-01
```

<https://github.com/tudelft3d/cityjson-versioning-prototype>

Discussion

- Versioning is a form of metadata (and metadata is important).
- It can reshape the content itself.
- Fits perfect with the GeoBIM world.
- We should be open to changes and collaboration.

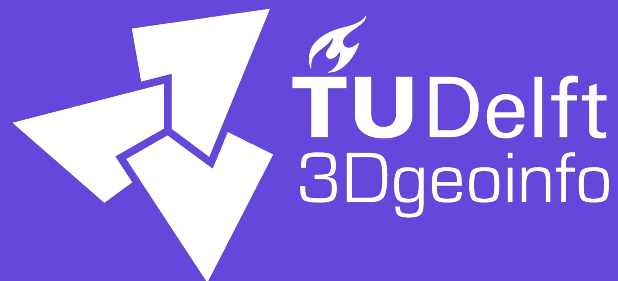
Future work

- Master's thesis (Gemeente Rotterdam)
- Evolve the CLI of prototype
- Develop a web UI

Thank you!

<https://github.com/tudelft3d/cityjson-versioning-prototype>

<https://doi.org/10.5194/isprs-annals-IV-4-W8-123-2019>



<https://3d.bk.tudelft.nl>



[@tudelft3d](https://twitter.com/tudelft3d)



[tudelft3d](https://github.com/tudelft3d)

<https://3d.bk.tudelft.nl/svitalis>



[@liberostelios](https://twitter.com/liberostelios)



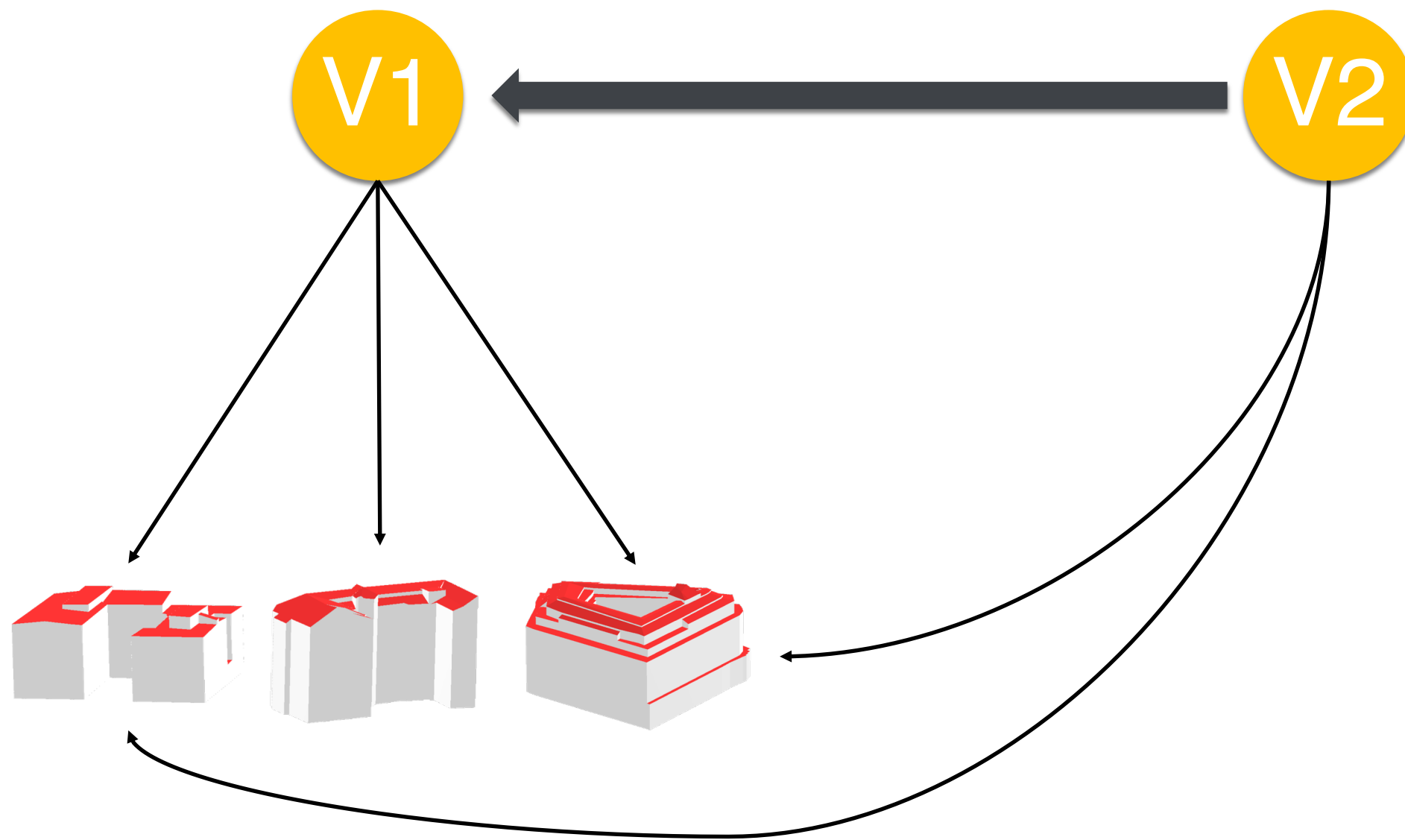
[liberostelios](https://github.com/liberostelios)

Identifiers

- Two city objects can't have the same ID
- Can use original ID and timestamp or version name to append id
- Best candidate is hashes (e.g. SHA256) and keeping id as attribute

```
▼ d38e4fc242c5d59fff9468c3ddabadba4df82afd:
  type: "Building"
  ► attributes: {}
  ► geometry: [...]
  cityobject_id: "{10A31B7E-8FC2-45EE-836A-1DB9FEF06E04}"
```

Data structure



Example?

```
"CityObjects": {  
  "building1": {  
    "type": "Building",  
    "geometry": [ ... ]  
  },  
  "building2": {  
    "type": "Building",  
    "geometry": [ ... ]  
  },  
  "building3": {  
    "type": "Building",  
    "geometry": [ ... ]  
  }  
}
```

```
"versions": {  
  "v1": {  
    "author": "John Doe",  
    "date": "2019-05-27T14:08:56.145473Z",  
    "message": "Initial commit",  
    "parent": [ ],  
    "objects": [  
      "building1",  
      "building2",  
      "building3"  
    ]  
  },  
  "v2": {  
    "author": "John Doe",  
    "date": "2019-05-29T14:08:56.145473Z",  
    "message": "Update renovated building",  
    "parent": [ "v1" ],  
    "objects": [  
      "building1",  
      "building3"  
    ]  
  },  
}
```

Example?

```
"CityObjects": {  
  "building1": {  
    "type": "Building",  
    "geometry": [ ... ]  
  },  
  " building2": {  
    "type": "Building",  
    "geometry": [ ... ]  
  },  
  " building1-renovated": {  
    "type": "Building",  
    "geometry": [ ... ]  
  }  
}
```

```
"versions": {  
  "v1": {  
    "author": "John Doe",  
    "date": " 2019-05-27T14:08:56.145473Z",  
    "message": "Initial commit",  
    "parents": [ ],  
    "objects": [  
      "building1",  
      "building2"  
    ]  
  },  
  "v2": {  
    "author": "John Doe",  
    "date": " 2019-05-29T14:08:56.145473Z",  
    "message": "Update renovated building",  
    "parents": [ "v1" ],  
    "objects": [  
      "building1-renovated",  
      "building2"  
    ]  
  },  
}
```

Refs (branches and tags)

```
"branches": {  
  "master": "v25",  
  "wind-simulation": "v28"  
},  
"tags": {  
  "release-2018": "v21",  
  "release-2017": "v14"  
},  
}
```