



SPA  MDR  CDR  SCE 

End to End Data Flow

Achieving the Holy Grail of Clinical Trials --
Challenges and Opportunities

October 10, 2019



Abstract

The pharma industry has always lagged behind in adopting innovative technologies and systems for managing the flow of information in clinical trials. This talk will describe a case study on how the sponsor, software vendors and CROs and other key stakeholder working collaboratively deployed and brought into production an innovative end to end data handling and processing solution in record time and under budget. The learnings from this deployment and the use of the system for over two years will be presented.



Agenda

- Defining the problem of end-to-end data flow
- The Dataflow
- Securing the flow
- Murphy's Law & other Challenges
- What's next?



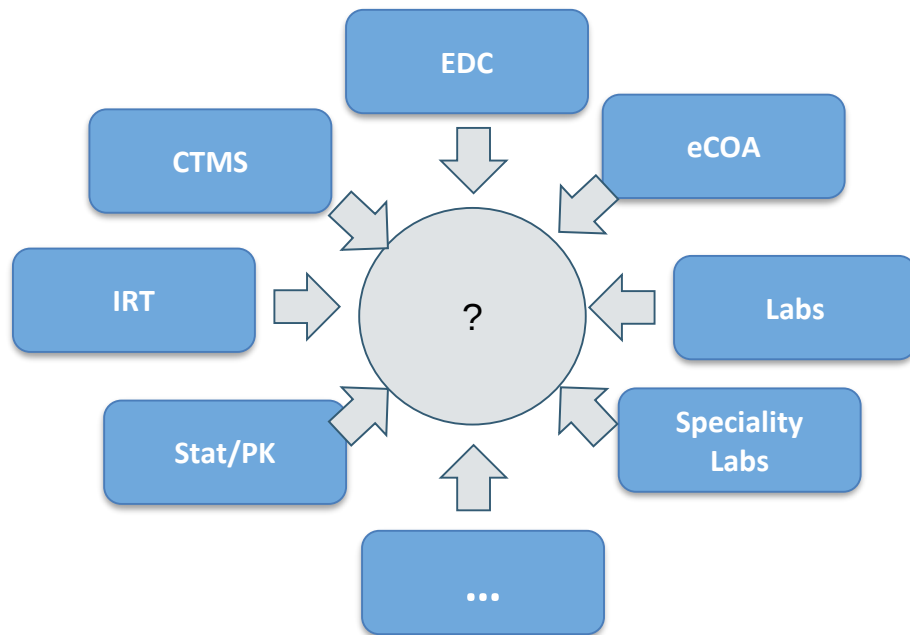
End-to-end Data Flow

Defining the Problem



Defining the Problem of End-to-end Data Flow

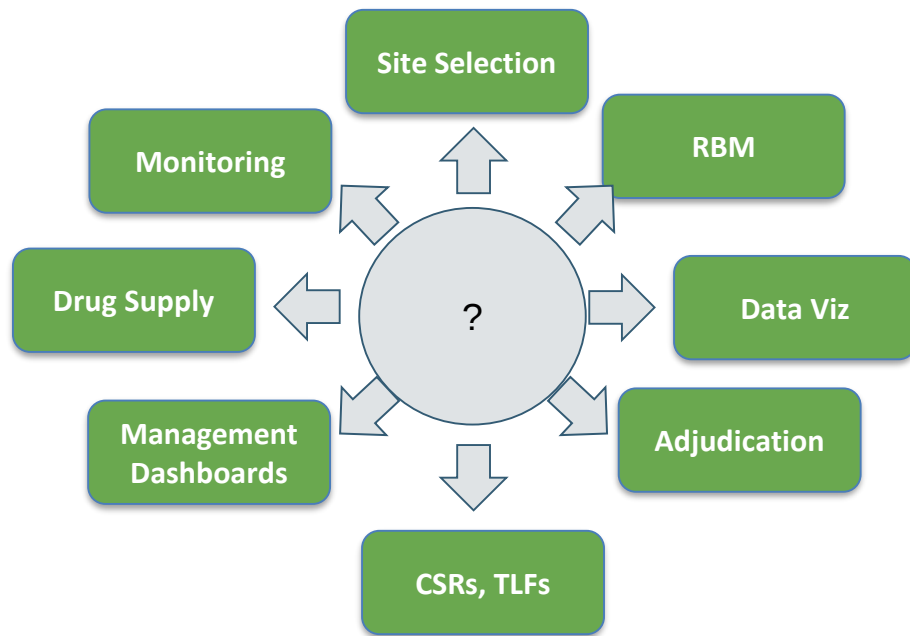
- Multiple Sources of Data
 - EDC
 - CTMS
 - RTSM
 - eCOA
 - Central Labs
 - Stat/PK
 - Speciality Labs



Defining the Problem of End-to-end Data Flow

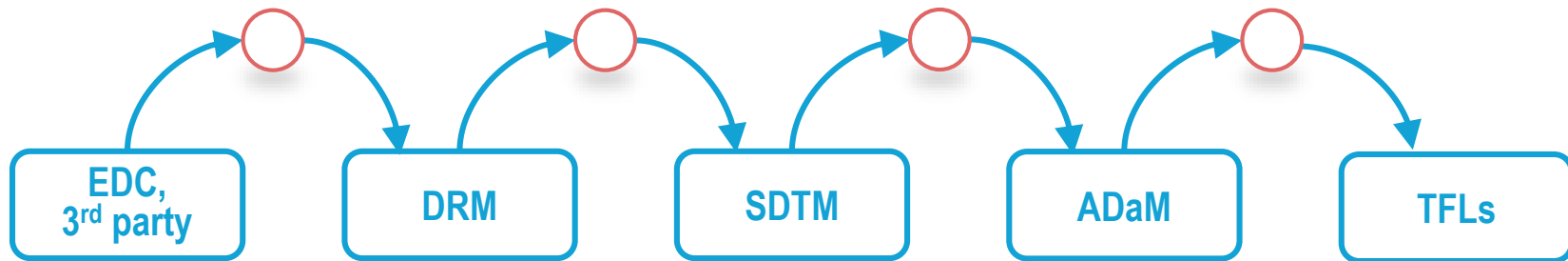
- Multiple Consumers of Data

- Site Selection
- Drug Supply
- Endpoint Adjudication
- Data Listings
- Risk based Monitoring
- Study Visualizations
- Management Reports
- TFLs for CSRs
- PK/PD Analysis
- Modeling and Simulation
- eTMF



Defining the Problem of End-to-end Data Flow

- Coordination between Sources and Consumers
 - Raw data from EDC to SDTM vendor
 - SDTM Data from Vendor
 - Passed to Visualization
 - Passed to RBM
 - Generate ADaM



Defining the Problem of End-to-end Data Flow

- Different Tools/Systems
 - Rave EDC SAS on Demand
 - SAS
 - sFTP
 - Spotfire
 - SAS VA
 - Pinnacle 21
 - Data Check Programs
 - eTMF



Defining the Problem of End-to-end Data Flow

- Given that we have ...
- Different sources
- Different consumers
- Dependencies between sources and consumers
- Different tools
- and...
- **All want their data and reports “now*”**

How do you make it happen?



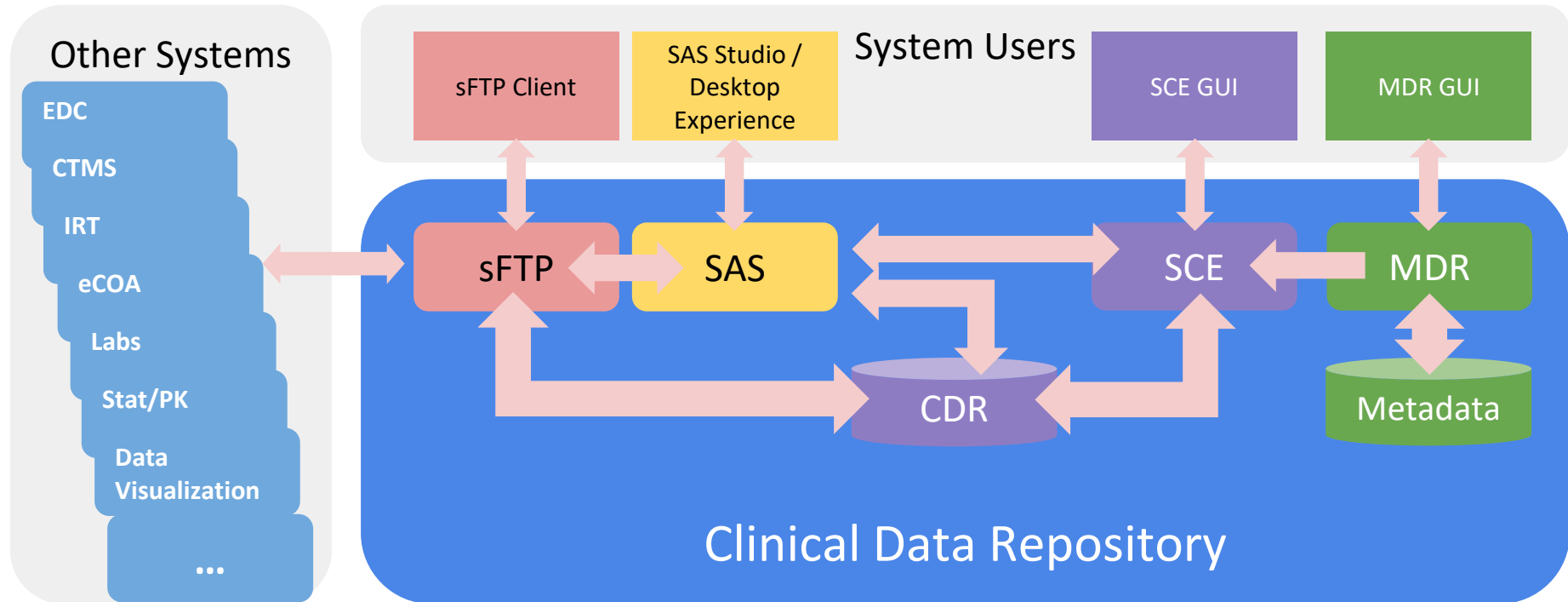
The Data Flow

Architecture of the flow

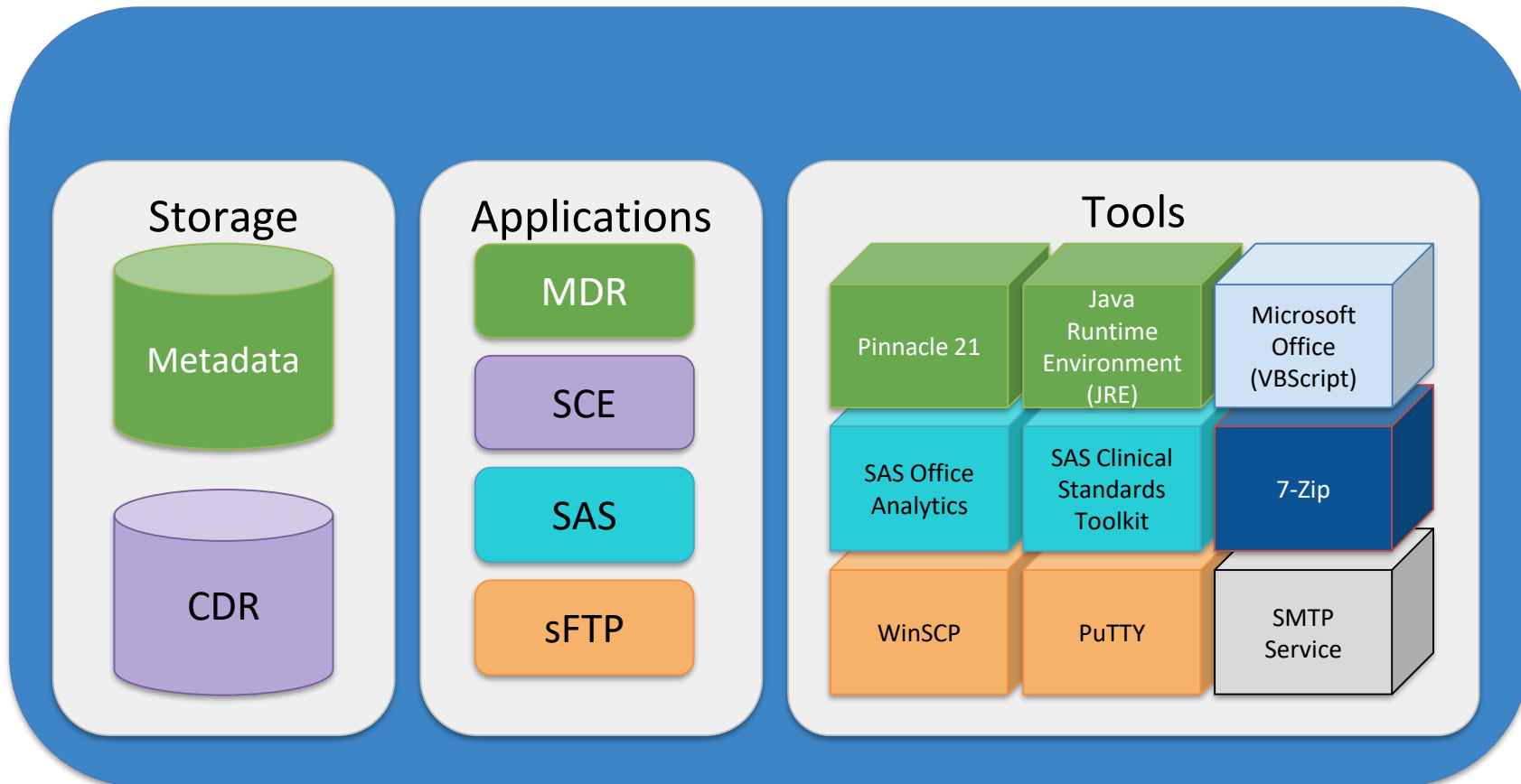


Clinical Dataflow

External and Internal Data Providers and Consumers



Clinical Data Repository Platform



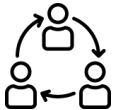
Component: Metadata Repository



Capture metadata standards including transformations/mappings



Bulk import or cherry pick from standards into study metadata for consumption



Collaboration on Metadata specification and data mismatches



Import from key or legacy systems / Export for CDR integration



Capture traceability, versioning, & lifecycle



Extend metadata, configure workflows, and integrate with CDR & SCE



Component: Clinical Data Repository



Standardize handling of trial data, program files, and data sets (collection to locking to archiving)



Handle any clinical data model cumulatively or incrementally



Run programs & batch jobs with traceable dependencies & automated re-runs upon update



Program in-app or from MS Windows desktop with versioning, diff tools, & role-based permissions



Easily define and manage role-based access, blinding, sequestering, and partner access



Reliably audit access & changes across all functions



Component: Statistical Computing Environment



Coordinate tasks, track work and enforce processes within the system



Maintain traceability and lineage across programs, data and outputs with version control



Program in-app or from MS Windows desktop with versioning, DIFF tools, & role-based permissions



Easily define and manage role-based access, blinding, sequestering, and partner access



Reliably audit access, changes and usage across all functions



Components: Other Tools & Systems



Secure File Transfer with FIPS-140 compliance



Transfer watcher and synchronization



Secure download



Notification and Alerting



Triggered Execution



Development Area/Workspaces



Securing Data Flow

Security, Authentication, Authorization



Securing Data Flow

sFTP Transfer

- FIPS-140 compliance
- Limited, secure set of cyphers
- Limited, secure set of key exchange algorithms



Securing Data Flow

Use of System Accounts

- sFTP and API users
- Should be Identified users
- Password profiles for long term and complexity



Securing Data Flow

APIs based interactions

- Token-based authorization
- Secure based on application rights definition
- Use secure encryption (no SSL or TLS 1.0) TLS 1.1 or higher



Securing Data Flow

Secured Analysis of Data

- Role and folder based security
- Blinded vs unblinding access
- Securing Data from all access Modalities
 - Securing Filesystems
 - Securing from the APP UI, and
 - Securing RDBMS



Securing Data Flow

Auditing Data Access

- Application Access
- File-system access auditing
- Auditing and Periodic Review
- Over all modalities



Murphy's Laws and other Challenges

What can go wrong will go ...



Challenges

Coordination and triggering of processing

- Multiple runs should be idempotent
- Timing and coordination of dependent Dataflows
- Triggering and Batch Triggering
- Write data checks



Challenges

sFTP Transfers

- Support for FIPS 140 and Cypher KEX issues
- Cloud-based blue-green deployment and IP/fingerprint consistency
- Coordination of different vendors & technology



Challenges

Data Volume and Processing needs

- Incremental vs Snapshot Processing
- Data sources don't always support incremental/transactional flow
- Need to track rows that were processed



Challenges

Faster, more frequent Transfer Cycles

- “Near Real Time” Visualizations and Dashboards
- Processing of complete data vs incremental
- Need for horizontal scale
- Fast processors and NVME disks
- Dozens of GB per transfer cycle



Opportunities

Things that can be better



Opportunities

Horizontal Scaling

- License cost can be limiting factor
- R and Python has excellent data processing libraries
- Can be run on multiple server



Opportunities

Handling Frequent update of Slowly Changing data

- Efficient storage of data using SCD-II
- Have up-to-date data yet be able to reproduce the data as of a particular date and time
- Relational Database with record versioning



Opportunities

Dealing with truly large data, transparently

- Zeta scale file systems
- In memory databases (SAS VA on Viya?)
- Horizontally-scaled RDBMS structure
- S3-style object storage



Questions?

