Modernization – What is it and how do I go about it?

Darryl Freinkel
Assignment 400 Group, Inc.
dfreinkel@assignment400.com
www.assignment400.com
770.321.8562 ext . 111

Welcome

- Good afternoon ladies and gentlemen. Today we will be flying at 40,000 feet, covering a lot of ground getting to a modernized solution.
- As with many journeys, you will land at a hub, change flights and resume on your way.
- Fasten your seat belts. We are expecting some bumpy weather along the way.
- Follow the flight rules and you will land safely at your destination.

What is Modernization?

- Modernization means different things to different people.
- It could be hardware, software, services, …
- For IBM i, **Front End** modernization primarily means replacing conventional 5250 screens (green screens) with a Graphical User Interface (GUI).

What is Modernization?

- Other areas of modernization are:
  - Back end Modernization.
    - Convert from procedural coding to event processing.
    - Convert all RPG structured code to FREE FORMAT.
    - Replace physical and logical files with SQL tables, views and indexes.
      - Change RPG to free form SQL RPG
    - Break up code into
      - Business rules
      - UI rules

Why Replace what is not Broke?

- Why replace the green screen?
  - The green screen is killing our income. Each day that users use it, is another nail in the coffin.
  - Employees entering the work force do not know how to use the green screen. Kids out of school cannot relate to the green screen and so they rebel. They need a mouse, iPod and Blackberry.
  - The number of 20 and 30 year olds, is large and they have become the new managers.

UX is compelling and differentiates

"A great user experience is one of the most compelling and important characteristics of a modern business application."

Forrester, March 2008

"Embrace legacy.
Let it flow to create new value."

Partha Iyengar, Gartner, Is Application Development dead? Aug 2008

"User Experience (UX) will increasingly become a differentiator for organizations delivering service."

Michael Barnes Gartner, Facing the Challenge of the SOA enterprise, July 2008

A recent study by Forrester Consulting, found that:

"legacy modernization would produce a five year return on investment (ROI) of 331 percent, with payback achieved in less than four months."

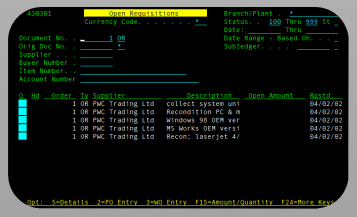
"Hearty endorsement of "update and extend" application-development strategies as opposed to the "rip and replace" idea for most organizations"

What is Modernization?

- Why replace the green screen?
  - So, who is to blame?
    - Well, we the development community and the companies that use the AS/400.
    - IBM made it known more than 10 years ago, that it would be replacing 5250 with client server technology. IBM has already brought out new replacements, either directly or indirectly.
  - If we do not replace 5250, we have nowhere to go.

What is Modernization?

your applications



presentation

application

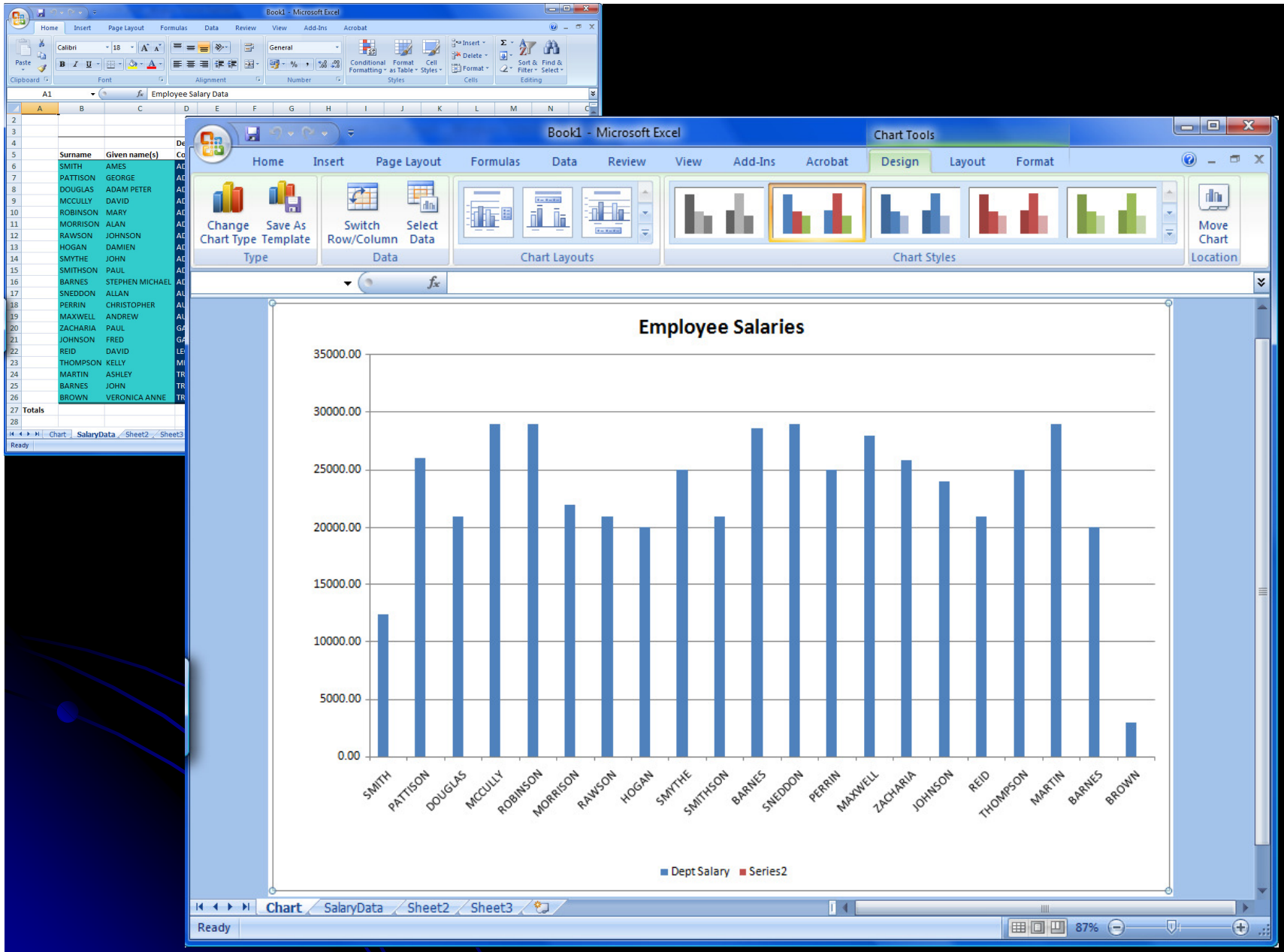
database

thin

mobile

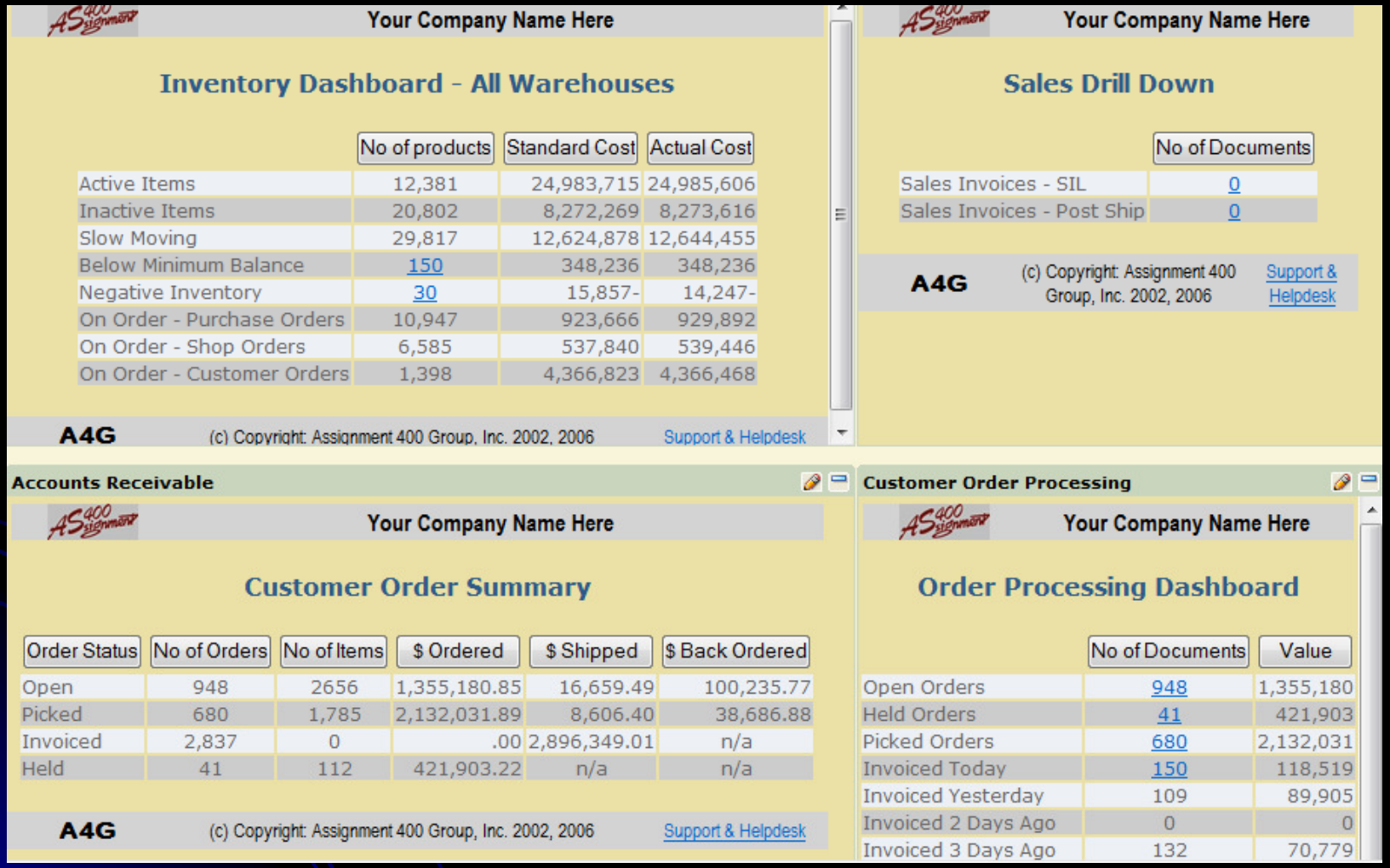
emulation

voice

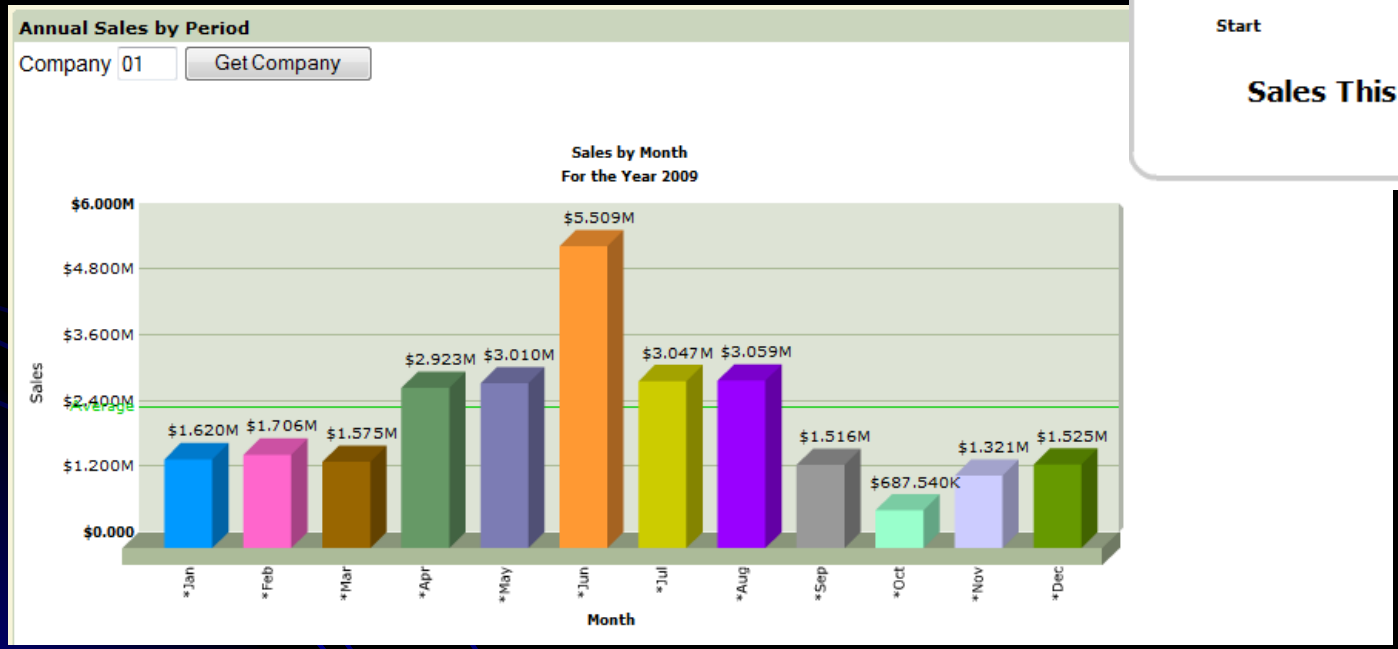


Default iPhone

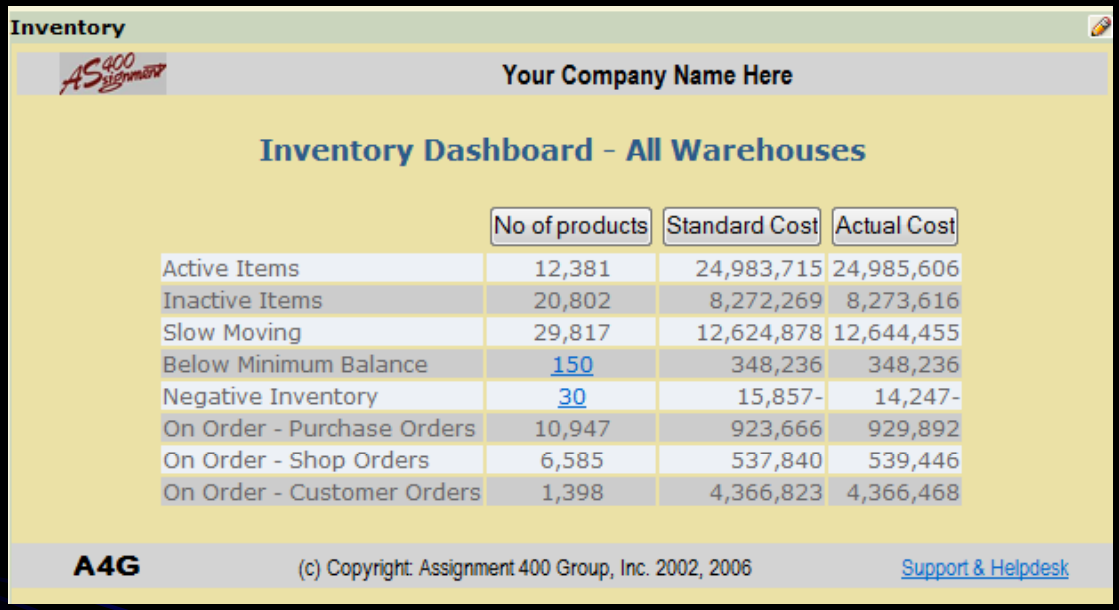
Portals

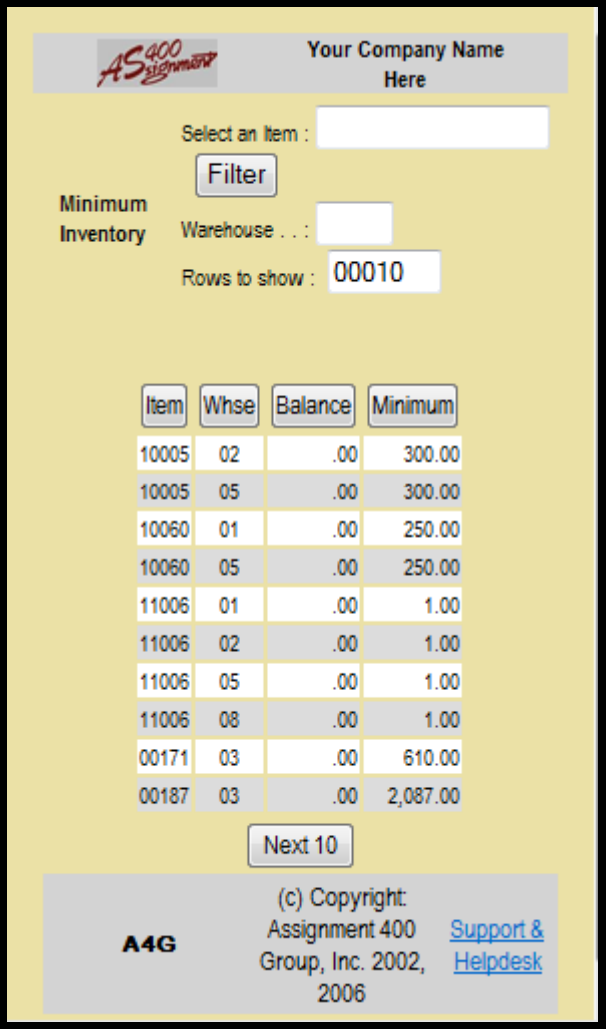


Portals



Portals





How do we Modernize?

- There are several ways to modernize.
  - Front End Modernization - Screen scrape
    - LookSoftware, Seagull, BCD, PHP, WebFace, HATS and others
    - Quick and easy
    - 7 to 60 days to be up and running.
  - Back-End Modernization
    - Re-develop your applications using your business rules
      - BCD WebSmart, PHP, EGL, LookSoftware, Lansa, Java, C#, Microsoft Visual Studio.
      - Slower and more intensive.
      - 3 to 12 months to implement.

How do we Modernize?

- 2 or 3 Step process.
  - Phase 1 - Start with a screen scraping technology.
  - Phase 2 - Deploy client server techniques using
    - IBM i as your work horse
    - Anything GUI tool as your User Interface (UI)
      - The past 10 years has shown this technology has a life cycle of approximately 3 years before being replaced.
      - Employ new blood for the UI. The kids of today love this area and they are good at it.
    - Re-develop your code from your existing code to be event driven, triggered by any call to the back office system.

How do we Modernize?

- 2 or 3 Step process.
  - Phase 2 - Deploy client server techniques.
    - If you have a **packaged solution**, you may not be able to deploy phase 2 directly.
    - Instead, find a tool like LookSoftware that will give you the same benefits.
      - Web services is the most important way to begin communication with others. Tools like LookSoftware provide web services, but instead of calling traditional API's, under the covers you call 1 or more programs in the package.

How do we Modernize?

- Phase 1 - Process your 5250 data stream further (Screen Scrape).
  - To do this you need a LookSoftware, HATS…
  - Think of the 5250 data stream as another source of information and not a screen. Modern tools like Look, allow you to do almost anything that a windows system can do. Users and managers won't believe it's the old AS/400 serving up the data.
  - 5250 Processors will bring you into the 21st century very, very quickly.
  - There is plenty of scope processing the 5250 data stream.

How do we Modernize?

- Phase 1 - Process your 5250 data stream further (Front End Modernization).
  - 5250 data stream is fast. You can have applications up and running in days.
  - Testing is reduced to a minimum. Your green screens are already tested. Yes, you do need to test.
  - The RPG guys will continue to develop and test before handing over to the GUI developers.
  - Most RPG folk can develop the GUI interface.
  - Simply build the rules and filters into the GUI and most of the work is done.
  - Use existing skills.
  - Learn a product like Look.

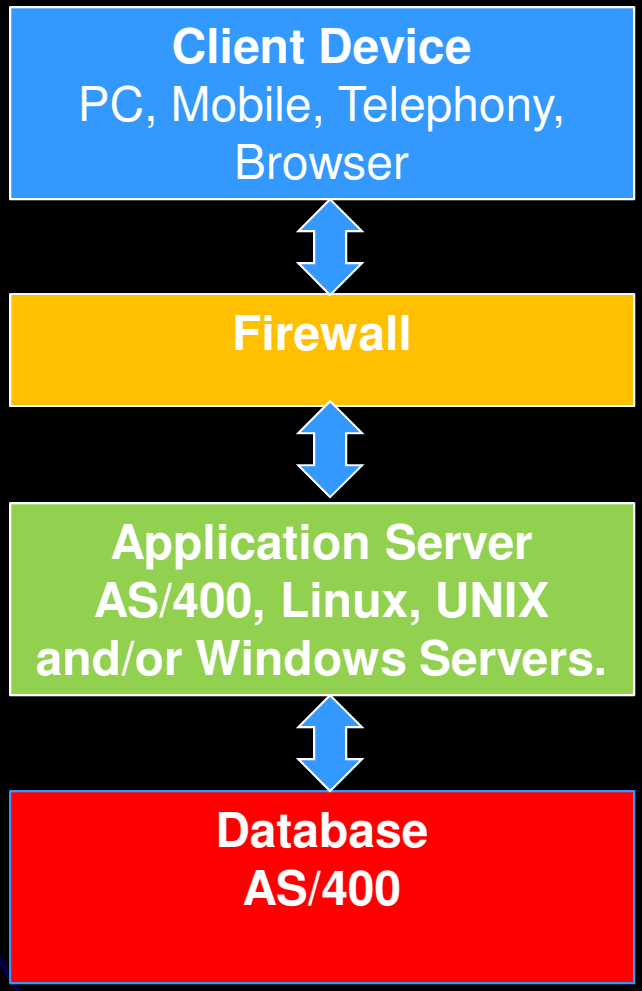
How do we Modernize?

- Phase 2 - Modify your applications (Back End Modernization).
  - To re-develop, you need to consider a few things.
    - How do you use your existing programs?
    - You need to understand client server processing.
      - The PC is the client.
      - The IBM i is the server.
    - Task 1 – Pick frequently used programs to begin with.
    - Task 2 – Take each program and decide how to separate the UI from the business rules.
    - The decisions for tasks 1 and 2 should be independent of the GUI (tool) to be used.

Understanding Client Server

- 2 Techniques
  - Procedural
    - We are all familiar with procedural. RPG programs are all procedural. You control the process from beginning to end.
    - The program has to get to a **return** or ***INLR** before releasing resources.
    - Memory is allocated. Swapping included.
    - Locks are held indefinitely.
    - Procedural is very restrictive.

Client Server Building Blocks



Understanding Client Server

- 2 Techniques
  - Event Driven
    - This is the way to go. There is no getting around it.
    - It is how you service **hundreds and thousands of users**.
    - Hot links will take you to places you never thought of. You may never return to your starting point. What will this do to your procedural programs?
    - All new applications should be considered programs for the Internet. Think about the internet programs you use.
    - This is easy work for RPG'ers.

Understanding Client Server

- Persistence and non persistent
  - Green Screen is **persistent**.
    - It executes in 1 job from beginning to end.
    - System assigns 1 job number.
    - Job attributes remain constant.
  - Intelligent clients and HTML are generally **non persistent**.
    - The first call may be assigned to run in job A.
    - Subsequent calls can be assigned to run in job B, C, D or A.
    - You need to set up your environment for each call.
    - Calls are API's. Each call is a self contained unit.
    - IBM i is very good at this.

Understanding Client Server

- In Event Driven programs:
  - Server Side
    - Sign on to the system for every transaction. (10,000 logons per hour is normal)
    - Set your library list
    - Call your program
    - Update all files
    - Save pointers and values in a session file.
    - Return data to the GUI application. The GUI will retain pointers and the session ID.
    - Program ends
    - Note: the server process will retain data from the last call.

Understanding Client Server

- In Event Driven programs:
  - Client Side (Intelligent Clients)
    - Sign on to the system once. The application will store your credentials.
    - Will call the server for data it needs.
    - Client could call multiple systems.
    - Will process the data and render the result onto the screen.
    - After users have entered data, client will determine the API to use and call it. Client activates the RPG program, not the other way round.
    - Server side program is called, processes, returns the data and ends.

Understanding Client Server

- In Event Driven programs:
  - Client Side (HTML)
    - Sign on to the system once. The browser or application will store your credentials.
    - Will call the server for data it needs.
    - Client calls one system only. Let the server call the other systems.
    - Will process the data and render the result onto the screen.
    - After users have entered data, client will determine the API to use and call it. Client activates the RPG program, not the other way round.
    - Server side program is called, processes, returns the data and ends.

Securing your Data

- In Event Driven programs:
  - Securing your data is very important.
    - Secure from outsiders
    - Secure from insiders
  - How do you do security.
    - Will need another session to review some techniques.

Operating System Requirements

- Most screen scrapes will work back as far as OS/400 version 4 and probably version 3. The stream remained consistent.
- For other client server tools, you will need OS/400 version 5.
  - Will depend on the tools used?

Example Tools

- Java
- LookSoftware
- Lansa
- IBM's EGL
- Seagull
- Microsoft's Visual Studio Express
- BCD's Websmart
- MRC
- ASNA
- CGIDEV

Skills

- RPG – You already know a lot
- GUI
  - Lansa, LookSoftware, EGL …
    - Easy to learn.
    - Developed by RPG folk.
  - PHP
    - Runs natively on the IBM I from V5R3.
    - Relatively easy to use.
    - Employ PHP specialists.
    - BCD's WebSmart PHP good for RPG folk.
  - Microsoft Visual Studio
    - Has several options and flavors.
    - .NET is different.
    - Employ Visual Studio specialists.
  - Java
    - Employ Java specialists.

Examples and Demonstration

- MECO's successful project.

Questions