

Smart Content Management



Martin Pengelly-Phillips
CTO, ftrack

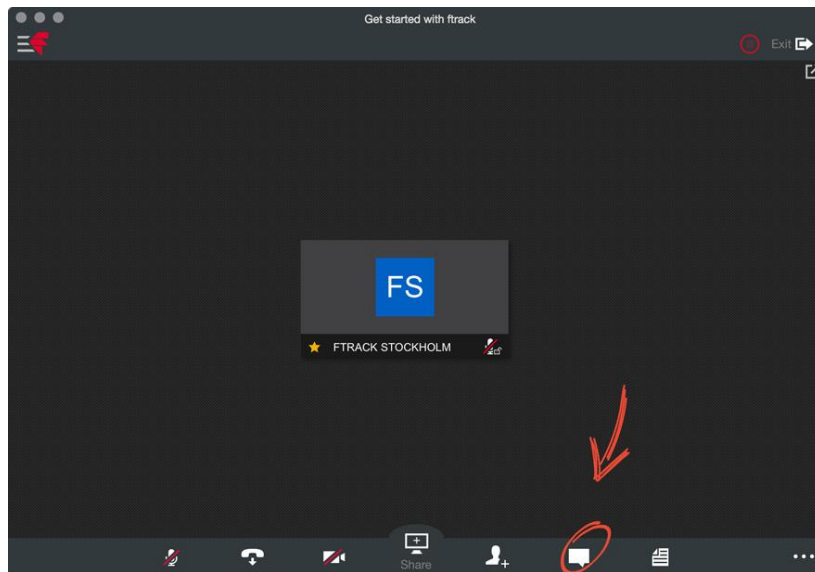
martin@ftrack.com

- Introduction
- What content?
- Traditional content management
- Smart content management
- What's next?
- Q&A

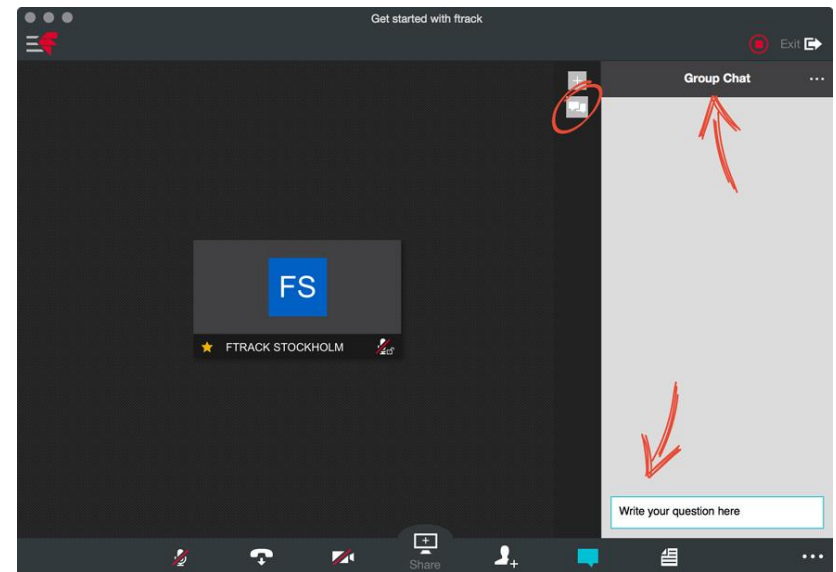
How to submit questions



Open Chat



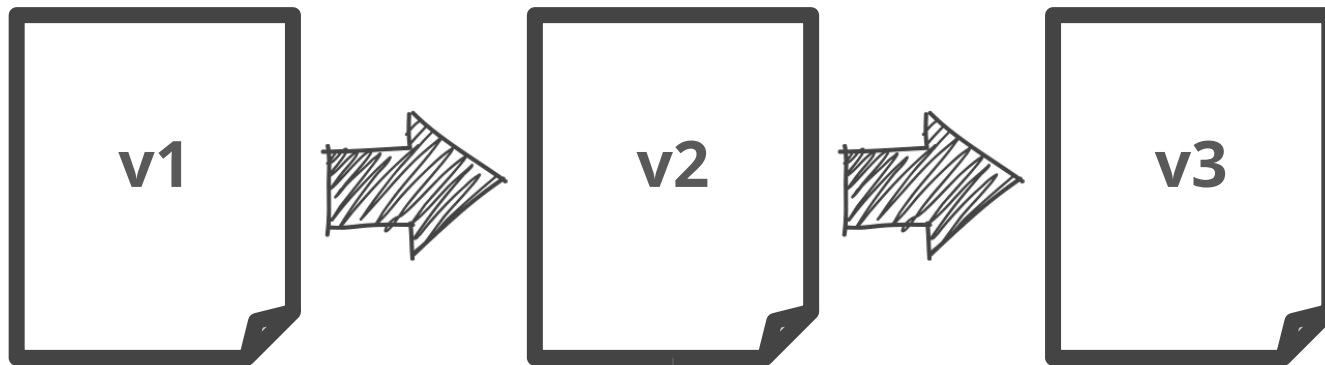
Submit questions to Group Chat



There will be a Q&A session at the end of the webinar

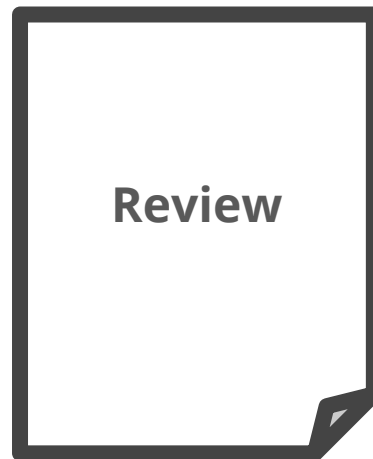
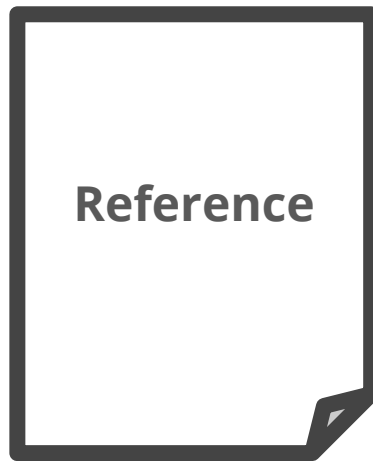
What content?

What content?



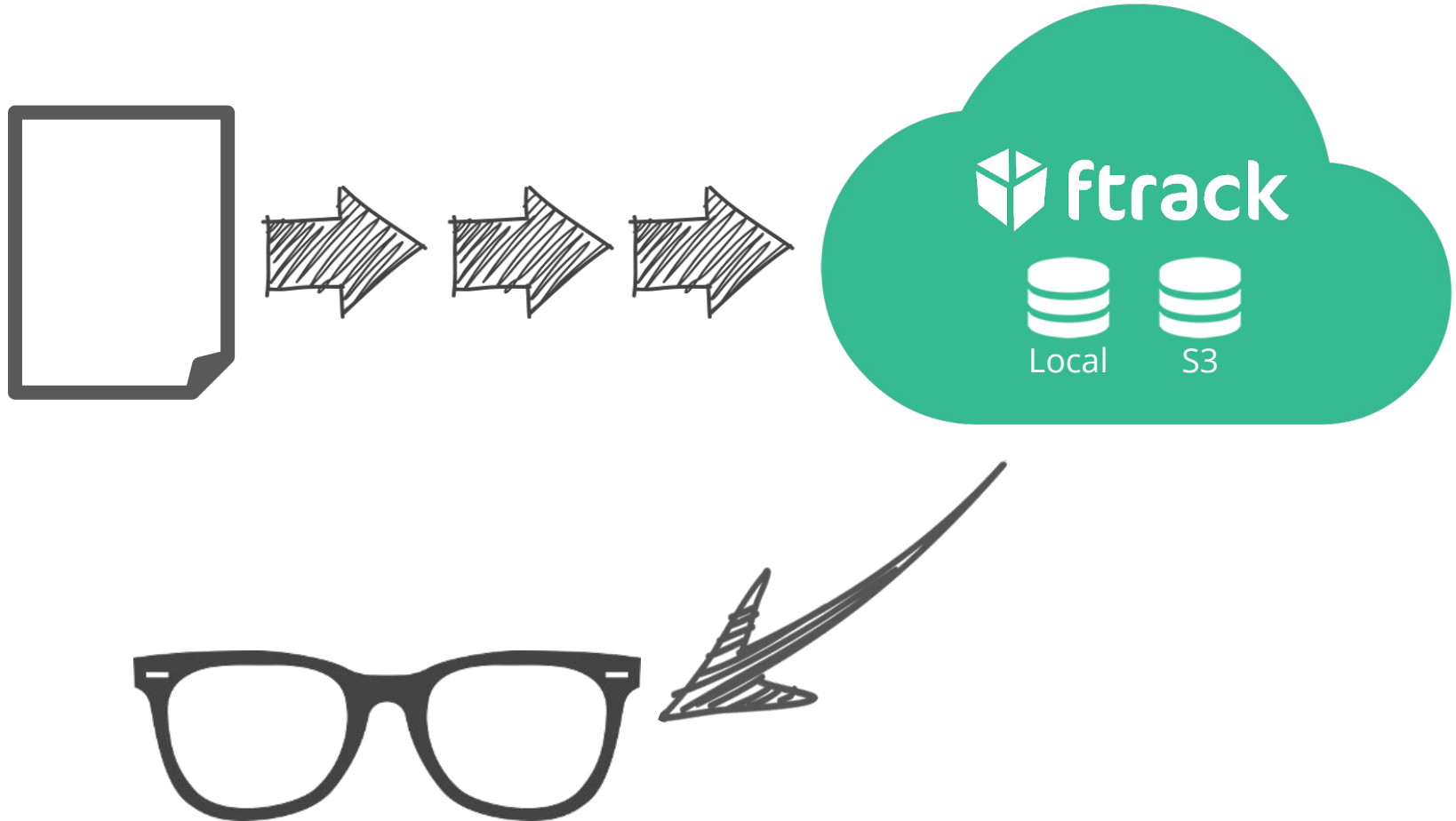
 `/job/ftr/assets/sam/model/002/ftr_assets_sam_model_002.sc`

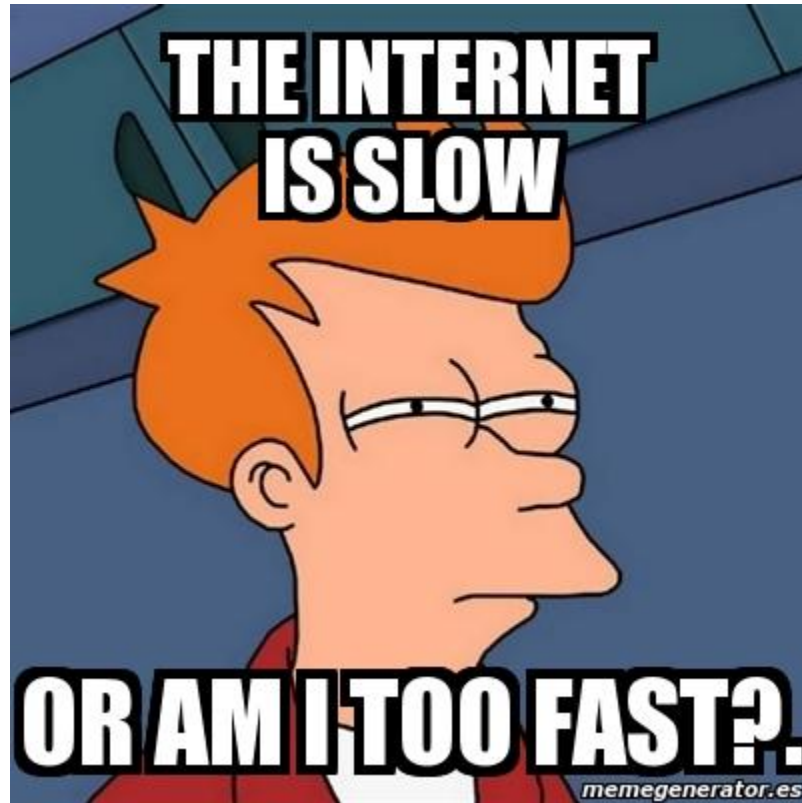
What content?



Traditional content management

Traditional Content Management





Traditional Content Management

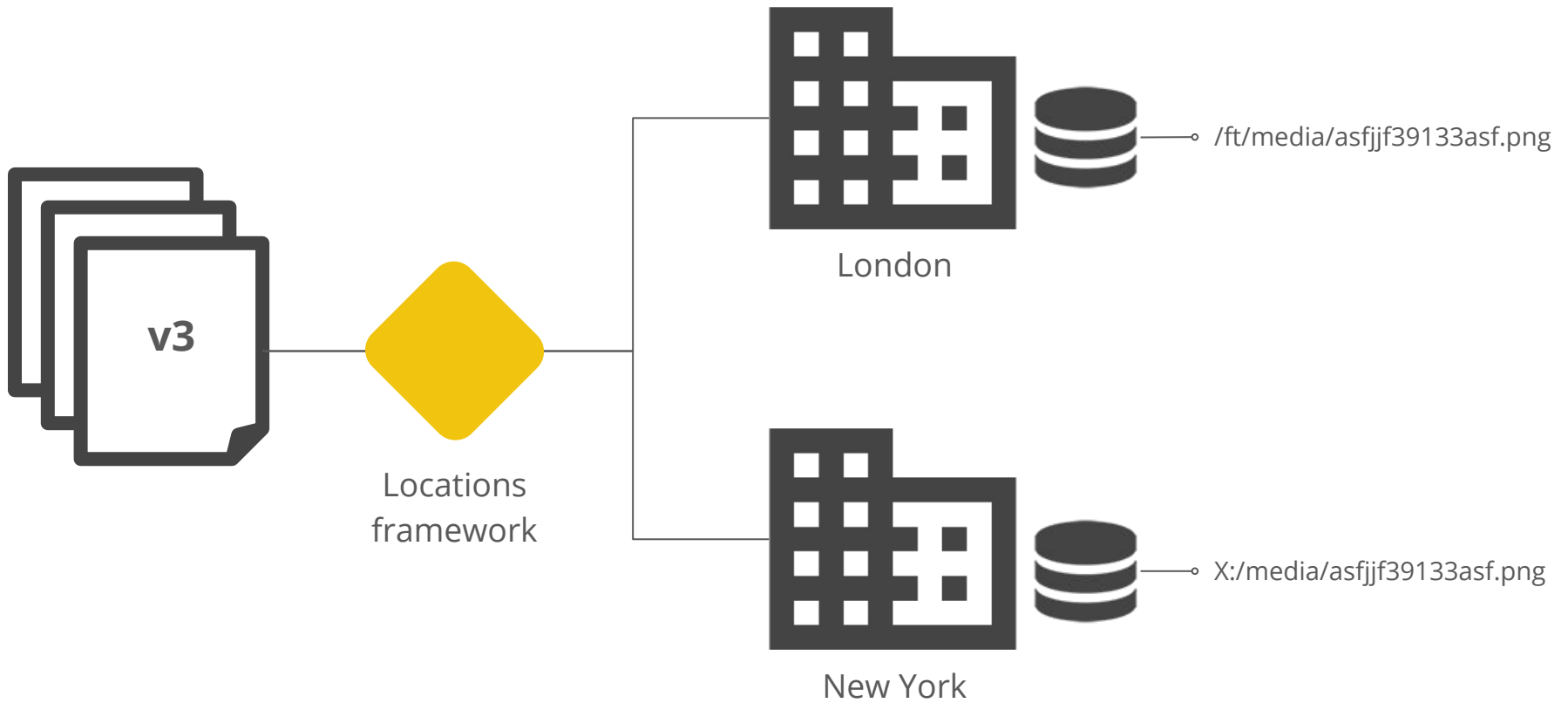


Traditional Content Management



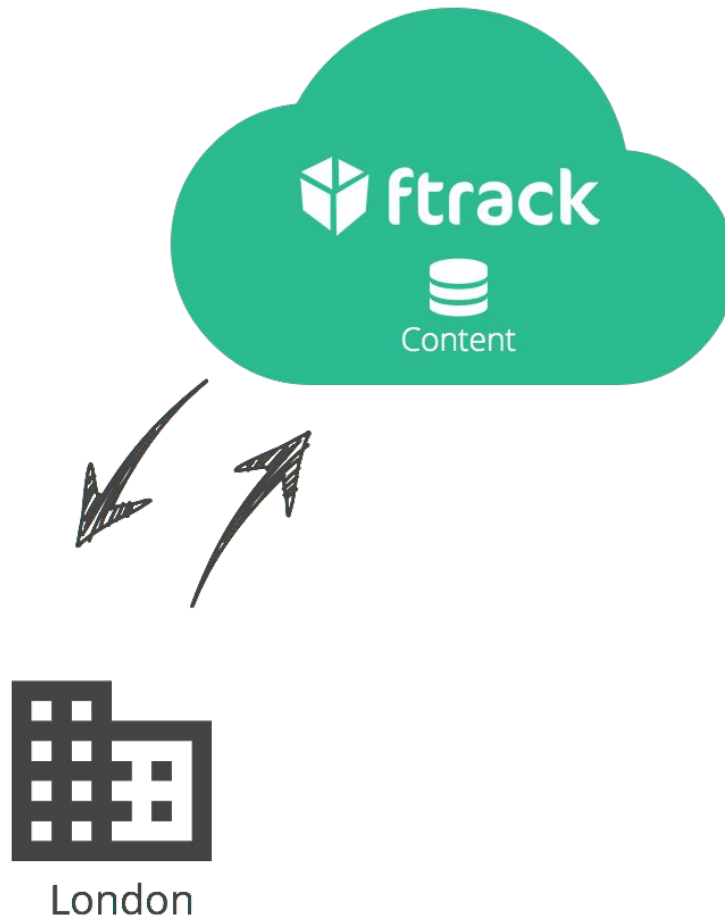
Smart content management

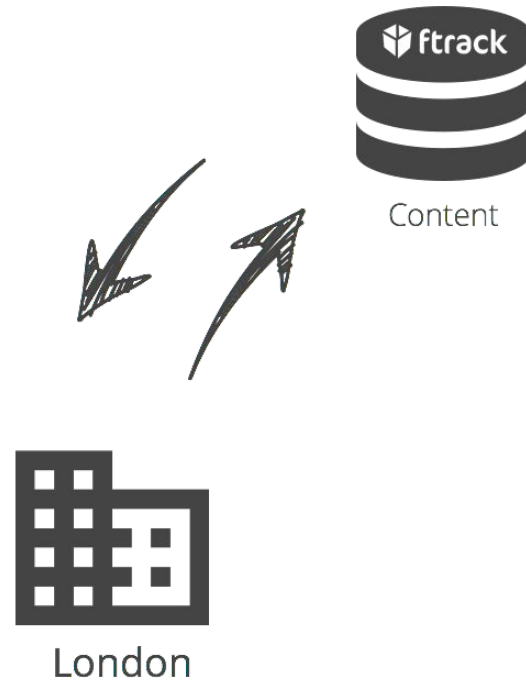
Now available in ftrack 3.3

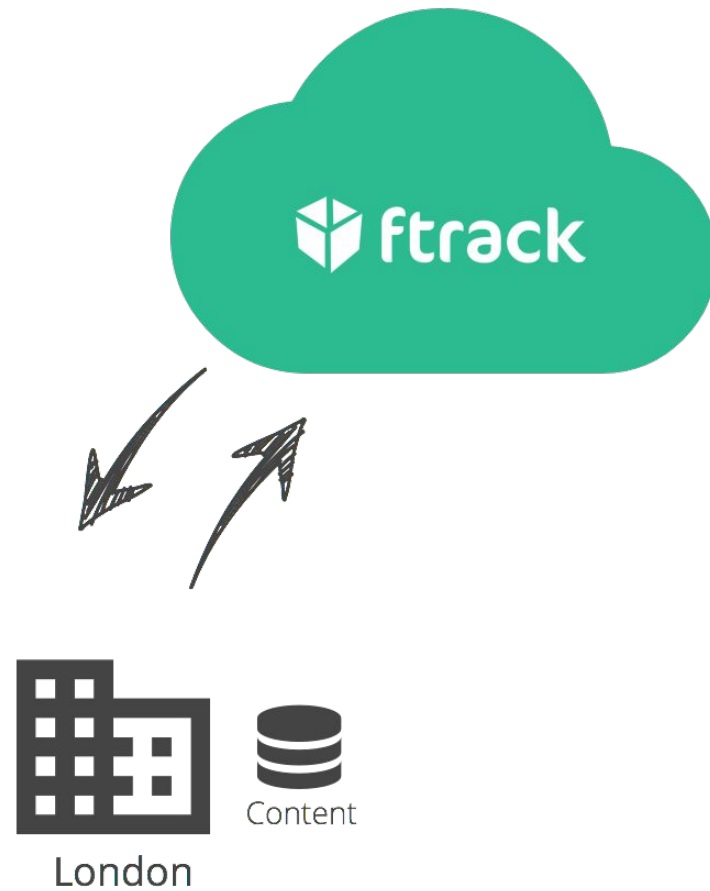




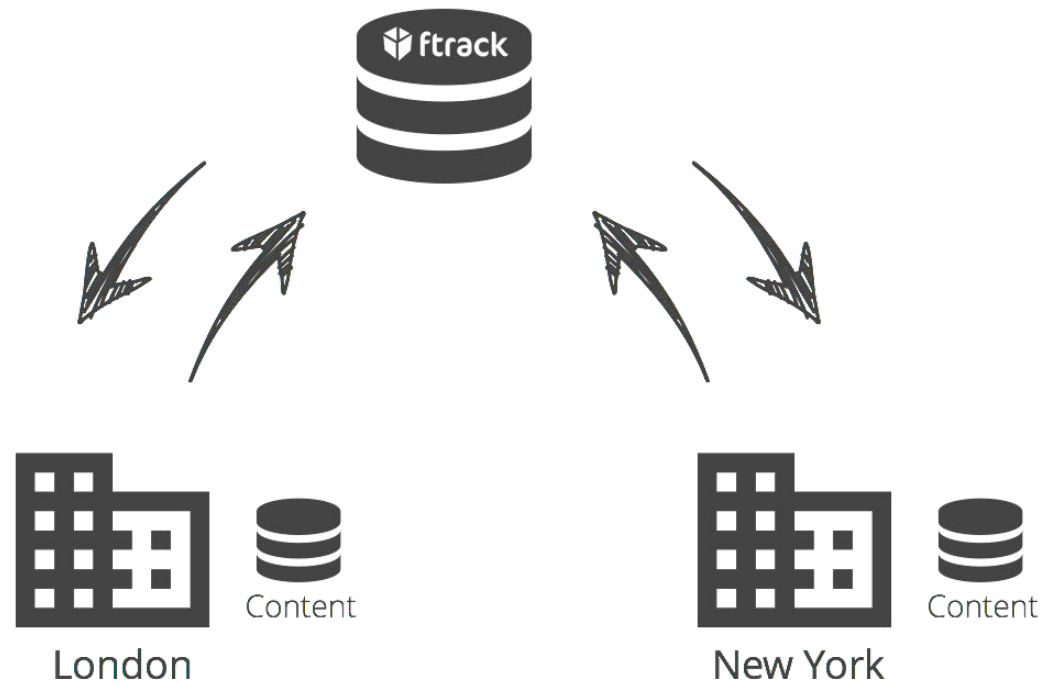
What's possible now?





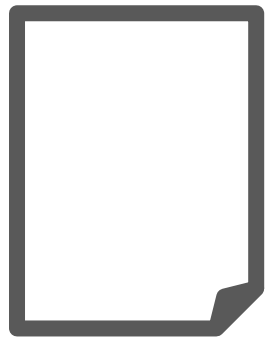


Fully local with separate content management



Details, details

Media upload



File information sent to
the ftrack server



ftrack responds with URL and
metadata to use for
uploading the file



01010101
11011010
10010101
00101010

File data uploaded
accordingly to storage

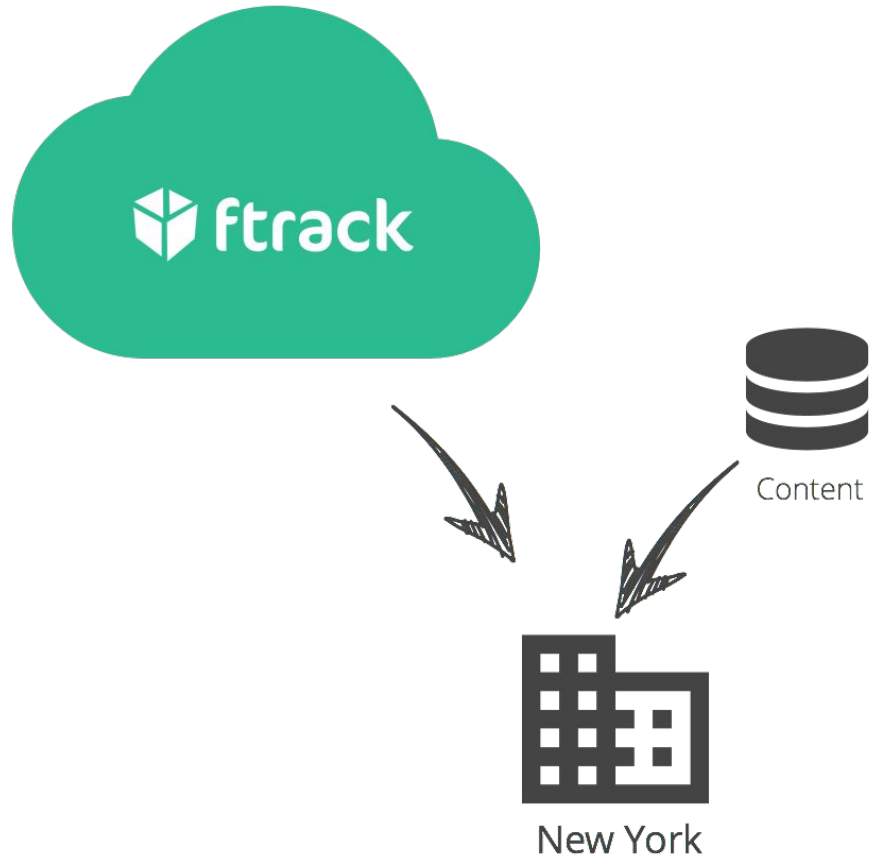


"ftrack.server"
location

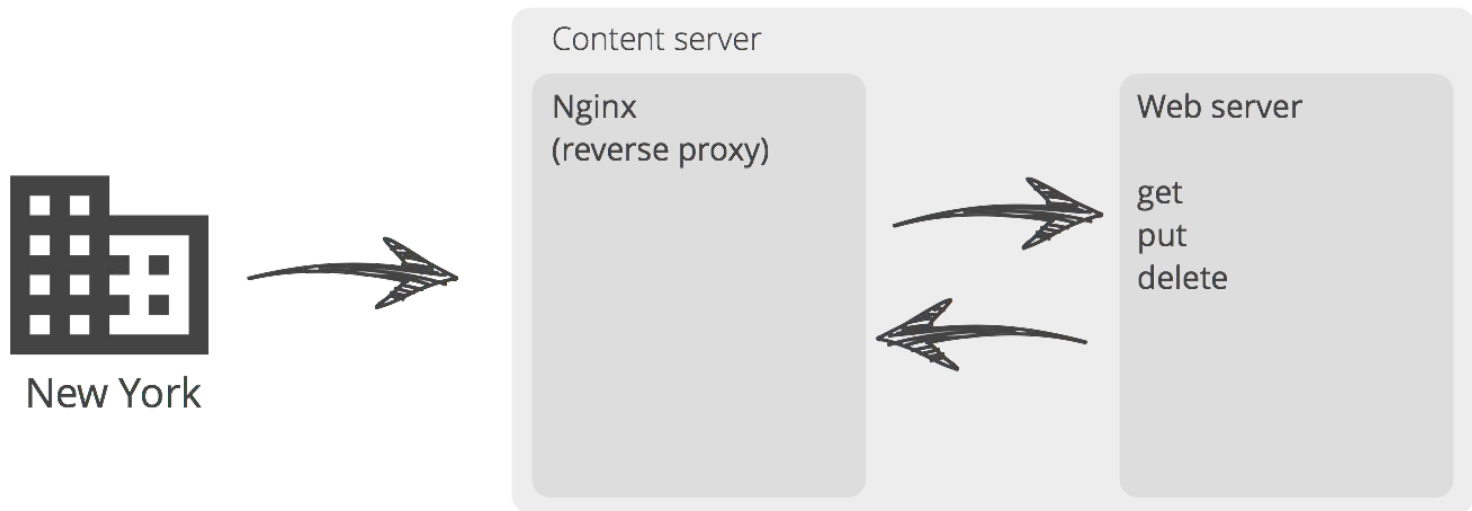
```
server_location = session.query('Location where name is "ftrack.server"').one()
```

```
component = session.create_component('/PATH/TO/SOURCE/MEDIA')
```

```
server_location.add_component(component)
```

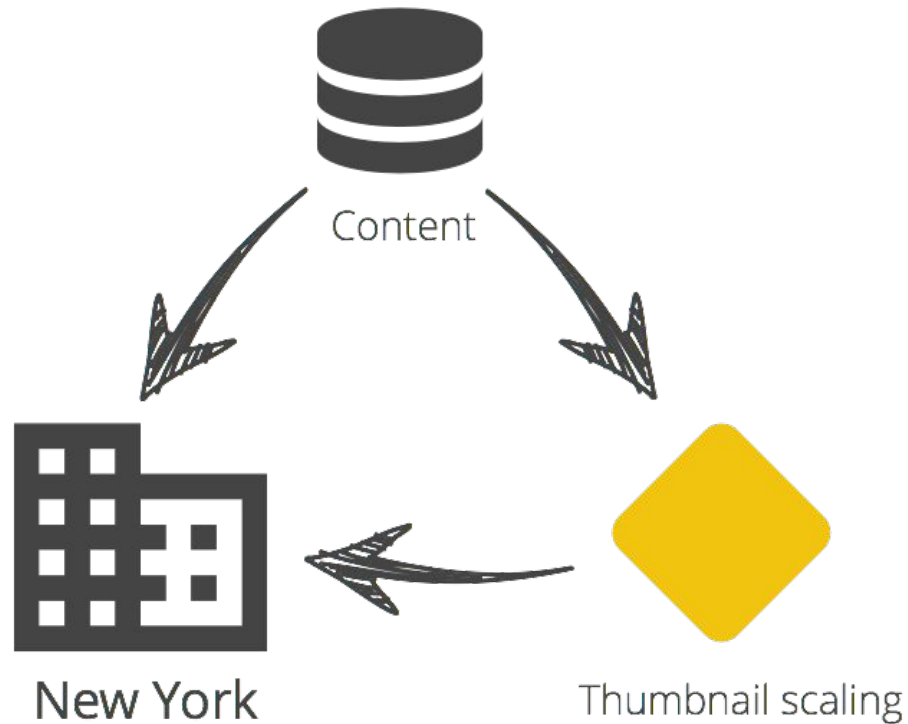


```
def put(id):  
    '''Write uploaded file represented by *id* to disk.  
  
    *id* is the component id and can be used to fetch additional  
    information from ftrack about the file such as extension.  
    '''  
  
def get(id):  
    '''Return the file represented by component *id*.  
  
    Also set Content-Type header based on file extension to ensure  
    browsers know how to read the file.  
    '''  
  
def remove(id):  
    '''Remove the file represented by component *id*.'''
```



Full example at

http://ftrack.rtd.ftrack.com/en/latest/administering/local_file_server.html



Thumbor

<https://github.com/thumbor/thumbor>

What's Next?

- Encoding as a service via API
- Improved controls over media
- Simpler views for non-asset file history
- Out of the box local content server
- Smarter content access and playback

Thank You

Questions?

Locations	http://ftrack.rtd.ftrack.com/en/latest/using/locations/index.html
Content Server	http://ftrack.rtd.ftrack.com/en/latest/administering/local_file_server.html
Python API	http://ftrack-python-api.rtd.ftrack.com/en/latest/

Email support	support@ftrack.com
Community forum	forum.ftrack.com