JP2's preservation capabilities within PLANETS

Volker Heydegger

DPC/BL Joint JPEG2000 Workshop

British Library Conference Centre, St. Pancras, London

25th June 2007

Overview

- JP2 in the context of PLANET's XCL
  - Preservation Characterisation Workpackage
  - What is the eXtensible Characterisation Language (XCL)?
  - Translating JP2 into the eXtensible Characterisation Extraction Language (XCEL)
- Choosing it for preservation - Some remarks about the preservation capabilities of JPEG2000

PLANETS Preservation Characterisation WP: Overall goals

- Characterise content to support preservation
  - Define an interoperable, format-independent standard to describe the properties of digital objects (XCDL)
  - Define a standard for describing how properties are encoded in specific object formats (XCEL)
- Identify digital objects
  - Build on TNAs PRONOM service for file format identification

PLANETS
Preservation Characterisation WP: Overall goals

- Build a characterisation registry
  - Provide persistent, unambiguous technical information about object types such as format, compression method or environmental aspects
  - Provide detailed information about inherent characteristics such as fonts used in a document, resolution of a raster image etc.

PLANETS Preservation Characterisation WP: Overall goals

- Support other preservation issues
  - Preservation planning: Understanding the preservation requirements of different digital objects, creating collection profiles
  - Preservation actions: Comparing the properties of objects transformed within a migration process, validation of it

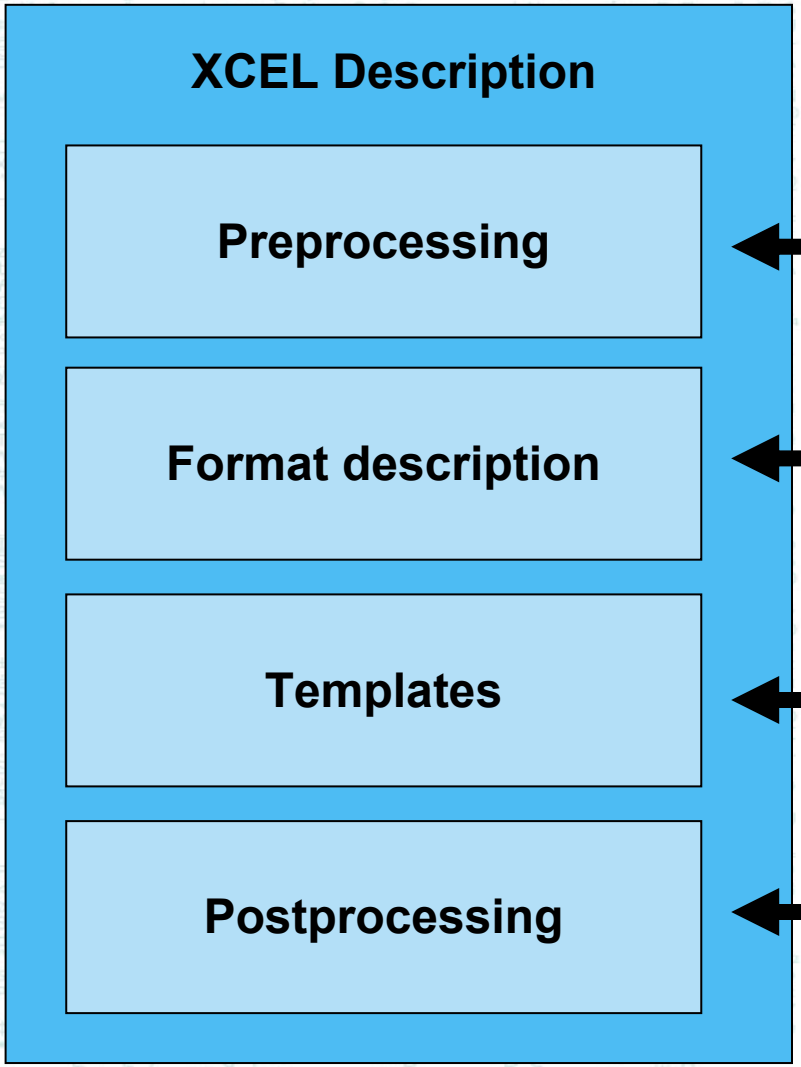
eXtensible Characterisation Language (XCL)

- Meta language, based on XML/ XML Schema
- eXtensible Characterisation Definition Language (XCDL):
  - Description of properties of a specific digital object
  - Content of a file with format type X, tagged in XML, according to the XCDL language definitions
  - Interpretable through an XCDL interpreter (e.g. Comparer)

eXtensible Characterisation Language (XCL)

- eXtensible Characterisation Extraction Language (XCEL):
  - Language to describe how properties of digital objects are stored
  - File format specification tagged in XML, according to the XCEL language definitions
  - Interpretable through an XCEL interpreter (e.g. Extractor)

XCEL: Global Architecture

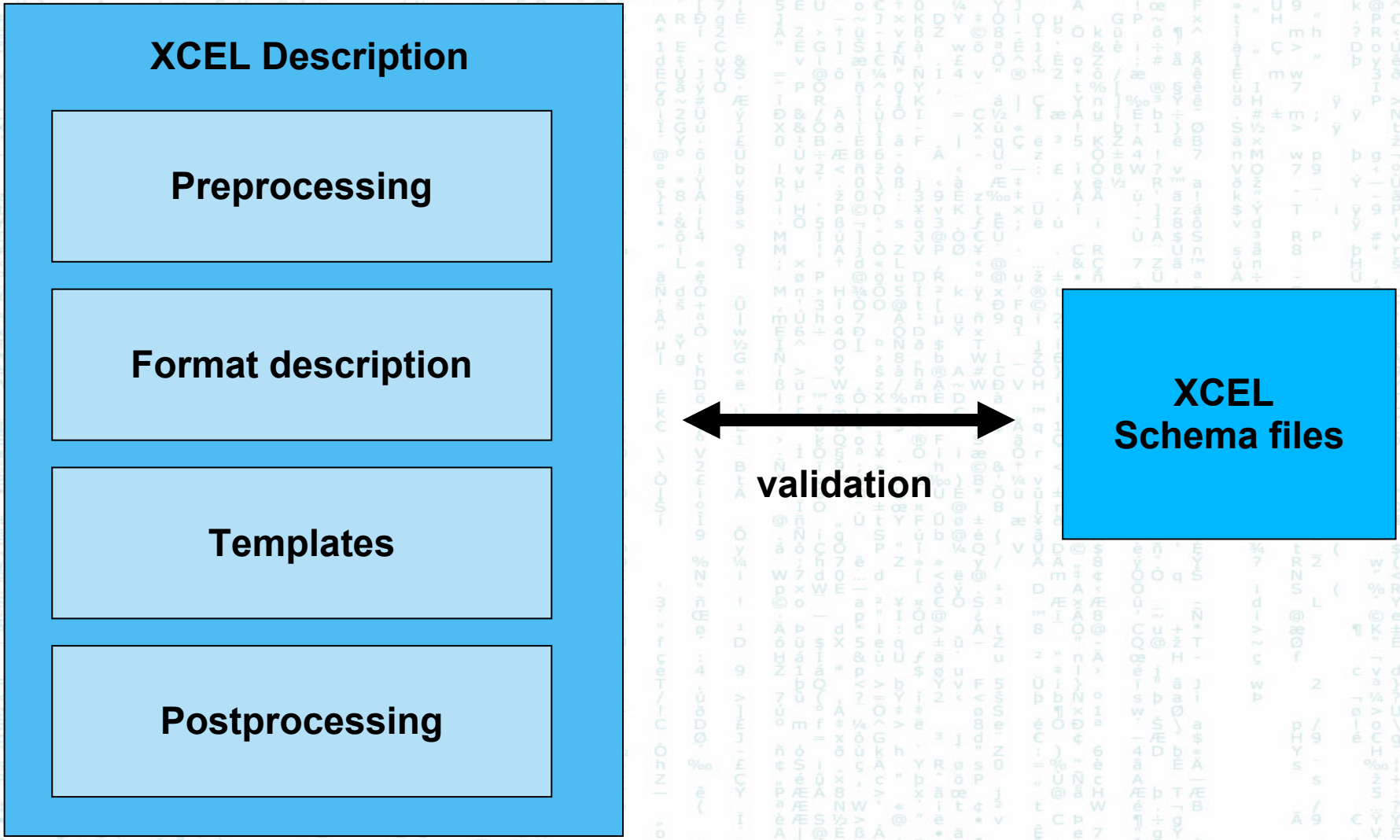


Configuration tasks, affecting the behaviour of the XCEL interpreter

Description of the structure of the object

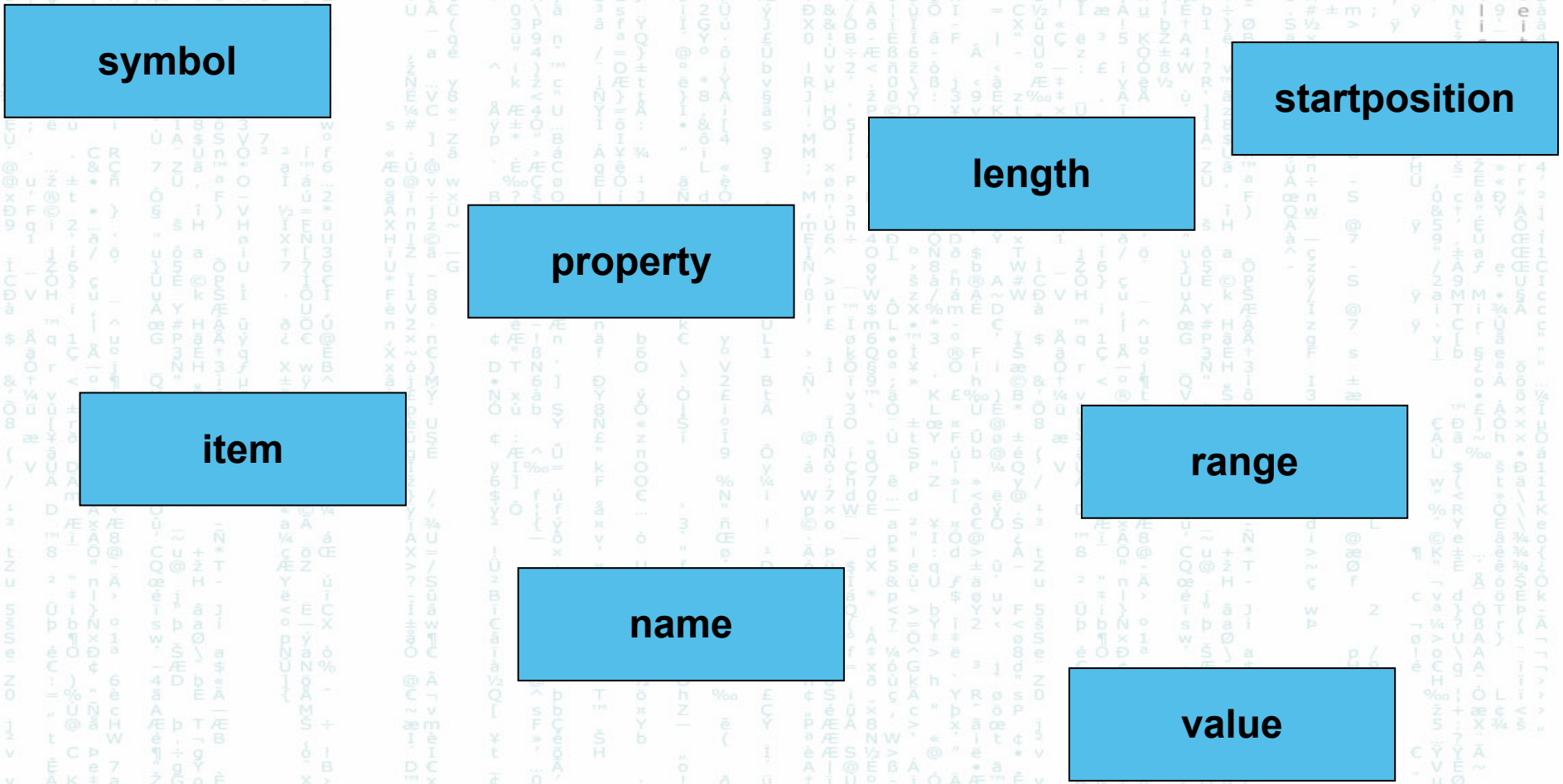
Description of recuring structures

Actions on the result of the format description processing


XCEL Description
Preprocessing
Format description
Templates
Postprocessing
validation
XCEL
Schema files

XCEL: Basic Structuring Elements

There are just a few elements sufficient enough to describe a file format:



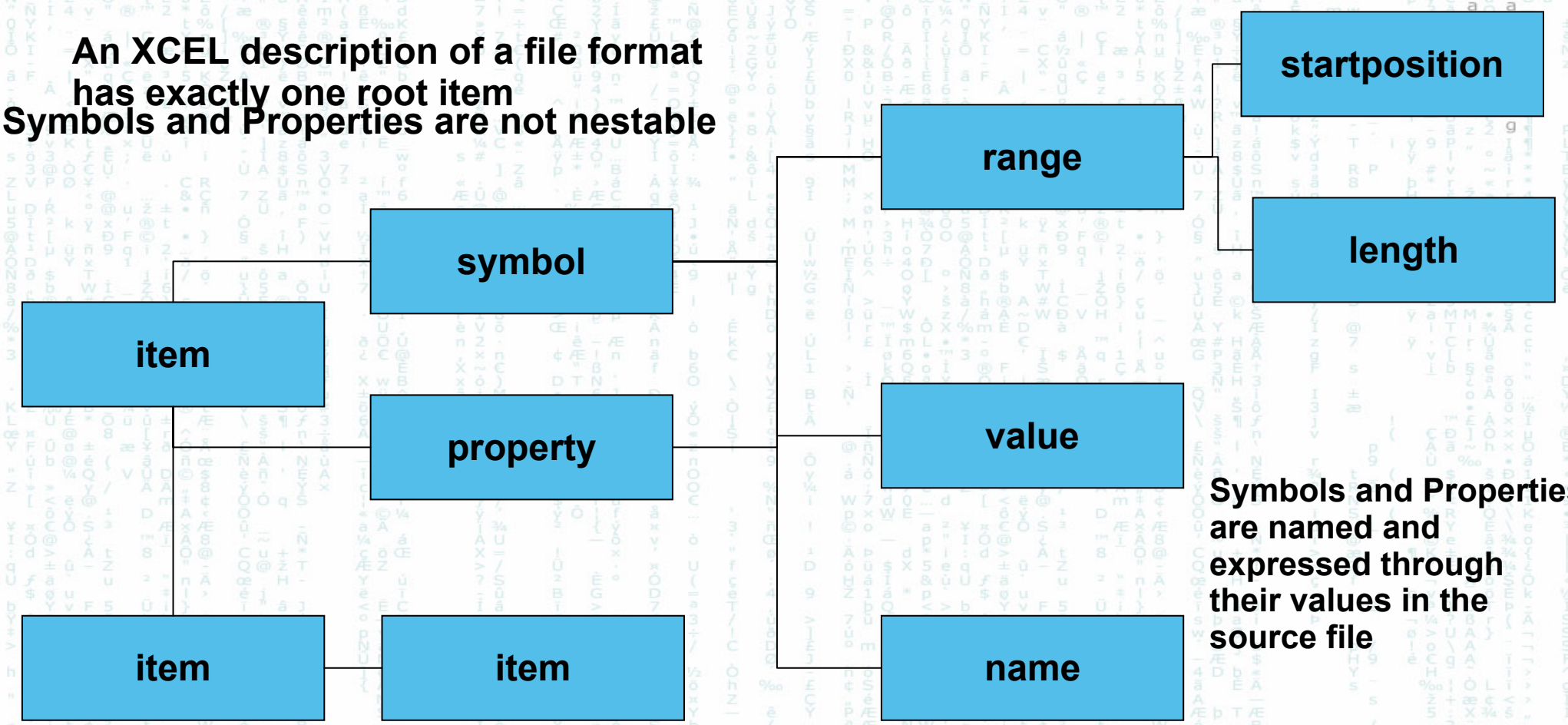
XCEL: Basic Structuring Elements

Symbols and properties can be located within a file by indicating their start position and length

These elements are structured in a tree-like hierarchy

An XCEL description of a file format has exactly one root item

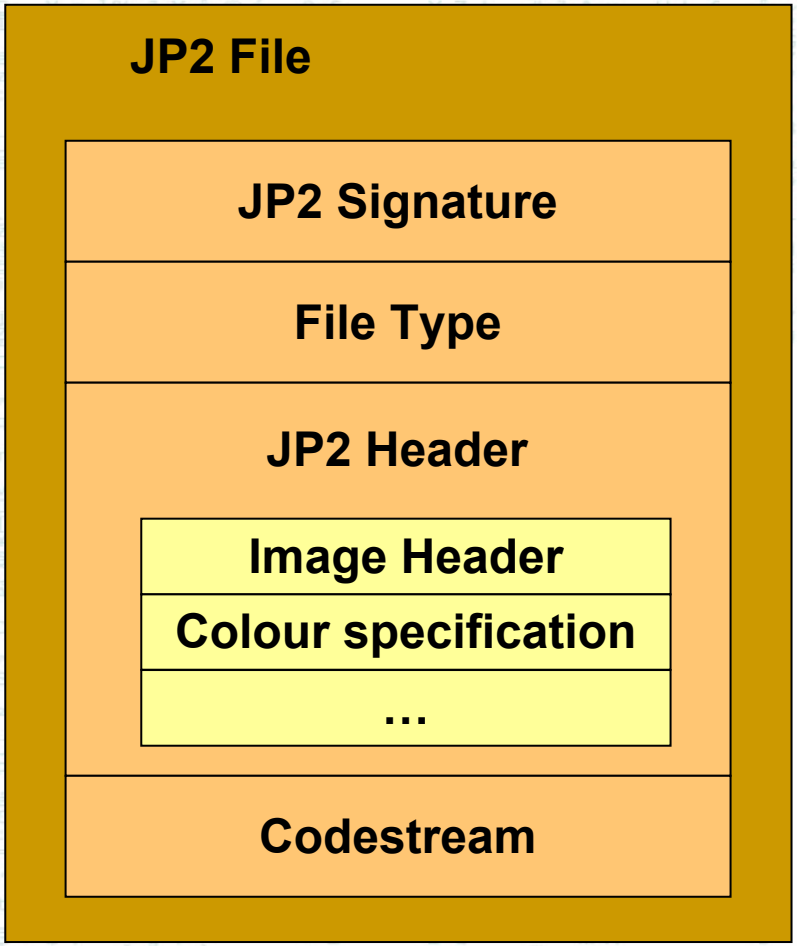
Symbols and Properties are not nestable



Symbols and Properties are named and expressed through their values in the source file

Items are nestable

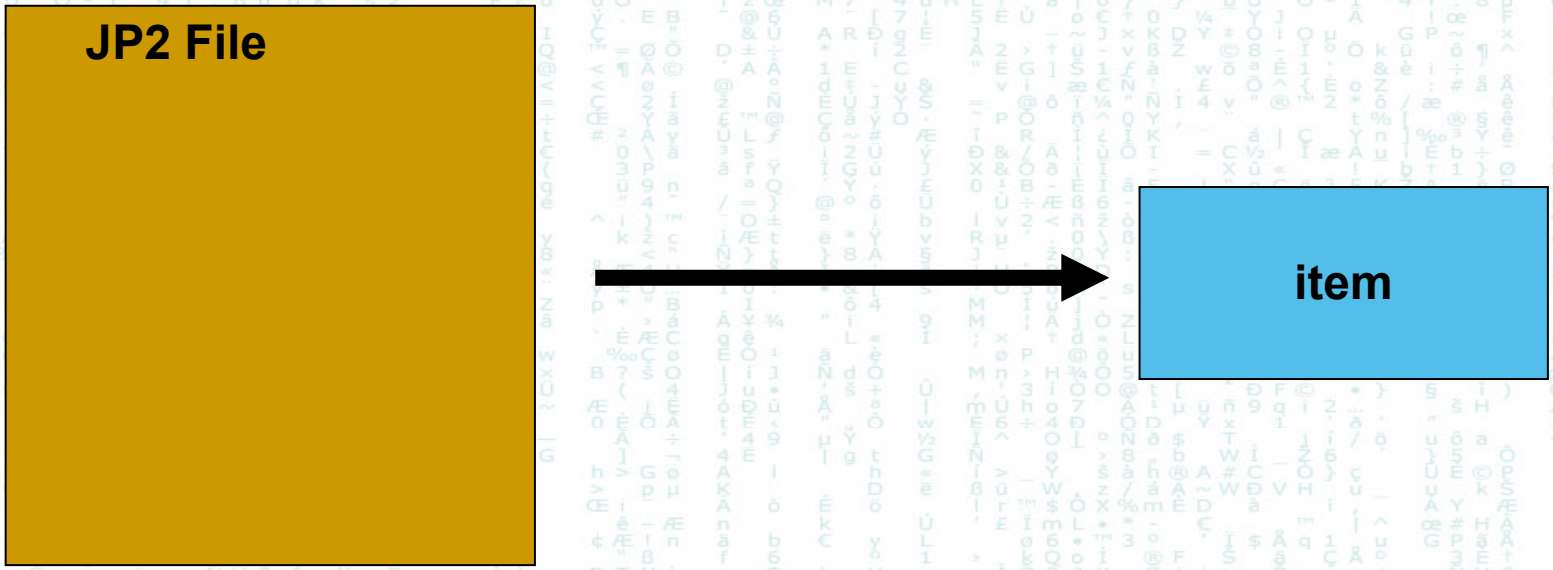
JP2 Architecture...



JP2 files are basically structured as sequences of ‚boxes'.

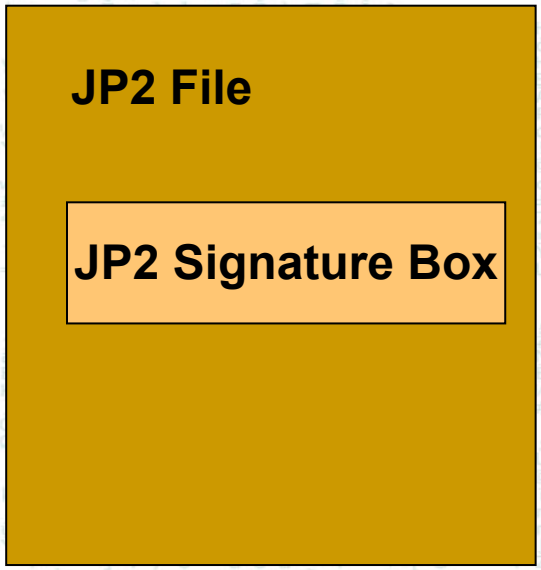
How can we transform this structure into the XCEL structure?

...transformed into XCEL

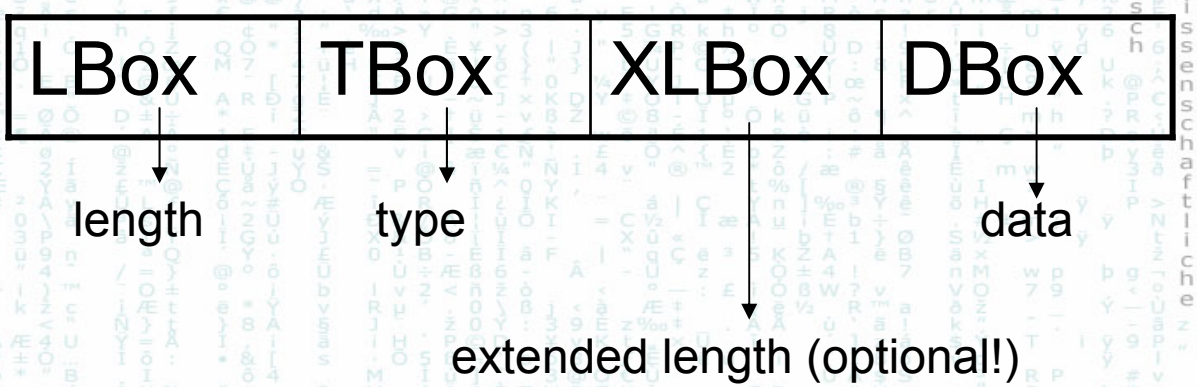


...

```
<formatDescription>
        <item identifier=„JP2RootItem" xsi:type="structuringItem"
order="sequence">
            ...
        </item>
</formatDescription>
...
```



Structure: Box



Box name	Type	Container box	Required?	Notes
JP2 signature box	‚jp\032\032‘	No	Required	This box uniquely identifies the file as a jp2 file

Box name	Type	Container box	Required?	Notes
JP2 signature box	‚jp\032\032‘	No	Required	This box uniquely identifies the file as a jp2 file

…

<formatDescription>

<item identifier=„JP2RootItem" xsi:type="structuringItem" order="sequence">

<item identifier=„JP2SignatureBox“ xsi:type=„structuringItem“ order=„sequence

optional=„false>

…

</item>

…

</item>

</formatDescription>

…

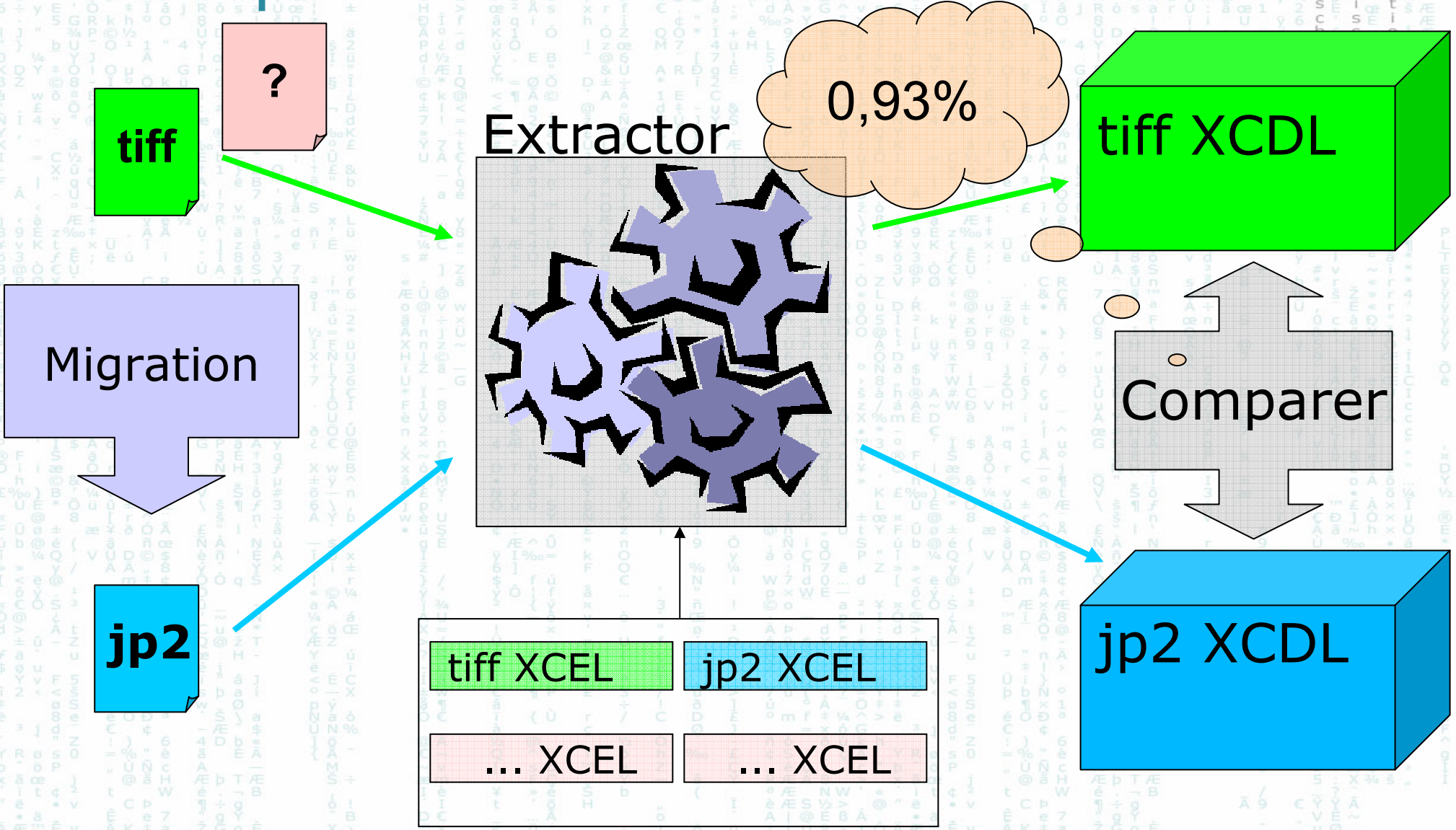
LBox	TBox	DBox

Each field has a length of 4 = 12 Byte

„The type of the JP2 signature box shall be ‘jP\032\032’ (X‘6A501A1A’). The length of this box shall be 12 bytes. The contents of this box shall be the 4-byte character string ‘<CR><LF><X’87’><LF>’ (X‘0D0A870A’). For file verification purposes, this box can be considered a fixed-length 12-byte string which shall have the value: X‘0000 000C 6A50 1A1A 0D0A 870A’“

```
…
<formatDescription>
  <item identifier=„JP2RootItem" xsi:type="structuringItem" order="sequence">
        <item identifier=„JP2SignatureBox“ xsi:type=„structuringItem“ order=„sequence“
optional=„false>
          <property identifier=„JP2Signature" interpretation=„uint8">
          <range>
            <startposition xsi:type="sequential"/>
            <length xsi:type="fixed">12</length>
          </range>
          <value>0 0 0 12 106 80 26 26 13 10 135 10</value>
                <name>signature</name>
      </property>
        </item>
   …
  </item>
</formatDescription>
…
```

XCL in practice



The Output: XCDL

```
...
<formatDescription>
  <item identifier=„JP2RootItem" xsi:type="structuringItem" order="sequence">
   <item identifier=„JP2SignatureBox“ xsi:type=„structuringItem“ order=„sequence“ optional=„false>
            <property identifier=„JP2Signature" interpretation=„uint8">
          <range>
            <startposition xsi:type="sequential"/>
            <length xsi:type="fixed">12</length>
          </range>
          <value>0 0 0 12 106 80 26 26 13 10 135 10</value>
                    <name>signature</name>
        </property>
          </item>
   ...
  </item>
</formatDescription>
...
  ...
  <property id="1" source="raw" cat="descr">
            <name>signature</name>
            <valueSet id="1">
                    <labVal>
                              <val>0 0 0 12 106 80 26 26 13 10 135 10</val>

                <type>uint8</type>
                    </labVal>
                    <dataRef ind="normAll“/>
            </valueSet>
  </property>
  ...
```

```
…
<property id="1" source="raw" cat="descr">
        <name>signature</name>
        <valueSet id=„i1">
                <labVal>
                        <val>0 0 0 12 106 80 26 26 13 10 135 10</val>

        <type>uint8</type>
                </labVal>
                <dataRef ind="normAll“/>
        </valueSet>
</property>
…
<property id=„13" source="raw" cat="descr">
    <name>height</name>
    <valueSet id=„i13" >
        <rawValue>0 0 0 5A</rawValue>
        <labValue>
            <val>95</val>
            <type>uint32</type>
        </labValue>
        <dataRef ind="normAll"/>
    </valueSet>
</property>
…
```

Preservation Capabilities: Robustness

- JPEG2000 has defined *error resilience* as one of its advantageous features
  - Primarily to lower transmission errors
  - Are there any other benefits resulting out of it?

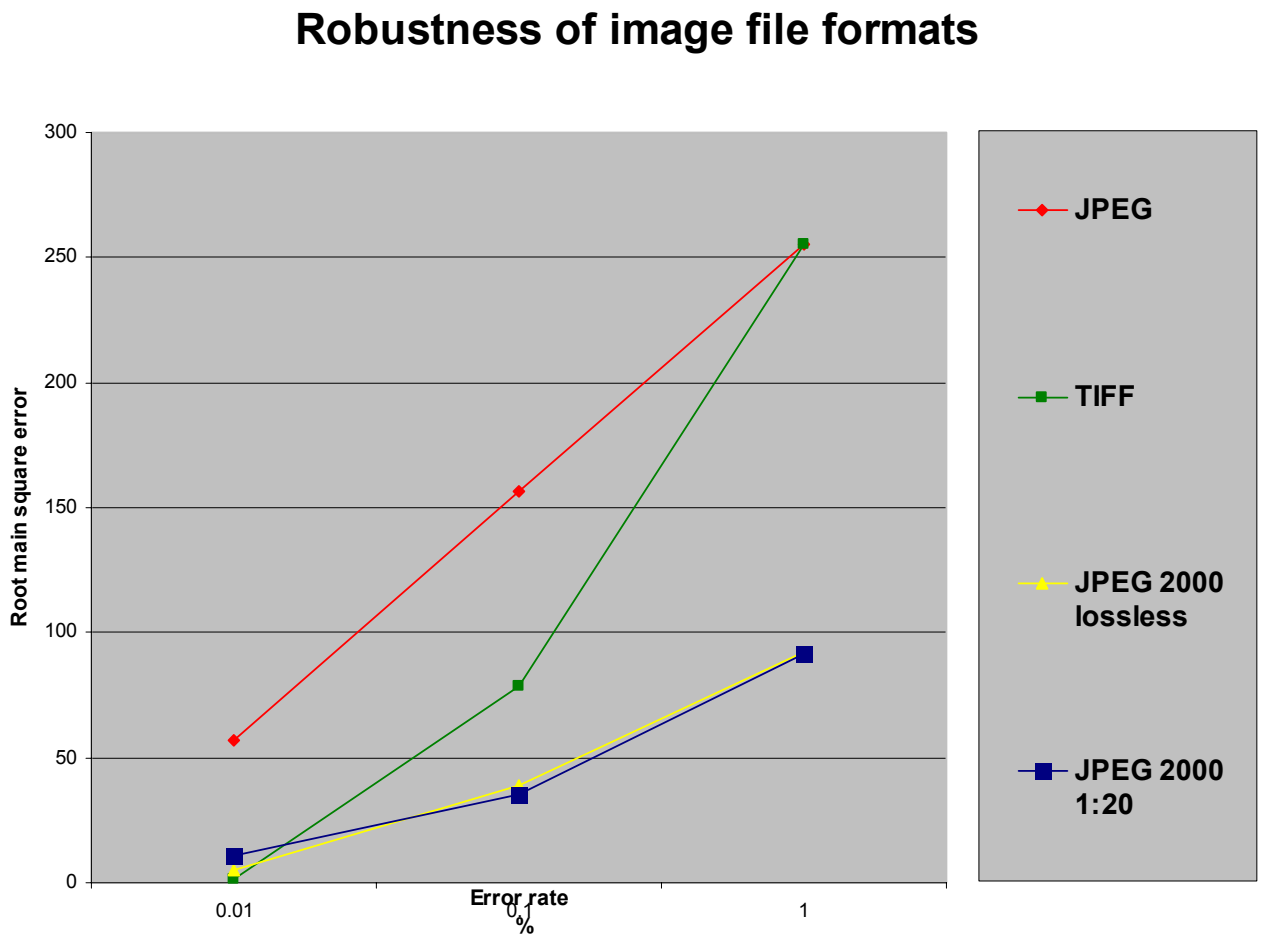
Error resilience

- Error resilience is mainly achieved by means of “markers” within the codestream
- Error resilience at the entropy encoding level:
  - Entropy coding of quantised coefficients within code-blocks
  - Termination of the arithmetic coder and resetting of contexts after each coding pass
  - Lazy coding mode

Error resilience

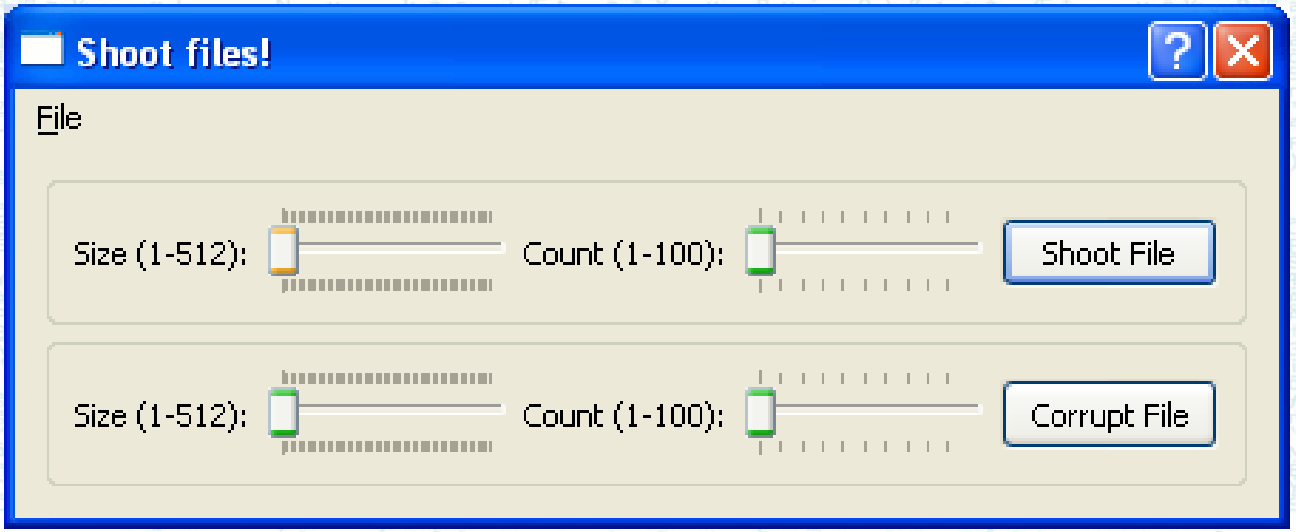
- Error resilience at the packet level:
  - Short packets
  - Resynchronisation markers preceding each single packet in a tile

Side effects?



Source: Dr. Paolo Buonora, Archivio di Stato di Roma,
Centro di Fotoriproduzione degli Archivi di Stato
http://www.cflr.beniculturali.it

Shooting files: Some results



Thank you!

Links:

www.planets-project.eu
www.hki.uni-koeln.de
www.cflr.beniculturali.it