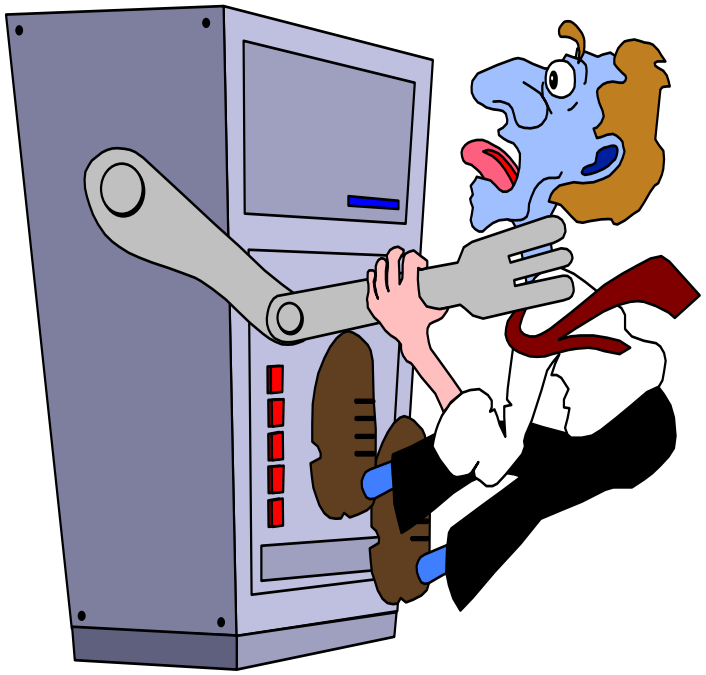
Costs and Barriers to Government Inter-Agency and Industry Data Exchange

Richard Eastman
President
The Eastman Group, Inc.

Hardware Centric 1960's



- LOTS OF DATA
- SOLUTION SPECIFIC
- DATA RETRIEVAL

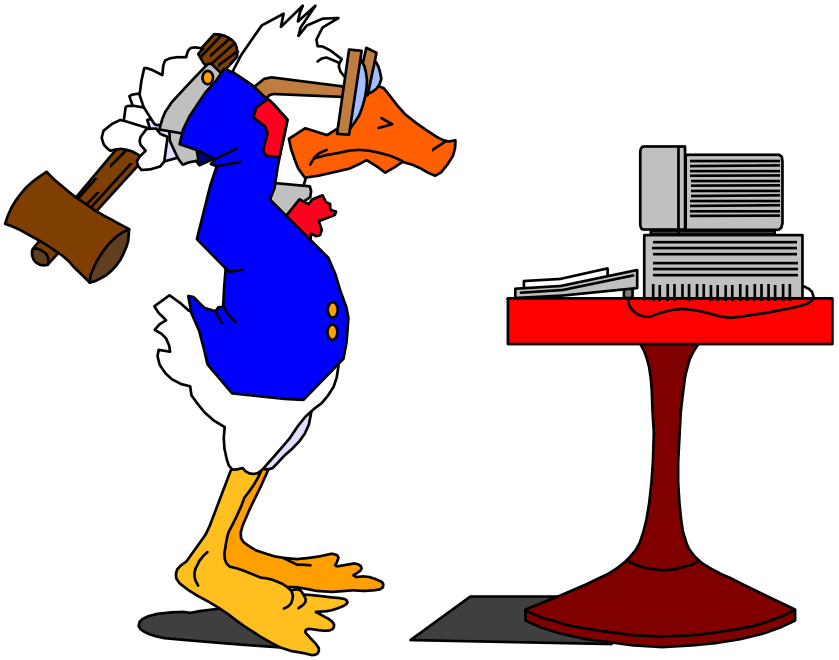
DRIVEN BY HARDWARE DESIGN ISSUES
MAINFRAME ARCHITECTURES

Software Centric 1970's

- DATA PROCESSING
- STRUCTURED
- "CONTAINERIZED"

PRIMARY/KEY BUSINESS SOLUTIONS
MINICOMPUTER ARCHITECTURES

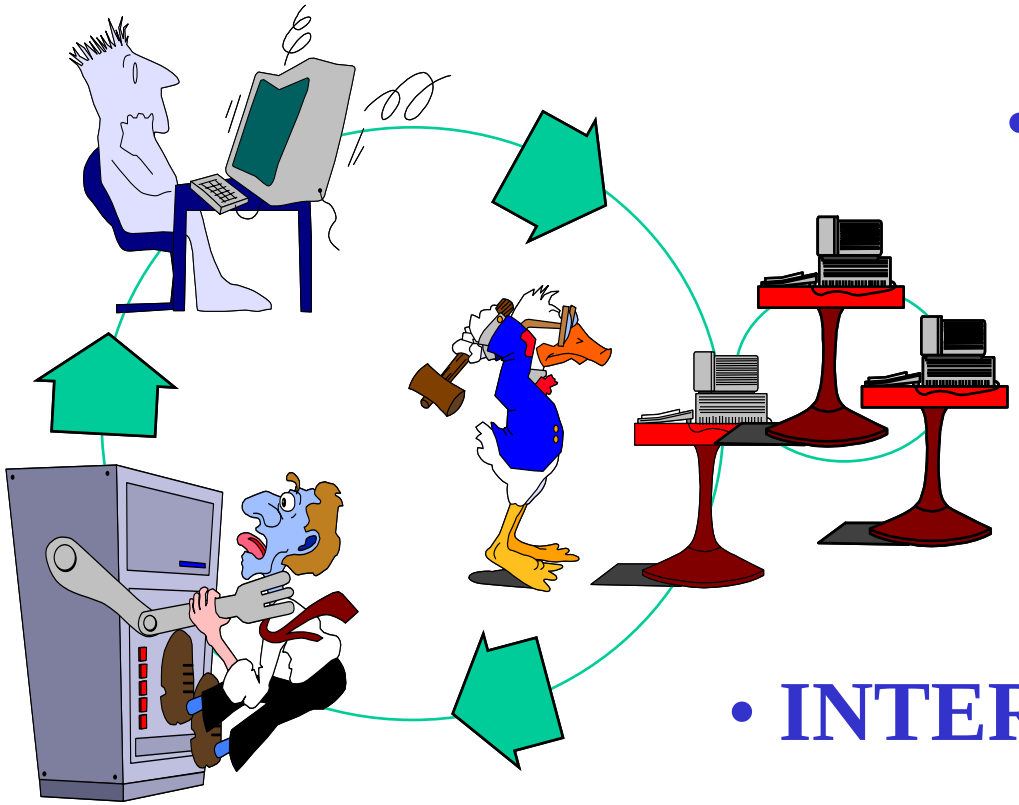
Task Centric 1980's



- TASK SPECIFIC
- ADAPTABLE
- PERSONAL USE

INDIVIDUALIZED BUSINESS TOOLS
MICROCOMPUTER (PC) ARCHITECTURE

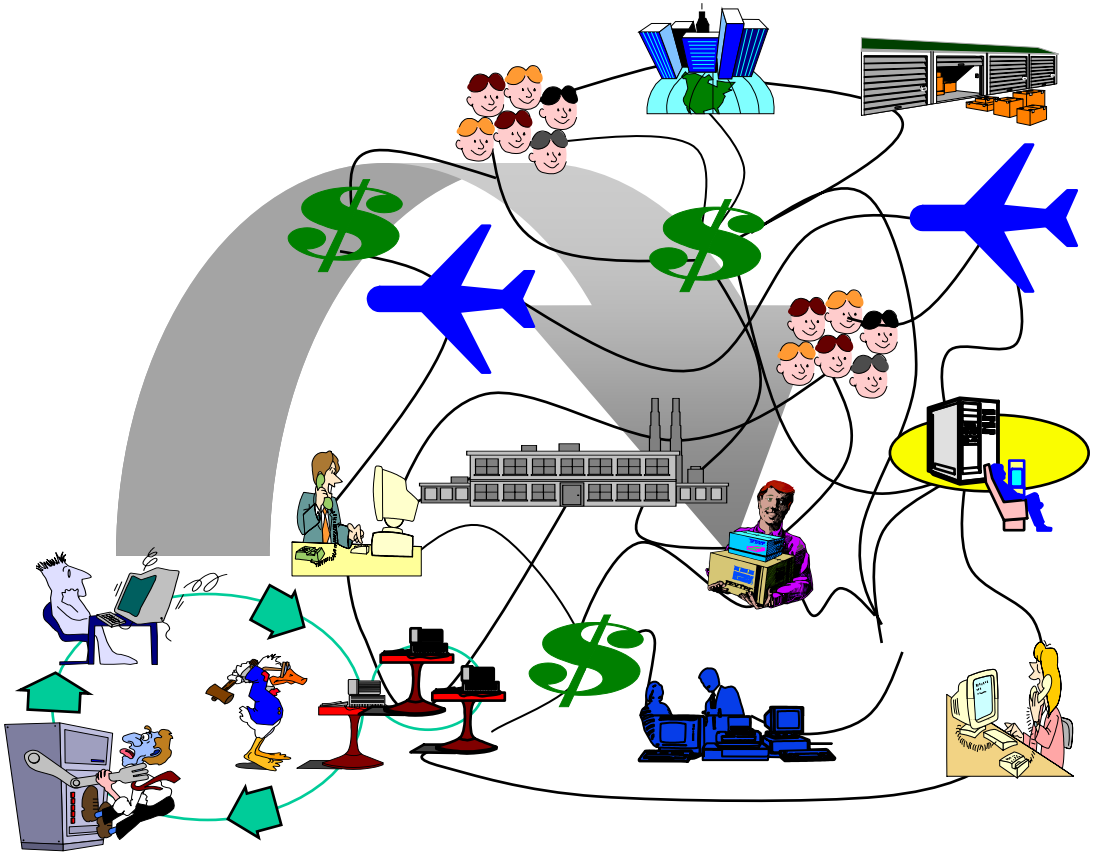
Communications Centric 1990's



- TRANSPARENT
- LAN/WAN's
- INTERNET/INTRANET

MAKING THE DIFFERENT PLATFORMS "TALK" TO EACH OTHER

Interactive Centric 2000's

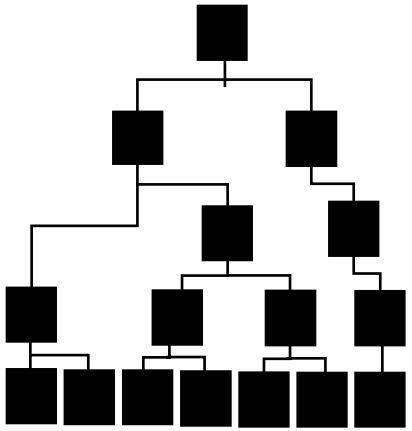


- Bi-Directional
- Broadband
- Multi-Media

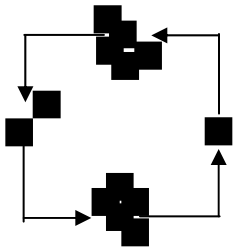
Integrating Information with Knowledge
…Transaction Processing Transparent

Transitioning Eras…

Industrial Age

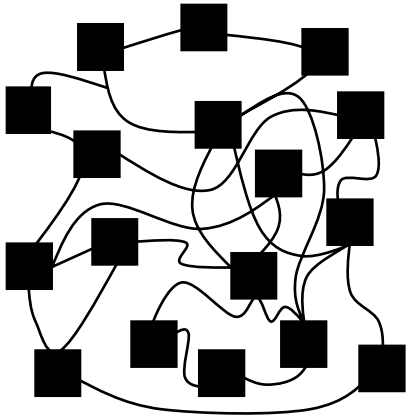


Information flows from hub-centers through controlled hubs or hosted gateways to users …



- ***Supply-Driven Information***

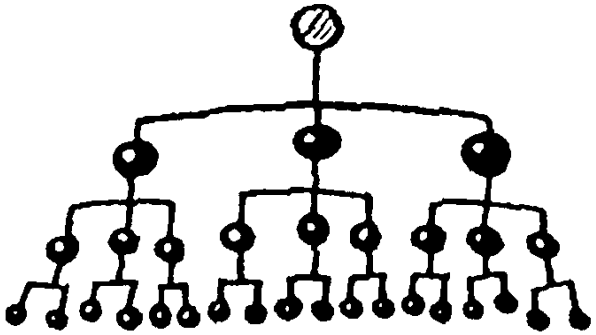
Information Age



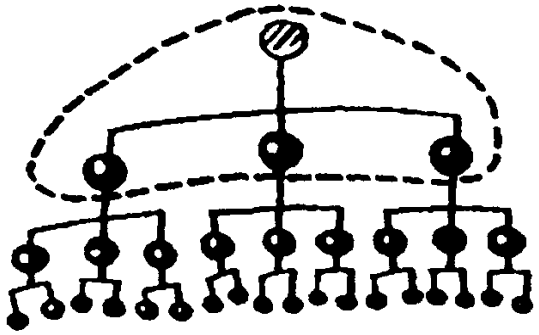
Information flows digitally and is shared by everyone … information becomes interactive…

- ***Demand-Driven Information***

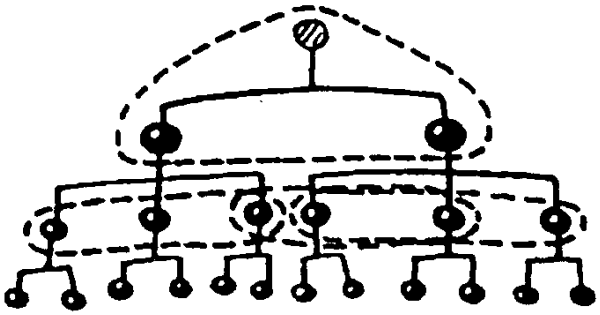
From "Imagin•i•zation : Gareth Morgan©1997



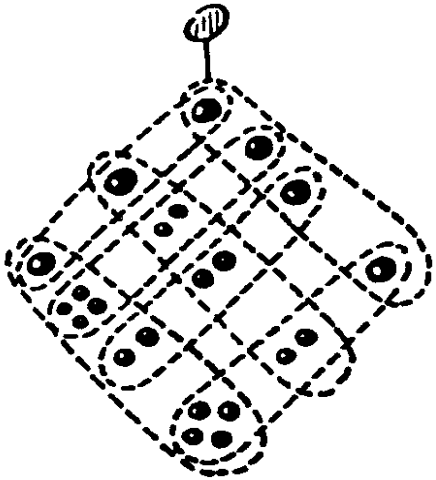
Model 1: The Rigid Bureaucracy



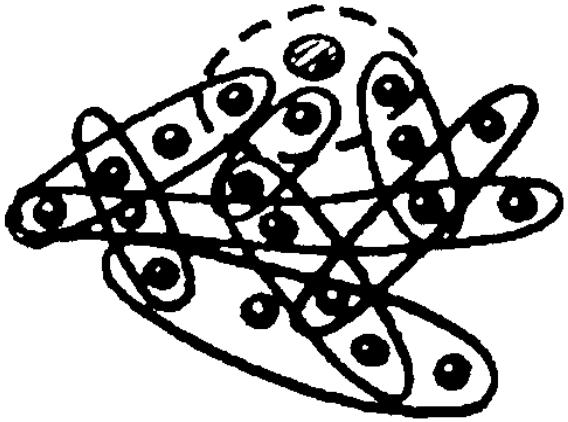
Model 2: The Bureaucracy With a Senior "Management Team"



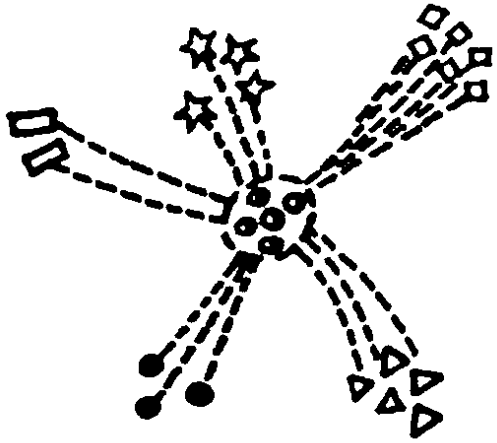
Model 3: The Bureaucracy With Project Teams and Task Forces



Model 4: The Matrix Organization



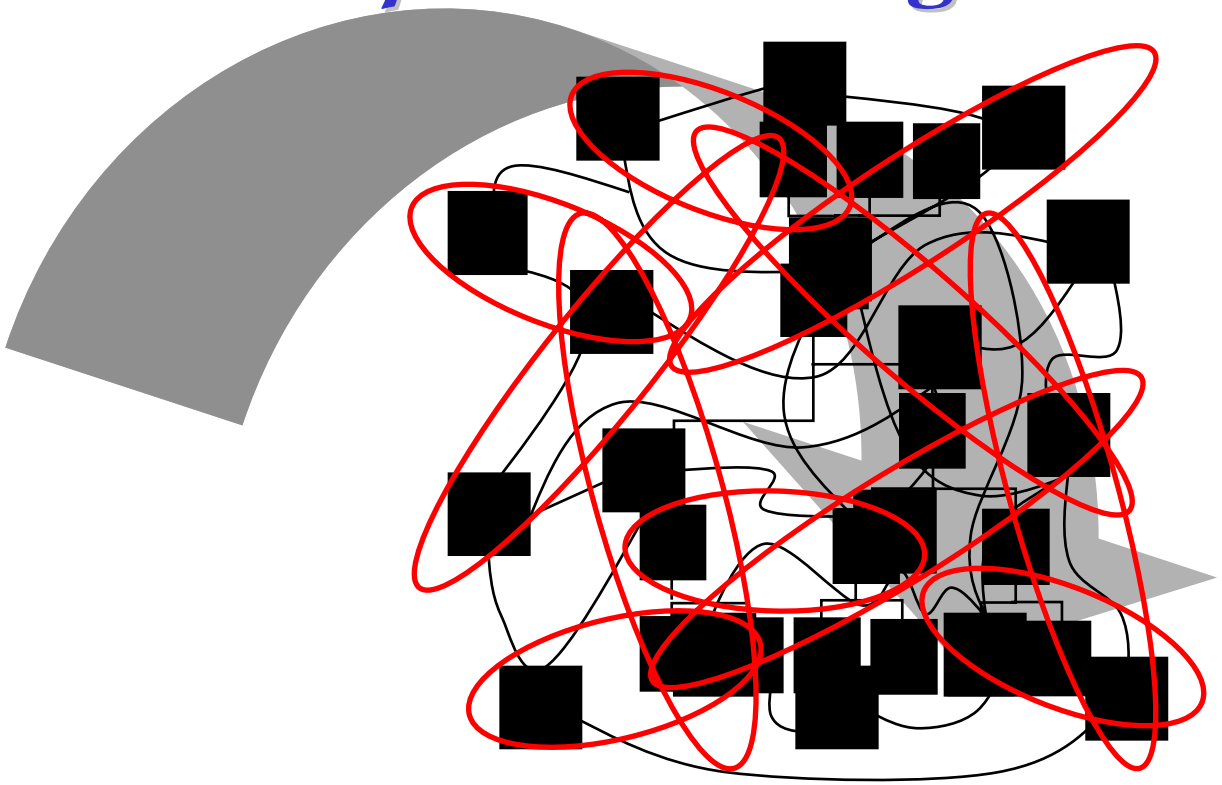
Model 5: The Project Organization



Model 6: The Loosely-Coupled Organic Network

Structures are changing the way people relate to one another

In the Information Age…



Information flows digitally and is shared by everyone… Transaction Processing becomes transparent…

Humans Drive Technology
Buyers Drive Suppliers
Operations Integrate with Information
Context More Important than Content

Effecting the World … Economics • Politics • Knowledge

Some Comparisons…

Programmer Salaries

Salaries for TPF-type or code-intensive programmers are 10-15% higher than highest comparable skills in newer programming languages[1] …

Since most are generally "dead" languages, the learning curve is generally more than double to acquire comparable programming proficiency

Range…

Programmer Analyst
$53,250 - $80,500
+10% for C++
+5-15% for XML
+5% for VB

Team Leader
$78,500 - $108,500
+ skill-type

Assuming long term retention, normal attrition, and burden … effective salary costs are more than 1.5 times competitive industry costs!

1. Source: Robert Half Technology

Some Comparisons…

Coding Requirements… Count Words; Assembler 1

```
;************************************************************************
; Program to count the number of words in a file.  Words are
; delimited by whitespace
;************************************************************************
        DOSSEG                  ;select standard segment-ordering
        .MODEL  SMALL           ;code and data each fit in 64K
        .STACK  200h            ;512-byte stack
        .DATA
Count           DW    0         ;used to count words
InWhitespace    DB    ?         ;set to 1 when the last
                                ; character read was whitespace
TempChar        DB    ?         ;temporary storage used by
                                ; GetNextCharacter
Result          DB    'Word count: ', 5 DUP (?)
                                ;string printed to report the count
CountInsertEnd  LABEL BYTE      ;used to find the end of the area the
                                ; count value string is stored in DB
                                ; 0dh,0ah,'$' OS fn #9 expects strings to
                                ; end with a dollar sign
        .CODE
ProgramStart:
        mov     ax,@data
        mov     ds,ax           ;point DS to the .DATA segment
        mov     [InWhitespace],1        ;assume we're in whitespace, since
                                ; the first non-whitespace we'll find
                                ; will mark the start of a word
CountLoop:
        call    GetNextCharacter        ;get next character to check
        jz      CountDone       ;...if any
        call    IsCharacterWhitespace   ;is it whitespace?
        jz      IsWhitespace    ;yes
        cmp     [InWhitespace],0        ;character is not
                                ; whitespace-are we currently in
                                ; whitespace?
        jz      CountLoop       ;we're not in whitespace, and the
                                ; character isn't whitespace, so
                                ; we're done with this character
        inc     [Count]         ;we are in whitespace, and the
                                ; character is not whitespace, so
                                ; we just found the start of a new word
        mov     [InWhitespace],0        ;mark that we're no longer in
whitespace
        jmp     CountLoop       ;do the next character
IsWhitespace:
        mov     [InWhitespace],1        ;mark that we're in whitespace
        jmp     CountLoop       ;do the next character
;
; We're done counting--report the results.
;
CountDone:
        mov     ax,[Count]      ;number to convert to a string
        mov     bx,OFFSET CountInsertEnd-1 ;point to the end
                                ;of the string to
                                ; put the number in
        mov     cx,5            ;number of digits to convert
        call    ConvertNumberToString   ;make the number a string
        mov     bx,OFFSET Result        ;point to result string
        call    PrintString     ;print the count
        mov     ah,4ch          ;DOS terminate program fn #
        int     21h             ;end the program
;
```

Some Comparisons…

Coding Requirements…

Count Words; Assembler 2

```
; Subroutine to get the next character from the standard input.
;
; Input: None
;
; Output:
;       AL = character, if one was available
;       Z flag = 0 (NZ) if character available,
;              = 1 (Z) if end of file reached
;
; Registers destroyed: AH, BX, CX, DX
;
GetNextCharacter        PROC
        mov     ah,3fh                  ;DOS read from file fn #
        mov     bx,0                    ;standard input handle
        mov     cx,1                    ;read one character
        mov     dx,OFFSET TempChar      ;put the character in TempChar
        int     21h                     ;get the next character
        jc      NoCharacterRead         ;if DOS reports an error, then treat
                                        ; it as the end of the file
        cmp     [TempChar],1ah          ;was it Control-Z?
                                        ; (marks end of some files)
        jne     NotControlZ             ;no
NoCharacterRead:
        sub     ax,ax                   ;mark no character read
NotControlZ:
        and     ax,ax                   ;set Z flag to reflect whether a
                                        ; character was read (NZ), or the
                                        ; end of the file was reached (Z).
                                        ; Note that DOS fn #3fh sets AX to
                                        ; the number of characters read
        mov     al,[TempChar]           ;return the character read
        ret                             ;done
GetNextCharacter        ENDP
;
; Subroutine to report whether a given character is whitespace.
;
; Input:
;       AL = character to check
;
; Output:
;       Z flag = 0 (NZ) if character is not whitespace
;              = 1 (Z) if character is whitespace
;
; Registers destroyed: none
;
IsCharacterWhitespace   PROC
        cmp     al,' '                  ;is it a space?
        jz      EndIsCharacterWhitespace   ;if so, it's whitespace
        cmp     al,09h                  ;is it a tab?
        jz      EndIsCharacterWhitespace   ;if so, it's whitespace
        cmp     al,0dh                  ;is it a carriage return?
        jz      EndIsCharacterWhitespace   ;if so, it's whitespace
        cmp     al,0ah                  ;is it a linefeed? If so,
                                        ; it's whitespace, so return Z;
                                        ; if not, it's not whitespace,
                                        ; so return NZ as set by cmp
EndIsCharacterWhitespace:
        ret
IsCharacterWhitespace   ENDP
;
; Subroutine to convert a binary number to a text string.
;
```

Some Comparisons…

Coding Requirements…

Count Words; Visual Basic

```
'*****************************************************************************
'WORDS(<ExpC1>[,<ExpC2>])
'Returns:Numeric
'Description:Returns the number of words in <ExpC> delimited by
any character in optional
'<ExpC2>.The default for <ExpC2> is a Whitespace.
'Example: WORDS("This is a test of FoxTools") (returns: 6)
'*****************************************************************************
Public Function WORDS(ByVal cExpression As String, Optional
cDelimiter = " ") As Integer

   Dim lcCharacter As String
   Dim lcPreviousCharacter As String
   Dim lnCounter As Integer
   Dim lnCount As Integer
   Dim lcStringBuffer As String
    lnCounter = 0
   lnCount = 0
   'Checking the essential parameter
   If Len(cExpression) = 0 Then 'Expression is Empty
     WORDS = lnCounter
     Exit Function
   End If
   'Appending a delimiter if reqd inorder to get the right word number
   If InStr(cDelimiter, Right(cExpression, 1)) = 0 Then
     cExpression = cExpression & Left(cDelimiter, 1)
   End If
   For lnCount = 1 To Len(cExpression)
     lcCharacter = Mid(cExpression, lnCount, 1)
     If InStr(cDelimiter, lcCharacter) > 0 Then
       If InStr(cDelimiter, lcPreviousCharacter) = 0 Then
         lnCounter = lnCounter + 1
         lcStringBuffer = ""
       End If
     Else
       lcStringBuffer = lcStringBuffer & lcCharacter
     End If
     lcPreviousCharacter = lcCharacter
    Next
    WORDS = lnCounter
End Function
```

Some Comparisons…

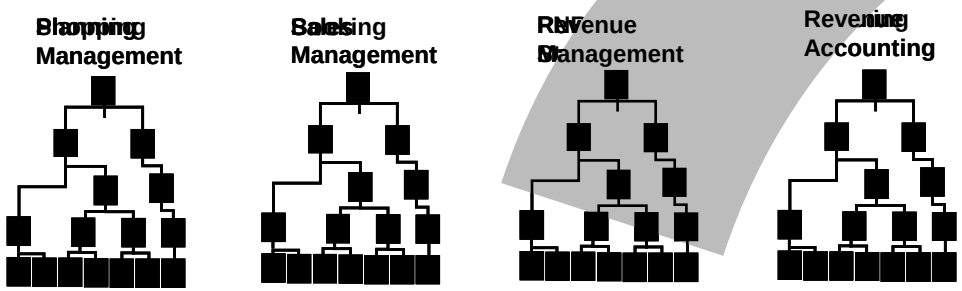
Coding Requirements… *Count Words; SQL*

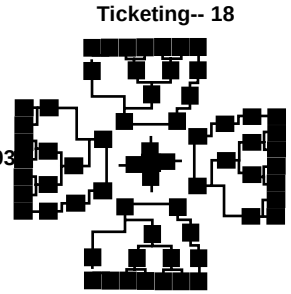
WORDS(lcString, " ")

Some Comparisons…

Comparative Architectures

Silo Structures[1]





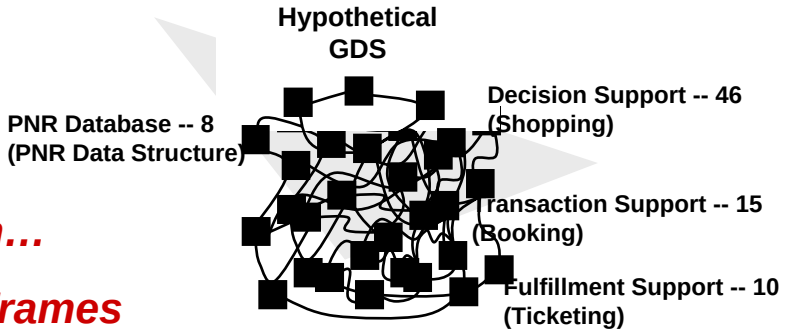
Plus …
Development and Project Management -- 64
Front-end & Gateways -- 124

GDS Hierarchal System…

14 Clustered IBM Mainframes
6.2 Million lines of TPF Code
4 Tanden Himalayas for pre-processing structured message formats
437 Programmers/Analysts

Comparable Nodal-Network[2]…

5-7 Clustered Sun Servers
40%-50% Licensed Code
500 Intel Pentium Processor Nodes running Linux
101 Programmers/Analysts



Plus…
Development & Project Management -- 15
External Adaptors – 7
(Gateways)

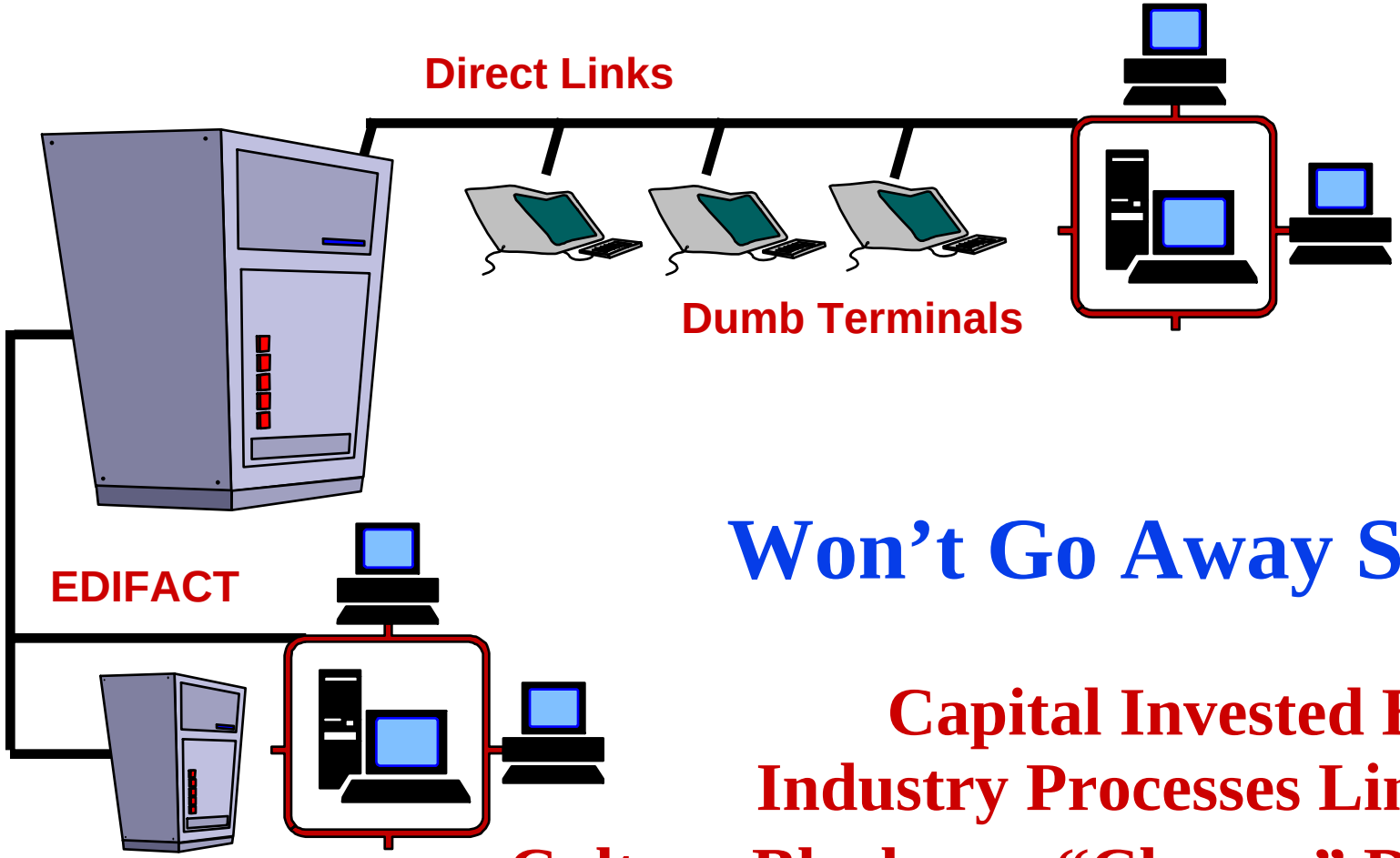
A. Hardware replacement costs estimated at under 25% of current IBM Environment.

B. Using similar salary costs, ongoing support is also under 25% of current base.

1. Source … Economics of Airline Automation, Handbook of Airline Economics, McGraw-Hill: Editor Daryll Jenkins : Author Richard Eastman

2. Source … 2001 Proprietary and Confidential Study by TEG and Integrasoft™

Existing Systems…

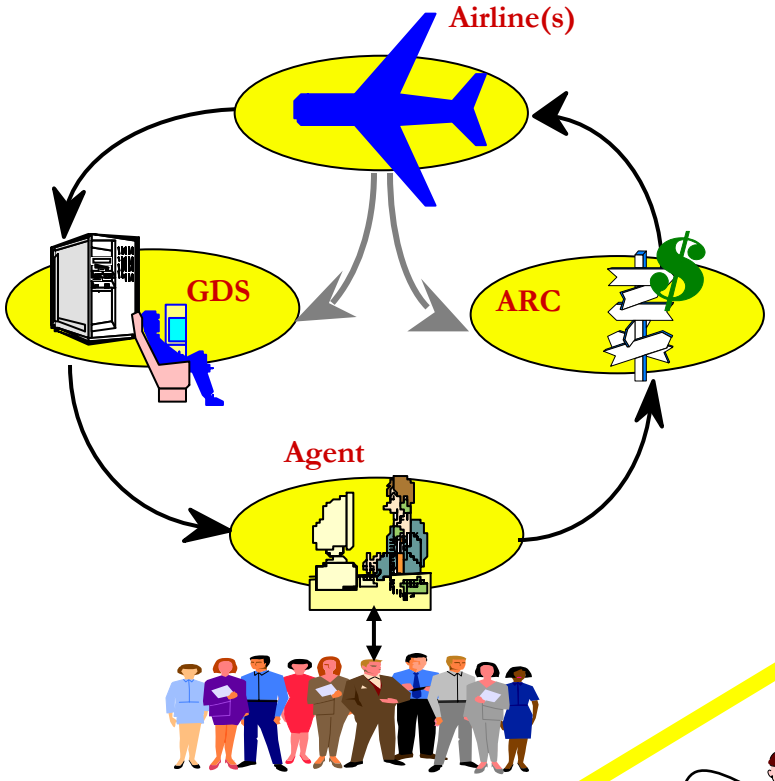


Won't Go Away Soon!

Capital Invested Base
Industry Processes Linkages
Culture Blocks vs. "Change" Drivers

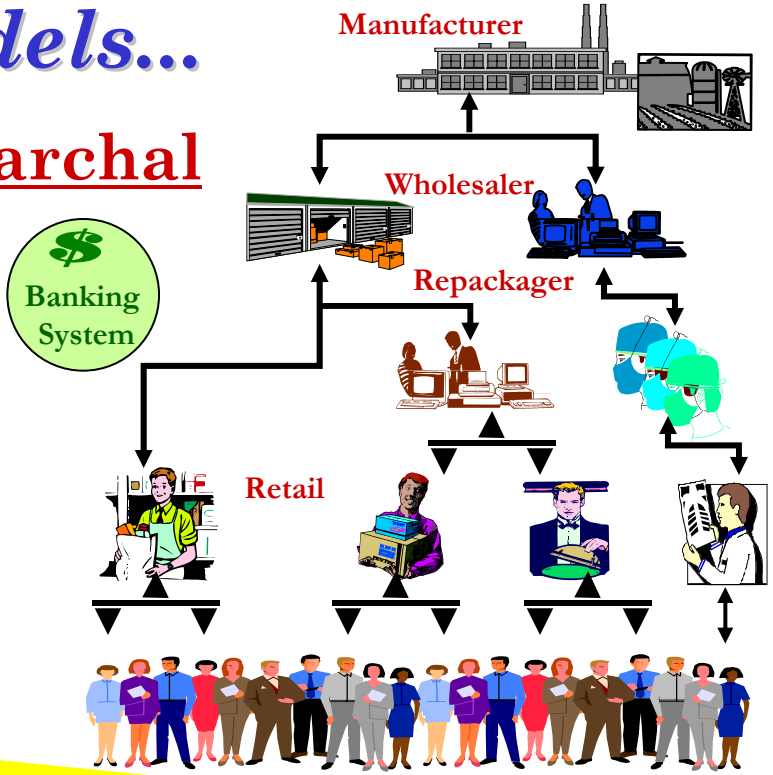
Systems Must Adapt and Blend into Existing Environments

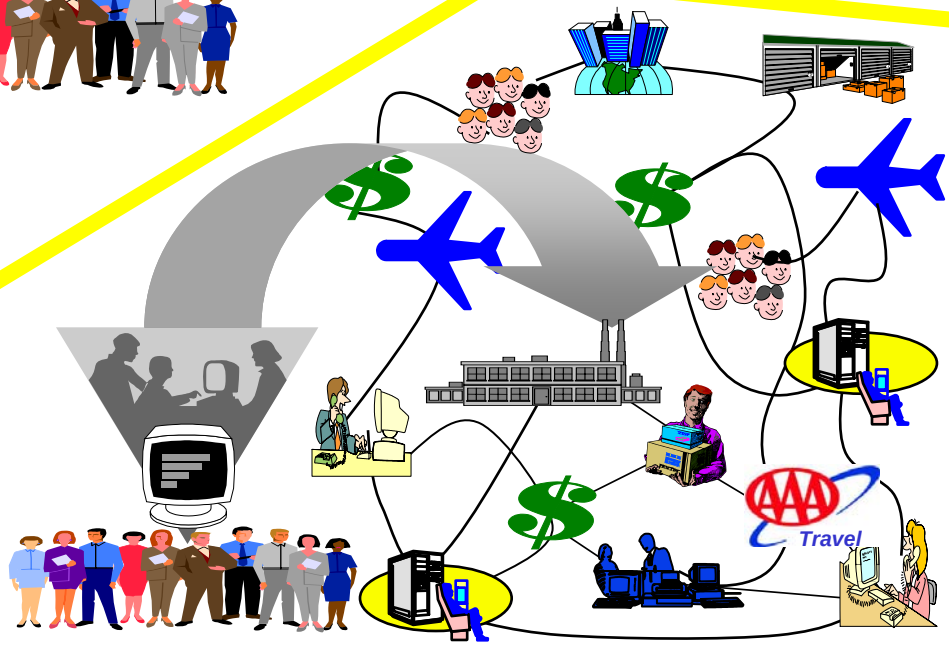
Business/Information Models...



Holistic

Hierarchal



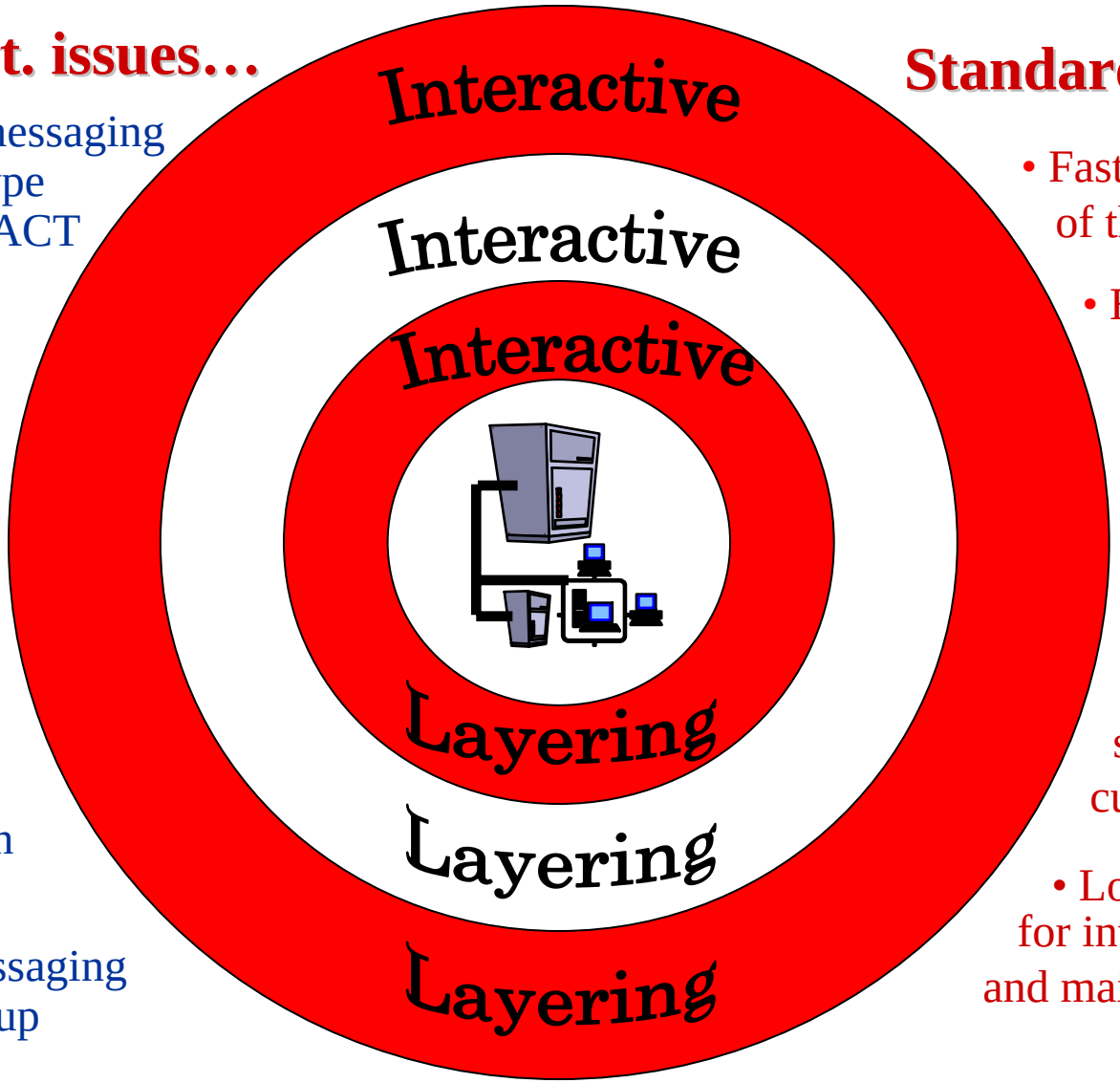


Hyperarchy

Blending Information Models…

Airline/Govt. issues…

- Two types of messaging
  Type B Teletype
  Type A EDIFACT
- Limitations in flexibility
- Markets evolving rapidly
- Suppliers are partnering and alliances forming
- Communication is increasing
- Traditional messaging is not keeping up



Standards/Gateway…

- Faster exploitation of the Internet
- Easier use of cross-industry technology
- Tighter integration with suppliers and customers
- Lower costs for internal support and maintenance

Confronted With…

OpenTravel Alliance

amadeus

WORLDSPAN

Sabre

GALILEO INTERNATIONAL

XML/edi Group

ASC X12

IFX Forum

UN/EDIFACT

AutoLink™

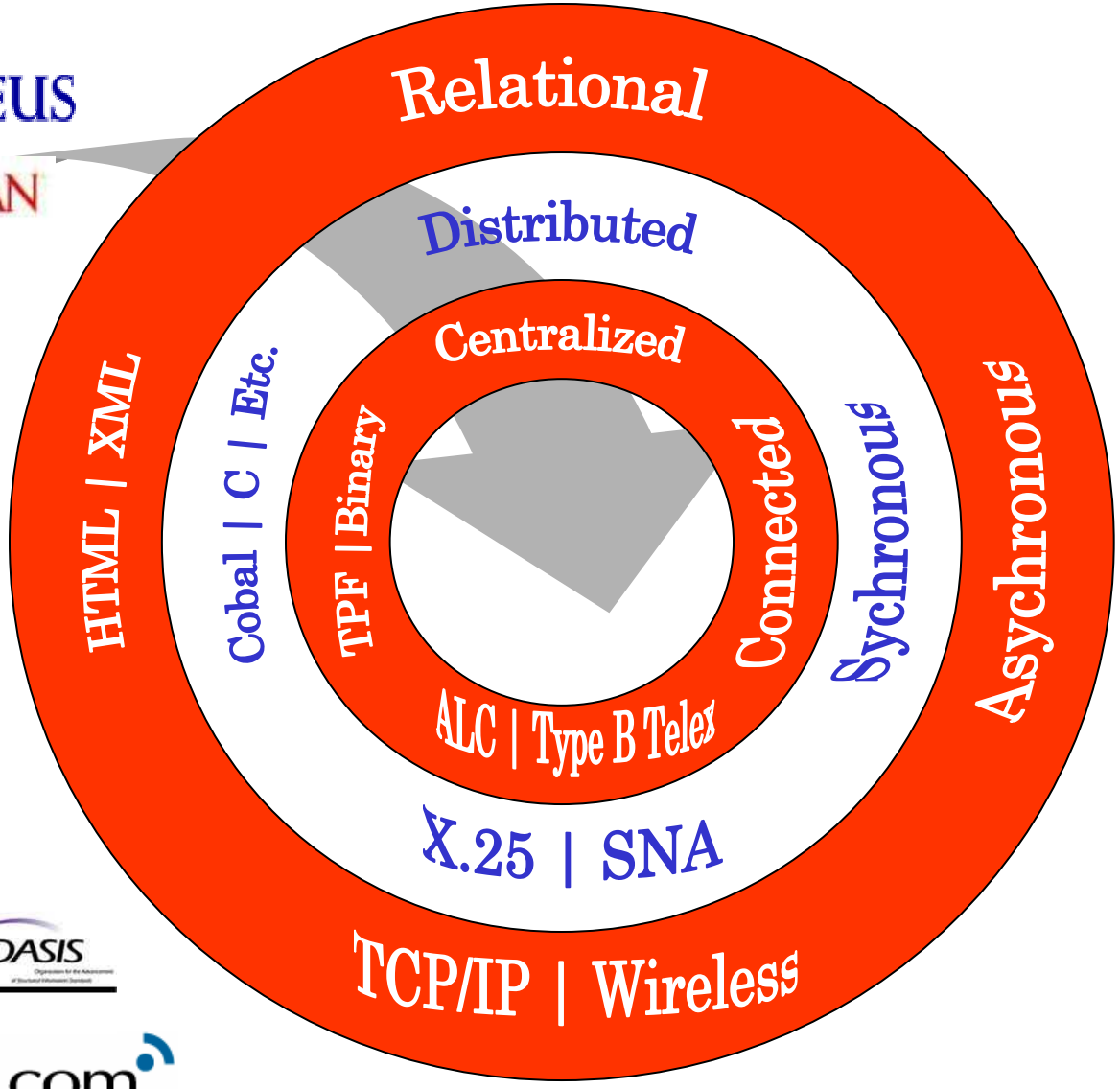
DATALEX TRAVEL SOLUTIONS

ebXML

OASIS

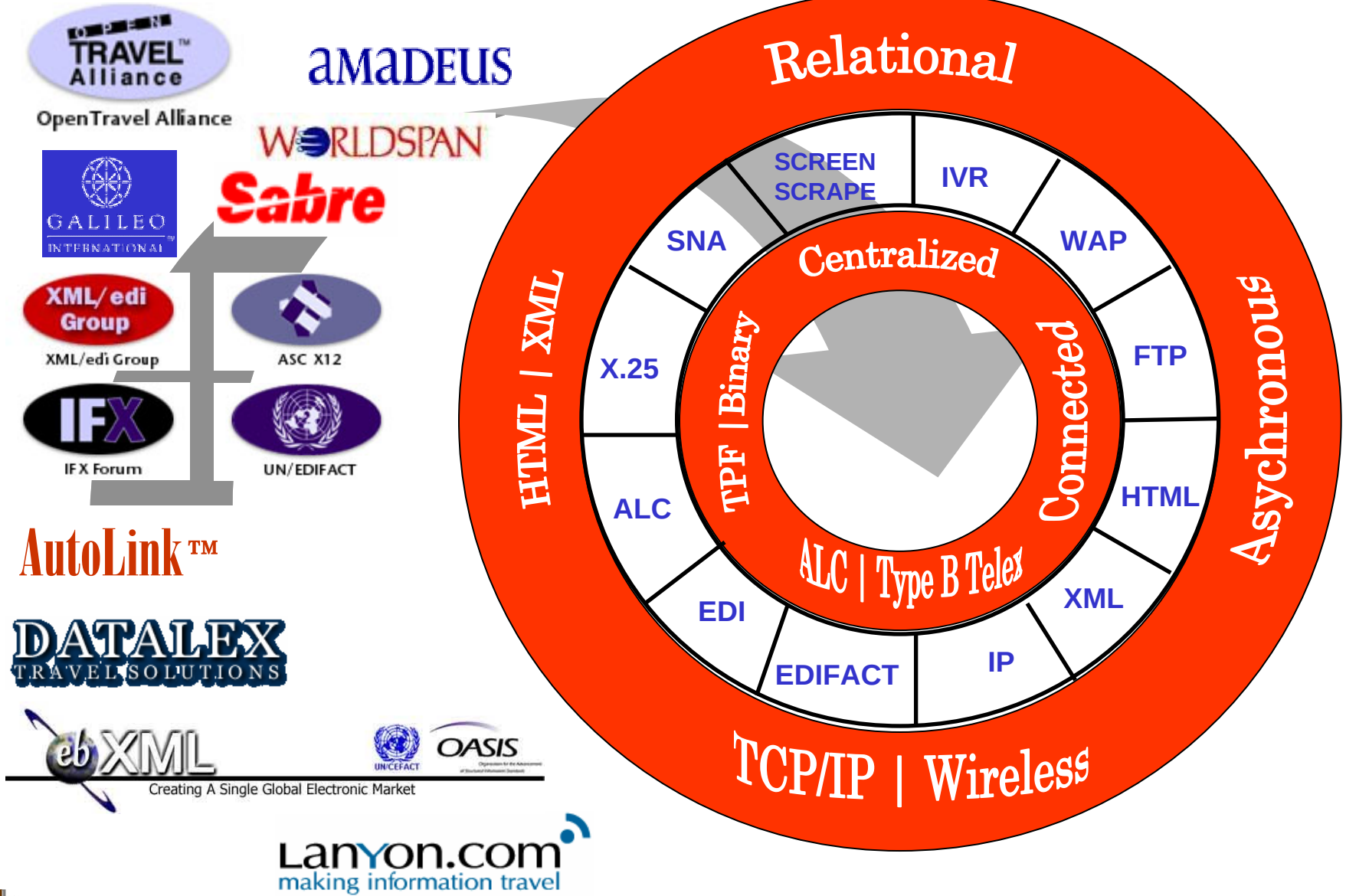
Creating A Single Global Electronic Market

Lanyon.com making information travel



Layered Messaging…

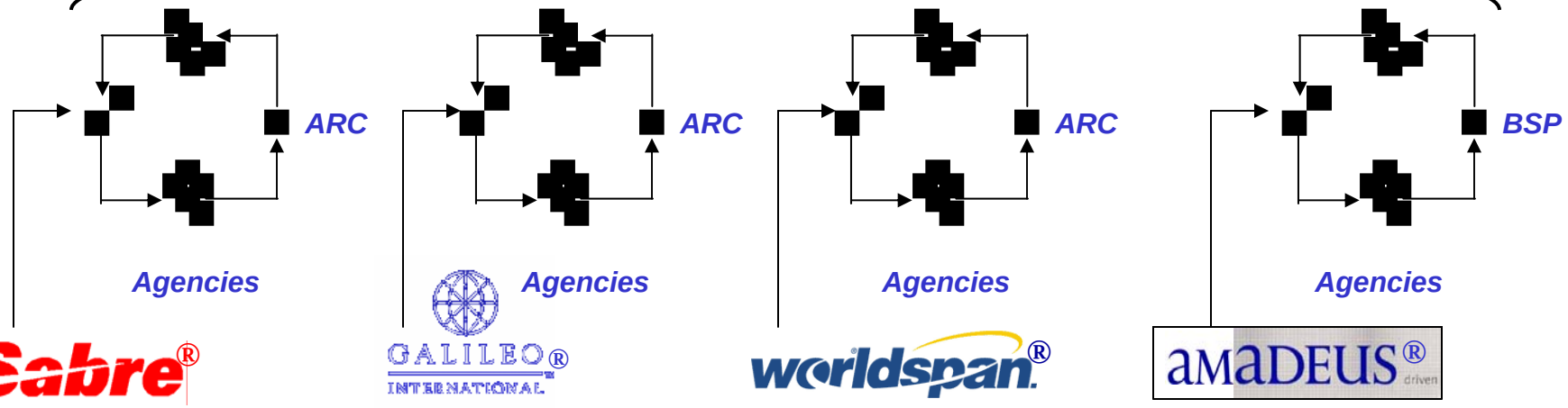
Confronted With...



Layered Messaging...

Proprietary "Silo" Structures…

AA, UL, DL, WN, NW, CO, HP, AS, AQ, HQ, TZ, LH, KL, AC, BA, AF, SR, QF, SQ, IB, MX, AM, JL, VP, …

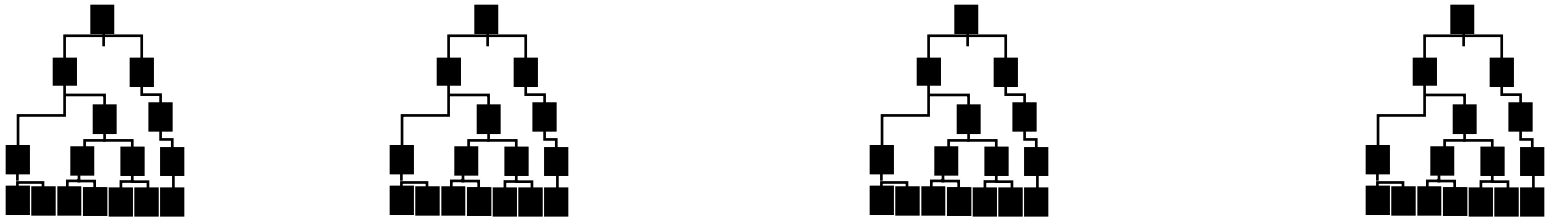


Federal Bureau Investigation

Central Intelligence Agency

Customs

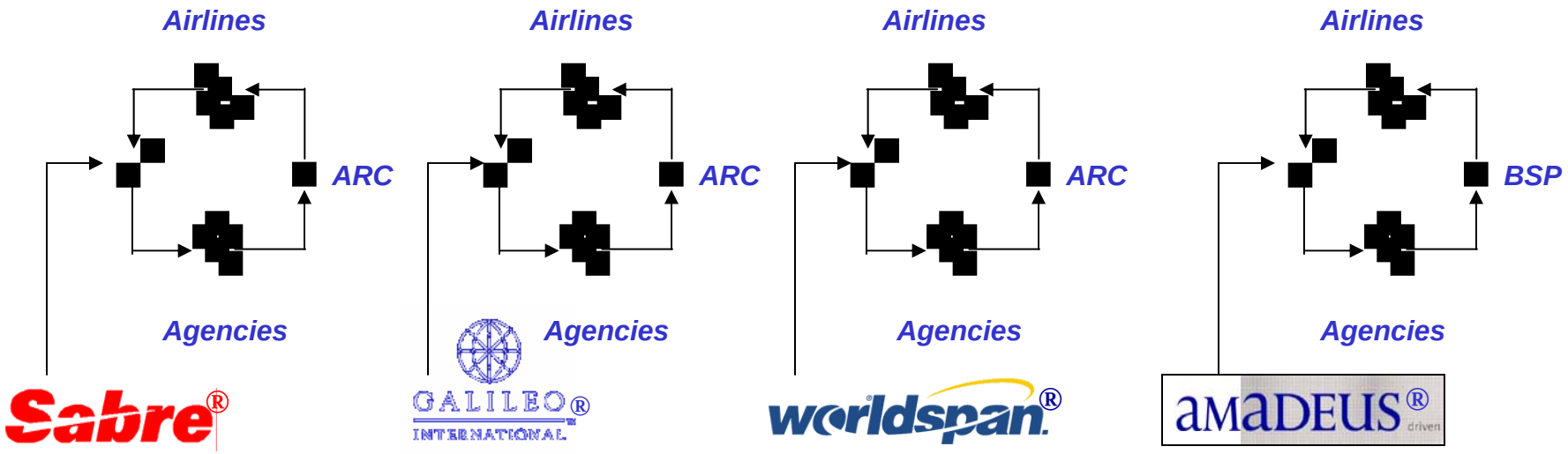
Immigration & Naturalization



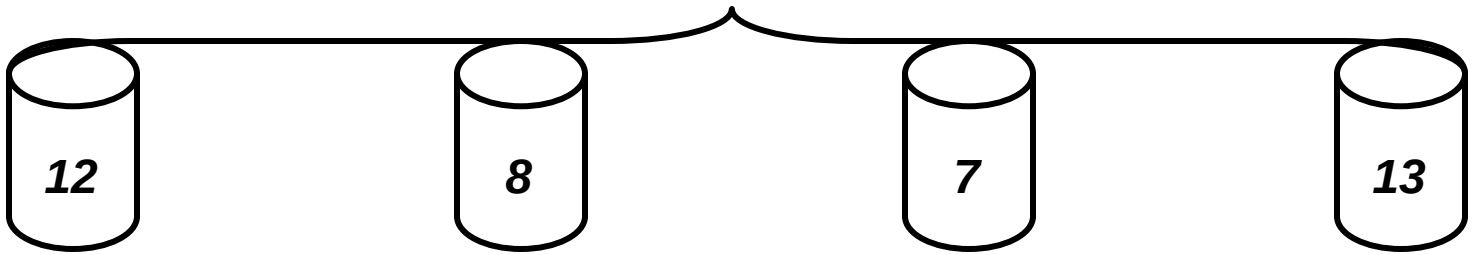
"Proprietary" necessary to serve the competitive business or unique agency function of each entity!!!

Case Study…

Silo Structures



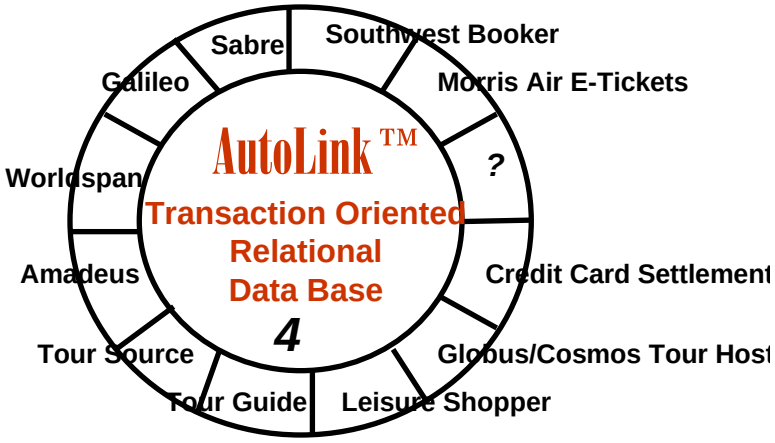
AQUA® Quality Control (1987-1993)



1. *1993 – TEG / AQUA Split*

2. *How to do with 4 …*
   what we had done with 40?

3. *Created Transaction Parser*
   using Relational Data Base



A Solution…

Banks
Government
Back Office
Accounting
Corporations

DavEl Limo
DoD Travel Systems
Skyjet
Globus
Brendan
Etc.

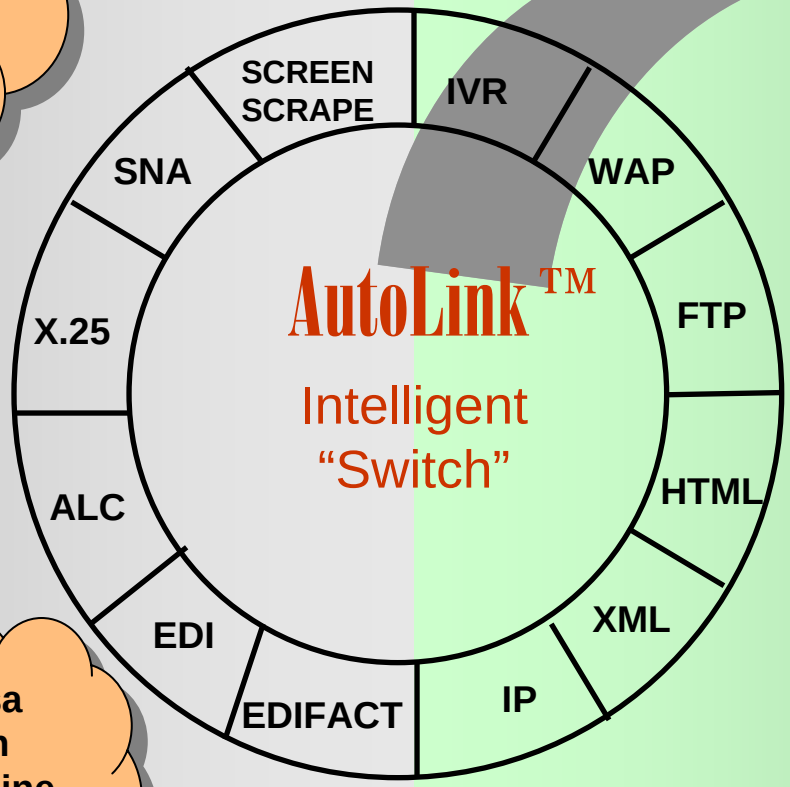
Sabre
Galileo
Worldspan
Amadeus
Abacus

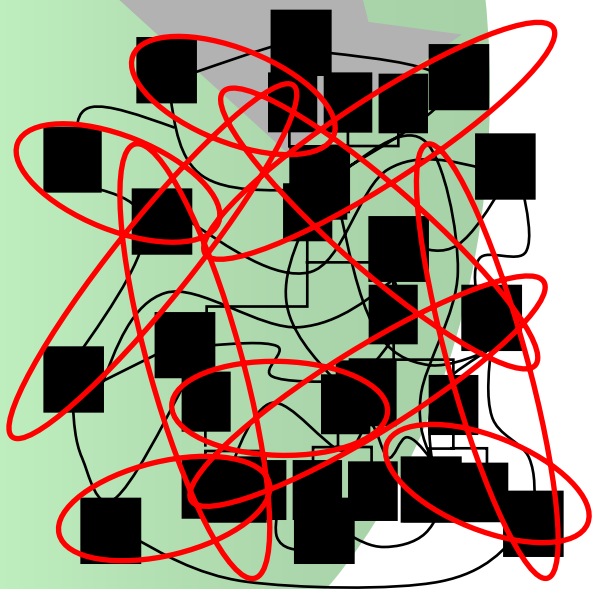
Lufthansa
Austrian
United Airline
Southwest
Sun Country
Etc.

PA Consulting Group

TRAP "rules engine"
Terrorist Research Apprehension Program

AcuFlight, Inc.
ARINC YOU WON'T BELIEVE WHAT WE CAN DO.™
Booking Control Suite





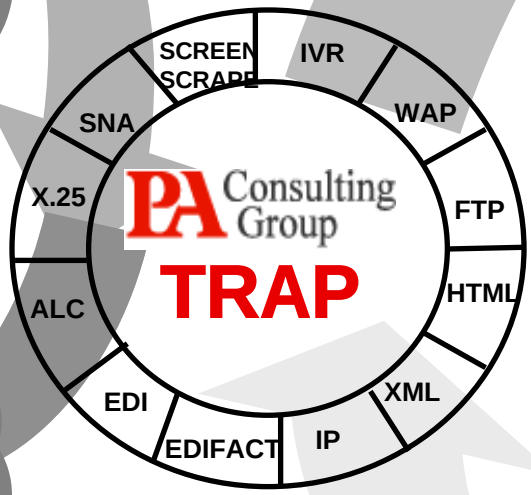
Hyperarchy of Information

A context "engine"…

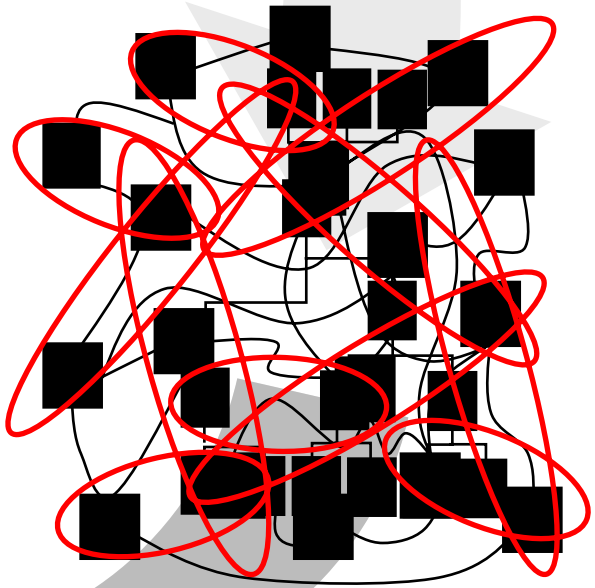
A Solution…

1. Switch Data To/From Different Entities
2. Rules Driven with Secured Access
3. Existing Systems Exist Unchanged
4. Can Function Independently or Integrated

Airline
GDS
Cargo
Shipping
Trucking



TSA
FBI
CIA
Customs
INS



Hyperarchy of Information

<u>The context</u> … meeting on-demand needs!

THE EASTMAN GROUP, INC.

Thank You!

PA Consulting Group

ARINC YOU WON'T BELIEVE WHAT WE CAN DO.™

AcuFlight, Inc.

Travel Software Solutions
Technology Consultants
2280 University Drive
Suite 104
Newport Beach, CA 92660-3328
USA
+1 (949) 574-1505 : Fax 574-8369