



Who Checks the AI? Making AI-Generated Analyses Auditable



Tomás Sabat Stöfsei

CEO & Co-Founder, Verisian

Code generation was never the constraint

Today, everyone can generate a lot of code quickly

The constraint is validation: how do I trust results in a high-stakes regulated environment?

AI-generated code can be deceptive

As AI models improve, outputs look more and more reasonable

But code can break logic silently: missing values or patients dropping out

Double programming may not catch this: two incorrect interpretations that agree do not make a right one

Code is the source of truth

Specs state what the code should do. Submission documents what the code supposedly does. The code is what actually happens.

Every assumption, imputation, grouping and derivation lives in the code

A study can be conformant to standards and still contain incorrect data.

Validity of content is validity of code.

Validation means code validation

You can only trust results if you trust the code that produced them

Data cannot be validated. Its content is a consequence of the code

Code validation is the single most important effort in analysis, submission and review

It starts with specifications

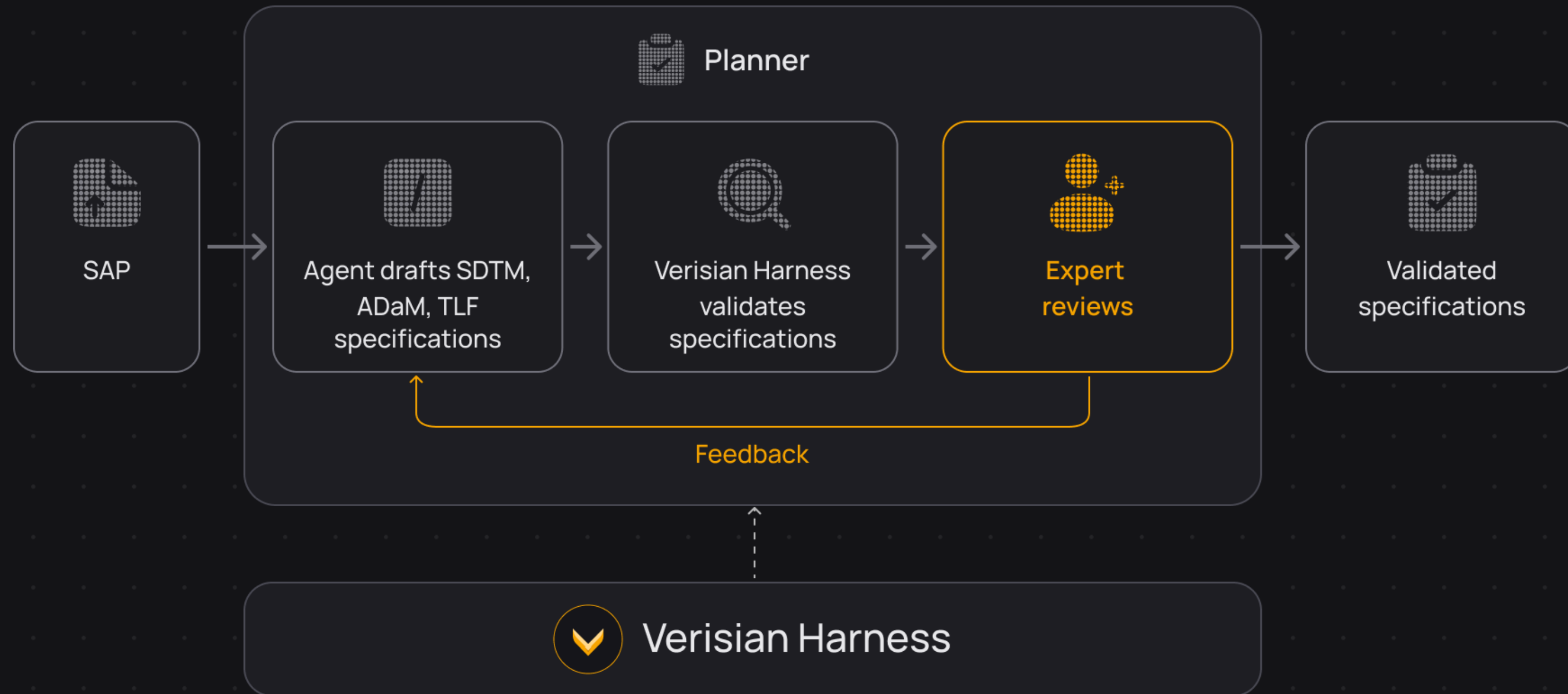
Every downstream process depends on them: implementation, validation, pooling, submission

Most are incomplete, ambiguous, or copied from a similar study

An output cannot be validated against an ambiguous specification, by agents or by experts

Specifications drift from the code the moment implementation begins

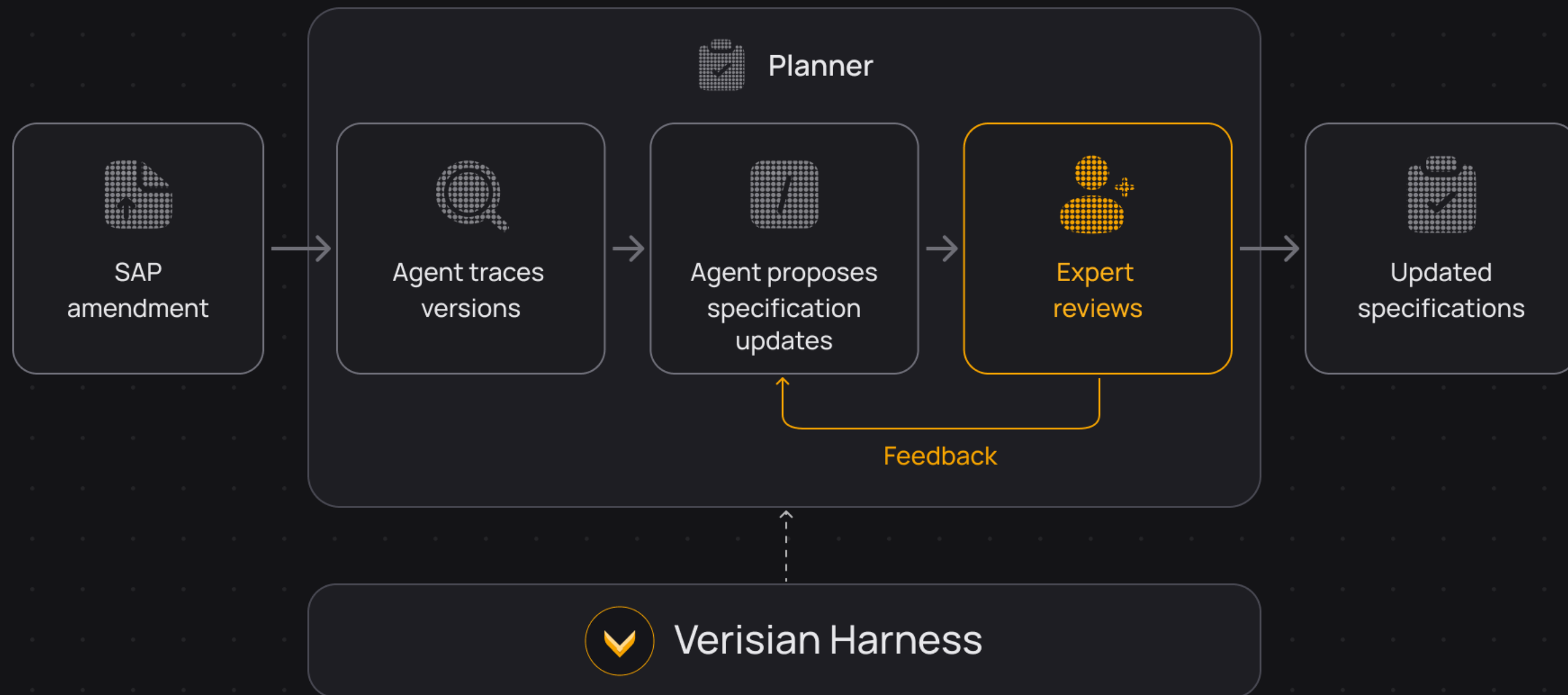
Planning: agents draft specifications, experts validate them



Specification maintenance

Specifications drift from the code the moment implementation begins

Planning: specifications kept aligned with the SAP



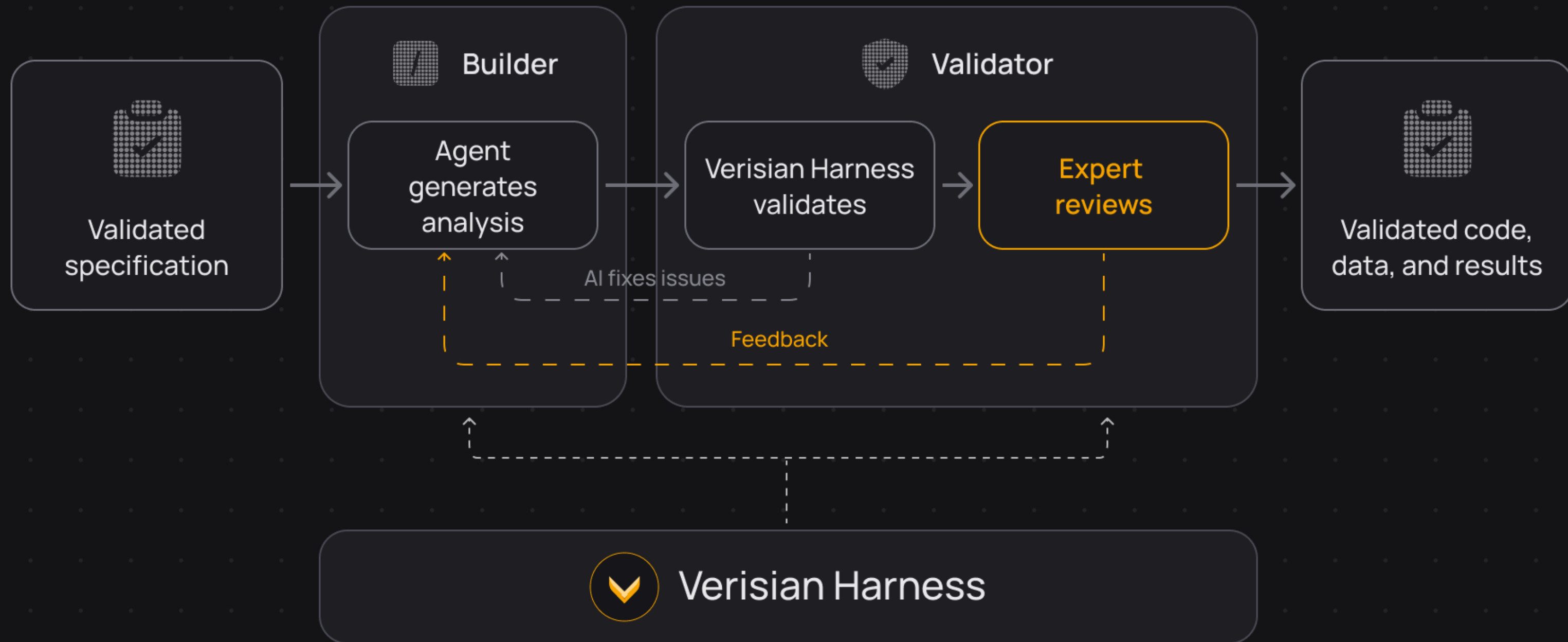
Analysis: why full automation is hard

Code generation alone shifts the bottleneck from generation to validation

Validation resists automation: tens of thousands of lines of code cannot be understood at scale

AI hallucinates when inputs get too large

Analysis: implementation and validation in one continuous cycle



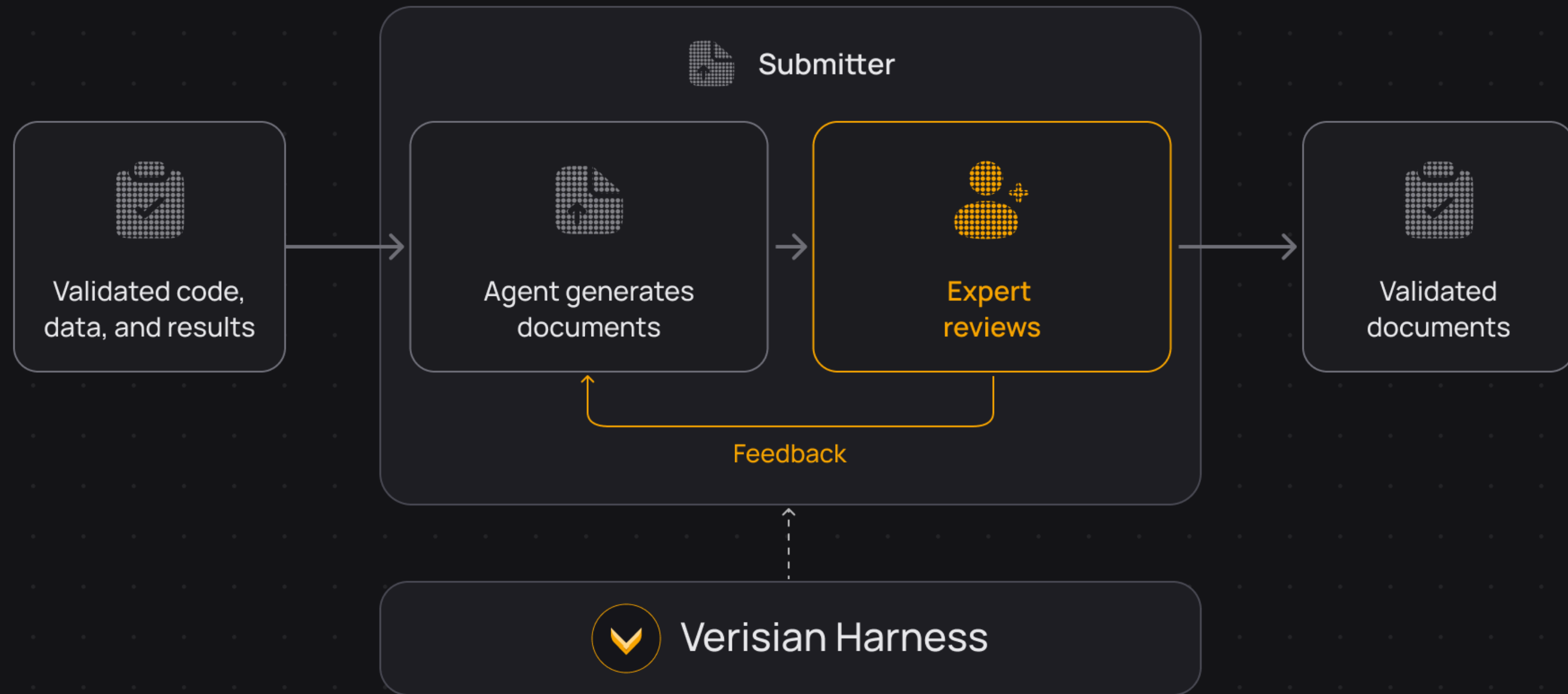
Generating submission documents with AI

Define-XML computation methods and key Reviewer's Guides sections are study-specific and cannot be templated

Generation is fast. Validation is the problem

The content lives in experts' heads and in specifications that have drifted from the code

Submission documents: why automating them is hard



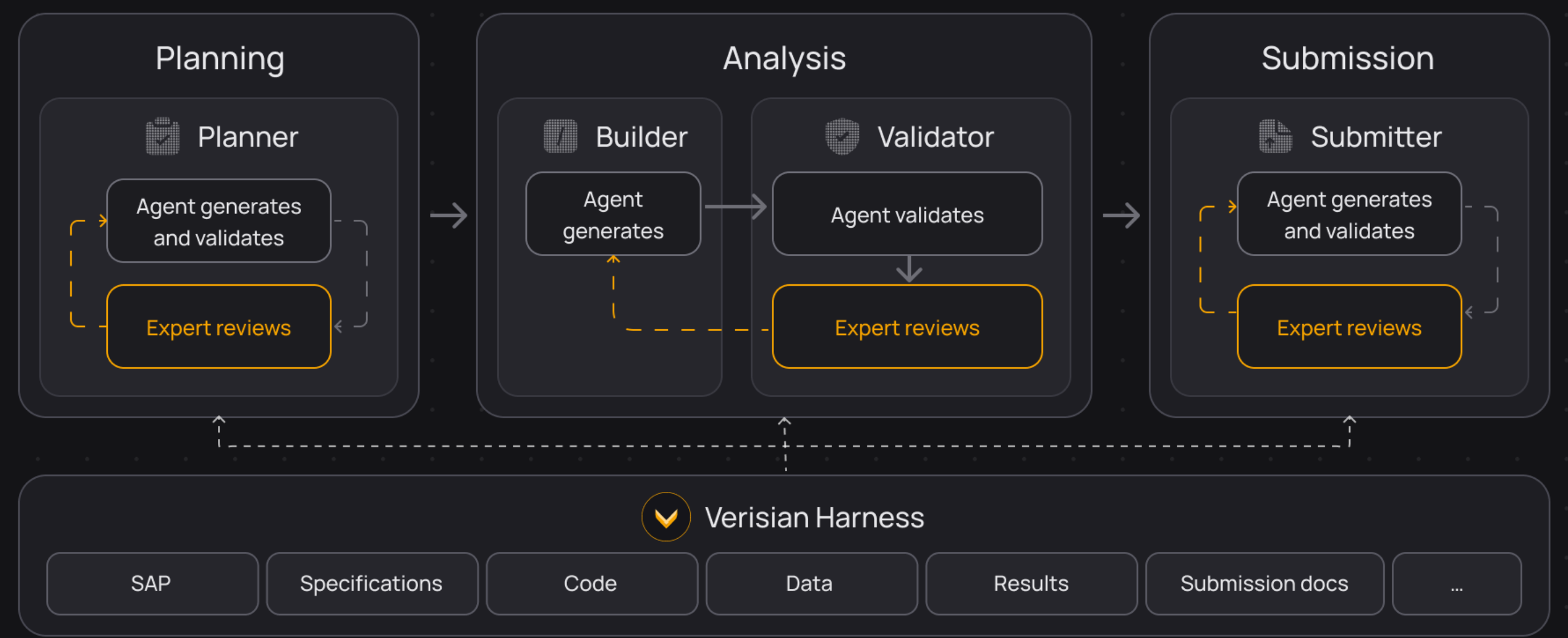
Customer story: Bayer uses Verisian to automate submission document generation



The Verisian AI infrastructure is enabling Bayer to take big steps towards delivering our data caterer vision, where at least 80% of the busywork is automated by trustworthy and evidence-based AI.

Sascha Ahrweiler VP, Head Statistical Programming & Analysis, Strategy & Operations, Bayer

Agentic automation with expert review



Verisian unifies every document, specification, code, dataset, and result into a fully deterministic, reproducible, and traceable system.



What is **age**?

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...



```
age.sas
...
firstdose=min(firstdose,startdt,enddt);
age=intck('year',dob,firstdose,'c');
```



age
36
...
...
...
...

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...



```
age.sas
...
firstdose=min(firstdose,startdt,enddt);
age=intck('year',dob,firstdose,'c');
```

Variable Traceability: SDTM.DM.AGE in a demo study

AGE is a simple concept, but can be derived in different ways

dob	firstdose
1974-02-05	2010-03-28
...	...
...	...
...	...
...	...



```
age.sas
...
firstdose=min(firstdose,startdt,enddt);
age=intck('year',dob,firstdose,'c');
```



age
36
...
...
...
...

How is AGE derived and what data does it depend on?

 macros	 source	 sdtm	 adam	 tlf
 %derive_baseline.sas	 demographic.sas	 dm.sas	 adae.sas	 t14_1_1.sas
 %dtc2dt.sas	 dosing.sas	 ae.sas	 adef.sas	 t14_2_1.sas
 %derive_age.sas	 labs.sas	 cm.sas	 adsl.sas	 t14_3_1.sas
 %flag_last.sas	 adverse.sas	 ex.sas	 adef.sas	 l162_1.sas
 %km_plot.sas	 exposure.sas	 lb.sas	 adtte.sas	 l16_2_2.sas
 %visit_window.sas	 vital_signs.sas	 vs.sas	 advs.sas	 f14_2_1.sas
 +14 more	 +18 more	 +30 more	 +16 more	 +174 more

dm.sas

```
proc sort data=source.dosing(keep=subject startdt enddt) out=dosing;
by subject startdt;
run;

data dose_sum;
set dose_raw;
by subject;

retain firstdose lastdose;

if first.subject then do;
firstdose=.;
lastdose=.;
end;

firstdose=min(firstdose,startdt,enddt);
lastdose=max(lastdose,startdt,enddt);

if last.subject;
run;

proc sort data=source.demographic out=demographic;
by subject;
run;

data demog_dose;
merge demographic dosing;
by subject;
run;

options missing=' ';

data sdtm.dm;
set EMPTY_DM demog_dose(rename=(race=_race));

studyid='XYZ123';
domain='DM';
usubjid=left(uniqueid);
subjid=put(subject,3.);
rfstdtc=put(firstdose,yymmdd10.);
rfendtc=put(lastdose,yymmdd10.);
rfxstdtc=put(firstdose,yymmdd10.);
rfxendtc=put(lastdose,yymmdd10.);
rficdte=put(icdate,yymmdd10.);
rfpendtc=put(lastdoc,yymmdd10.);
dthfl='N';
siteid=substr(subjid,1,1) || "00";
brthdte=put(dob,yymmdd10.);
age=intck('year', dob, firstdose, 'c');
if age ne . then ageu='YEARS';
sex=put(gender,sex_demographic_gender.);
race=put(_race,race_demographic_race.);
armcd=put(trt,armcd_demographic_trt.);
arm=put(trt,arm_demographic_trt.);
actarmcd=put(trt,armcd_demographic_trt.);
actarm=put(trt,arm_demographic_trt.);
country="USA";
run;
```

dm.sas

```
proc sort data=source.dosing(keep=subject startdt enddt) out=dosing;
by subject startdt;
run;

data dose_sum;
set dose_raw;
by subject;

retain firstdose lastdose;

if first.subject then do;
firstdose=.;
lastdose=.;
end;

firstdose=min(firstdose,startdt,enddt);
lastdose=max(lastdose,startdt,enddt);

if last.subject;
run;

proc sort data=source.demographic out=demographic;
by subject;
run;

data demog_dose;
merge demographic dosing;
by subject;
run;

options missing=' ';

data sdm.dm;
set EMPTY_DM demog_dose(rename={race=_race});

studyid='XYZ123';
domain='DM';
usubjid=left(uniqueid);
subjid=put(subject,3.);
rfstdtc=put(firstdose,ymmddl0.);
rfendtc=put(lastdose,ymmddl0.);
rfxstdtc=put(firstdose,ymmddl0.);
rfxendtc=put(lastdose,ymmddl0.);
rficdte=put(icdate,ymmddl0.);
rfpendtc=put(lastdoc,ymmddl0.);
dthfl='N';
siteid=substr(subjid,1,1) || "00";

age=intck('year', dob, firstdose, 'c');

sex=put(gender,sex_demographic_gender.);
race=put(_race,race_demographic_race.);
armcd=put(trt,armcd_demographic_trt.);
arm=put(trt,arm_demographic_trt.);
actarmcd=put(trt,armcd_demographic_trt.);
actarm=put(trt,arm_demographic_trt.);
country="USA";
run;
```


How is AGE derived and what data does it depend on?

macros	source	sdtm	adam	tlf
 %derive_baseline.sas	 demographic.sas	 dm.sas	 adae.sas	 t14_1_1.sas
 %dtc2dt.sas	 dosing.sas	 ae.sas	 adef.sas	 t14_2_1.sas
 %derive_age.sas	 labs.sas	 cm.sas	 adsl.sas	 t14_3_1.sas
 %flag_last.sas	 adverse.sas	 ex.sas	 adef.sas	 l162_1.sas
 %km_plot.sas	 exposure.sas	 lb.sas	 adtte.sas	 l16_2_2.sas
 %visit_window.sas	 vital_signs.sas	 vs.sas	 advs.sas	 f14_2_1.sas
 +14 more	 +18 more	 +30 more	 +16 more	 +174 more

```
demographics.sas

proc format;
value trt 1="Active" 0="Placebo";
value gender 1="Male" 2="Female";
value race 1="White" 2="Black" 3="Other";
run;

data source.demographic;
set raw.demographic;
label subject="Subject Number" trt="Treatment" gender="Gender"
race="Race"
orace="Other Race Specify" dob="Date of Birth" uniqueid=
"Company Wide Subject ID" randdt="Randomization Date" icdate=
"Informed Consent Date" lastdoc="Last date of contact";
input subject 1-3 trt 5 gender 7 race 9 orace $ 11-20 +1 dob
mmddyy10. +1
randdt mmddyy10. +1 icdate mmddyy10. +1 lastdoc mmddyy10.;
uniqueid='UNI' || put(subject,3.);
format dob randdt icdate lastdoc mmddyy10.;
run;

data _null_;
rc = filename('xptdir', "&xptdir");
if fexist('xptdir') = 0 then rc = dcreate("xpt",
"/mnt/viya-share/homes/demo-study/data/xpt");
run;

libname xpt_demo xport "&xptdir./demographic.xpt";
data xpt_demo.demograp;
set source.demographic;
run;
libname xpt_demo clear;
```

```
dosing.sas

* dosing.sas;
data source.dosing;
label subject="Subject Number" startdt="Dosing start date" enddt=
"Dosing end date" ddose="Daily dose taken (pills)" uniqueid=
"Company Wide Subject ID" startmm="Month of Start Dose" startdd=
"Day of Start Dose" startyy="Year of Start Dose" endmm=
"Month of End Dose" enddd="Day of End Dose" endyy=
"Year of End Dose";
input subject 1-3 startmm 5-6 startdd 8-9 startyy 11-14 endmm 16-17
enddd
19-20 endyy 22-25 ddose 27;
uniqueid='UNI' || put(subject,3.);
startdt=mdy(startmm, startdd, startyy);
enddt=mdy(endmm, enddd, endyy);
format startdt enddt mmddyy10.;
datalines;
;
run;
```

```
dm.sas

proc sort data=source.dosing(keep=subject startdt enddt) out=dosing;
by subject startdt;
run;

data dose_sum;
set dose_raw;
by subject;

retain firstdose lastdose;

if first.subject then do;
firstdose=.;
lastdose=.;
end;

firstdose=min(firstdose,startdt,enddt);
lastdose=max(lastdose,startdt,enddt);

if last.subject;
run;

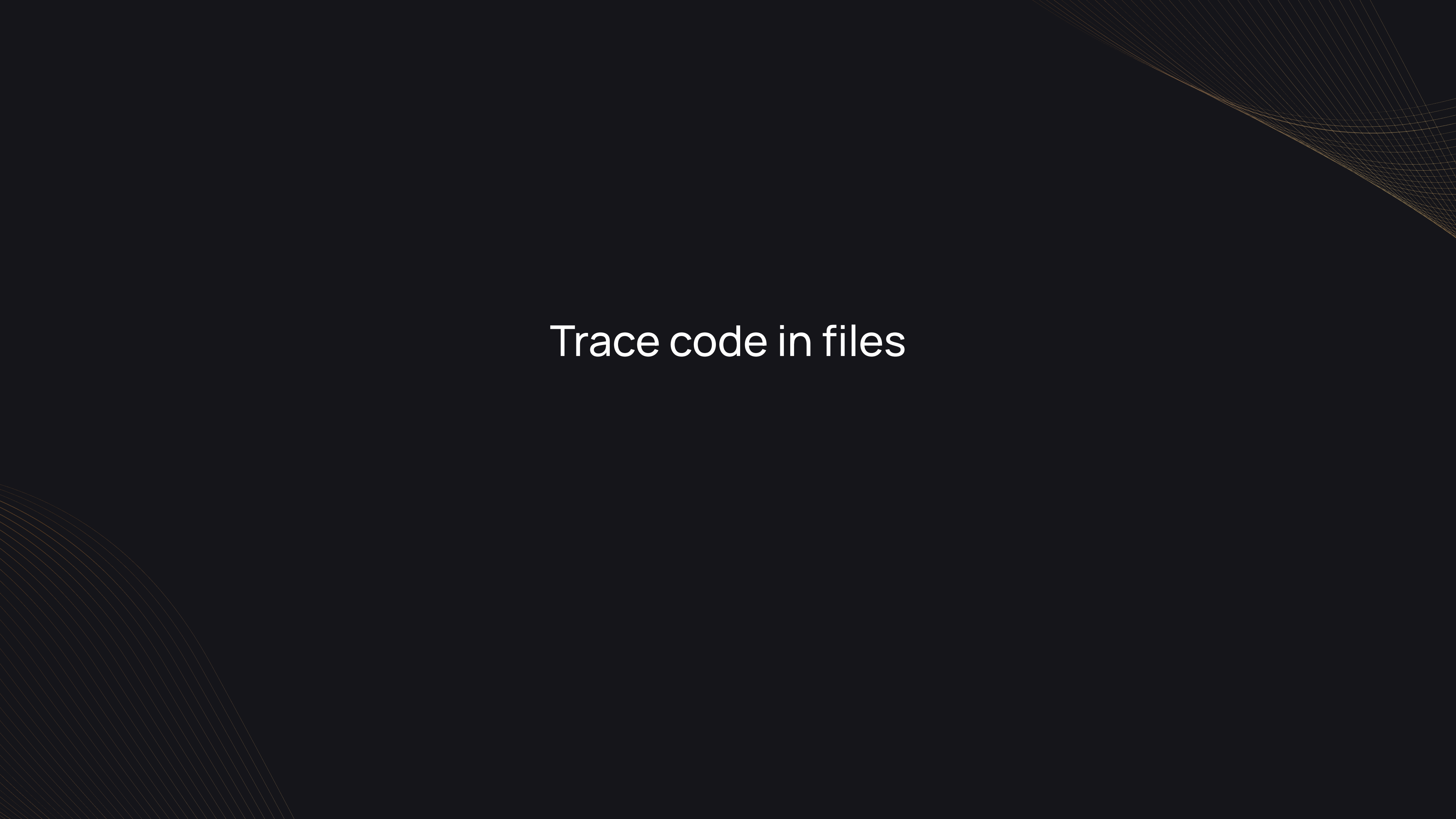
proc sort data=source.demographic out=demographic;
by subject;
run;

data demog_dose;
merge demographic dosing;
by subject;
run;

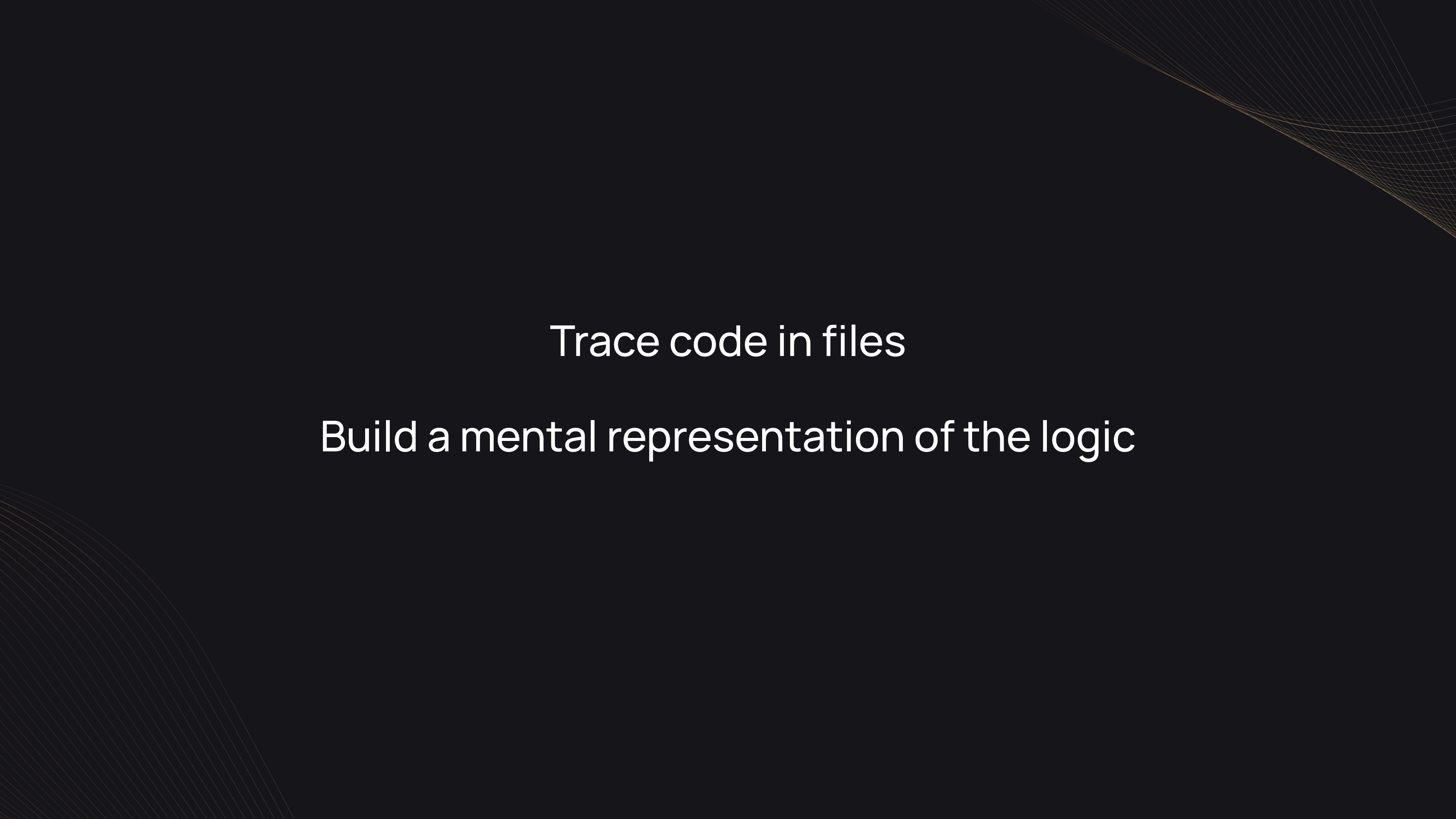
options missing=' ';

data sdtm.dm;
set EMPTY_DM demog_dose(rename=(race=_race));

studyid='XYZ123';
domain='DM';
usubjid=left(uniqueid);
subjid=put(subject,3.);
rfstdtc=put(firstdose,ymmdd10.);
rfendtc=put(lastdose,ymmdd10.);
rfxstdtc=put(firstdose,ymmdd10.);
rfxendtc=put(lastdose,ymmdd10.);
rficdtc=put(icdate,ymmdd10.);
rfpendtc=put(lastdoc,ymmdd10.);
dthfl='N';
siteid=substr(subjid,1,1) || "00";
brthdte=put(dob,ymmdd10.);
age=intck('year', dob, firstdose, 'c');
if age ne . then ageu='YEARS';
sex=put(gender,sex_demographic_gender.);
race=put(_race,race_demographic_race.);
armcd=put(trt,armcd_demographic_trt.);
arm=put(trt,arm_demographic_trt.);
actarmcd=put(trt,armcd_demographic_trt.);
actarm=put(trt,arm_demographic_trt.);
country="USA";
run;
```

Trace code in files



Trace code in files

Build a mental representation of the logic

What variables does **AGE** depend on?

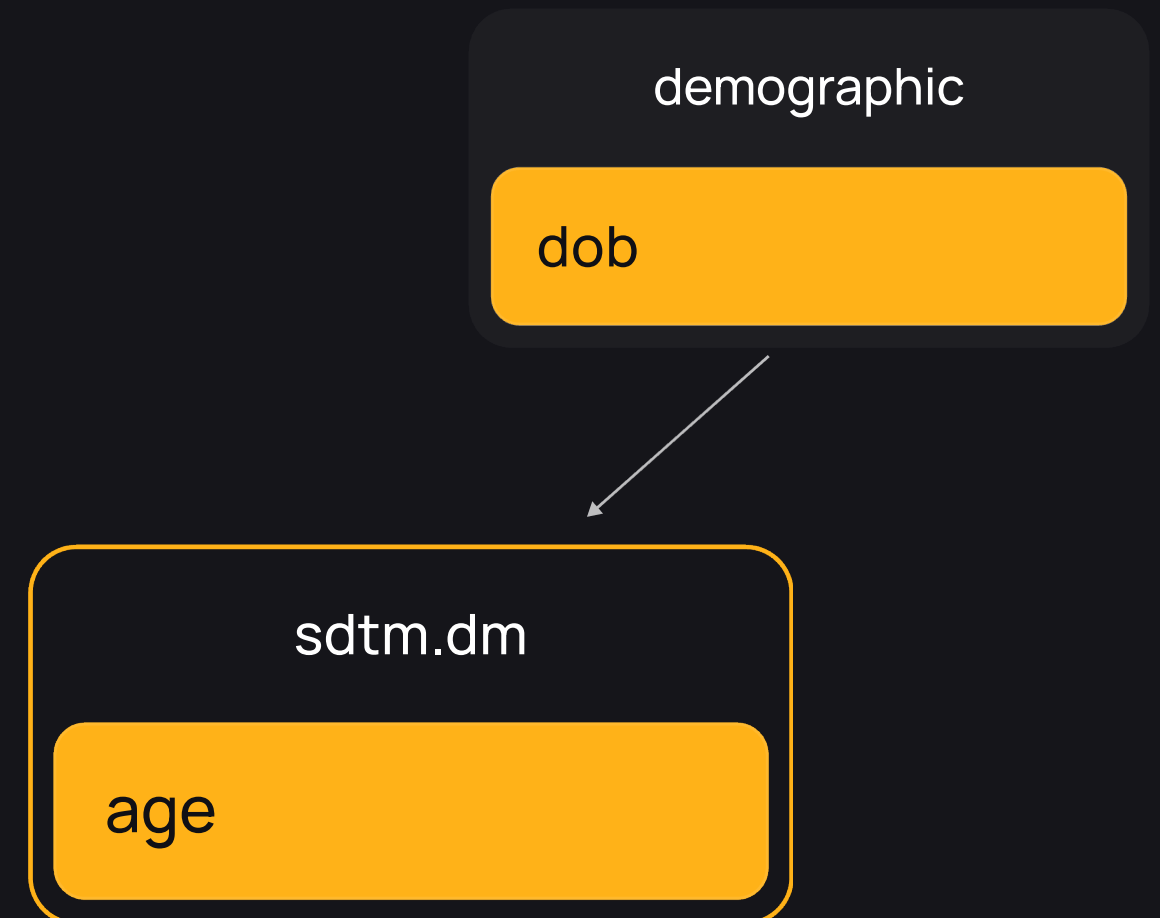
```
data sdtm.dm;  
age=intck('year',dob,firstdose,'c');
```

sdtm.dm

age

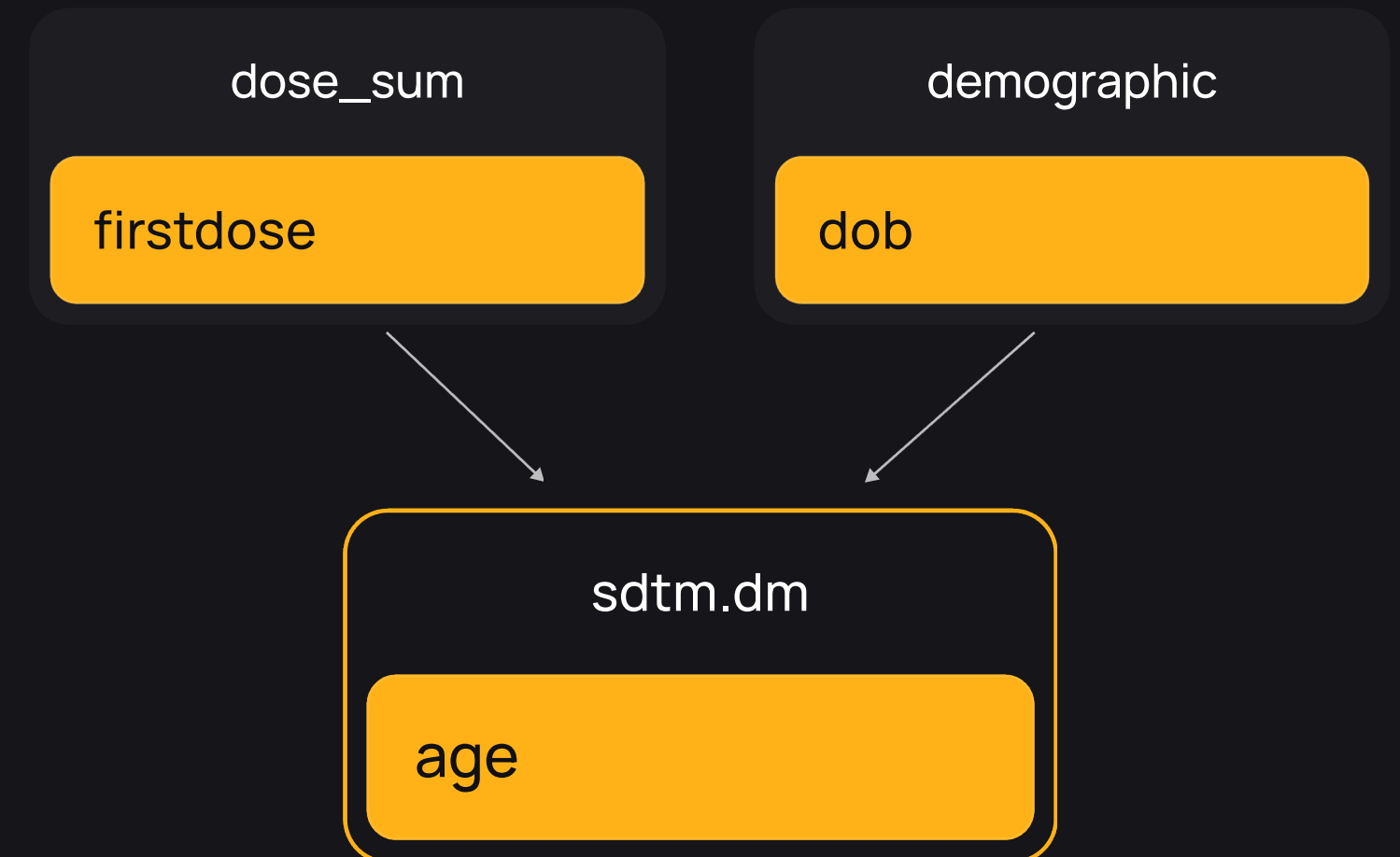
What variables does **AGE** depend on?

```
data demographic;  
set raw_demographics;  
  
data sdtm.dm;  
set demographic dose_sum;  
age=intck('year', dob, firstdose, 'c');
```



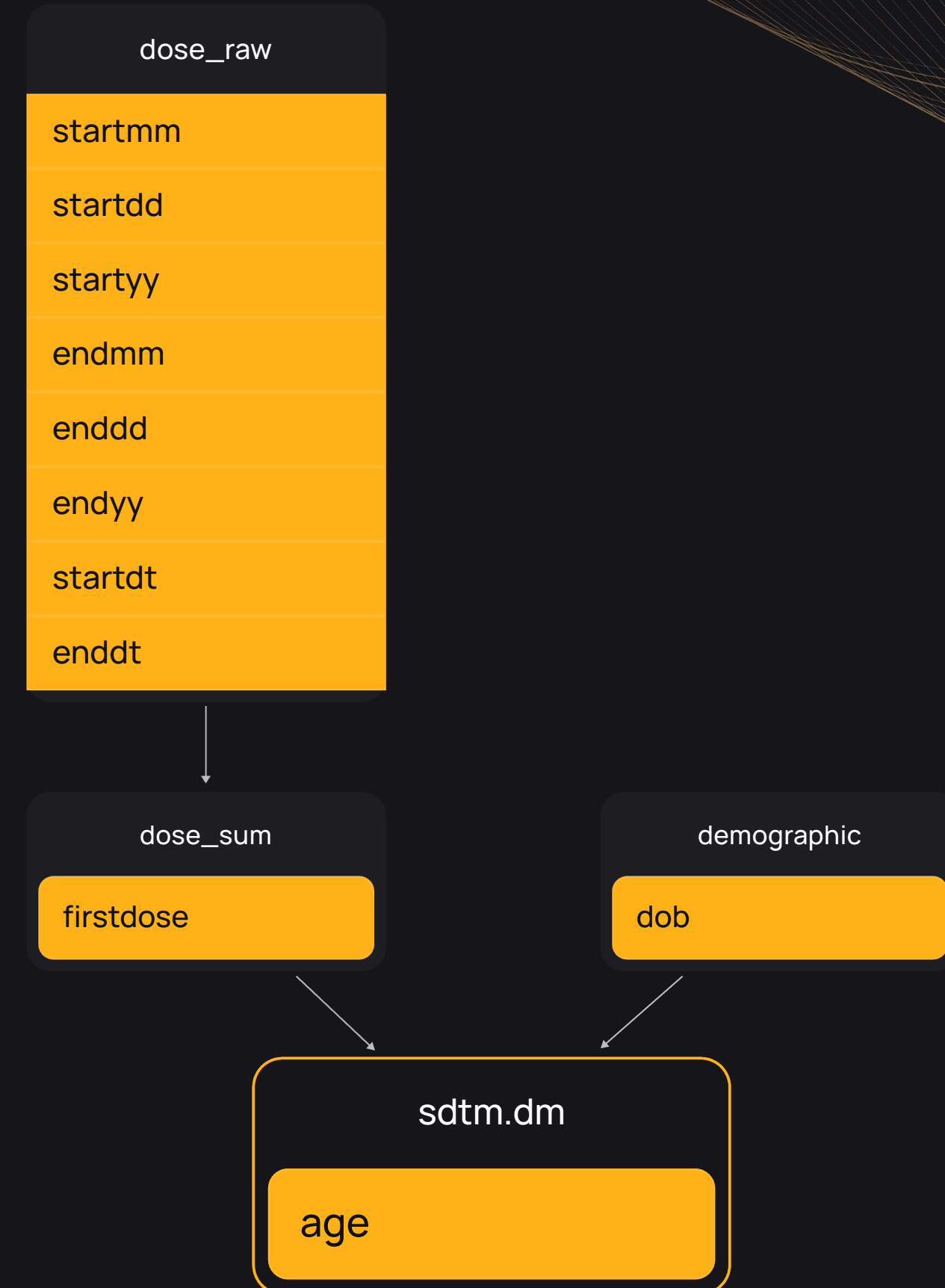
What variables does **AGE** depend on?

```
data dose_sum;  
if first.subject then do;  
firstdose=.;  
firstdose=min(firstdose, startdt, enddt);  
  
data demographic;  
set raw_demographics;  
  
data sdtm.dm;  
set demographic dose_sum;  
age=intck('year', dob, firstdose, 'c');
```



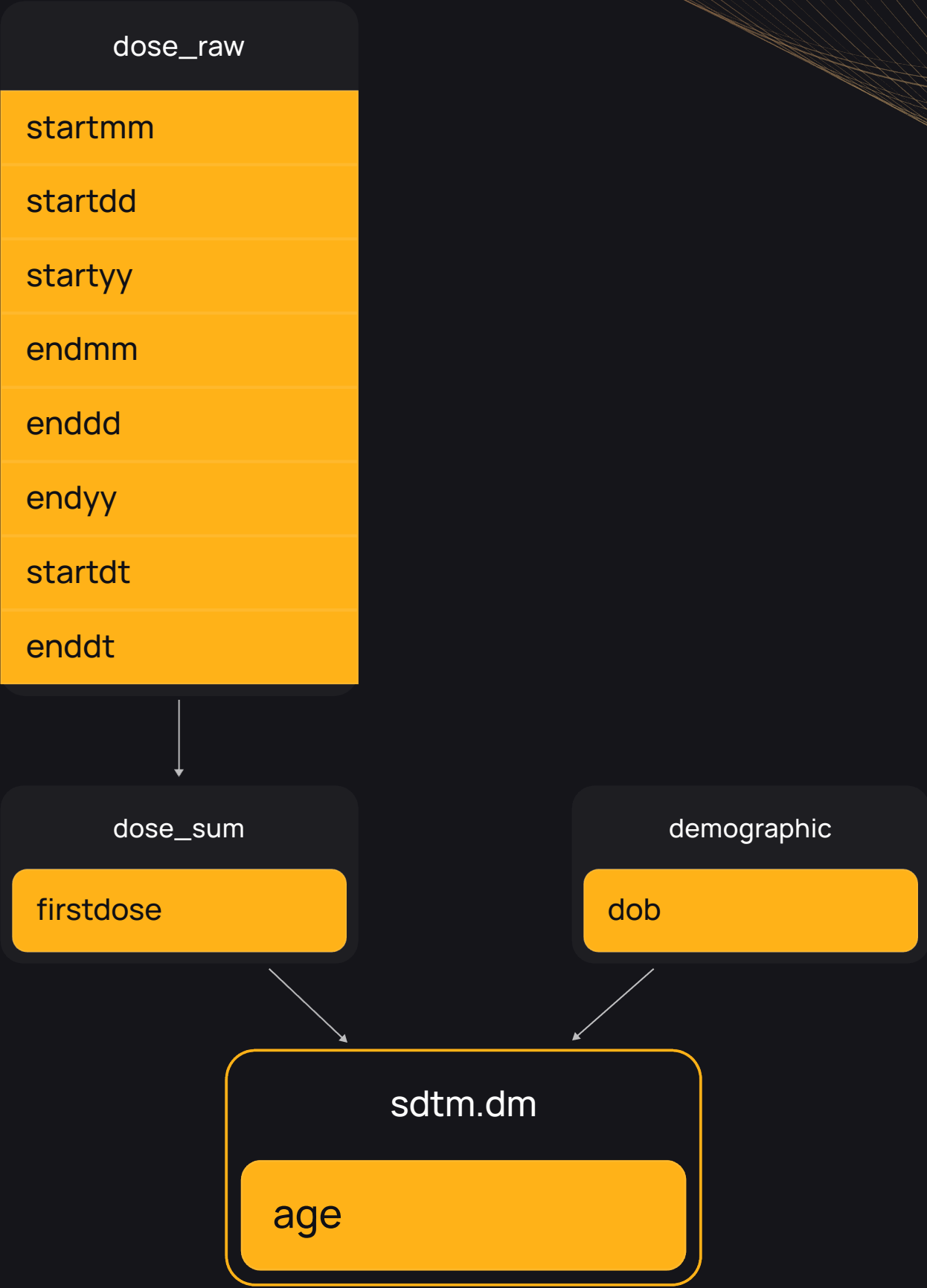
What variables does **AGE** depend on?

```
data dose_raw;  
startdt=mdy(startmm, startdd , startyy);  
enddt=mdy(endmm, enddd, endyy);  
run;  
  
data dose_sum;  
if first.subject then do;  
firstdose=.;  
firstdose=min(firstdose,startdt,enddt);  
run;  
  
data demographic;  
set raw_demographics;  
run;  
  
data sdtm.dm;  
set demographic dose_sum;  
age=intck('year', dob, firstdose, 'c');
```



What variables does AGE depend on?

Dataset	Variable
dose_raw	startmm
dose_raw	startdd
dose_raw	startyy
dose_raw	endmm
dose_raw	enddd
dose_raw	endyy
dose_raw	startdt
dose_raw	enddt
dose_raw	firstdose
demographic	dob



Verisian automates on this foundation

Verisian generates your analyses, validates them continuously, and produces your key submission documents.

01

Grounded in evidence

The infrastructure our AI runs on: one unified, traceable system built from your study's documents, data, and code.



02

Automated with AI

Analyses, datasets, validation, and submission documents are generated by Verisian AI.



03

Approved by your experts

Nothing moves forward without expert review and sign-off at defined gates.

The result is trustworthy, traceable, evidence-based automation across analysis, submission, and regulatory review.

Key takeaways

1. Generation is not the constraint. Validation is.
2. Make specifications AI-ready before automating anything.
3. Validate code: a result is only as valid as the code that produced it.
4. Put experts at defined quality gates with the evidence in front of them.
5. Make every result traceable to its specification, code and data, for your team and for the regulator.

Agentic automation with expert review

