



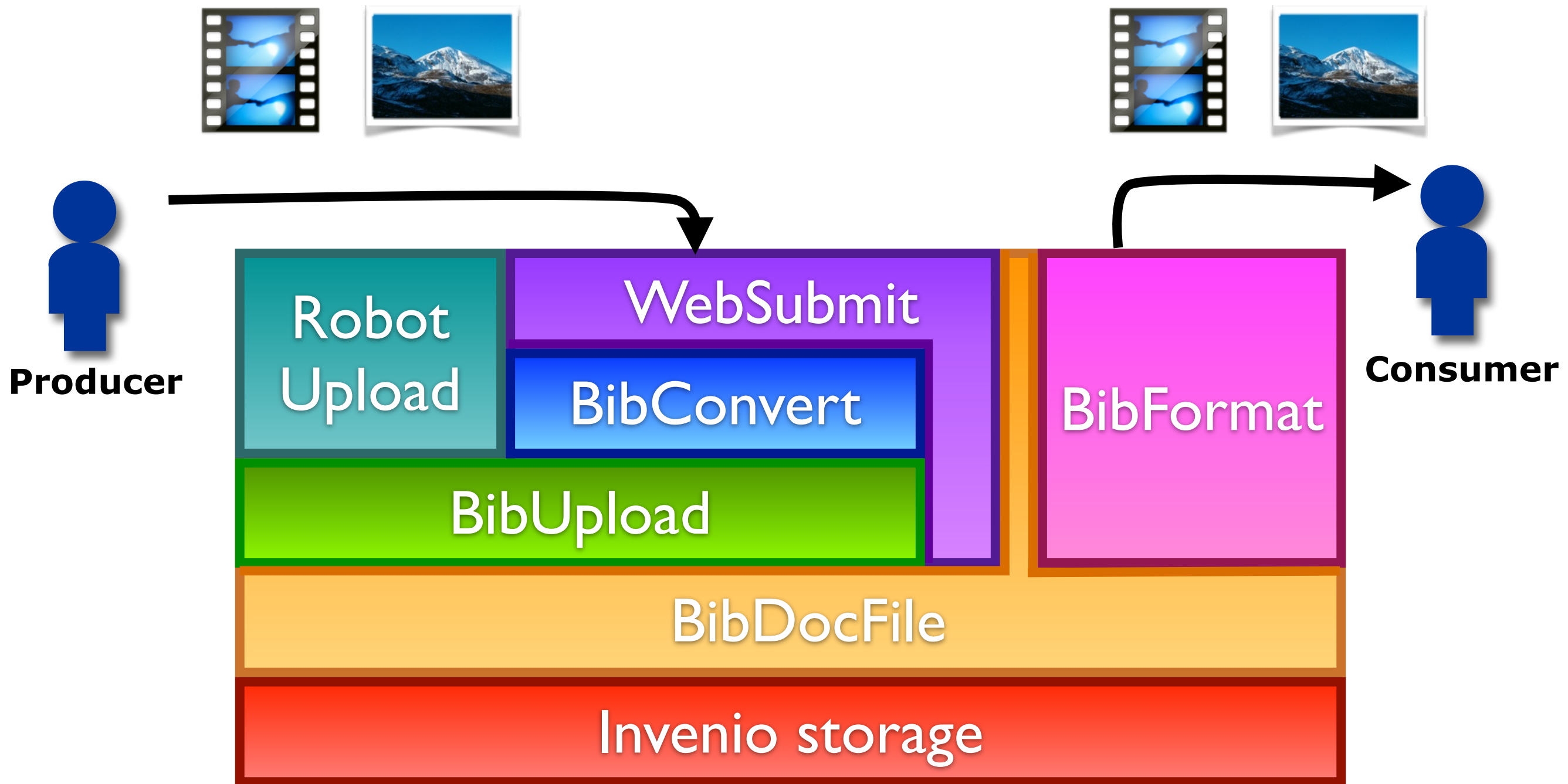
Multimedia Files Management

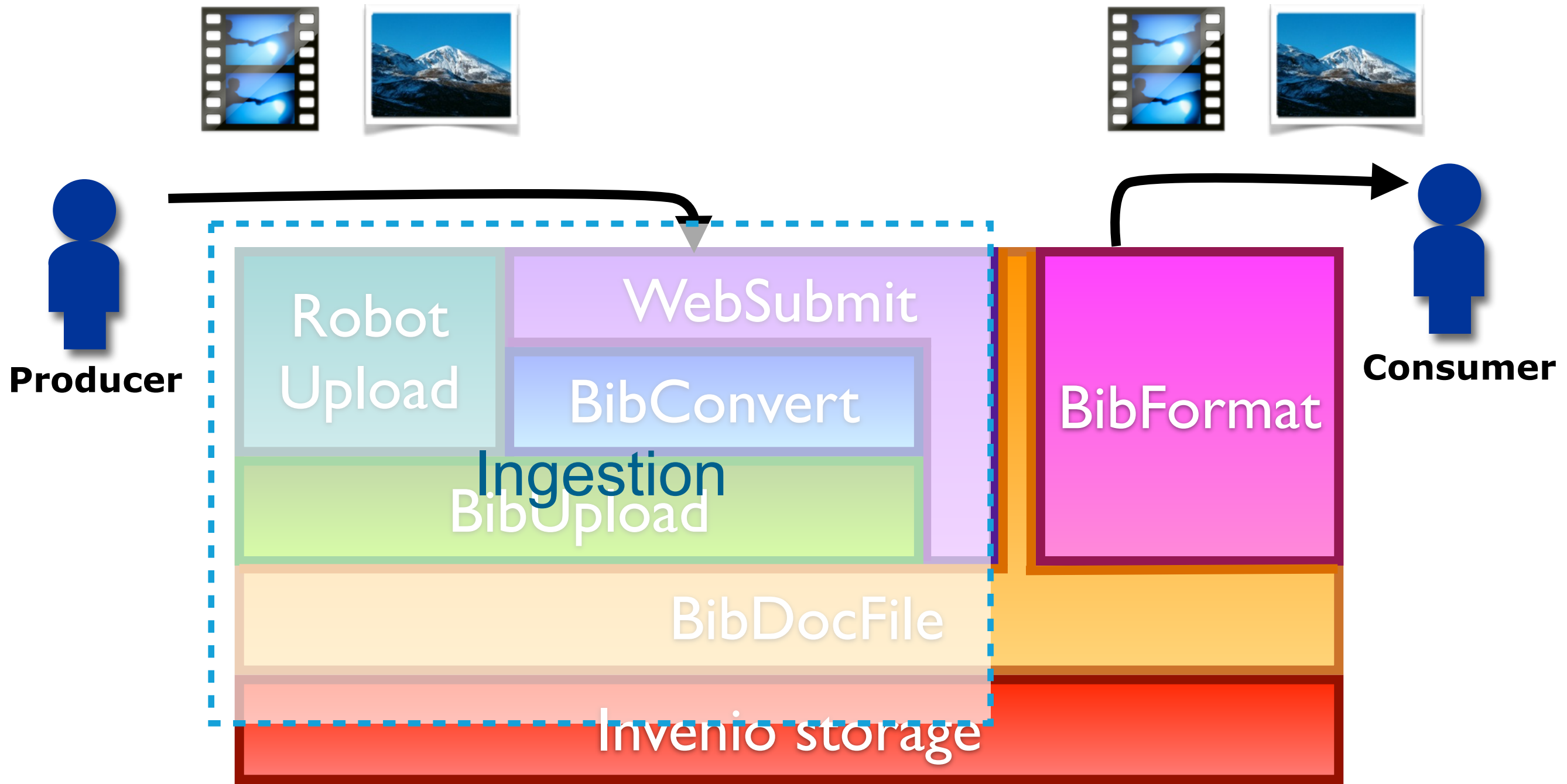
Jérôme Caffaro
CERN

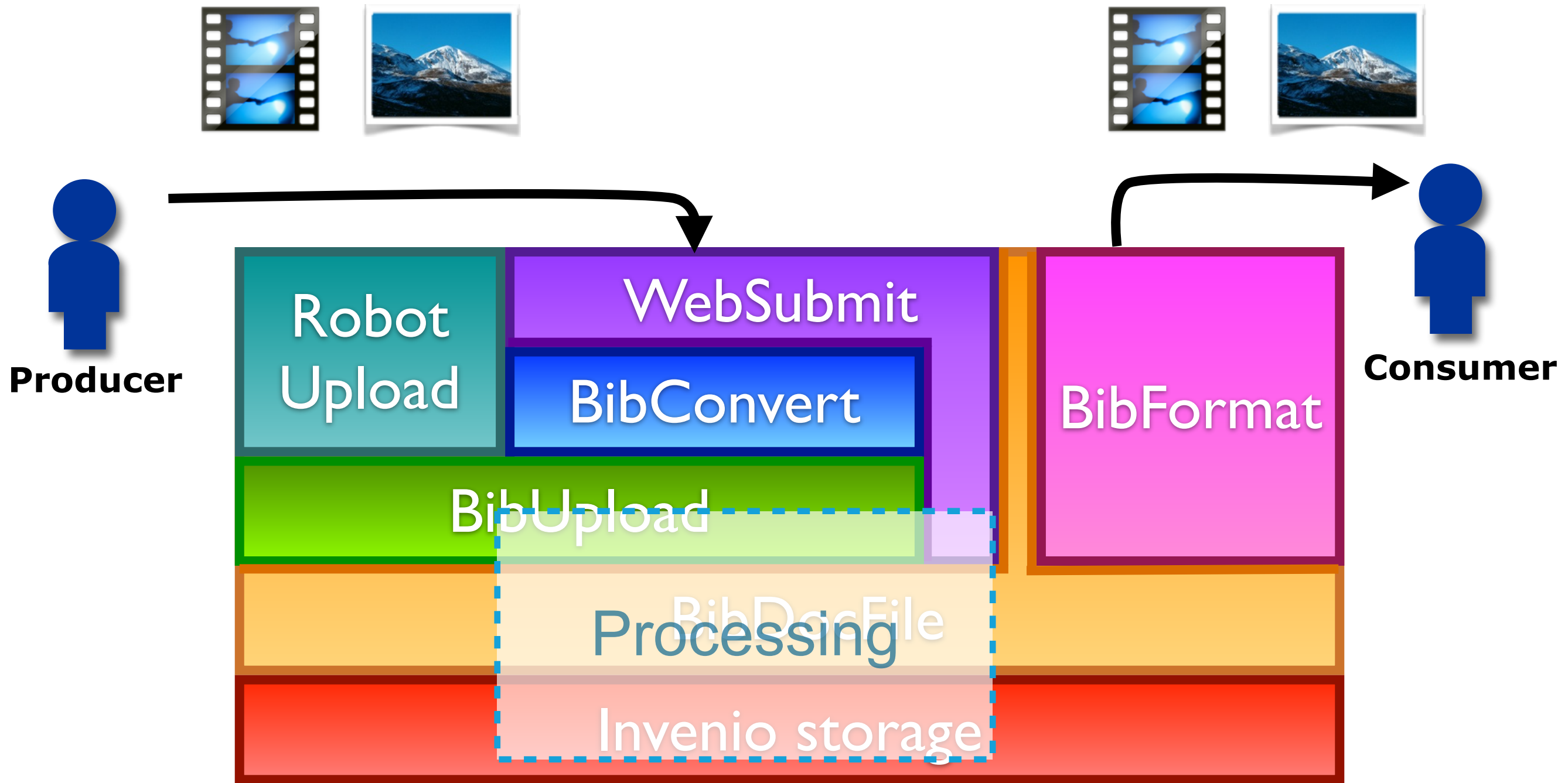
Invenio User Group Workshop
Jülich, Germany
17–20 November 2013

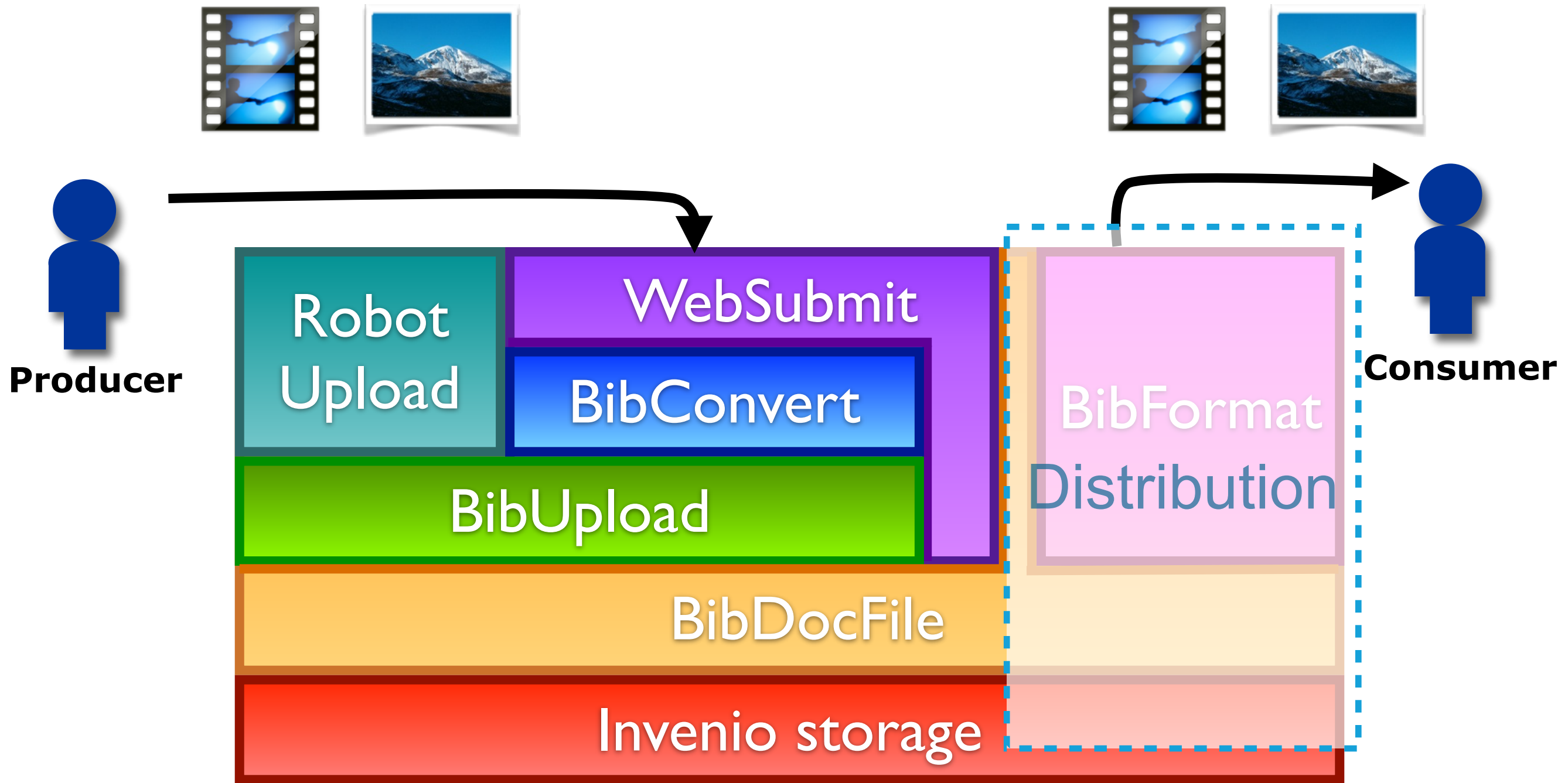
INVENIO) Outline

- 1 Ingestion
- 2 Processing
- 3 Distribution









- document-type agnostic file management stack, but:
 - large files (videos) in large quantities (photo rolls) requires specific upload techniques
 - web-specific ("light") versions must be created
 - specific file distribution (Streaming, Podcasts, etc.)

- document-type agnostic file management stack, but:

- large files (videos) in large photo rolls requires special techniques

- Avoid "select" - "click" - "upload", "select" - "click", etc.
- Upload progress feedback

- web-specific ("light") versions must be created

- specific file distribution (Streaming, Podcasts, etc.)

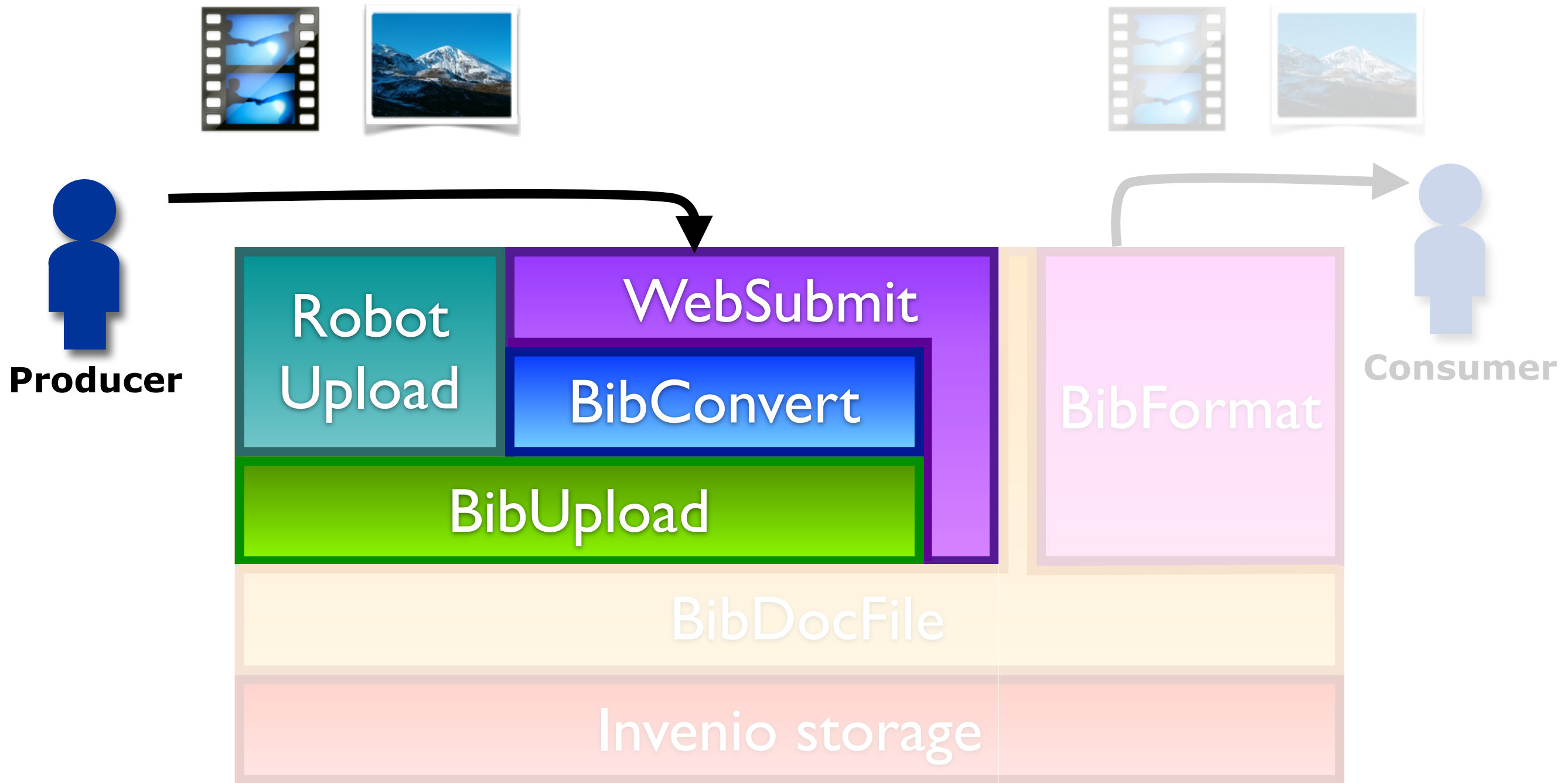
- document-type agnostic file management stack, but:
 - large files (videos) in large quantities (photo rolls) requires specific upload techniques
 - web-specific ("light") version created
 - specific file distribution (Streaming, Podcasts, etc.)

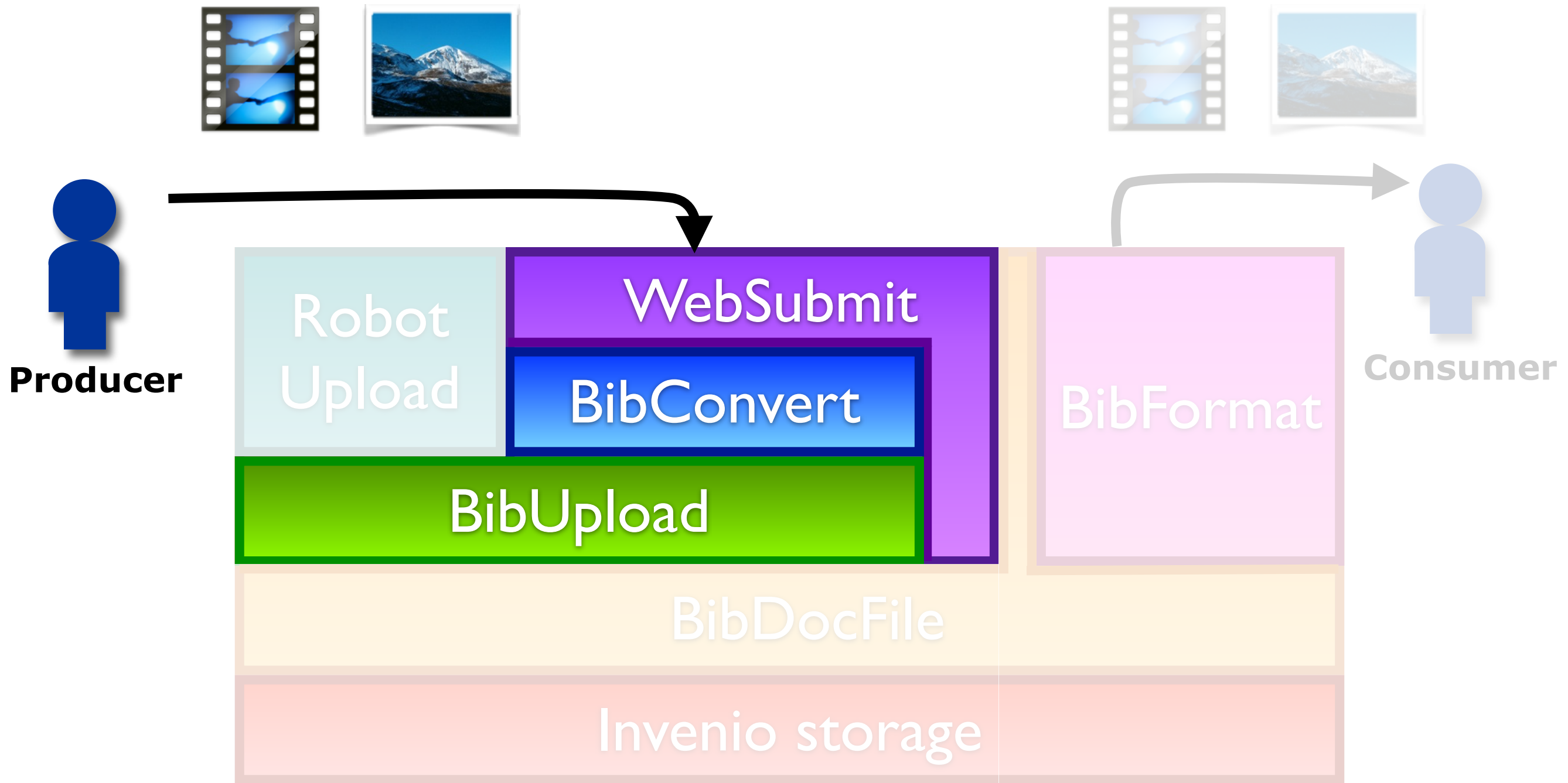
- Generate server-side
- Without overloading the web server
- Automate

- document-type agnostic file management stack, but:
 - large files (videos) in large quantities (photo rolls) requires specific upload techniques
 - web-specific ("light") versions must be created
 - specific file distribution (Podcasts, etc.)
 - Download vs embed
 - Progressive download vs stream

1

Ingestion





INVENIO) » Multimedia Submission

INVENIO) » Multimedia Submission






Photos:

INVENIO) » Multimedia Submission



Photos:

		
<input type="text" value="Flower 1"/>	<input type="text" value="Flower 2"/>	<input type="text"/>



Browse

Upload_Photos
Response element

INVENIO) » Multimedia Submission



Photos:

Flower 1 Flower 2

Browse

*Upload_Photos
Response element*



Videos:

INVENIO) » Multimedia Submission



Photos:

Flower 1 Flower 2 []

Browse

Upload_Photos
Response element



Videos:

Upload Video Finish Submission

CERN-MOVIE-2011-060.... (1518.76MB) - 9%

"BibEncode"
Response element

INVENIO) » Multimedia Submission



Photos:

Flower 1 Flower 2

Browse

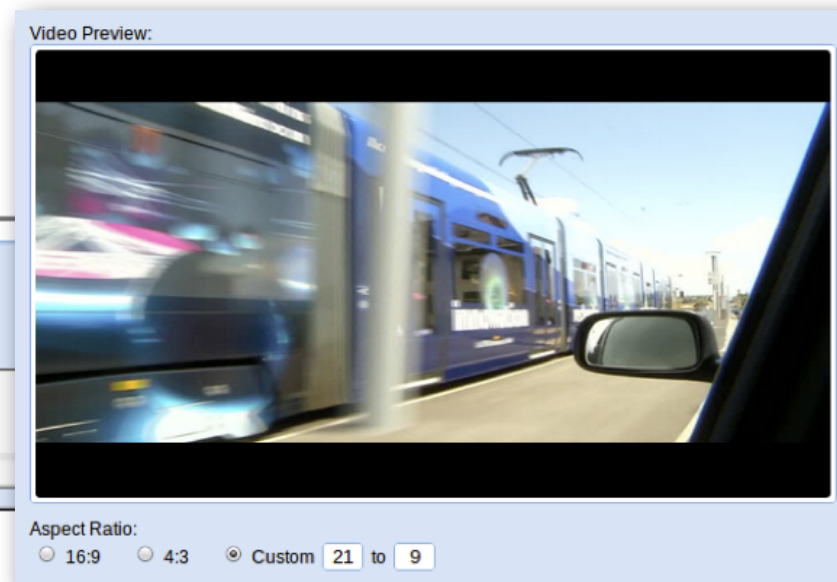
*Upload_Photos
Response element*



Videos:

Upload Video

CERN-MOVIE-2011-060.... (1518.76MB) - 9%



*"BibEncode"
Response element*

INVENIO) » Multimedia Submission



Photos:

Flower 1 Flower 2 []

[]
Browse

*Upload_Photos
Response element*



Videos:

Upload Video

CERN-MOVIE-2011-060.... (1518.76MB) - 9%



Not yet fully read-to-use response element: currently include complete submission form

*"BibEncode"
Response element*

INVENIO) » Multimedia Submission



Photos:

Flower 1 Flower 2

Browse

*Upload_Photos
Response element*



Videos:

Upload Video

CERN-MOVIE-2011-060.... (1518.76MB) - 9%

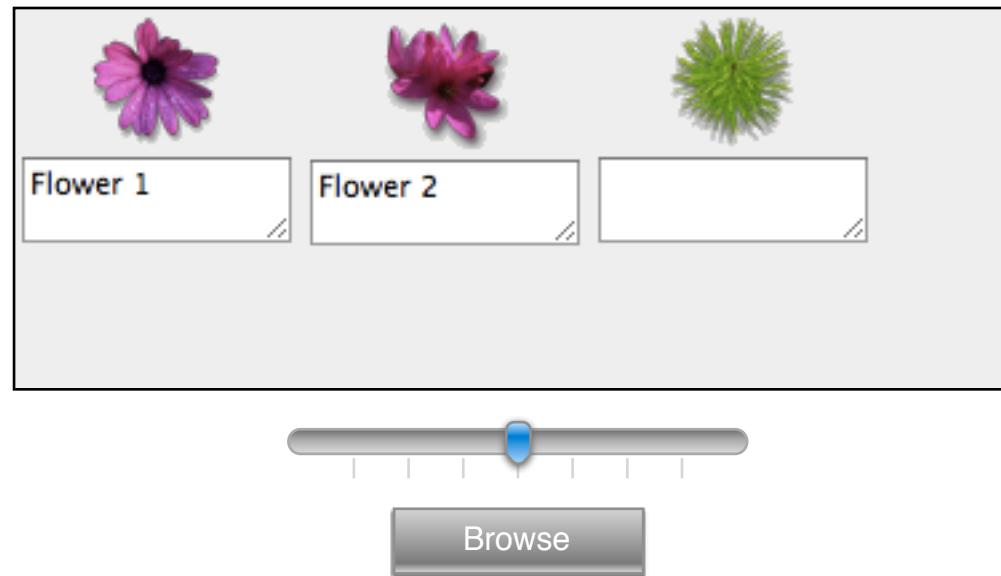


Not yet fully read-to-use response element: currently include complete submission form

*"BibEncode"
Response element*

INVENIO) » Multimedia Submission

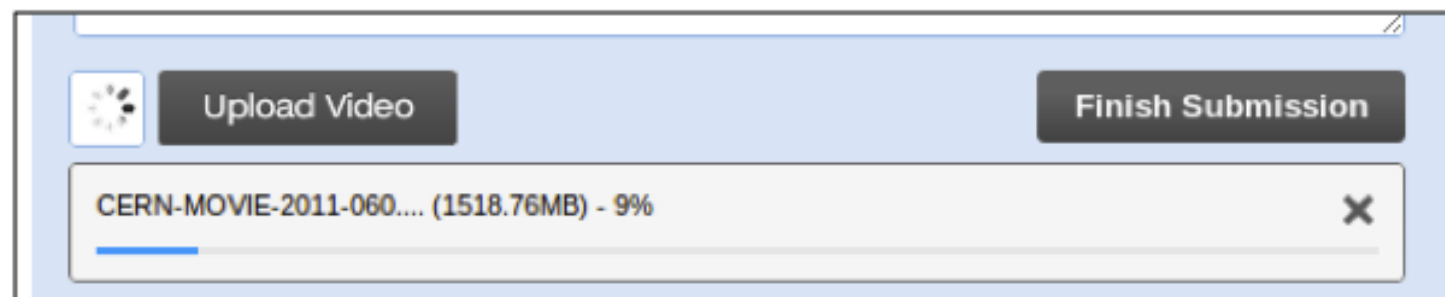
response element



function

+

Move_Photos_to_Storage



+

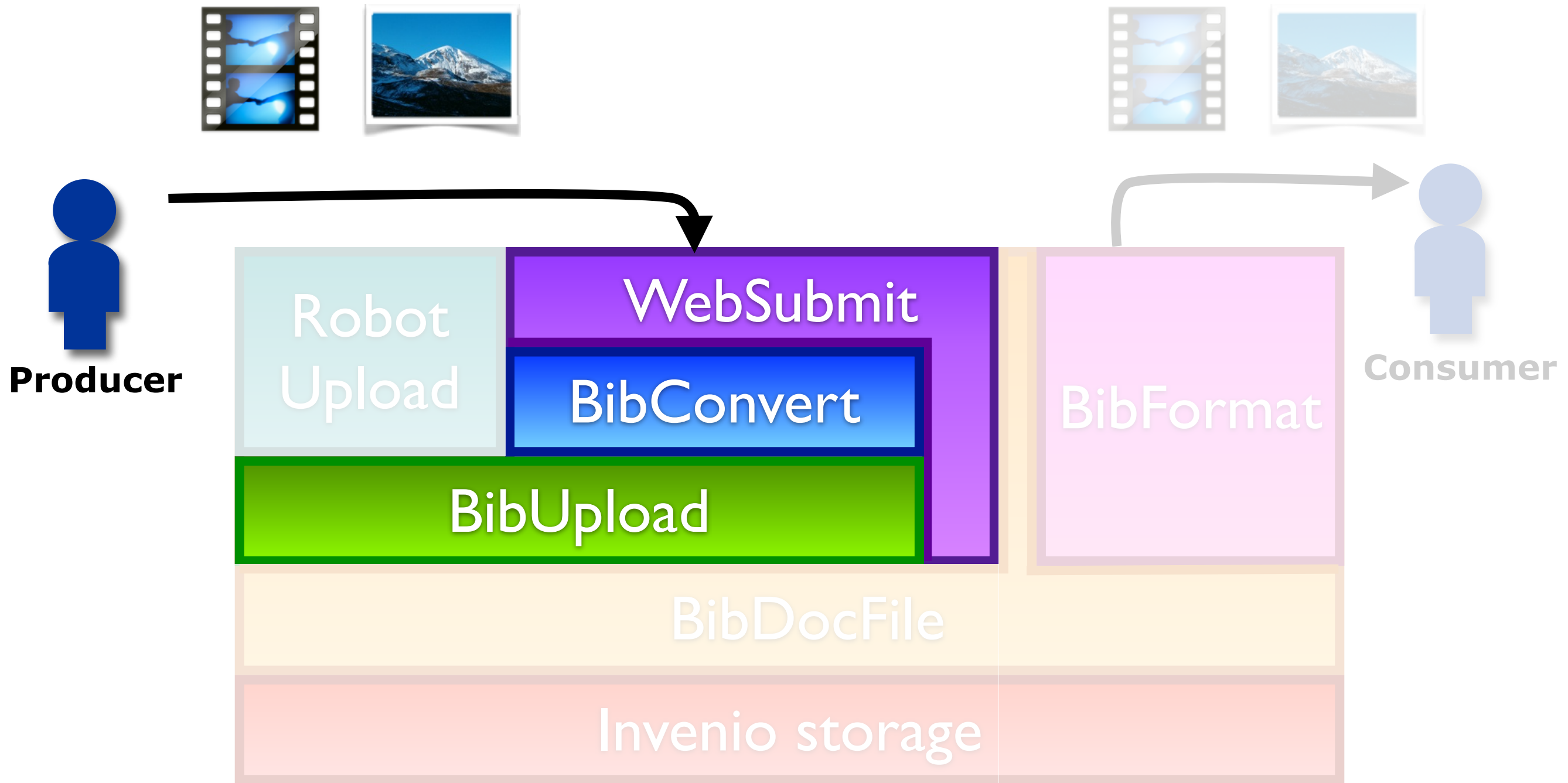
Video_Processing

See: <http://invenio-demo.cern.ch/help/admin/websubmit-admin-guide#5.6>

Demo

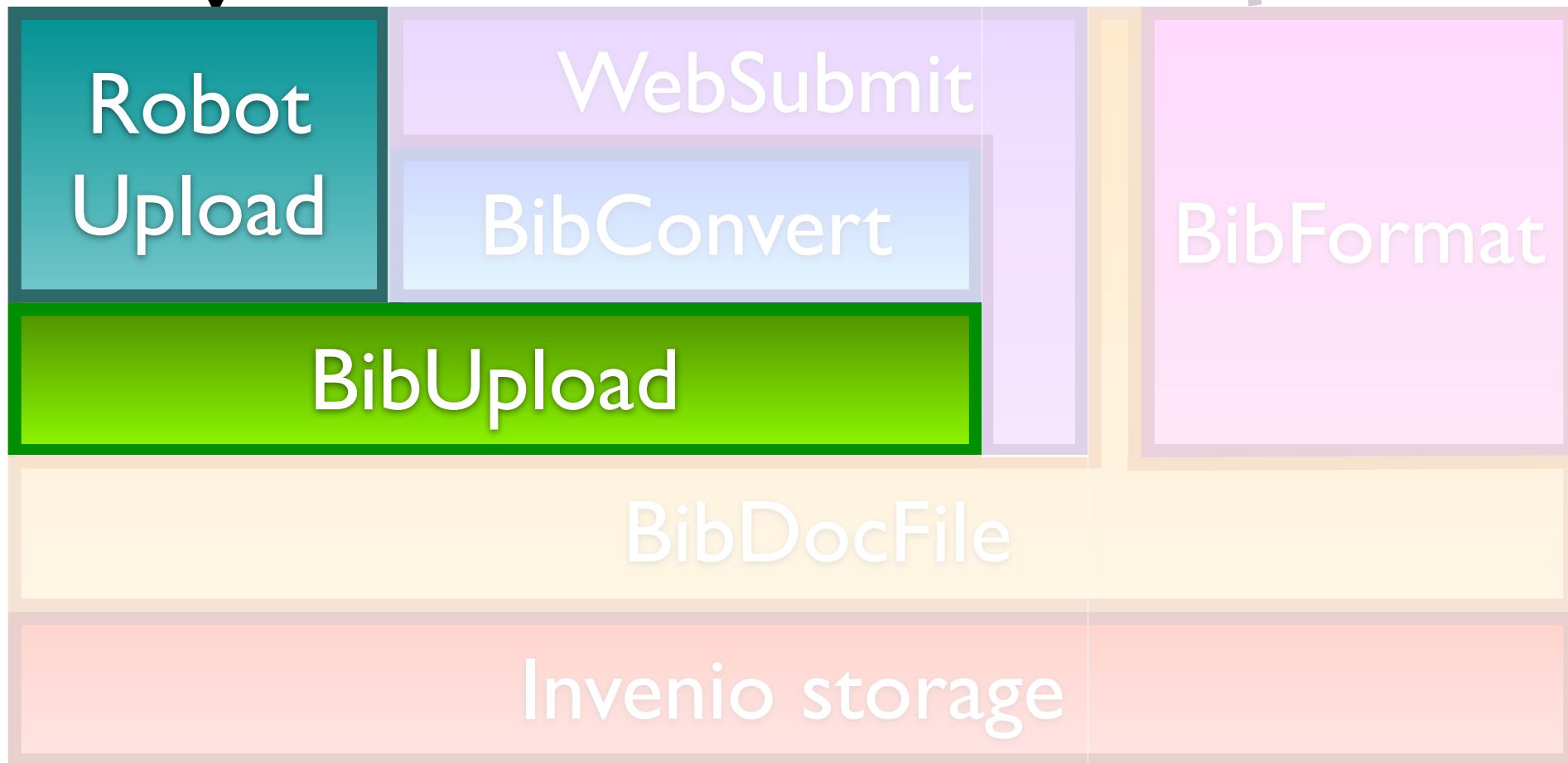
INVENIO) » Shortcomings

- **photos**
 - re-ordering of photos not yet supported
- **videos**
 - first iteration only of BibEncode
 - « upload » response element not yet well isolated from other parts of the form





Producer



Consumer

INVENIO) » RobotUpload

- REST API to push records
- provides seamless integration with existing tools (systems, desktop applications, etc)
 - external professional creation tools, such as Final Cut Server, LightRoom, etc. through their plugin/export infrastructure
 - custom institutional systems, scripts

INVENIO) » RobotUpload

INVENIO) » RobotUpload

```
$ less my_video.xml
```

INVENIO) » RobotUpload

```
$ less my_video.xml
```

```
<record>
  <datafield tag="245" ind1=" " ind2=" ">
    <subfield code="a">My Title</subfield>
  </datafield>
  <datafield tag="980" ind1=" " ind2=" ">
    <subfield code="a">VIDEO</subfield>
  </datafield>
  <datafield tag="FFT" ind1=" " ind2=" ">
    <subfield code="a">http://invenio-software.org/download/invenio-demo-site-
files/CERN-MOVIE-2010-075.mpg;master</subfield>
    <subfield code="f">mpg</subfield>
    <subfield code="n">CERN-MOVIE-2010-075</subfield>
    <subfield code="t">Main</subfield>
  </datafield>
</record>
```


INVENIO) » RobotUpload

```
$ curl -T my_video.xml \  
http://serv.er/batchuploader/robotupload/insert?  
callback_url=http://anotherserver/feedback \  
-A invenio_webupload \  
-H "Content-Type: application/marcxml+xml"
```


INVENIO) » RobotUpload

```
$ curl -T my_video.xml \  
http://serv.er/batchuploader/robotupload/insert?  
callback_url=http://anotherserver/feedback \  
-A invenio_webupload \  
-H "Content-Type: application/marcxml+xml"
```

```
$ curl -T my_video.xml \  
http://serv.er/batchuploader/robotupload/insert?  
&nonce=1234&callback_url=http://anotherserver/  
feedback \  
-A invenio_webupload \  
-H "Content-Type: application/marcxml+xml"
```

INVENIO) » RobotUpload

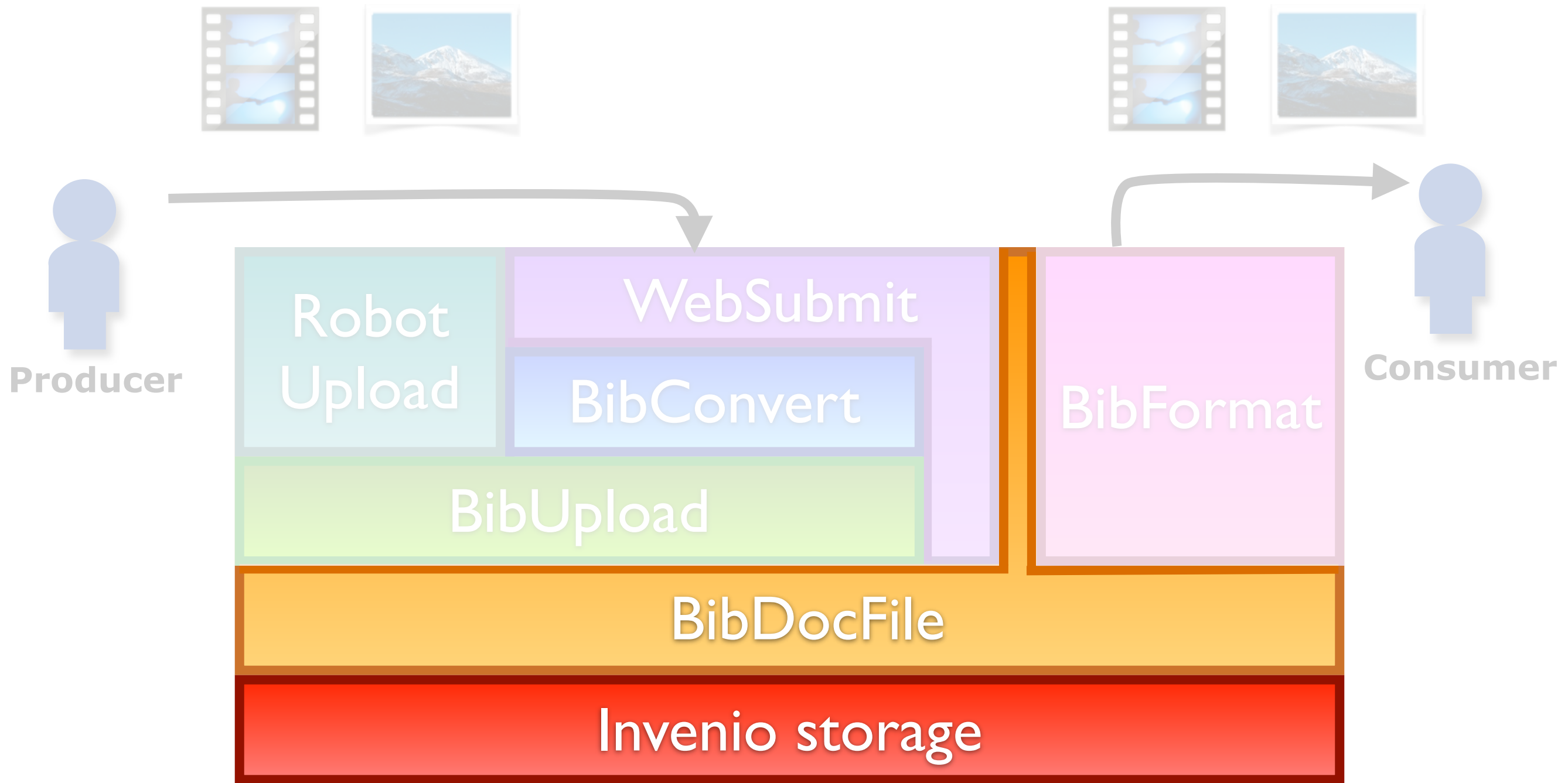
- daemon mode available too: look for jobs deposited (eg. via *scp*) in directory `CFG_BATCHUPLOADER_DAEMON_DIR`:
 - Metadata
`/opt/invenio/var/batchupload/metadata/{append|correct|insert|replace}`
 - Files
`/opt/invenio/var/batchupload/documents/{append|revise}`
Matching by *report number* or *recid*

see <http://invenio-demo.cern.ch/help/admin/bibupload-admin-guide>

Demo

2

Processing



INVENIO » Processing

- sub-formats support: one BibDoc can have several formats (`.jpg`, `.png`, `.tiff`), and each of these can have several sub-formats
(`.jpg;icon-180`, `.jpg;icon-640`, `.jpg;icon-1024`)
- metadata extraction and re-use (indexing, embedding, etc.)
- `bst_create_icons` tasklet (images)
- BibEncode (videos)



Icon creation tasklet



- creates web optimized (slaves) version of images
 - watches for missing images in a given collection
- runs offline, in BibSched
 - tasklet: script running as BibTask



- creates web optimized (slaves) version of images
 - watches for missing images in a given collection
- runs offline, in BibSched
 - tasklet: script running as BibTask

```
$ /opt/invenio/bin/bibtasklet -T bst_create_icon \  
-a collection=Pictures \  
-a icon_sizes=180,700,1024 \  
-s 5m
```



- creates web optimized (slaves) version of images
 - watches for missing images in a given collection
- runs offline, in BibSched
 - tasklet: script running as BibTask

```
$ /opt/invenio/bin/bibtasklet -T bst_create_icon \  
-a collection=Pictures \  
-a icon_sizes=180,700,1024 \  
-s 5m
```

ID	PROC [PRI]	USER	RUNTIME	SLEEP	STATUS	HOST	PROGRESS
14	bibrank	admin	2013-11-16 14:53:19	5m	WAITING		Will execute: '/opt/invent
15	bibsort	admin	2013-11-16 14:53:24	5m	WAITING		Will execute: '/opt/invent
11	bibindex	admin	2013-11-16 14:53:29	5m	WAITING		Will execute: '/opt/invent
18	bibencode:daemon	admin	2013-11-16 14:53:34	5m	WAITING		Will execute: '/opt/invent
90	bibtasklet	admin	2013-11-16 14:57:27	5m	RUNNING	jerome	Done 7 out of 9.
13	webcoll	admin	2013-11-16 14:59:50	5m	WAITING		Will execute: '/opt/invent
12	bibreformat	admin	2013-11-16 15:08:14	20m	WAITING		Will execute: '/opt/invent

Automatic Mode [A Manual] [1/2/3 Display] [P Purge] [l/L Log] [O Opts] [E Edit motd] [Q Quit] 2013-11-16 14:58:57



ATLANTIS INSTITUTE OF FICTIVE SCIENCE admin :: [logout](#)

[Search](#)
[Submit](#)
[Personalize ▾](#)
[Help](#)
[Administration ▾](#)

[Home](#) > [Multimedia & Arts](#) > [Pictures](#) > [European Molecular Biology Conference](#) > [Access to Fulltext](#)

[Information](#)
[Discussion](#)
[Usage statistics](#)

European Molecular Biology Conference - CERN-HI-6902127

Main file(s):



6902127

	6902127.gif (icon)	[17.34 KB]	11 Nov 2013, 15:27
	6902127.jpeg	[285.15 KB]	11 Nov 2013, 15:27
version 1	6902127.jpg (icon-1024)	[284.98 KB]	16 Nov 2013, 14:58
	6902127.jpg (icon-180)	[20.06 KB]	16 Nov 2013, 14:58
	6902127.jpg (icon-700)	[145.44 KB]	16 Nov 2013, 14:58

[Similar records](#)



- both for WebSubmit-ingested images AND batch/robotuploaded images
- coming soon...
- WebSubmit function `Move_Photos_to_Storage` to schedule icon creation immediately



BibEncode



- encodes videos in BibSched
 - multi-nodes support
 - daemon waiting to collect new encoding jobs
- support for metadata extraction and enrichments
- sub-formats support
- optimized handling of large files
- support for re-processing existing videos

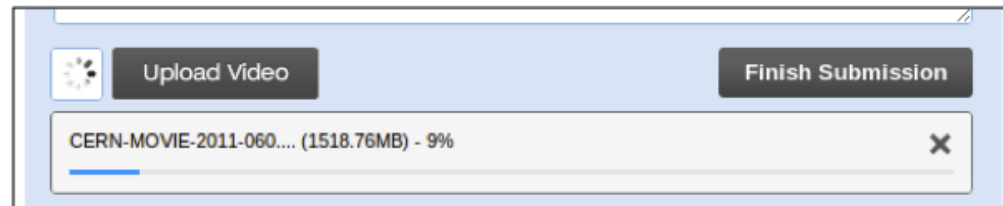


Dependencies:

- **ffmpeg**, with following options enabled
(defined in `CFG_BIBENCODE_FFMPEG_CONFIGURATION_REQUIRED`):
`--enable-gpl --enable-version3 --enable-nonfree --enable-libfaac
--enable-libtheora --enable-libvorbis --enable-libvpx --enable-
libx264`
- **mediainfo, ffprobe**
- *path to the above binaries set in:*
`CFG_PATH_FFMPEG, CFG_PATH_FFPROBE, CFG_PATH_MEDIAINFO`



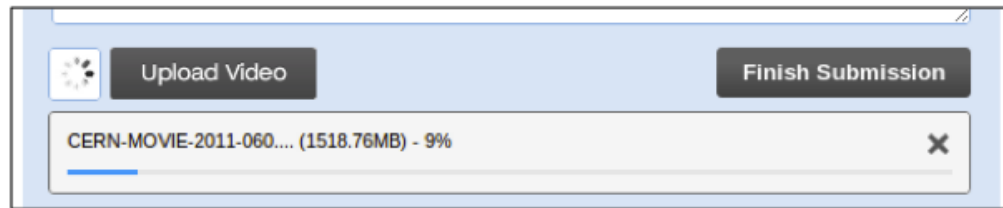
DEMOVID_SINGLE Response element





DEMOVID_SINGLE
Response element

function

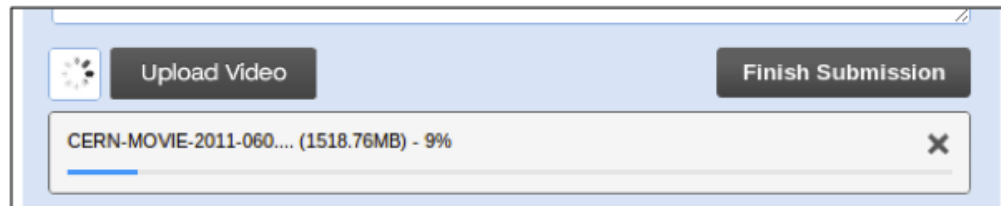


+

Video_Processing



DEMOVID_SINGLE Response element



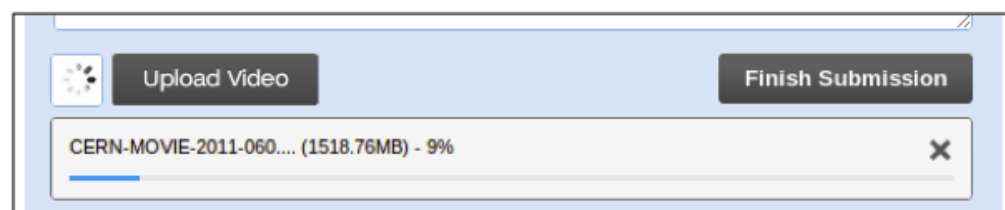
+

Video_Processing

Configure to use an encoding template



DEMOVID_SINGLE Response element



+

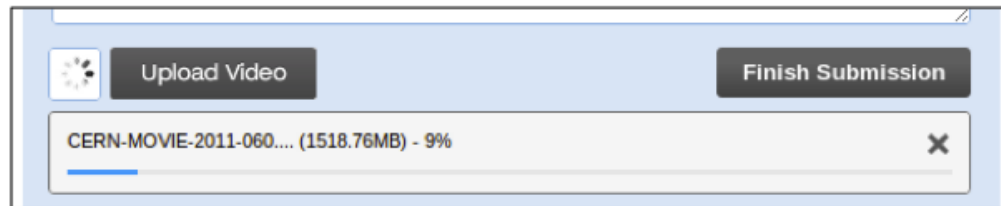
Video_Processing

Configure to use an
encoding template

```
$ /opt/invenio/bin/bibencode -m daemon -s 5m
```



DEMOVID_SINGLE Response element



+

Video_Processing

Configure to use an encoding template

```
$ /opt/invenio/bin/bibencode -m daemon -s 5m
```

ID	PROC [PRI]	USER	RUNTIME	SLEEP
76	bibencode:daemon	admin	2012-05-06 20:31:01	5m

Automatic Mode [A Manual] [1/2/3 Display] [P Purge] [L/L Log] [0 Opts]

INVENIO) »



Batch template

```
$ less /opt/invenio/etc/bibencode/batch_template_submission.json
```



```
$ less /opt/invenio/etc/bibencode/batch_template_submission.json
// Batch Job Template Example for a fresh submission
{
  // Defines that this JSON document describes a batch job
  "isbatch": true,

  // Defines if the master video (input file) should be appended to the record
  "add_master": true,

  // Defines the bibdoc name for the master and all slaves, neces. for the demo
  // websubmit workflow because we use temporary filenames
  "bibdoc_master_docname": "DEMOVIDEO",

  // Defines bibdoc extension for the master, necessary for the demo
  // websubmit workflow because we use temporary filenames without extension
  "bibdoc_master_extension": "mp4",

  "jobs": [
    {
      "mode": "encode",
      "profile": "WEBM_480P",
      "bibdoc_subformat": "480p"
    },
    [...]
  ]
}
```



```
$ less /opt/invenio/etc/bibencode/batch_template_submission.json
// Batch Job Template Example for a fresh submission
{
  // Defines that this JSON document describes a batch submission
  "isbatch": true,

  // Defines if the master video (input file) should be added to the output
  "add_master": true,

  // Defines the bibdoc name for the master and all slaves, neces. for the demo
  // websubmit workflow because we use temporary filenames
  "bibdoc_master_docname": "DEMOVIDEO",

  // Defines bibdoc extension for the master, necessary for the demo
  // websubmit workflow because we use temporary filenames without extension
  "bibdoc_master_extension": "mp4",

  "jobs": [
    {
      "mode": "encode",
      "profile": "WEBM_480P",
      "bibdoc_subformat": "480p"
    },
    [...]
  ]
}
```

You probably will have to update these arguments for your version of *ffmpeg*



```
$ less /opt/invenio/etc/bibencode/batch_template_submission.json
// Batch Job Template Example for a fresh submission
{
  // Defines that this JSON document describes a batch job
  "isbatch": true,

  // Defines if the master video (input file) should be appended to the record
  "add_master": true,

  // Defines the bibdoc name for the master and all slaves, neces. for the demo
  // websubmit workflow because we use temporary filenames
  "bibdoc_master_docname": "DEMOVIDEO",

  // Defines bibdoc extension for the master, neces. for the demo
  // websubmit workflow because we use temporary filenames
  "bibdoc_master_extension": "mp4",

  "jobs": [
    {
      "mode": "encode",
      "profile": "WEBM_480P",
      "bibdoc_subformat": "480p"
    },
    [...]
  ]
}
```

Currently if metadata extraction fails, you might want to set:

```
'assure_quality': false
'add_master_metadata': true
```




```
$ less /opt/invenio/etc/bibencode/batch_template_submission.json
// Batch Job Template Example for a fresh submission
{
  // Defines that this JSON document describes a batch job
  "isbatch": true,

  // Defines if the master video (input file) should be appended to the record
  "add_master": true,

  // Defines the bibdoc name for the master and all slaves, neces. for the demo
  // websubmit workflow because we use temporary filenames
  "bibdoc_master_docname": "DEMOVIDEO",

  // Defines bibdoc extension for the master, necessary for the demo
  // websubmit workflow because we use temporary filenames without extension
  "bibdoc_master_extension": "mp4",

  "jobs": [
    {
      "mode": "encode",
      "profile": "WEBM_480P",
      "bibdoc_subformat": "480p"
    },
    [...]
  ]
}
```

Define list of encoding jobs, referring to encoding profiles (next slide)



```
$ less /opt/invenio/etc/bibencode/batch_template_submission.json
// Batch Job Template Example for a fresh submission
{
  // Defines that this JSON document describes a batch job
  "isbatch": true,

  // Defines if the master record should be appended to the record
  "add_master": true,

  // Defines the bibdoc name, slaves, neces. for the demo
  // we submit workflow because we use temporary filenames
  "bibdoc_master_docname": "DEMOVIDEO",

  // Defines bibdoc extension for the master, necessary for the demo
  // we submit workflow because we use temporary filenames without extension
  "bibdoc_master_extension": "mp4",

  "jobs": [
    {
      "mode": "encode",
      "profile": "WEBM_480P",
      "bibdoc_subformat": "480p"
    },
    [...]
  ]
}
```

Video_Processing function will use this default template if no other specified

INVENIO) »



Profiles

```
$ less /opt/invenio/etc/bibencode/encoding_profiles_local.json
```



```
$ less /opt/invenio/etc/bibencode/encoding_profiles_local.json
```

```
"MP4_480P": {
  "videocodec": "libx264",
  "audiocodec": "libfaac",
  "videobitrate": "1000k",
  "audiobitrate": "128k",
  "width": 854,
  "height": 480,
  "passes": 1,
  "extension": "mp4",
  "special": "-vpre fast -vpre baseline -ac 2"
},
"WEBM_720P": {
  "videocodec": "libvpx",
  "audiocodec": "libvorbis",
  "videobitrate": "2500k",
  "audiobitrate": "128k",
  "width": 1280,
  "height": 720,
  "passes": 1,
  "extension": "webm",
  "special": "-ac 2 -g 320 -qmax 63"
},
```

You probably will have to update these arguments for your version of *ffmpeg*



```
$ less /opt/invenio/etc/bibencode/encoding_profiles_local.json
```

```
"MP4_480P": {
  "videocodec": "libx264",
  "audiocodec": "libfaac",
  "videobitrate": "1000k",
  "audiobitrate": "128k",
  "width": 854,
  "height": 480,
  "passes": 1,
  "extension": "mp4",
  "special": "-vpre fast -vpre baseline -ac 2"
},
"WEBM_720P": {
  "videocodec": "libvpx",
  "audiocodec": "libvorbis",
  "videobitrate": "2500k",
  "audiobitrate": "128k",
  "width": 1280,
  "height": 720,
  "passes": 1,
  "extension": "webm",
  "special": "-ac 2 -g 320 -qmax 63"
},
```



- `Video_Processing` function schedules a job according to template set for its parameter `batch_template`
- job file is created in directory `CFG_BIBENCODE_DAEMON_DIR_NEWJOBS`
(`/opt/invenio/var/tmp-shared/bibencode/jobs`)
- BibEncode daemon periodically processes jobs in this directory
- consumed jobs are moved to `CFG_BIBENCODE_DAEMON_DIR_OLDJOBS`
(`/opt/invenio/var/tmp-shared/bibencode/jobs/done`)



- metadata enrichment enabled in batch template
- « batch » mode exists too:
 - watch for records missing the slaves in a given collection

See: <http://cds.cern.ch/record/1450898>



- metadata enrichment enabled in batch template
- « batch » mode exists too:
 - watch for records missing the slaves in a given collection

```
$ /opt/invenio/bin/bibencode -m batch \  
-collection="Videos" \  
-i /opt/invenio/etc/bibencode/my_batch_template.json  
-s 5m
```




- metadata enrichment enabled in batch template
- « batch » mode exists too:
 - watch for records missing the slaves in a given collection

```
$ /opt/invenio/bin/bibencode -m batch \  
-collection="Videos" \  
-i /opt/invenio/etc/bibencode/my_batch_template.json \  
-s 5m
```

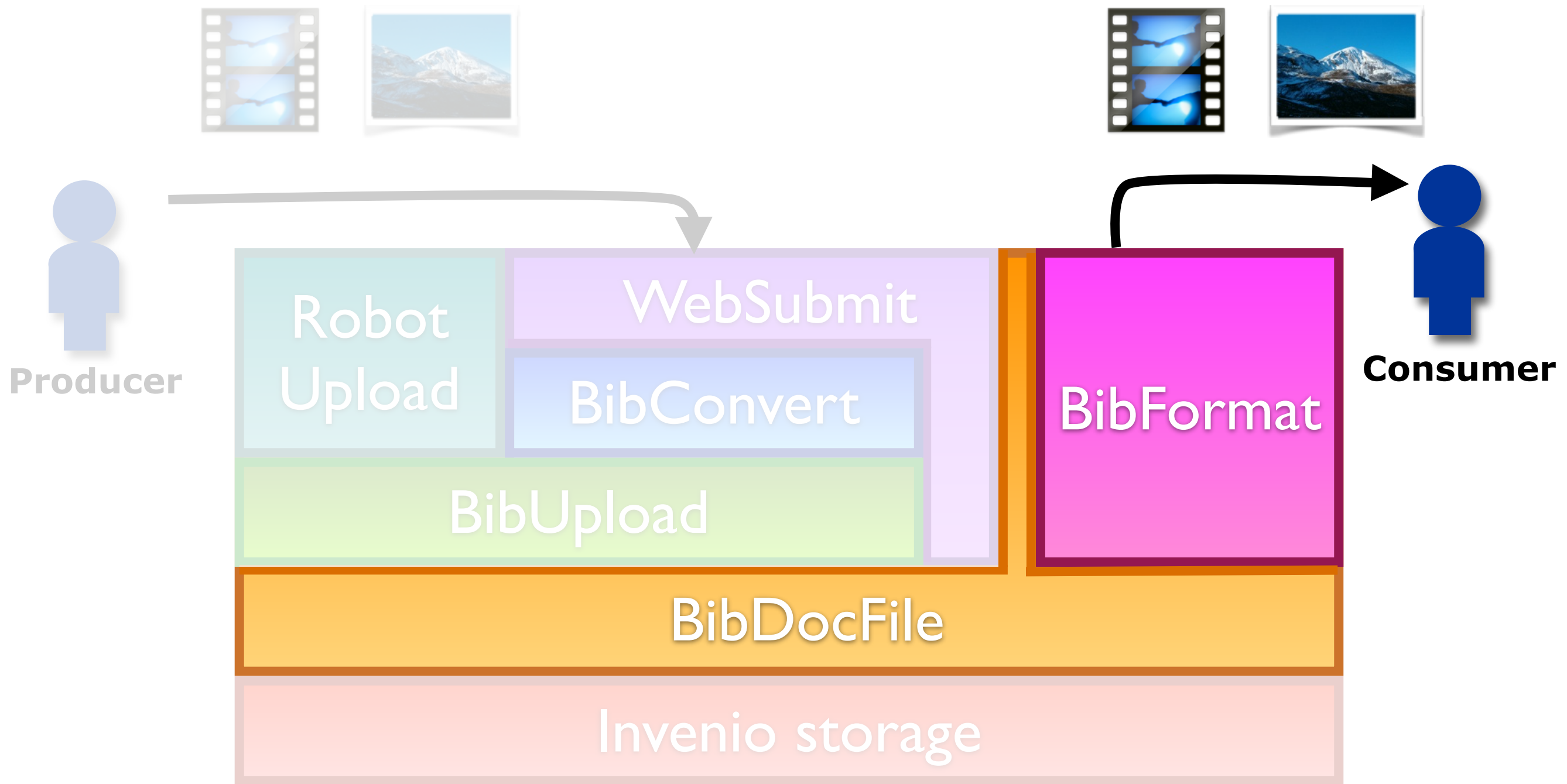
- alternative:
 - create jobs via an external service

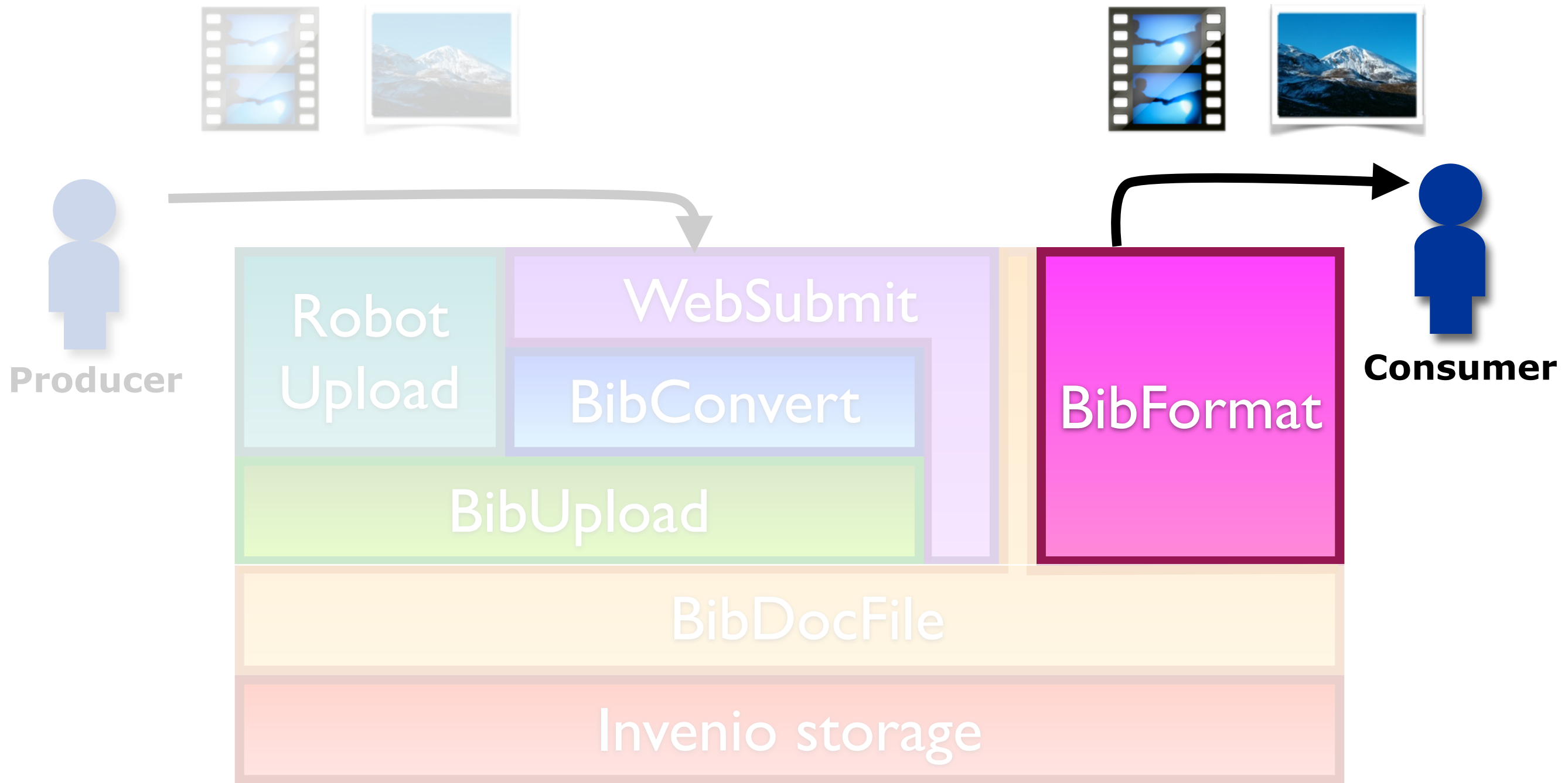
See: <http://cds.cern.ch/record/1450898>

Demo

3

Distribution





- a few handful default resources
 - `Picture_HTML_*.bft`, `bfe_photos`
 - `Video_HTML_*.bft`, `bfe_video_*`
 - using `MedialEment.js`
- customize, use external libraries as needed

INVENIO » Formats / Export

- podcast (*xr*) output format, `podcast.xml`
- meta (*hdm*) output format,
`default_html_meta.bft`, `bfe_meta*`
 - adds meta tags to page headers
 - OpenGraph, GoogleScholar, Twitter Card

INVENIO) » Formats / Export

Swisscom 19:07 88 %

← Tweet 🔍 ✍️

 **CMS Publications**
@CMSpapers 

[#CMSEXO](#) 12-051: Search for Jet Extinction in the Inclusive Jet pT spectrum at $\sqrt{s}=8\sim\text{TeV}$: cds.cern.ch/record/1545286
[#CMSPAS](#)

06.05.13 10:50



 **CERN Document Server**

Search for Jet Extinction in the Inclusive Jet pT spectrum at...

A search for the extinction of QCD jet production is presented using data collected with the CMS detector corresponding to 10.7 \$





Accueil Connecter Découvrir Moi

 **Jim Boulton**
@jim_boulton 

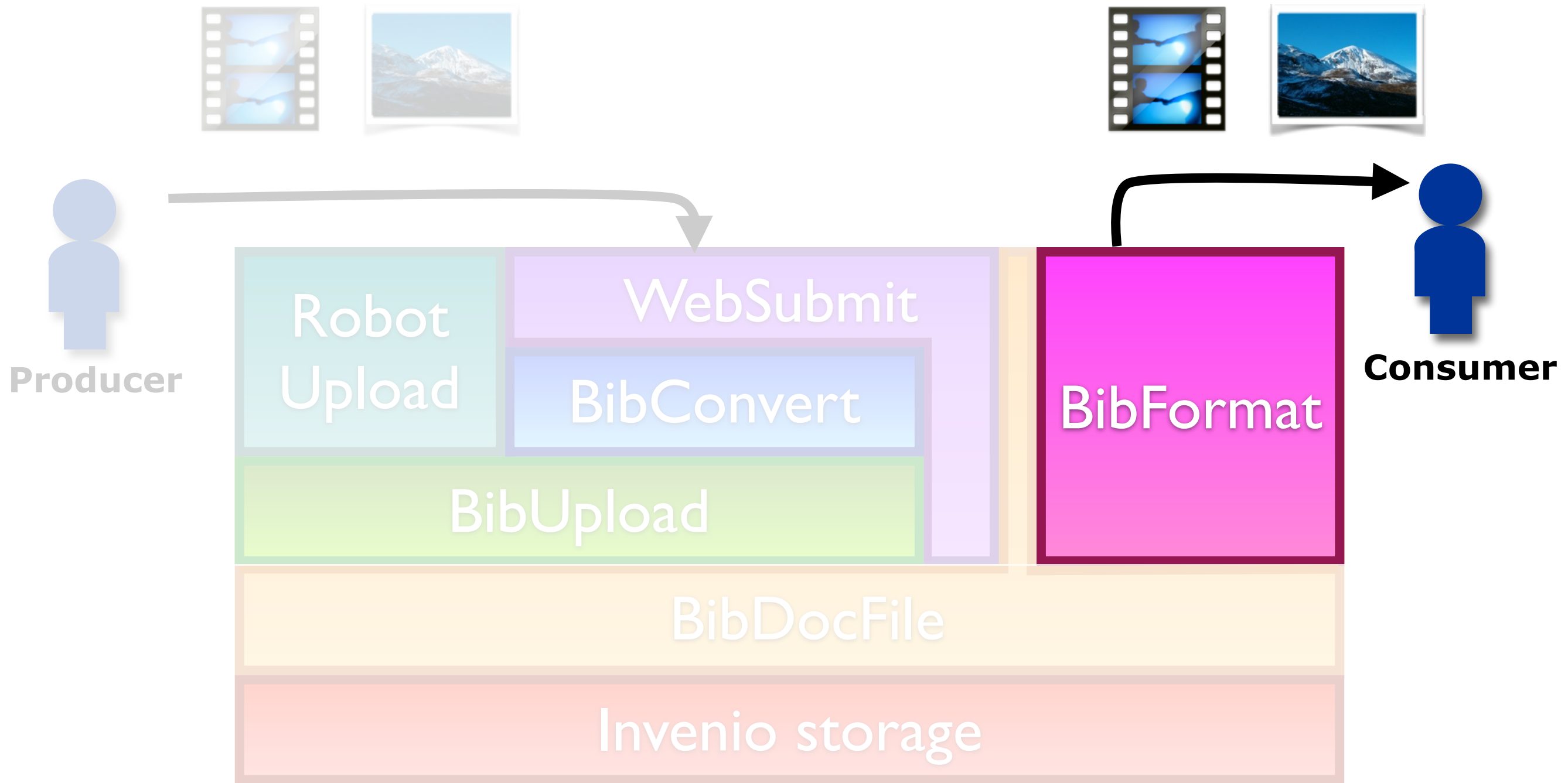
CERN service TBLs NeXT, the first web server, in 2009 cds.cern.ch/record/1547556#

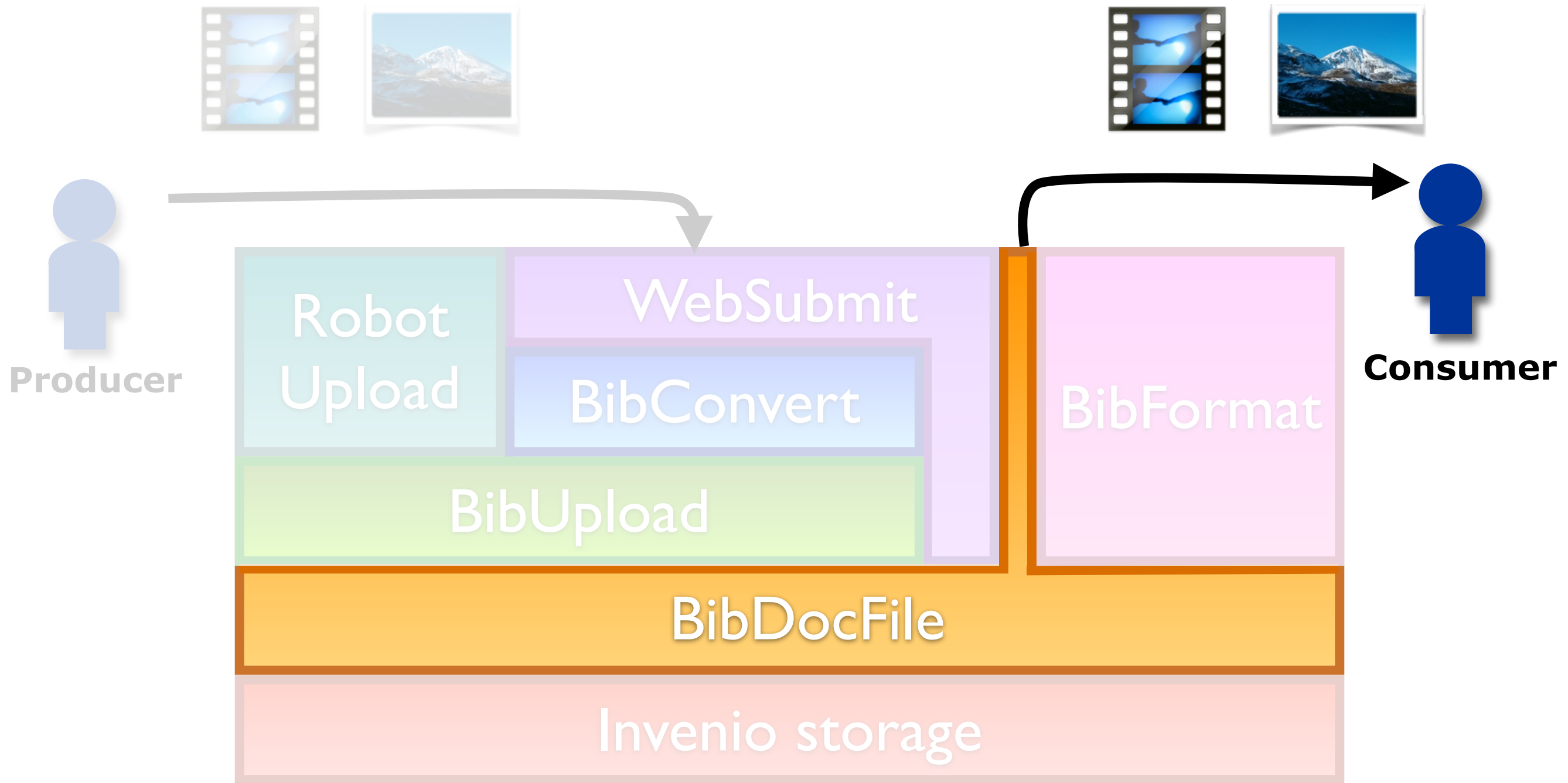
10:46 AM - 27 May 2013



 **CERN Document Server @CERN_CDS**

← ↻ ★





- load-balancing; maximize bandwidth
- use mod_xsendfile
`CFG_BIBDOCFILE_USE_XSENDFILE = 1`
- HTTP-live streaming, or other streaming technologies?

That's all folks!