Monitoring of Unstructured Trial Data (MOUNTD)

10Oct2024

Thomas Ellebæk
Clinical Data Scientist
Global Biometrics

Helping people live better lives

Agenda

1. Introduction
2. High-level design
3. Data flow
4. Creating a check
5. Check output
6. System monitoring

1. Introduction

FERRING PHARMACEUTICALS

Abstract

“The Global Biometrics department at Ferring had a visionary goal: to streamline clinical trial image collection and automate quality control and reconciliation tasks in-house. This year, we’re proud to introduce our new system, Live File Checker (LFC). Developed in-house on the Azure cloud platform, LFC comprises a rich array of cloud components, including container apps and message queues. It is integrated with EDC (Viedoc) and our clinical database (LSAF). Notably, LFC empowers Python custom coding for quality checks, automatically triggered upon the upload of new files (such as images) to EDC.”

Abstract

"The Global Biometrics department at Ferring had a visionary goal: to streamline clinical trial image collection and automate quality control and reconciliation tasks in-house. This year, we're proud to introduce our new system, Live File Checker (LFC). Developed in-house on the Azure cloud platform, LFC comprises a rich array of cloud components, including container apps and message queues. It is integrated with EDC (Viedoc) and our clinical database (LSAF). Notably, LFC empowers Python custom coding for quality checks, automatically triggered upon the upload of new files (such as images) to EDC."

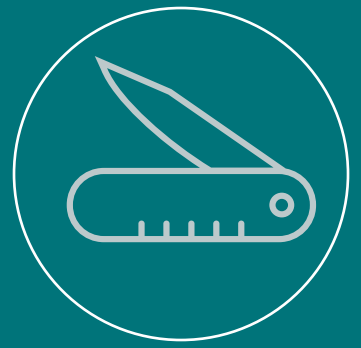
TEAM:

Mads Raad	Project Statistician
Jens Langendorf	System Development Manager
Sebastian Gilson	Compliance and Process Officer
Thomas Ellebæk	Clinical Data Scientist

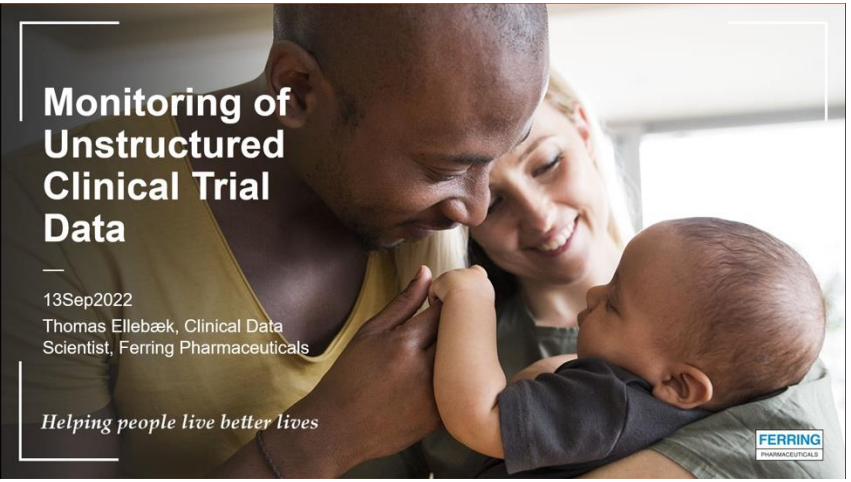
Takeaways

More and more trials include unstructured data, so we need a cheap way to collect it

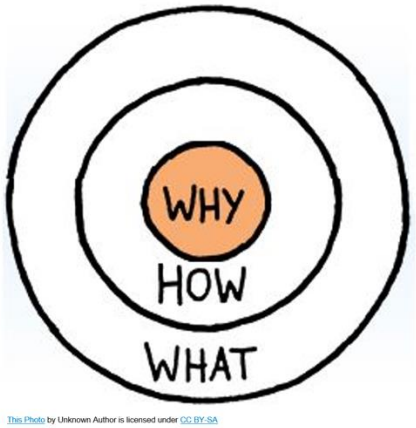
Unstructured data should be cleaned in the same pace as any other data



Vision presented September 2022 at Nordic Innovation Day



Agenda



This Photo by Unknown Author is licensed under CC BY-SA

For Public Use - Public

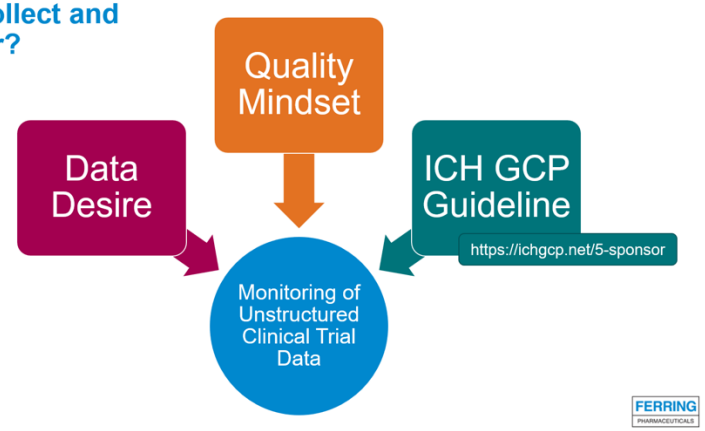
WHAT do we mean by monitoring of unstructured clinical trial data?

- Monitoring
  to watch and check a situation carefully for a period of time in order to discover something about it:
  https://dictionary.cambridge.org/dictionary/english/monitoring
  Centralized Monitoring
- Unstructured Data
  Unstructured data is digital information that cannot be stored efficiently in a relational database (RDMS) because it does not use pre-set data models or schemas.
  https://www.techtarget.com/searchbusinessanalytics/definition/unstructured-data
- Clinical Trial Data
  Data collected during a clinical trial.
  https://medical-dictionary.thefreedictionary.com/clinical+trial+data

Unstructured Clinical Trial Data
13Sep2023:
Medical images, e.g. ultrasound scans, embryo development time-lapse movies.

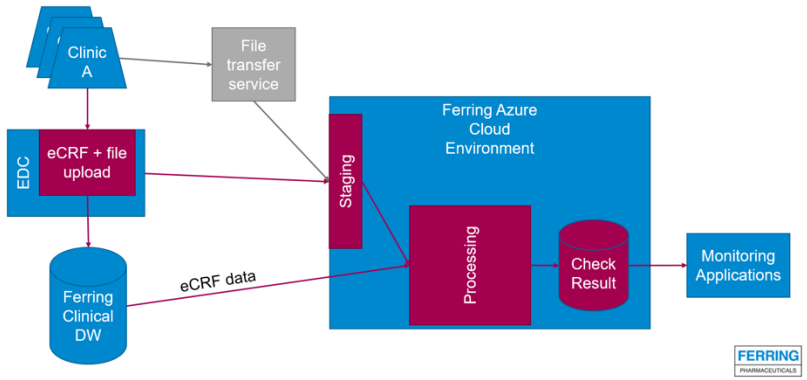
For Public Use - Public

WHY collect and monitor?



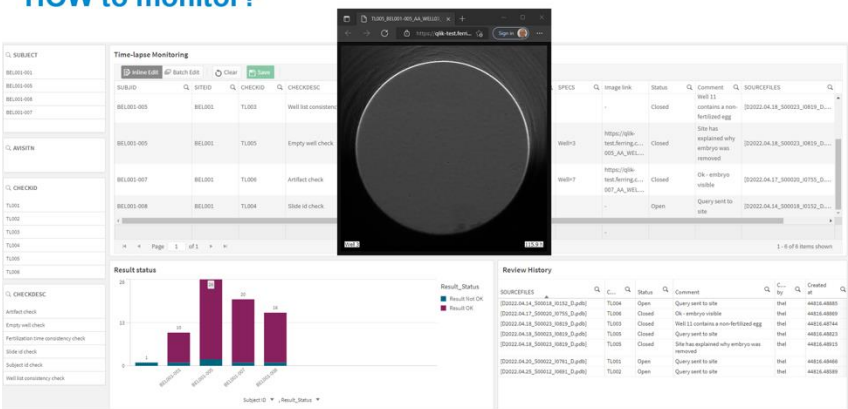
For Public Use - Public

HOW to collect and process?



For Public Use - Public

HOW to monitor?

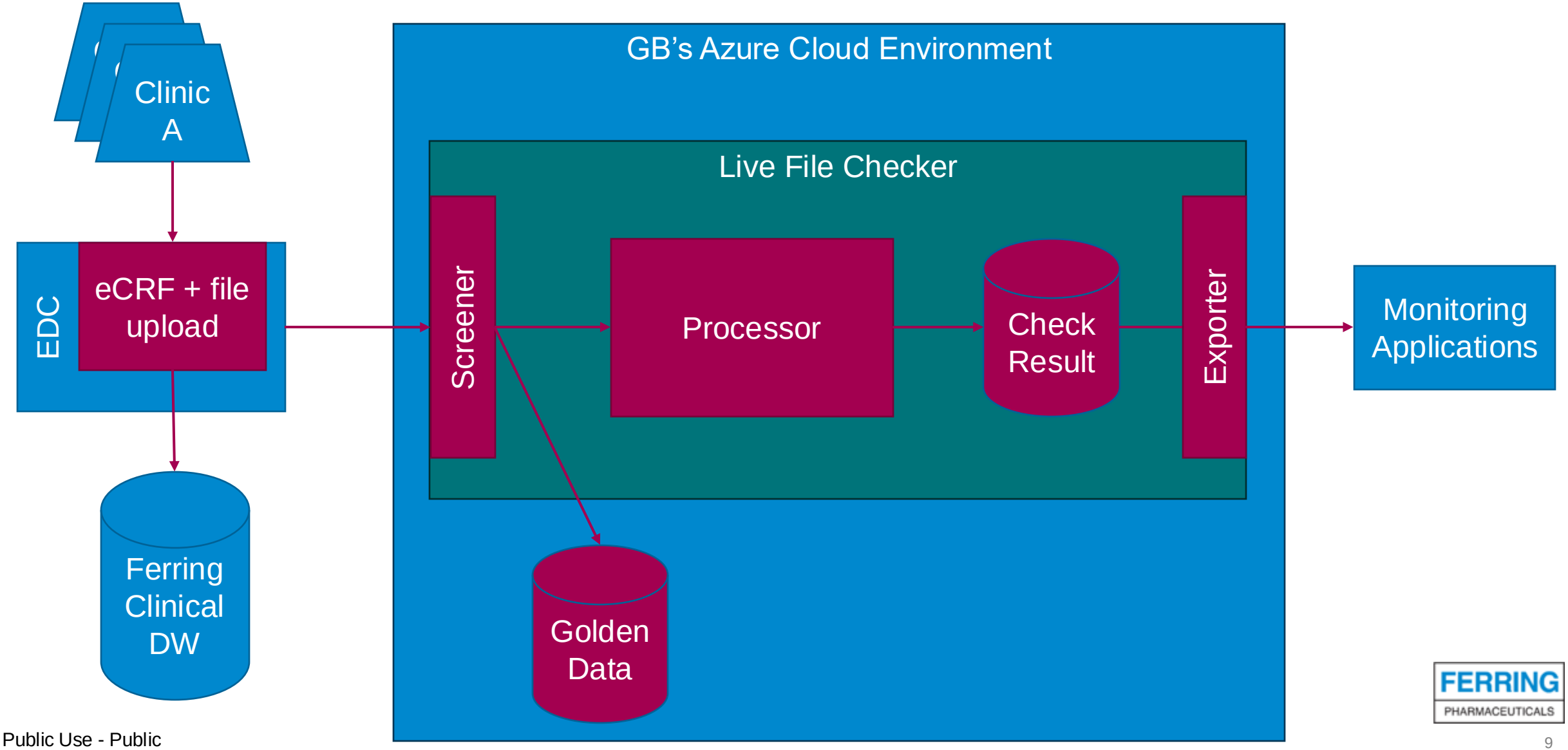


For Public Use - Public

2. High-level design

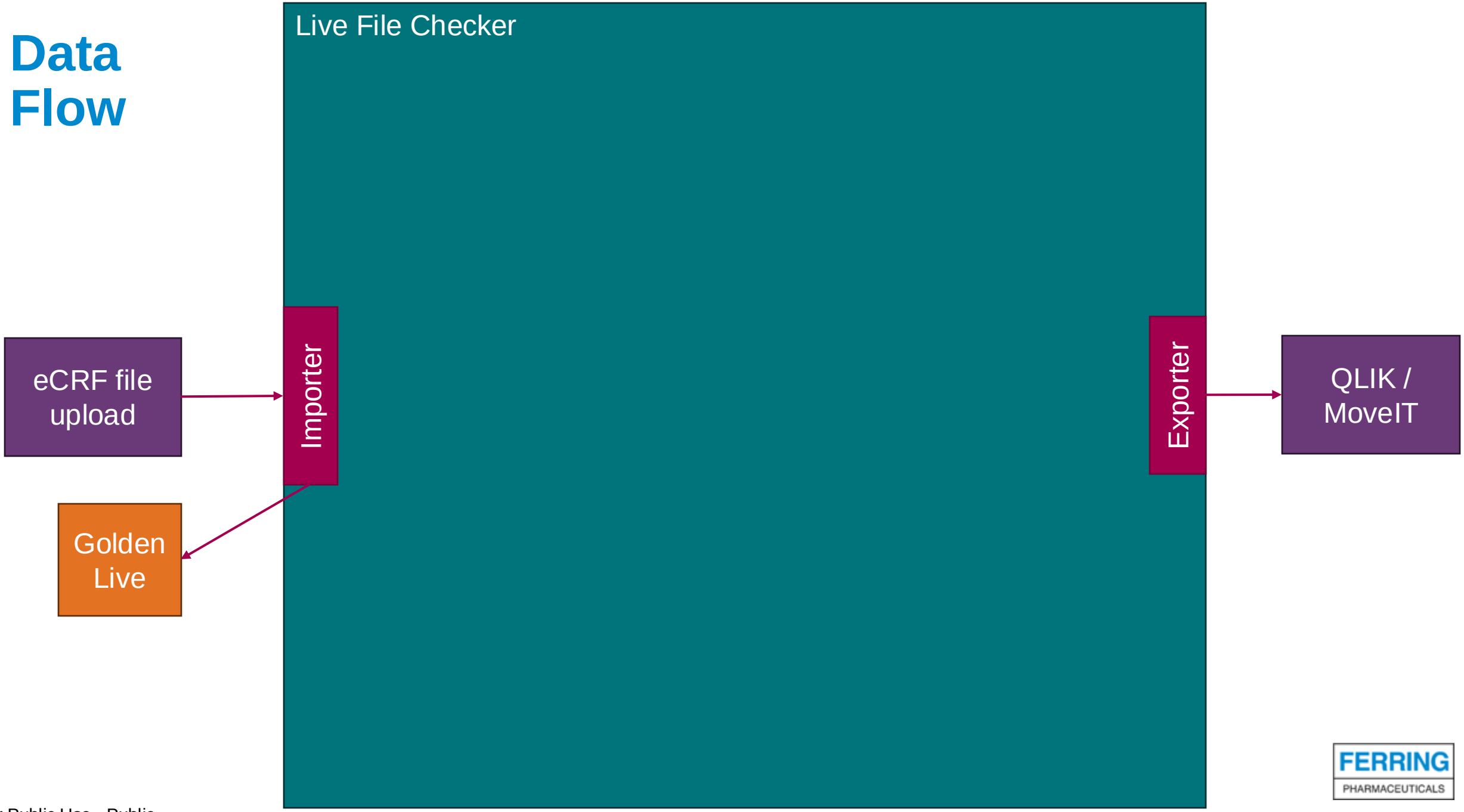
FERRING PHARMACEUTICALS

MOUNTD HLD

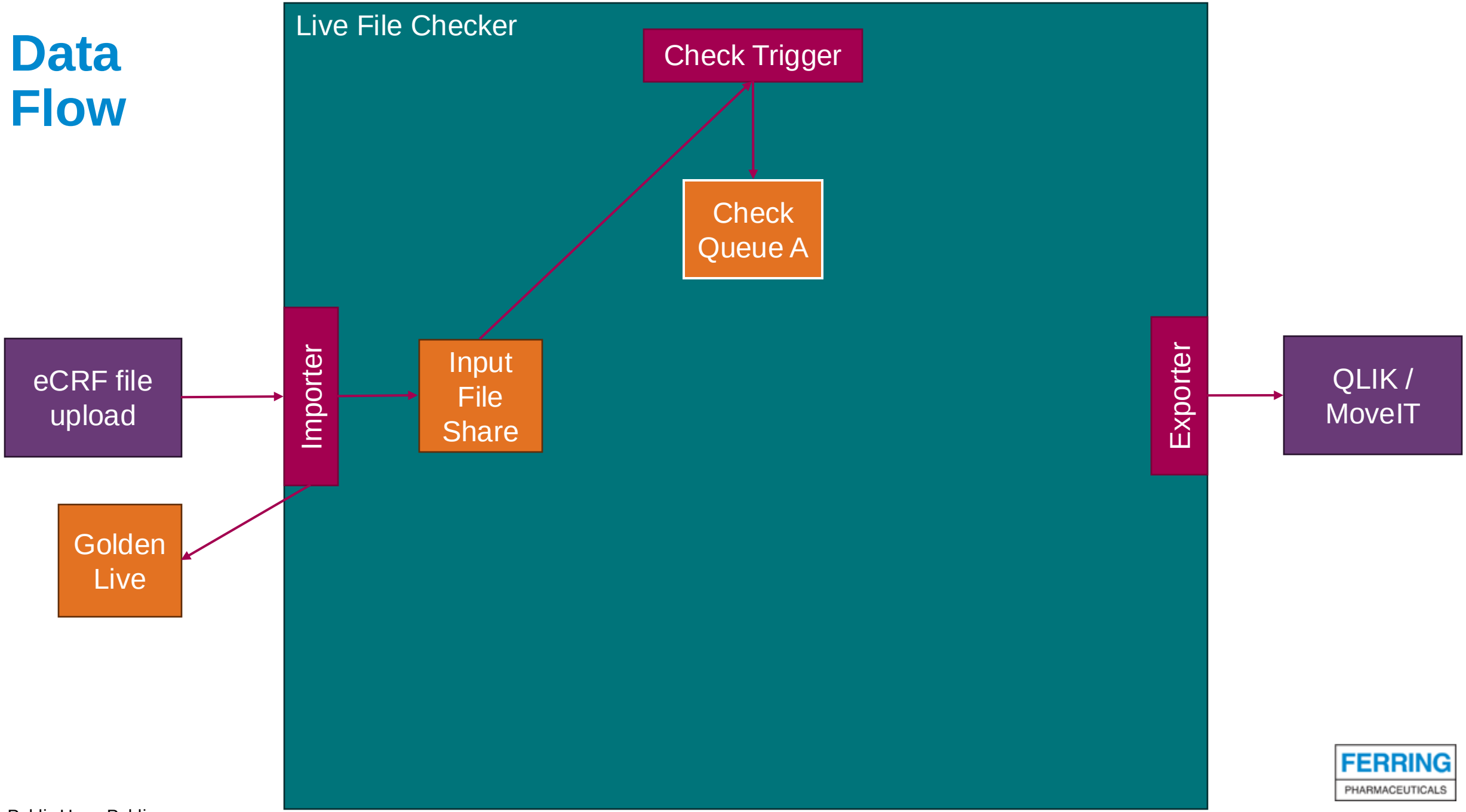


3. Data flow

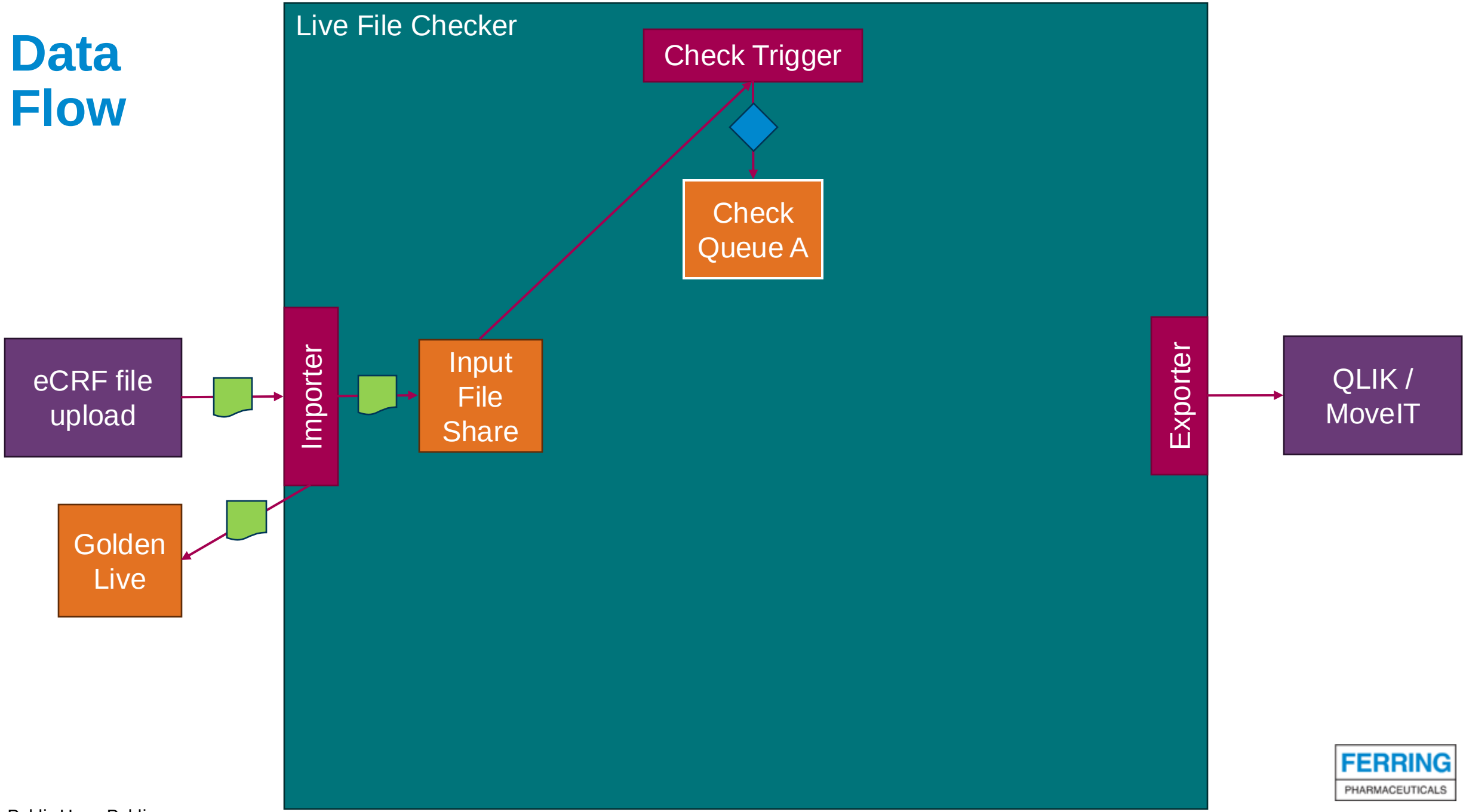
Data Flow



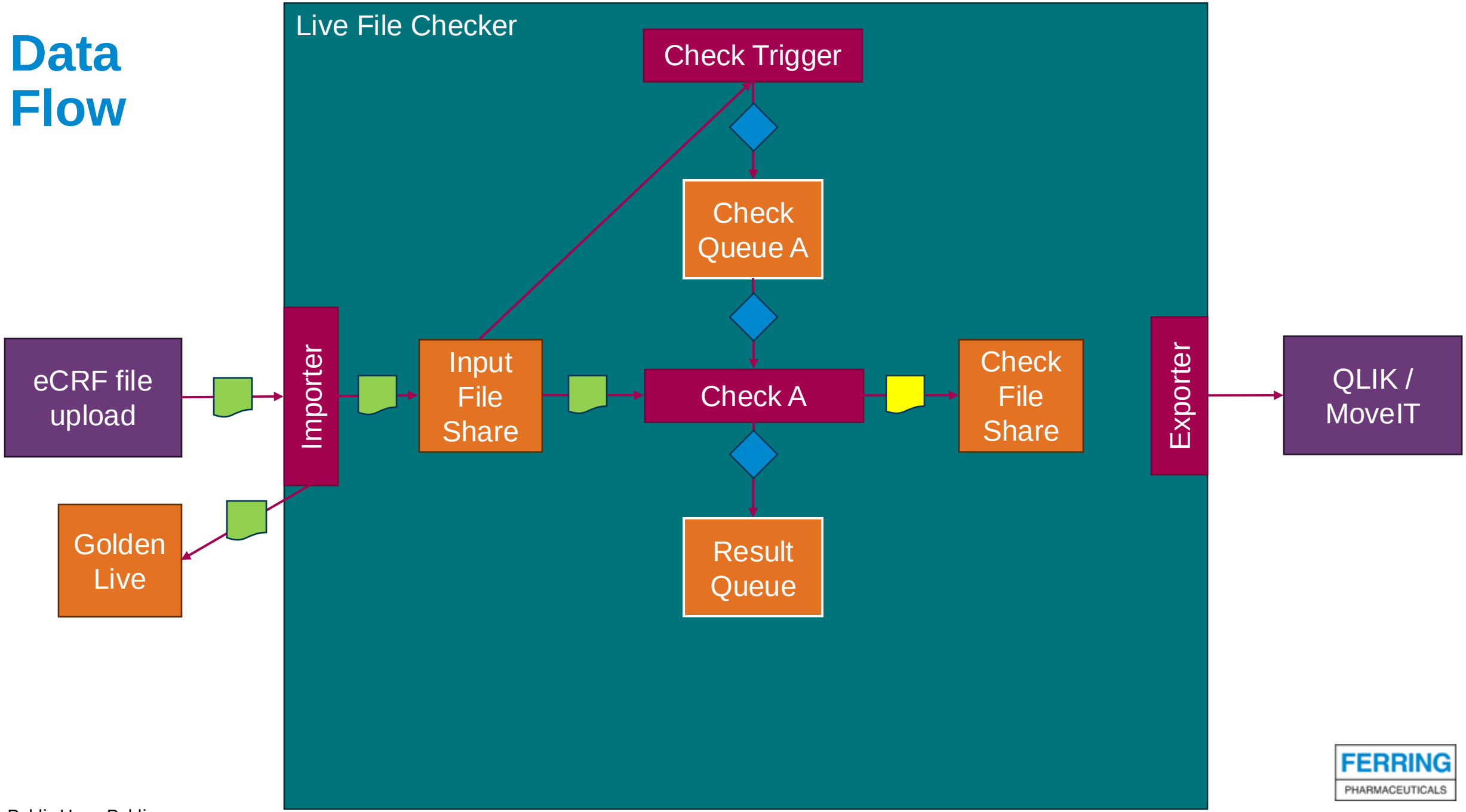
Data Flow



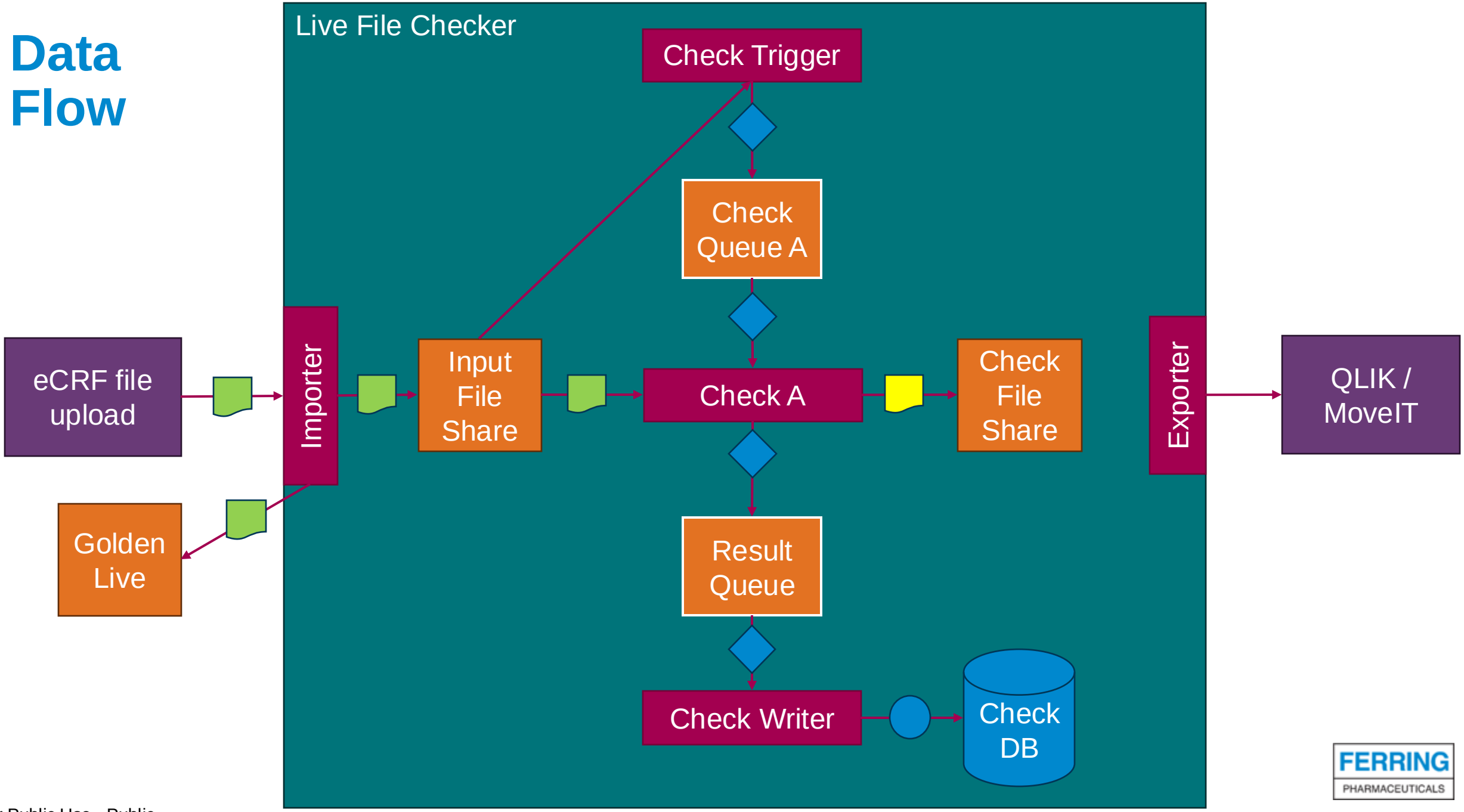
Data Flow



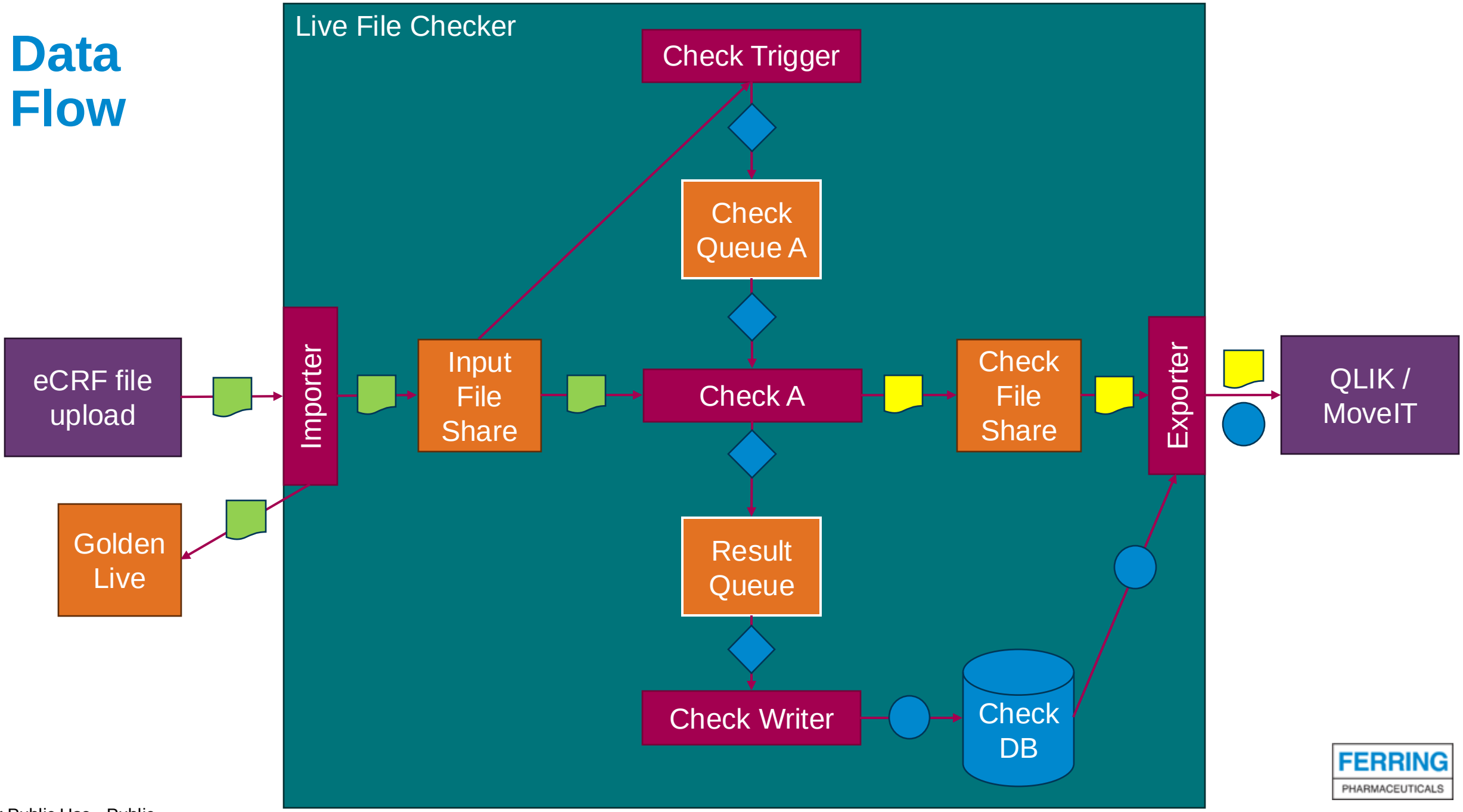
Data Flow



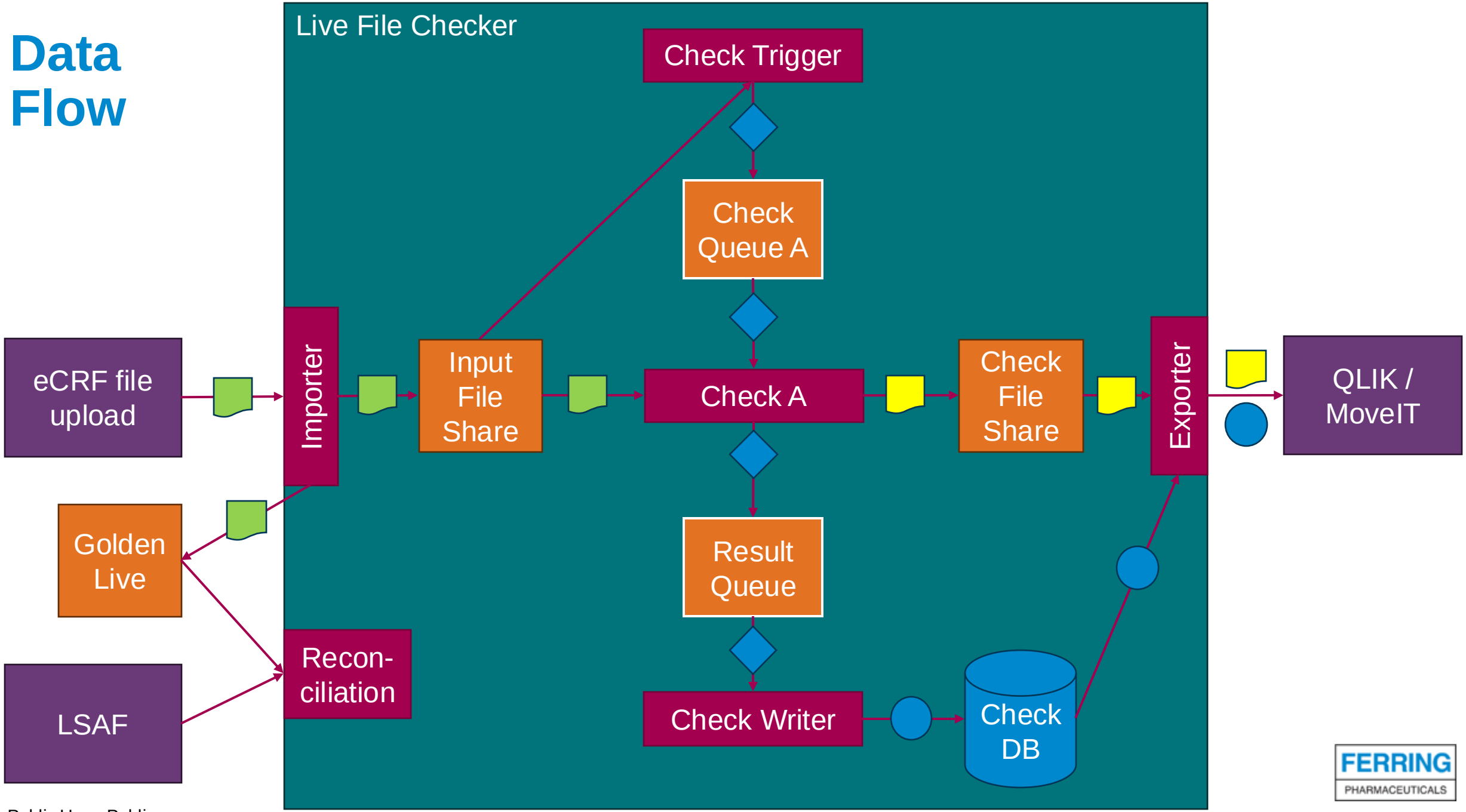
Data Flow



Data Flow



Data Flow



4. Creating a check

Check Example

```python
import warnings
from lfc_common.check_db_interface.py_models.models.entries import ResultEntry
from src.base_check import BaseCheck
from lfc_common.lfc_logging.py.src.log_msg import Logger, LogLevel
import pydicom
import os
import copy
from PIL import Image


class Dicom2DStillImageCheck(BaseCheck):
    """
        Expecting all files to be DICOM 2D still images.
        Set finding=True when file does not have exact 1 image region.
        Set finding=True when pixel_array shape has not exactly 3 dimensions (width, heigh, color_channel).
        Skipping check if not a dicom file.
    """

    def check(self, result_entry_inc: ResultEntry) -> [ResultEntry]:
        data_dir = self.data_dir(result_entry_inc)
        output_dir = self.trial_output_dir(result_entry_inc)
        free_text = "multiple_img_regions"
        result_entries = []

        for file in os.listdir(data_dir):
            result_entry_copy = copy.deepcopy(result_entry_inc)
            result_entry_copy.finding = False
            result_entry_copy.specs = f"source={file}"
            file_path = os.path.join(data_dir, file)
            file_without_extension, file_extension = os.path.splitext(file)
            file_extra = "_" if file_extension == "" else ""

            if file_extension.lower() in ['.dcm', '']:
                try:
                    dicom_data = pydicom.dcmread(file_path)
                    if os.getenv('DEP_ENV') in ['test', 'prod']:
                        warnings.filterwarnings("ignore", category=UserWarning)
                    pixel_data = dicom_data.pixel_array
                    warnings.resetwarnings()
```

Check Example Continued

```
41                # NUMBER OF REGIONS PART
42                regions = dicom_data.SequenceOfUltrasoundRegions
43                # Check looks at number of PhysicalDeltaX attribute to determine number of regions.
44                num_physical_delta_x = 0
45                for region in regions:
46                    if region.get('PhysicalDeltaX') is not None:
47                        num_physical_delta_x += 1
48
49                if num_physical_delta_x == 0:
50                    result_entry_copy.result_text = f"DICOM contains no image regions. "
51                    result_entry_copy.finding = True
52                elif num_physical_delta_x > 1:
53                    result_entry_copy.result_text = f"DICOM contains {num_physical_delta_x} image regions, " \
54                                                    f"only 1 expected. "
55                    result_entry_copy.finding = True
56
57                    # save image_ref so it is viewable in Qlik
58                    if len(pixel_data.shape) == 3:
59                        output_file_name = f"{self.check_id()}_{result_entry_copy.screening_id}" \
60                                           f"_{free_text}_{file_without_extension}{file_extra}.jpg"
61                        result_entry_copy.image_ref = os.path.join(result_entry_inc.trial_id, output_file_name)
62                        new_file_path = os.path.join(output_dir, output_file_name)
63                        Image.fromarray(pixel_data).save(new_file_path)
64
65                # NUMBER OF DIMENSIONS PART
66                if len(pixel_data.shape) > 3:
67                    result_entry_copy.result_text = result_entry_copy.result_text + \
68                                                    f"DICOM contains more than a 2D still image, " \
69                                                    f"{pixel_data.shape[0]} frames detected."
70                    result_entry_copy.finding = True
71                elif len(pixel_data.shape) < 3:
72                    result_entry_copy.result_text = result_entry_copy.result_text + \
73                                                    f"DICOM pixel array has an unexpected number of dimensions " \
74                                                    f"({len(pixel_data.shape)})."
75                    result_entry_copy.finding = True
76
```

Check Example Continued

```
77                  # NO FINDING
78                  if not result_entry_copy.finding:
79                      result_entry_copy.result_text = f"DICOM contains a 2D still image."
80
81              except Exception as e:
82                  Logger.log(msg=f"Error for '{file_path}': {str(e)}",
83                             lvl=LogLevel.WARN, scope=result_entry_inc.trial_id)
84                  result_entry_copy.result_text = f"Error processing DICOM file: {str(e)}"
85                  result_entry_copy.finding = False
86              finally:
87                  result_entries.append(result_entry_copy)
88
89          else:
90              result_entry_copy.result_text = f"Check skipped. File is not a dicom file."
91              result_entry_copy.finding = False
92              result_entries.append(result_entry_copy)
93
94      return result_entries
```

5. Check output

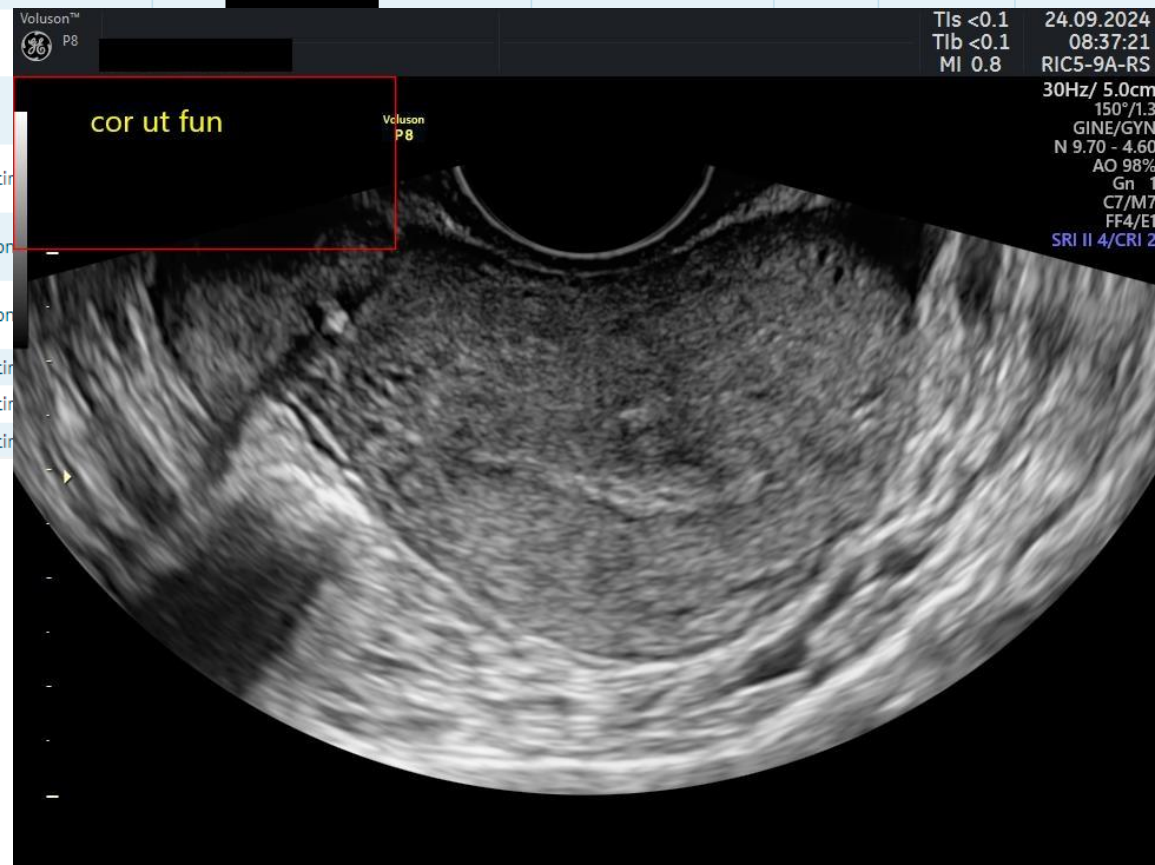
Data Monitoring Application

Accepted abbreviation 'cor ut fund' not part of detected text '| cor ut fun v'.

Check results table

Inline Edit | Batch Edit | Clear | Save | Export

Screen No	Site ID	Check ID	Check description	Finding	Result text	Visit event ID	Specs	Tmestamp (UTC)	Image link	Status	Comment
[redacted]	ESP001	cel-002	Accepted abbreviation check.	True	Accepted abbreviation 'cor ut fund' not part of detected text '| cor ut fun v'.	Stimulation_Day_1	source=[redacted]_6.dcm	2024-09-24T14:00:51	link	Open	
	ESP001	cel-002	Accepted abbreviation check.	True	Accepted abbreviation 'cor ut fund' not part of detected text 'Voluson â ,'.	Transfer	source=[redacted]_6.dcm	2024-09-23T20:00:30	link	Open	
	ESP001	cel-004	2D still image check.	True	DICOM contains more than a 2D still image, 198 frames detected.	Transfer					
	ESP001	cel-002	Accepted abbreviation check.	True	Accepted abbreviation 'cor ut fund' not part of detected text 'cor ut fun â '.	Transfer					
	ESP001	cel-002	Accepted abbreviation check.	True	Accepted abbreviation 'sag lus' not part of detected text ''.	End_of_Stir					
	ESP001	cel-002	Accepted abbreviation check.	True	Accepted abbreviation 'sag lus' not part of detected text 'sag lu:'.	Stimulation					
	ESP001	cel-002	Accepted abbreviation check.	True	Accepted abbreviation 'cor ut fund' not part of detected text 'cor ut fun'.	Stimulation					
	ESP001	cel-001	DICOM readable check.	False	File is a readable DICOM file.	End_of_Stir					
	ESP001	cel-001	DICOM readable check.	False	File is a readable DICOM file.	End_of_Stir					
	ESP001	cel-001	DICOM readable check.	False	File is a readable DICOM file.	End_of_Stir					



6. System monitoring

FERRING PHARMACEUTICALS

Reconciliation

Log Analytics Workspace

LogLevel	INFO
LogMessage	This reconciliation run is between audit trail in LSAF and GDL in Azure. Files in total (LSAF and Azure combined): 80 Files in LSAF only: 0 Files in Azure only: 0 Files in both LSAF and Azure: 80 Files with size equal: 80 Files with size unequal: 0 Files with hash equal: 80 Files with hash unequal: 0 Files with order of timestamps right: 80 Files with order of timestamps wrong: 0
LogStamp	2024-09-30T07:00:36.268483

Alerts Rules

Name	Condition	Severity
lfc_prod_ca_audit	Table rows > 0	1 - Error
lfc_prod_ca_error	Table rows > 0 In selected dimensions	1 - Error
lfc_prod_ca_log_format	Table rows > 0 In selected dimensions	1 - Error
lfc_prod_ca_not_running_0	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_not_running_1	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_not_running_2	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_not_running_3	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_warn	Table rows > 0 In selected dimensions	2 - Warning
lfc_prod_sa_audit	Table rows > 0	1 - Error
lfc_prod_sa_dir_del	Table rows > 0	2 - Warning
lfc_prod_sa_file_del	Table rows > 0	2 - Warning
lfc_prod_sa_fs_request	Table rows > 0	2 - Warning
lfc_prod_sa_result_q	Table rows <= 0	3 - Informational
lfc_prod_sql_audit	Table rows > 0	1 - Error
lfc_prod_sql_non_uami	Table rows > 0	2 - Warning

Alerts Rules

Name	Condition	Severity
lfc_prod_ca_audit	Table rows > 0	1 - Error
lfc_prod_ca_error	Table rows > 0 In selected dimensions	1 - Error
lfc_prod_ca_log_format	Table rows > 0 In selected dimensions	1 - Error
lfc_prod_ca_not_running_0	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_not_running_1	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_not_running_2	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_not_running_3	Table rows <= 0 In selected dimensions	2 - Warning
lfc_prod_ca_warn	Table rows > 0 In selected dimensions	2 - Warning
lfc_prod_sa_audit	Table rows > 0	1 - Error
lfc_prod_sa_dir_del	Table rows > 0	2 - Warning
lfc_prod_sa_file_del	Table rows > 0	2 - Warning
lfc_prod_sa_fs_request	Table rows > 0	2 - Warning
lfc_prod_sa_result_q	Table rows <= 0	3 - Informational
lfc_prod_sql_audit	Table rows > 0	1 - Error
lfc_prod_sql_non_uami	Table rows > 0	2 - Warning

lfc_prod_sa_fs_request

Log search alert details

Copy link Go to alert rule

Summary History

Alert fired

Action group triggered

Action group LFC PROD Action Group triggered

Configured by Name
Alert rule lfc_prod_sa_fs_request

User response changed

THEL-ADM@ferring.com

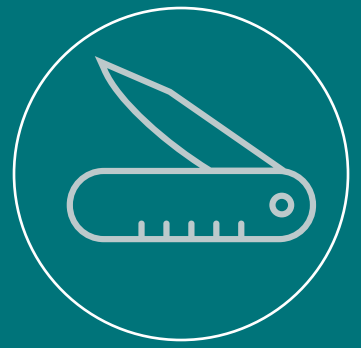
Changed from New To Closed

Testing new version of storage explorer (1.35). Data file share loaded but no files inspected.

Takeaways

More and more trials include unstructured data, so we need a cheap way to collect it

Unstructured data should be cleaned in the same pace as any other data



New platforms, tools and skills are needed