



[Lessons learned from]
Euclid Data Model qualification

Teake Nutma¹, Christophe Dabin², Edwin A. Valentijn¹



euclid



**kapteyn astronomical
institute**



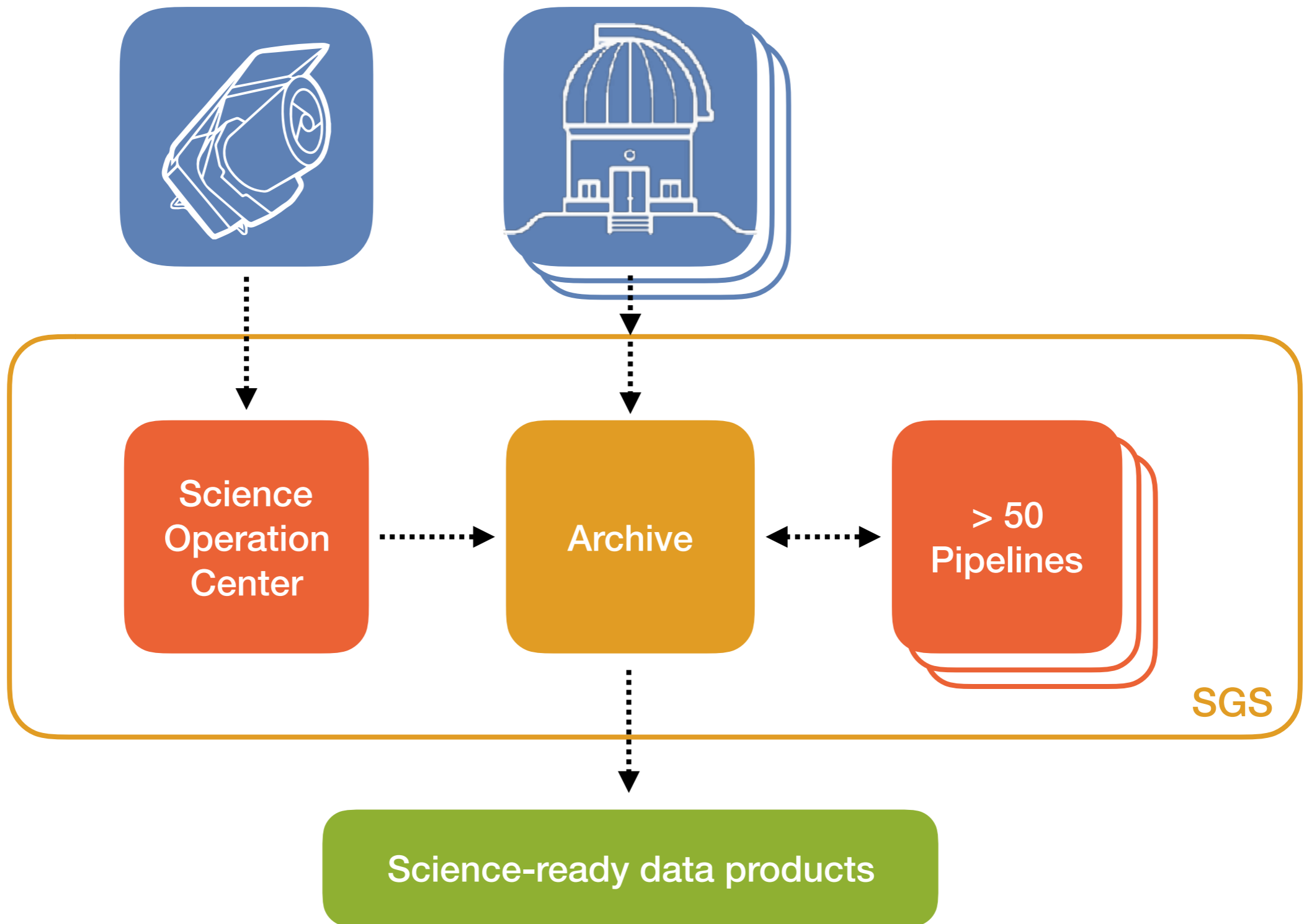
OmegaCEN

¹ Kapteyn Astronomical Institute, University of Groningen, The Netherlands

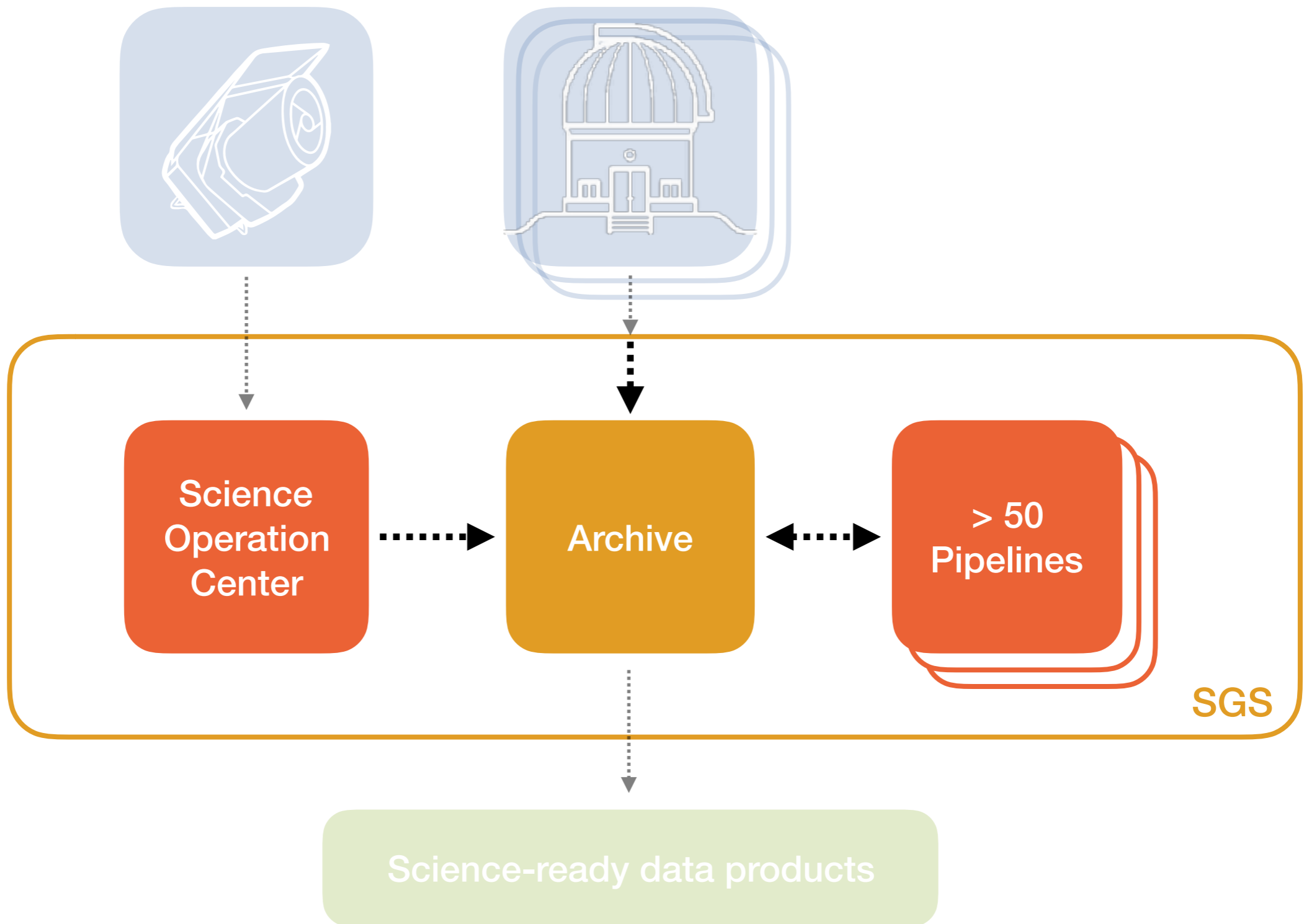
² Centre National d'Études Spatiales, Toulouse, France



The SGS and the Data Model



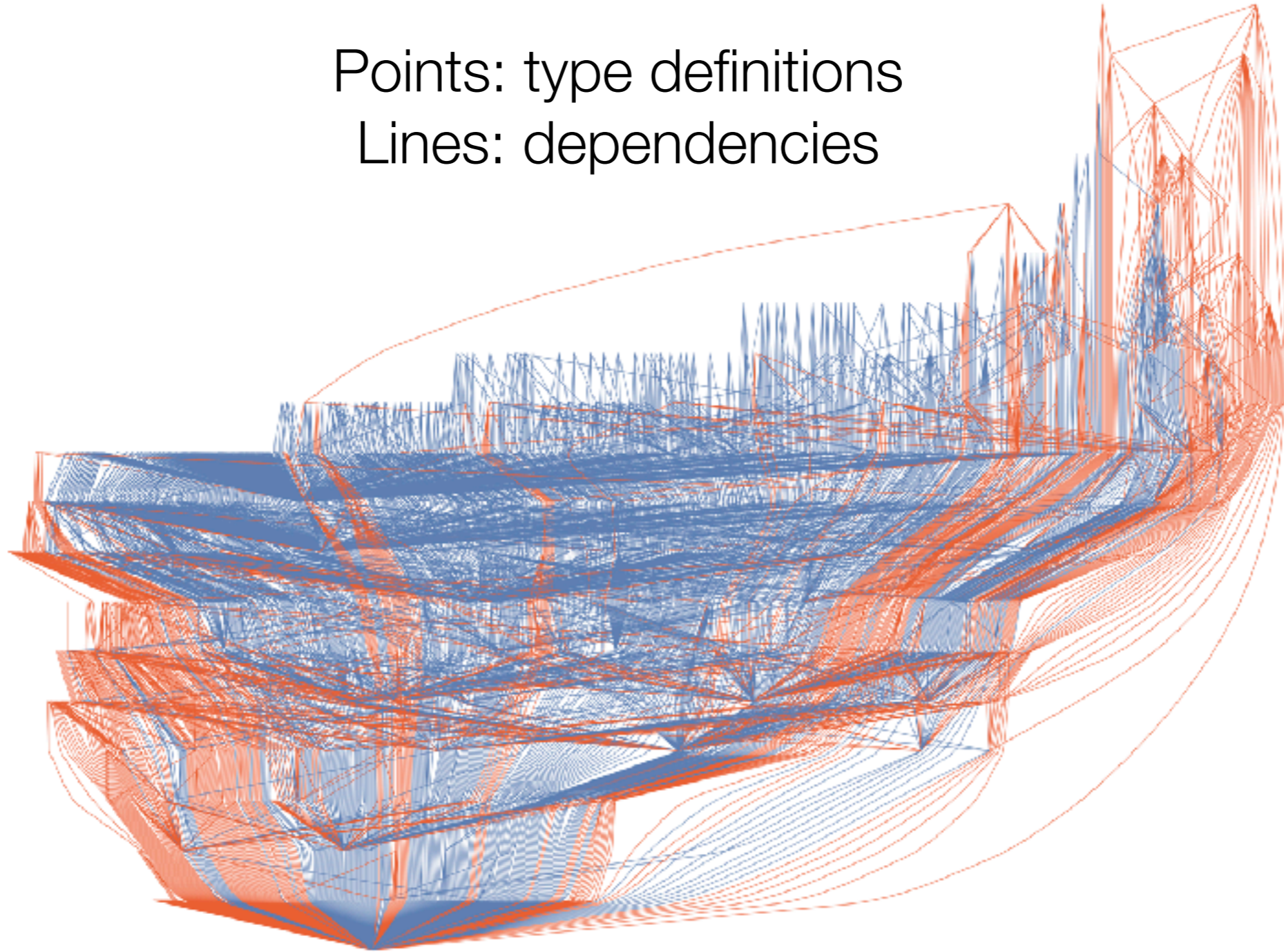
The SGS and the Data Model



The Euclid Data Model

Points: type definitions

Lines: dependencies



Data Product Definitions

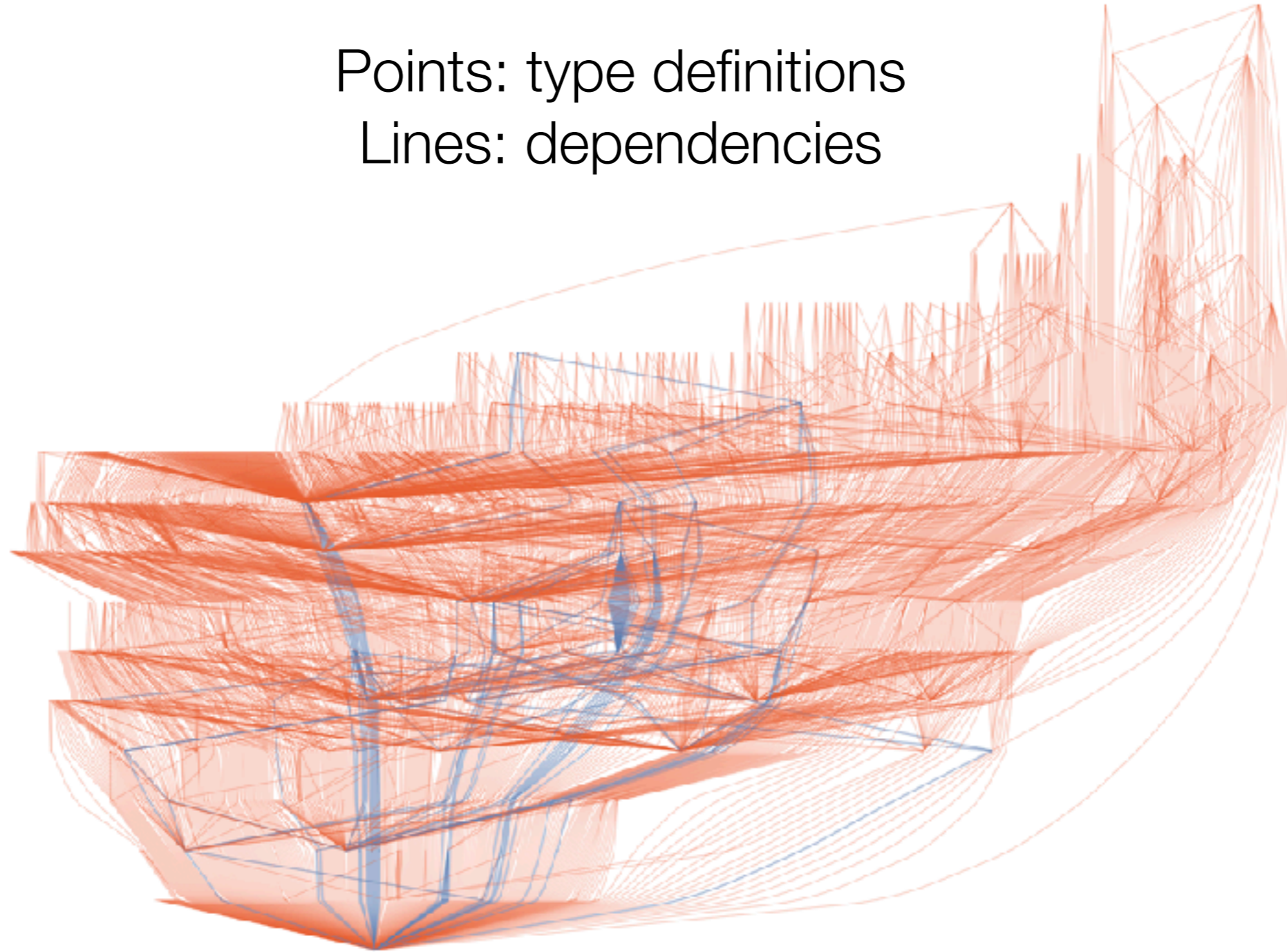


Other

The Euclid Data Model

Points: type definitions

Lines: dependencies



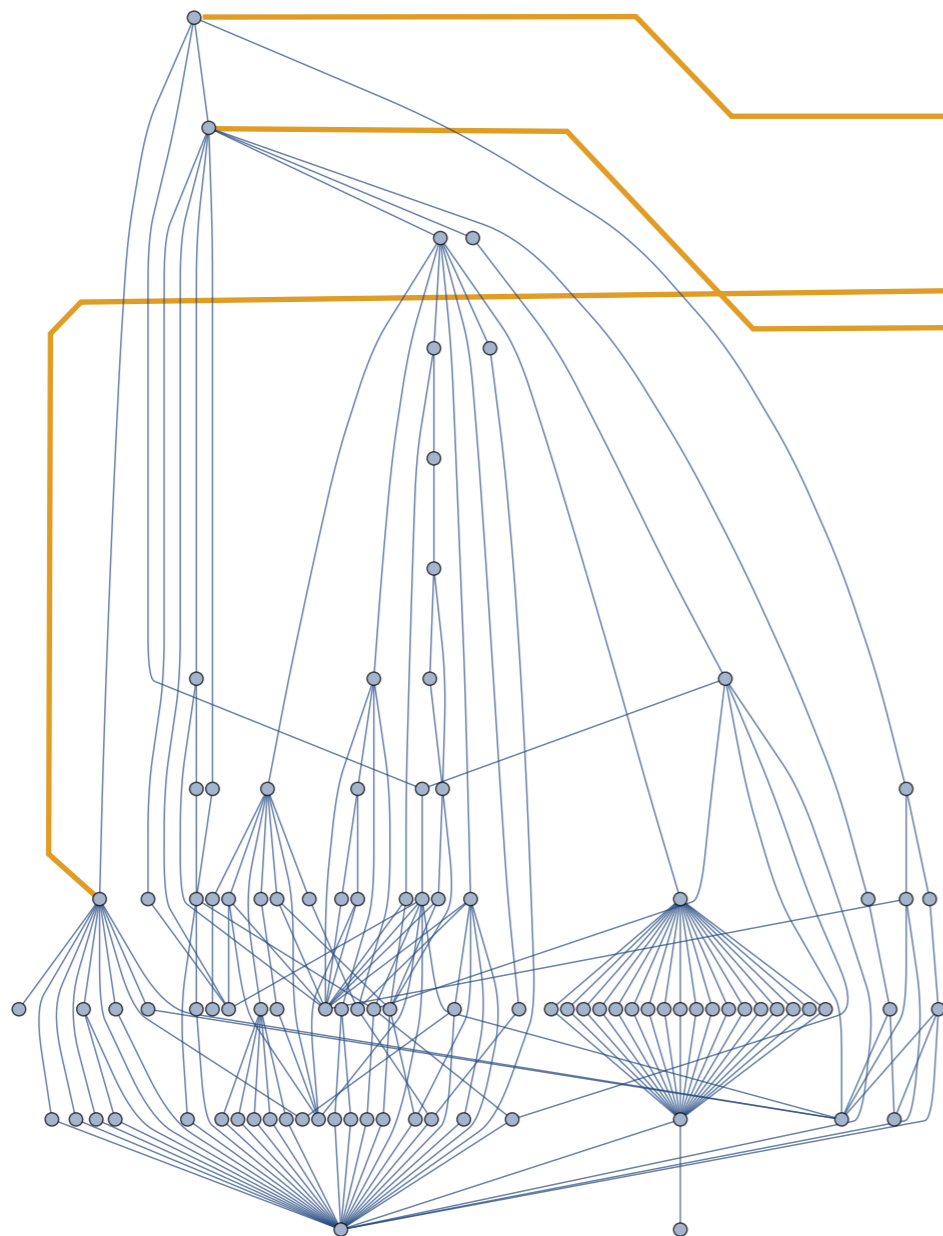
MER Mosaic DPD



Other

The Euclid Data Model

MER Mosaic DPD



```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  ◯ <xs:element name="DpdMerMosaic" type="dpdMerMosaic">  
    <xs:complexType name="dpdMerMosaic">  
      <xs:sequence>  
        ◯ <xs:element name="Header" type="genericHeader"/>  
        ◯ <xs:element name="Data" type="merMosaic"/>  
        <!-- ... -->  
      </xs:sequence>  
    </xs:complexType>  
  
    <xs:complexType name="merMosaic">  
      <xs:complexContent>  
        <xs:extension base="stackedImage">  
          <xs:sequence>  
            <xs:element name="TileIndex" type="xs:integer"/>  
            <!-- ... -->  
          </xs:sequence>  
        </xs:extension>  
      </xs:complexContent>  
    </xs:complexType>  
  
    <!-- ... -->  
  </xs:element>  
</xs:schema>
```

XSD Data Model file

The Euclid Data Model

MER Mosaic DPD

Validation



```
<DpdMerMosaic>
  <Header>
    <!-- ... -->
  </Header>
  <Data>
    <!-- ... -->
    <TileIndex>741773</TileIndex>
    <!-- ... -->
  </Data>
  <!-- ... -->
</DpdMerMosaic>
```

XML metadata file

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DpdMerMosaic" type="dpdMerMosaic">
    <xs:complexType name="dpdMerMosaic">
      <xs:sequence>
        <xs:element name="Header" type="genericHeader"/>
        <xs:element name="Data" type="merMosaic"/>
        <!-- ... -->
      </xs:sequence>
    </xs:complexType>
  <xs:complexType name="merMosaic">
    <xs:complexContent>
      <xs:extension base="stackedImage">
        <xs:sequence>
          <xs:element name="TileIndex" type="xs:integer"/>
          <!-- ... -->
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!-- ... -->
</xs:schema>
```

XSD Data Model file

Implementing the Data Model

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DpdMerMosaic" type="dpdMerMosaic">
    <xs:complexType name="dpdMerMosaic">
      <xs:sequence>
        <xs:element name="Header" type="genericHeader"/>
        <xs:element name="Data" type="merMosaic"/>
        <!-- ... -->
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="merMosaic">
      <xs:complexContent>
        <xs:extension base="stackedImage">
          <xs:sequence>
            <xs:element name="TileIndex" type="xs:integer"/>
            <!-- ... -->
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <!-- ... -->
  </xs:schema>
```

XSD Data Model file

Code generation



```
from wise_orm import DBObject, persistent
# ...

class MerMosaic(StackedImage):
    tile_index = persistent('', int, None)
    # ...

class DpdMerMosaic(DBObject):
    header = persistent('', GenericHeader, None)
    data = persistent('', MerMosaic, None)
    # ...
```

EAS-DPS Python binding

Implementing the Data Model

- XSD `xs:integer` is unbounded
- But database numeric types are bounded!
- So impose rule:
Don't use unbounded XSD types (EUCLID-STD-E-HB-80-503_0210)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DpdMerMosaic" type="dpdMerMosaic">
    <xs:complexType name="dpdMerMosaic">
      <xs:sequence>
        <xs:element name="Header" type="genericHeader"/>
        <xs:element name="Data" type="merMosaic"/>
        <!-- ... -->
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="merMosaic">
      <xs:complexContent>
        <xs:extension base="stackedImage">
          <xs:sequence>
            <xs:element name="TileIndex" type="xs:integer" />
            <!-- ... -->
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <!-- ... -->
  </xs:schema>
```

XSD Data Model file

```
from wise_orm import DBObject, persistent
# ...

class MerMosaic(StackedImage):
    tile_index = persistent('', int, None)
    # ...

class DpdMerMosaic(DBObject):
    header = persistent('', GenericHeader, None)
    data = persistent('', MerMosaic, None)
    # ...
```

EAS-DPS Python binding

Implementing the Data Model

- XSD `xs:integer` is unbounded
- But database numeric types are bounded!
- So impose rule:
Don't use unbounded XSD types
(*EUCLID-STD-E-HB-80-503_0210*)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DpdMerMosaic" type="dpdMerMosaic">
    <xs:complexType name="dpdMerMosaic">
      <xs:sequence>
        <xs:element name="Header" type="genericHeader"/>
        <xs:element name="Data" type="merMosaic"/>
        <!-- ... -->
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="merMosaic">
      <xs:complexContent>
        <xs:extension base="stackedImage">
          <xs:sequence>
            <xs:element name="TileIndex" type="xs:long" />
            <!-- ... -->
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <!-- ... -->
  </xs:schema>
```

XSD Data Model file

```
from wise_orm import DBObject, persistent
# ...

class MerMosaic(StackedImage):
    tile_index = persistent('', int, None)
    # ...

class DpdMerMosaic(DBObject):
    header = persistent('', GenericHeader, None)
    data = persistent('', MerMosaic, None)
    # ...
```

EAS-DPS Python binding

XML rules

- EUCLID-STD-E-HB-80-503_0210
- EUCLID-STD-E-HB-80-503_0010
- EUCLID-STD-E-HB-80-503_0110
- EUCLID-STD-E-HB-80-503_0130
- EUCLID-STD-E-HB-80-503_0160
- EUCLID-STD-E-HB-80-503_0170
- EUCLID-STD-E-HB-80-503_0240
- EUCLID-STD-E-HB-80-503_0215
- EUCLID-STD-E-HB-80-503_0260
- EUCLID-STD-E-HB-80-503_0280
- EUCLID-STD-E-HB-80-503_0310
- EUCLID-STD-E-HB-80-503_0320
- EUCLID-STD-E-HB-80-503_0330
- EUCLID-STD-E-HB-80-503_0340
- EUCLID-STD-E-HB-80-503_0350
- EUCLID-STD-E-HB-80-503_0370
- EUCLID-STD-E-HB-80-503_0380
- EUCLID-STD-E-HB-80-503_0386
- EUCLID-STD-E-HB-80-503_0387
- EUCLID-STD-E-HB-80-503_0400
- EUCLID-STD-E-HB-80-503_0426
- EUCLID-STD-E-HB-80-503_0427
- EUCLID-STD-E-HB-80-503_0430
- EUCLID-STD-E-HB-80-503_0450
- EUCLID-STD-E-HB-80-503_0451
- EUCLID-STD-E-HB-80-503_0452
- EUCLID-STD-E-HB-80-503_0453
- EUCLID-STD-E-HB-80-503_0470
- EUCLID-STD-E-HB-80-503_0480
- EUCLID-STD-E-HB-80-503_0530
- ...

XML rules

- We have a Word document containing all XML rules (*EUCL-CNE-SGS-INF-00239*)
- 74 pages, 60 rules
- Not super convenient for 67 data modellers
- Need program to automatically check for these rules



datamodelvalidator

```
$ datamodelvalidator /path/to/datamodel
```

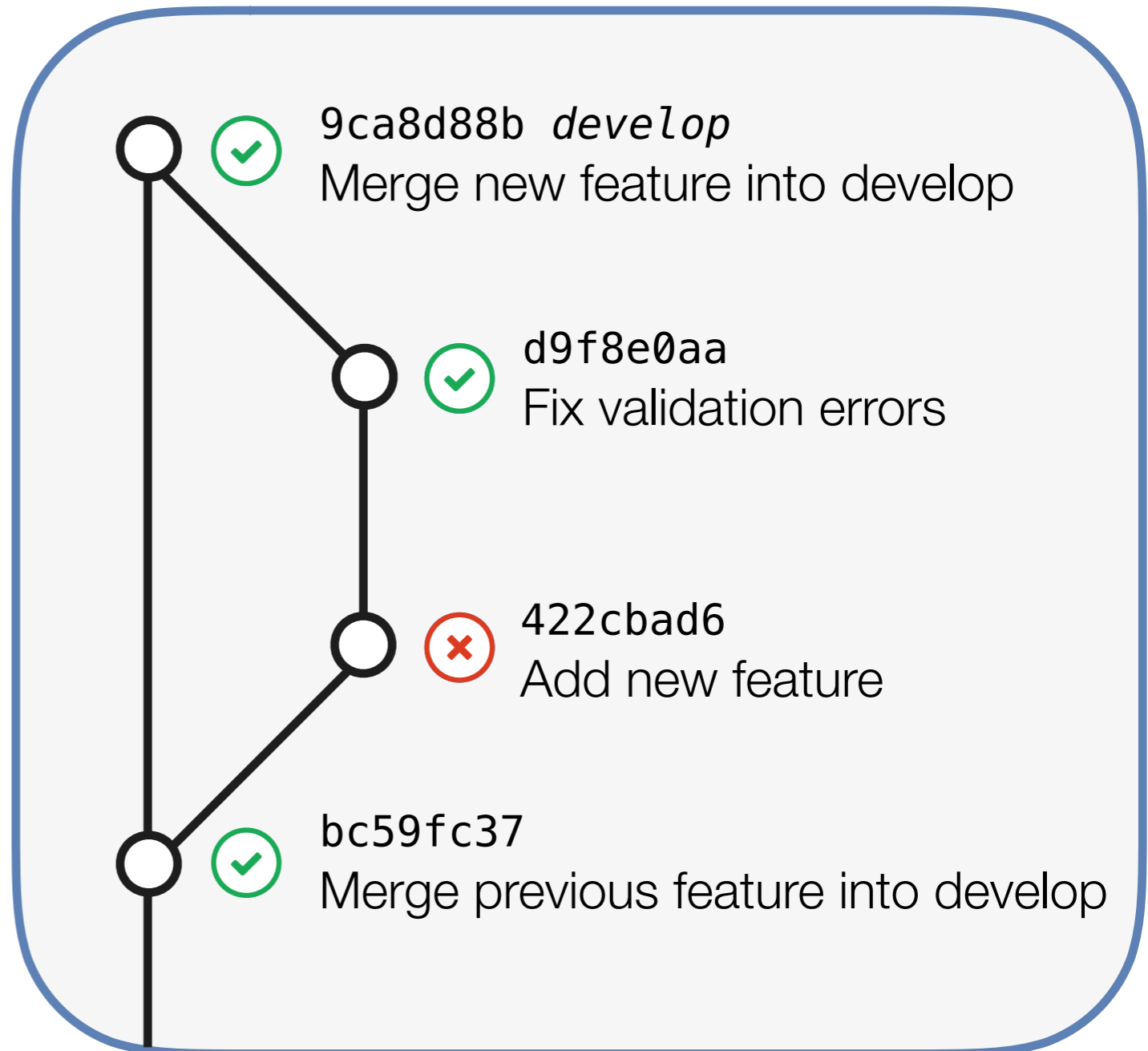
```
WARNING euc-test-stc.xsd:255 Type 'allSkyType' is empty.  
Data model is valid but contains issues (1 warning).
```

```
$ █
```

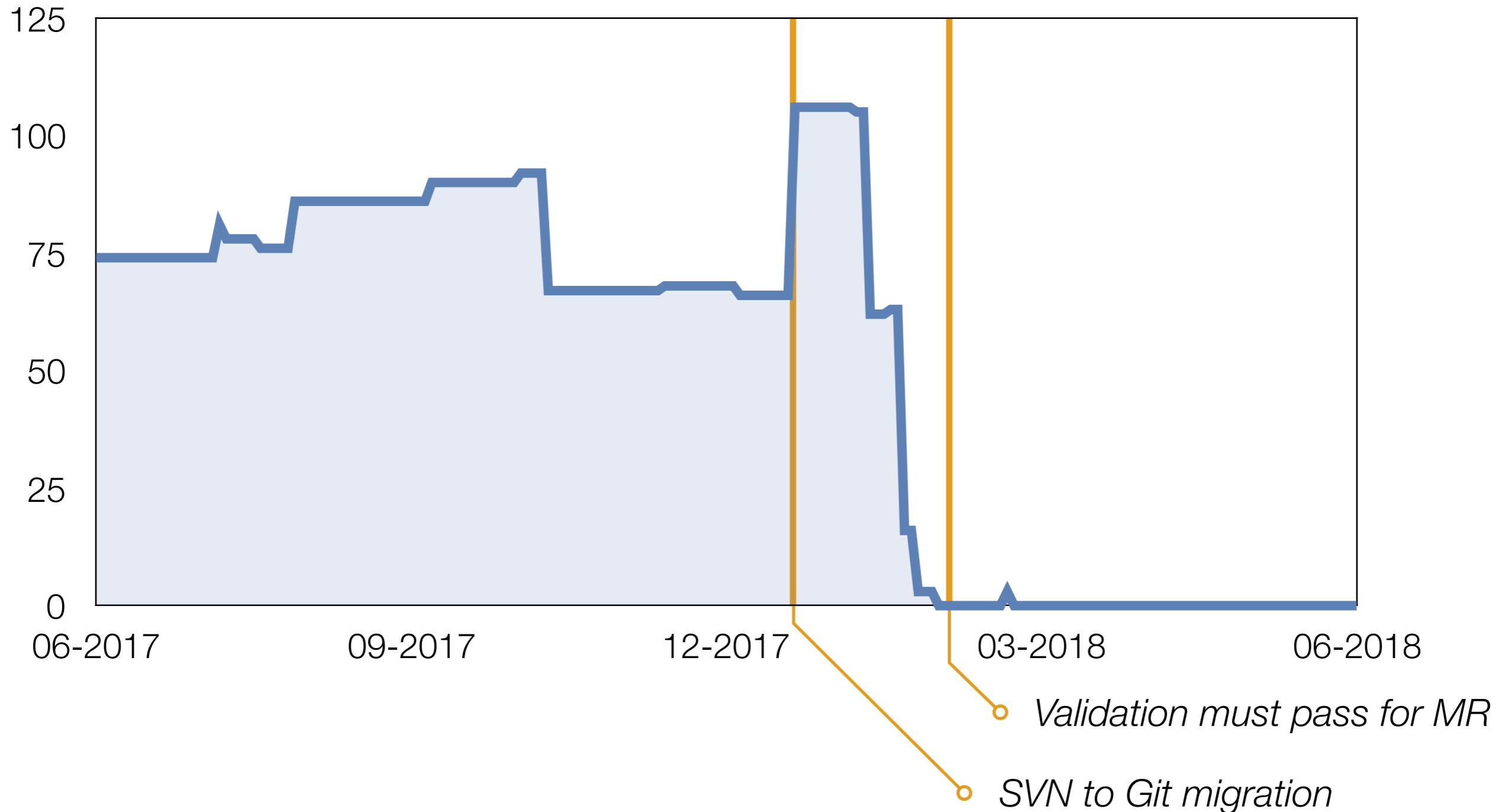
- CLI program that validates Data Model w.r.t. XML rules
- Schema for schemas and Python code
- Ensures database schema can be build from Data Model
- Data Modellers can run it locally before committing

Validation on GitLab

- Validation is run on every commit
- Pipeline **must pass** before changes can be merged
- (Almost) no possibility to introduce errors!



Euclid Data Model errors



Lessons learned

- XSD modelling space is too big for our domain-specific applications:
 - Relational databases
 - Object-oriented languages (C++, Python)
- Need rules to restrict to subspace of W3C XSD.

