

Implementing and Administrating Redundant PI-Advanced Computing Engine (ACE) Servers



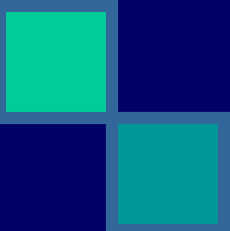

Craig Taylor
PI Administrator



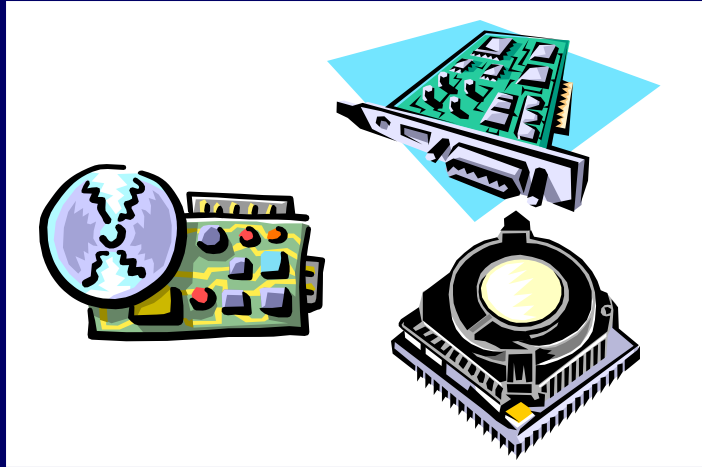
CALIFORNIA ISO



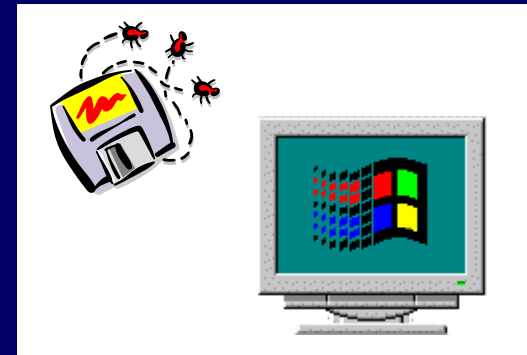
Agenda:

- 
- Redundancy and the ISO PI Infrastructure
 - PI-ACE Calculations ISO style/technique
- 

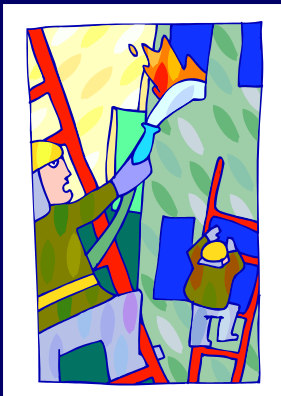
Why redundancy?



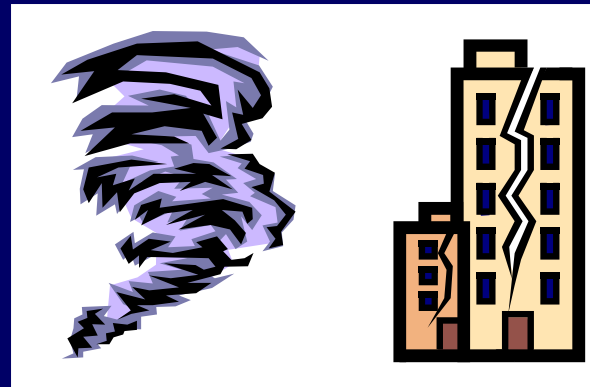
■ Hardware Failure



■ Software Failure




■ Site Disaster



■ Natural Disaster

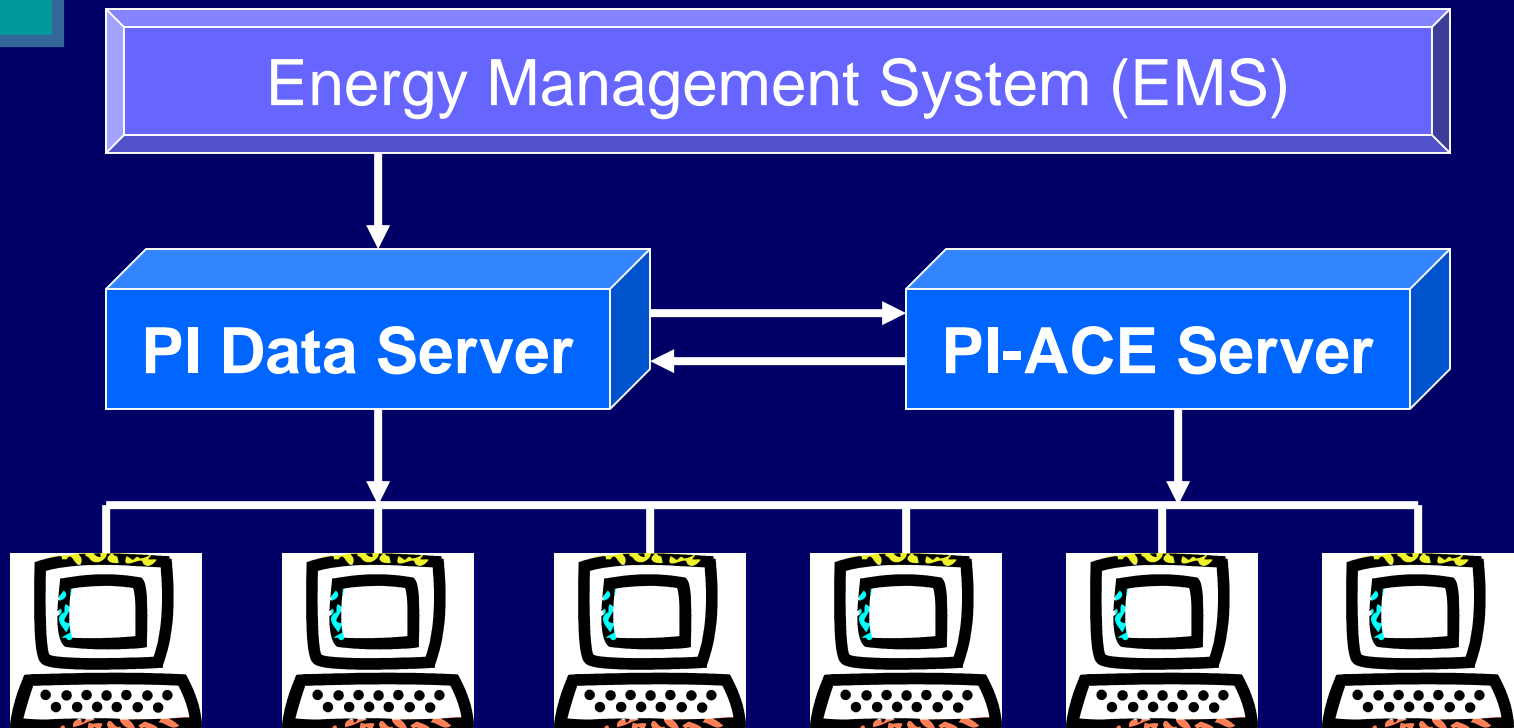


Other Redundancy Benefits...

- Further Ensures Data is Available
 - Lowers the risk of Data Loss
 - Eases Server Maintenance
 - Reduces human errors in Production
- 

ISO PI Infrastructure

- Paired PI Data servers with PI ACE servers
- Majority of data provided via EMS
- PI-ACE Server also Provides Module Database

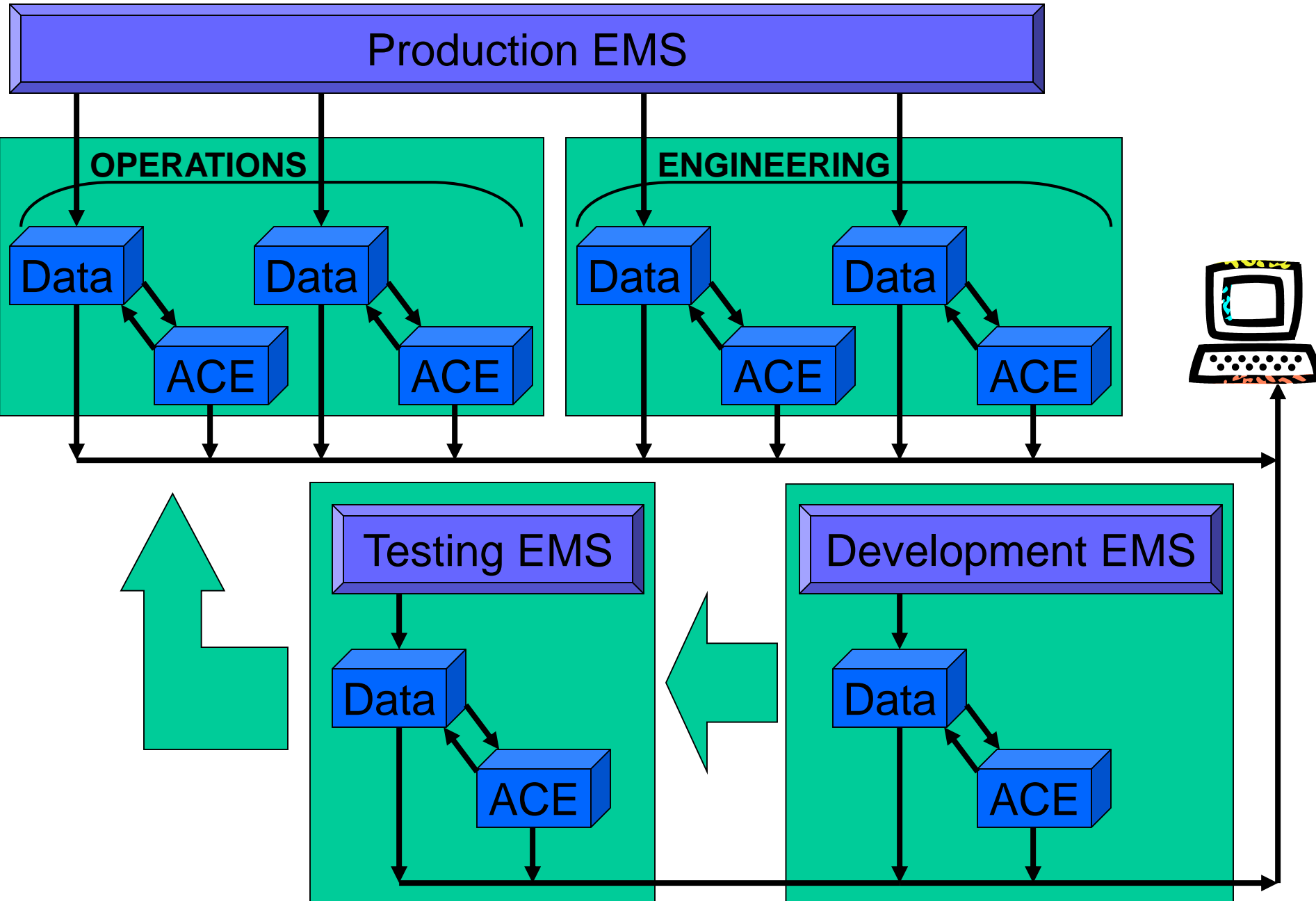




Server Specifications

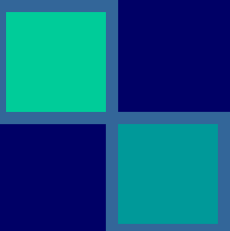

Component	PI Data Servers	PI-ACE Servers
System	HP DL 580	HP DL 380
CPU's	4 x 700Mhz	2 x 2.1 Ghz
Memory	1 Gigabytes	2 Gigabytes
Controller	Built into Mother Board	Built into Mother Board
Outside Network	100 Megabit Full Duplex	100 Megabit Full Duplex
Backup Network	FDDI	N/A
Disk Storage	HP HSG80 Controller Brocade Silkworm 2800 300 Gigabytes	2 x 18 GB Mirrored

ISO PI Infrastructure





Issues with ISO Infrastructure

- 
- 1st: Synchronizing PI Point Database
 - 2nd: Synchronizing PI Module Database
 - 3rd: Distributing the PI-ACE Executables
- 

1st Issue: Synchronizing Point DB

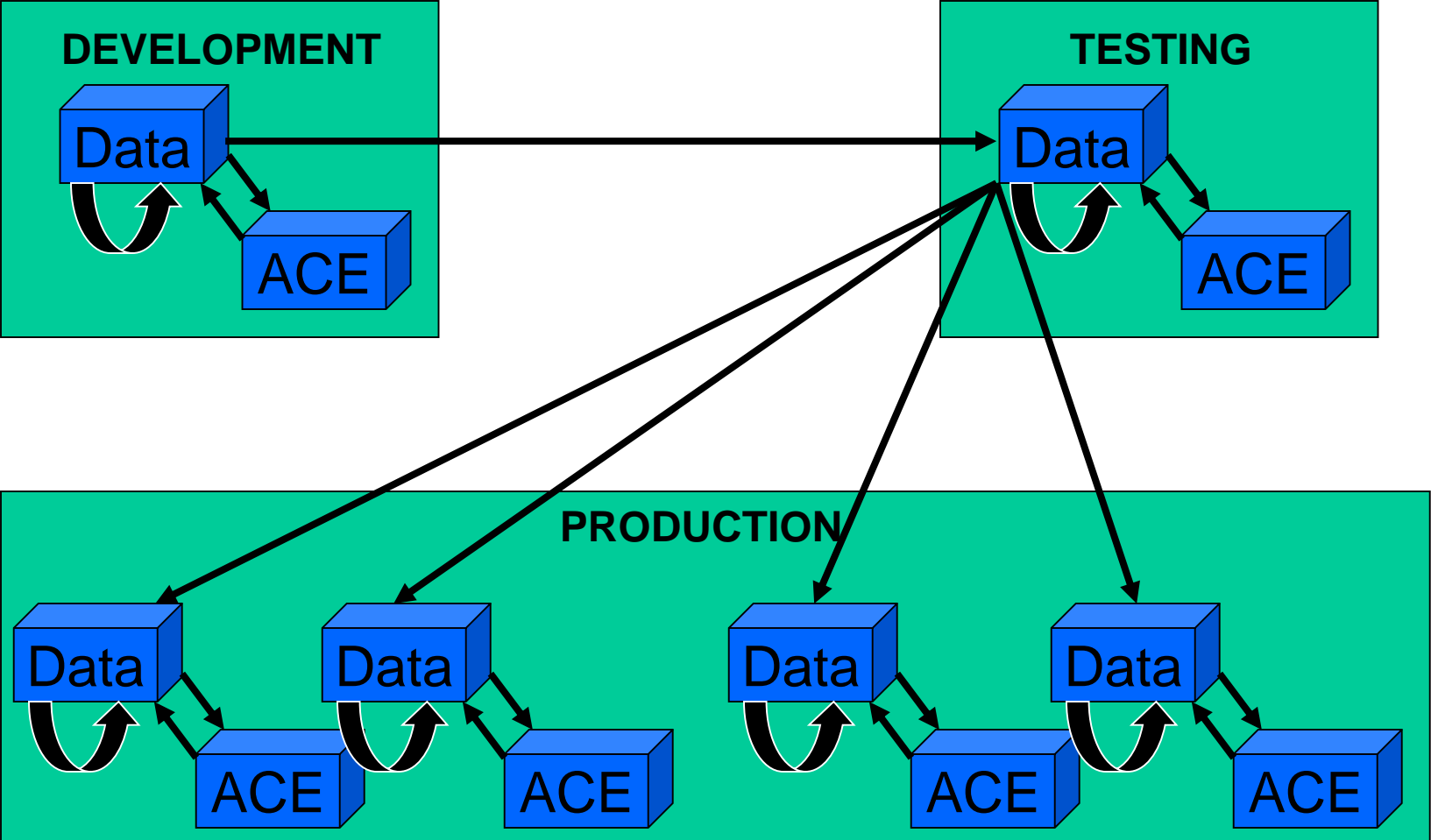
- The PI point database must be identical on all Production PI servers



- Could have achieved this by manual updates to all machines but decided to automate the synchronization to reduce human errors
- Created VB Program to apply PI config files
- Program runs on all servers, PDS distributes

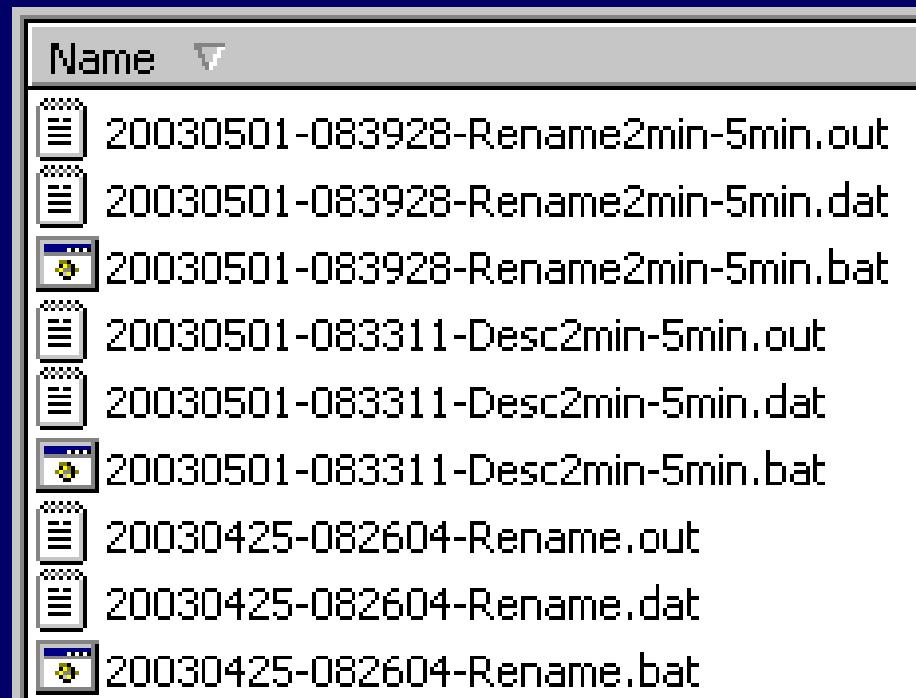
PIAutoConfig Visual Basic Service

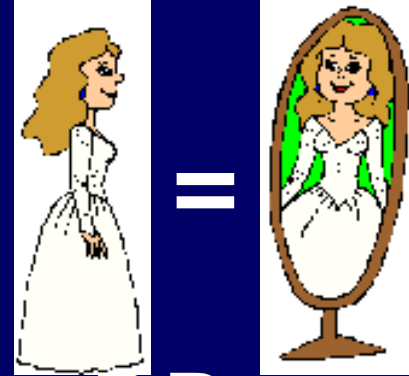
↪ Applies the PI config files to all servers



Key Takeaways: Synchronizing Point DB

- Simple application served very useful purpose
- Audit trail created by retained PI config files



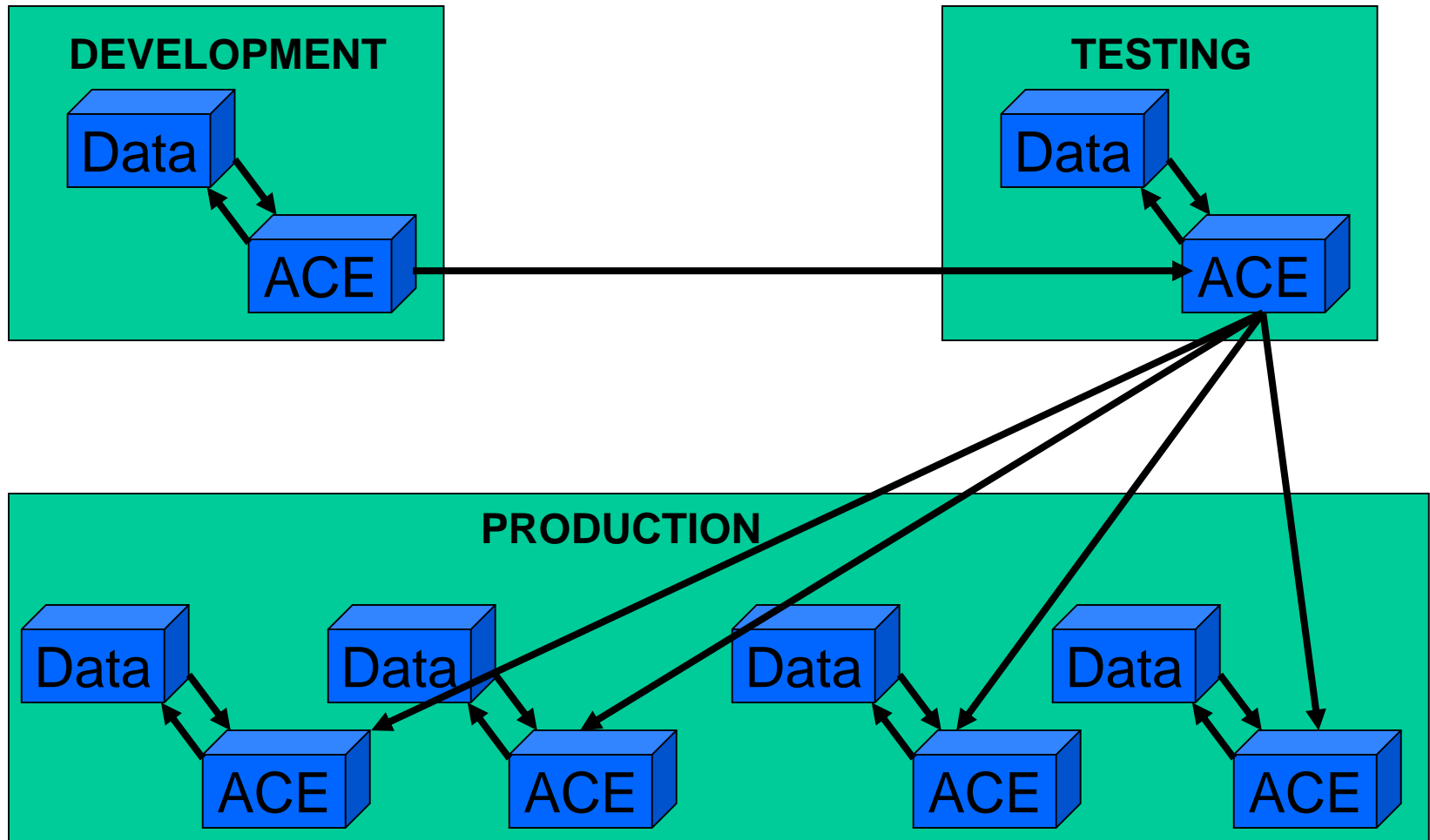


2nd Issue: Synchronizing Module DB

- The PI Module Database must also be identical on all Production PI servers
- Again, could have achieved this by manual updates to all machines but decided to automate the synchronization
- Created VB Program to auto-copy PI modules from PDS to other servers – used recursion!!!

Auto-Copy Modules Visual Basic Service

- Auto-copies the PI Modules from server to server



Module Database Recursion

Public Sub DupModules(DestMod As PISDK.PIModules, _
LocalMod As PISDK.PIModule)

'Check that all Sub-Modules exist, if not then create

'Check that all Aliases are identical, if not then create

'Check that all Properties are identical, if not then create

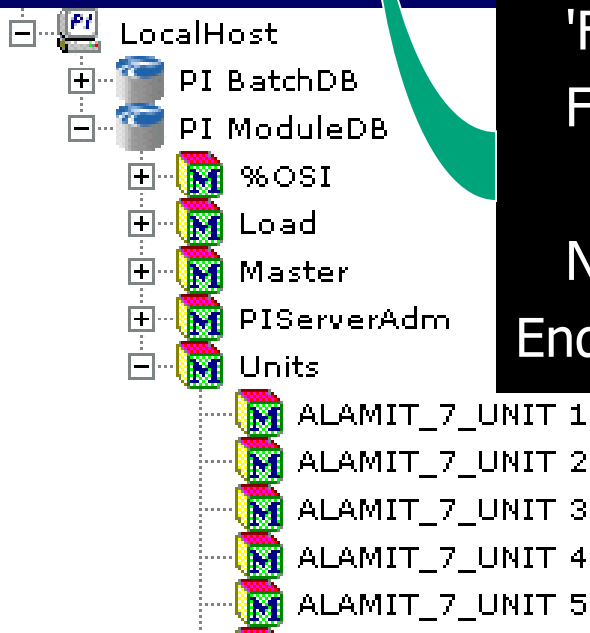
'For all Sub-Modules, call this Sub-routine...

For Each piModTmp In LocalMod.PIModules

 Call DupModules(DestModRec, piModTmp)


Next

End Sub






Key Takeaways: Synchronizing Module DB

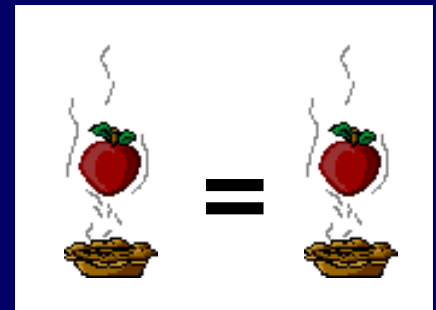
- Using recursion simplified the update code to just a few lines
 - The structure created on the Development PI server pair was easily replicated to all other server pairs (Testing and Production)
- 



3rd Issue:

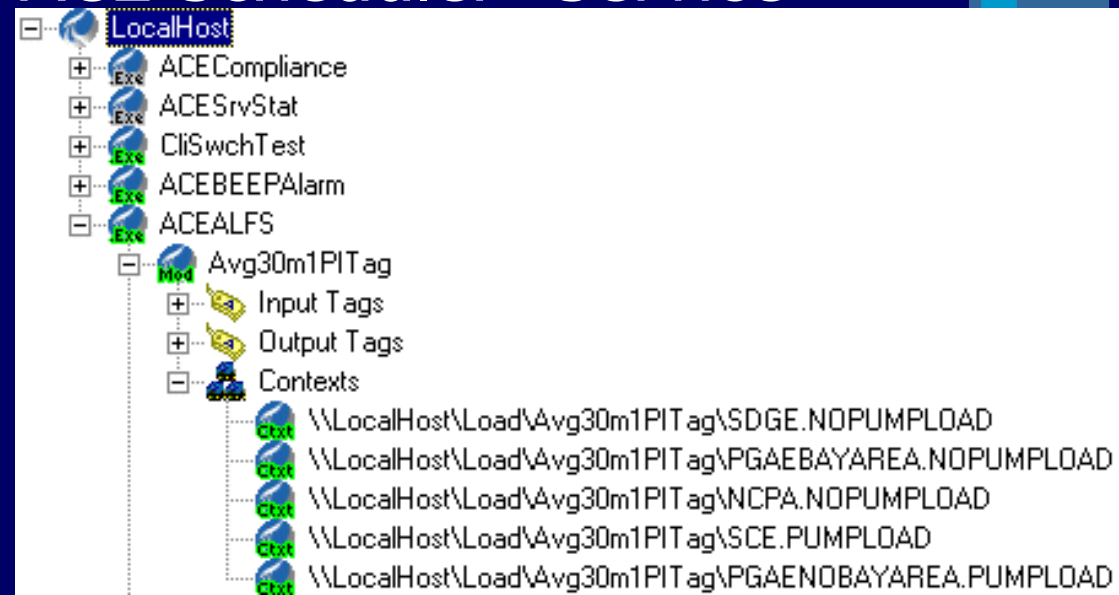
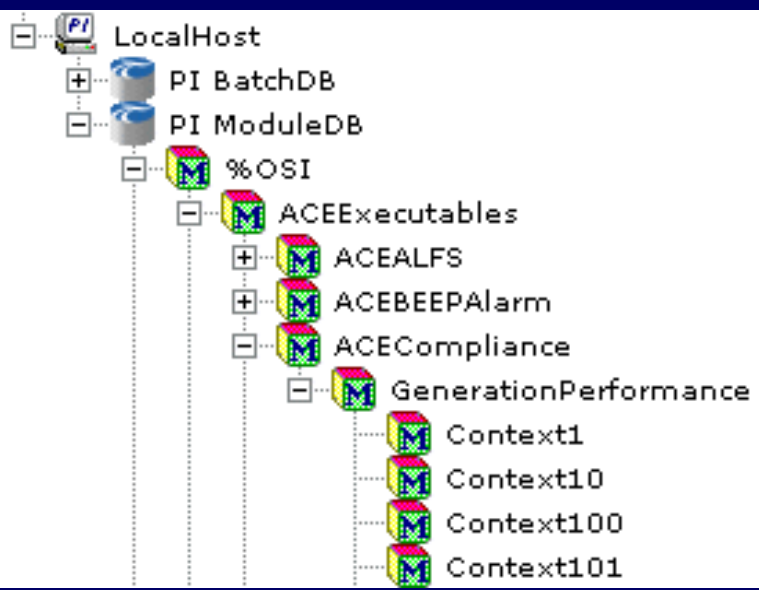
Distributing PI-ACE Executables

- Needed a way to easily distribute the PI-ACE executables from Development system
 - Added code to the PIAutoConfig program running on each server
- 



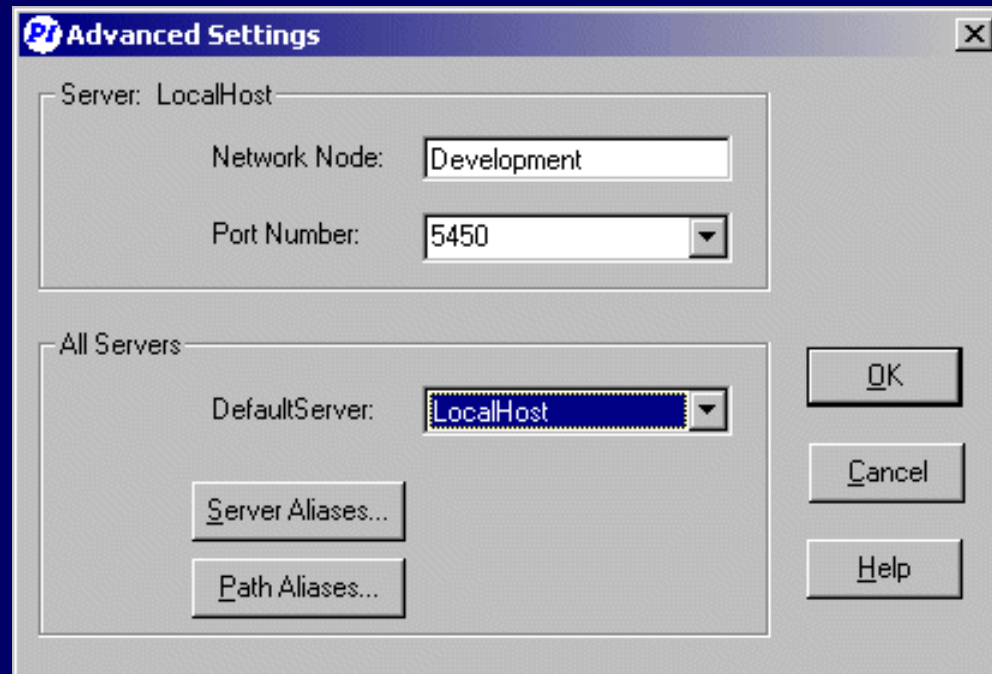
Steps to Distribute PI-ACE Executables

1. Stop the "PI-ACE Scheduler" Service
2. Make EXE Directory and copy EXE file
3. Initialize the EXE
4. Copy EXE MDB Structure (Under %OSI)
5. Turn Executable, Modules and Contexts ON
6. Start the "PI-ACE Scheduler" Service

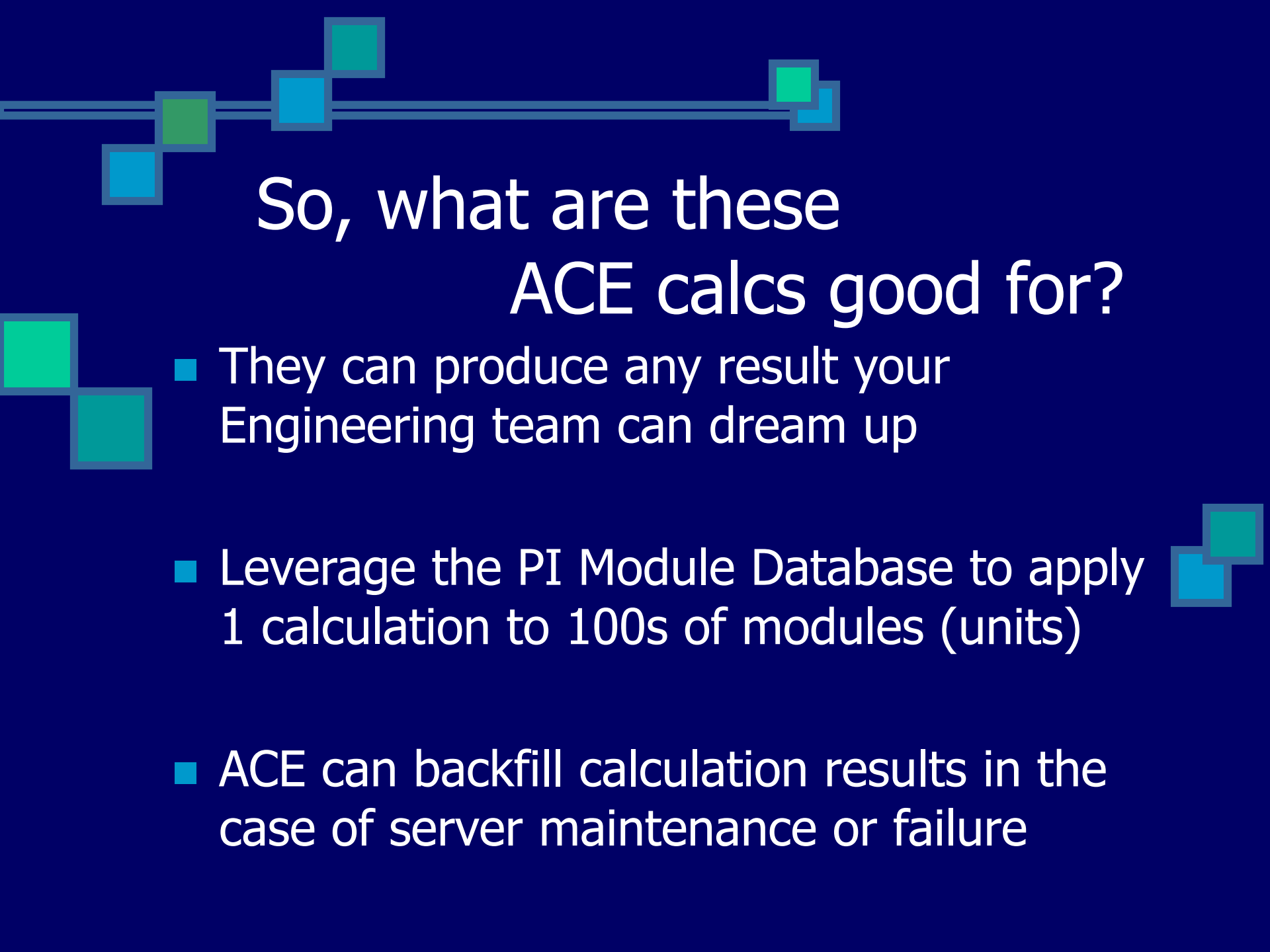


Key Takeaways: Distributing Executables

- Build all executables against the Development system referenced in the PI-SDK as a connection named "LocalHost"

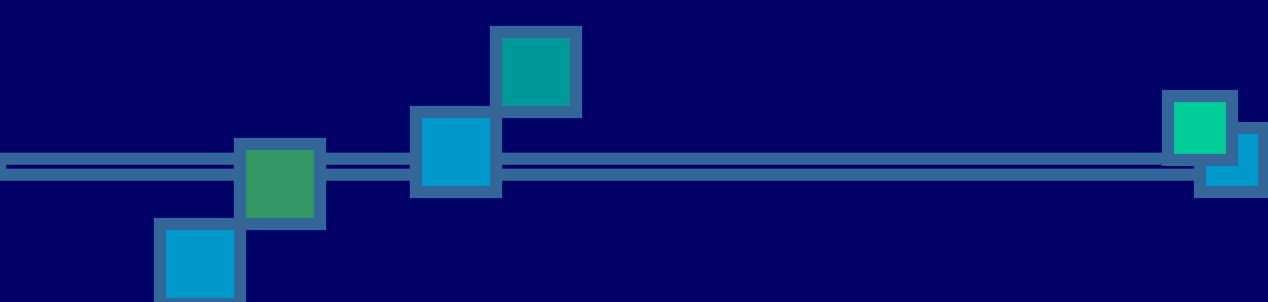


- Each PI-ACE server has a "LocalHost" defined as its own server




So, what are these ACE calcs good for?

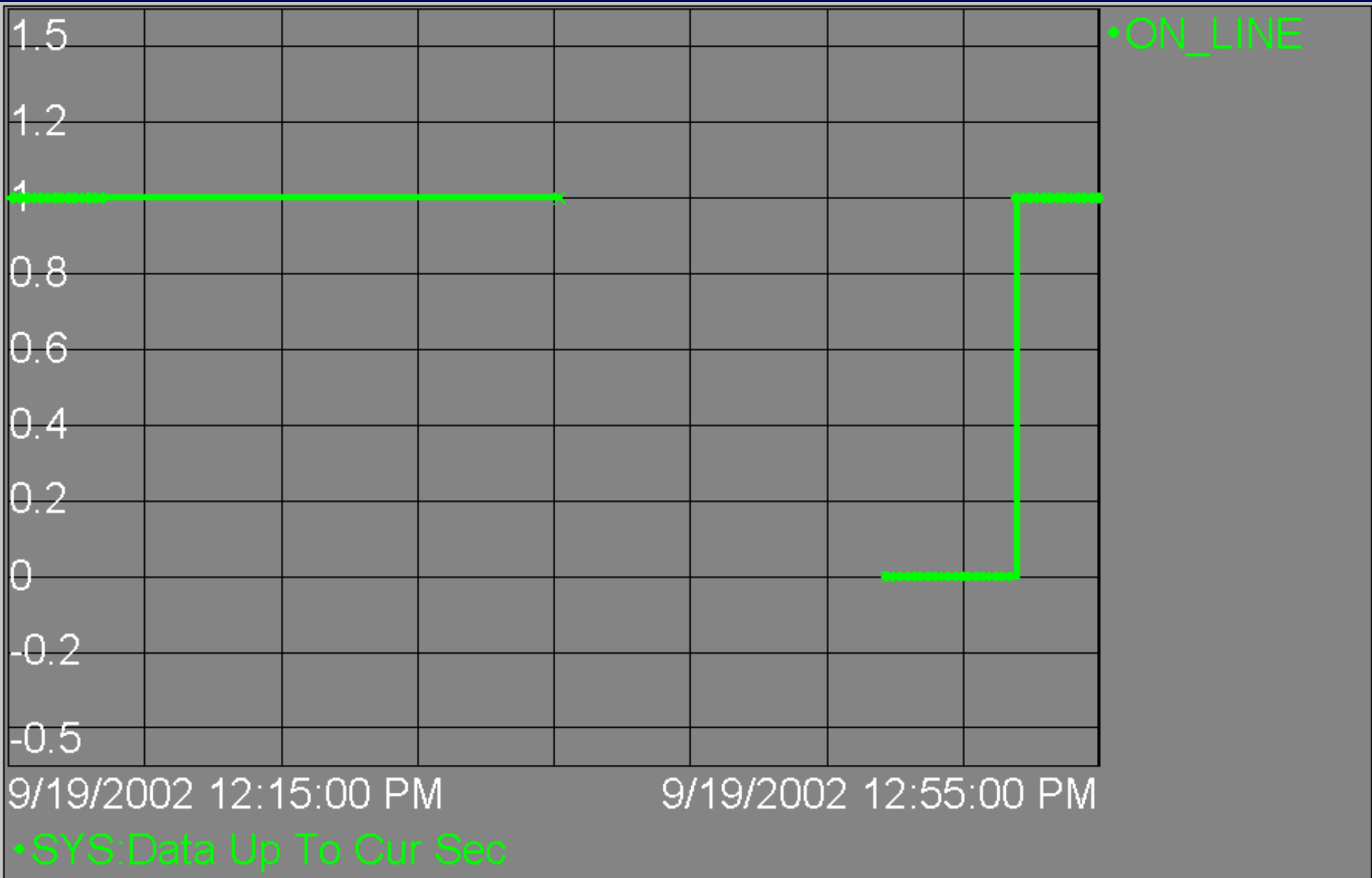
- They can produce any result your Engineering team can dream up
- Leverage the PI Module Database to apply 1 calculation to 100s of modules (units)
- ACE can backfill calculation results in the case of server maintenance or failure



Tip #1: Make a Master Data Tag to Drive your Calculations

- This tag indicates whether data is up-to-date or stale on the PI server
 - If data not up-to-date then skip calculation
 - When data is up-to-date then PI-ACE backfills calculation results
- 

Example Master Data Tag

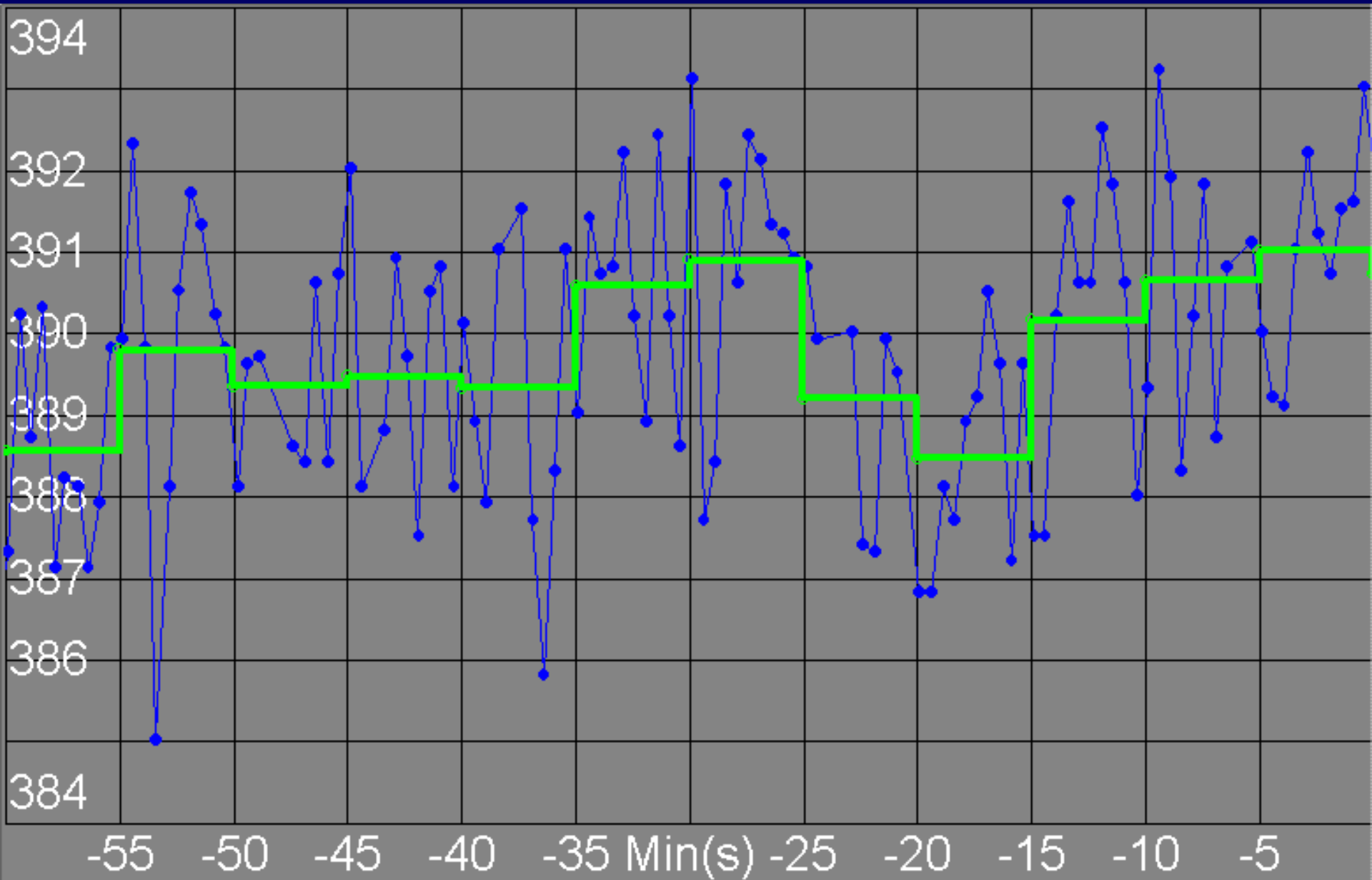


Typical Backfilling Calculation

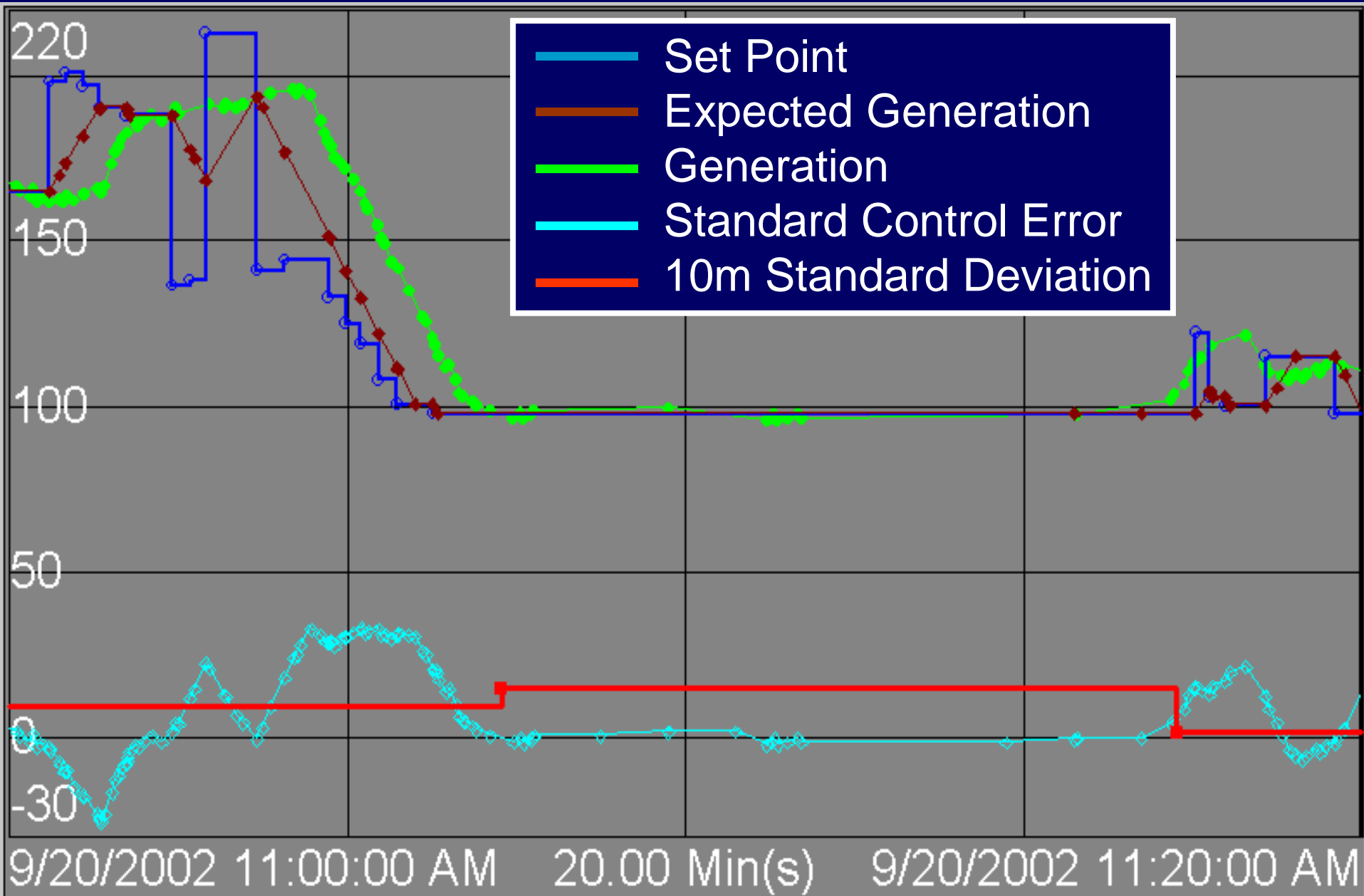
Tip #2: Steps to Backfill ACE Data

1. Check current data flowing into PI Database via "Master Data Tag"
2. Set calculation start time equal to last event written to output tag
3. Set calculation end time as either:
 - Current time
 - Start time + 3 hours
4. Gather all data for defined time period
5. Calculate results from data
6. Write results to output tags

ACE Results (simple average)



ACE Results (Performance Tracking)





Conclusion

- ISO Redundant PI Infrastructure
 - PI-ACE Calculations ISO style/technique
- 