

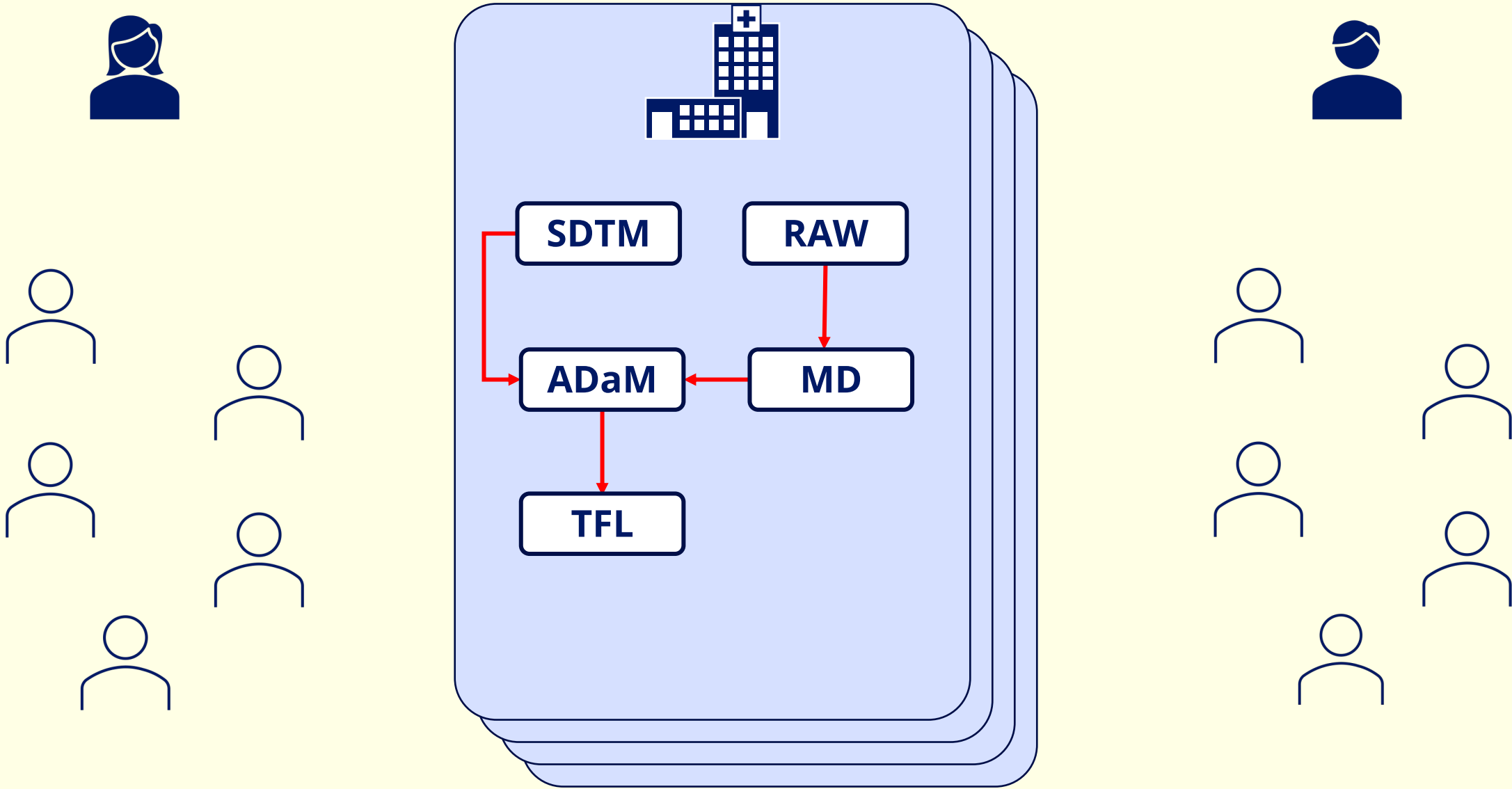
Slicing Codes:

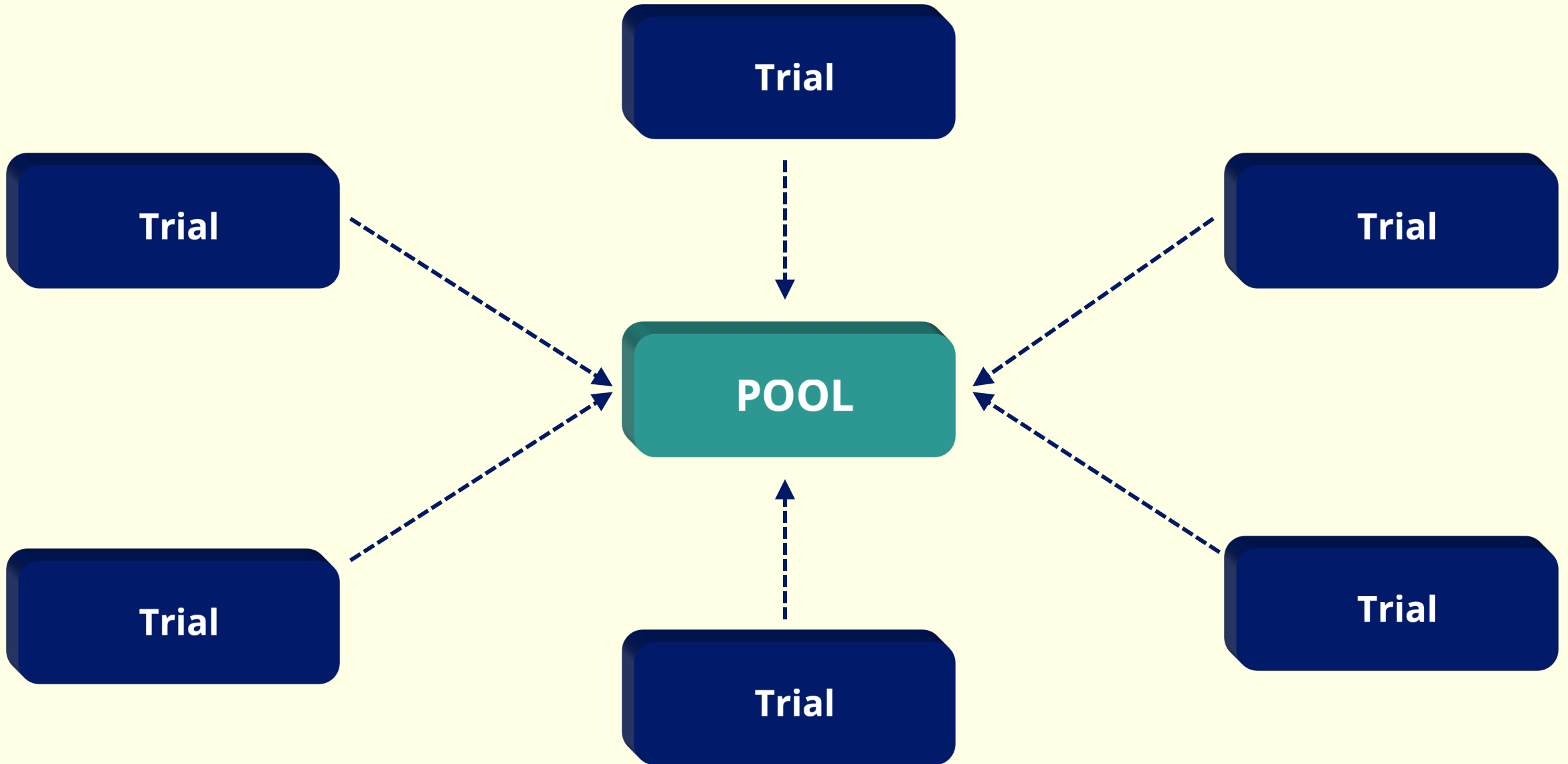
A Modular Approach to ADaM Programming

Mahendran Venkatachalam
Statistical Programming Specialist

DISCLAIMER:

The views and opinions expressed in this presentation belong solely to the presenter and do not necessarily reflect those of the presenter's employer, organization, committee, or any other group or individual. The purpose of this presentation is to highlight the experience gained in one of our projects. No data or confidential information belonging to Novo Nordisk has been utilized in this presentation.



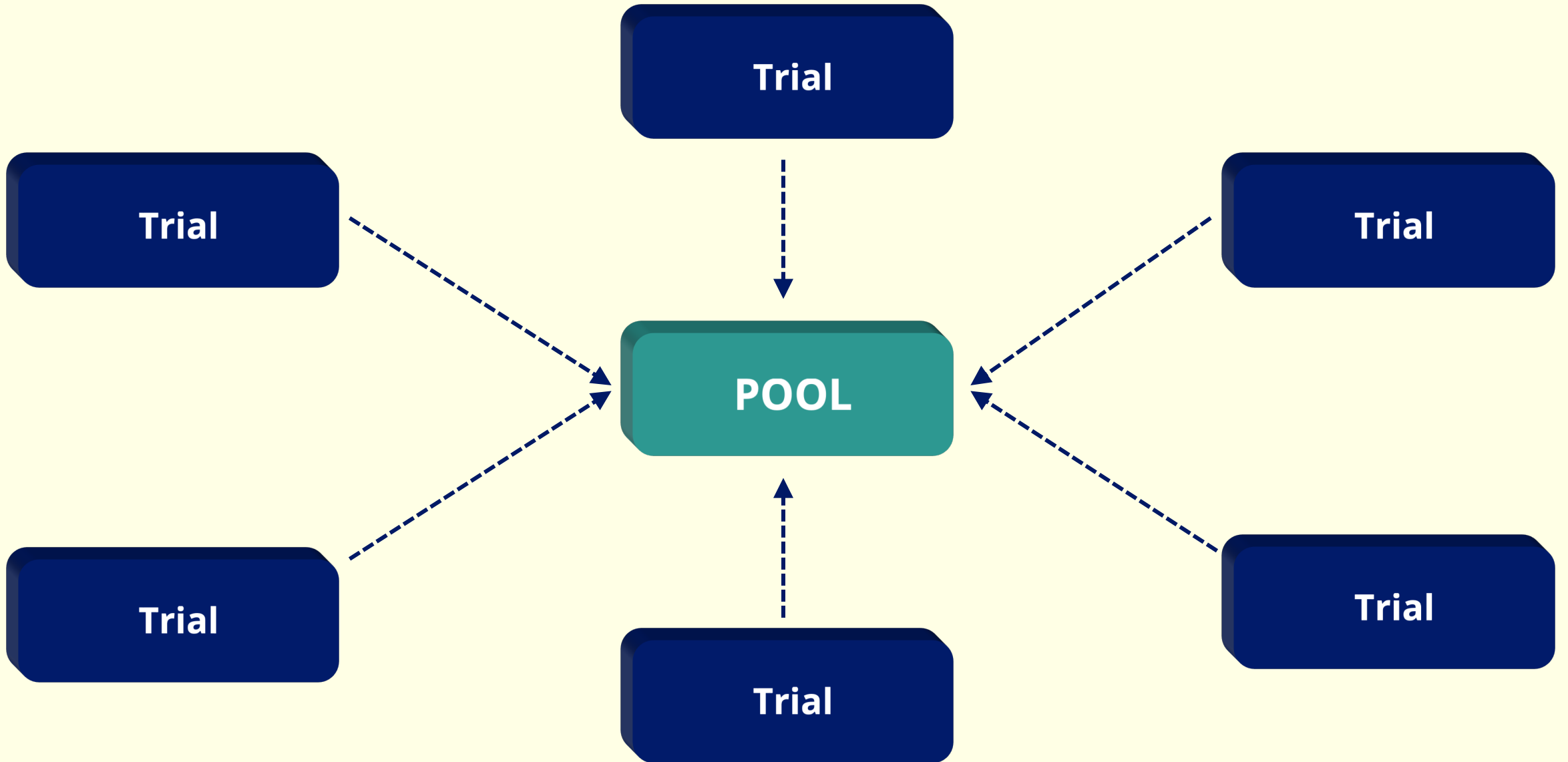


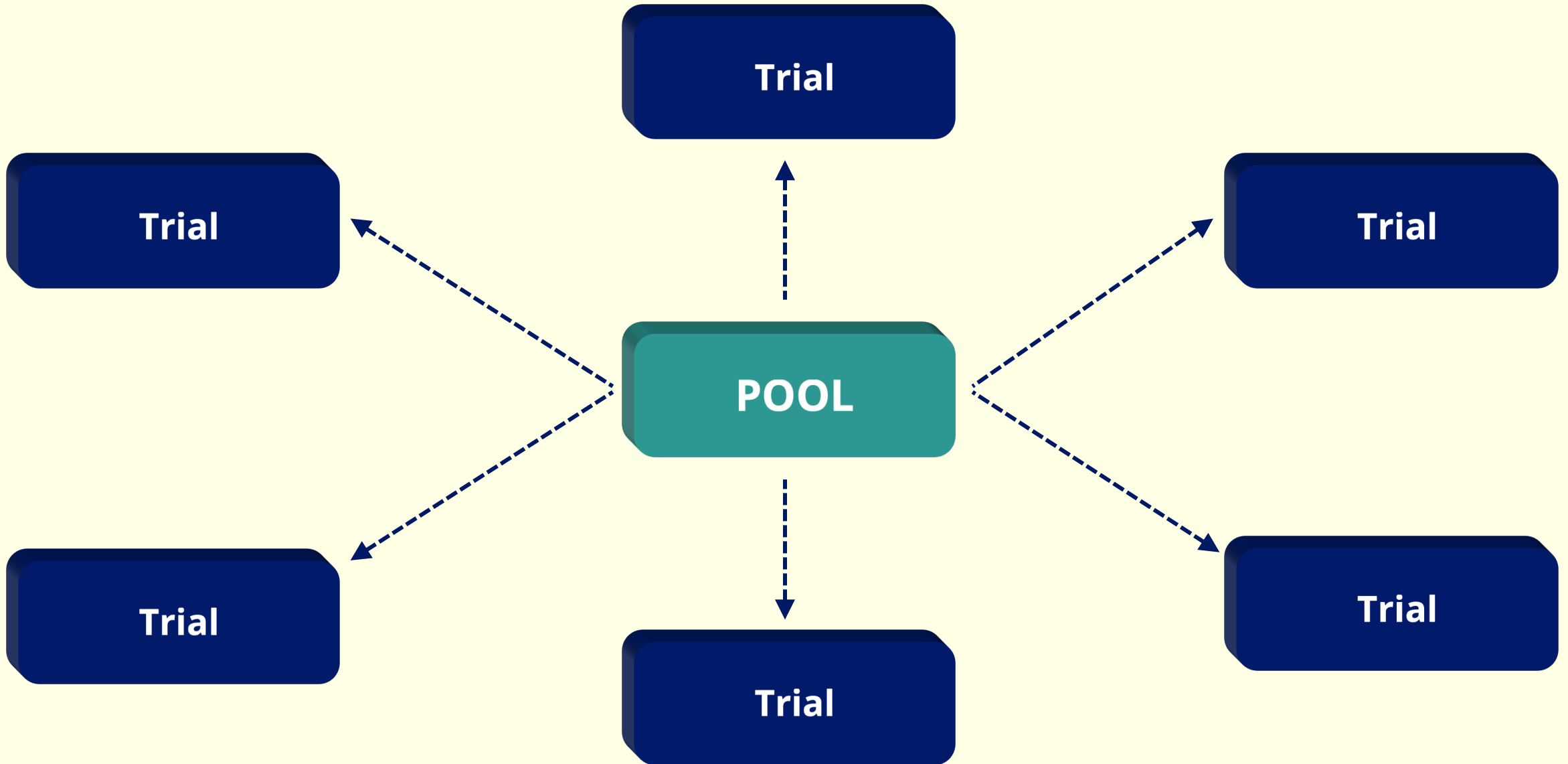
Alignment

Code Duplicates

Resources

Cost & time





“Pooled” Programming

WHY

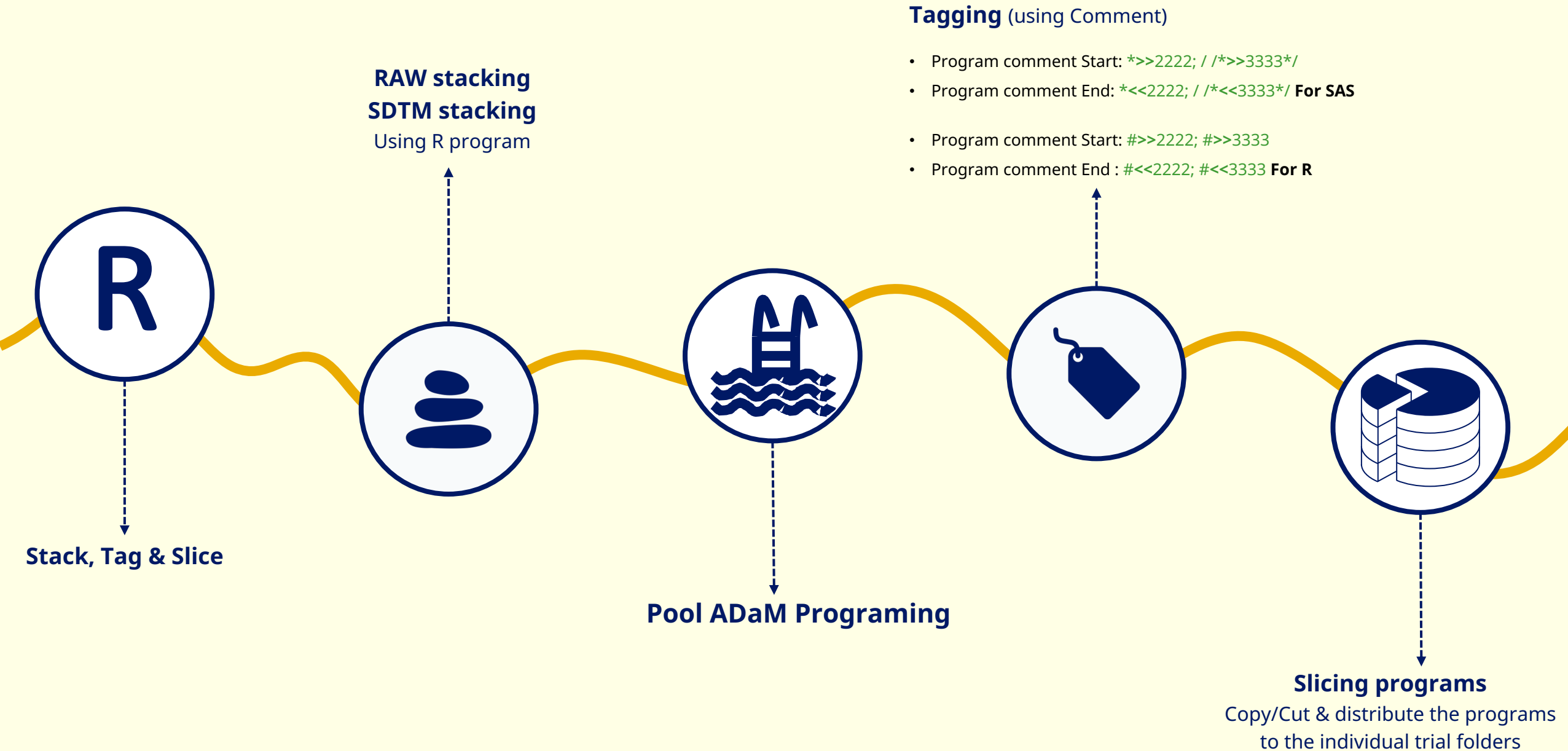
- Highly repetitive tasks
- Submission required pooled data
- Number of programmers are limited
- Alignment & Save time

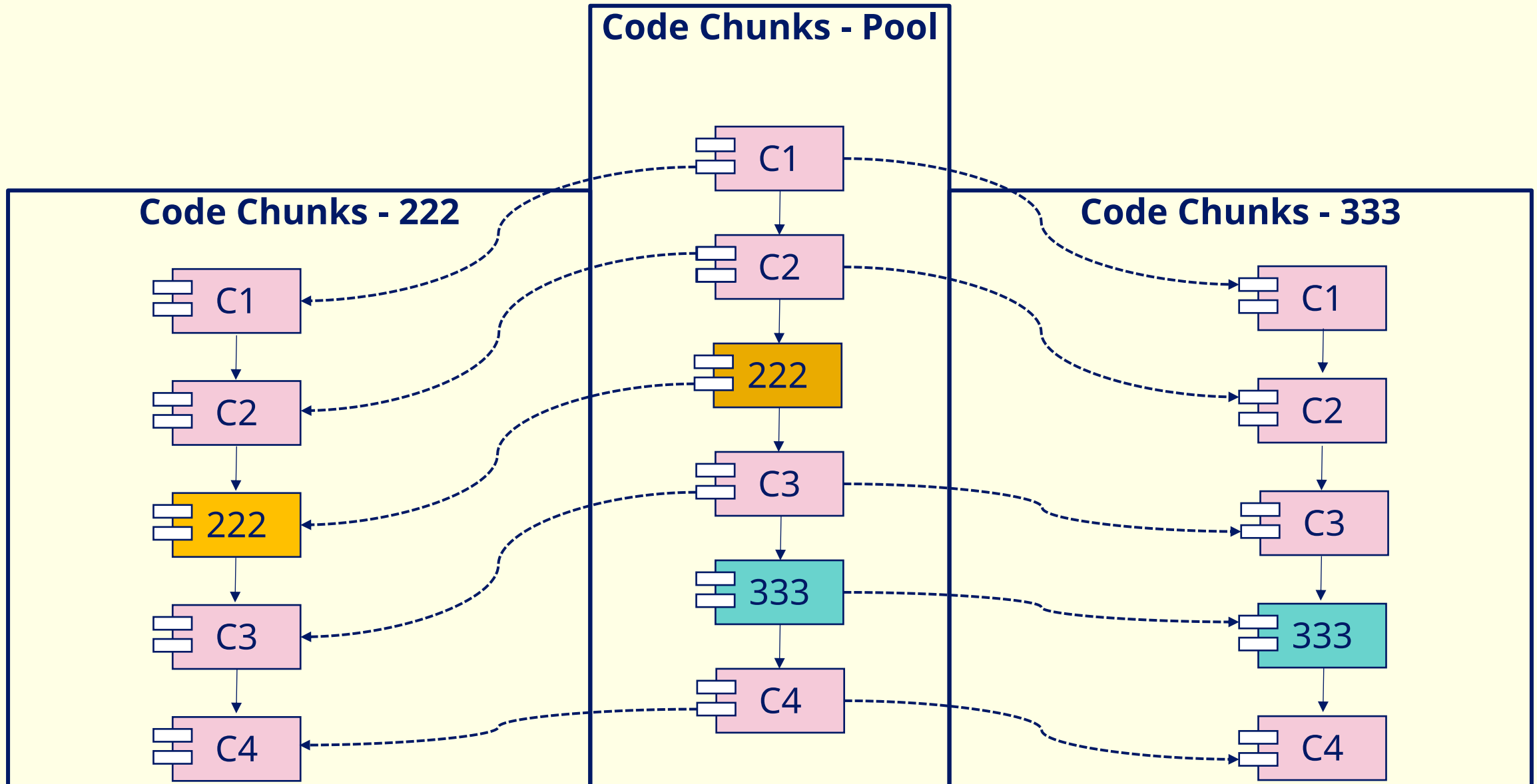
HOW

- Pooled data from the start
- Codes in one repository & “slice” to individual repo
- Everyone is responsible for all pooled trials
- You must coordinate – one repository

Trials are out

Pools are in





Code Chunks in Pool

```
*** sorting adlb;  
proc sort data= sdtm.ae out=ae;  
  by usubjid aerefid;  
run;
```

C1

```
*** get adsl & merge;  
proc sql;  
  create table ae_1 as  
  select a.*, b.trtsdt, b.trtedt  
  from ae as a inner join  
    adam.adsl(where = ("SOME FILTER")) as b  
  on a.usubjid = b.usubjid  
  ;  
quit;
```

C2

```
*** derive variables;  
*>>222;*>>444;  
proc sql;  
  create table ae_1 as  
  select a.*, b.cerefid  
  from ae_1 as a left join  
    sdtm.ce as b  
  on a.usubjid = b.usubjid and  
    a.aelnkid = b.celnkid  
  order by usubjid, aerefid  
  ;  
quit;  
*<<222;*<<444;
```

222

```
*** final code;  
data final;  
  set ae_1;  
  "Anything else to derive?";  
run;
```

C4

Code Chunks in Pool

```

*** sorting adlb;
proc sort data= sdtm.ae out=ae;
  by usubjid aerefid;
run;

*** get adsl & merge;
proc sql;
  create table ae_1 as
  select a.*, b.trtsdt, b.trtedt
  from ae as a inner join
    adam.adsl(where = ("SOME FILTER")) as b
  on a.usubjid = b.usubjid
  ;
quit;

*** derive variables;
*>>222;*>>444;
proc sql;
  create table ae_1 as
  select a.*, b.cerefid
  from ae_1 as a left join
    sdtm.ce as b
  on a.usubjid = b.usubjid and
    a.aelnkid = b.celnkid
  order by usubjid, aerefid
  ;
quit;
*<<222;*<<444;

*** final code;
data final;
  set ae_1;
  "Anything else to derive?";
run;

```

C1

C2

222

C4

>>Trial#

Codes

<<Trial#

Code Chunks in Pool

```

*** sorting adlb;
proc sort data= sdtm.ae out=ae;
  by usubjid aerefid;
run;

*** get adsl & merge;
proc sql;
  create table ae_1 as
  select a.*, b.trtsdt, b.trtedt
  from ae as a inner join
    adam.adsl(where = ("SOME FILTER")) as b
  on a.usubjid = b.usubjid
  ;
quit;

*** derive variables;
*>>222;*>>444;
proc sql;
  create table ae_1 as
  select a.*, b.cerefid
  from ae_1 as a left join
    sdtm.ce as b
  on a.usubjid = b.usubjid and
    a.aelnkid = b.celnkid
  order by usubjid, aerefid
  ;|
quit;
*<<222;*<<444;

*** final code;
data final;
  set ae_1;
  "Anything else to derive?";
run;

```

C1

C2

222

C4

Code Chunks in 222

```

*** sorting adlb;
proc sort data= sdtm.ae out=ae;
  by usubjid aerefid;
run;

*** get adsl & merge;
proc sql;
  create table ae_1 as
  select a.*, b.trtsdt, b.trtedt
  from ae as a inner join
    adam.adsl(where = ("SOME FILTER")) as b
  on a.usubjid = b.usubjid
  ;
quit;

*** derive variables;
*>>222;*>>444;
proc sql;
  create table ae_1 as
  select a.*, b.cerefid
  from ae_1 as a left join
    sdtm.ce as b
  on a.usubjid = b.usubjid and
    a.aelnkid = b.celnkid
  order by usubjid, aerefid
  ;|
quit;
*<<222;*<<444;

*** final code;
data final;
  set ae_1;
  "Anything else to derive?";
run;

```

C1

C2

222

C4

Code Chunks in Pool

```

*** sorting adlb;
proc sort data= sdtm.ae out=ae;
  by usubjid aerefid;
run;

*** get adsl & merge;
proc sql;
  create table ae_1 as
  select a.*, b.trtsdt, b.trtedt
  from ae as a inner join
    adam.adsl(where = ("SOME FILTER")) as b
  on a.usubjid = b.usubjid
  ;
quit;

*** derive variables;
*>>222;*>>444;
proc sql;
  create table ae_1 as
  select a.*, b.cerefid
  from ae_1 as a left join
    sdtm.ce as b
  on a.usubjid = b.usubjid and
    a.aelnkid = b.celnkid
  order by usubjid, aerefid
  ;|
quit;
*<<222;*<<444;

*** final code;
data final;
  set ae_1;
  "Anything else to derive?";
run;

```

C1

C2

222

C4

Code Chunks in 333

```

*** sorting adlb;
proc sort data= sdtm.ae out=ae;
  by usubjid aerefid;
run;

*** get adsl & merge;
proc sql;
  create table ae_1 as
  select a.*, b.trtsdt, b.trtedt
  from ae as a inner join
    adam.adsl(where = ("SOME FILTER")) as b
  on a.usubjid = b.usubjid
  ;
quit;

*** derive variables;
*** final code;
data final;
  set ae_1;
  "Anything else to derive?";
run;

```

C1

C2

C4

```

* Variables from ADAM.ADSL are attached.;
data dat_3;
  length usubjid $60.;
  merge dat 2(in=a) adam.adsl(in=b keep=usubjid randdt trtsdt
/*>>1234*//*>>4567*/
  montsdt montedt
/*<<1234*//*<<4567*/
/*>>1234*//*>>2222*//*>>3333*/
  bolperdt
/*<<1234*//*<<2222*//*<<3333*/
  saffl fasfl sex age

);
by usubjid;
if a and b;
run;

```

Pool

```

* Variables from ADAM.ADSL are attached.;
data dat_3;
  length usubjid $60.;
  merge dat 2(in=a) adam.adsl(in=b keep=usubjid randdt trtsdt
/*>>1234*//*>>4567*/
  montsdt montedt
/*<<1234*//*<<4567*/
  saffl fasfl sex age

);
by usubjid;
if a and b;
run;

```

4567

Specs in Pool

include_in_trial	include_in_submission	deliverable	corefl	table	column	label	order	type	length
Y	Y	1234		ADSL	DCTFL	Discontinuation from Treatment Flag		C	2
Y	Y	-1234		ADSL	DCTFL	Discontinuation from Treatment Flag		C	2
Y	Y			ADSL	DCTREAS	Reason for Discontinuation of Treatment		C	200
Y	Y			ADSL	DCTREASP	Reason Specify for Discont of Treatment		C	200

Specs in 1234

include_in_trial	include_in_submission	deliverable	corefl	table	column	label	order	type	length
Y	Y	1234		ADSL	DCTFL	Discontinuation from Treatment Flag		C	2
Y	Y			ADSL	DCTREAS	Reason for Discontinuation of Treatment		C	200
Y	Y			ADSL	DCTREASP	Reason Specify for Discont of Treatment		C	200

```

1 # to Trial 1234 -----
2
3 trial <- "1234"
4
5 # Slice Place, CST and external data if any
6 metaslice(
7   "place",
8   "cst",
9   "external_data",
10  in_instance = "current",
11  in_trial = "pool",
12  in_project = "someNNproject",
13  out_trial = trial
14 )
15
16 # Slice programs
17 slice(
18   "macros",
19   "metadataprogram",
20   "adamprog" = c("program1.R", "program2.R", "program3.R", "program4.R",
21                 "program1.sas", "program2.sas", "program3.sas", "program4.sas" ),
22   "statprog",
23   "parprog/adamprog",
24   "parprog/statprog",
25   in_trial = "pool",
26   in_project = "someNNproject",
27   out_trial = trial
28 )|

```

Graph

Commit



AUTO: metadataprogram re-generated from pool.

62dbde73 MVEH(Mahendran Venkatachalam) Today at 10:05 AM



AUTO: macros re-generated from pool.

b7eea16d MVEH(Mahendran Venkatachalam) Today at 10:05 AM

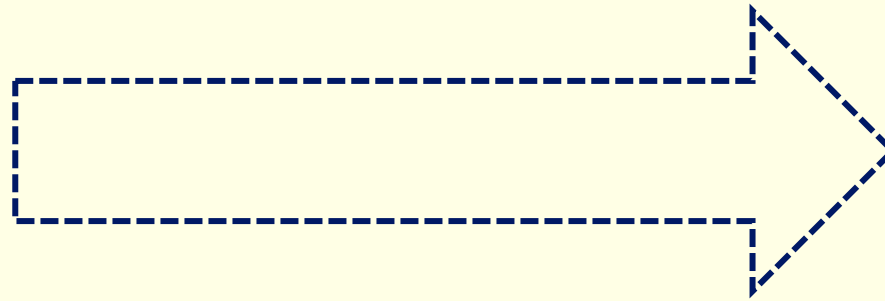
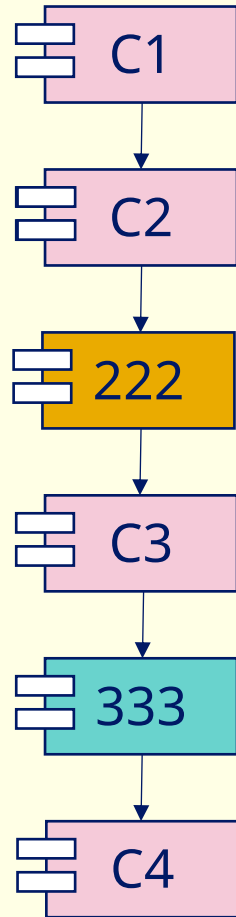


AUTO: adamprog re-generated from pool.

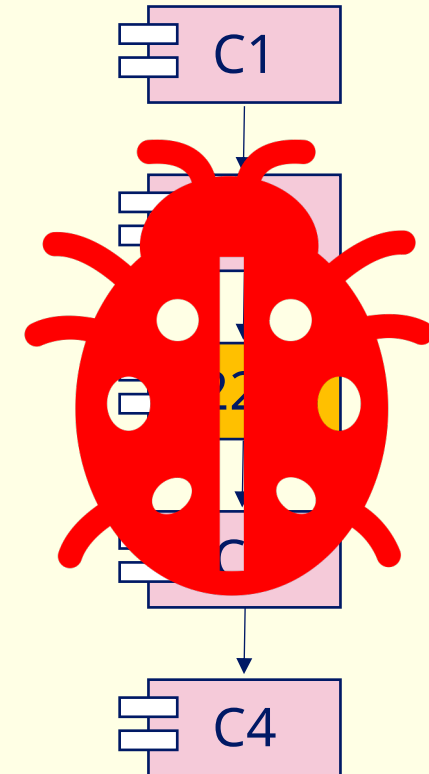
7c485abd MVEH(Mahendran Venkatachalam) Today at 9:52 AM

- **Any external data's**
- **Metadata programs**
- **Macros**
- **Parallel program**
- **Stat Analysis**
- **Output metadata**
- **Documentations**

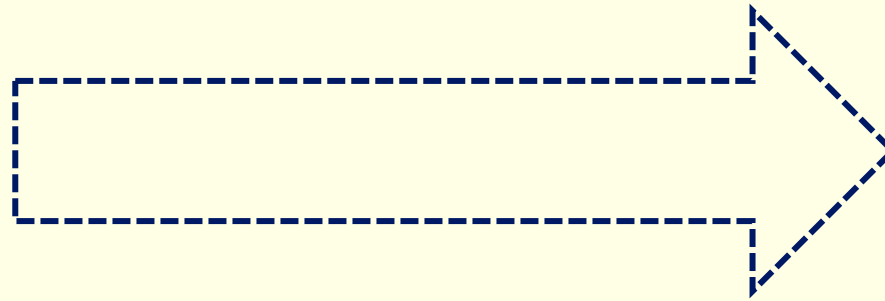
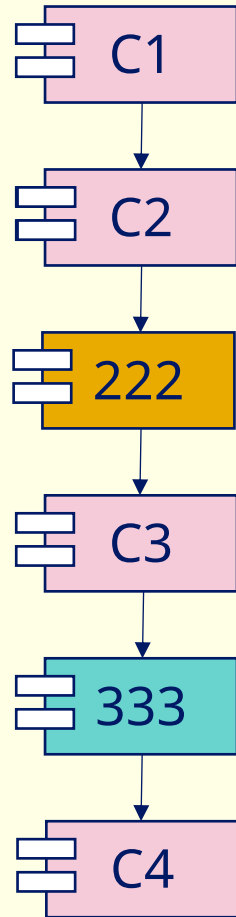
Pool



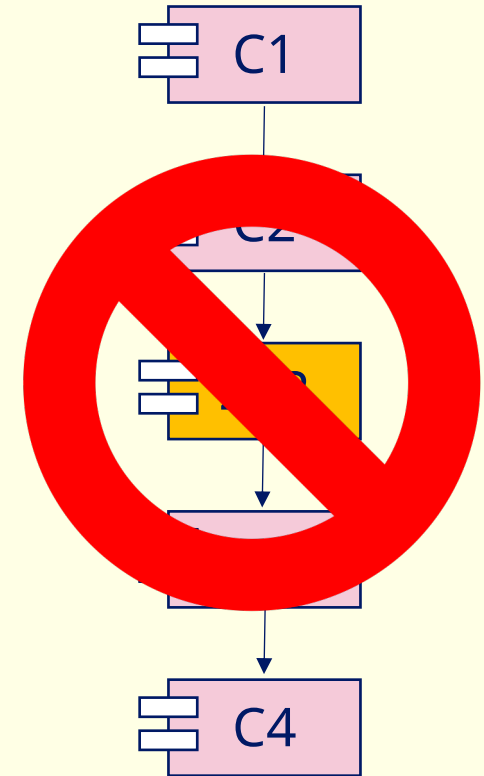
Trial (222)

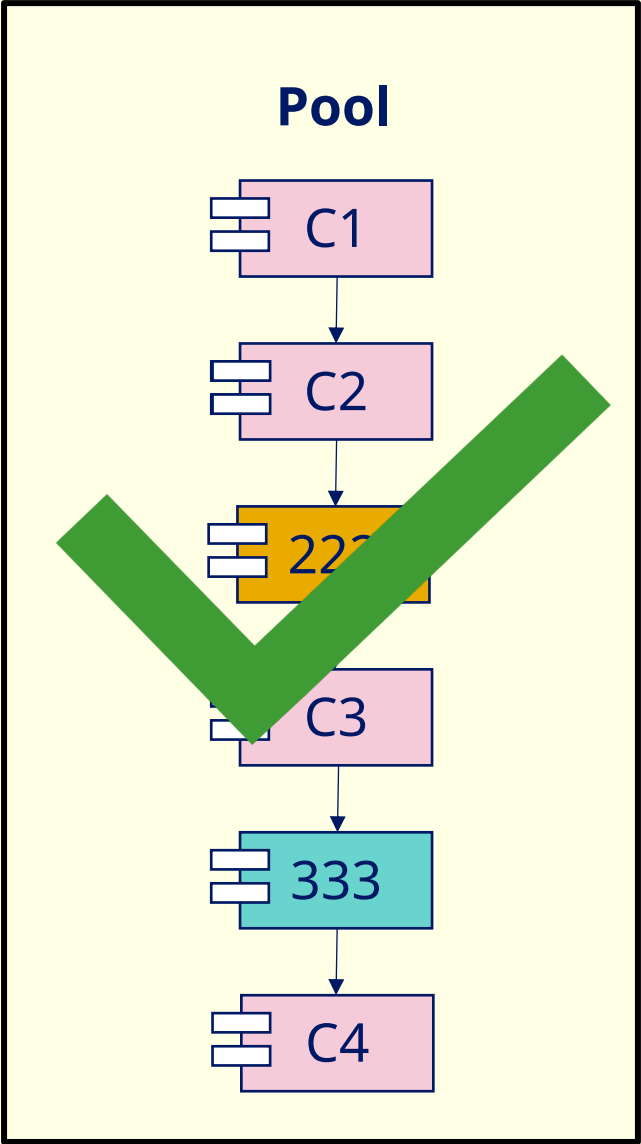


Pool

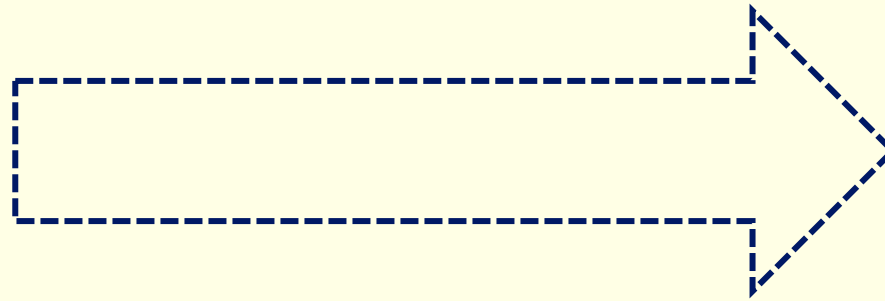
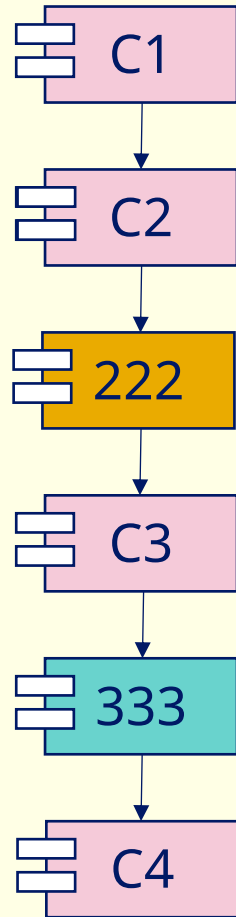


Trial (222)

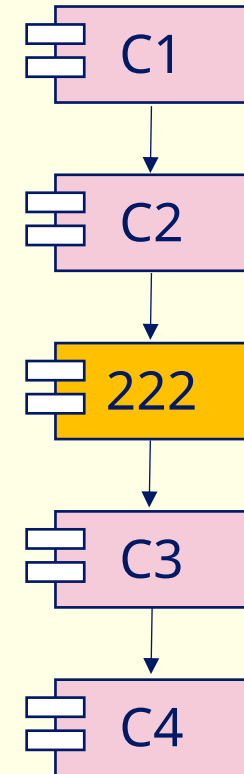




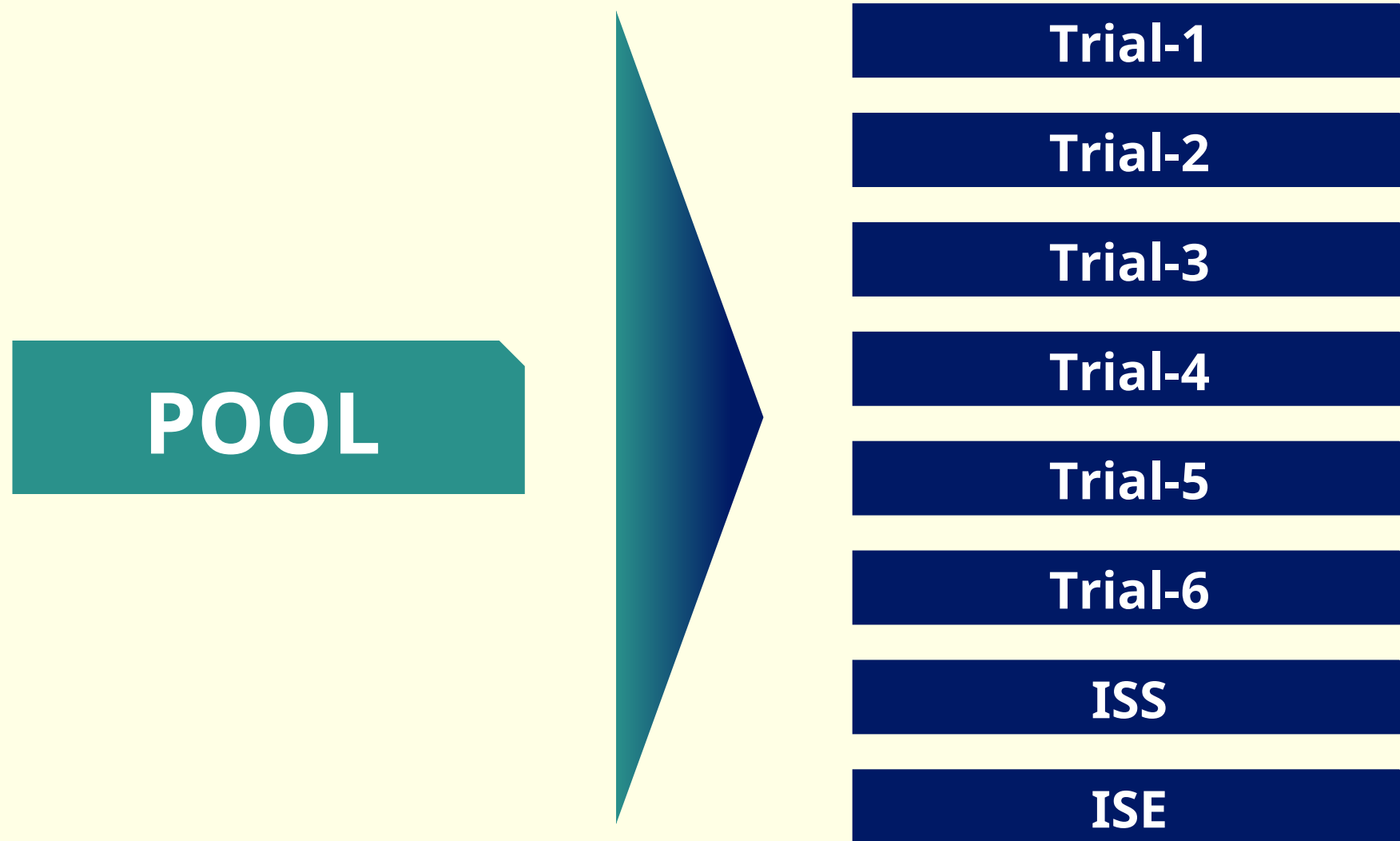
Pool



Trial (222)



**Do not touch
trial folders**



☐ **Pooled data ready for submission**

☐ **Limited resources**

☐ **Time**

☐ **Alignment**

Trials are out

Pools are in

Thank you!

Slicing Codes:

A Modular Approach to ADaM Programming

Mahendran Venkatachalam
Statistical Programming Specialist

DISCLAIMER:

The views and opinions expressed in this presentation belong solely to the presenter and do not necessarily reflect those of the presenter's employer, organization, committee, or any other group or individual. The purpose of this presentation is to highlight the experience gained in one of our projects. No data or confidential information belonging to Novo Nordisk has been utilized in this presentation.