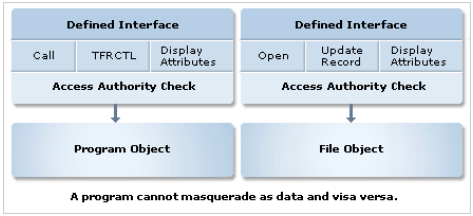
The Evolution of the IBM i Architecture

Steve Will
IBM i Chief Architect
@Steve_Will_IBMi
http://bit.ly/you_and_i_blog

IBM i Architecture

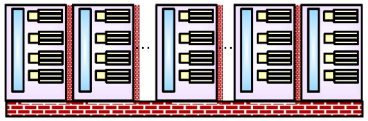
DB2 for i & Single Level Storage

Security & Integrity - Object Based

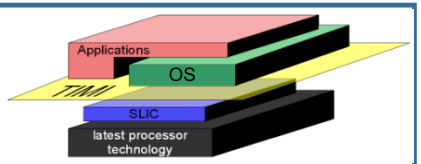


Integration & PASE

Multi-Workload Virtualization



Technology Independent Machine Interface



A system designed for business

Hey! Wait a minute.

Hey! Wait a minute.

Hey! Wait a minute.

2008

2000

IBM iSeries

IBM Power Systems

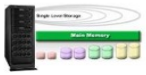
1988

AS/400®

System/38 (1978)
System/36 (1983)

ARCHITECTURE

DB2 for i & Single Level Store



Automate & optimize storage management

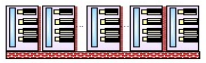
Object Based Architecture

Enables integrity, security, virus-resistance

Integration

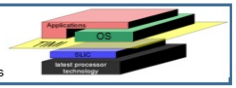
Integrates business components, e.g. DB2 database

Virtualized Work Management



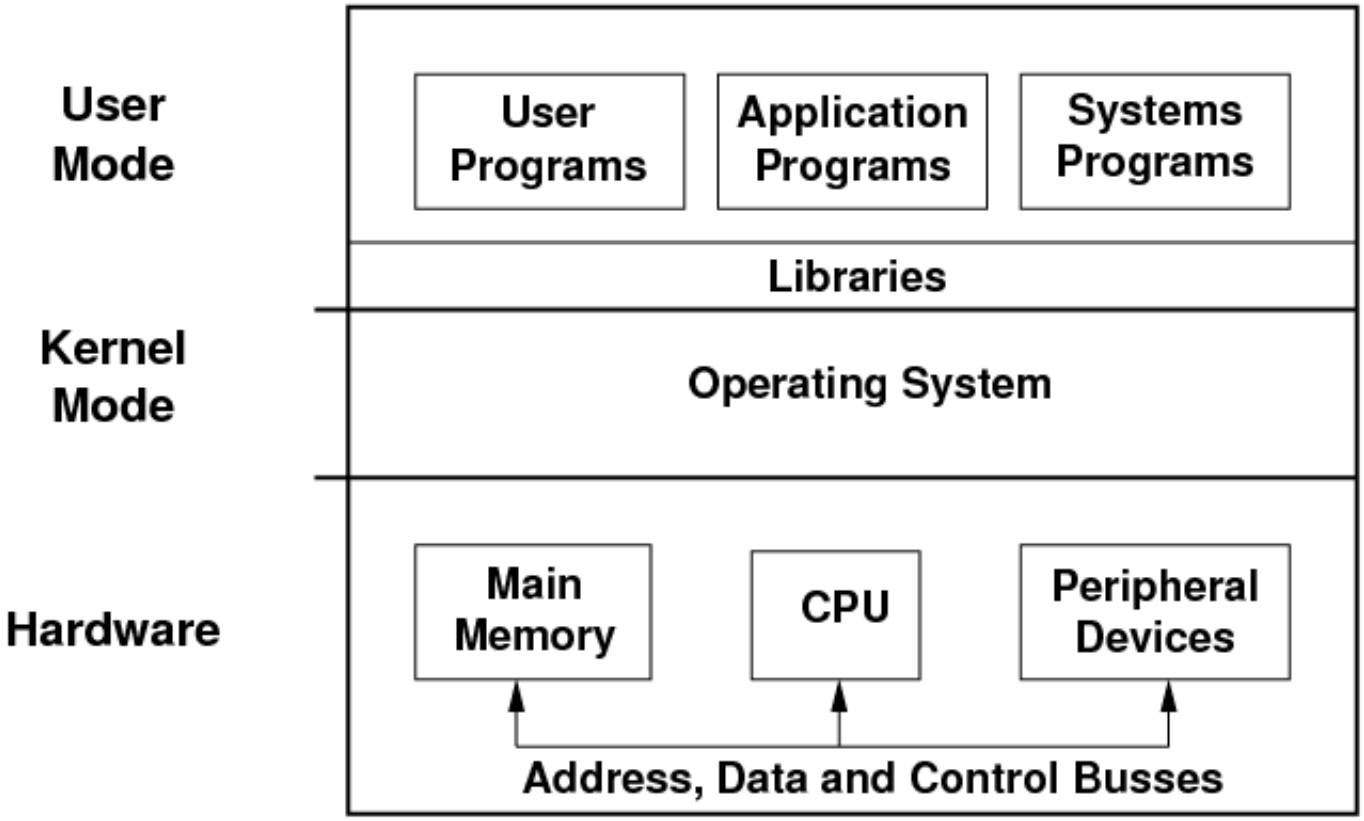
Provides built-in application virtualization

Technology Independent Machine Interface



Ensures application compatibility across multiple technology generations

General View of an OS Architecture



http://minnie.tuhs.org/CompArch/Lectures/week01.html

Layered Architecture of OS/400

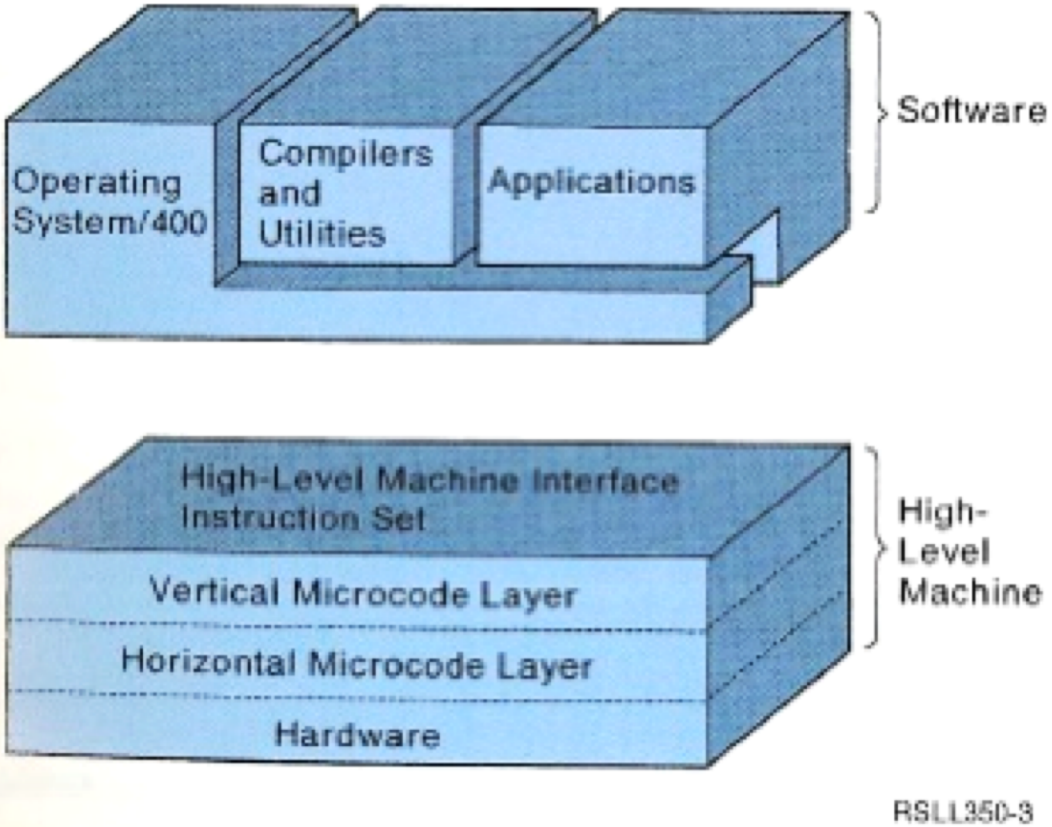
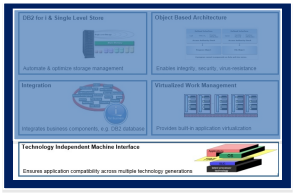


Figure 1 AS/400 Layered Architecture

Applications are compiled to an intermediate language, not processor instructions.

The “MI” (or “TIMI”) is the defined set of these instructions.



Layered Architecture of OS/400

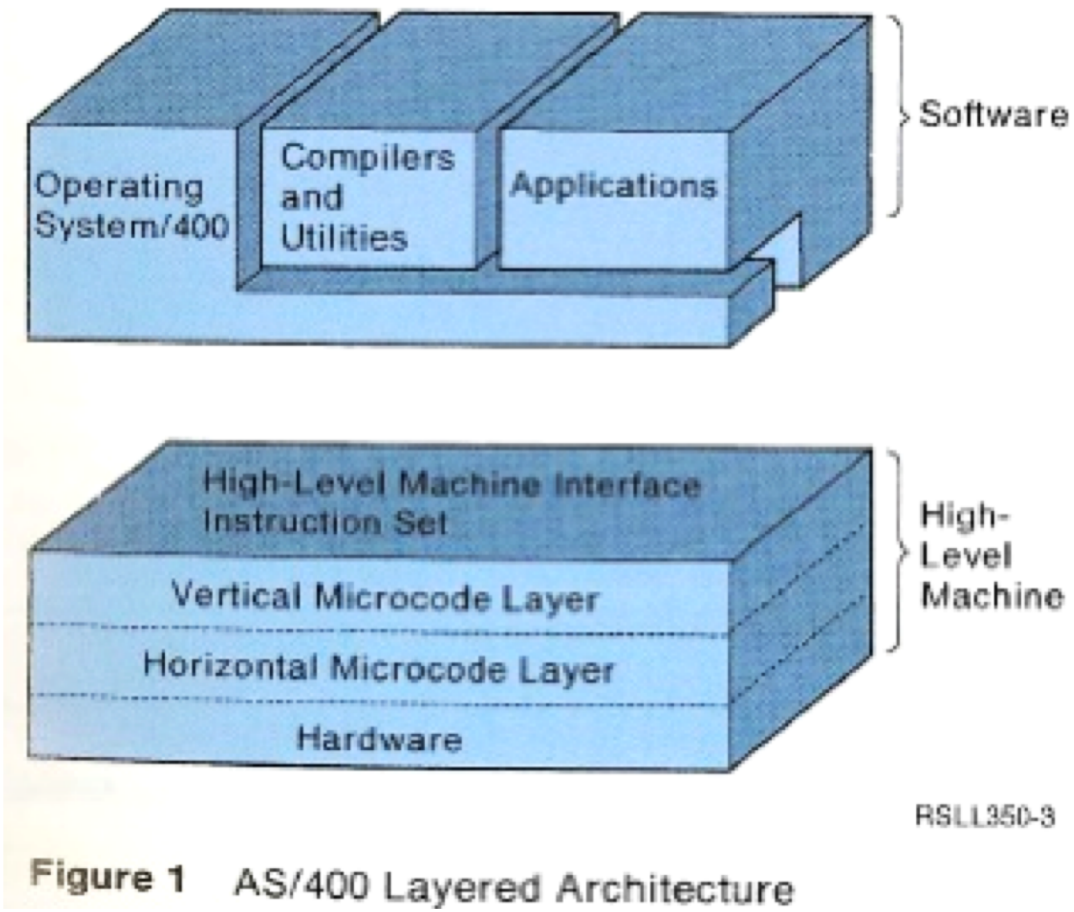


Figure 1 AS/400 Layered Architecture

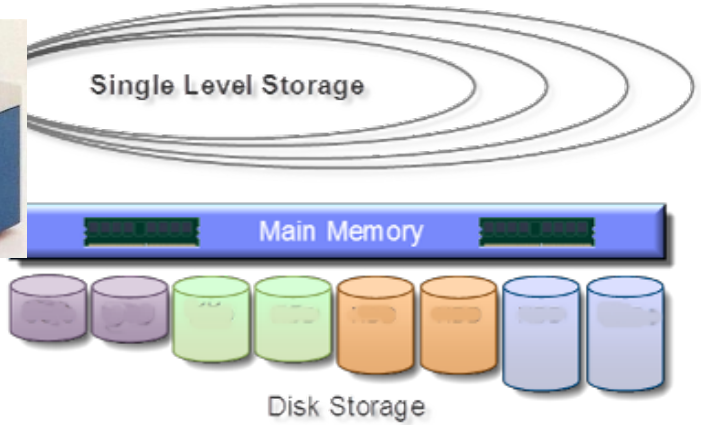
Applications are compiled to an intermediate language, not processor instructions.

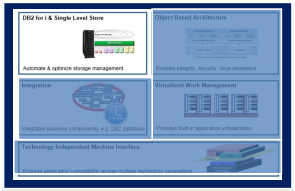
The “MI” (or “TIMI”) is the defined set of these instructions.

Storage Model – Back to System/38

All storage on the system is treated as a single contiguous set of memory, so mapping storage required special methods and knowledge of storage devices.

System/38 and initial AS/400 used 48-bit addresses for what became known as “Single Level Storage.”

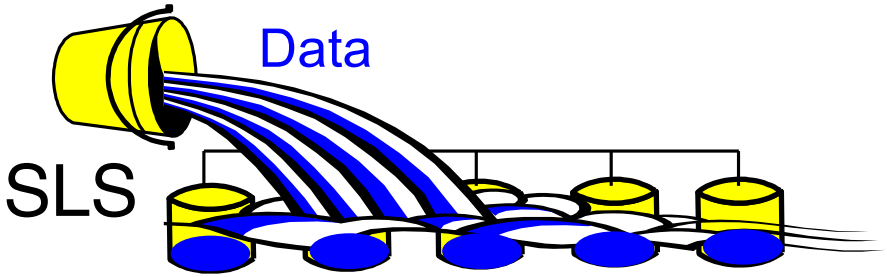


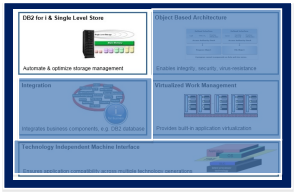


Storage Model – Back to System/38

All storage on the system is treated as a single contiguous set of memory, so mapping storage required special methods and knowledge of storage devices.

System/38 and initial AS/400 used 48-bit addresses for what became known as “Single Level Storage.”





Storage Model – Back to System/38

All storage on the system is treated as a single contiguous set of memory, so mapping storage required special methods and knowledge of storage devices.

System/38 and initial AS/400 used 48-bit addresses for what became known as “Single Level Storage.”

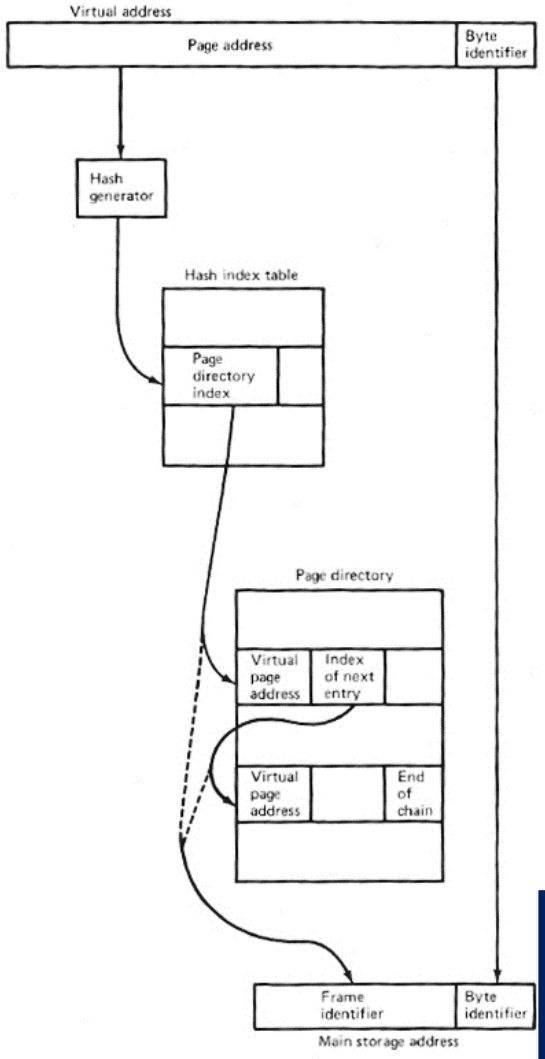
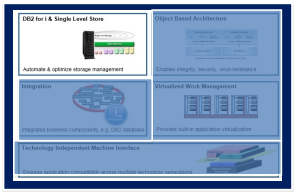


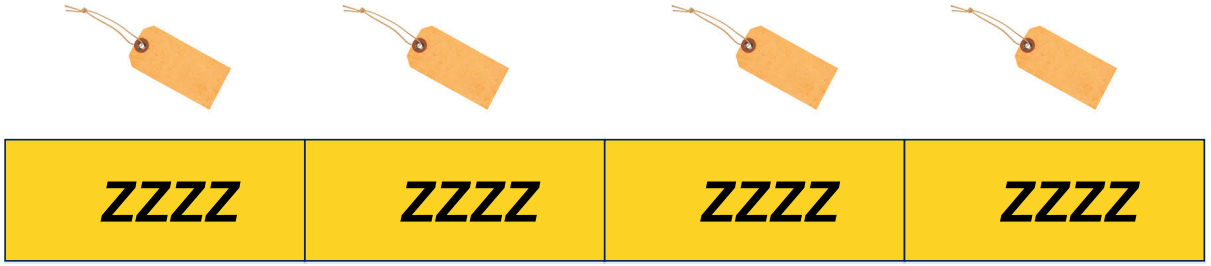
Figure 1 Virtual address translation

Tags

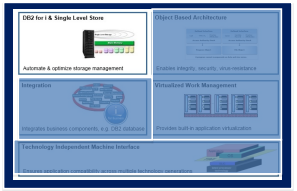
ZZZZ	ZZZZ	ZZZZ	ZZZZ



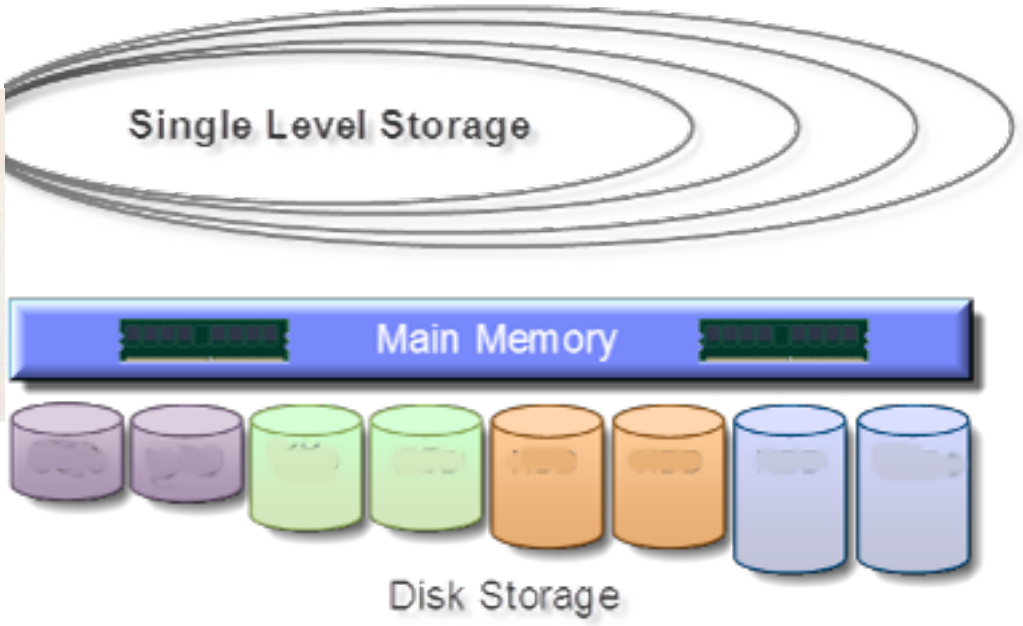
Tags

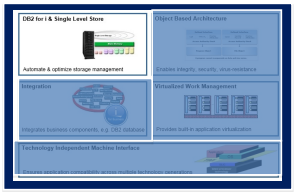


A “tag” per 32-bit word to indicate it’s part of a pointer.

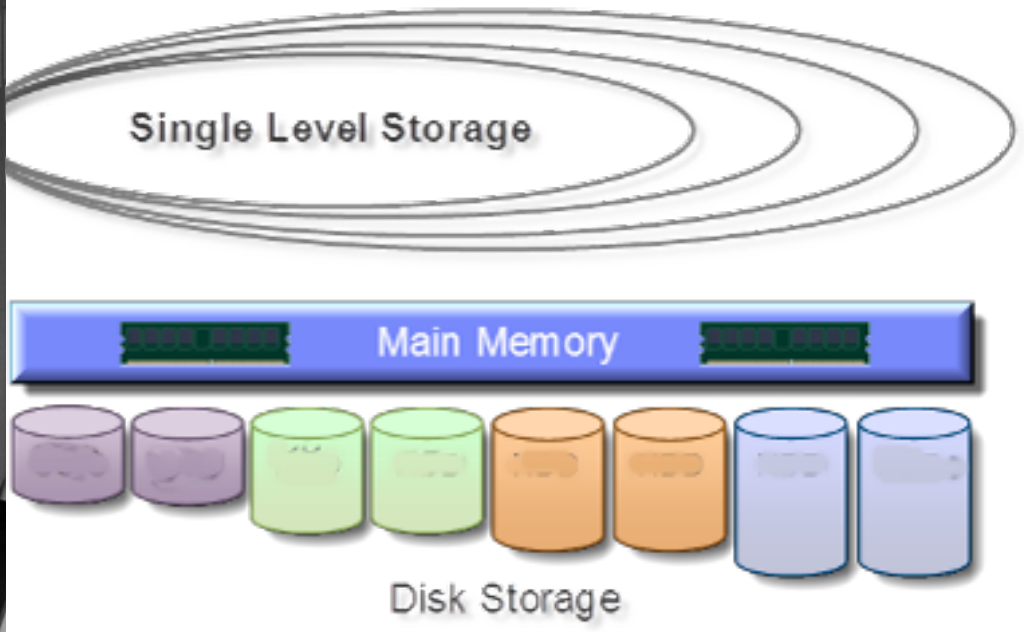


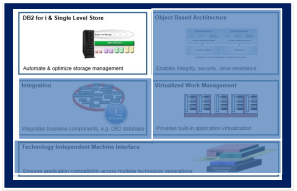
System/38 Single Level Storage



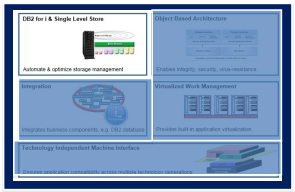


AS/400 Single Level Storage – The Same?





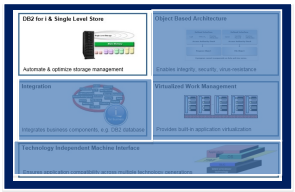
From System/38 to IBM i – “Single” Changed



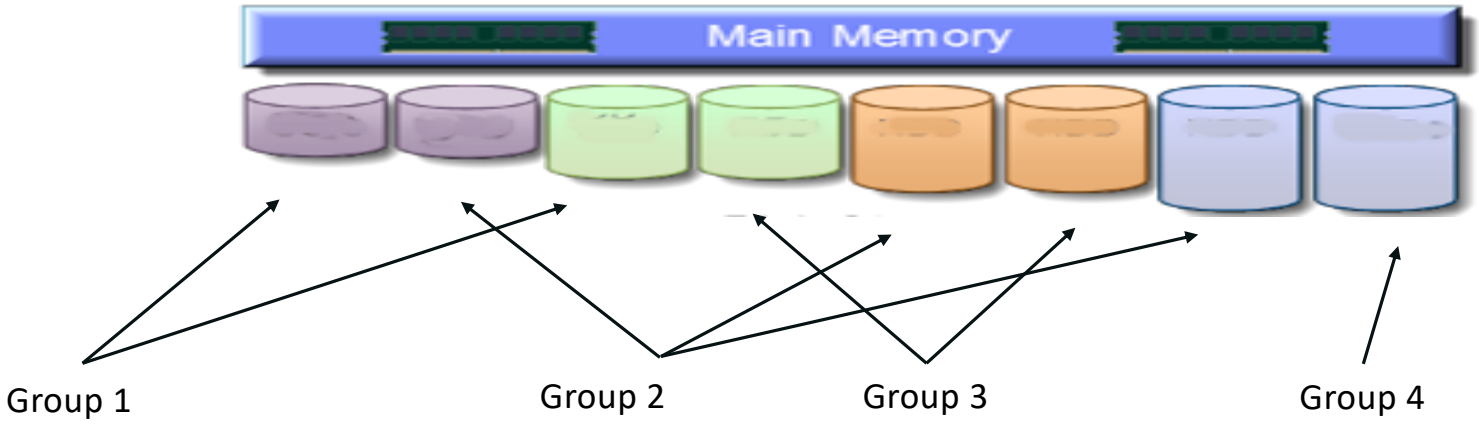
From System/38 to IBM i – “Single” Changed

Group 1

The AS/400 architects decided they wanted to have groups of disks.

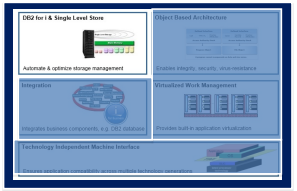


From System/38 to IBM i – “Single” Changed



The AS/400 architects decided they wanted to have groups of disks.

These groups were called Auxiliary Storage Pools.



Integrated DB

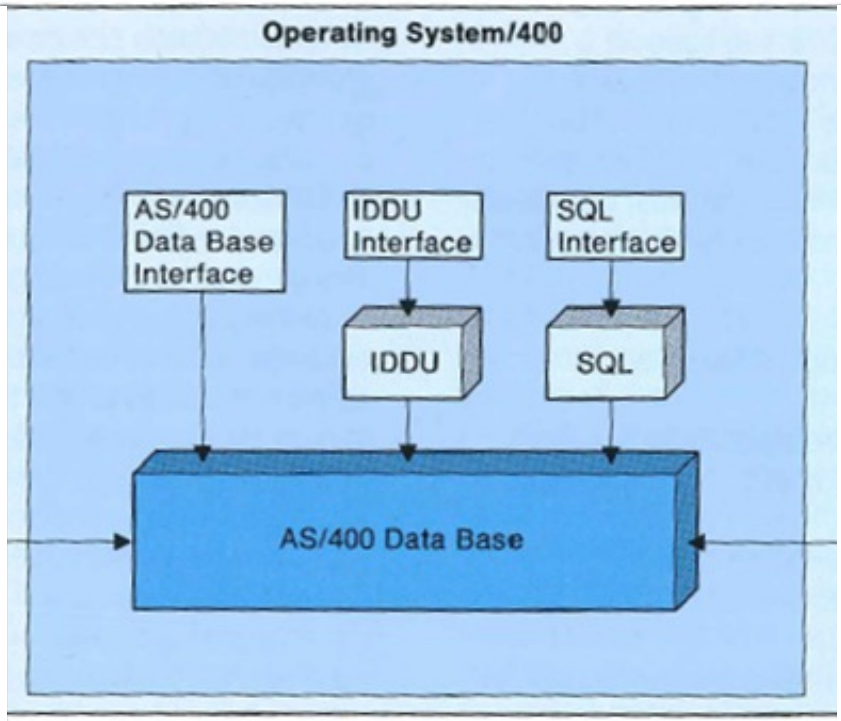
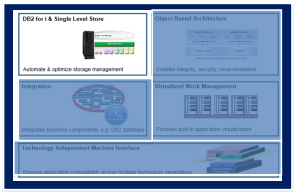


Figure 1 Interface to AS/400 Data Base



Integrated DB

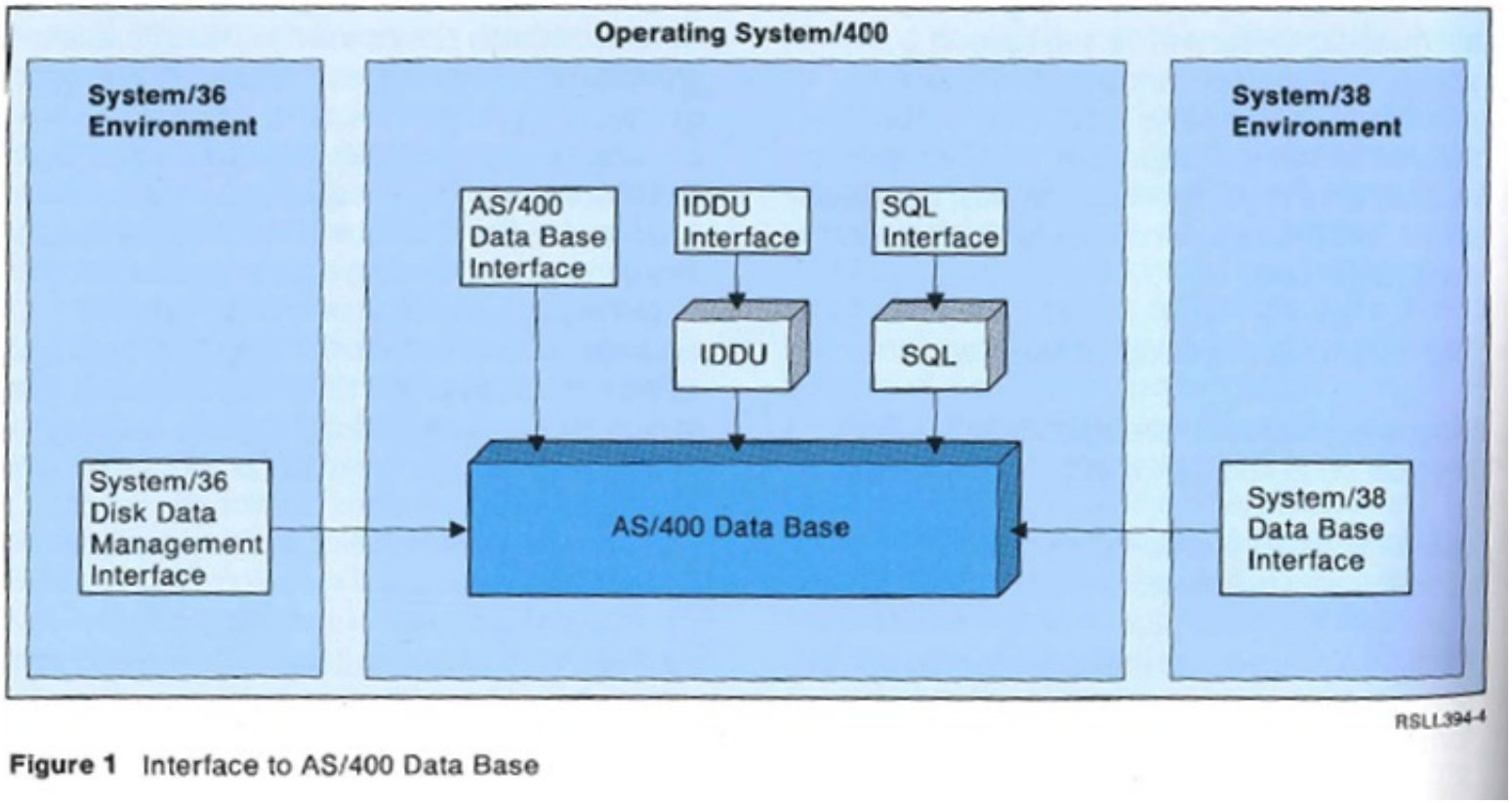
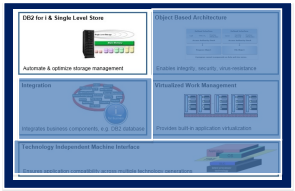


Figure 1 Interface to AS/400 Data Base



Object Orientation

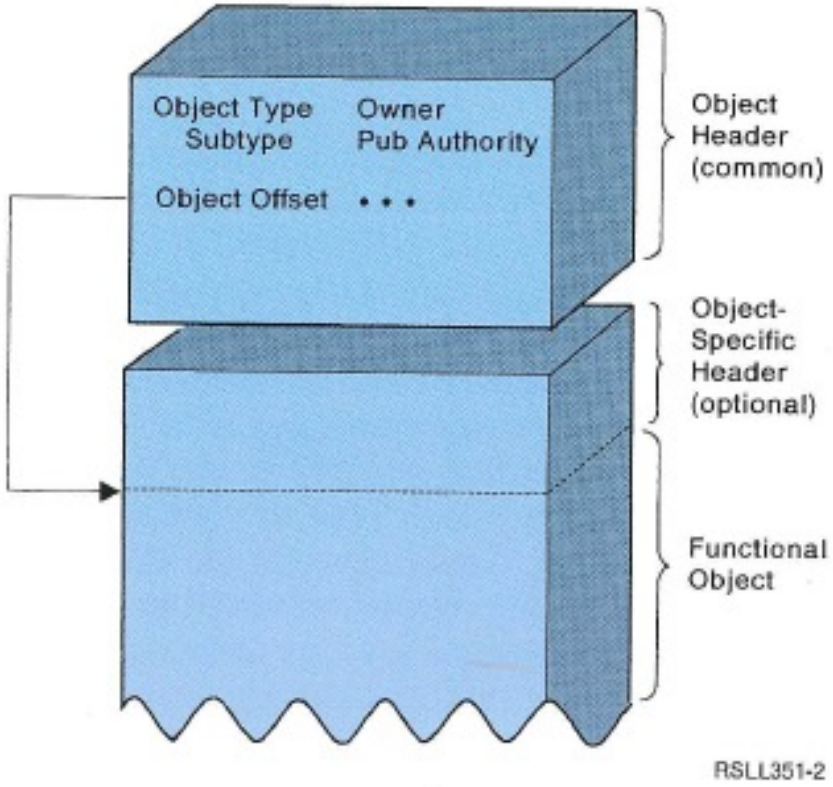
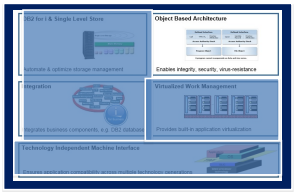


Figure 2 Structure of Generic Object

Objects protect the integrity of the system and customer data, while also allowing a strict object-based security architecture.



What About Integration & Work Management?

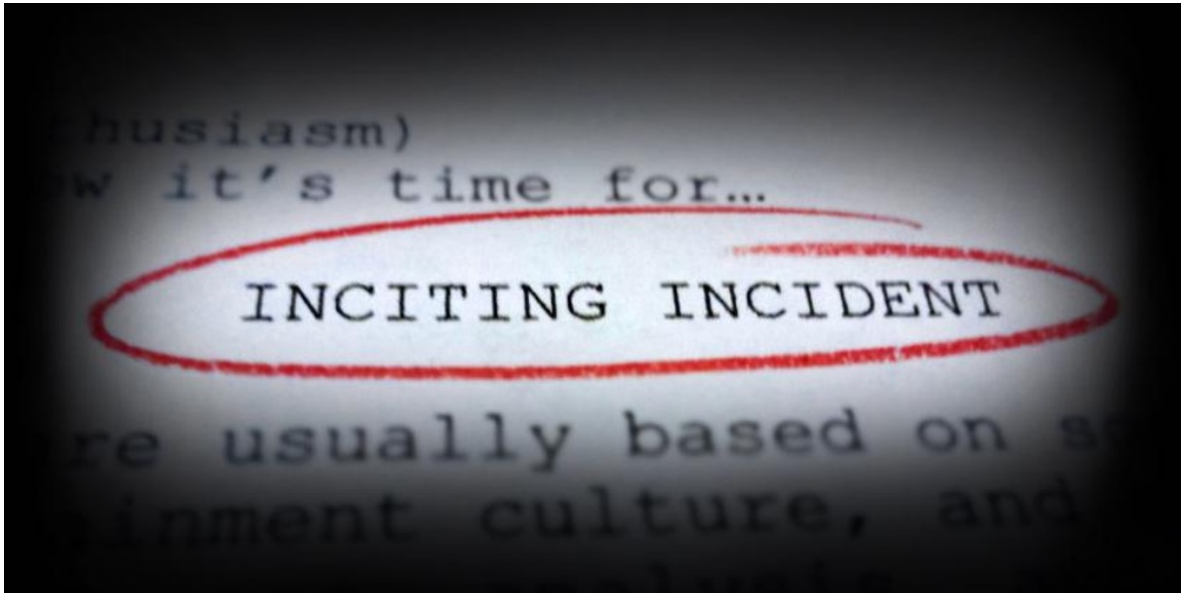
What About Integration & Work Management?

So, AS/400 V1R1 had the perfect architecture …

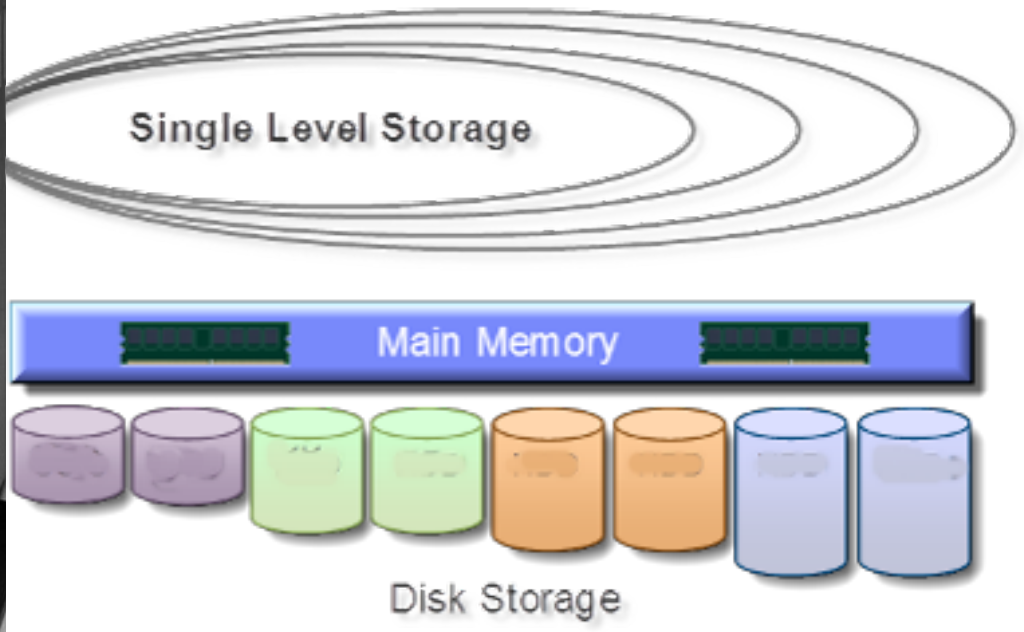
So, AS/400 V1R1 had the perfect architecture …

And then …

And then …



Remember that Single Level Storage Thing?



Programming Languages and Integrity

Programming Languages and Integrity

First:

Programming Languages and Integrity

First:

But Then:

Programming Languages and Integrity

First: RPG COBOL

But Then: THE C PROGRAMMING LANGUAGE

(Pointers!)

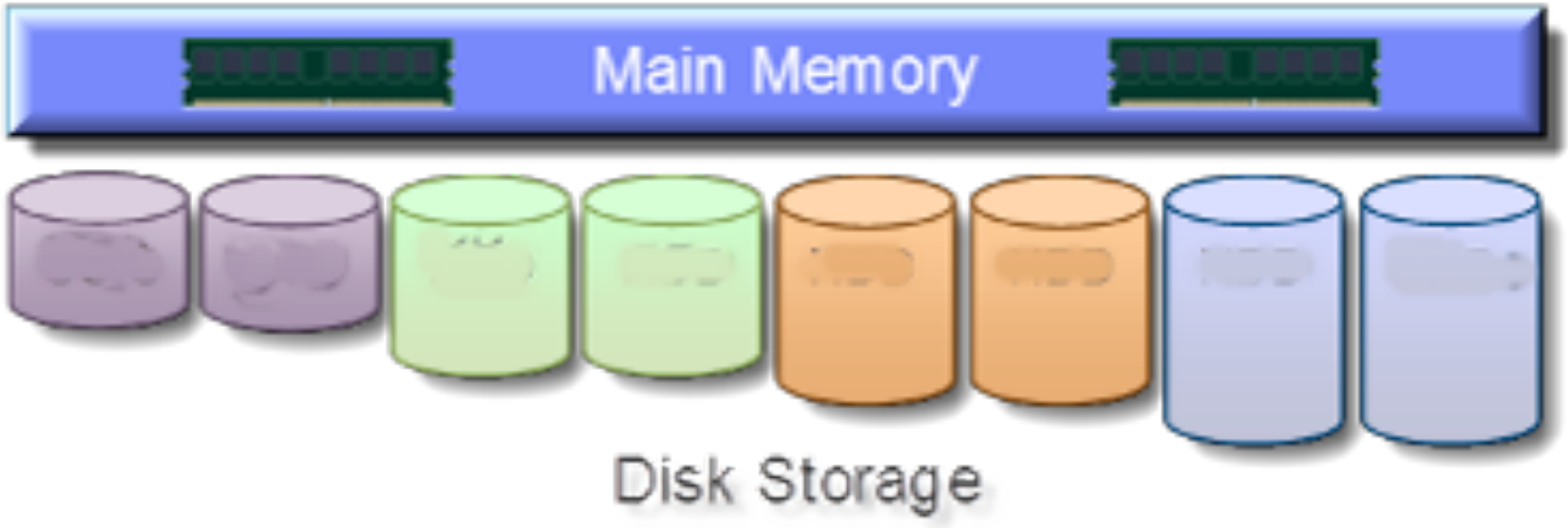
Programming Languages and Integrity

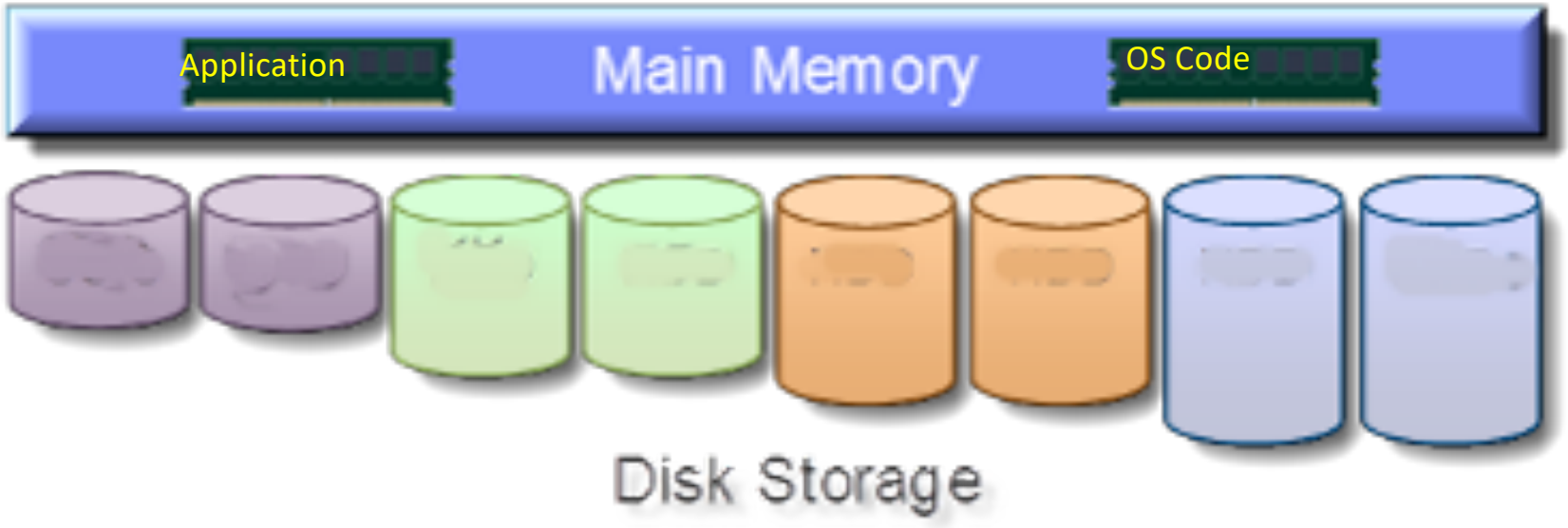
First: RPG COBOL

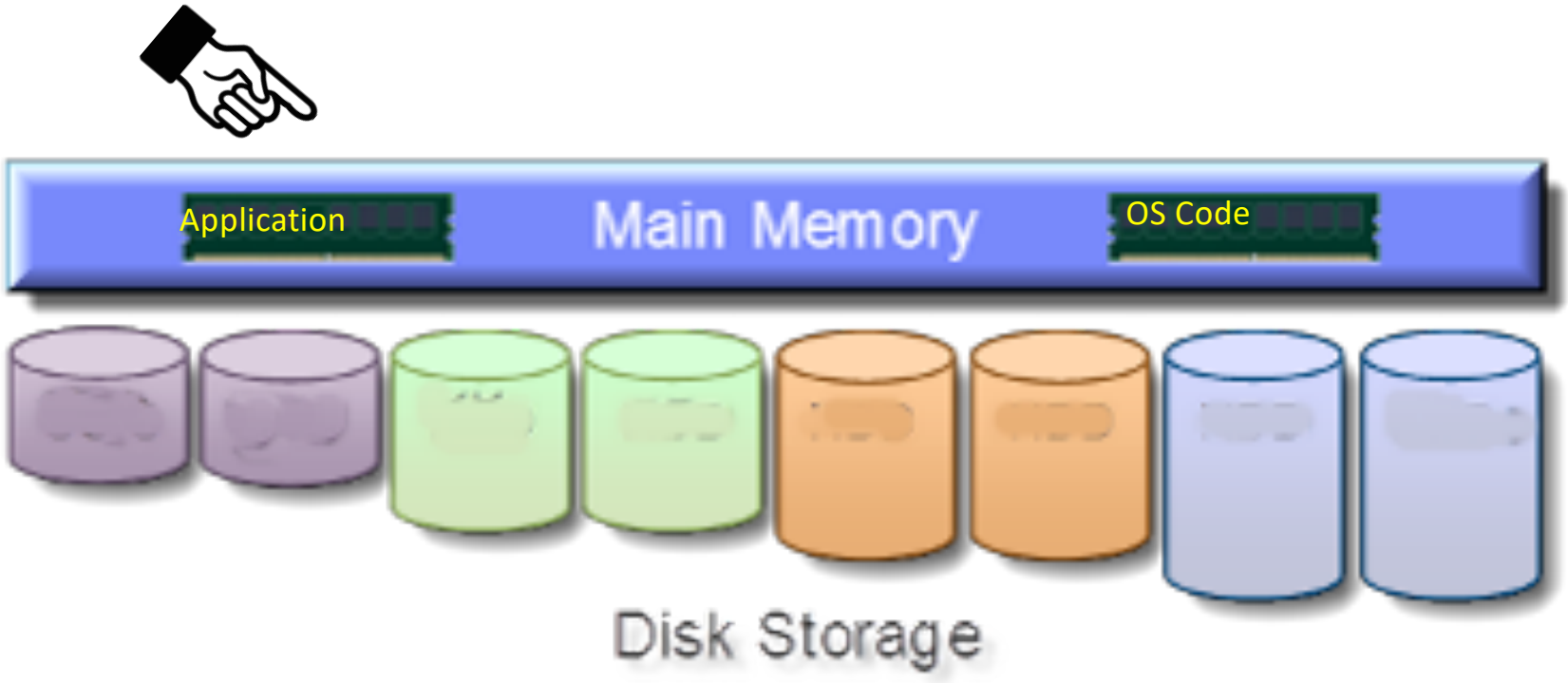
But Then: THE C PROGRAMMING LANGUAGE

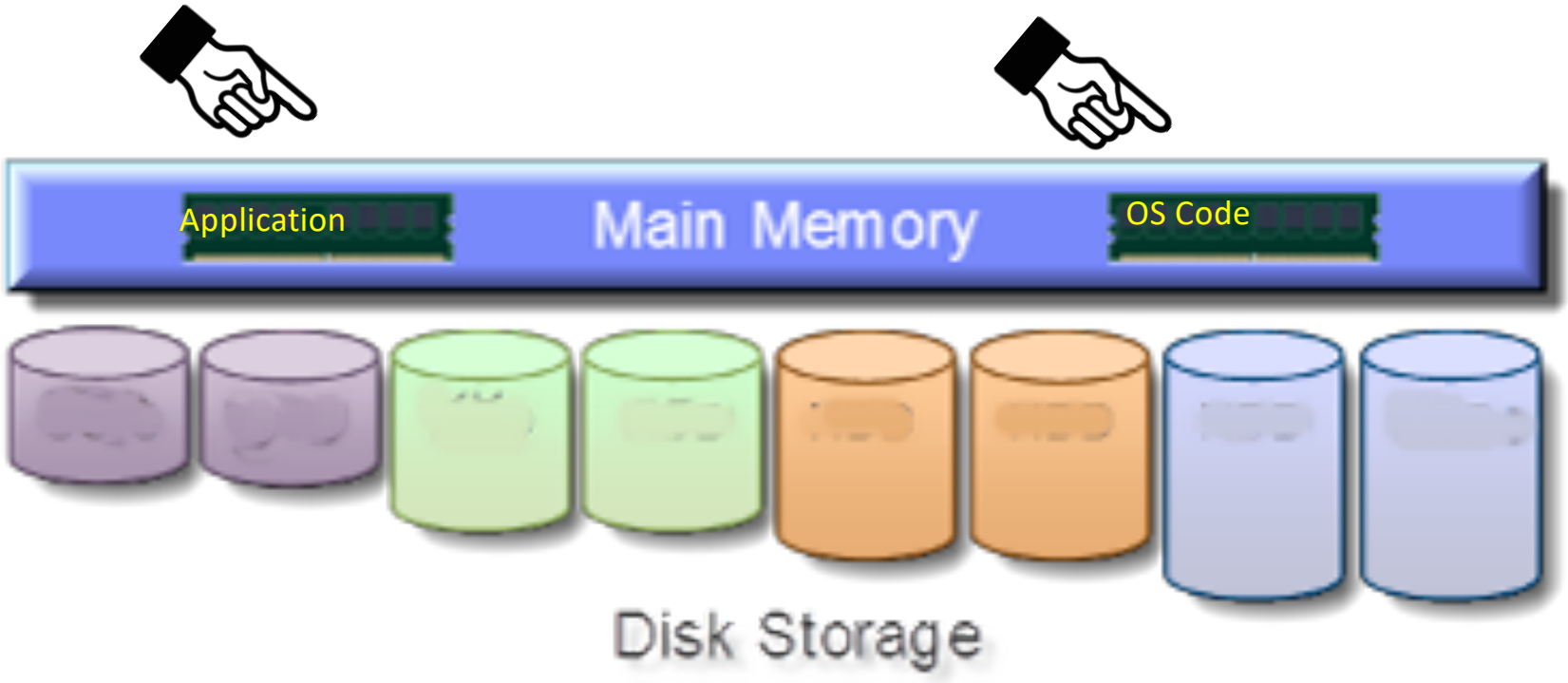
(Pointers!)

So What?









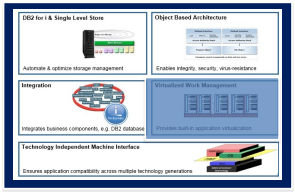
Programming Languages and Integrity

First: RPG COBOL

But Then: THE C PROGRAMMING LANGUAGE

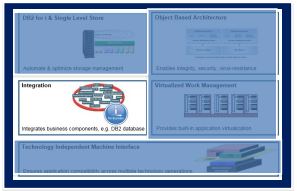
So …

Security level 50 & initial HW support for Integrity (Storage Protection, Privileged/Problem state support)



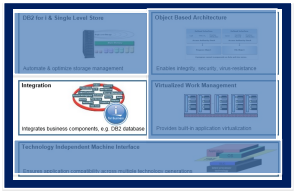
File Systems

- System/38 had one file system – QSYS.LIB – which was (is) “flat”
  - By the late ’70s, hierarchical file systems were proving their value



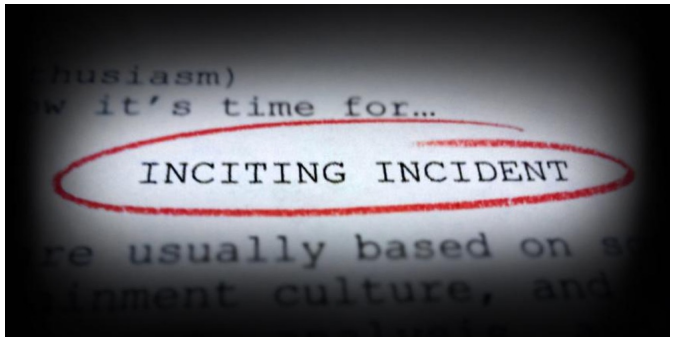
File Systems

- System/38 had one file system – QSYS.LIB – which was (is) “flat”
  - By the late ’70s, hierarchical file systems were proving their value
- System/36 implemented QDLS and AS/400 included it as a system-wide architecture
  - So, again, the AS/400 architecture was more than just the S/38

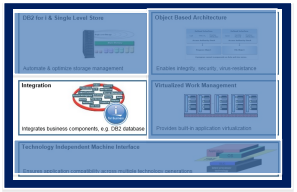


File Systems

- System/38 had one file system – QSYS.LIB – which was (is) “flat”
  - By the late ’70s, hierarchical file systems were proving their value
- System/36 implemented QDLS and AS/400 included it as a system-wide architecture
  - So, again, the AS/400 architecture was more than just the S/38
- The industry defined a standard hierarchical file system, as part of the POSIX standards

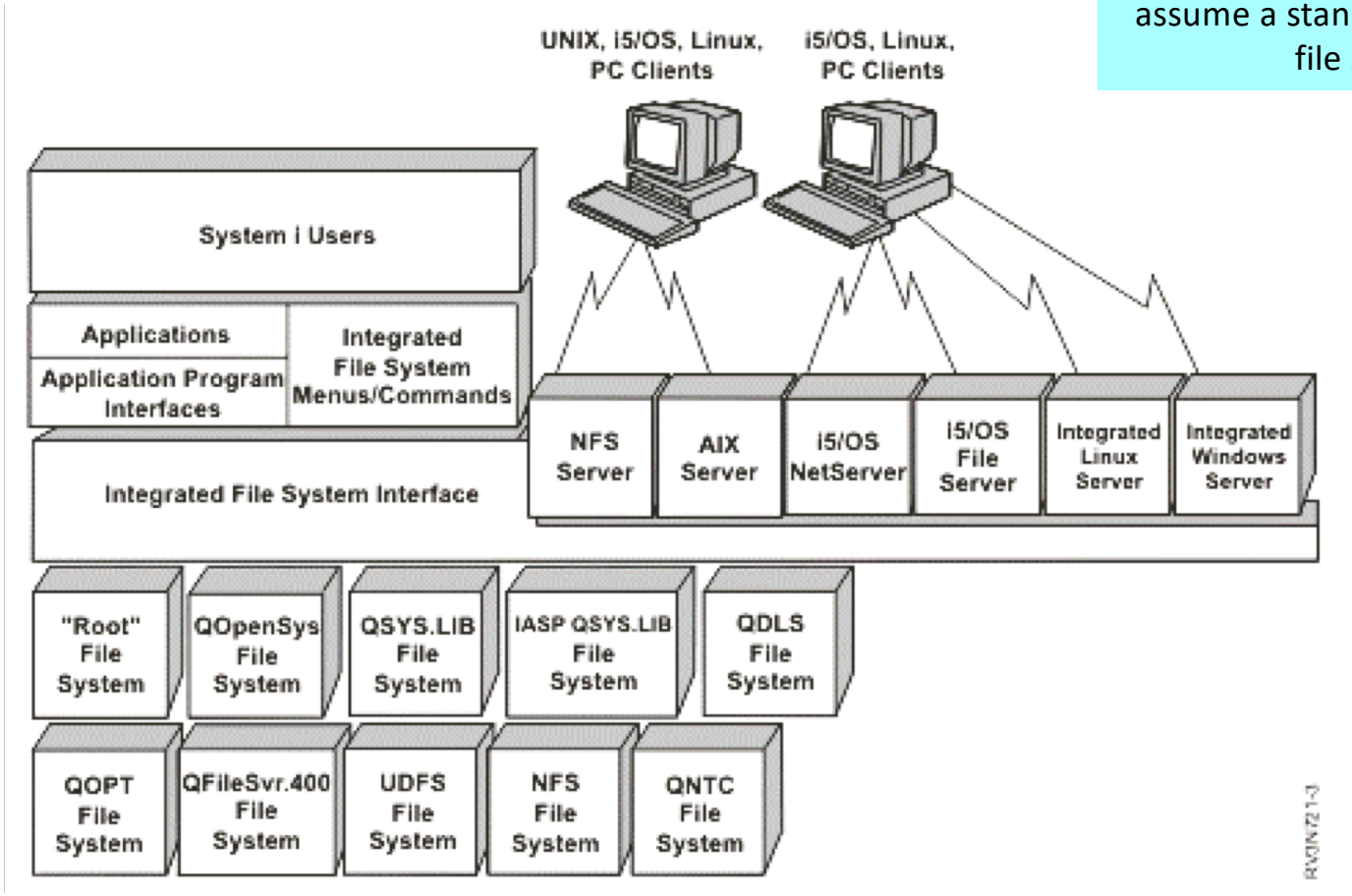


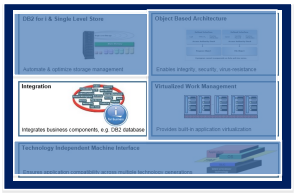
POSIX



Integrated File System - IFS

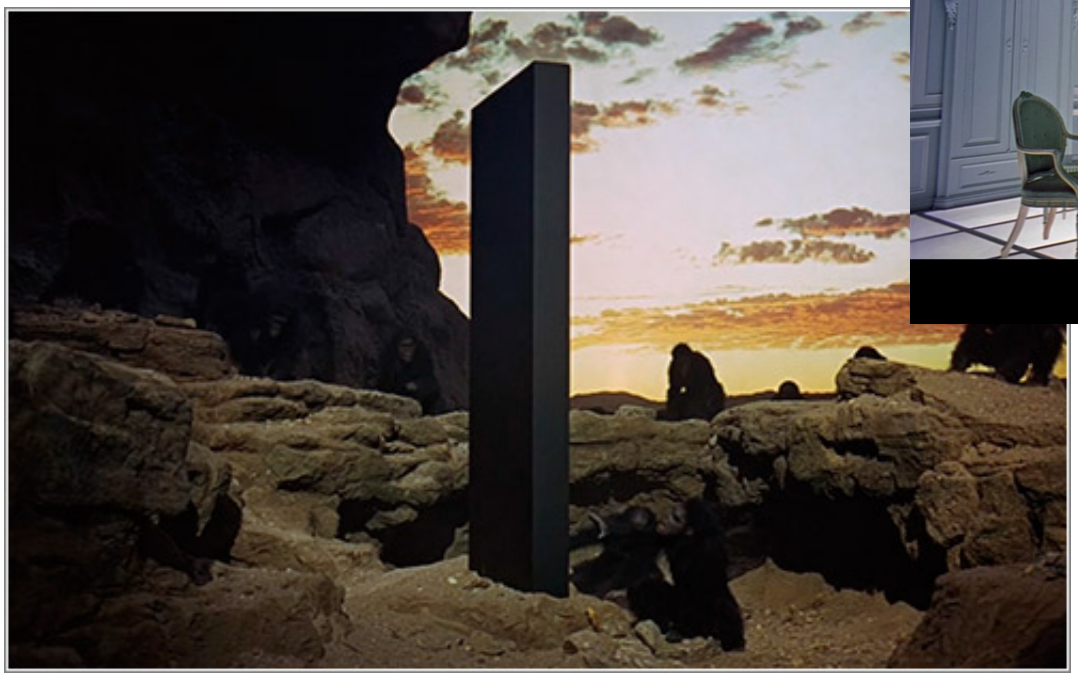
Today's modern applications assume a standard, hierarchical file system





Once Upon a Time

For very good reasons, early software was written in large, **monolithic**, programs.



The Monolith

But the Programming World Kept Changing

But the Programming World Kept Changing

Computer architectures made resources more available, compilers made calls between programs more efficient, and languages developed to assume **modularity**.

But the Programming World Kept Changing

Computer architectures made resources more available, compilers made calls between programs more efficient, and languages developed to assume **modularity**.

ILE

The Integrated Language Environment: aka ILE, aka New Program Model

The creation of ILE required the creation of the **Activation Engine**

- A new architectural component
- Controls the birth, life, and death of the process and its **activation groups**

ILE Benefits

- Binding
- Modularity
- Reusable Components
- Common Runtime Services through bindable APIs
- Source Debugger
- Better Control Over Resources

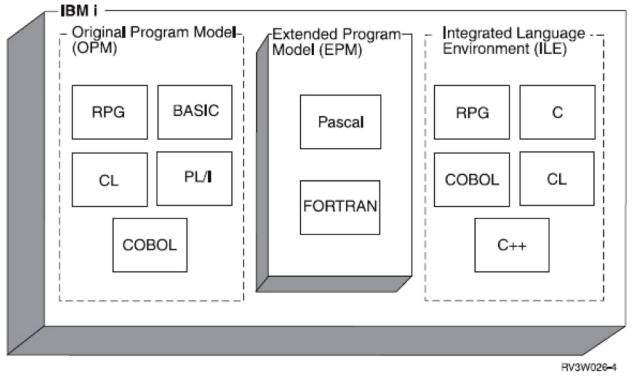
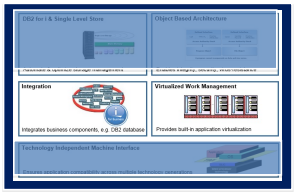
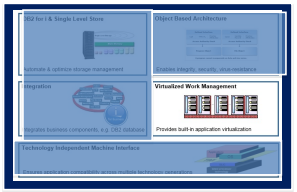


Figure 2-5. Relationship of OPM, EPM, and ILE to IBM i



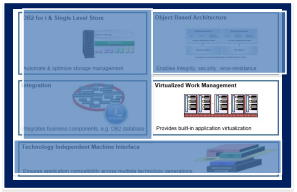
Jobs and Processes Were Enough, Until They Weren't

- Job: the user-visible, user definable container of work
- Process: the underlying above-MI construct for processing, which has information about
  - The process itself
  - What's running in the process



Jobs and Processes Were Enough, Until They Weren't

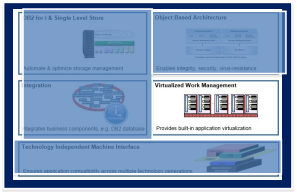
- Job: the user-visible, user definable container of work
- Process: the underlying above-MI construct for processing, which has information about
  - The process itself
  - What's running in the process
- Both of these take extensive time to set up



Jobs and Processes Were Enough, Until They Weren't

- Job: the user-visible, user definable container of work
- Process: the underlying above-MI construct for processing, which has information about
  - The process itself
  - What's running in the process
- Both of these take extensive time to set up

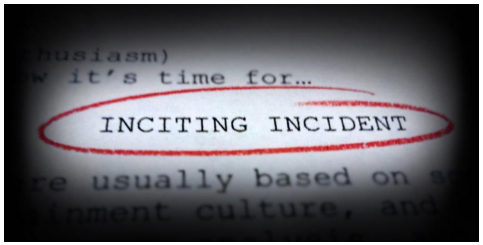
- Applications wanted to kick off work asynchronously and frequently.

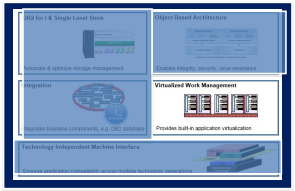


Jobs and Processes Were Enough, Until They Weren't

- Job: the user-visible, user definable container of work
- Process: the underlying above-MI construct for processing, which has information about
  - The process itself
  - What's running in the process
- Both of these take extensive time to set up

- Applications wanted to kick off work asynchronously and frequently.





Applications Drive Adaptation

Applications Drive Adaptation

Applications Drive Adaptation

Applications Drive Adaptation

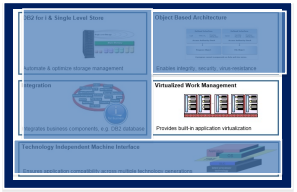
Lotus Notes

Applications Drive Adaptation

Lotus Notes

Applications Drive Adaptation

Lotus Notes



CISC to RISC

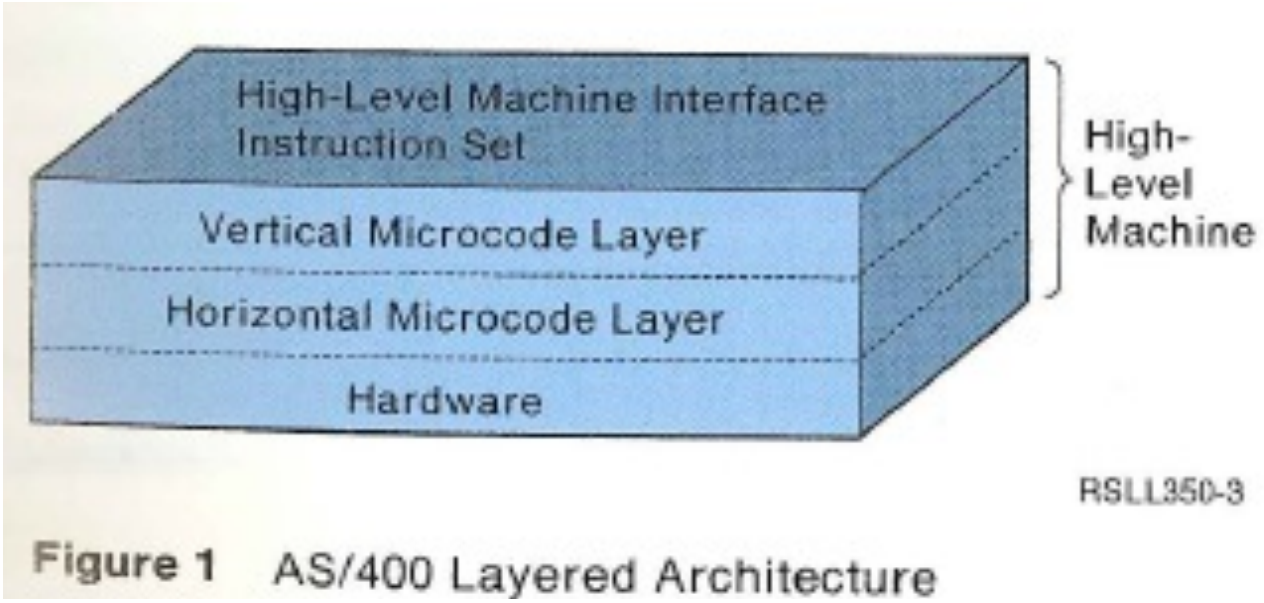
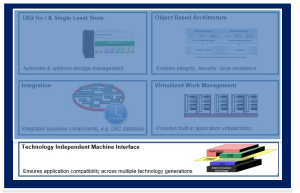
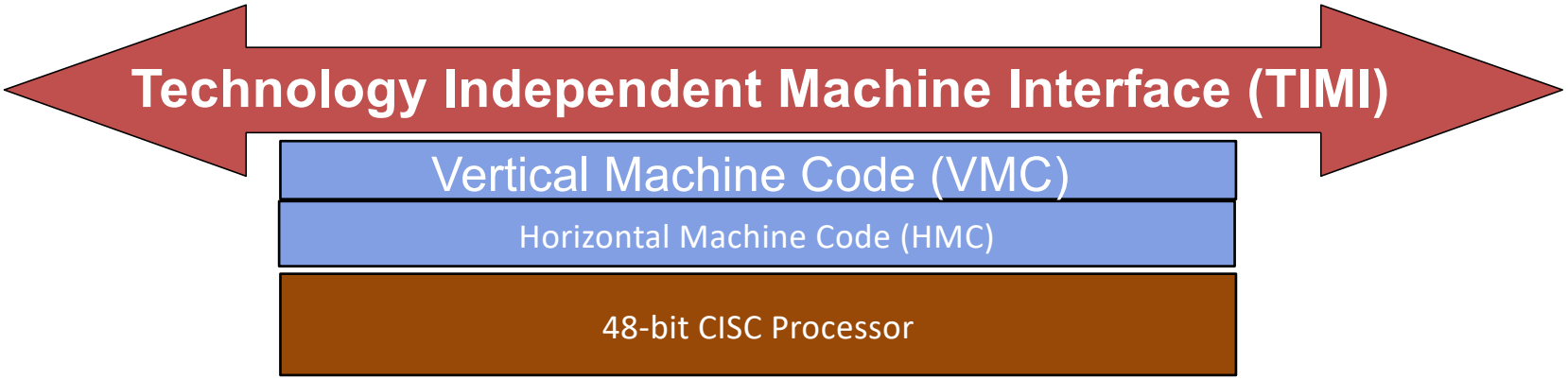


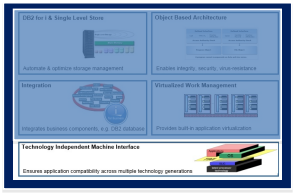
Figure 1 AS/400 Layered Architecture

The original AS/400 hardware was a 48-bit processor, which implemented a “Complex Instruction Set.”

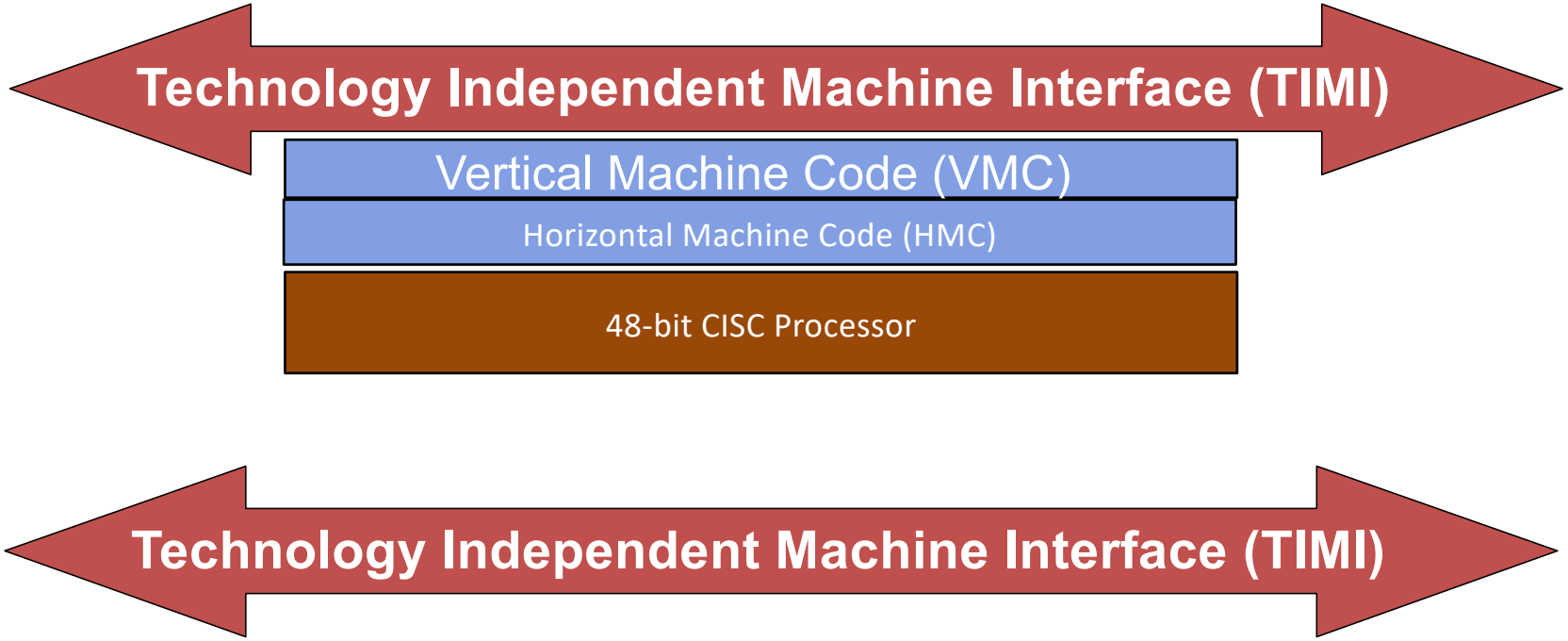


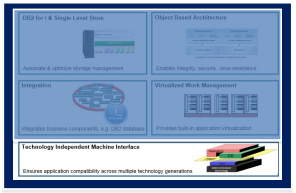
CISC to RISC



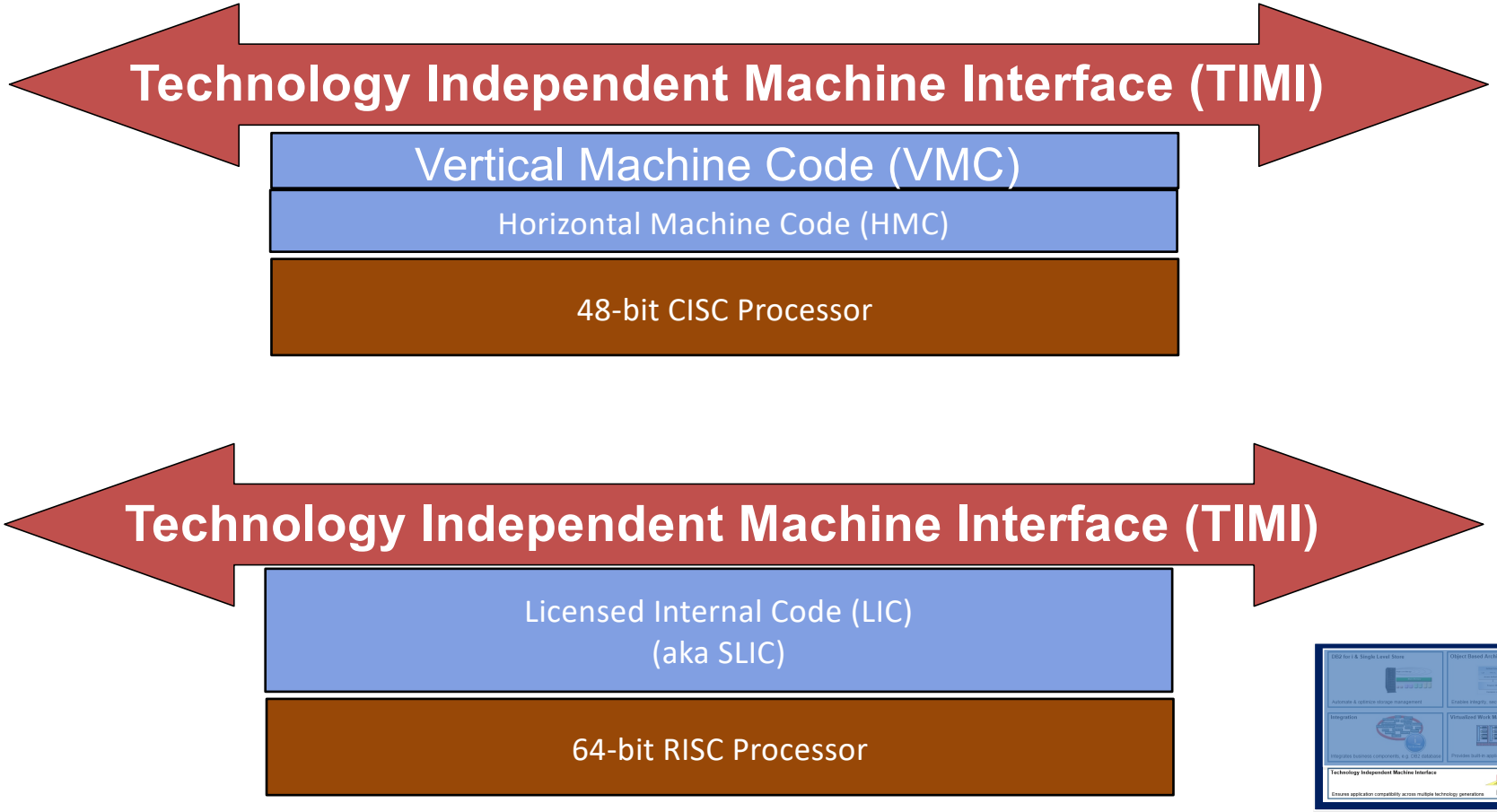


CISC to RISC

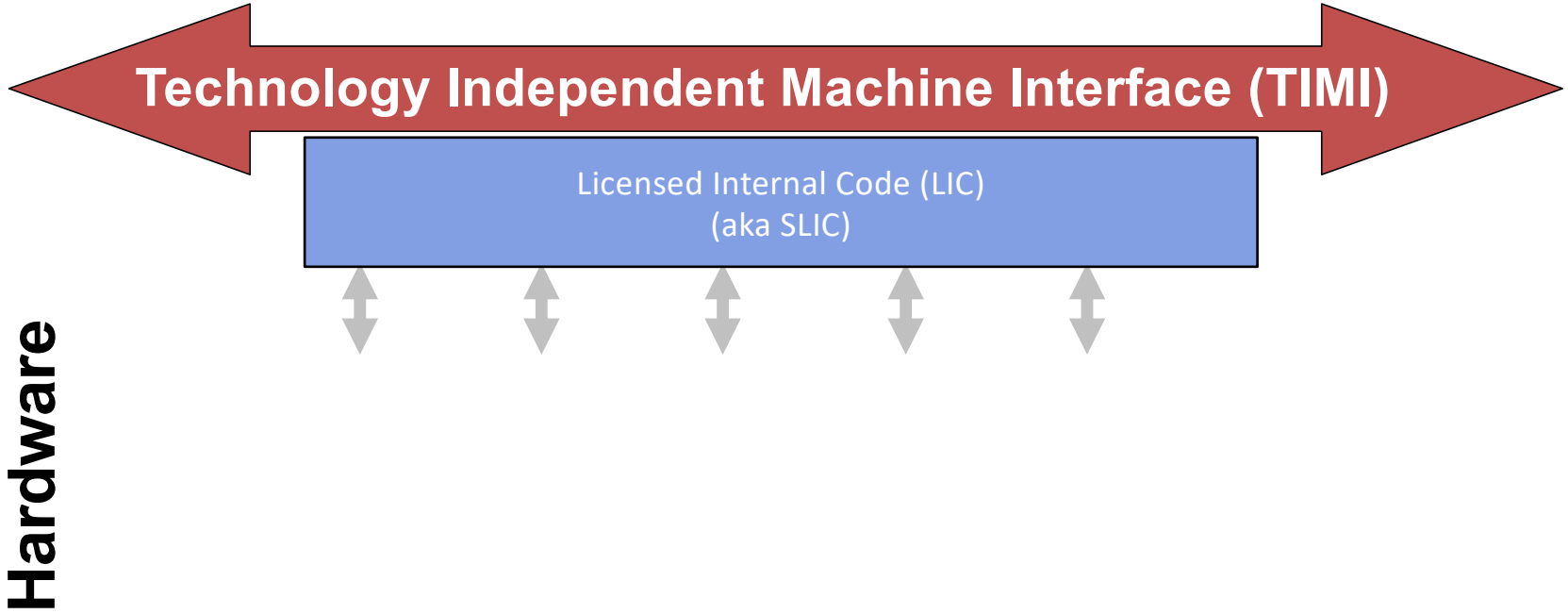




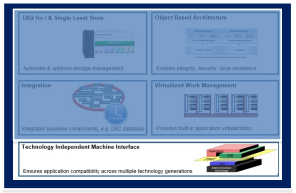
CISC to RISC



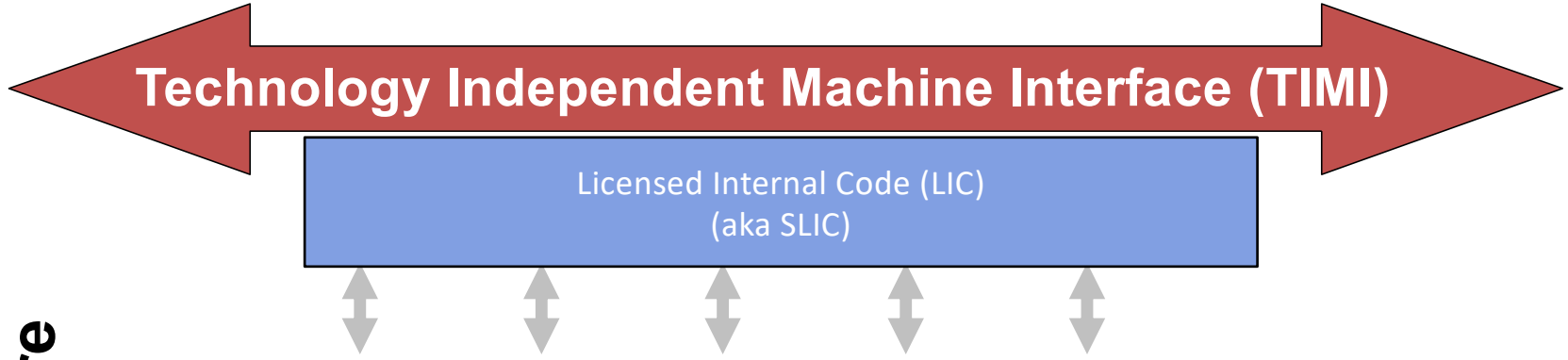
The Architecture – post-CISC



So, did the "Architecture" change?

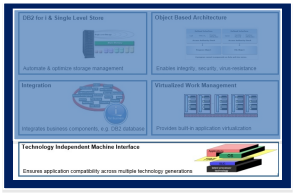


The Architecture – post-CISC

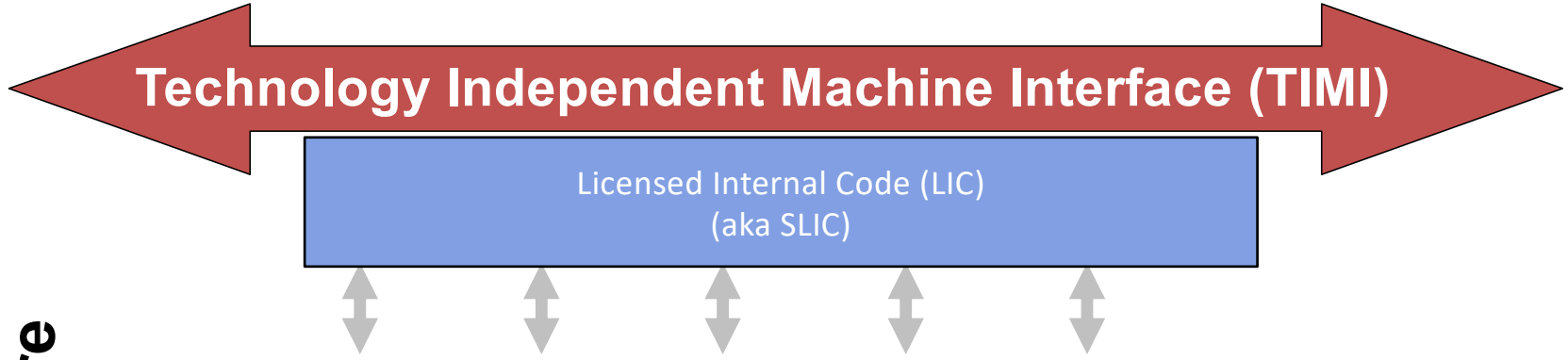


So, did the "Architecture" change?

Above TIMI? Not so much.

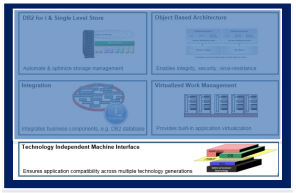


The Architecture – post-CISC

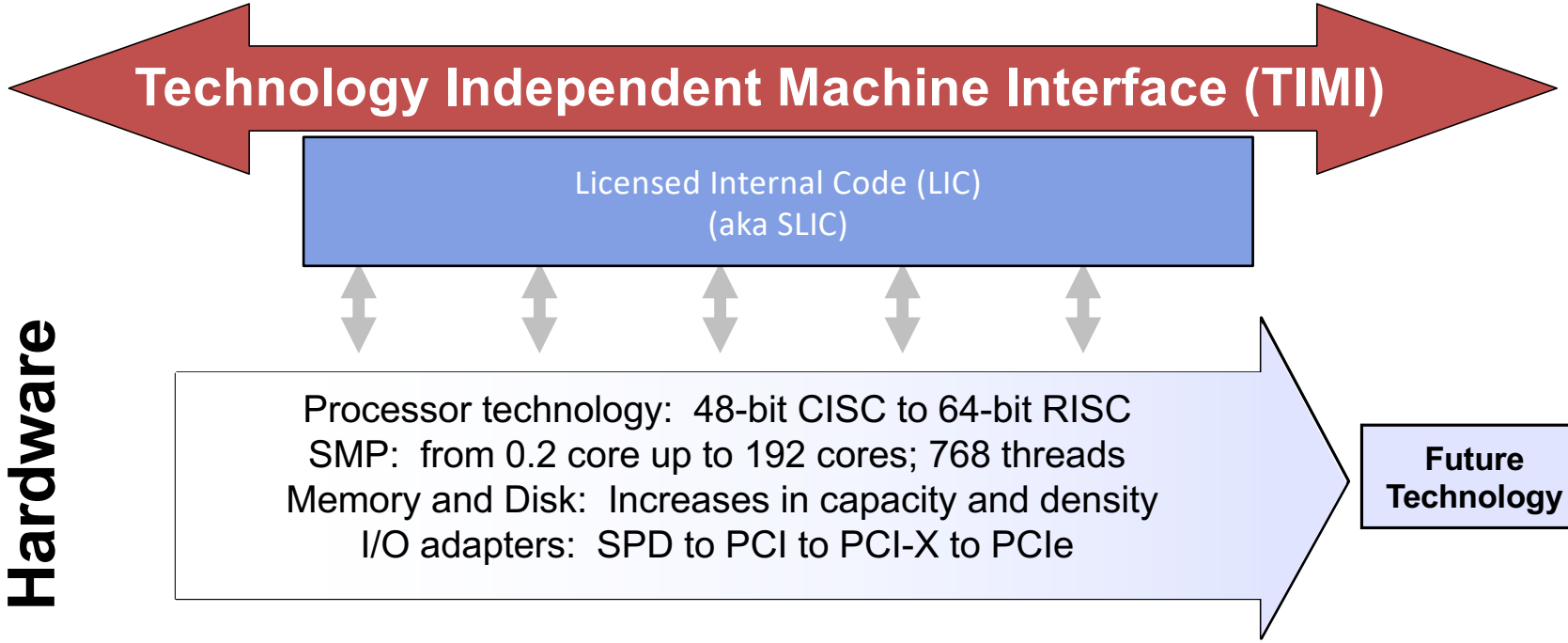


So, did the "Architecture" change?

Above TIMI?	Not so much.
Below TIMI?	You betcha!

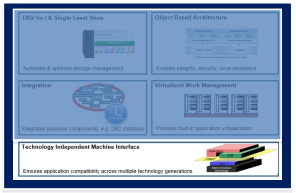


The Architecture – post-CISC

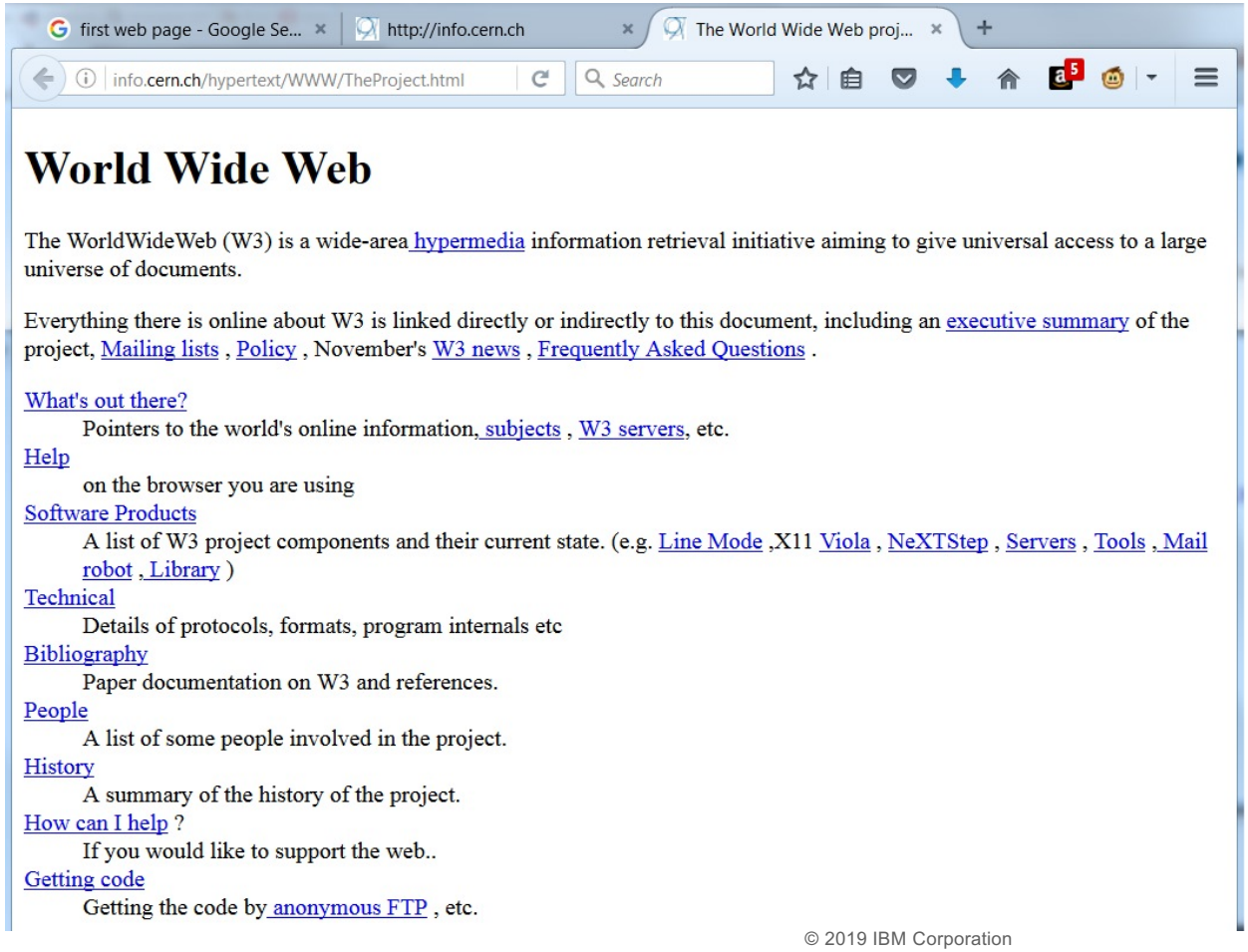


So, did the "Architecture" change?

Above TIMI?	Not so much.
Below TIMI?	You betcha!



And Then Came This …

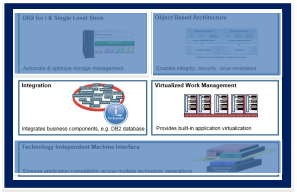


Hypertext Transfer Protocol (HTTP)

The first web page ever!

Web Serving Drove Changes

- HTTP was gaining strength as a presentation protocol
  - With imbedded data retrieval from a web server
- V3R7 – CERN-based web server
  - Single instance only; V4R1 – Multiple servers allowed
- V4R5 – Apache-based web server

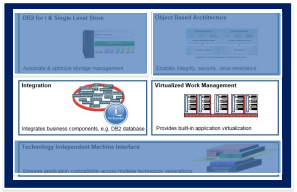


Web Serving Drove Changes

- HTTP was gaining strength as a presentation protocol
  - With imbedded data retrieval from a web server
- V3R7 – CERN-based web server
  - Single instance only; V4R1 – Multiple servers allowed
- V4R5 – Apache-based web server

- Benefited from Multi-threading

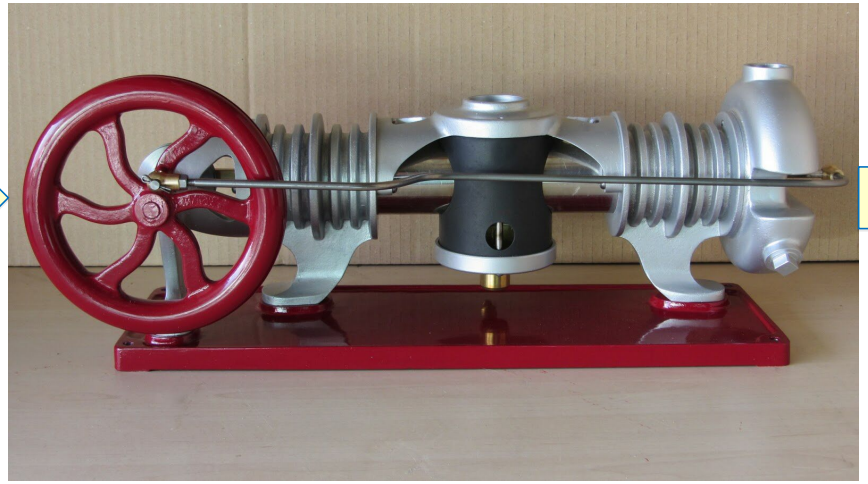
- Drove requirements for
  - Digital Certificates
  - Server Architecture
  - Teraspace



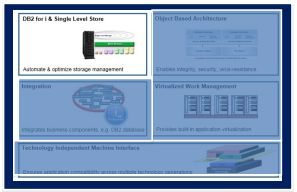
What about the Database?

The Classic Query Engine - CQE

User Question in the Form of a Query



DB2 Answer



DB2 Architects Saw Opportunities

- DB2 Architects wanted a Query Engine which could

DB2 Architects Saw Opportunities

- DB2 Architects wanted a Query Engine which could
  - Take advantage of knowing it was dealing with SQL

SQL

DB2 Architects Saw Opportunities

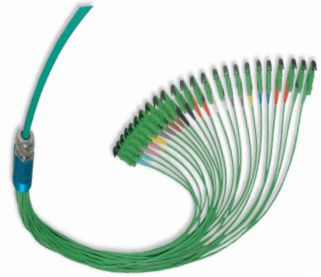
- DB2 Architects wanted a Query Engine which could
  - Take advantage of knowing it was dealing with SQL
  - Learn from the past

SQL

DB2 Architects Saw Opportunities

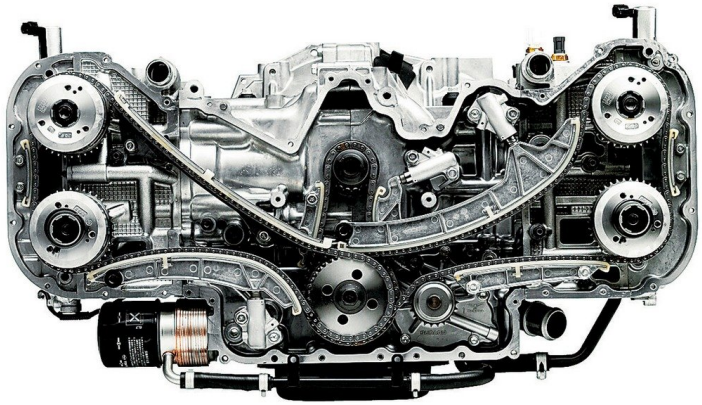
- DB2 Architects wanted a Query Engine which could
  - Take advantage of knowing it was dealing with SQL
  - Learn from the past
  - Use what it learned from queries for one application to improve queries for other applications.

SQL

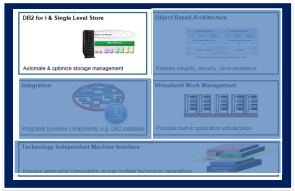


The SQL Query Engine

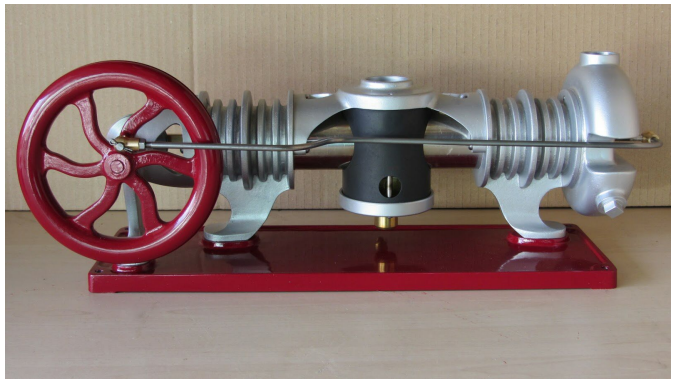
User Question in the Form of an **SQL** Query



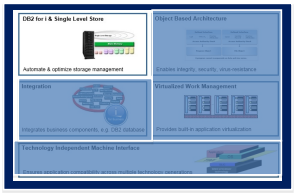
DB2 Answer



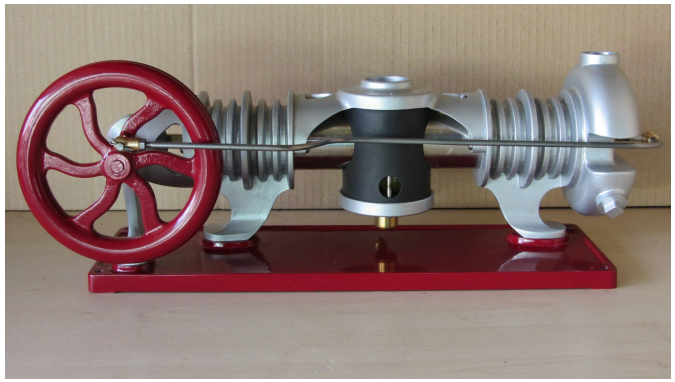
Avoiding Disruption Was Paramount

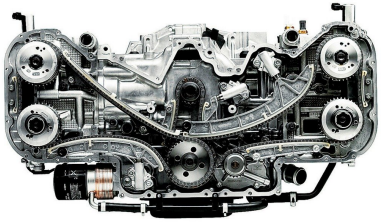


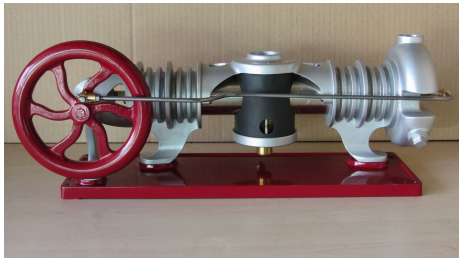
Time



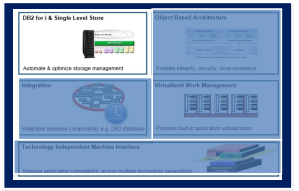
Avoiding Disruption Was Paramount



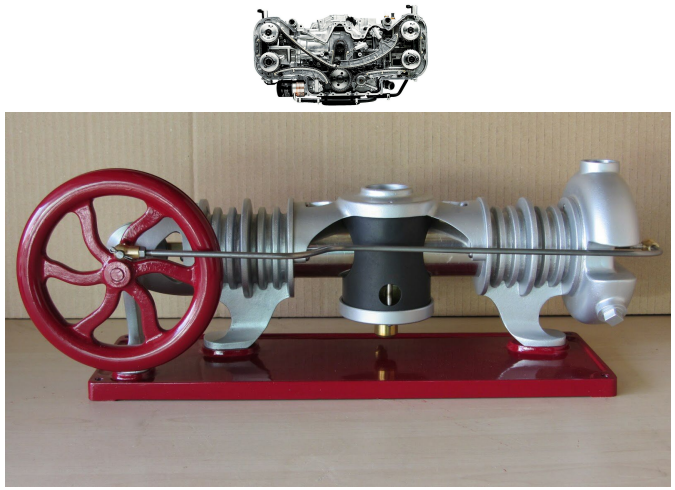


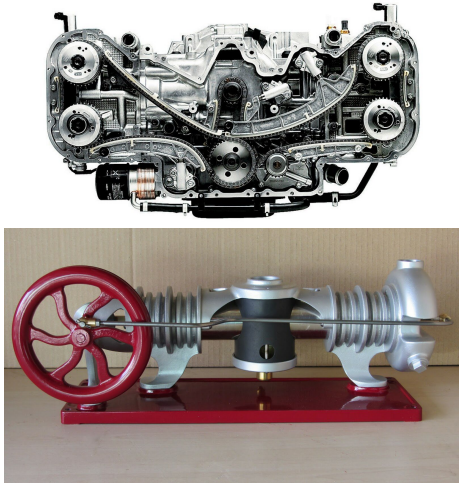


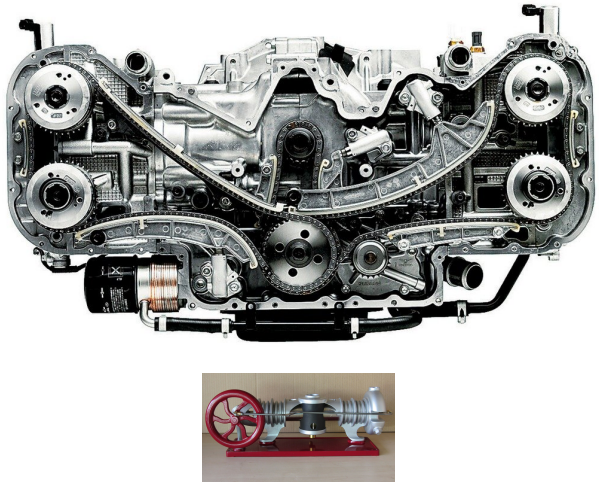
Time



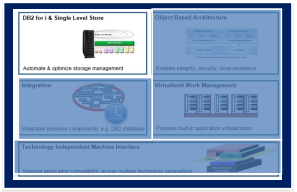
Avoiding Disruption Was Paramount





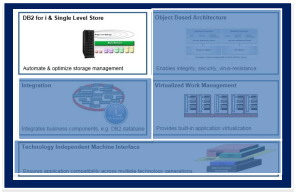


Time



Teraspace: Storage Revolution

- 1 Terabyte (= 2^{40} bytes)
  - flat (non-segmented), process-local storage
  - Temporary
- Lightweight 8-byte pointers available
  - high performance, untagged
- And definitely NOT Single-Level Store!



2008

2000

IBM iSeries

IBM Power Systems

1988

AS/400®

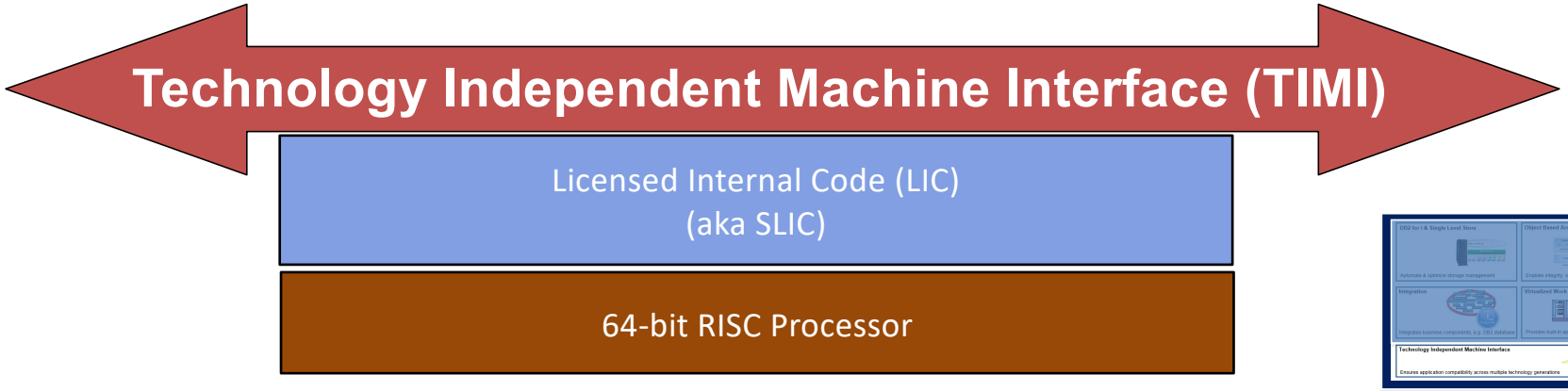
System/38 (1978)
System/36 (1983)

6.1

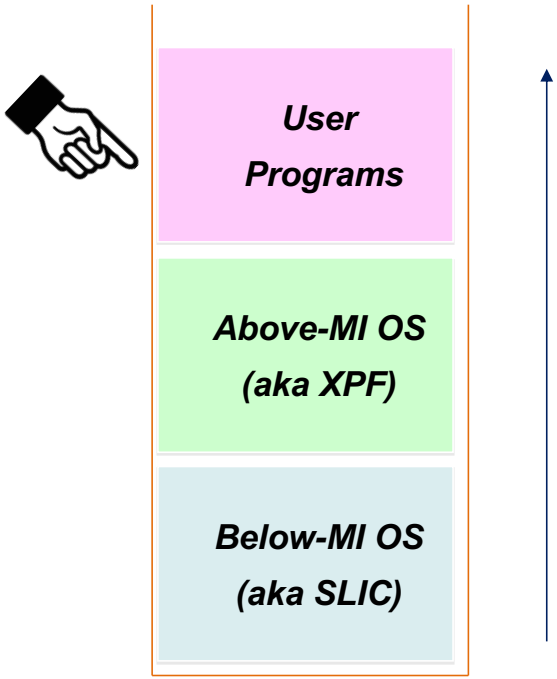
Retranslation at 6.1

In IBM i 6.1, Retranslation occurred to for three purposes:

- Improving security & integrity
- Performance
- Removing limits

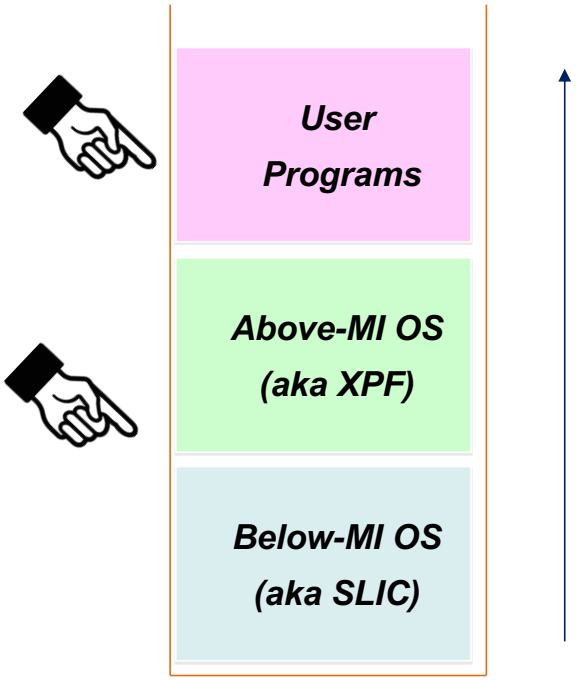


6.1 Retranslation Integrity Part 1: The Execution Stack



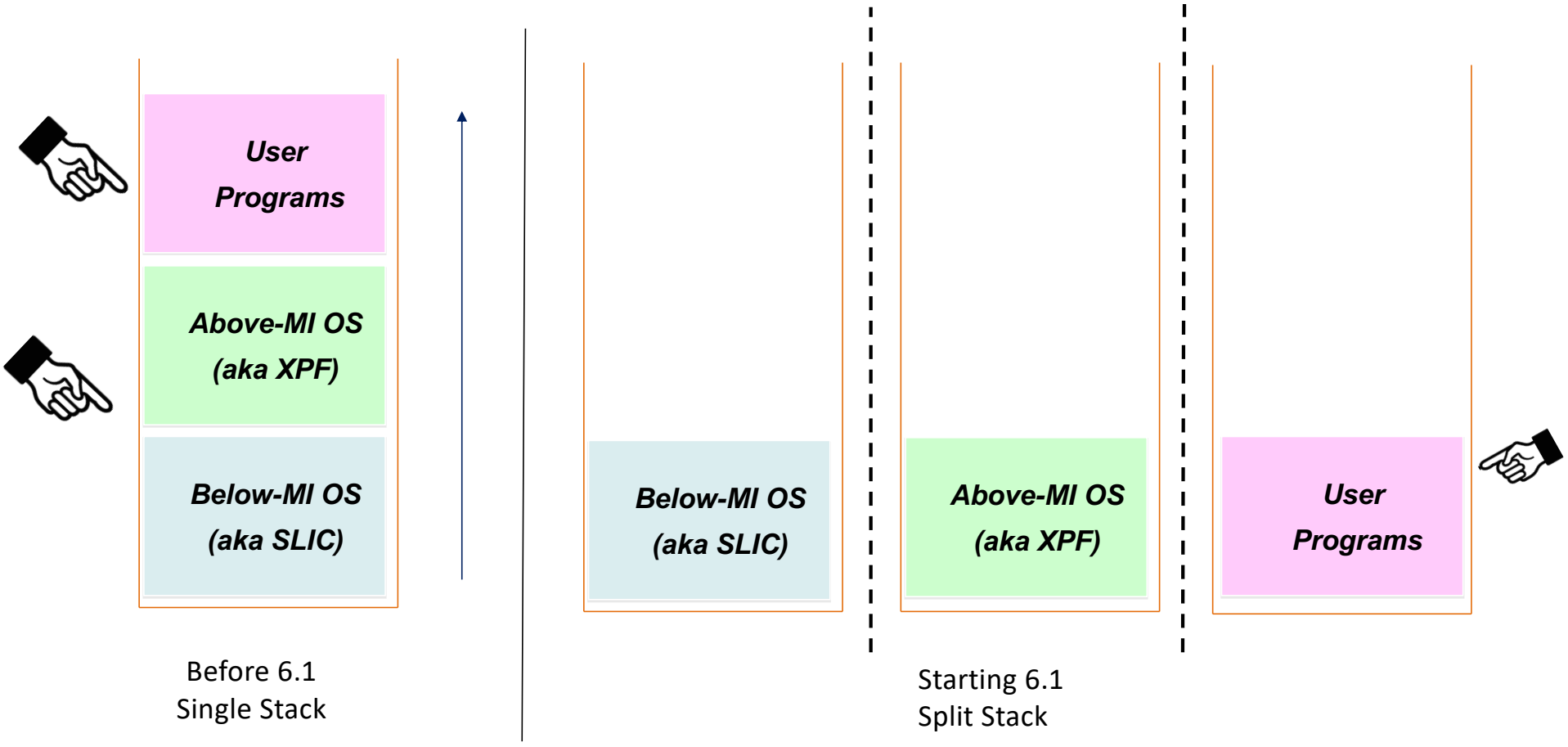
Before 6.1
Single Stack

6.1 Retranslation Integrity Part 1: The Execution Stack



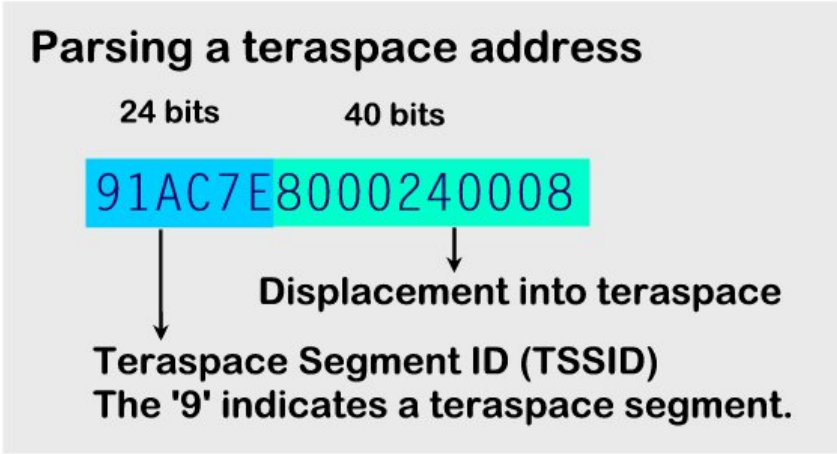
Before 6.1
Single Stack

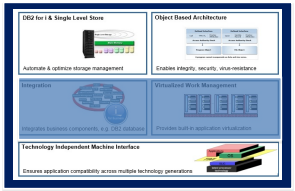
6.1 Retranslation Integrity Part 1: The Execution Stack



Retranslation Integrity & Performance Part 2: Teraspace

- Teraspace initially implemented in software – in SLIC
- Over time, processor support was added to improve use and integrity of addressing.
- By 6.1, all supported machine types had the necessary processor support to do teraspace “right.”
- Retranslation accomplished true “hardware protection” and use of teraspace.





Retranslation Integrity Part 3

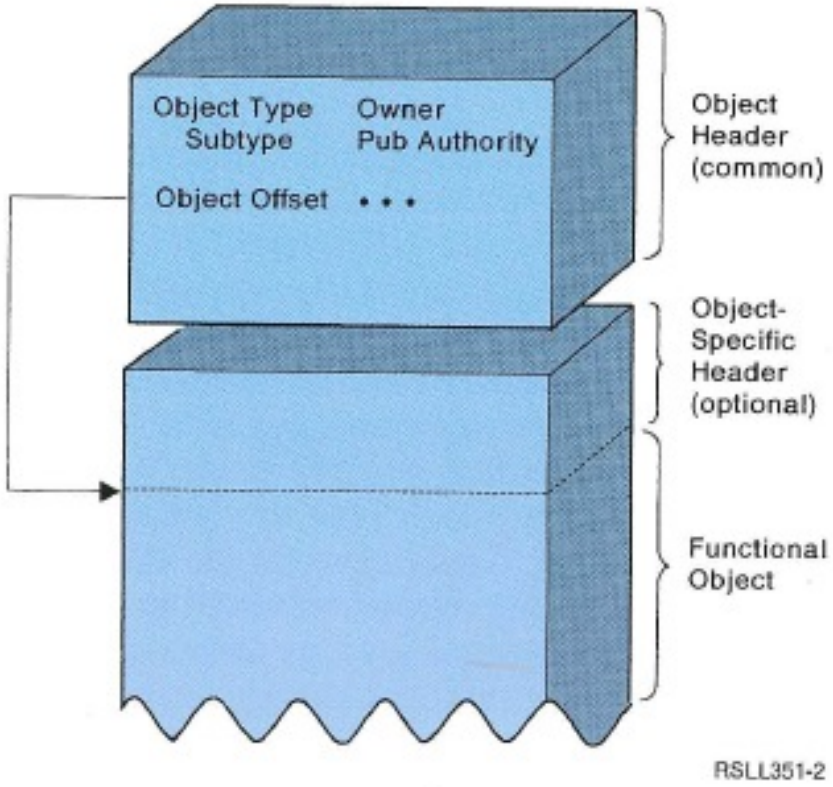
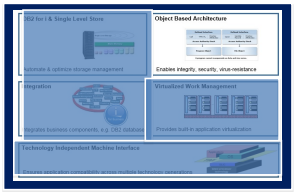


Figure 2 Structure of Generic Object

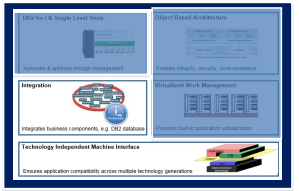
To protect the integrity of objects:

- Privileged/Problem domain
- User/System state
- 6.1: Hardware Protection for accessing privileged domain from user state
- 6.1: Close any potential way to masquerade as system state.



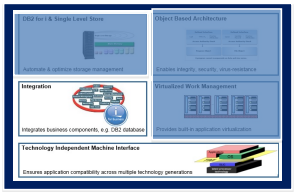
PASE – Portable Application Solutions Environment

- By 2000, AIX and OS/400 were able to run on the same POWER processors.



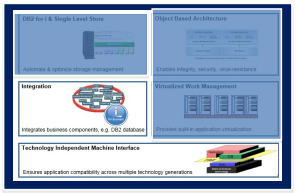
PASE – Portable Application Solutions Environment

- By 2000, AIX and OS/400 were able to run on the same POWER processors.
- This created the possibility for executables which are MI-based and AIX-based to run on the same hardware in the same partition.
- PASE makes it possible for those binaries to run in the same process.

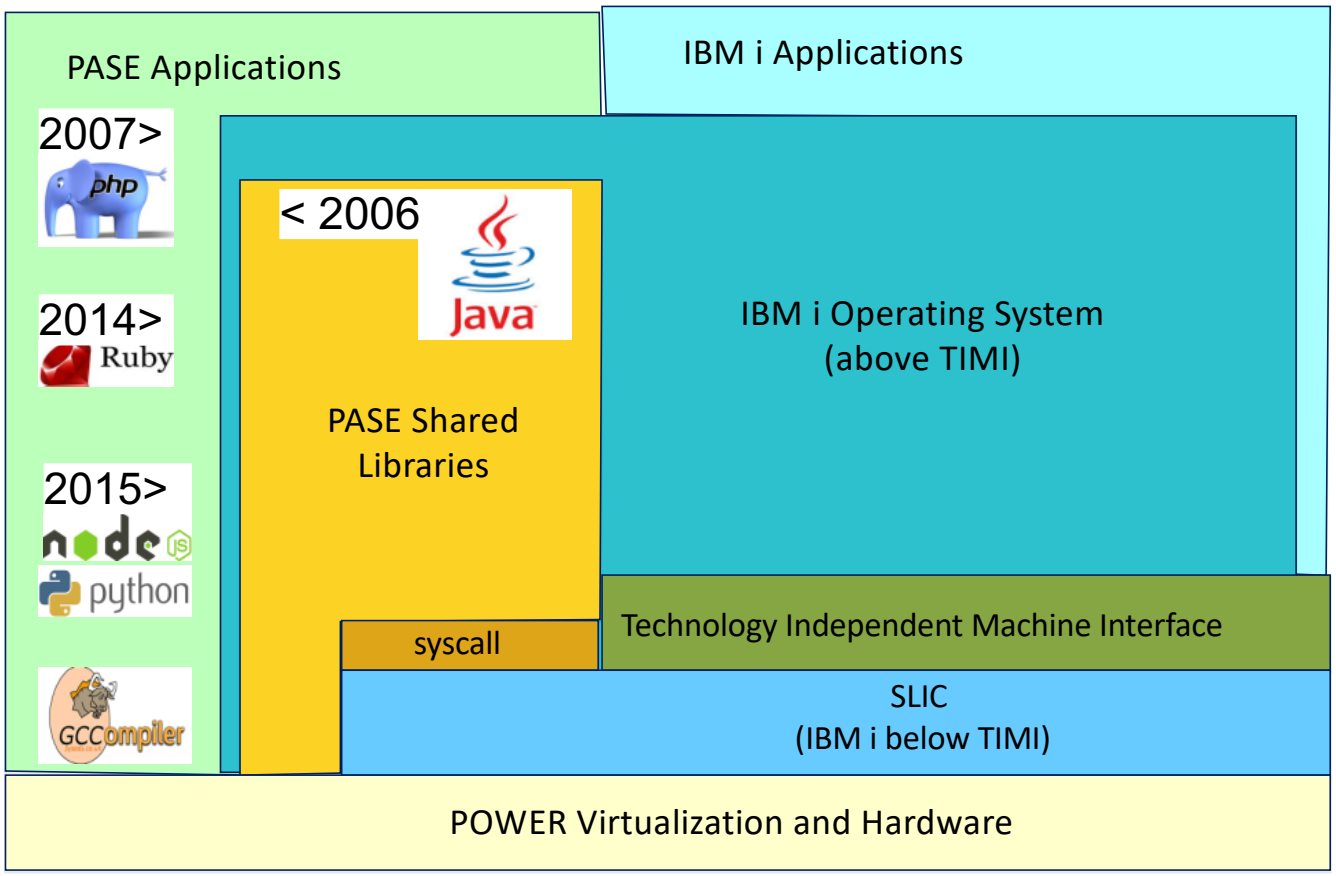


PASE – Portable Application Solutions Environment

- By 2000, AIX and OS/400 were able to run on the same POWER processors.
- This created the possibility for executables which are MI-based and AIX-based to run on the same hardware in the same partition.
- PASE makes it possible for those binaries to run in the same process.
- PASE is a release of AIX
  - Fitted to talk to SLIC rather than directly to the AIX kernel.
- PASE gets the memory from same SLIC teraspace pools used by ILE
  - for program run stack, heap, and shared memory
  - PASE can ONLY see memory that PASE acquired through its own syscall APIs

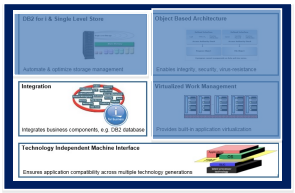


Pictorial View of PASE



PASE allows IBM i to host

- Java
  - SAP
- PHP
- Ruby
- And many other open source options



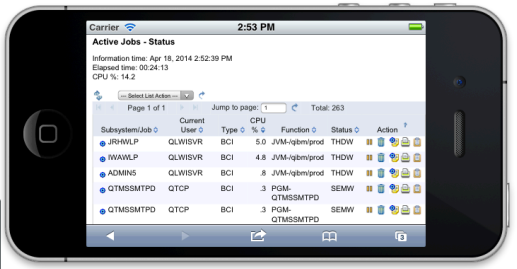
Managing/Accessing the System

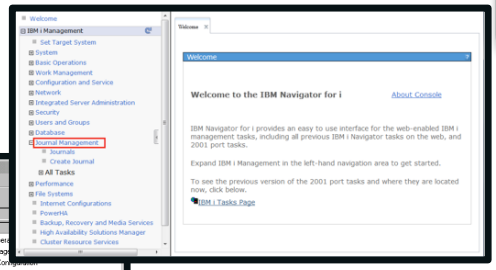
Java & Web +
Server Jobs

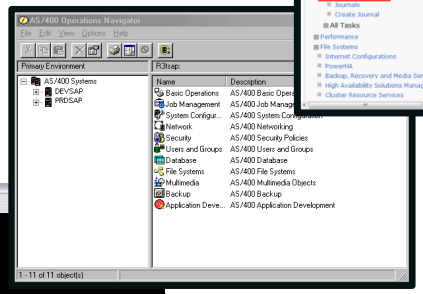
GUI
Emulated
5250

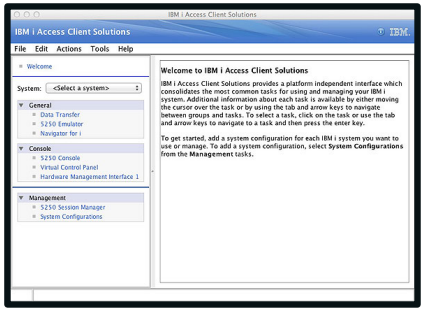
Emulated
5250

5250

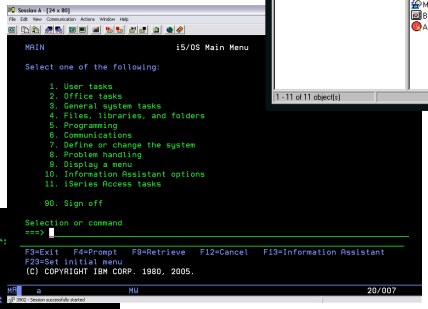


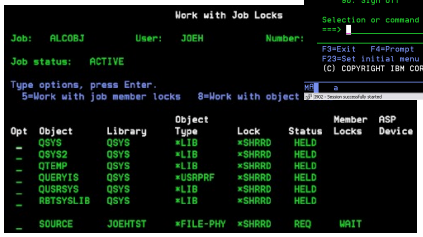


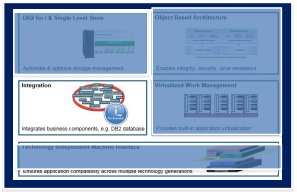




Java, Mobile, BYOD
SQL Services

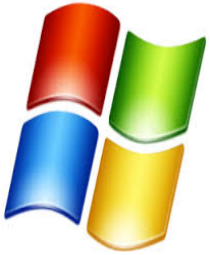


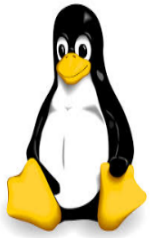




IBM i Access and Management

Limited to Windows Install





Any Device

Some things I'll just mention

- Development Tools
- National Language Support
- Display & Print
- LDAP
- Object Signing
- Enterprise Identity Mapping
- Scaling
- Nodal Affinity
- Independent ASPs
- Logical Partitioning
- N-2 Support
- Technology Refreshes
- Evolution to Waiting Servers
- Integration of Open Source

And things I'll mention for a particular point

- Storage Technology
  - From 520-byte **proprietary,** to 512-byte commodity, to SAN

- I/O
  - During the CISC-to-RISC transition, the move from **IOP-based I/O** to IOA-based I/O was happening

- Networking
  - O. M. G. Does anyone remember **SNA**? Used to be interwoven. I mean, look at this:

AS/400 Integrated Data Communications

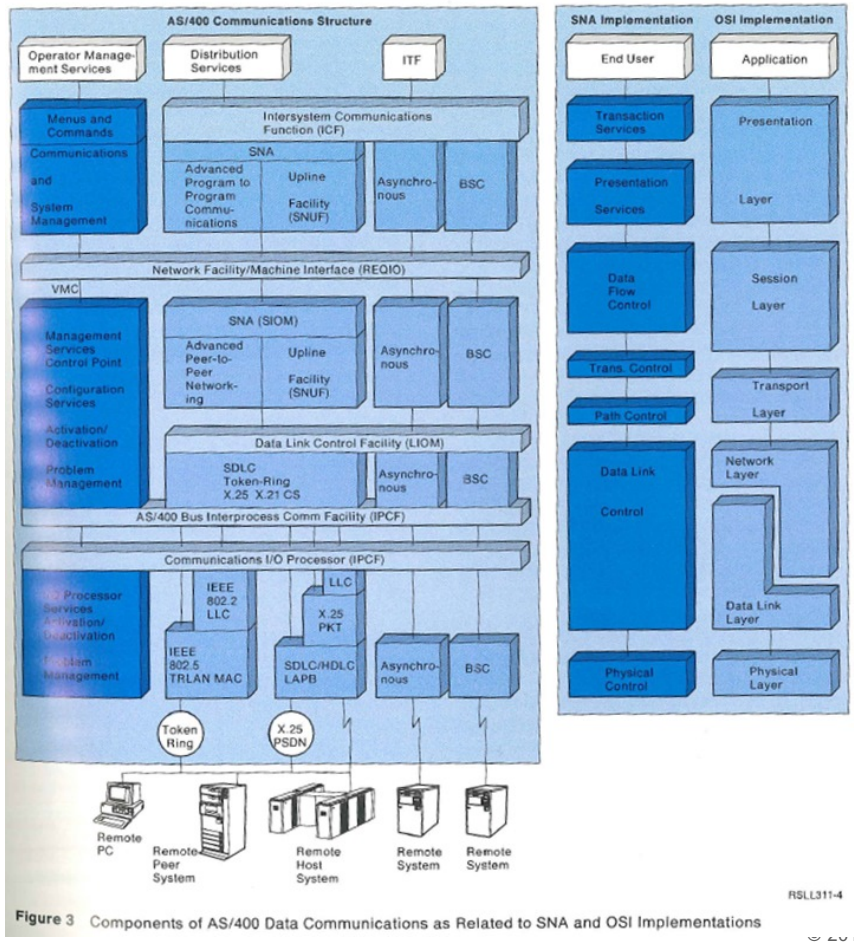
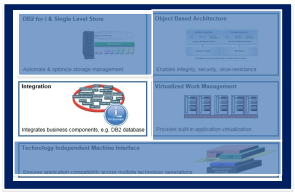
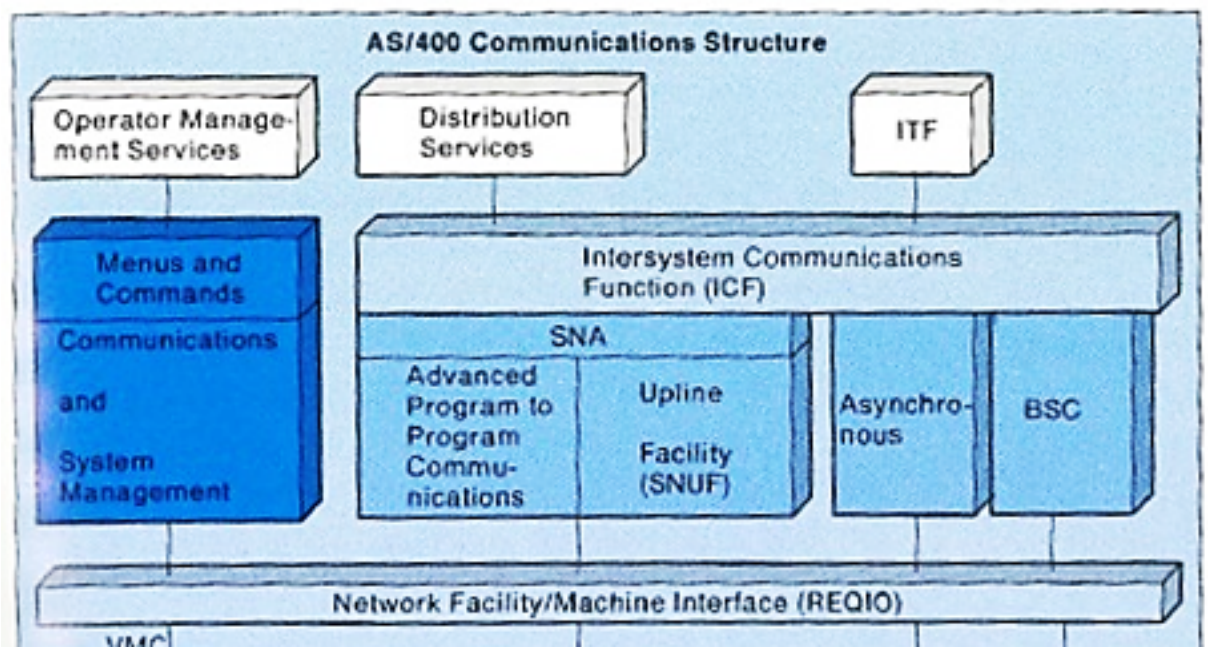


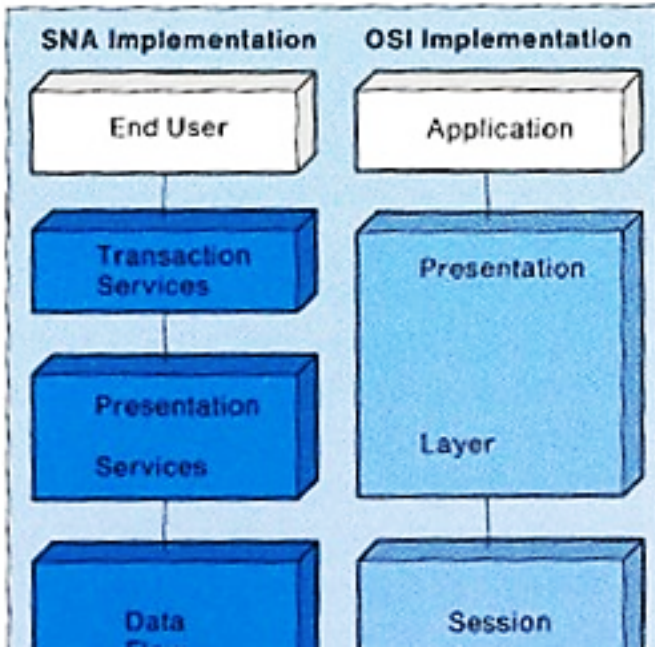
Figure 3 Components of AS/400 Data Communications as Related to SNA and OSI Implementations

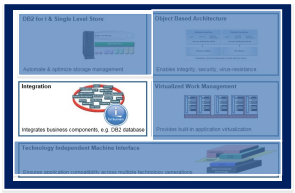
- You can't read that, can you?
- Let's take a closer look at part of it.



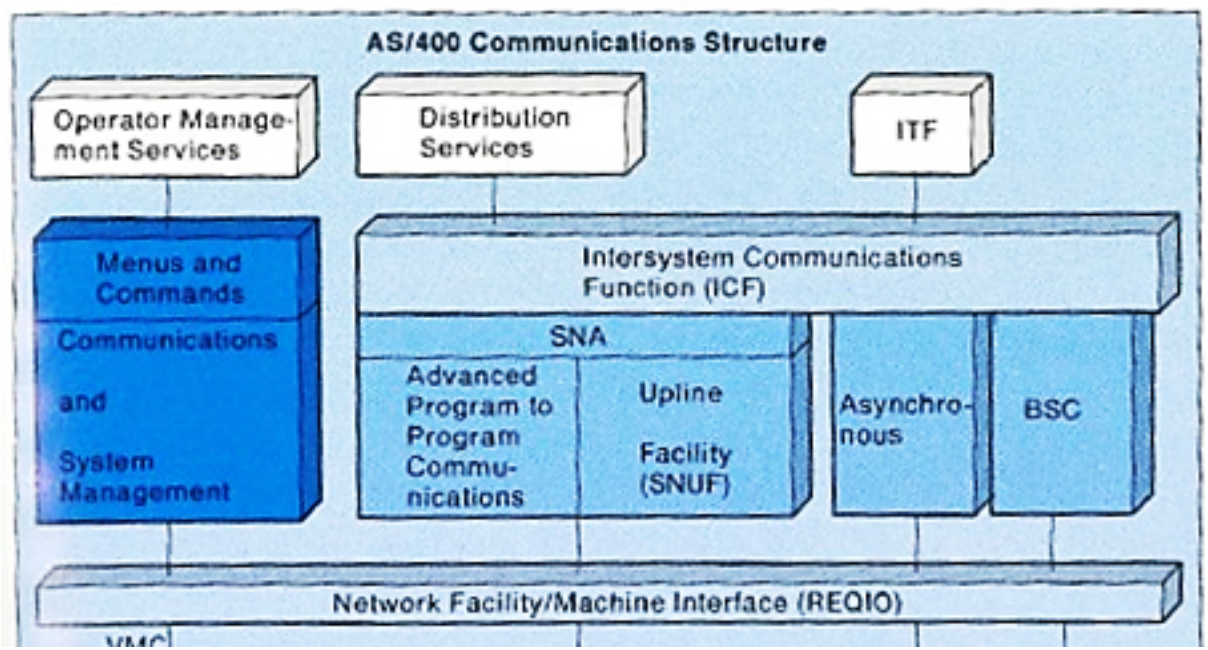
AS/400 Integrated Data Communications

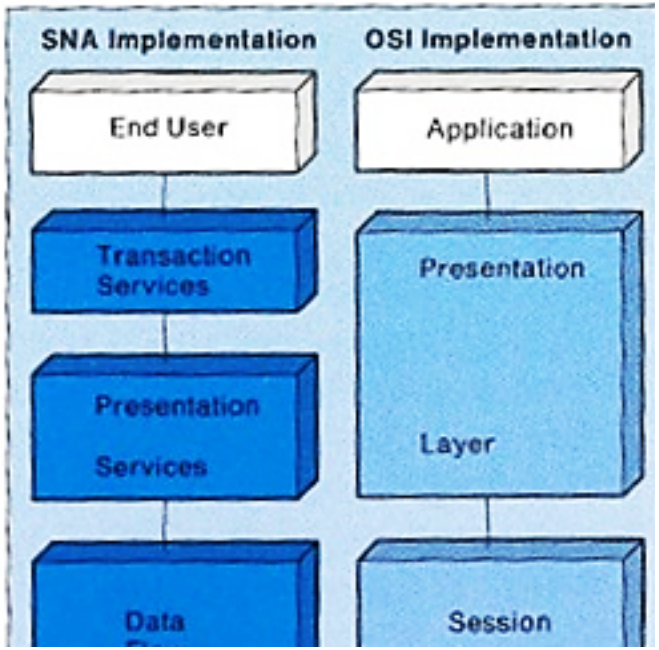




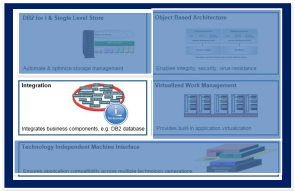


AS/400 Integrated Data Communications





No TCP/IP in sight.
The REQIO TIMI instruction survived.



And things I'll mention for a particular point

- Storage Technology
  - From 520-byte **proprietary,** to 512-byte commodity, to SAN

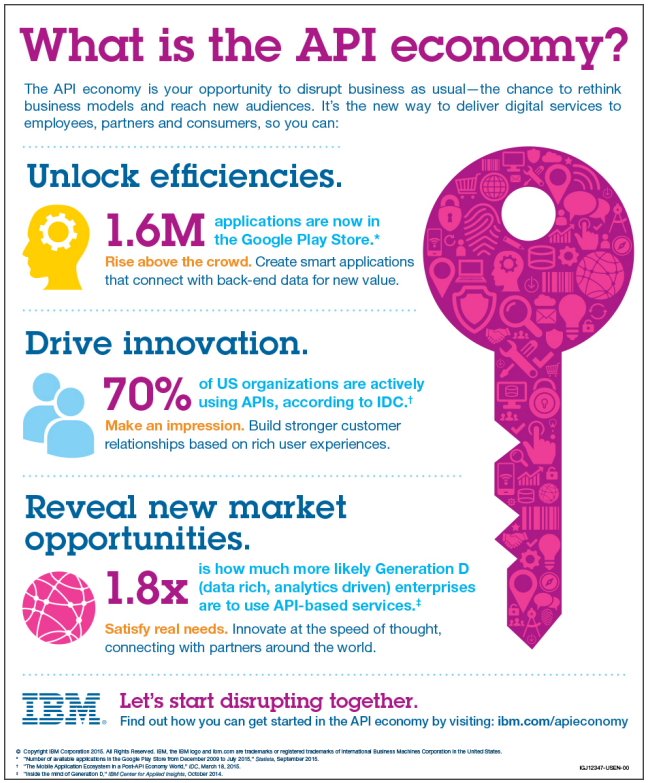
- I/O
  - During the CISC-to-RISC transition, the move from **IOP-based I/O** to IOA-based I/O was happening

- Networking
  - O. M. G. Does anyone remember **SNA**? Used to be interwoven. I mean, look at this:

Major pieces of the architecture of S/38 & AS/400:
Gone or transformed to be nearly unrecognizable.

What now?

Continuous Availability



IBM Db2 Mirror for i

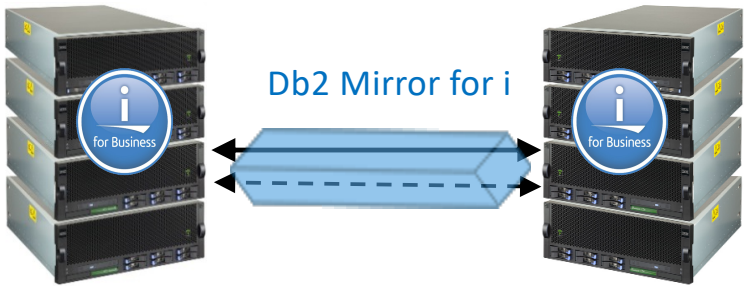
Operating System Synchronous Replication

Continuous Availability

24 x 7 Up Time

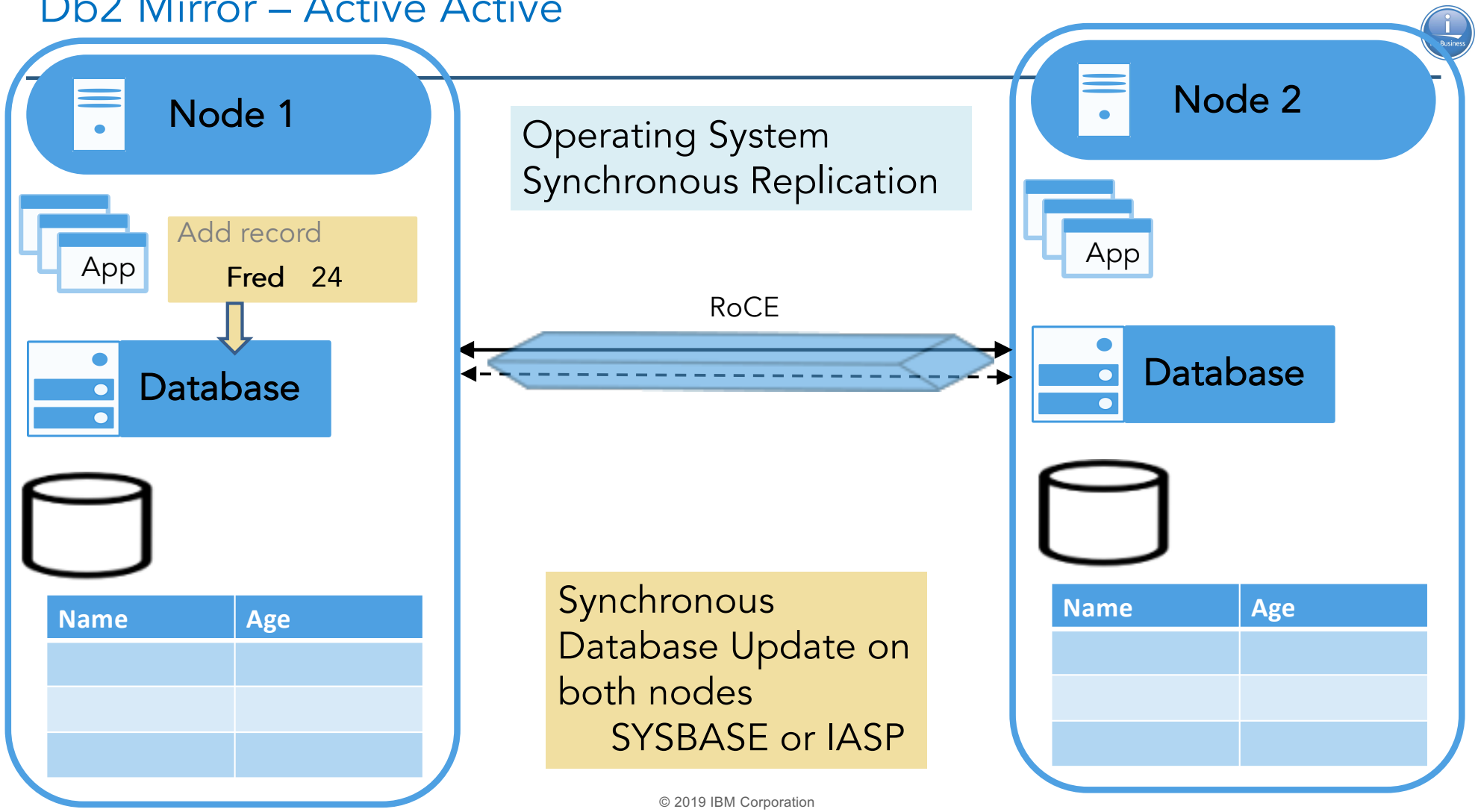
Rolling Upgrades

RTO/RPO Near Zero



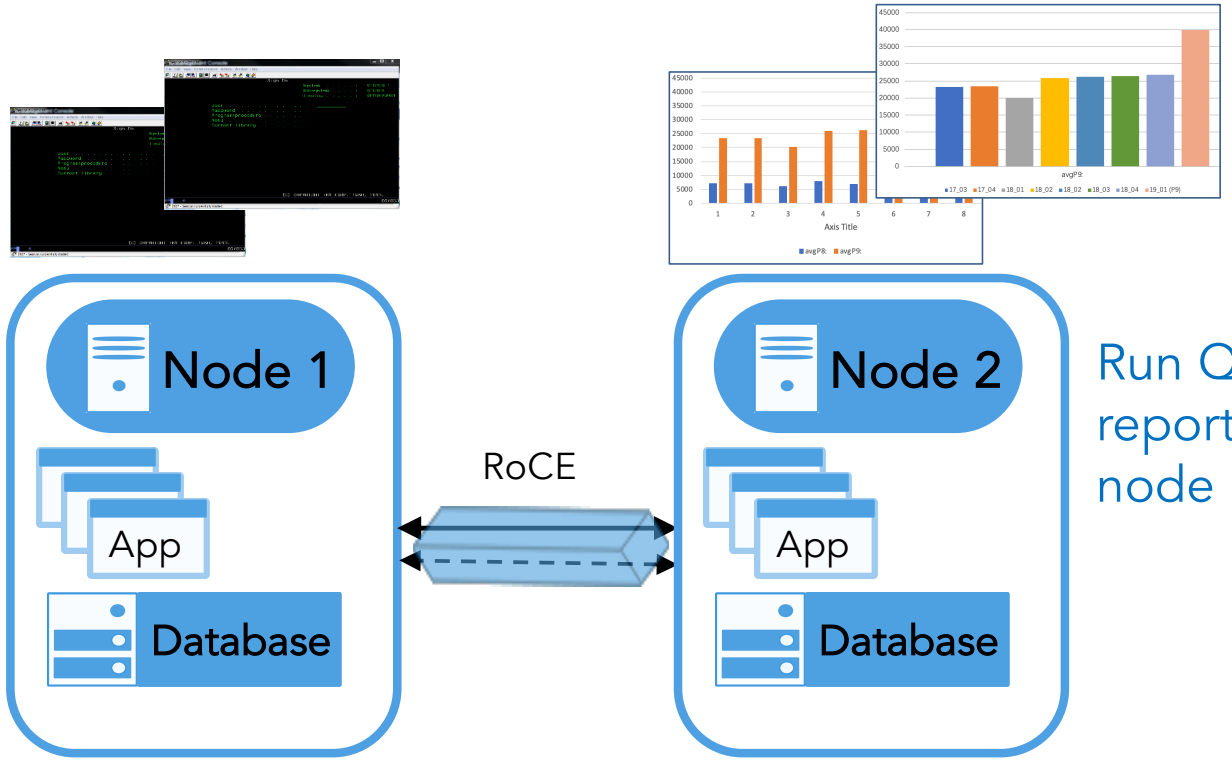
POWER8 or later & IBM i 7.4

Db2 Mirror – Active Active



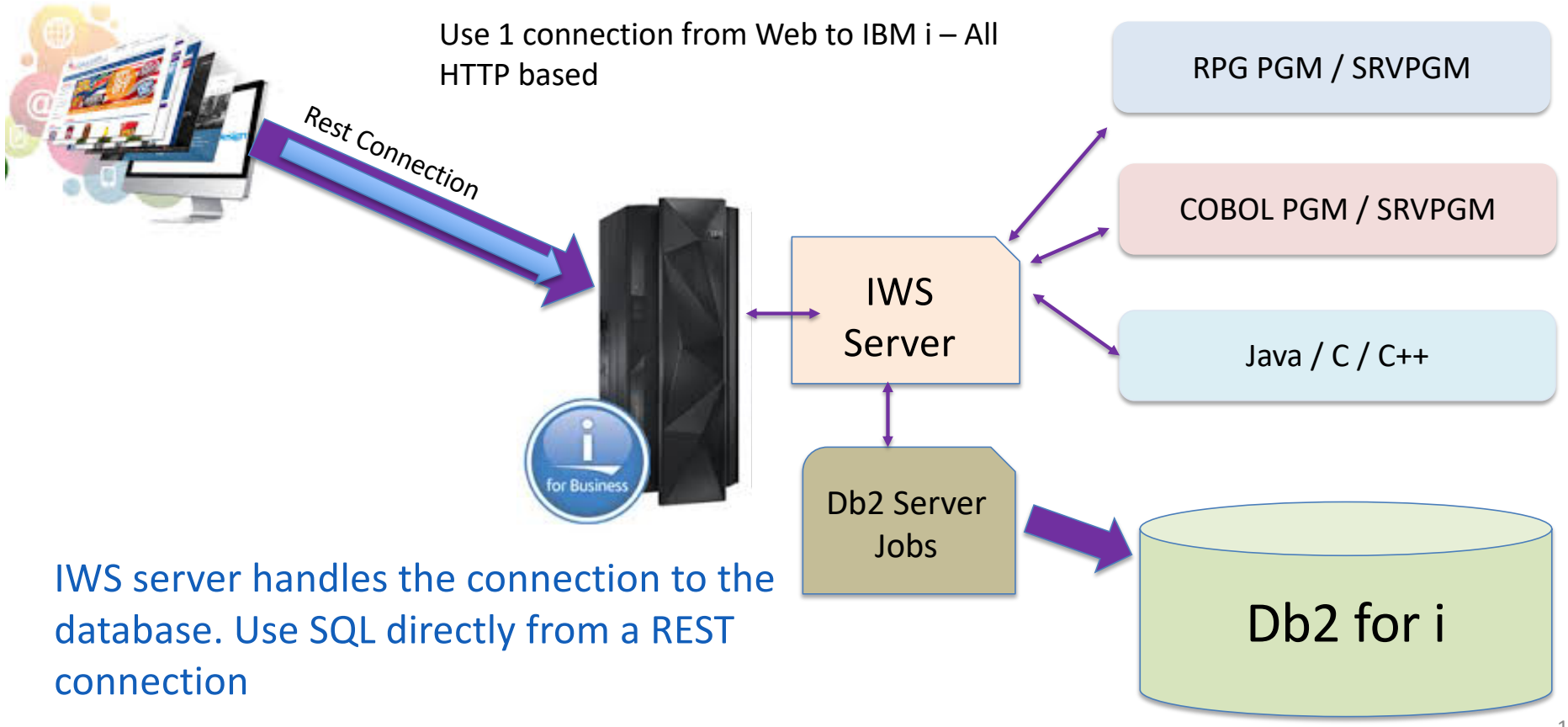
Db2 Mirror – Active Passive

Run Production Workloads on this node

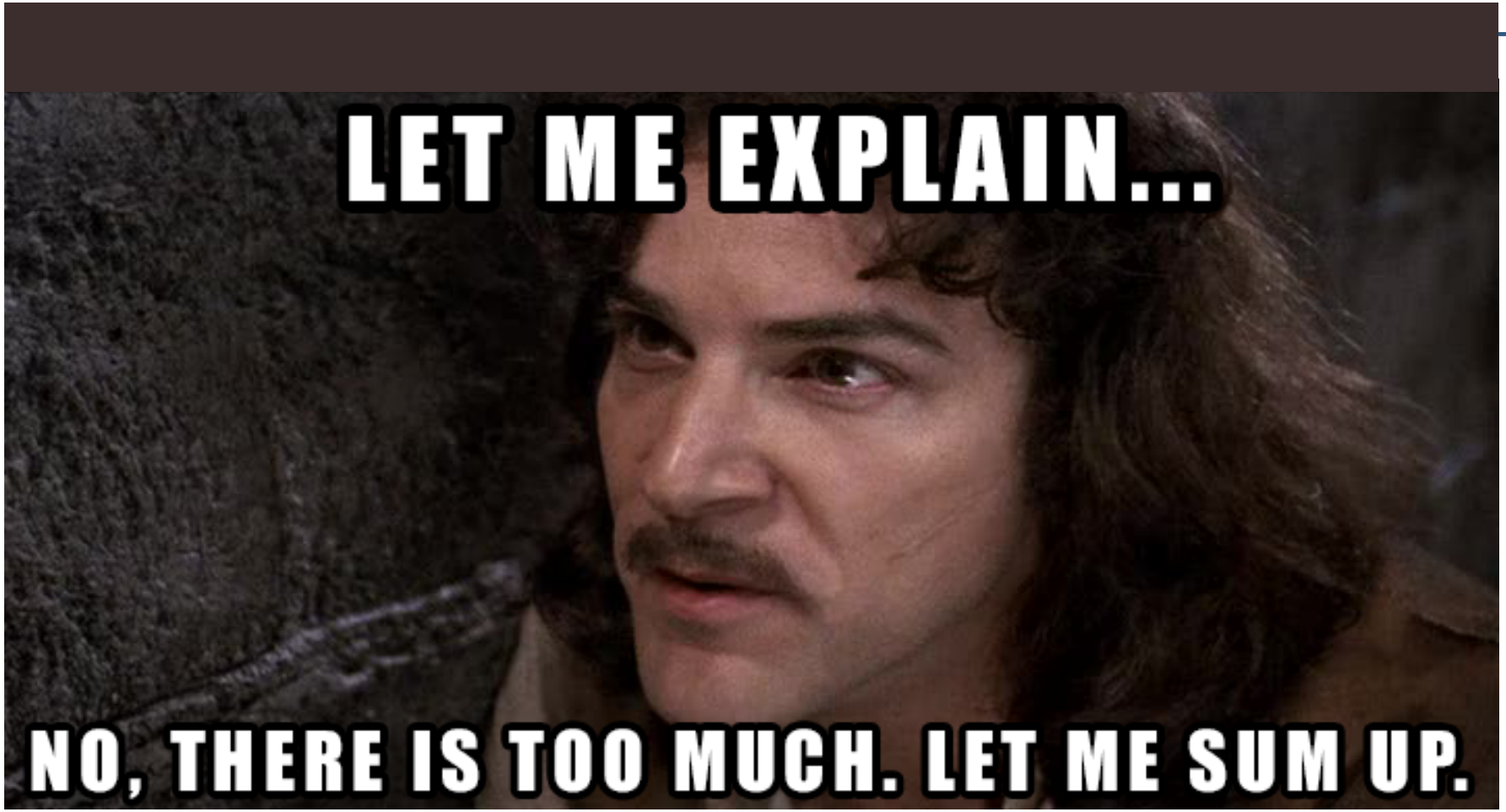


Run Queries and reports on this node

IBM i in the “Services” Era



IWS server handles the connection to the database. Use SQL directly from a REST connection



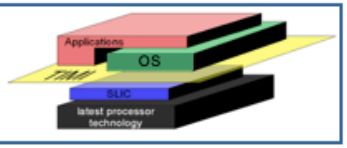
DB2 for i & Single Level Storage

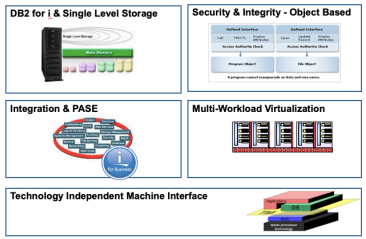
Security & Integrity - Object Based

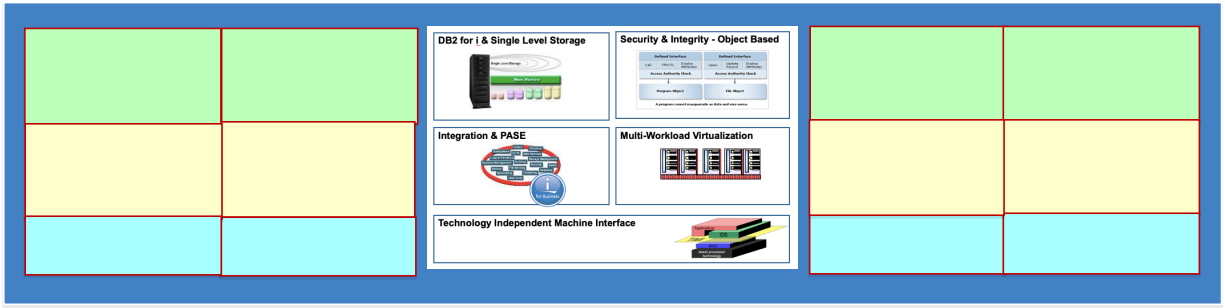
Integration & PASE

Multi-Workload Virtualization

Technology Independent Machine Interface







IBM i - "Technology will change and IBM i is built to change with it"

Link

ithankyou