



Content

- Introduction and definition
- Forward and Backward Traceability
- How to realize file level traceability
- Documentation and checks
- Conclusion



Traceability - Definition

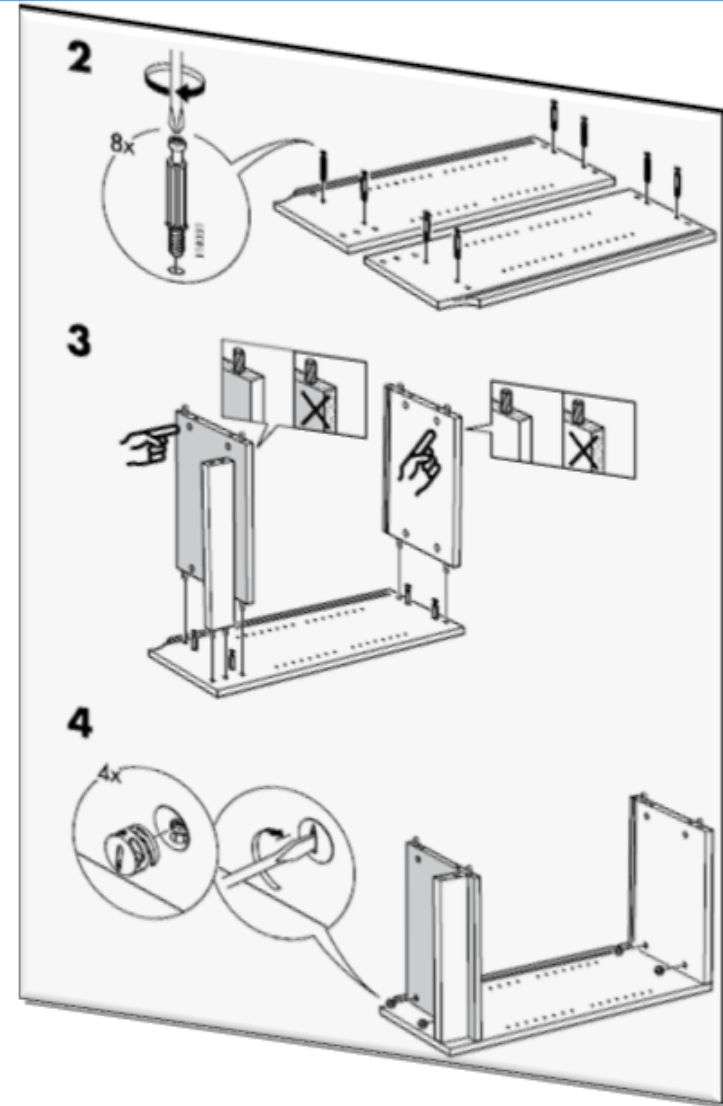
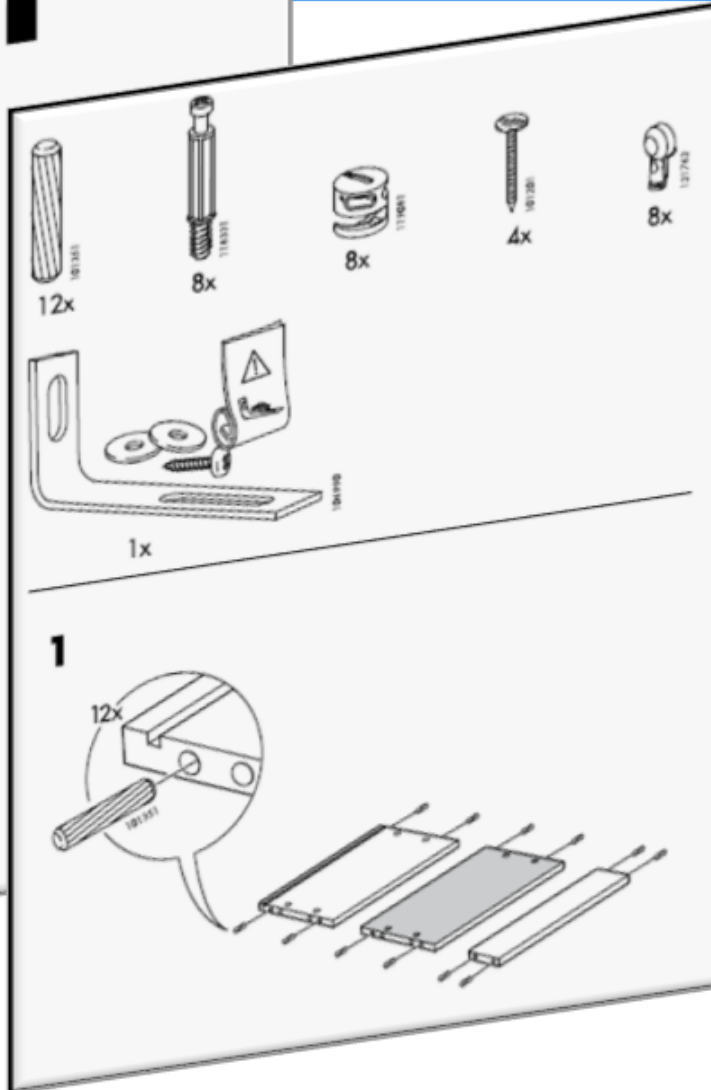
“Traceability refers to the completeness of the information about every step in a process chain.” (Wikipedia)

- Data becomes traceable if sequence of source data → transformation logic → target data is available for the whole study
- If Traceability Metadata are available for a whole study, it should be possible to reproduce the study results.
- Traceability on different level:
 - File level → focus in this presentation
 - Data set level
 - Data item level

BILLY

The diagram illustrates the assembly of a Billy bookshelf. On the left, a perspective view shows a partially assembled unit with four shelves. On the right, a detailed callout shows the components for attaching a shelf: a wooden dowel (12x), a metal screw (8x), a metal bracket (1x), and a warning triangle. Below this, a circular inset shows a close-up of the wooden dowel being inserted into a pre-drilled hole in the shelf, with a screwdriver being used to tighten the screw.

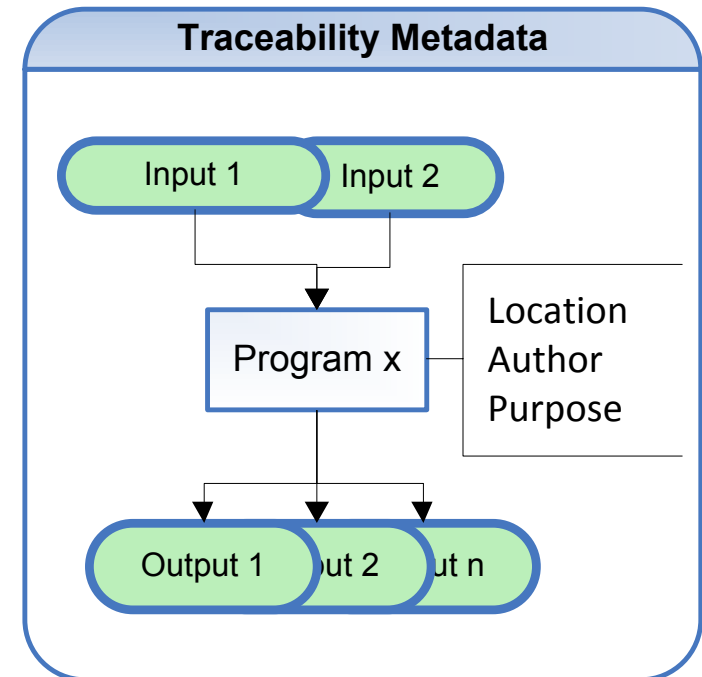
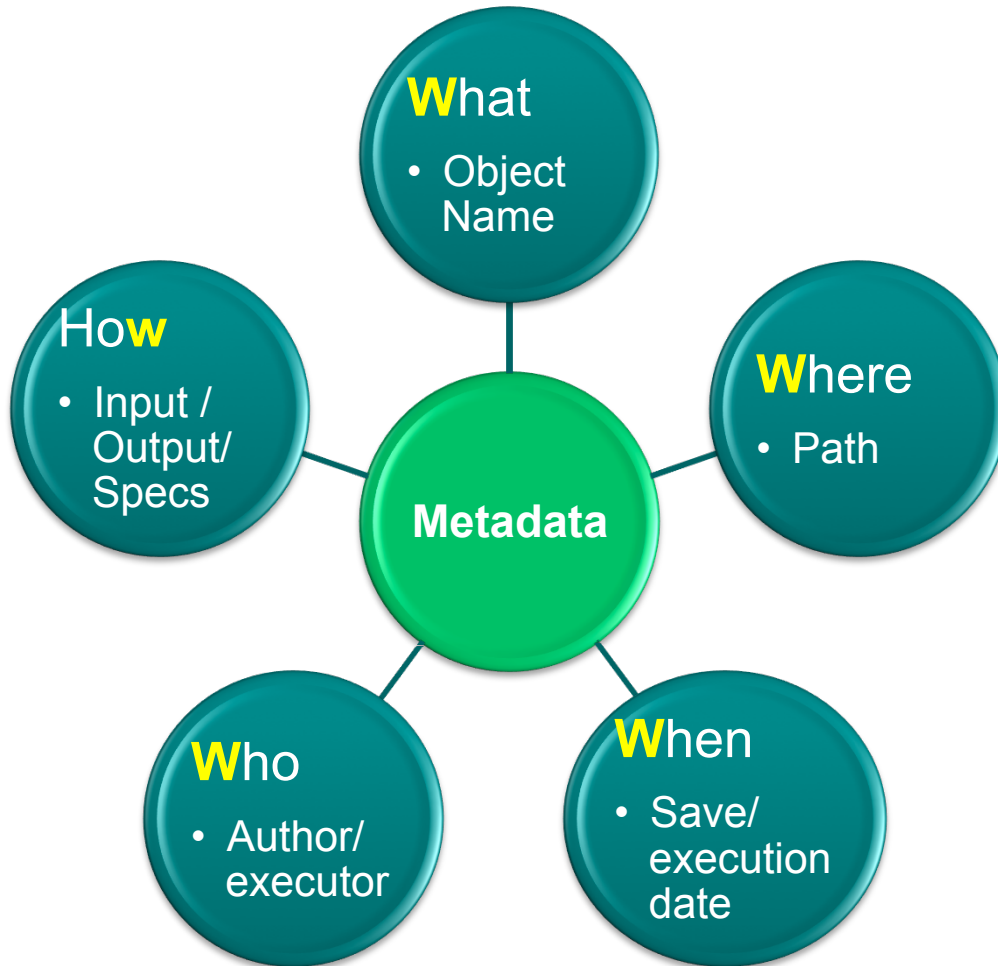
Traceability illustrated

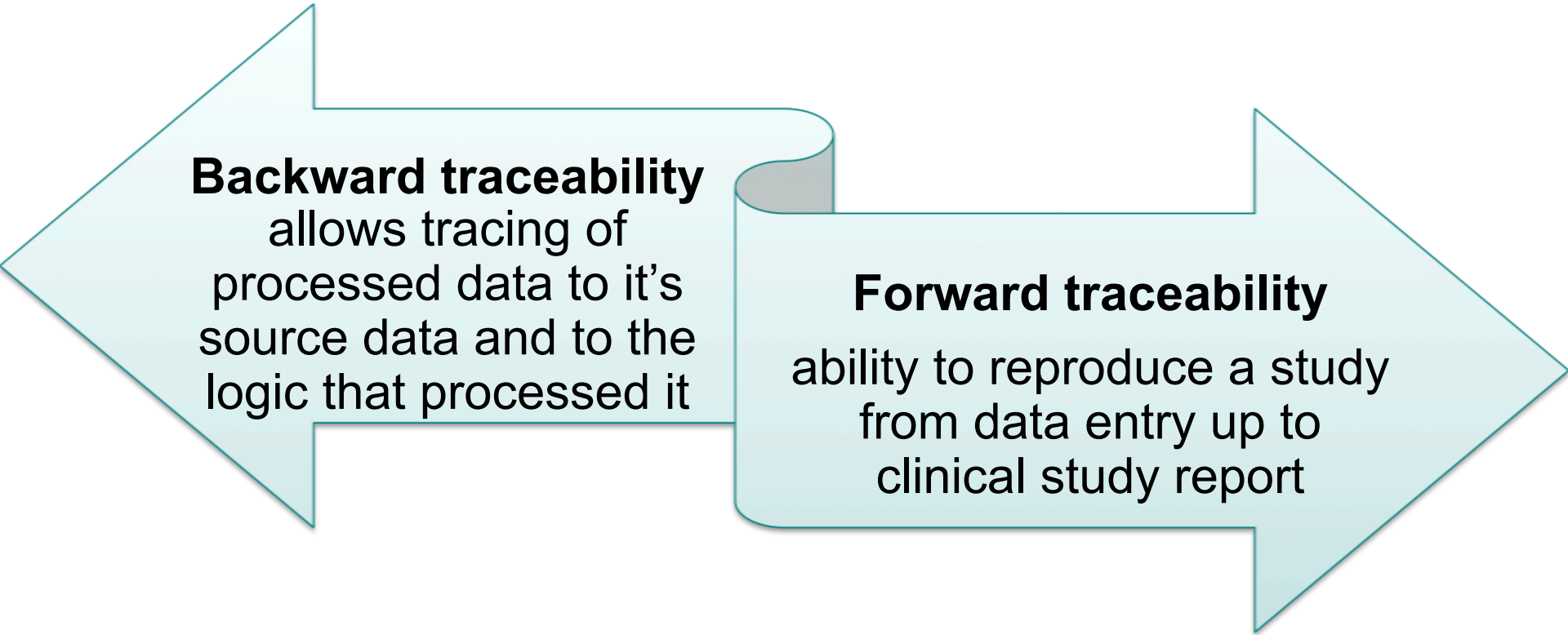


Metadata



The following metadata are needed to ensure traceability of an object (“W” questions):



A diagram illustrating the concepts of backward and forward traceability. It features two large, light blue arrows pointing in opposite directions. The left arrow points to the left and contains text about backward traceability. The right arrow points to the right and contains text about forward traceability. The two arrows are connected at their inner ends by a grey, rounded rectangular shape that resembles a folded piece of paper or a bridge.

Backward traceability

allows tracing of
processed data to its
source data and to the
logic that processed it

Forward traceability

ability to reproduce a study
from data entry up to
clinical study report

Why traceability?



- All electronic data processing steps are reproducible
- Comprehensive and submission documentation for HA reviewers
- Answer authority questions quickly
- Efficient and fast investigations even years after study reporting
- Working together in teams
- High quality

An unexpected urgent inquiry

A typical atypical case



Table 8-23: Duration of medication A as pre-medication of the index pulmonary embolism before randomization (ITT population)

	Drug 1 n = 419 (100%)	Drug 2 n = 2413 (100%)
Duration pre-medication of the index (categorical)		
No pre-medication	82	17
1 day	17	46
2 days		
>2 days		
Duration of (LMW) pre-medication of the index (days)		
Mean		
Standard deviation		
Minimum		
Maximum		

Request about a very old analysis:

- We are trying to understand what CRF was used for this table?
- We cant figure out the maximum duration (last row)
- Can we get the patient numbers of Drug 2 >2days?

3

Who the heck did this?

Why me???

Where do I find the program?

What are the algorithms that were used?

What was the RAW data that the analysis was derived from?

From what year is this? Was I even in the company?



Question: How do I find the program?



Table 8-23: Duration of medication A as pre-medication of the index pulmonary embolism before randomization (ITT population)

	Drug 1 n = 419 (100%)	Drug 2 n = 2413 (100%)
Duration pre-medication of the index (categorical)		
No pre-medication	182 (7.5%)	190 (7.9%)
1 day	1754 (72.5%)	60 (2.9%)
2 days	460 (19.0%)	43 (1.8%)
>2 days	23 (1.0%)	20 (0.8%)
Duration of (LMW) pre-medication of the index (days)		
Mean	1.14	
Standard deviation	1.56	
Minimum	0.0	
Maximum	7.0	

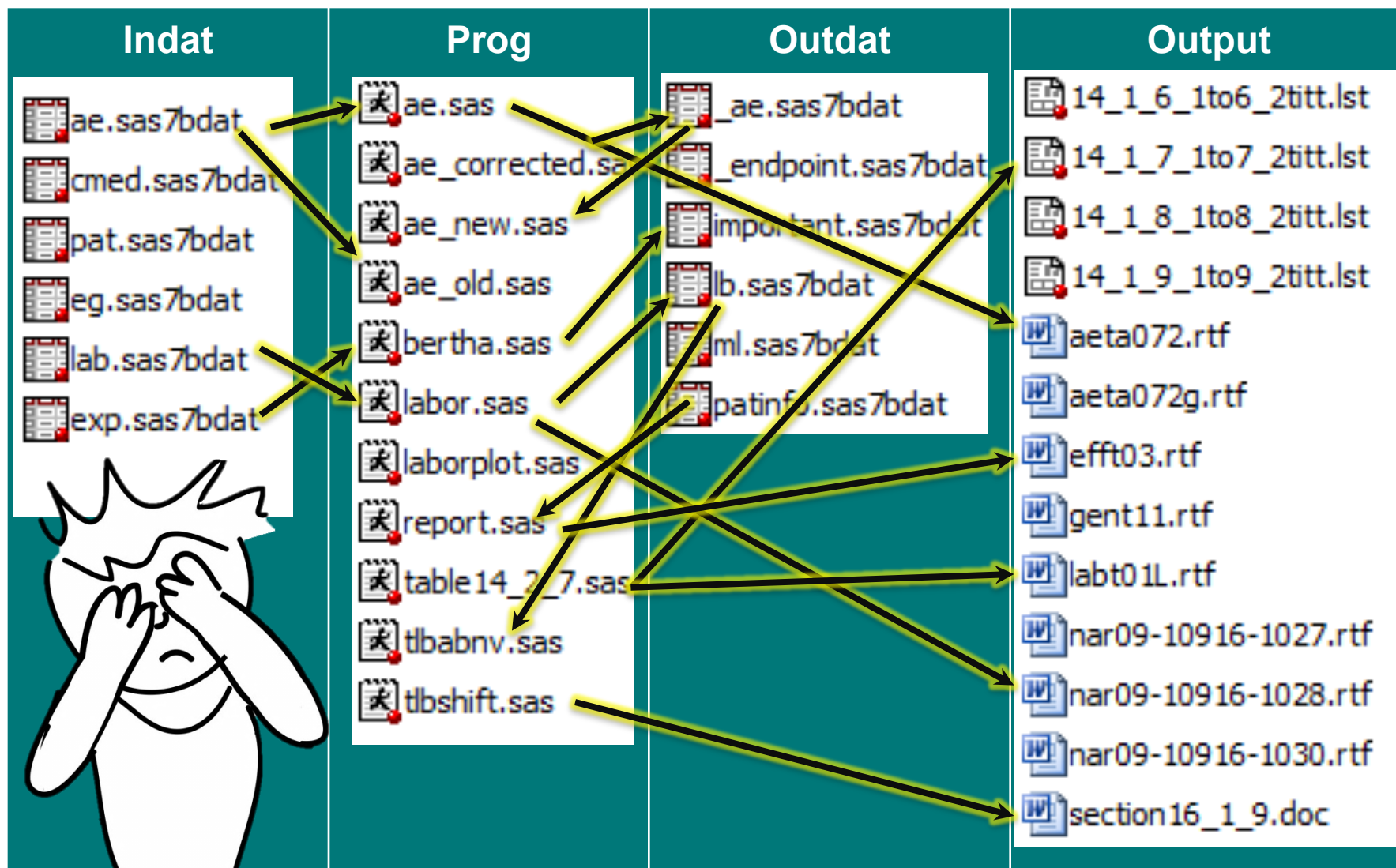
Lost without a trace!



- Each Output file (table/ listing/ figure) ALWAYS has to have an ID footer with:
 - ↳ A Keyword so that tools can identify the ID footer
 - ↳ SAS program name and path
 - ↳ program status (draft/final)
 - ↳ user id (optional)
 - ↳ date and time stamp
 - Graphics: create vector graphic files instead of pixel graphics
 - footer and title are still readable by tools
 - RTF files: ID footer is not stored in the document footer but in the document body
 - Footer has to stay with the output everywhere it is used
- Quickly trace back to SAS program. Even after years



Analysis file naming



Analysis file naming (2)



PROGRAM NAMING CONVENTION

- Meaningful program names – so that others can make use of it
→ **this improves team work and helps reviewers**
- 1:1 - Program name = output file name= output data set = log file name
1:n - Program name = beginning of output file name
→ **this ensures a quite good quality of Output File Traceability**
- Define SAS macro variable &progrname and use it whenever output file name have to be specified or referenced
→ **this ensures consistency regardless of potential program name changes**
- Use CDISC domain abbreviations (AE, CM, DM, ...) in program names
- More company internal naming rules as deemed helpful

Program header



```

/*****
*** study: 12345/ project:      ****?*??
*** program name: table-ae.sas  ****?*
*** specs:      ???           ****?*
*** validation: May 2012      ****?*
*** modified:   Oct 2012      ****?*
*****/
%let proname = table-ae-corrected.sas;
  
```

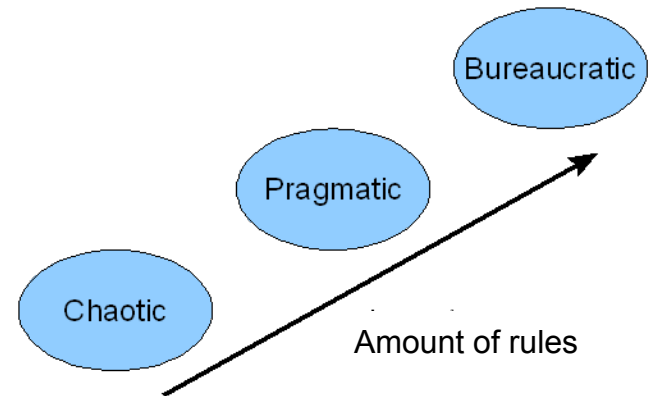
author???

Program header (2)

- The only thing worse than no header is a wrong header
- A correct and complete header can be quite a task and it's known to be not a favorite job of a programmer
- Typical header metadata are: *Author, purpose, SAS version, source data, output data, external macros, output files (ASCII, graphics....), global macro variables, options changed, validation level*
- **Fat or thin program header?**

For the sake of reliable header:

- Reduced to what is really useful
- No redundant information
 - source of inconsistencies
 - work to collect
 - unless automatically filled by the program editor



Program header (3)



```
/** *****  
*** study: 12345/ project: *****  
*** program name: table-ae.sas *****  
*** specs: *****  
*** validation: May 2012 *****  
*** modified: Oct 2012 *****  
***** */
```

```
%let proname = table-ae-corrected.sas;
```

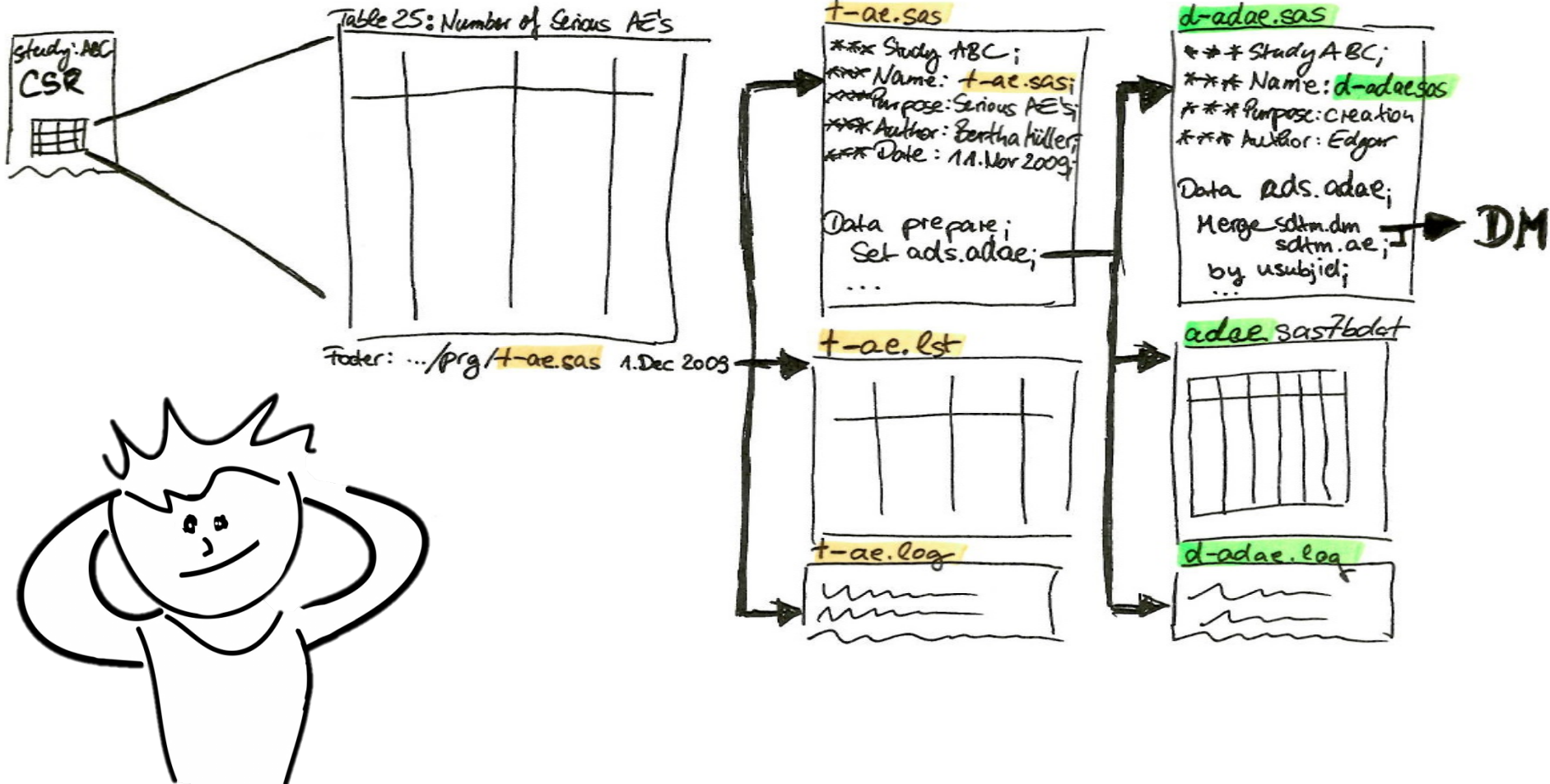
```
/** *****  
*** study: 12345/ project: phuse *****/  
%let proname = table-ae.sas;  
/** specs: adverse events table *****  
*** author: bertha *****  
*** date: Oct 2012 *****  
***** */
```

Backward Traceability

...programming rule based



Backward Traceability

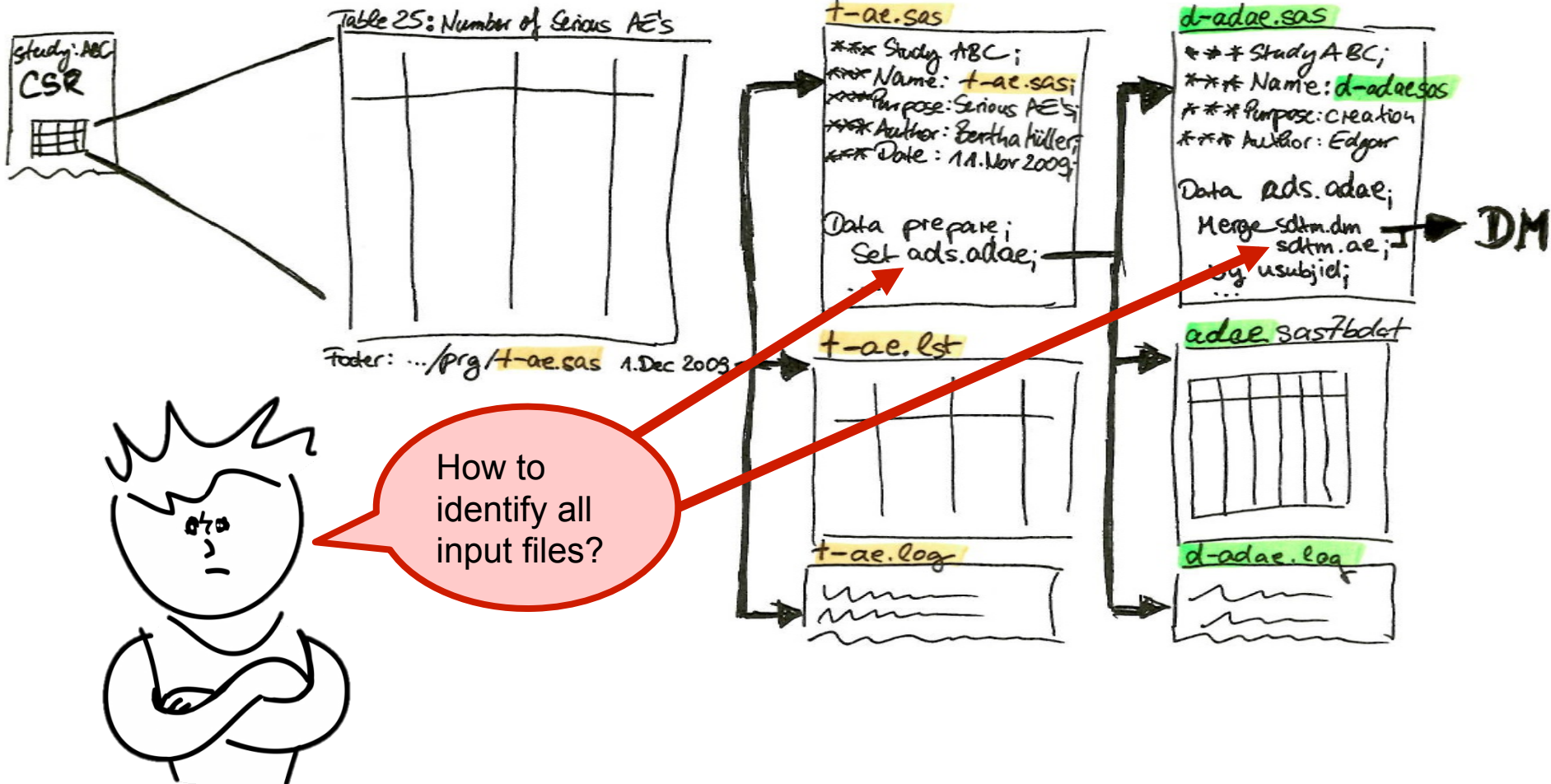


Backward Traceability

...programming rule based



Backward Traceability

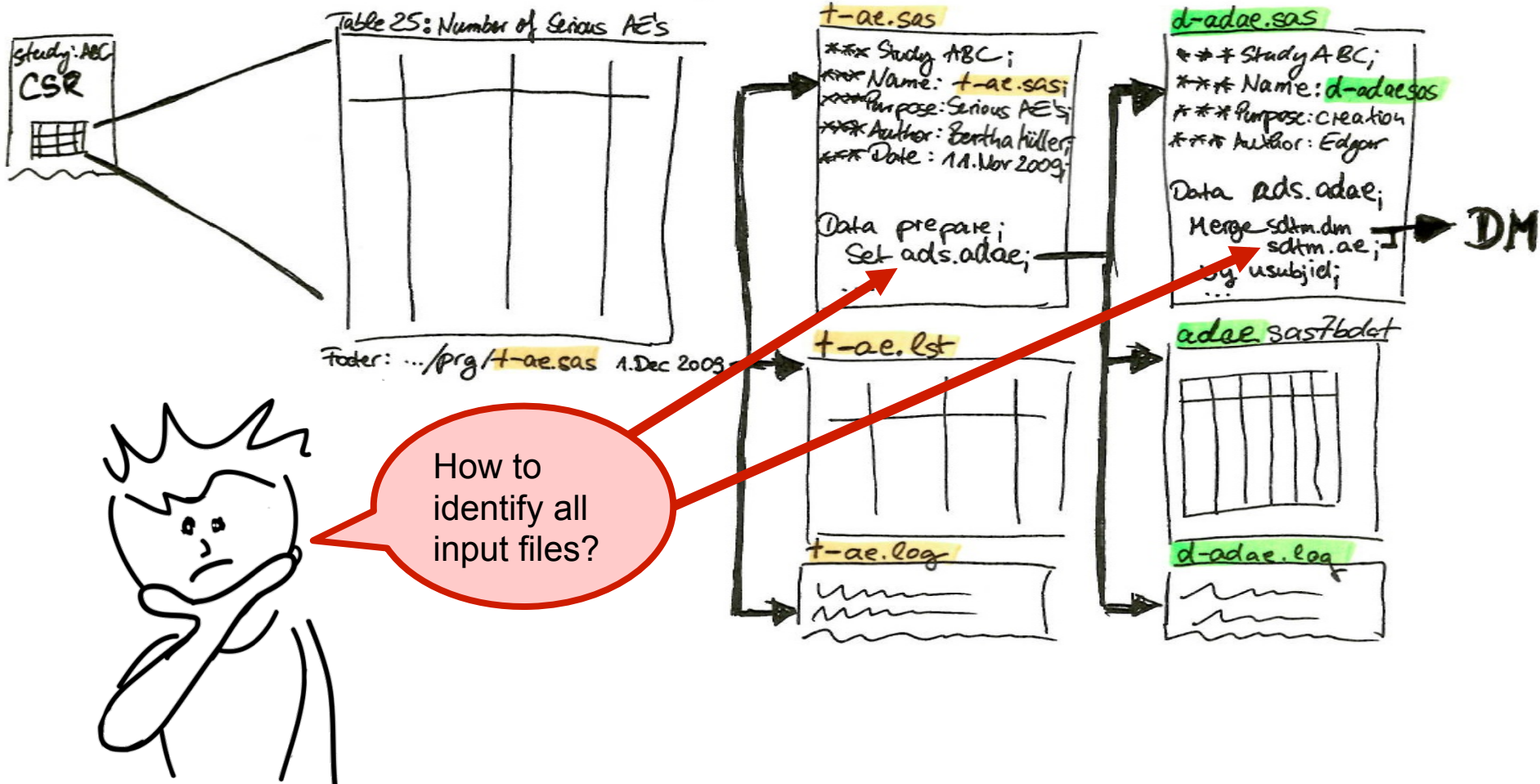


Backward Traceability

...programming rule based



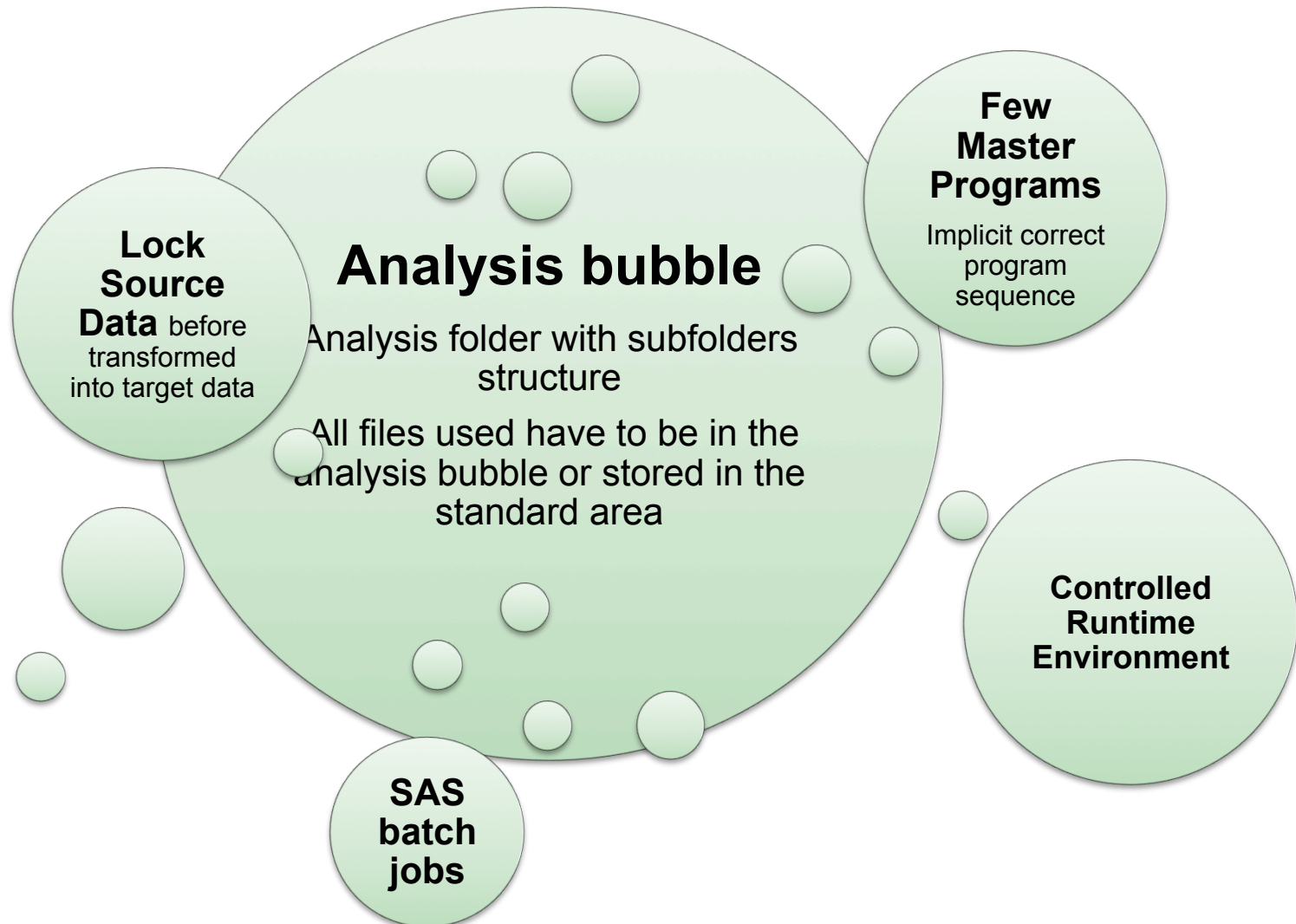
Backward Traceability



ANALYSIS BUBBLE



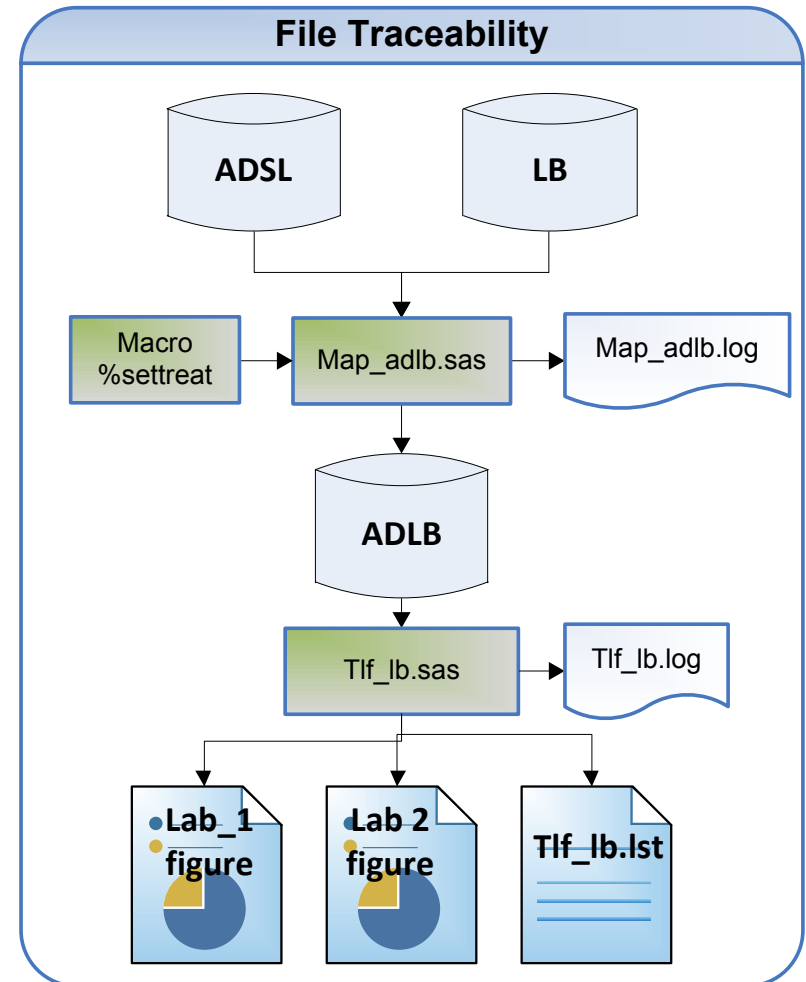
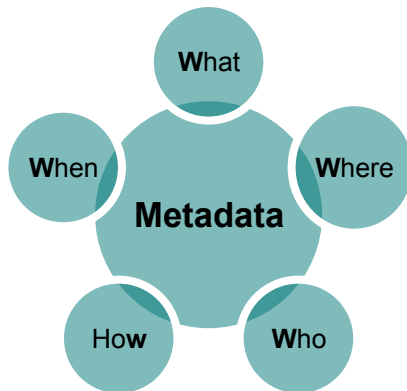
Ensure an uninterrupted chain of source and targets in chronological order



Ways to achieve Analysis File Traceability



The goal is to find a user friendly way to reach complete analysis file traceability



Ways to achieve Analysis File Traceability (2)

Way 1) Programming Rule and File System

- Good naming convention to get between program and outputs
- Output footer as a backward link to the SAS program.
- Program header to document program author and purpose
- From file system take file path and modification time stamp

→ Limitation: Input data sets and other input files
Programming rules have to be followed STRICTLY

Way 2) Manual tracking of input and output files

- Define macro variables in the program header for each input and output file

```
%let indat1 = sdtm.ae;  
%let indat2 = thes.meddra;  
%let outfile1 = lb-plot1.emf;  
%let outdat1 = ads.adlb;
```

→ Limitation: Manual overhead, quality?



RTrace - Traceability



Way 3) SAS Code parser

Tool that scans programs, log and output files for key words to detect links between programs, macros, data sets, output files

DATA ads.adae	→ ads.adae is written (OUT)	
SET oad.lb	→ oad.lb is read	(IN)
%INCLUDE "&prgdir/t-ae.SAS"	→ sub-programs used (IN)	

→ Limitation:

- Effort to get such a tool, ignore comments and program header
- Difficult to get 100% of all INs and OUTs (file names created per macro loop, or dynamic number of figure files (mprint file helps)

Way 4) RTRACE

Embed SAS programs within RTRACELOC option to protocol all input and output files

```
OPTIONS RTRACE = ALL RTRACELOC = "&logdir./mytrace.log;
```

```
*** run all sas programs here ***;
```

```
OPTIONS RTRACELOC = "/dev/null";
```



RTrace



...
 File closed: /by-sasp/patdb/projects/de03757/310101/stat/test_1/results/t-eff-primary.lst
 File closed: /by-sasp/patdb/projects/de03757/310101/stat/test_1/pgms/t-eff-primary.sas
 File referenced: /by-sasp/patdb/projects/de03757/310101/stat/test_1/pgms/t-ex.sas
 File opened: /by-sasp/patdb/projects/de03757/310101/stat/test_1/pgms/t-ex.sas
 File opened: /opt/SAS-El/by-sasp/sas91/sas91/sashelp/vtitle.sas7bview
 File closed: /opt/SAS-El/by-sasp/sas91/sas91/sashelp/vtitle.sas7bview

File opened
 File opened
 File opened
 File opened
 File closed
 File closed
 File closed
 File opened
 File opened
 File closed
 ...

```

  <- /by-sasp/patdb/projects/865028/14371/stat/test/pgms/d-adv.sas
  <- /by-sasp/patdb/projects/865028/14371/stat/test/pgms/d-adsl.sas
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/ad.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/ds.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/ta.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/ex.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/se.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/dm.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adv.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/xg.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/su.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/prod/metadata/dm.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/fmt_hpx.sas7bcat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/pgms/d-adae.sas
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/adae.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/ad.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adae.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/prod/metadata/ae.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/prod/metadata/dm.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/fmt_hpx.sas7bcat
  
```

/sas7/_titles.sas7bdat.lck

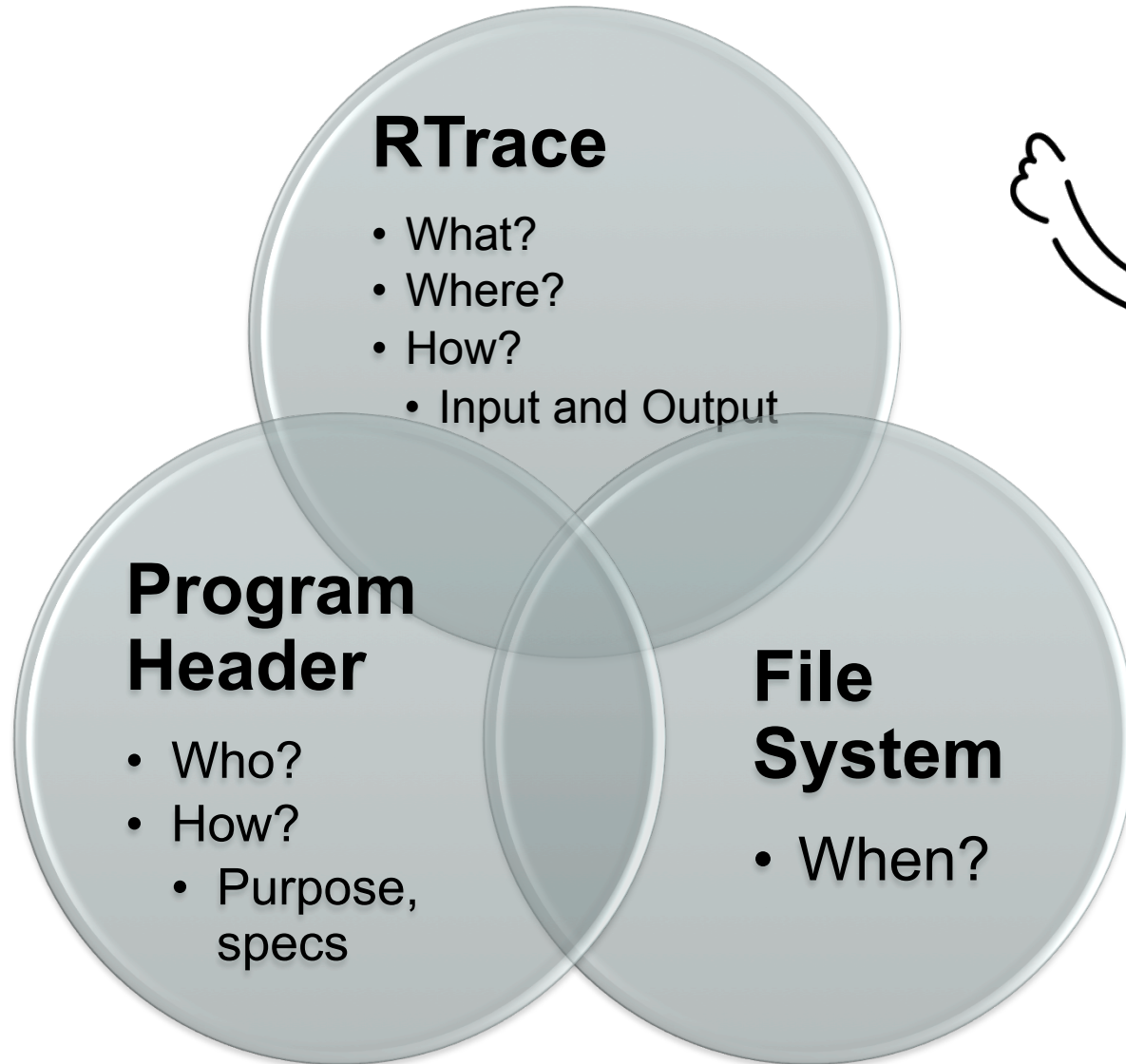
/sas7/#tf7436.sas7butl
 /sas7/#tf7437.sas7bndx.lck
 sas7/#tf7436.sas7butl
 sas7/#tf7437.sas7bndx.lck

↔ read write ← read → write

```
+ <- /by-sasp/patdb/projects/865028/14371/stat/test/pgms/d-adv.sas
- <- /by-sasp/patdb/projects/865028/14371/stat/test/pgms/d-adsl.sas
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/ad.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/ds.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/ta.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/ex.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/se.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/dm.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adv.sas7bdat
  <-> /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/xg.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/pre_prod/data/ad/su.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/prod/metadata/dm.sas7bdat
  <-> /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/fmt_hpux.sas7bc
- <- /by-sasp/patdb/projects/865028/14371/stat/test/pgms/d-adae.sas
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/adae.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/ad.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adsl.sas7bdat
  <-> /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/adae.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/stat/test/data/adsmeta/adsl.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/prod/metadata/ae.sas7bdat
  <- /by-sasp/patdb/projects/865028/14371/dm/prod/metadata/dm.sas7bdat
  <-> /by-sasp/patdb/projects/865028/14371/stat/test/data/ads/fmt_hpux.sas7bc
```

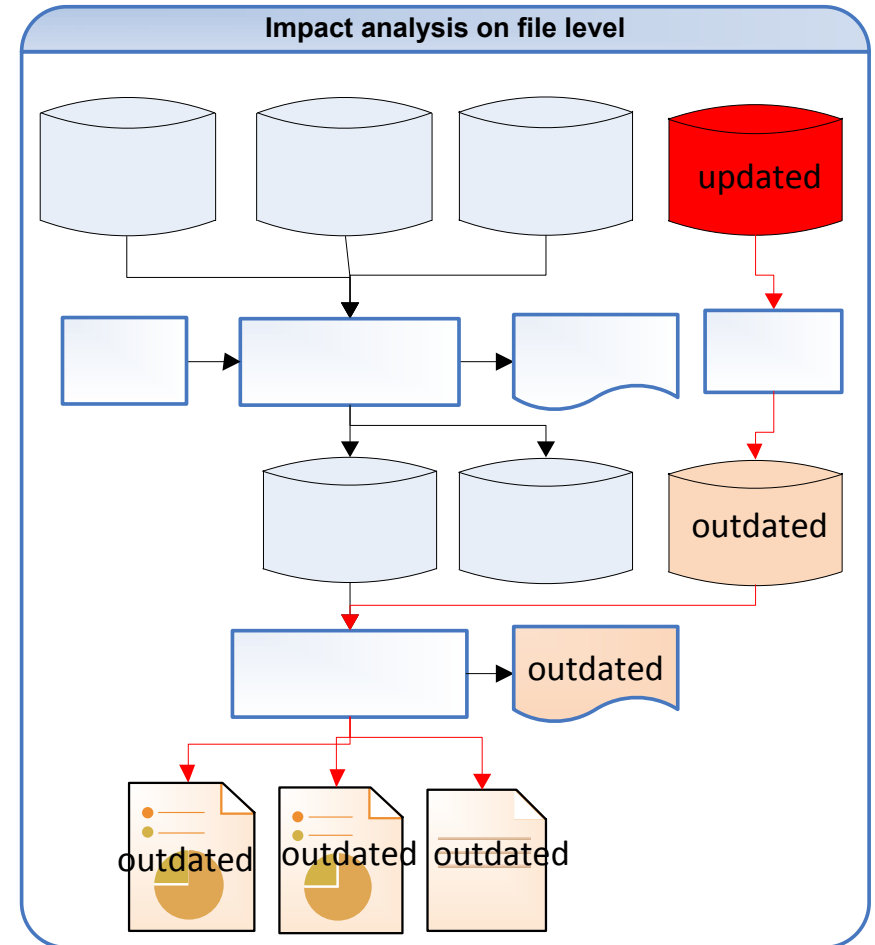
Solution

...for Analysis traceability on file level



Impact Analysis

- Chronological order of all objects in the work flow
- What impact has a data update?
- Typical cases in statistical analysis where impact analysis can be helpful:
 - In case of a data base reopening where data changed
 - In case SAP supplement or specification update that lead to updated SAS programs
- Observe input folders for new input files that aren't used. called loose ends.



Consistency and plausibility checks



- ☐ Automatic checks should be performed over all collected metadata
- ☐ Metadata available redundantly → cross checks to ensure consistency
- ☐ Check for example that:
 - ✓ ...possible input and output files documented in header are in sync or even update header automatically (before productive run)
 - ✓ ...SAS Dataset name are not longer than 8 characters
 - ✓ ...program name = beginning of output file names
 - ✓ ...all used files are stored in the analysis bubble or standards area
 - ✓ ...only validated and release data and programs were used in the productive run
 - ✓ ...no loose beginnings or loose ends exist

A lot less manual checks ☺



Lost without a trace....



Thank
You
For
Your
Attention!

