ENDURADATA

PROTECTING, DELIVERING AND LEVERAGING DATA

Automating data aggregation, ingestion and analysis from multiple health sources.

A. El Haddi Founder/CTO -- linkedin.com/in/aelhaddi

DISCLAIMERS

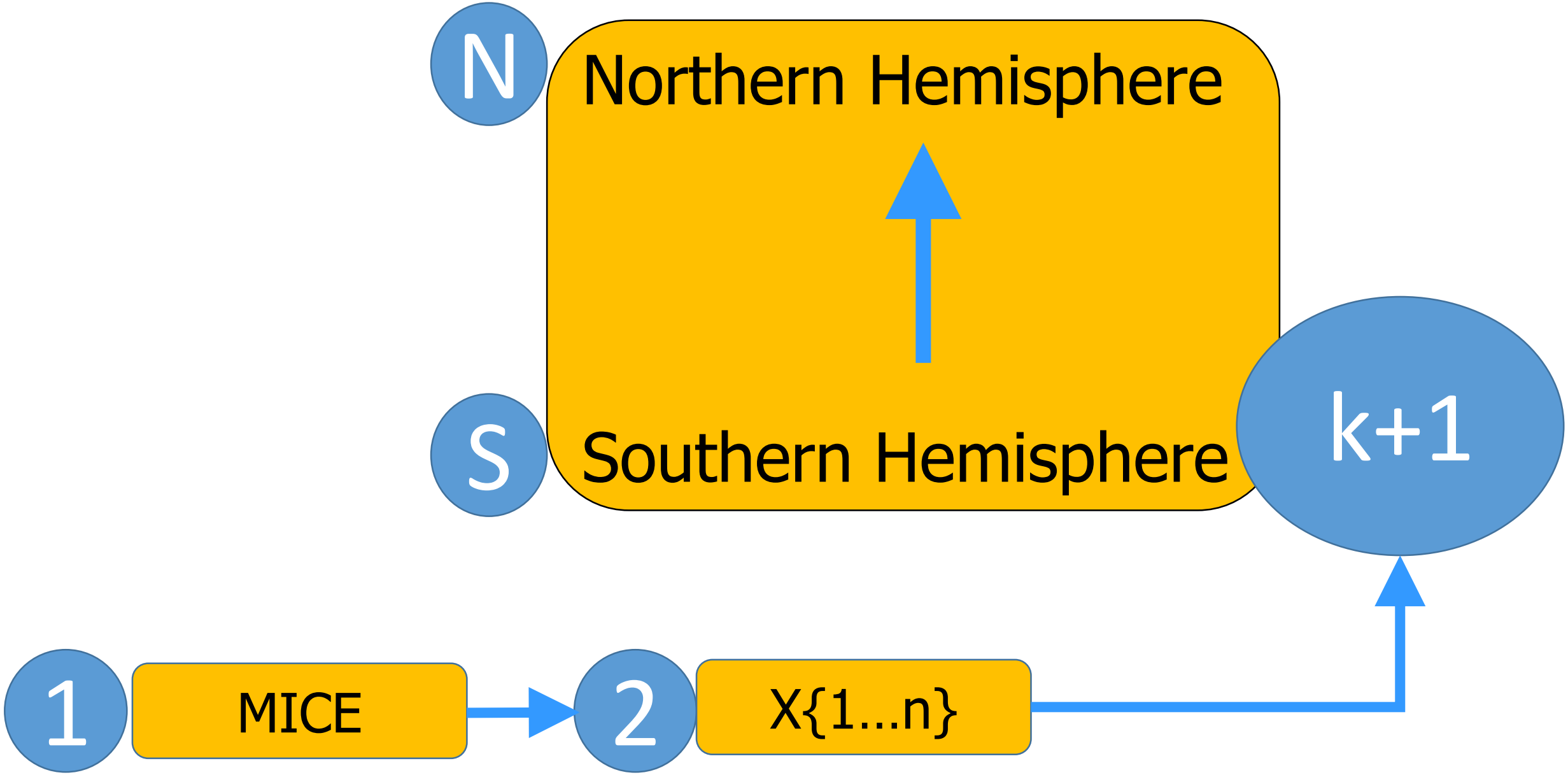
- I am not a medical doctor
- But my title in the 80s was scientist & data manager.

Very Sensitive Data

- Enrollment information (i.e., entire family's info including SSN!)
- PHI
- Doctors' notes
- Operations Documents/spreadsheets
- Images (Xrays, CT Scans, MRI ….)
- Clinical research data
- Pharma & Device manufacturer info
- Scientific literature, reports, etc.

Reference: Rd.com

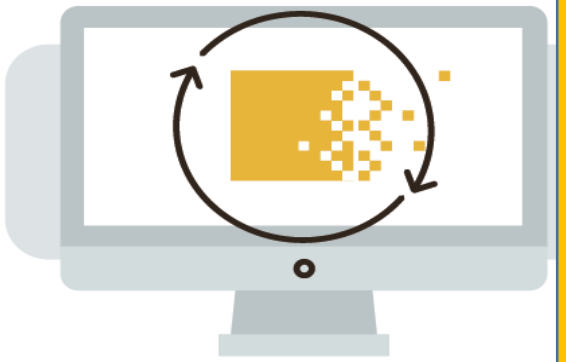
Global Trials … different Arms



Variety of business challenges

- Process optimization
- Data transfer/delivery
- Automatic && **reproducible** exploratory analysis
- Leveraging data
- Data protection
- Reducing errors & risks.

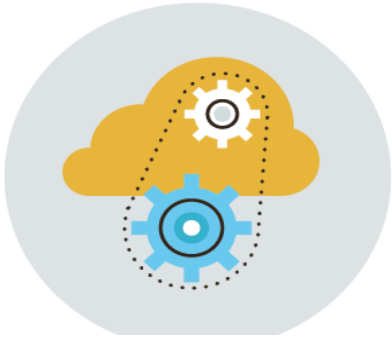
More technical challenges & externalities:



- Need continuous secure access to data
- Data where needed & when needed
- Time zones
- Networks & providers
- Governments & regulations
- Need to automate reproducible reporting.

Multiple Platforms

Multiple Sites



Multiple Clouds

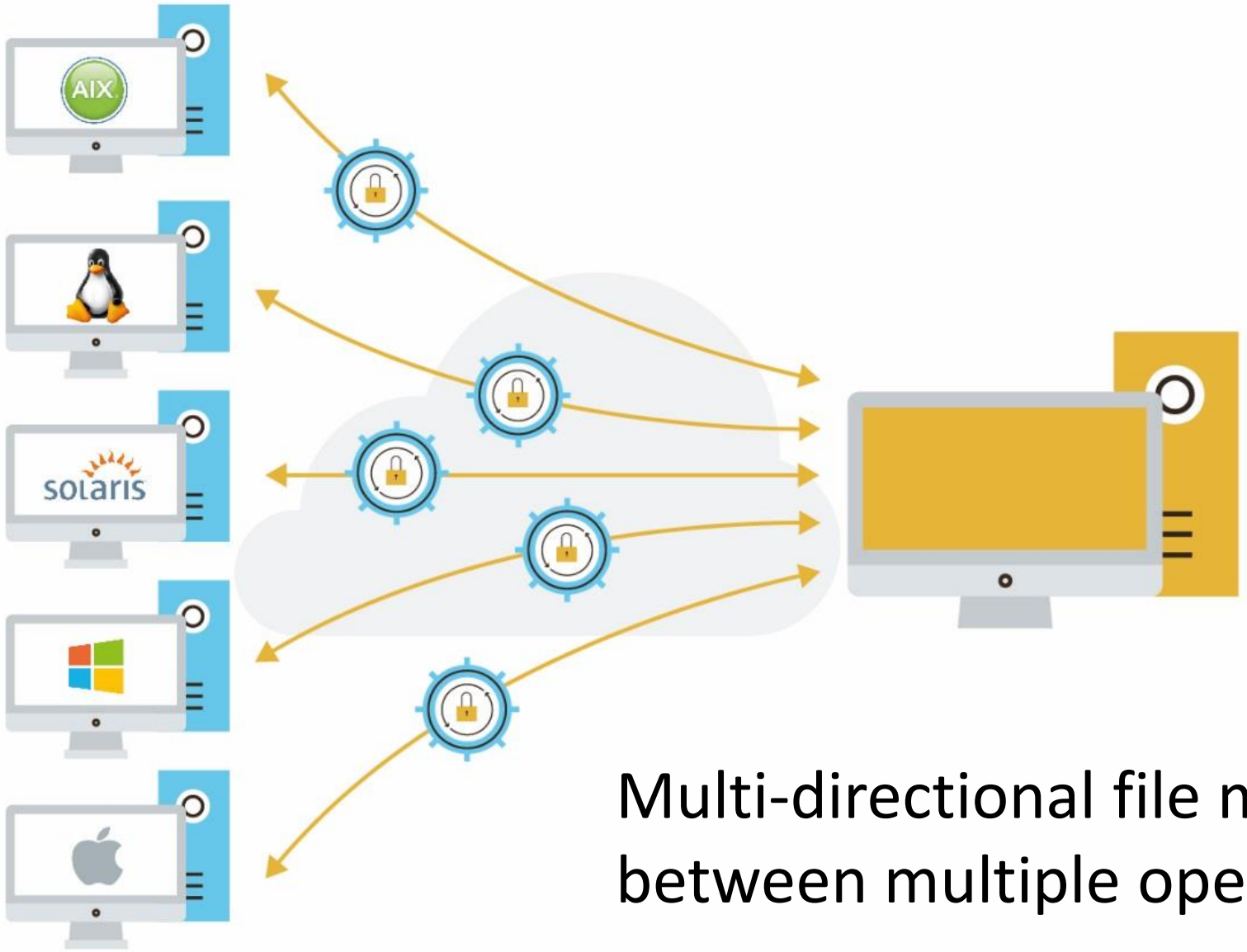
Leveraging data

Examples of topologies & how data is moved & synchronized

Bi-directional, real-time, file mirroring for Windows.

Bi-directional, real-time, file mirroring for LINUX.

ENDURADATA
PROTECTING, DELIVERING AND LEVERAGING DATA

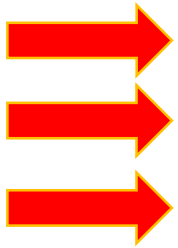


Multi-directional file mirroring between multiple operating systems.

ENDURADATA
PROTECTING, DELIVERING AND LEVERAGING DATA

What CROs are dealing with?

- { Cross border data movement } + { Sensitive data }



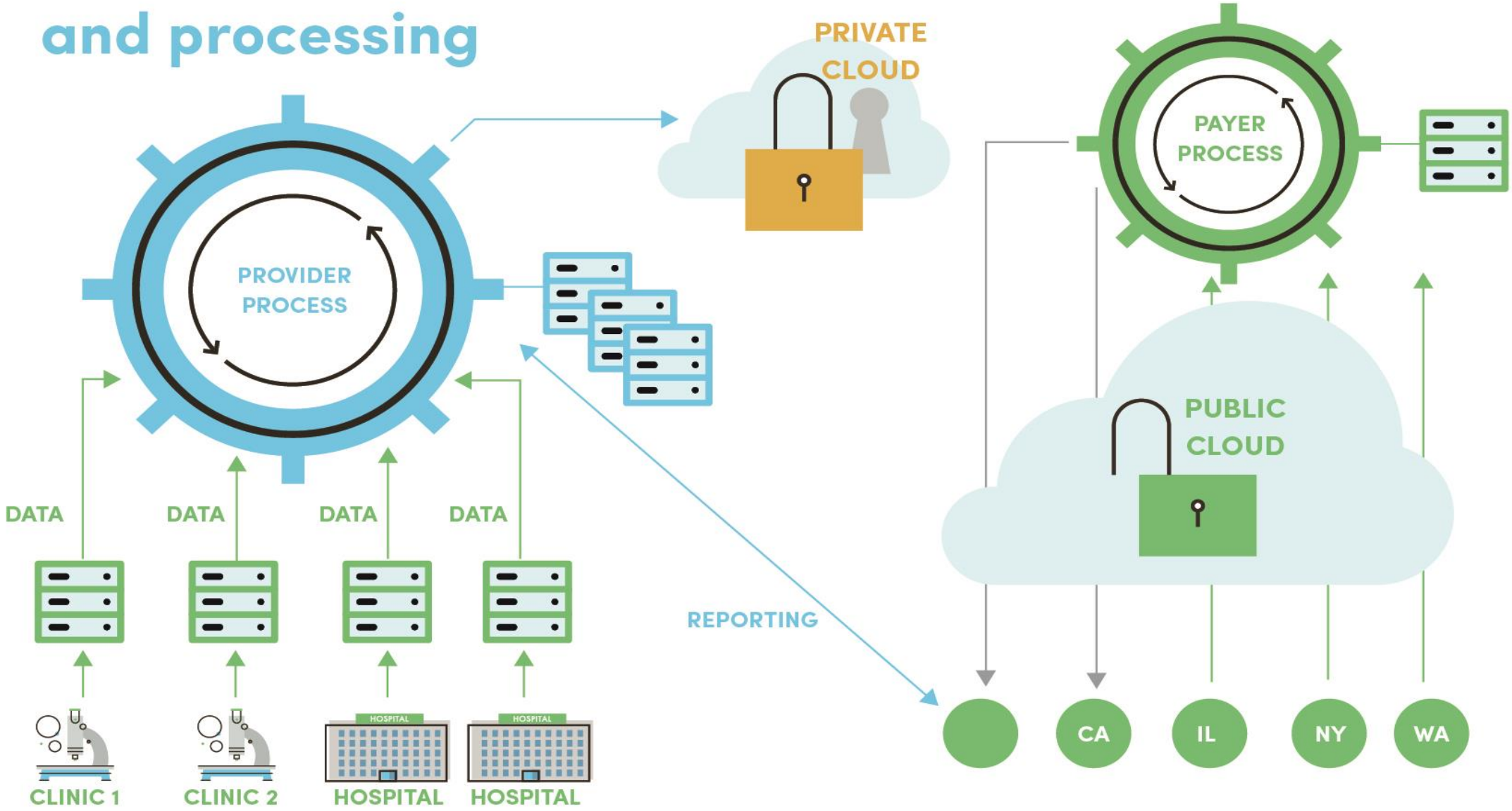
Anonymization
Strip personal info from everything
Replace all identifiable info

→ Part of Compliance

- Data leaks + Industrial espionage → Compliance + IP
- Data ingestion
- Process/Report/Communicate. → Compliance + IP + OPS

$$$$$

Healthcare:Automated data movement and processing

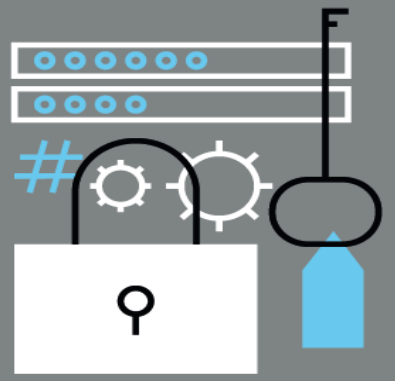


Security & Encryption

Data streams are encrypted (AES 128 by default)

Data can remain encrypted at rest

Other rules: Regex on both sender and receiver.



Multiple authentications:

- Hosts allowed
- Passwords for management
- Passwords for transport
- File encryption keys
- Transport encryption keys
- Link identification.

Bandwidth & Compression

Send only deltas
Monitor in real time

Bandwidth throttling
Parallel I/O

Adaptive compression

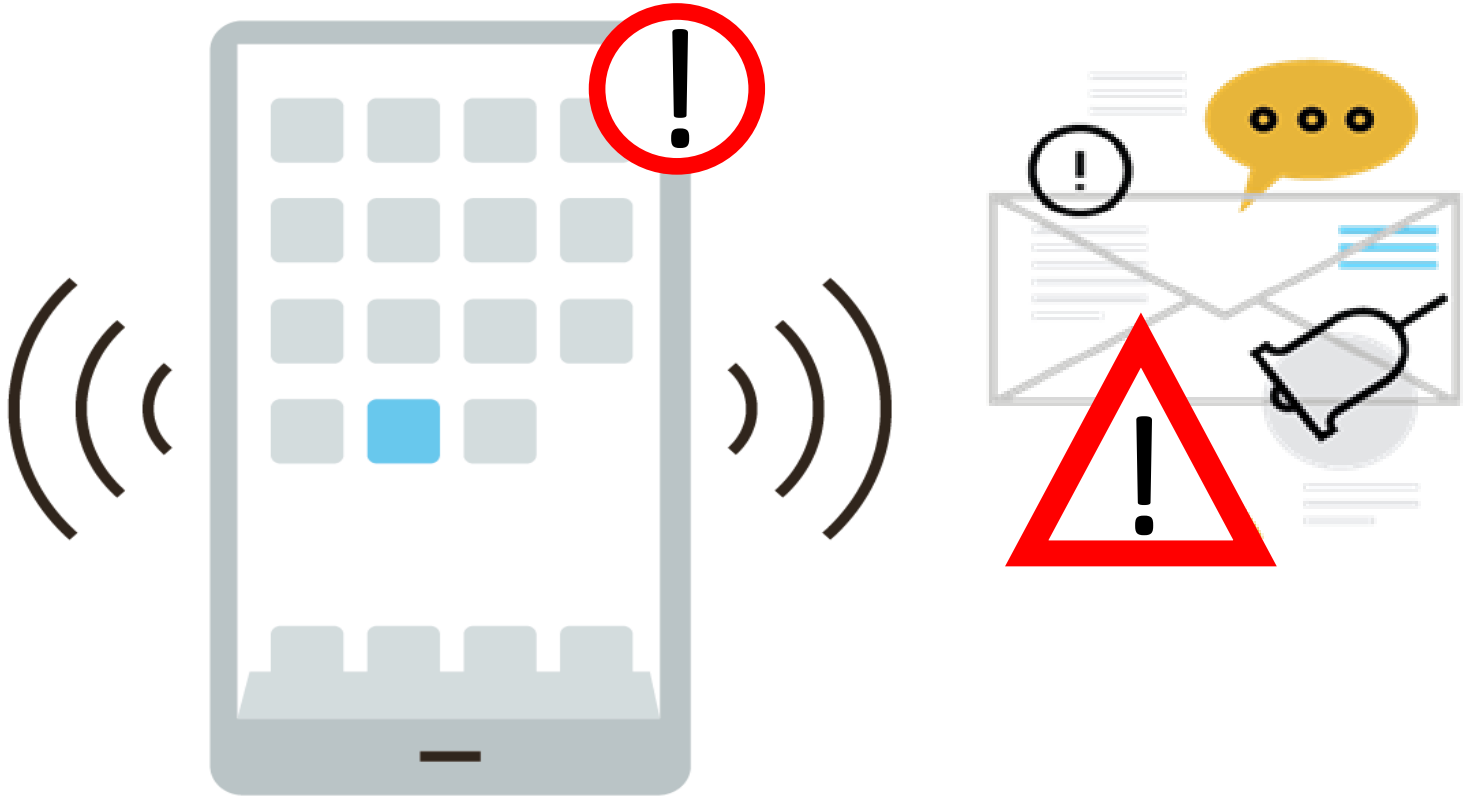
Pause, Resume
Caching, Deduplication

Email Alerts

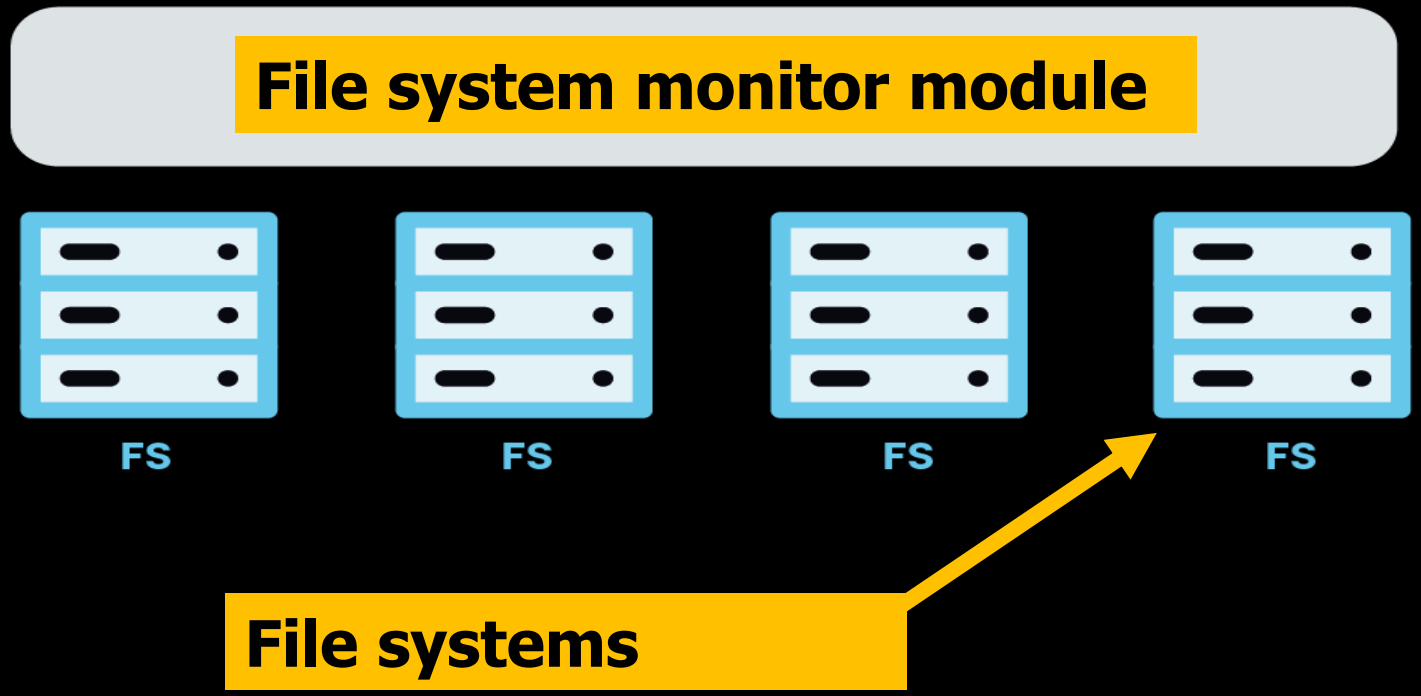
Notifications:

- **System problems**
- **Network problems??**
- **Disk space**
- **Failures.**

- **Alerts from sender**
- **Alerts from receiver**
- **Or from both.**



Sender side: real time module: Monitors data and metadata changes



Monitor I/O mutator operations:

- Write
- Truncate
- Chmod
- Chgrp
- Chown
- Rename
- Delete
- Symlink
- Mkdir
- Rmdir
- Change Attributes.

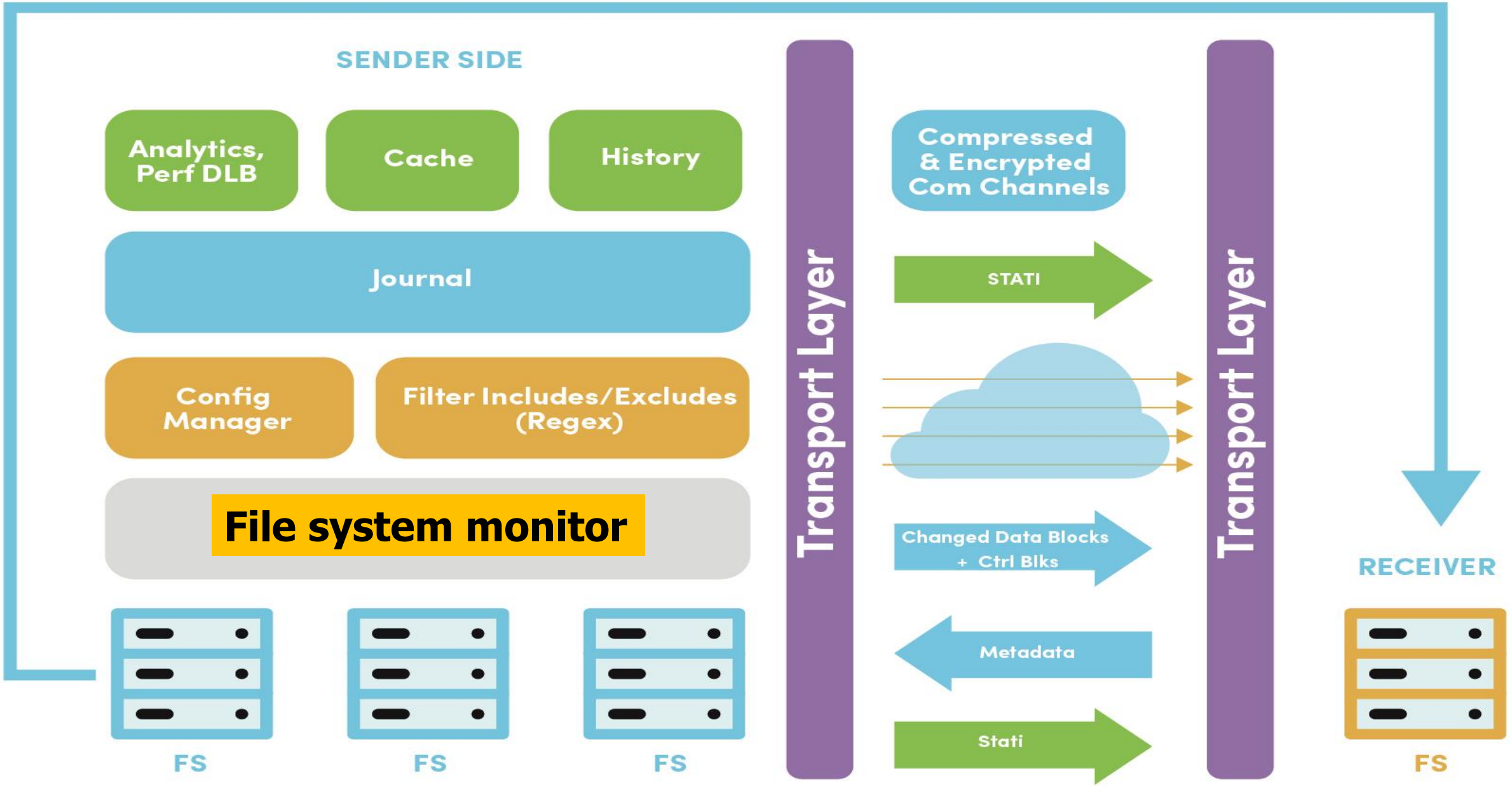
Deal with open files

Windows:

- VSS
- Snapshots

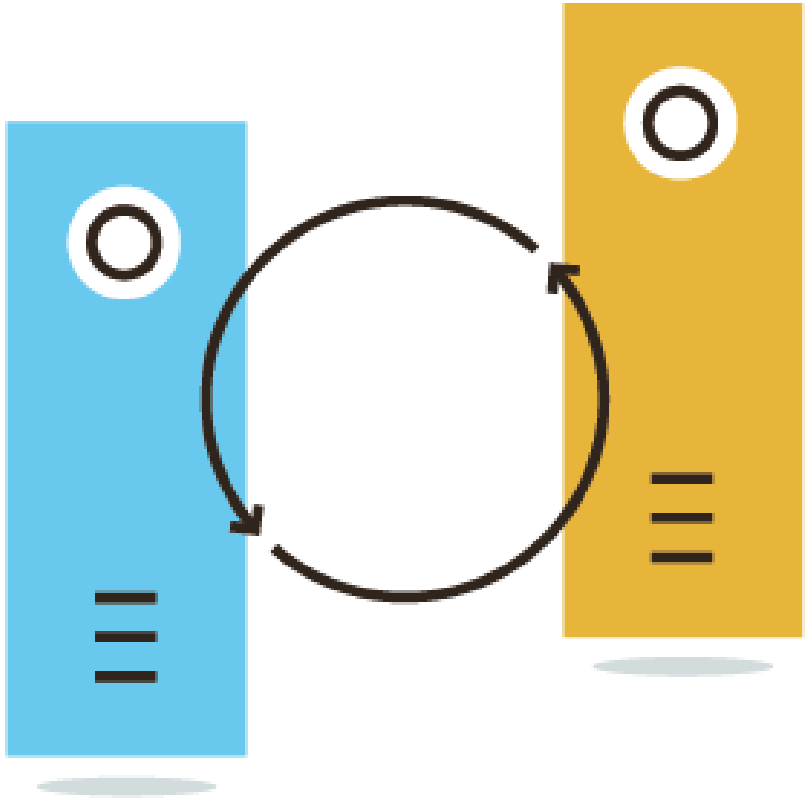
Re-queue changes

On the Windows side: exclusive locks


SENDER SIDE
Analytics, Perf DLB
Cache
History
Journal
Config Manager
Filter Includes/Excludes (Regex)
File system monitor
FS
FS
FS
Transport Layer
Compressed & Encrypted Com Channels
STATI
Changed Data Blocks + Ctrl Blks
Metadata
Stati
Transport Layer
RECEIVER
FS

Data Transfer:

- Use multiple streams
- Get info from journal
- Slice & dice to balance payloads
- Group compressible blocks
- Encrypt
- Consolidate
- …
- Send.



But you said you will talk about R?

Sender Post & Pre-processing

- FS module detects changes
- May invoke a pre-processing (i.e., R scripts, ….)
- Synchronize
- May invoke post-processing scripts.

Receiver Post & Pre-processing

- Receive request to sync
- May invoke a pre-processing (i.e., R scripts, ….)
- Synchronize
- May invoke post-processing scripts.

Example of using post processing with R

Reproducible research & communicating results

- Once data is delivered:
- Invoke post processing:
  - Ingests data into mysql, …
  - Uses R, markdown and Knitr
  - Generates reports automatically: PDF and HTML
  - Posts to internal web site & resyncs results to distribute
- For long term data: Used SAS (typically for GLM).

nature International weekly journal of science

NATURE | NEWS FEATURE

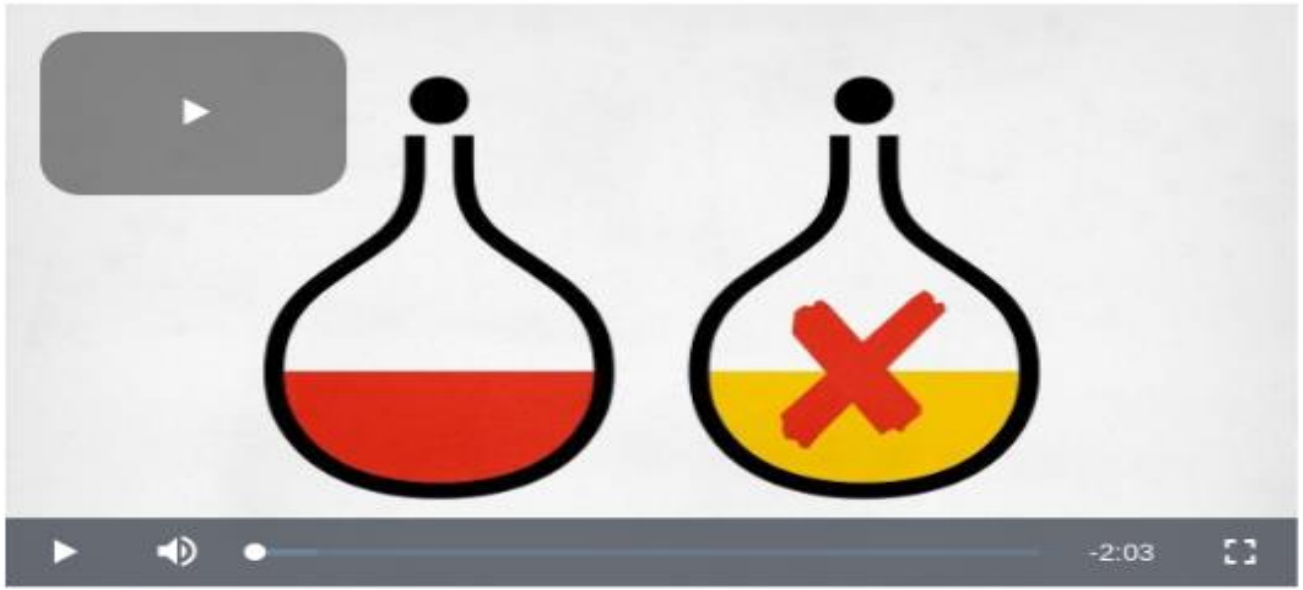
1,500 scientists lift the lid on reproducibility

Survey sheds light on the 'crisis' rocking research.

Monya Baker

25 May 2016 | Corrected: 28 July 2016

PDF Rights & Permissions



More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments. Those are some of the telling figures that emerged from *Nature*'s survey of 1,576 researchers who took a brief online

Reproducible research

Use Knitr && Markdown && avoid the "Bob" factor:

```
```{r  blockid}
Statement 1
Statement 2

...
Statement n
```
```

Reproducible research

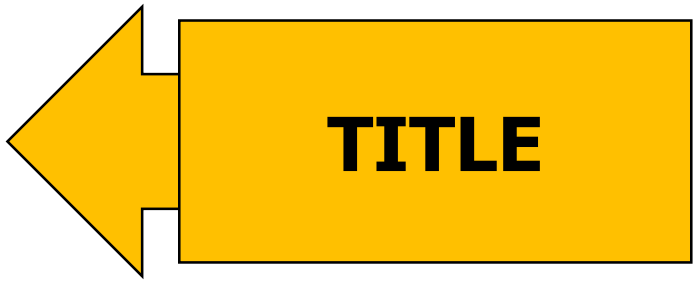
1. Save the code in a file rmed1.Rmd
2. Process to get the pdf & html output
3. Example of a bash script to process multiple RMDs.

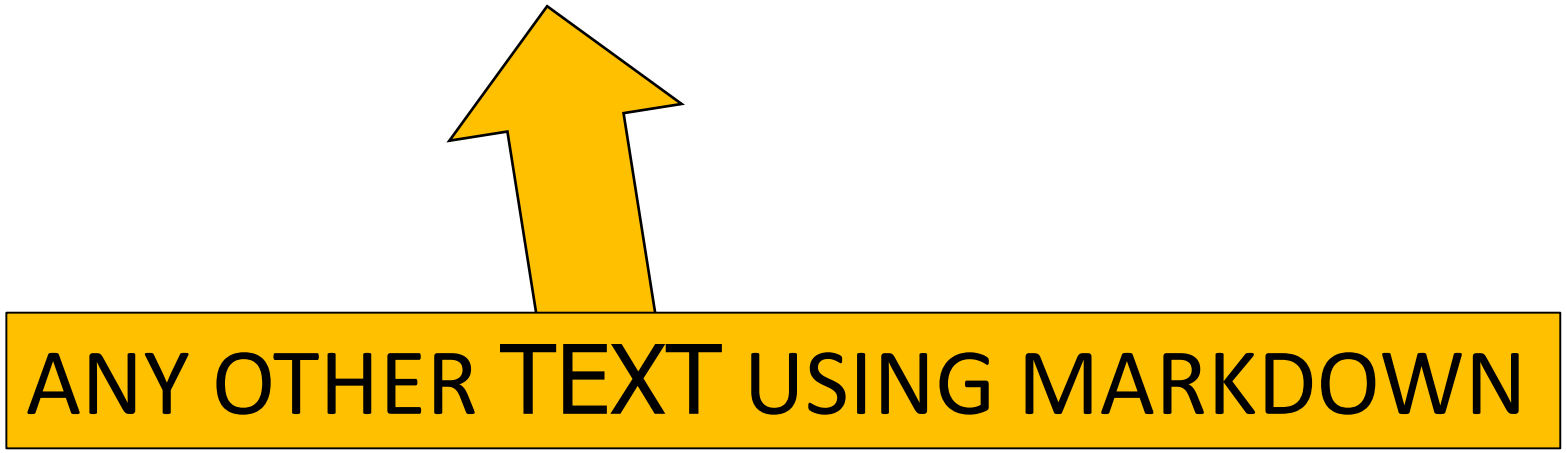
```
for f in $*
do
        R -e "rmarkdown::render('$f', c('html_document', 'pdf_document'))"
done
```

Example of R Markdown (part 1)


```
---
title:  Simulation of exponential distribution using rexp.
---
#1. Materials and methods

The simulation shows that as the sample size increases, the exponential distribution tends towards a normal distribution. We need to increase the sample size for a convergence of both the mean and the sigma to a standard normal N(0,1).
```





R Markdown (part 2)

....
#2. Simulation of the exponential distribution
```` {r simparams}
**knitr::opts_chunk$set(cache=FALSE)**
**lambda <- 0.2    #rate**
**theormean <- 1/lambda # Theoretical mean**
**u0 <- theormean;**
**theorsd <- 1/lambda   # Theoretical stdev**
**nobs <- 40        #number of observations**
**maxns <- 1000    #number of simulations**
````

* Theoretical mean: 1/lambda `r theormean`
* Theoretical stddev 1/lambda= `r theorsd`

SECTION HEADLINE

Your R code block

R Markdown (part 3)


```
```{r simulation}
rexpsimulation1 <- rexp(40, lambda);
mu1=mean(rexpsimulation1)
sdemp1 <- sd(rexpsimulation1)
varemp1 <- var(rexpsimulation1)

meanrexp <- NULL
for ( ns in 1:maxns ) { # Generate ns simulations each nobs observations using lambda as our rate
      rexpsimulation <- rexp(nobs, lambda);
      meanrexp <- c(meanrexp, mean(rexpsimulation))
}
```
```

← START R CODE Block

<--------- END R Code block

Sample: mean `r mu1` Empiric stddev = `r sdemp1` variance:`r varemp1 `
Notice that both sample mean and stdev are closer to the theoretical mean (`r theormean`) and standard dev (`r theorsd`)

R Markdown (part 4)

The differences will converge towards zero as the sample size increases later.

```` {r plotsim}
par(mfrow=c(1,2))
hist(rexpsimulation1, col="gray", xlab="rexp ", main="A. Exponential dist")

#theoretical mean
abline(v = c(theormean), col = "blue", lty=1,lw="2")
abline(v = c(mu1), col = "red", lty=1,lw="2") # set line dist mean sample1
mu2=mean(meanrexp)
sdsimmean = sd(meanrexp)
varsimmean = var(meanrexp)
hist(meanrexp, col="green", xlab="rexp simulation", main="B. Mean exponential dist")
abline(v = c(mu2), col = "red", lty=1,lw="2")
````

#3 Conclusions

When a file arrives, it starts the bash file to process the R scripts.

Here is an example of the output that gets replicated to all subscriber systems.

PDF or HTML output:

Simulation of exponential distribution using rexp.

1. Overview

The simulation shows that as the sample size increases the exponential distribution tends towards a normal distribution. We need to increase the sample size for a convergence of both the mean and the sigma to a standard normal N(0,1).

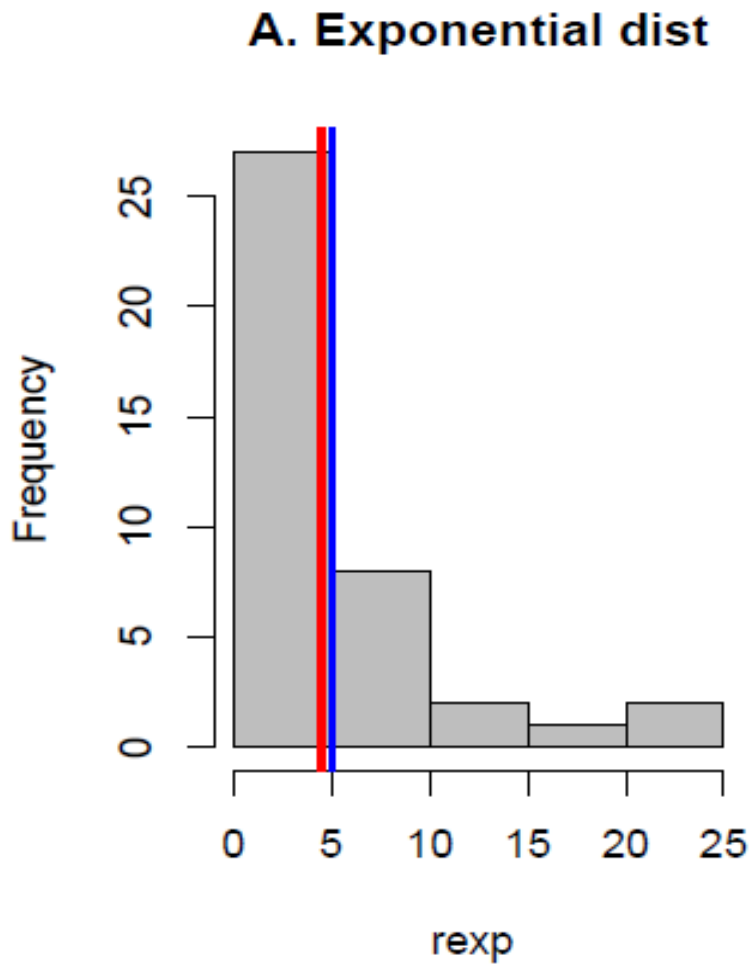
2. Simulation of the exponential distribution

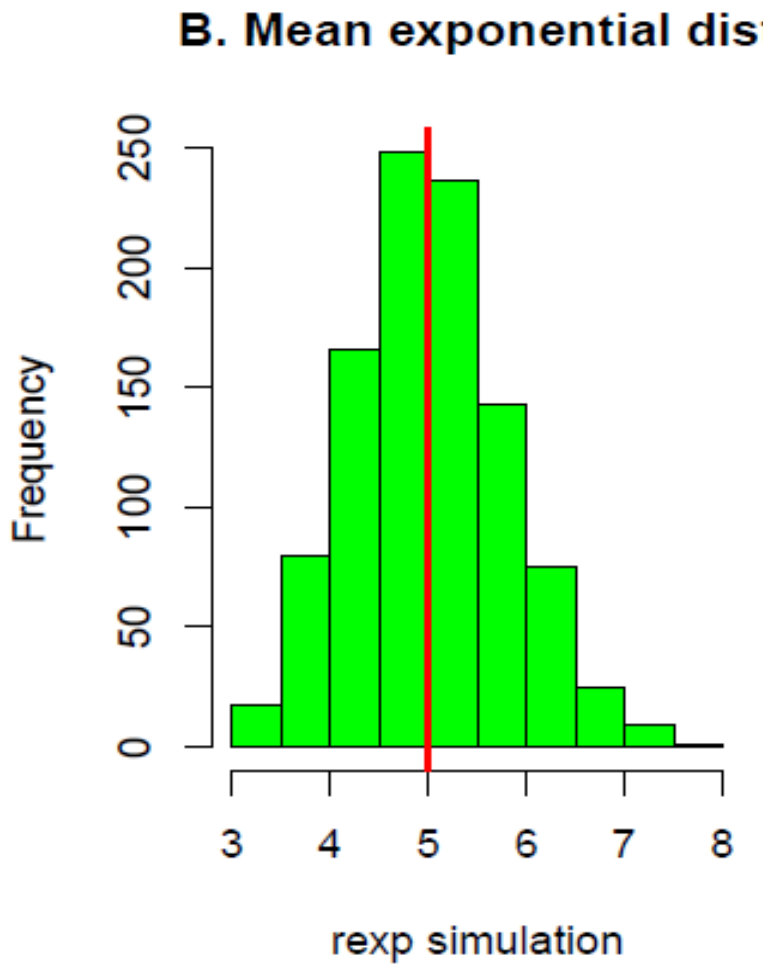
```
knitr::opts_chunk$set(cache=FALSE)
lambda <- 0.2    #rate
theormean <- 1/lambda # Theoretical mean
u0 <- theormean;
theorsd <- 1/lambda   # Theoretical stdev
nobs <- 40          #number of observations
maxns <- 1000     #number of simulations
```

Theoretical mean: 1/lambda 5 Theoretical stddev 1/lambda= 5

PDF or HTML output (continued) :


```
hist(meanrexp, col="green", xlab="rexp simulation", main="B. Mean exponential dist")
abline(v = c(mu2), col = "red", lty=1,lw="2")
```





3 Conclusions

TAKEAWAYS

- Combine data movement with data ingestion
- Automate all steps:
- Reproducible research:
  - ➢Must include all inputs
  - ➢Must include all transformations
  - ➢No manual edits
  - ➢Preserve the truth for validation

- Reduce risks, errors
- Reduce labor costs
- Reduce delivery time
- Monitoring
- Anyone can verify your work.

DO NOT DO THIS!!!

- Do not edit manually(i.e., vi, excel …..)
- Do not do one time manual changes: errors, no one can verify what you did.

Above all: Here is your kill switch:

- Delete all interactive data editing software
- No one can reproduce point and click.

Thank you.
Questions ?

elhaddi@enduradata.com

The Bob Factor:

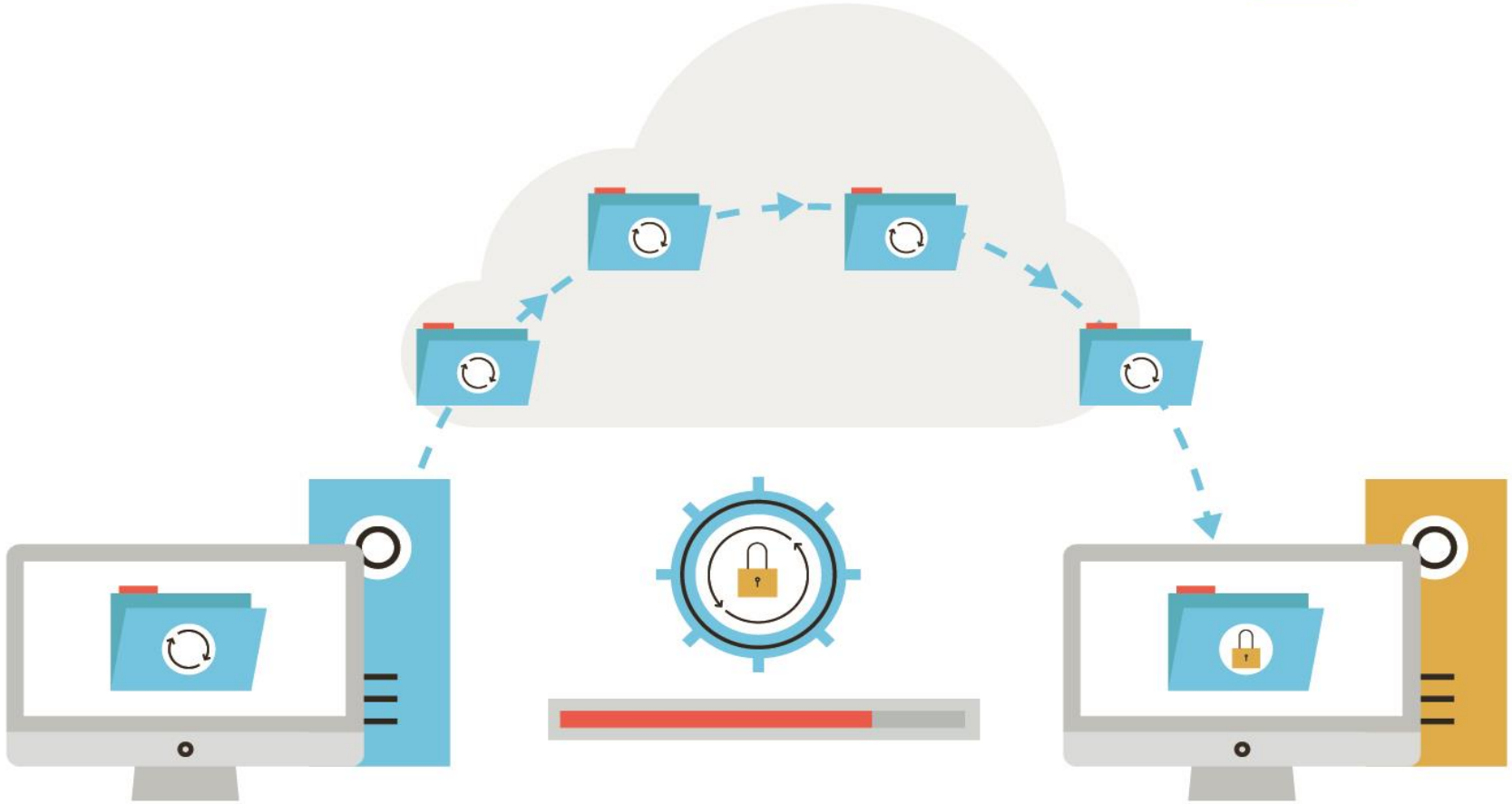
https://www.enduradata.com/real-time-sync-file-replication-videos/

Intentionally left blank

The end!

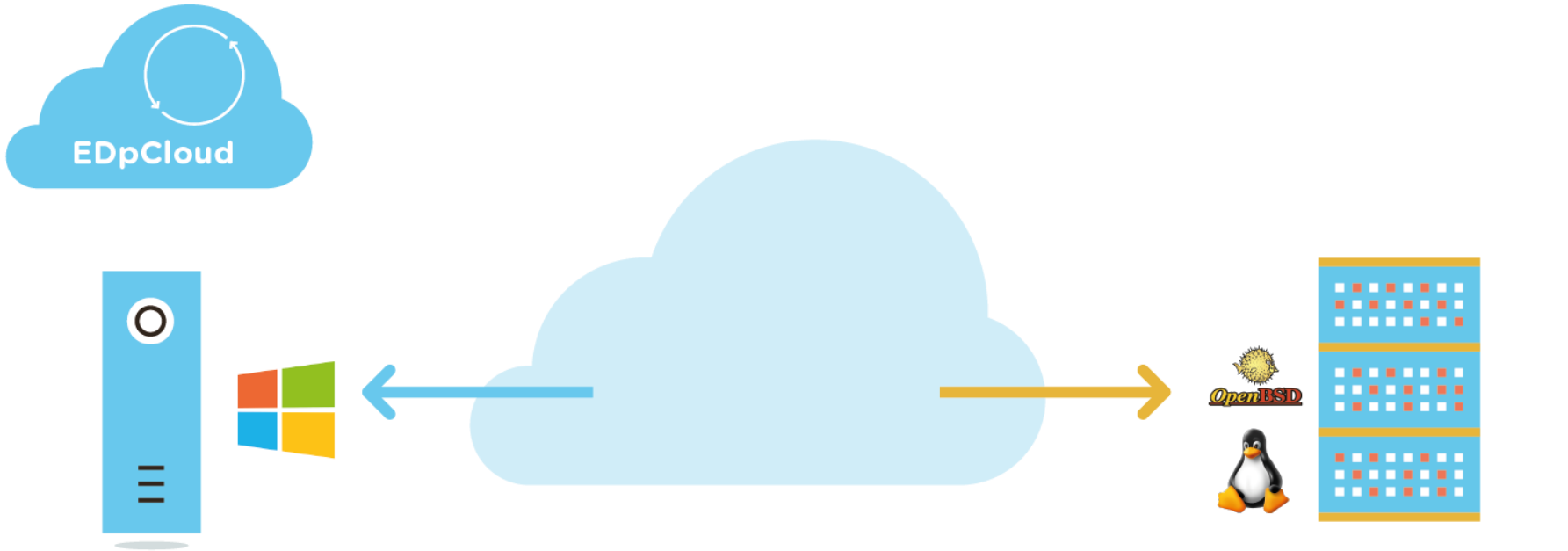
Additional Information

Appendix



Automatic & secure file replication between locations & systems.

Ransomware Protection with multi OS, versioning & isolation



DENVER

Replicate/Back up Data from Windows servers

ATLANTA

Save on/Restore from a LINUX/OpenBSD server

Other information

- https://www.enduradata.com/edwadds/data-synchronization-software-edpcloud-enduradata.pdf
- https://www.enduradata.com/edpcloud-data-synchronization-software-used-in-healthcare-information-exchange/
- https://www.enduradata.com/downloads/