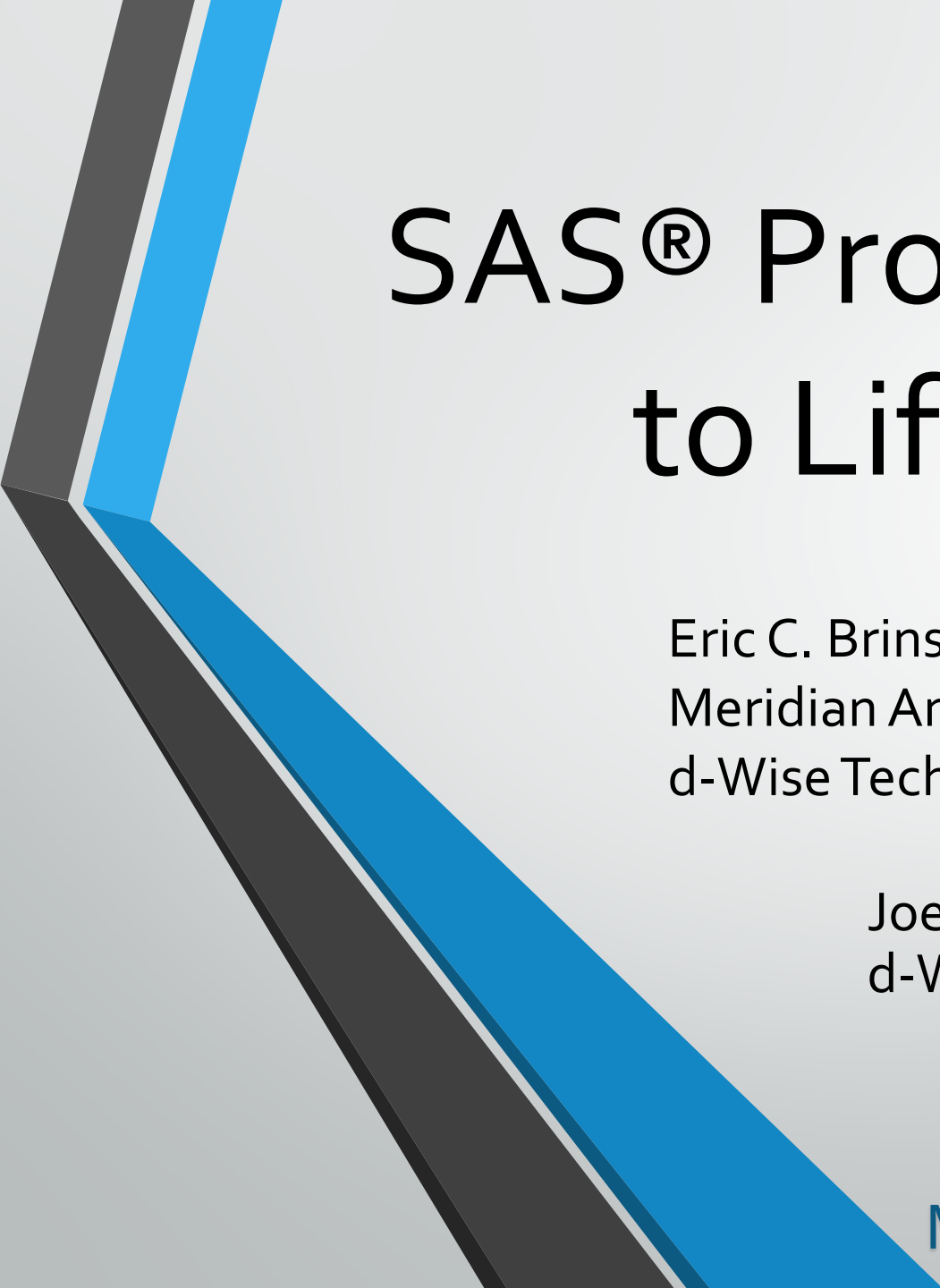




SAS® Programmer's Guide to Life on the SAS Grid

Eric C. Brinsfield

Meridian Analytics, Virginia Beach, VA USA

d-Wise Technologies, Morrisville, NC USA

Joe Olinger

d-Wise Technologies, Morrisville, NC USA

Meridian Analytics

d-wise

Preliminaries

- Please hold questions to the end, unless you need a clarification.
- Survey – raise your hand if:
 - You know what SAS Grid Manager is?
 - You may be moving to the SAS Grid soon?
 - You are already running on SAS Grid?
 - You are curious if you might want to move to the SAS Grid?
 - Not sure why you are here?

Objectives

Today, I want to stress three major points:

1. SAS on the SAS Grid is the same SAS you already know
2. Understanding the SAS Grid architecture will help you
3. Learning SAS Grid functions and features are not essential, but do provide great benefits for the right situation.



Introduction

Use Case

Justification

Use Case

Many configurations of SAS Grid are possible, but the configuration for today's discussion is such that:

- All SAS processing executes on the SAS Grid Platform
- SAS clients (not SAS) are installed on Citrix
 - All SAS clients must connect to the SAS Grid
- All PC SAS and other SAS licenses will expire after migration
- SAS Grid Manager is installed on Linux servers

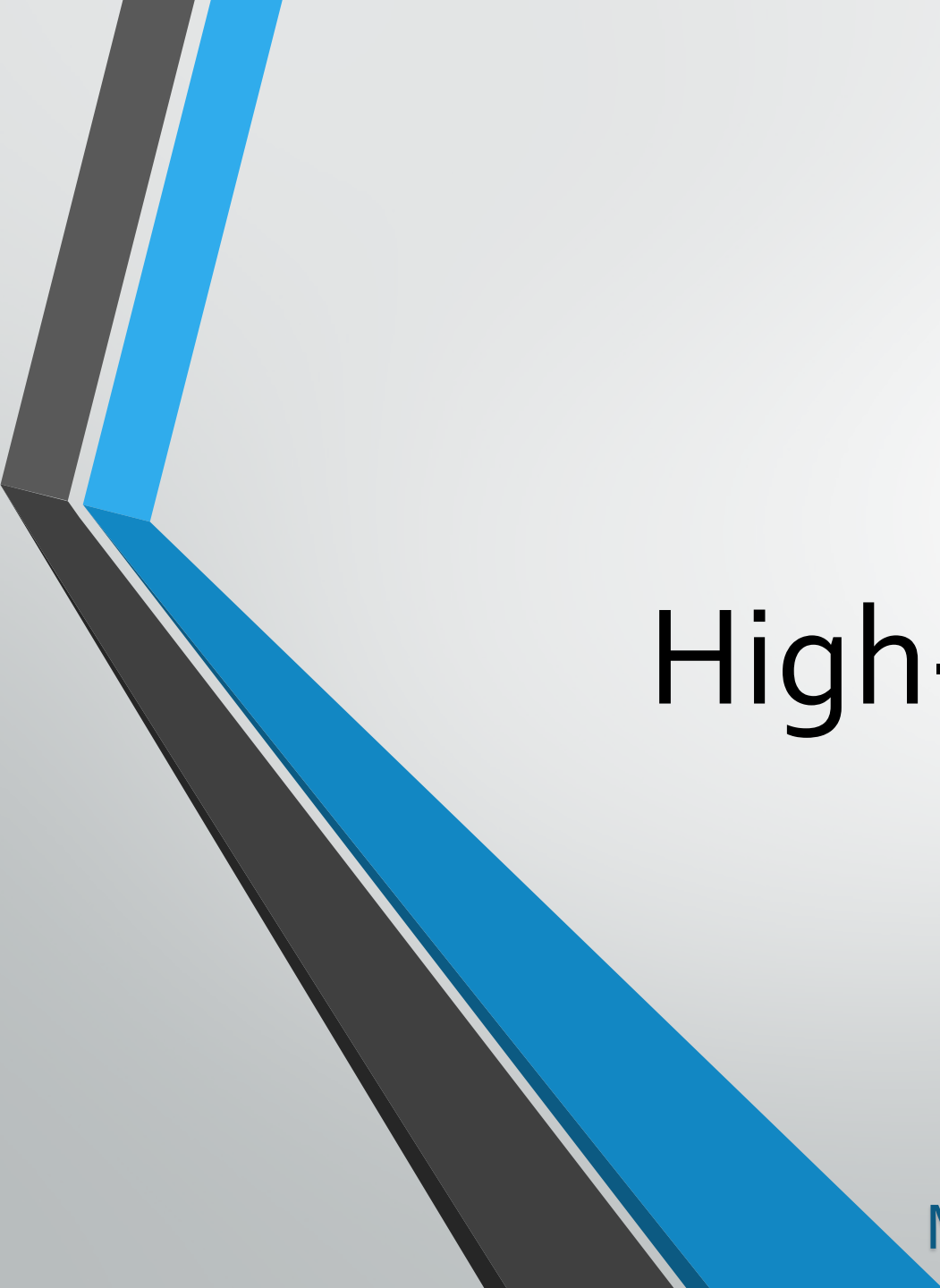
Justification with implications

Pros

- Reduced licensing costs
- Better inventory control
- Increased license utilization
- Centralized support
- Centralized maintenance
- Enhanced capability per person
- Reduced overall run-time for big jobs

Cons

- Downtimes for maintenance create global impact
- Network speeds are critical
- Platform sizing is critical
- Centralized support
- Centralized maintenance
- Users must share resources
- PC SAS users could be sad
- Windows users must adapt to Linux



High-Level Implications

Key Concepts

Mostly - No Change

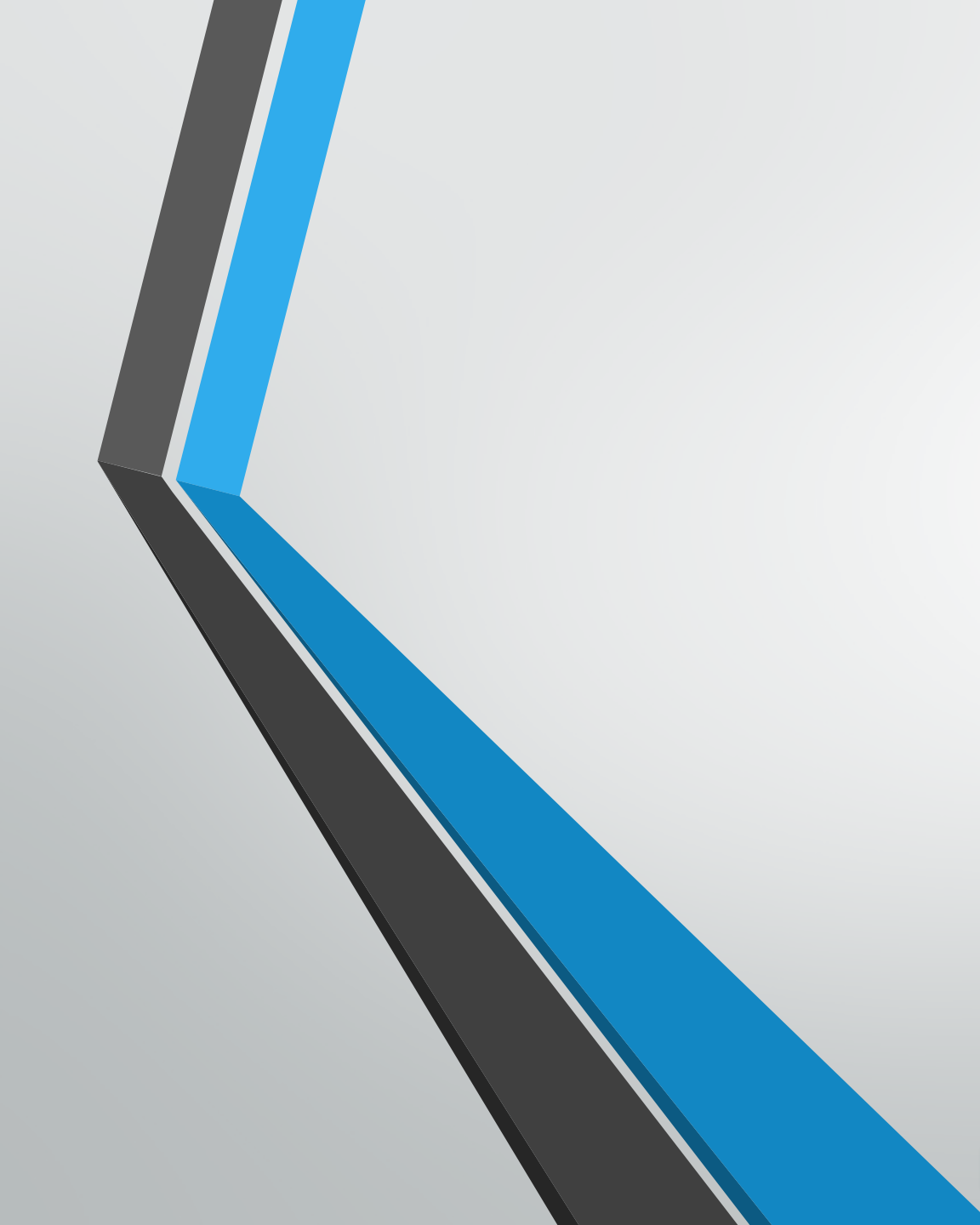
After connecting Enterprise Guide or SAS Studio to the SAS Grid,

- Both clients work the same way they do on your PC
 - You do not have to use the parallel processing
 - You can defer learning all the:
 - “grid” functions and statements
 - RSUBMIT features for the SAS Grid
- Get up and running first, then venture into new capabilities

Batch processing is a little bit different, but not hard to learn

SAS Clients

- Very little SAS Grid knowledge required
- Examples of SAS Clients that submit to the SAS Grid
 - SAS Studio and SAS Enterprise Guide
 - SAS Enterprise Miner and SAS IML Studio
 - SAS Add-in for Microsoft Office
 - SAS Management Console
- Not recommended
 - SAS Display Manager running directly on Linux
(unless you have a customized version that can submit to the SAS Grid)



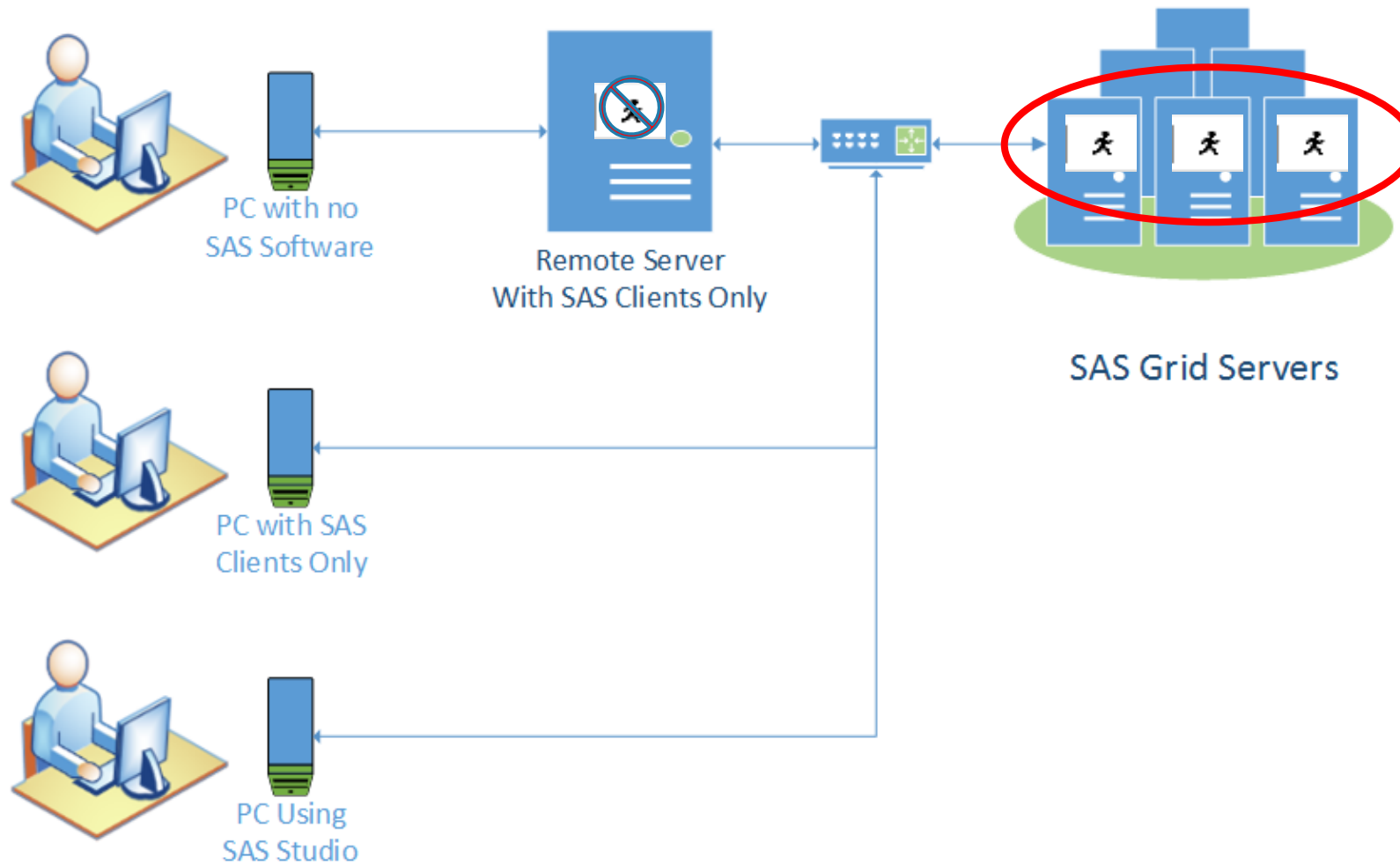
Understanding the Architecture

Helps you use the system more effectively

Meridian Analytics

d-wise

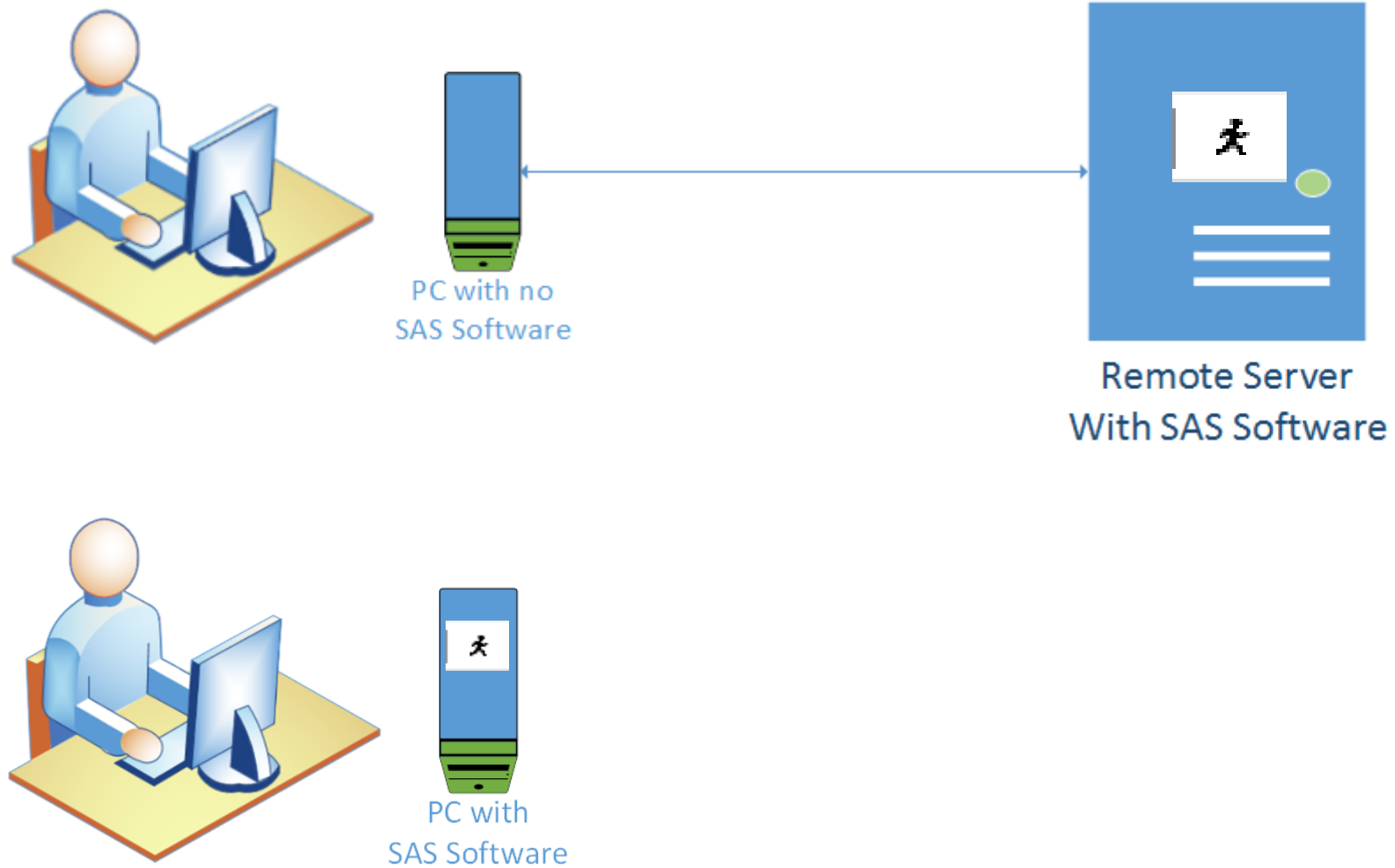
SAS Grid Platform



Meridian Analytics

d-wise

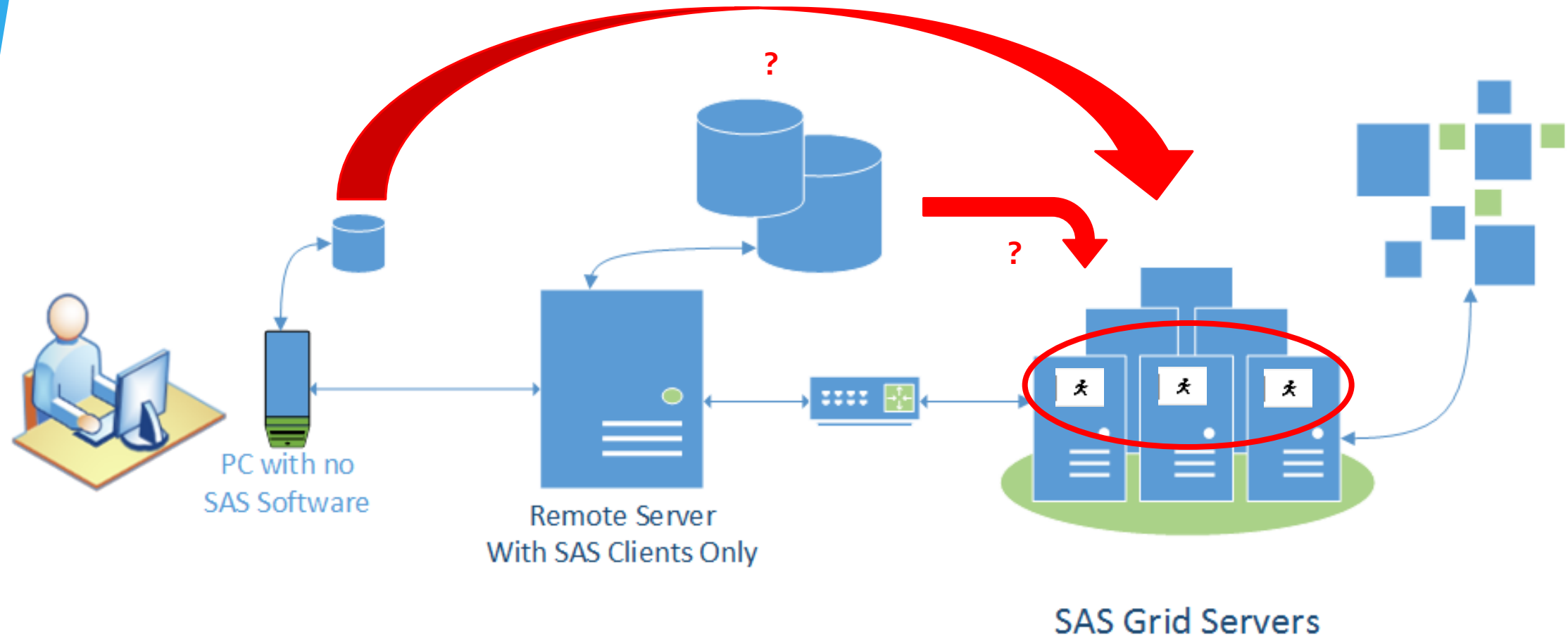
Common “from” Configurations



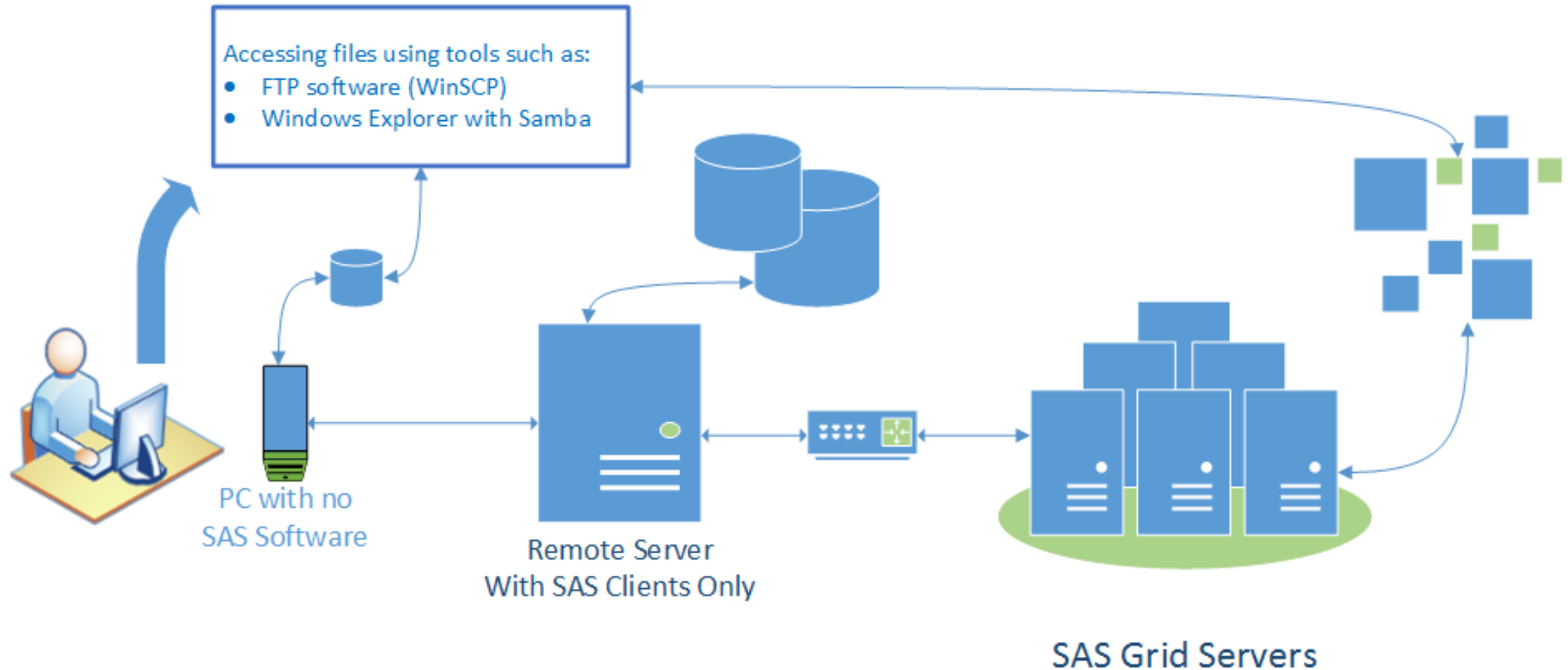
Meridian Analytics

d-wise

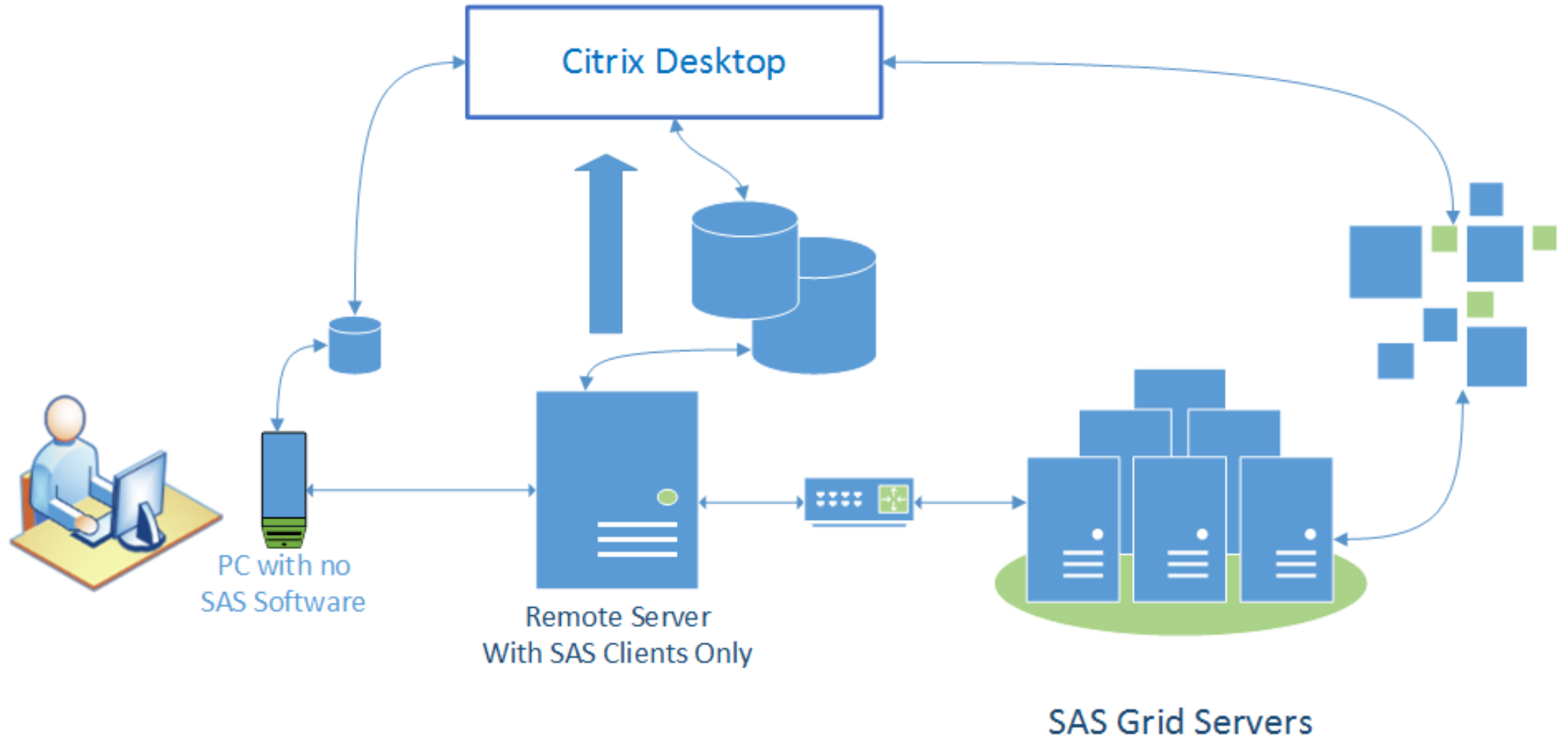
Where are the data? Programs?



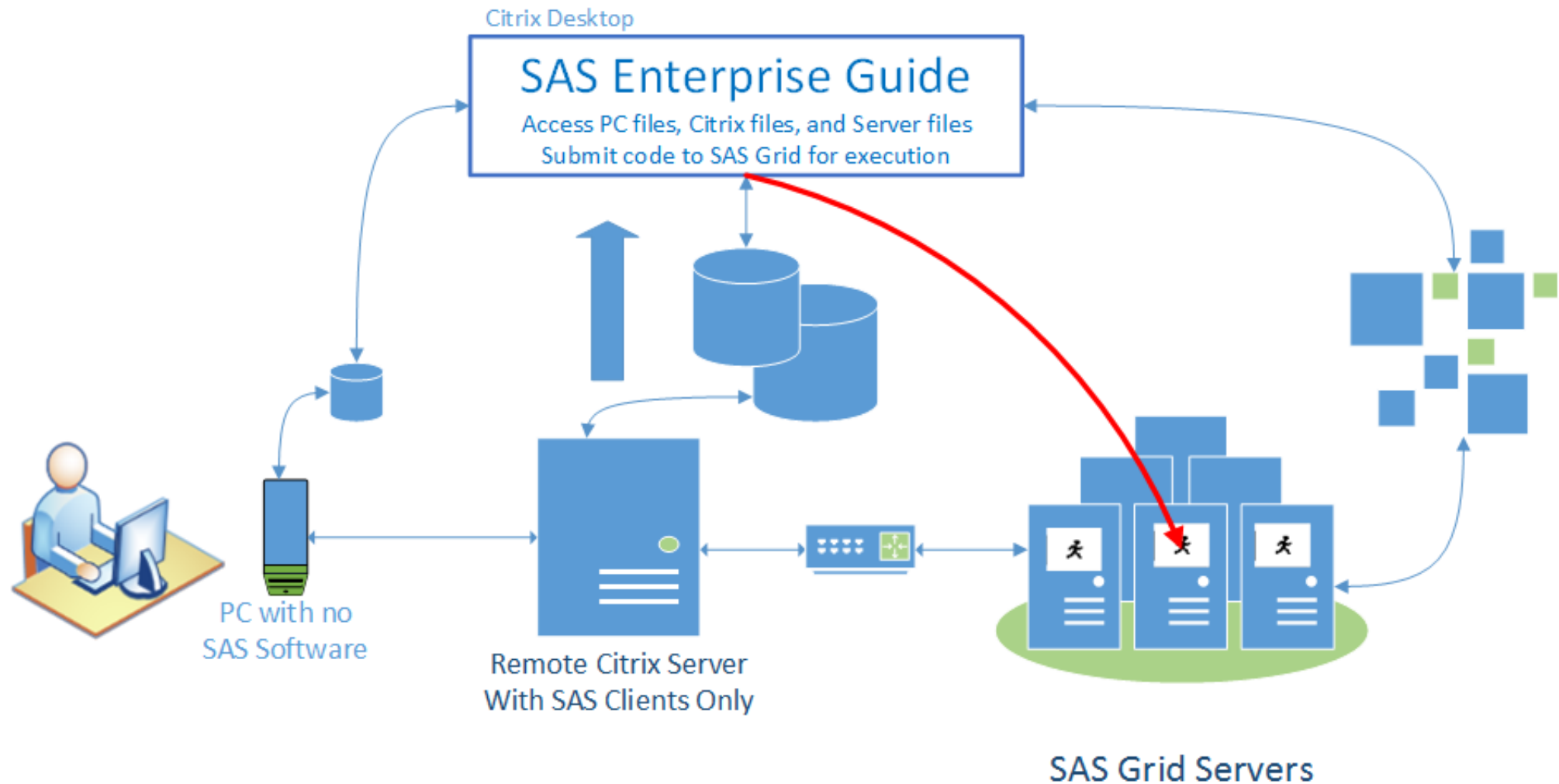
Moving files between PC and Linux



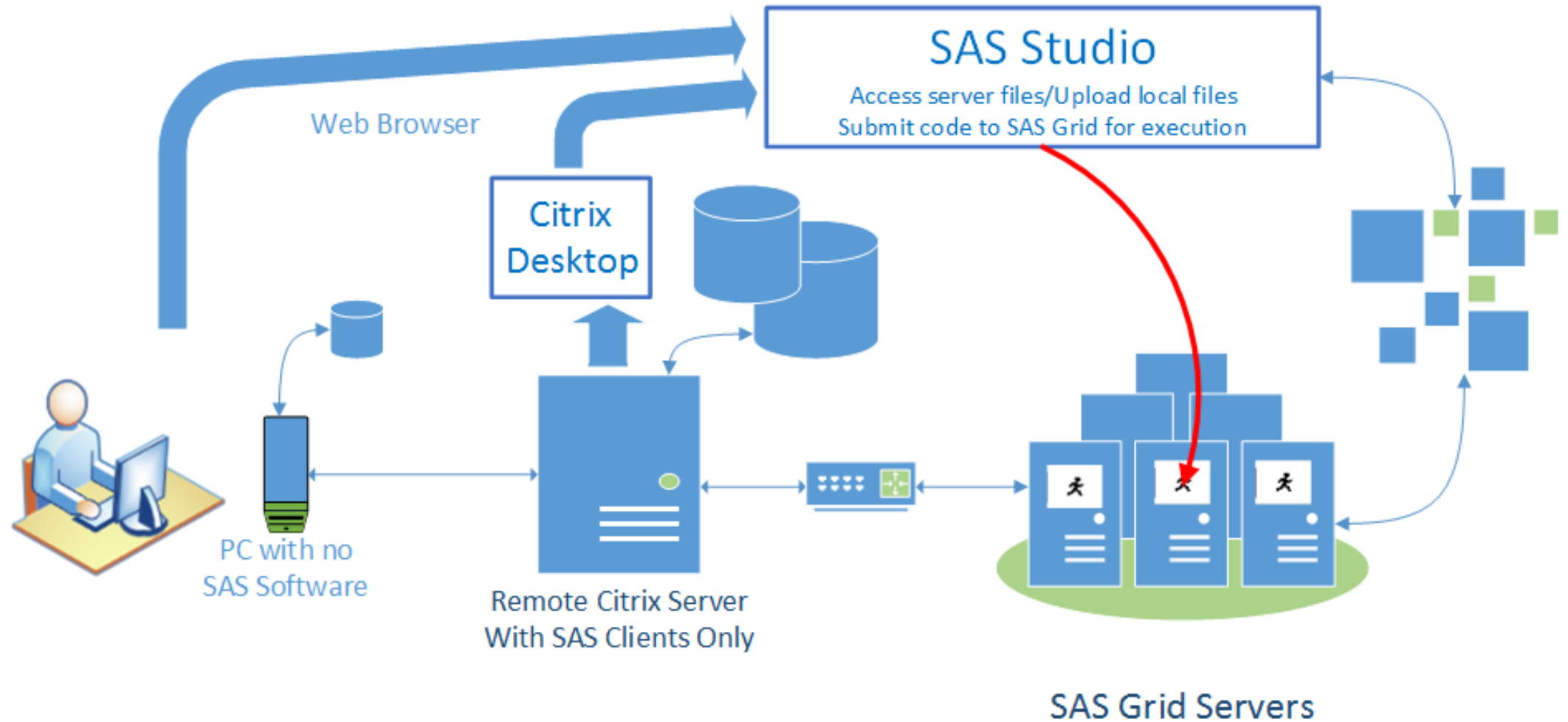
Using your remote desktop options



Accessing files from EG



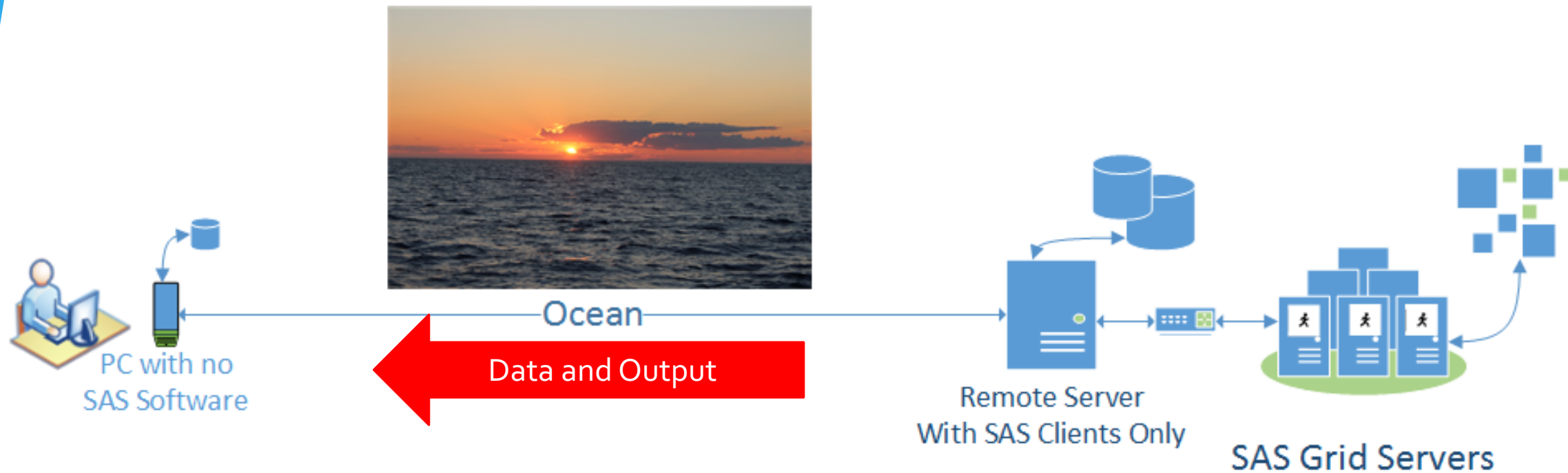
Accessing files from SAS Studio



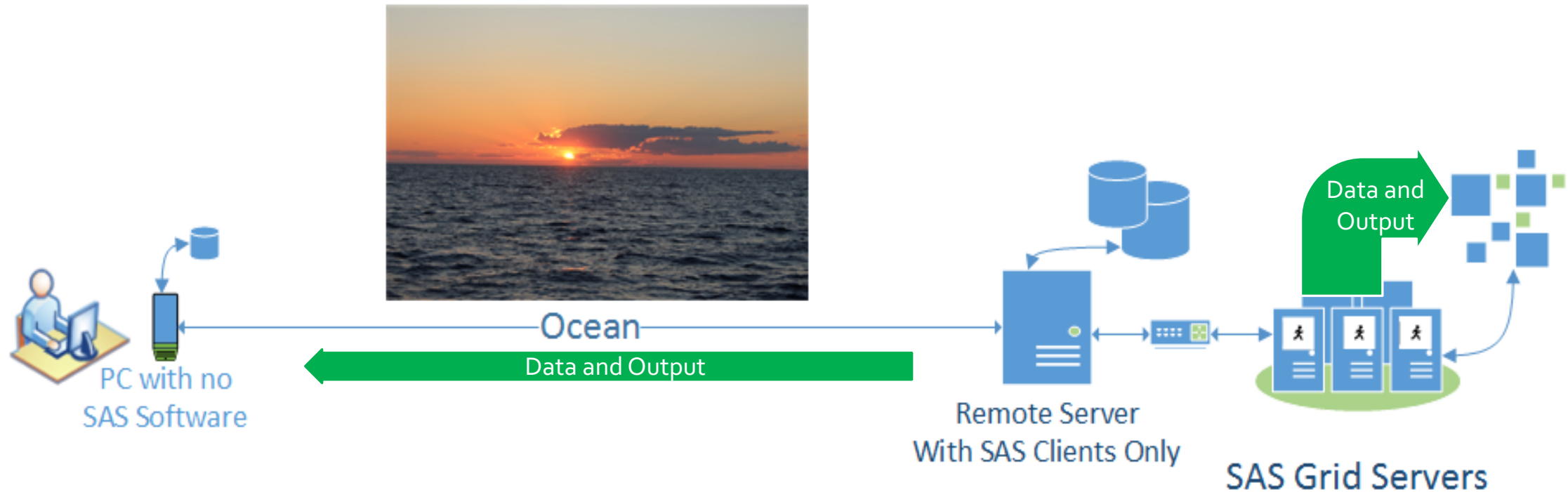
How far away is your grid platform?



Where is the SAS Grid platform?



Route large output to SAS Grid connected storage.



Routing Output

When in EG or SAS Studio, route output to storage that is co-located with the SAS Grid servers using:

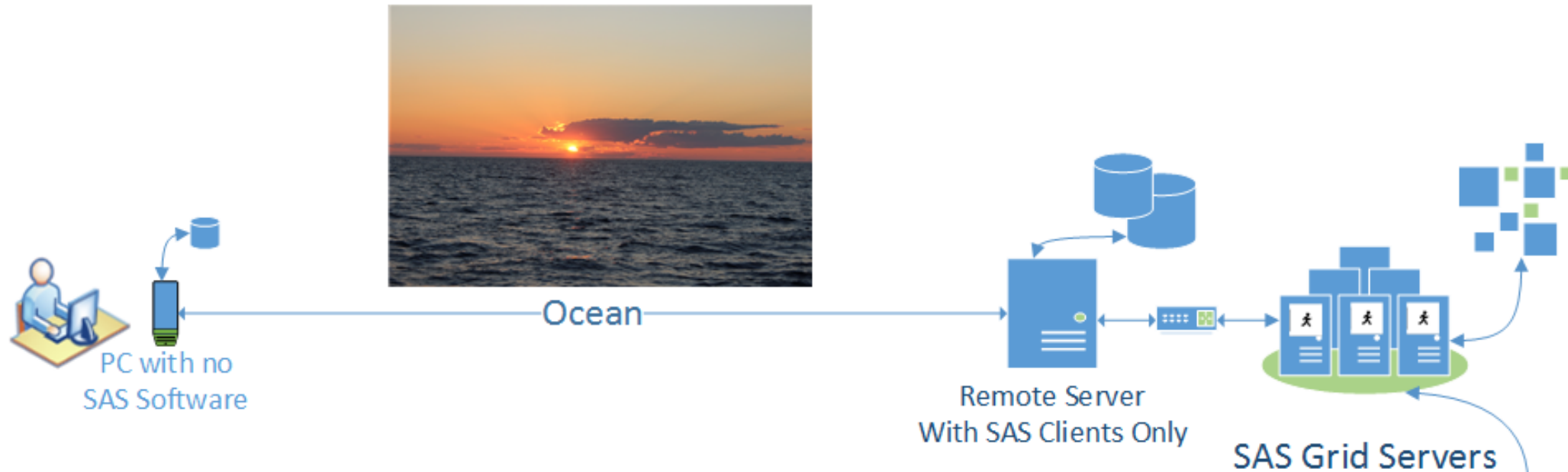
- ODS statements
- PRINTTO Procedure
- FILE statement

Use ODS to route large output to a desired ODS format such as PDF, RTF, or HTML.

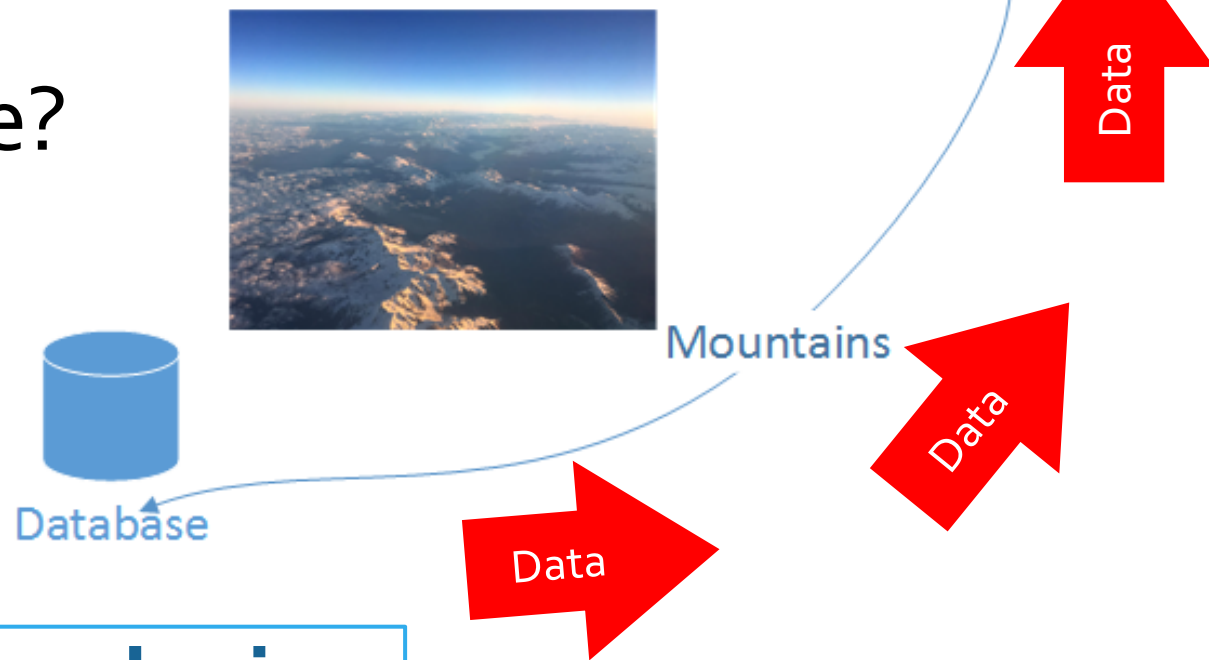
```
ODS _ALL_ CLOSE;  
ODS PDF FILE="&projroot/output/myanalysis.pdf";  
/*--- all of the code to generate the report */  
ODS PDF CLOSE;  
ODS HTML;
```

Even the SAS log can cause problems and all or sections of the SAS log can be routed to a file:

```
PROC PRINTTO LOG="&projroot/saslogs/mc_sim_001.log" NEW;  
RUN;  
/* Insert code here that creates a large SAS log file */  
PROC PRINTTO LOG=LOG; RUN;
```

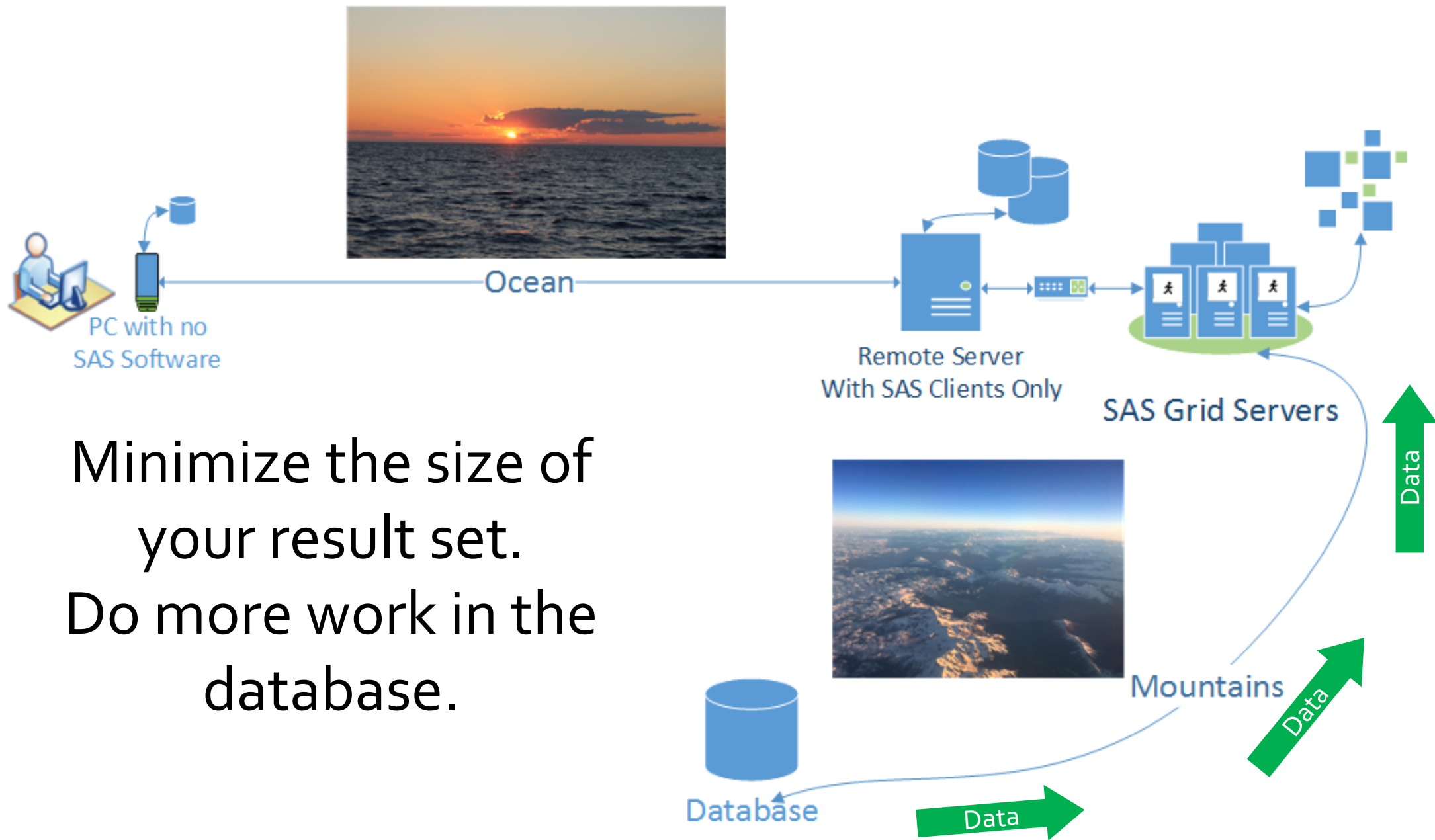


Where is your database?



Meridian Analytics

d-wise



Minimize the size of
your result set.
Do more work in the
database.



SAS Client Considerations

Meridian Analytics

d-wise

Check Enterprise Guide options

Options

General
Project Views
Project Recovery
Results
Results General
Viewer
SAS Report
HTML
RTF
PDF
Graph
Excel
PowerPoint
Stored Process
Data
Data General
Performance
Query
OLAP Data
Tasks
Tasks General
Custom Code
Output Library
SAS Programs
File Comparison
Program History
Security
Administration
Application Logging

Results > Results General

Result Formats

☒ SAS Report	☐ HTML	☐ PDF
☐ RTF	☐ Text output	☐ PowerPoint
☐ Excel		

Default: SAS Report

Managing Results

Replace results: Prompt before replacing

- ☒ Display SAS log when errors occur
- ☐ Automatically open data or results when generated
- ☐ Automatically add output data to the project tree
- ☒ Link handcoded ODS results
- ☒ Change task icon when warnings occur
- ☒ Show generated wrapper code in SAS log

Prompt before opening results larger than: 5 MB

Maximum number of output data sets to add to the project: 50

Check SAS Studio options

Preferences

General

Editor

Results

Tasks

Repositories

Batch

☒ Display warning if results are larger than: 4 MB

HTML output style: Htmlblue

☐ Produce PDF output

PDF output style: Pearl

☐ Generate the default table of contents

☐ Produce RTF output

RTF output style: Rtf

☐ Automatically open generated output data

Reset to Defaults

Save Cancel

Warning: System Options

Both Enterprise Guide and SAS Studio change the following default system options:

- **VALIDVARNAME=ANY vs V7**
Specifies the rules for valid SAS variable names that can be created and processed during a SAS session
- **VALIDMEMNAME=EXTEND or CAPATIBLE**
Specifies the rules for naming SAS data sets, SAS data views, and item stores



Changes in Operating System

Executing SAS on Linux

When coming from
Windows

- Change in data set structure
- Windows integration not supported
 - DDE not supported
 - SAS PC File Server still available through Windows
- Change in system commands
- DMS commands not supported in recommended Grid clients
- Change in scripting language
- Adjustments necessary with file editors
- Check SAS system options

Tools you may need

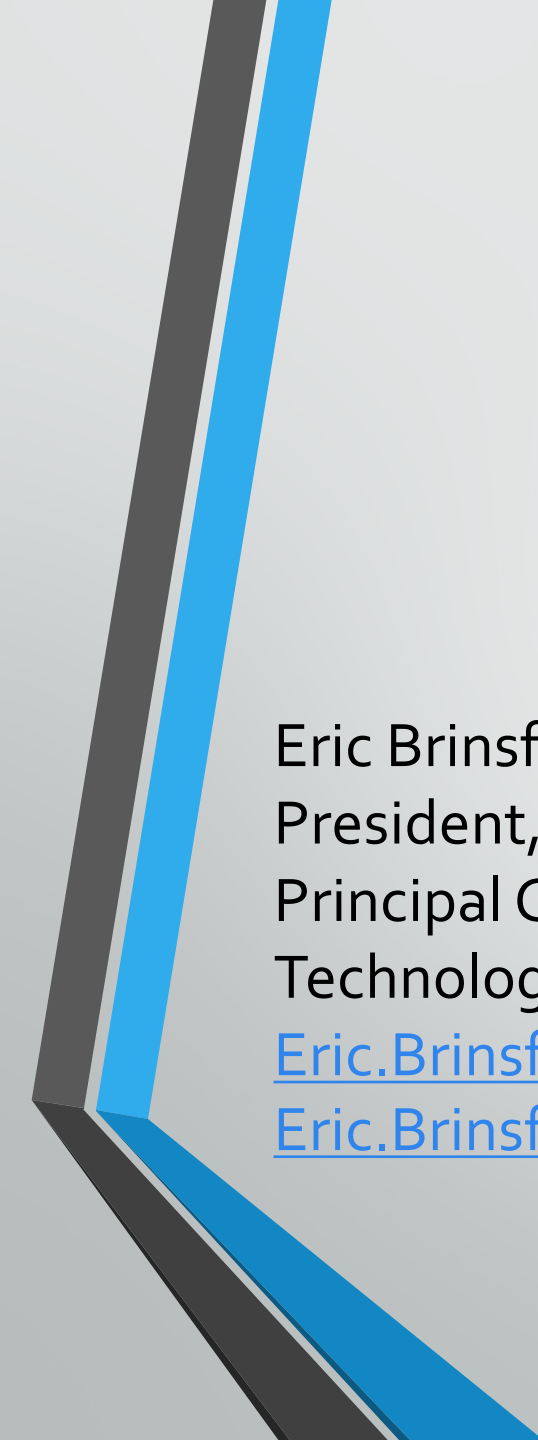
- PROC MIGRATE
- PC File Server
- SAS Add-in for Microsoft Office
- SFTP software
- Third-party editors
- SAS system options

Conclusion

- SAS programming is the same
- Go with the flow (of your architecture)
- Use the capabilities of the SAS Grid when you can
- When you are comfortable on the SAS Grid, try out parallel processing

Recommended Reading

- *Grid Computing in SAS® 9.4, Fifth Edition*
- *Moving and Accessing SAS® Files, Third Edition.*
- *SAS® 9.4 System Options: Reference, Fifth Edition*
- IBM Platform LSF Command Reference.” © Copyright IBM Corporation 1994, 2017. Available at https://www.ibm.com/support/knowledgecenter/SSETD4_9.1.3/lfs_kc_cmd_ref.html



Thank You!

Questions?

Contact Information:

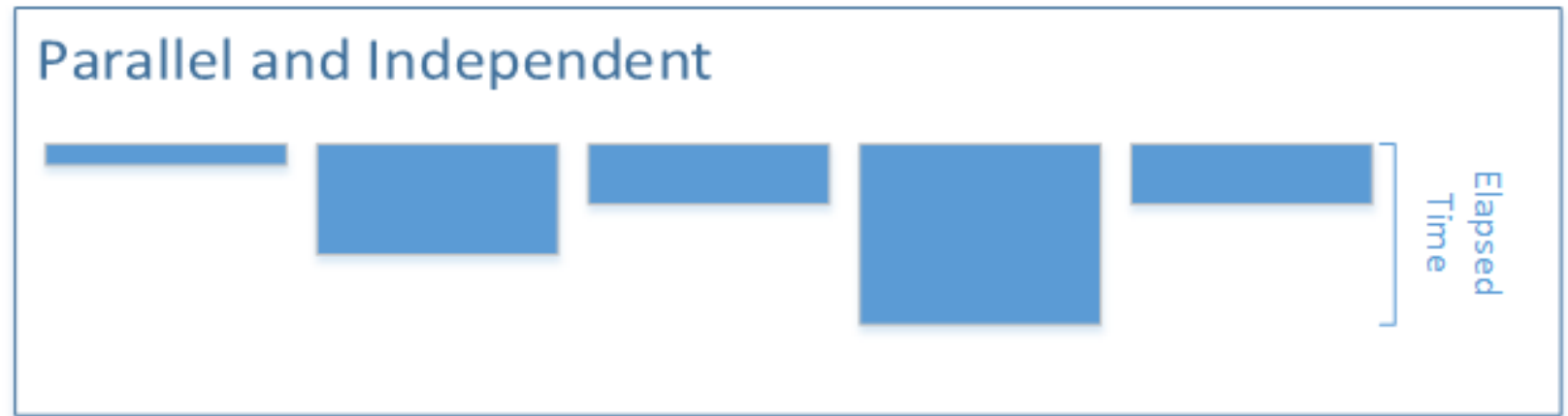
Eric Brinsfield
President, Meridian Analytics
Principal Consultant, d-Wise
Technologies
Eric.Brinsfield@meridiananalytics.com
Eric.Brinsfield@d-wise.com

Joe Olinger
Senior Consultant, d-Wise Technologies
Joe.Olinger@d-wise.com

Meridian Analytics

d-wise

Take Advantage of Parallel Processing



Comparing the Elapse Execution time of a job running programs sequentially vs. running programs in parallel if the programs are independent and parallel processors (nodes) are available.

Sequential time= the sum of the parts

Parallel time = the weakest link

Parallel processing with RSUBMIT

```
%let grid_rc=%sysfunc(grdsvc_enable(_all_,resource=SASApp));  
  
signon batch cmacvar=signonstatus;  
  
rsubmit batch wait=no;  
<<< Enter your full program here >>>  
endrsubmit;
```

Sample “parallel and combine”: setup

```
%let workroot=/SASDATA/shared/workshr;  
%makeName(outname=workdir);  
%mkdir(&workroot,&workdir);  
%let datawrk=&workroot/&workdir;  
libname tempshr "&datawrk";  
%let grid_rc=%sysfunc(grdsvc_enable(_all_,resource=SASApp));
```

Sample “parallel and combine”: job 1

```
%let j=1;  
signon tsk&j cmacvar=signonstatus inheritlib=(tempshr);;  
%syslput j=&j;  
%syslput workdir=&workdir;  
rsubmit tsk&j wait=no connectpersist=no;  
<<< enter code  
endrsubmit;
```

Repeat for job 2

```
%let j=2;  
signon tsk&j cmacvar=signonstatus inheritlib=(tempshr);;  
%syslput j=&j;  
%syslput workdir=&workdir;  
rsubmit tsk&j wait=no connectpersist=no;  
<<< enter code  
endrsubmit;
```

Sample “parallel and combine”: wait and combine

```
Waitfor _all_;
```

```
data combine;
```

```
merge tempshr.test1
```

```
tempshr.test2;
```

```
by .....
```

```
run;
```