



Your Business ■ Your Agenda ■ Our Passion
Maximizing the value of IT together

Rocket Software

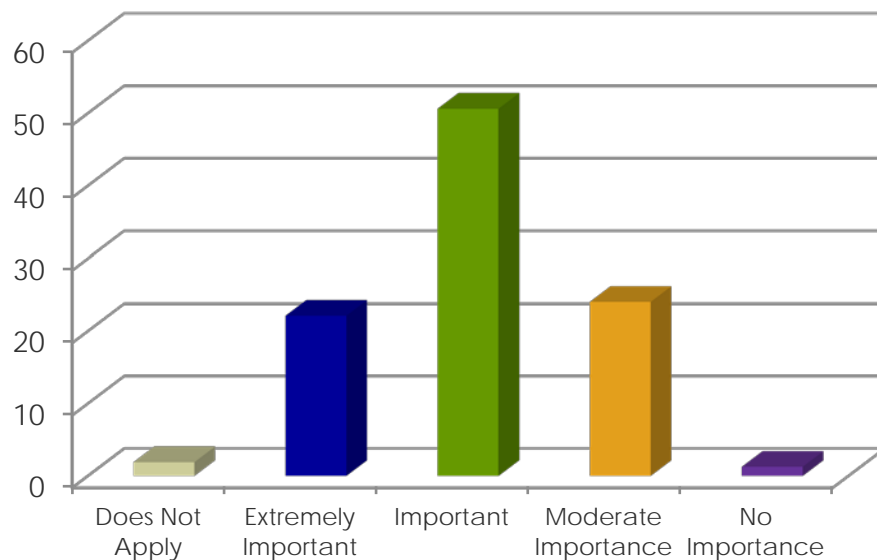
Putting your IMS/DBs into
Business Analytics / Intelligence

Virtual IMS User Group

Analytics and Mainframe Data



- How important is it for your organization to integrate non-relational data with a business analytics or business intelligence initiative?

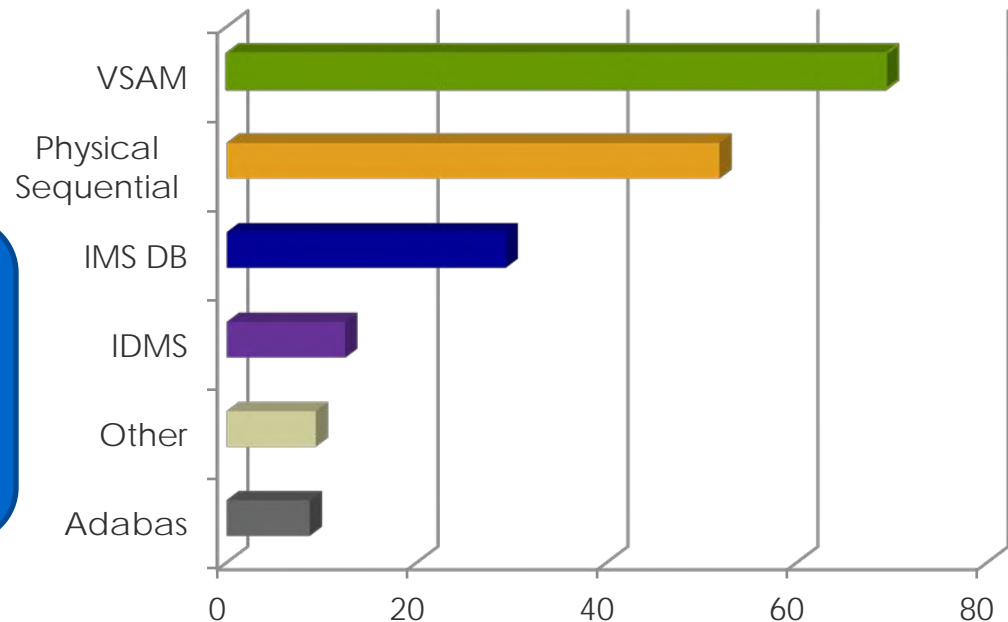


more than 70% signified high level of importance for integrating non-relational data with analytics

Analytics & Non-relational Data



- Which non-relational data source will need to be exposed to your business analytics or business intelligence application?



69% of respondents listed VSAM as the most prevalent non-relational data source needed for analytics

Data is Growing Exponentially



Web and Social Media



Consumer Data



Mobile data



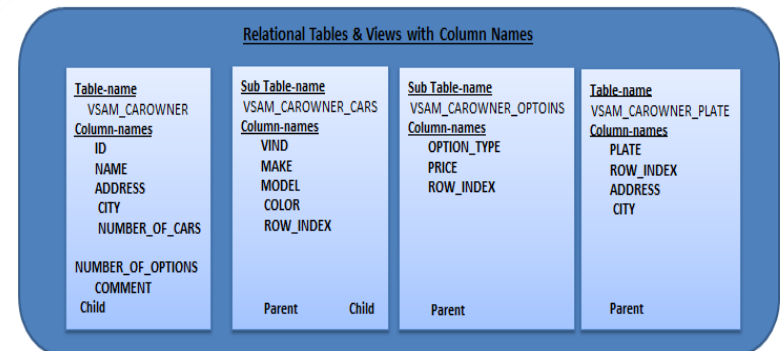
Logistics Data



Problem – Data Incompatibility

- Structure of non-relational data (VSAM, IMS, Adabas) isn't compatible with analytics or BI tools

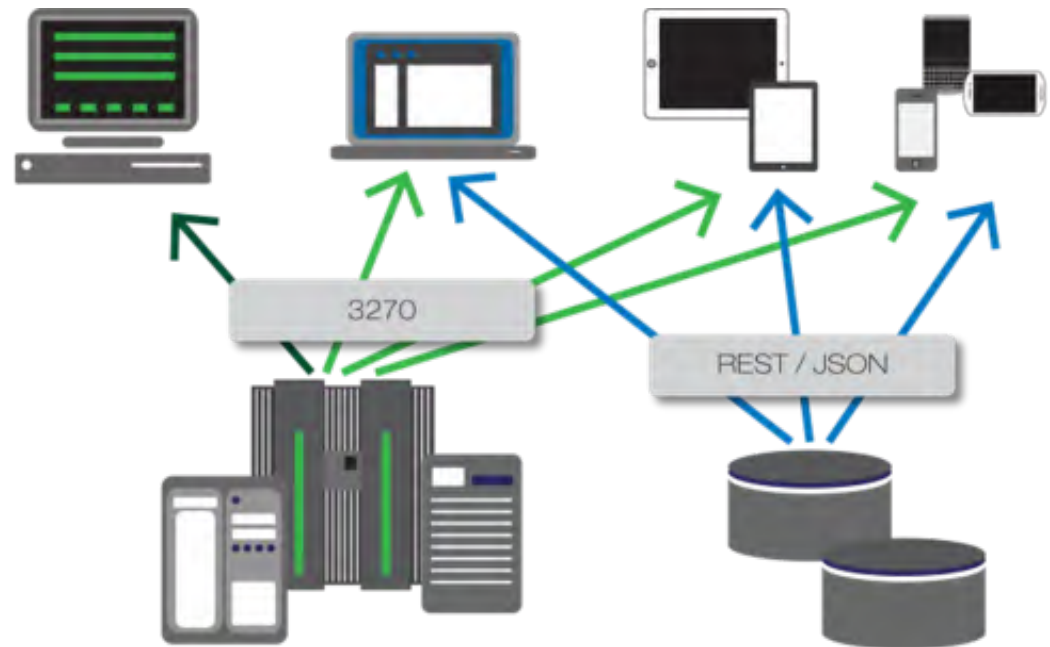
```
01 CAROWNER.  
 05 ID          PIC $9(08) COMP.  
 05 NAME        PIC X(18).  
 05 ADDRESS     PIC X(19).  
 05 CITY        PIC X(15).  
 05 NUMBER-OF-CARS PIC $9(08) COMP.  
 05 NUMBER-OF-OPTIONS PIC 9(4) COMP.  
 05 CARS OCCURS 0 TO 3 TIMES DEPENDING ON NUMBER-OF-CARS.  
 10 VIN         PIC X(13).  
 10 PLATE OCCURS 10 TIMES PIC X(1).  
 10 MAKE        PIC X(10).  
 10 MODEL       PIC X(10).  
 10 COLOR       PIC X(10).  
 10 OPTIONS OCCURS 0 TO 2 TIMES DEPENDING ON NUMBER-OF-OPTIONS.  
 15 OPTION-TYPE PIC X(10).  
 15 PRICE       PIC $9(5)V9(2) USAGE COMP-3.  
 05 COMMENTS   PIC X(50).
```



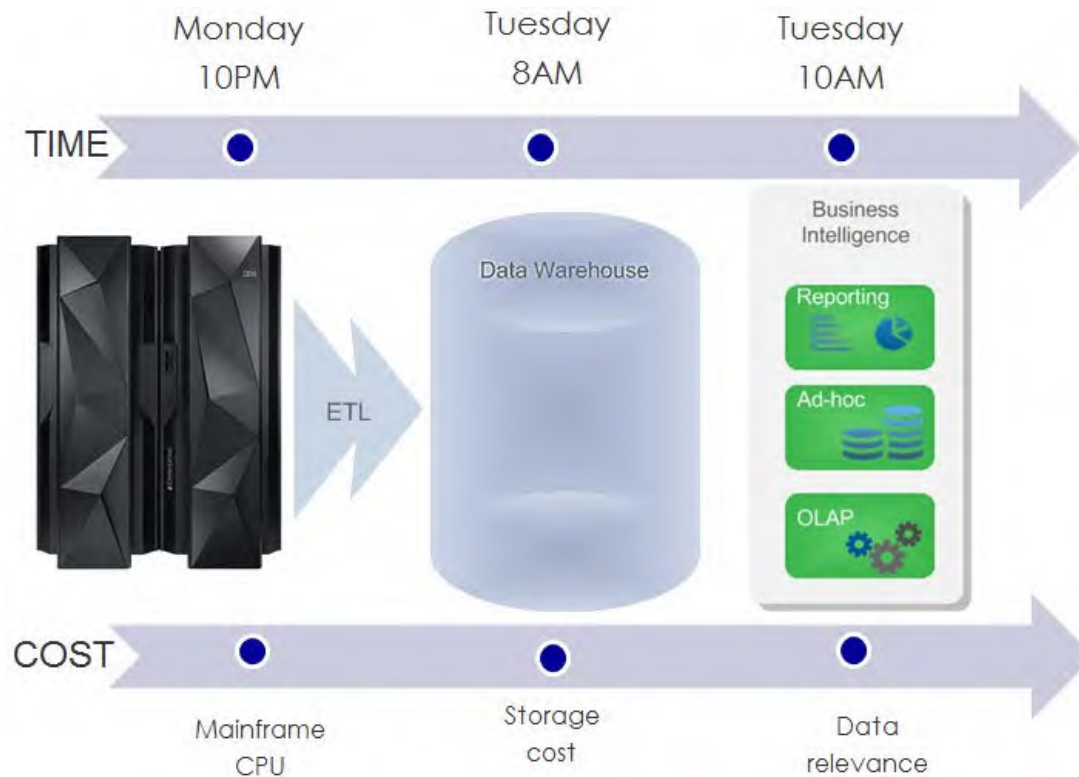
Unique Mainframe Challenges



- Programming skills in COBOL or PL/1 are in short supply
- Limited ability to code or manually manipulate data places business critical information at risk



Unique Mainframe Challenges



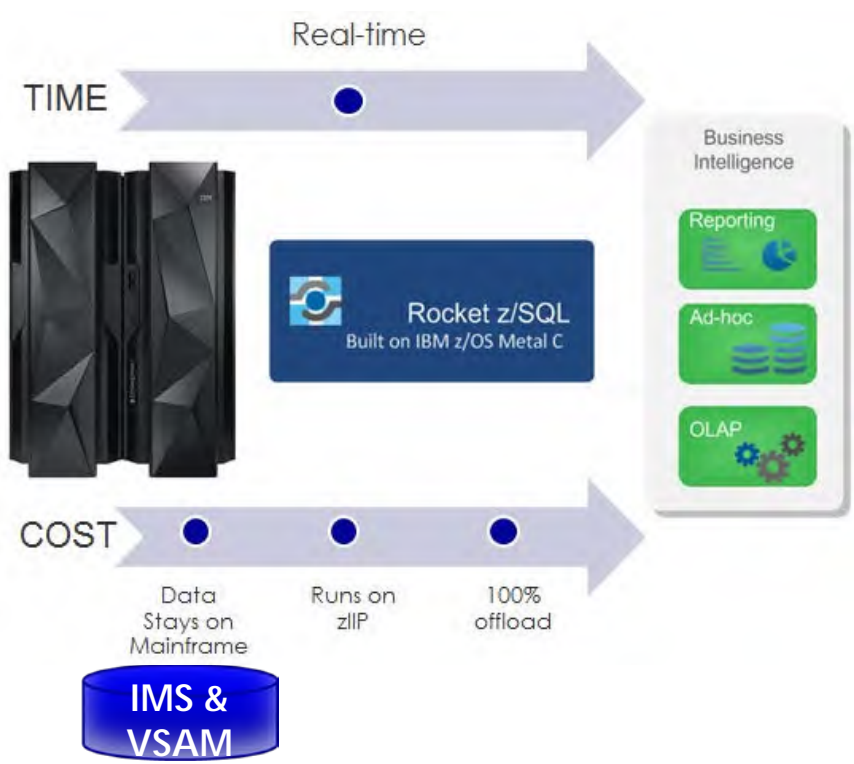
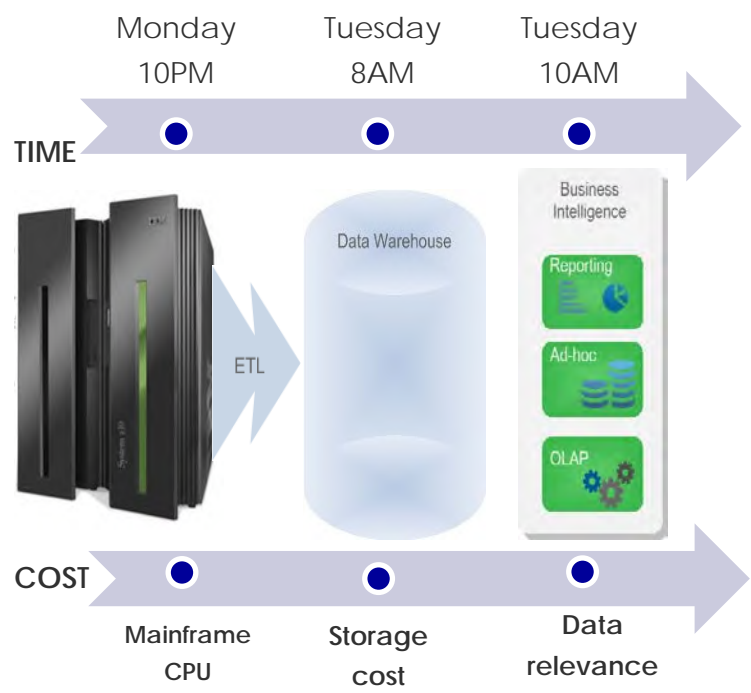
- Older data access methods are time consuming and expensive
- Issues with data latency and consistency



Analytics: Leverage Investment Direct Real Time to IMS/VSAM

Before: DW=Not Real Time

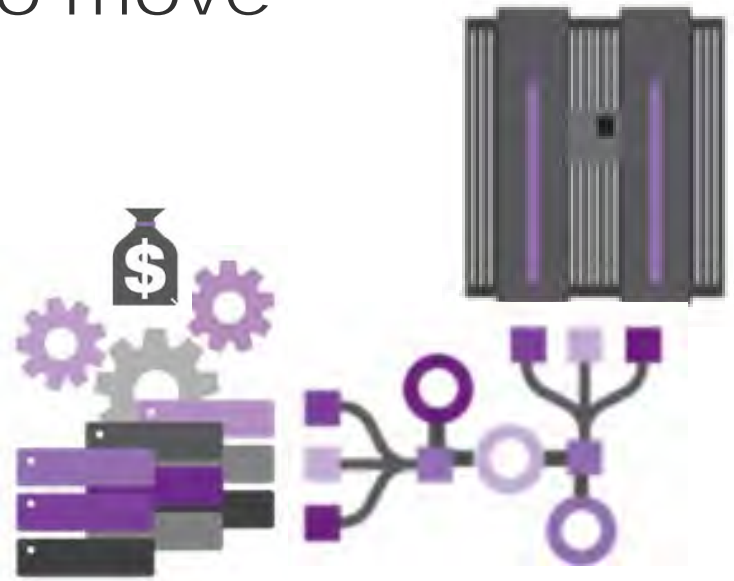
After: Direct & Real Time





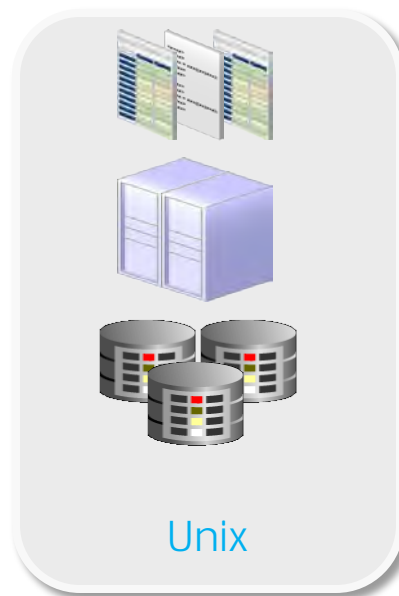
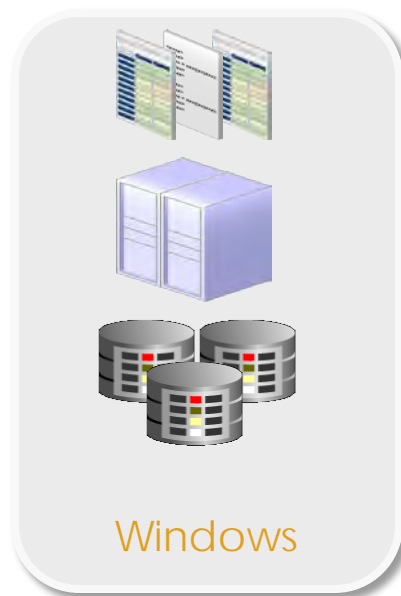
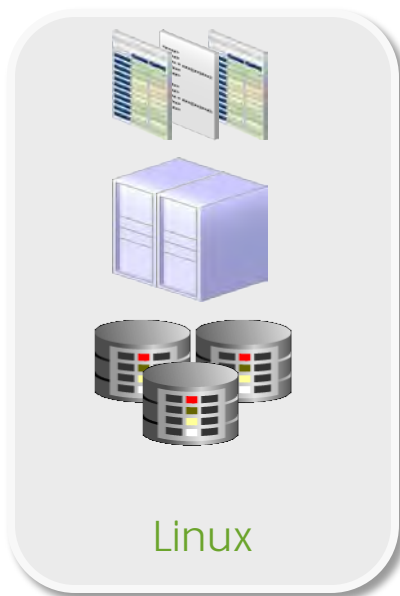
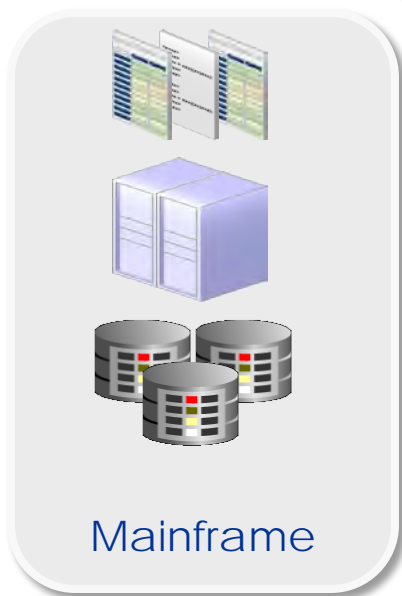
Why Move Mainframe Data

- Avoiding mainframe processing costs drives the decision to move data off host
- Data access based on mid-tier gateway servers or replication adds complexity, cost and risk



Mainframe Data and Analytics

Fractured View of Analytics

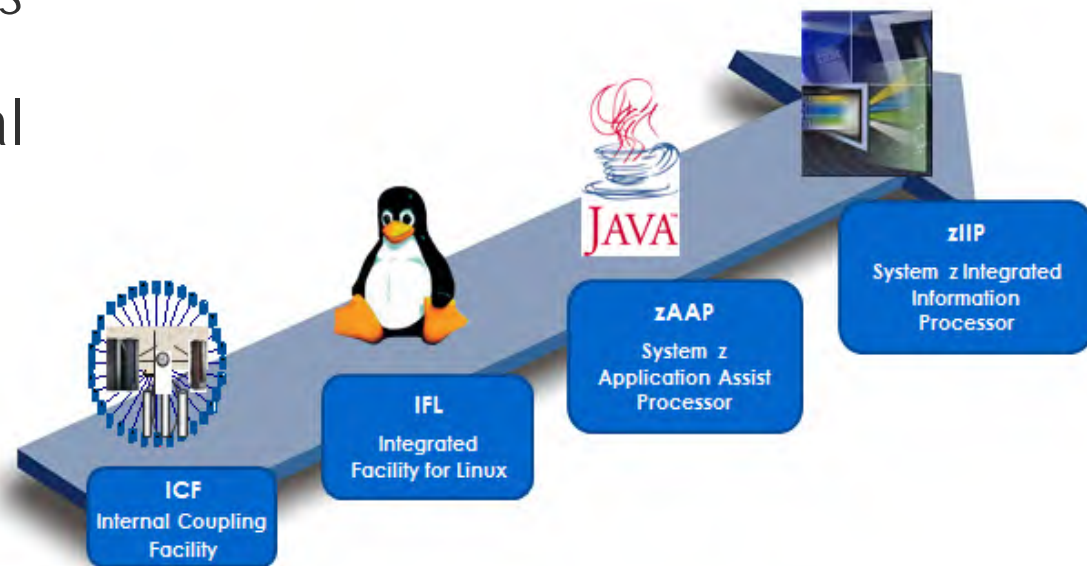


- ETL to handle legacy data transformation and correlation
- Need to create data repository to handle mainframe data
- Separate solutions for mainframe/non-mainframe systems
- Data replication to access non-relational data



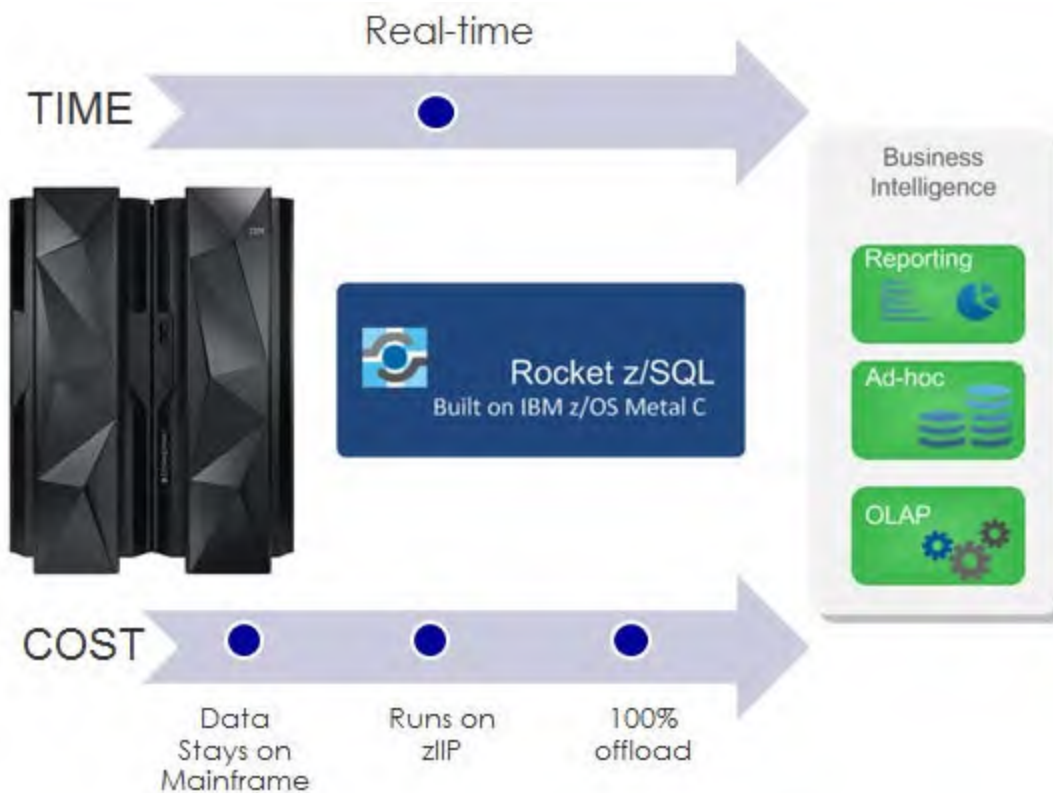
A Virtual Data Engine

- Integrating mainframe data in place can now be done cost effectively via specialty engines
- SQL to non-relational processing can be made eligible to run on the zIIP specialty engine, making it a virtual mainframe data engine





Real-time Data Access

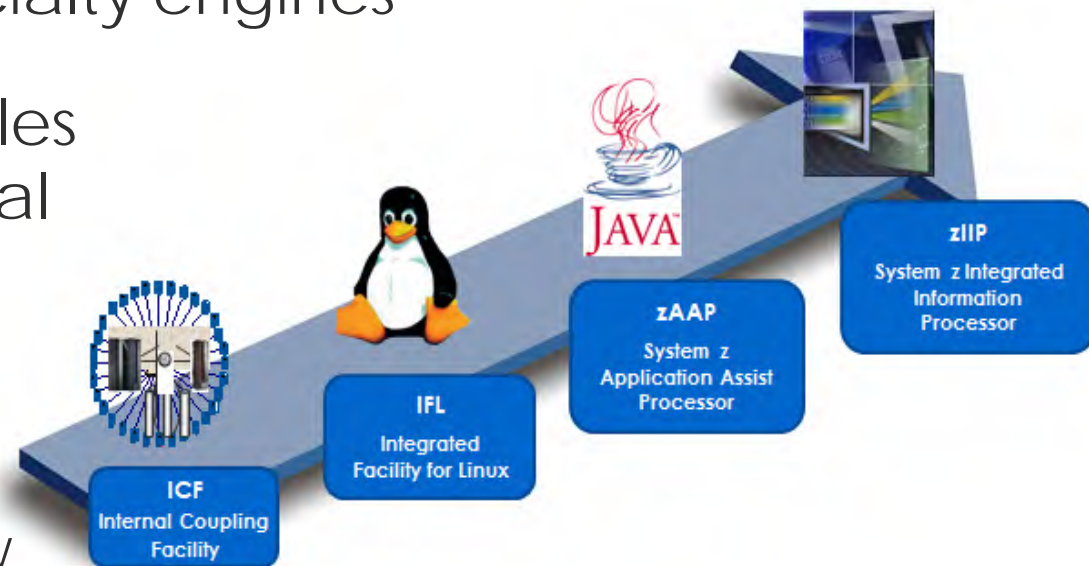


- Real-time data access with true ANSI 92 SQL
- Industry standard APIs
 - ODBC, JDBC, ADO.NET
- Runs on zIIP with up to 99% offload

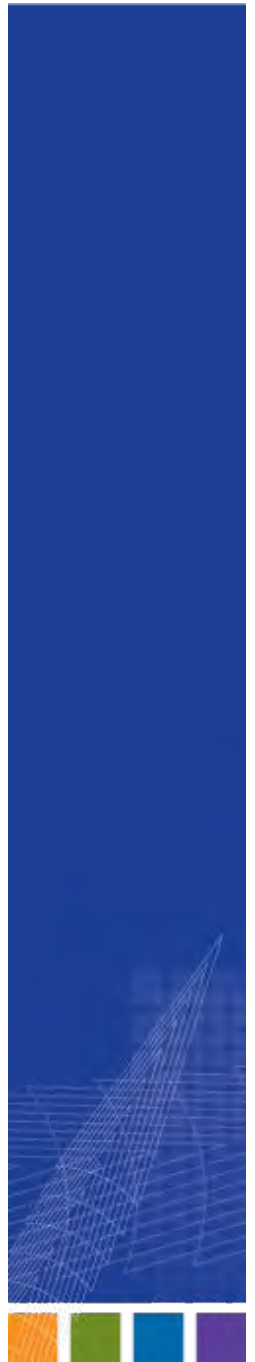


Leveraging the zIIP Engine

- Integrating mainframe data in place can now be done cost effectively via specialty engines
- Rocket z/SQL enables SQL to non-relational data processing to run on the zIIP specialty engine
 - Up to 99% of Shadow workload – for significantly reduced TCO



Dynamically transforming your IMD/DB to virtual relational tables



DBD & PSB to Virtual Relational Tables



DBD:DI21PART

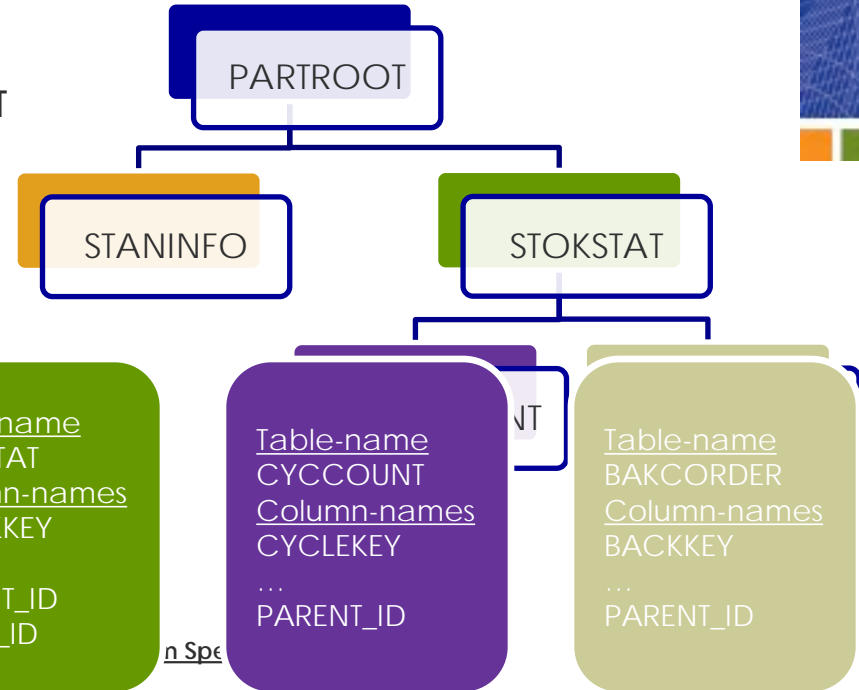


Table-name	Table-name	Table-name	Table-name	Table-name
PARTROOT	STANINFO	STOKSTAT	CYCCOUNT	BAKCORDER
Column-names	Column-names	Column-names	Column-names	Column-names
PARTKEY	STANKEY	STOCKKEY	CYCLEKEY	BACKKEY
...
CHILD_ID	PARENT_ID	PARENT_ID CHILD_ID	PARENT_ID	PARENT_ID

```

SIZE=(2048,2048) RECORD=(618,618)
SEGMENT=PARTROOT PARENT=0,BYTES=50, FREQ=250
FIELD NAME=(PARTKEY,SEQ),TYPE=C,BYTES=17,START=1
SEGMENT=STANINFO PARENT=PARTROOT,BYTES=85, FREQ=1
FIELD NAME=(STANKEY,SEQ),TYPE=C,BYTES=2,START=1
SEGMENT=STOKSTAT PARENT=PARTROOT, BYTES=160, FREQ=2
FIELD NAME=(STOCKKEY,SEQ),TYPE=C,BYTES=16,START=1
SEGMENT=CYCCOUNT PARENT=STOKSTAT, BYTES=25, FREQ=1
FIELD NAME=(CYCLKEY,SEQ),TYPE=C,BYTES=2,START=1
SEGMENT=BACKORDR PARENT=STOKSTAT, BYTES=75, FREQ=0
FIELD NAME=(BACKKEY,SEQ),TYPE=C,BYTES=10,START=1
DBDGEN
FINISH
END
  
```

```

DBPCB01 PCB TYPE=DB,DBDNAME=DI21PART,PROCOPT=GOT,
KEYLEN=43
SENSEG NAME=PARTROOT
SENSEG NAME=STANINFO,PARENT=PARTROOT
SENSEG NAME=STOKSTAT,PARENT=PARTROOT
SENSEG NAME=CYCCOUNT,PARENT=STOKSTAT
SENSEG NAME=BACKORDR,PARENT=STOKSTAT
PSBGEN LANG=COBOL,PSBNAME=DFSSAM07
END
  
```



Virtual Relational Tables from copybooks associated to DBD SEGMENTS



Column Name	Name	Num...	SQL Type	Format	Rede...	Occurs	Depe...	Offset	Length	Pre
STANKEY	STAN-REC	1	SQL_CHAR	CHARACTER				0	85	
FILLER	FILLER		SQL_CHAR	CHARACTER				0	2	
PROCUREMENT_CODE	PROCUREMENT-CODE	2	SQL_CHAR	CHARACTER	<input type="checkbox"/>			18	2	
INVENTORY_CODE	INVENTORY_CODE	2	SQL_CHAR	CHARACTER	<input type="checkbox"/>			20	1	

Table-name:
IMS_PARTREC

Column-names
PARTKEY
PREFIX
PARTNO
DESCRIPTION
...
CHILD_ID

Table-name:
IMS_STANREC

Column-names
STANKEY
PROCUREMENT_CODE
INVENTORY_CODE
MAKE_DEPT
...
PARENT_ID

Table-name:
IMS_STOKREC

Column-names
STOCKKEY
SS_AREA
SS_DEPT
SS_PROJ
...
PARENT_ID
CHILD_ID

Table-name:
IMS_CYCLEREC

Column-names
CYCLEKEY
PHYSICAL_COUNT
TOTAL_STOCK
...
PARENT_ID

Table-name:
IMS_BACKREC

Column-names
BACKKEY
WORK_ORDER
WO_QTY
...
PARENT_ID

Providing SQL access to IMS/DB



SELECT

```
a.PARTNO,  
a.DESCRPTION,  
a.RECORD_ID, b.STANKEY,  
b.INVENTORY_CODE
```

FROM

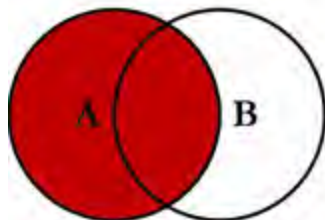
```
SHADOW.IMS_PARTREC a
```

LEFT OUTER JOIN

```
SHADOW.IMS_STANREC b
```

ON

```
a.CHILD_ID = b.PARENT_ID;
```



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

	PARTNO	DESCRIPTION	RECORD_ID	STANKEY	INVENTORY_CODE
0	AN960C10	WASHER	02AN960C10	02	2
1	CK05CW181K	CAPACITOR	02CK05CW181K	02	2
2	CSR13G104KL	KR1J50KS	02CSR13G104KL	02	2
3	JAN1N976B	WASHER	02JAN1N976B	02	2
4	MS16995-28	SCREW	02MS16995-28	02	2
5	N51P3003F000	SCREW	02N51P3003F000	02	2
6	RC07GF273J	RESISTOR	02RC07GF273J	02	2
7	106B1293P009	RESISTOR	02106B1293P009	02	2
8	123456789ABCDE		02123456789ABCDE	02	
9	250236-001	CAPACITOR	02250236-001	02	2
10	250239	TRANSISTOR	02250239	02	2
11	250241-001	CONNECTOR	02250241-001	02	2
12	250794	RESISTOR	02250794	02	2
13	250796	SWITCH	02250796	02	2
14	250891	SERVO VALVE	02250891	02	2
15	252252-003	COUPLING	02252252-003	02	2
16	3003802	CHASSIS	023003802	02	2
17	3003806	SWITCH	023003806	02	2
18	3007228	HOUSING	023007228	02	2
19	3008027	CARD FRONT	023008027	02	A
20	3009228	CAPACITOR	023009228	02	2
21	3009270	HOUSING	023009270	02	2
22	3009280	HOUSING CONV	023009280	02	2
23	3013405-002	MOUNTING	023013405-002	02	
24	3013412	COVER	023013412	02	2

68 rows | SQL Messages

Providing SQL access to IMS/DB

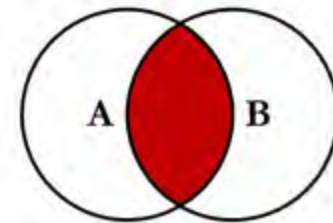


```
SELECT
  a.PARTNO,
  a.DESCRPTION,
  a.RECORD_ID,
  b.STANKEY,
  b.INVENTORY_CODE
```

```
FROM
  SHADOW.IMS_PARTREC a,
  SHADOW.IMS_STANREC b
```

```
WHERE
  b.PARENT_ID =
  a.CHILD_ID
```

```
AND
  b.INVENTORY_CODE='6';
```



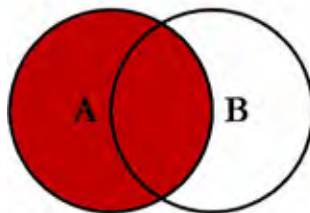
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

	PARTNO	DESCRIPTION	RECORD_ID	STANKEY	INVENTORY_CODE
0	930331-102	FILTER	02930331-102	02	6
1	930331-123	FILTER	02930331-123	02	6
2	930333-001	DISCRIMINATO	02930333-001	02	6

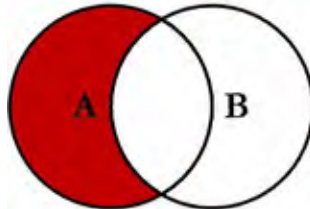
SQL Joins



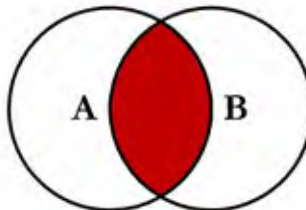
SQL JOINS



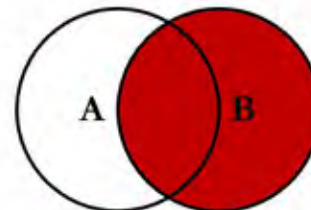
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



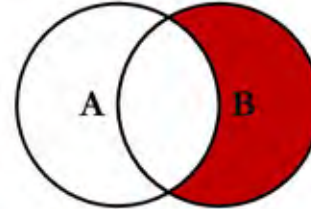
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



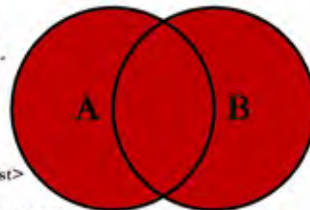
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



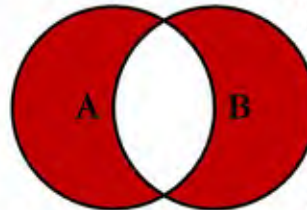
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

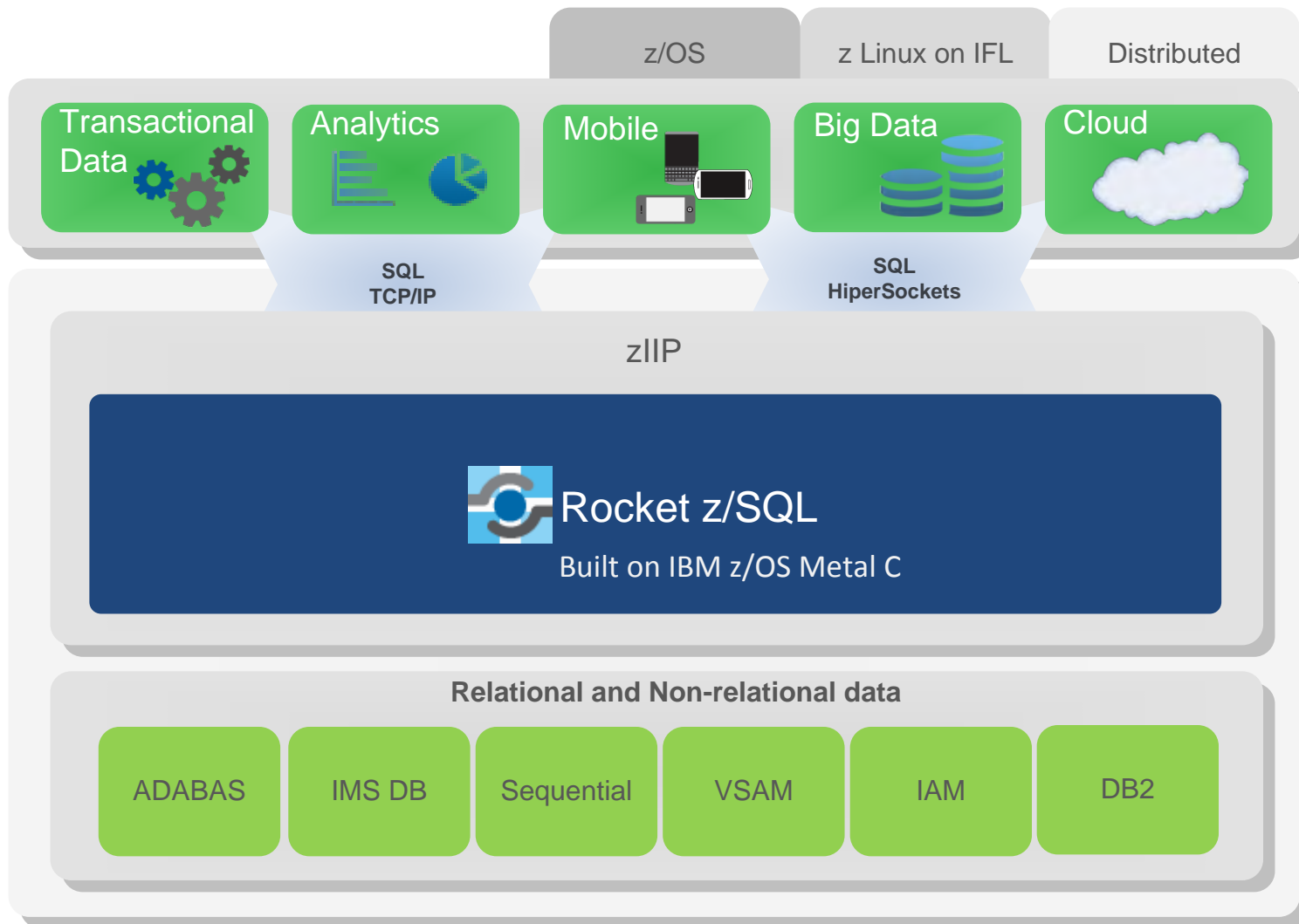


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Rocket z/SQL

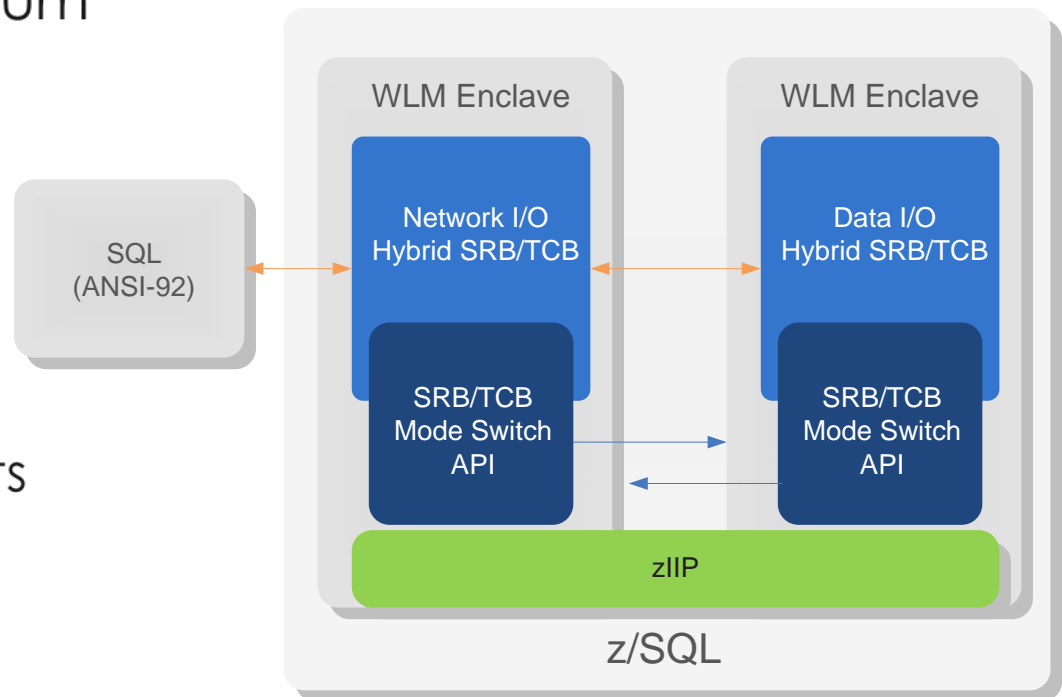
Workloads and Deployment Scenarios



Rocket z/SQL

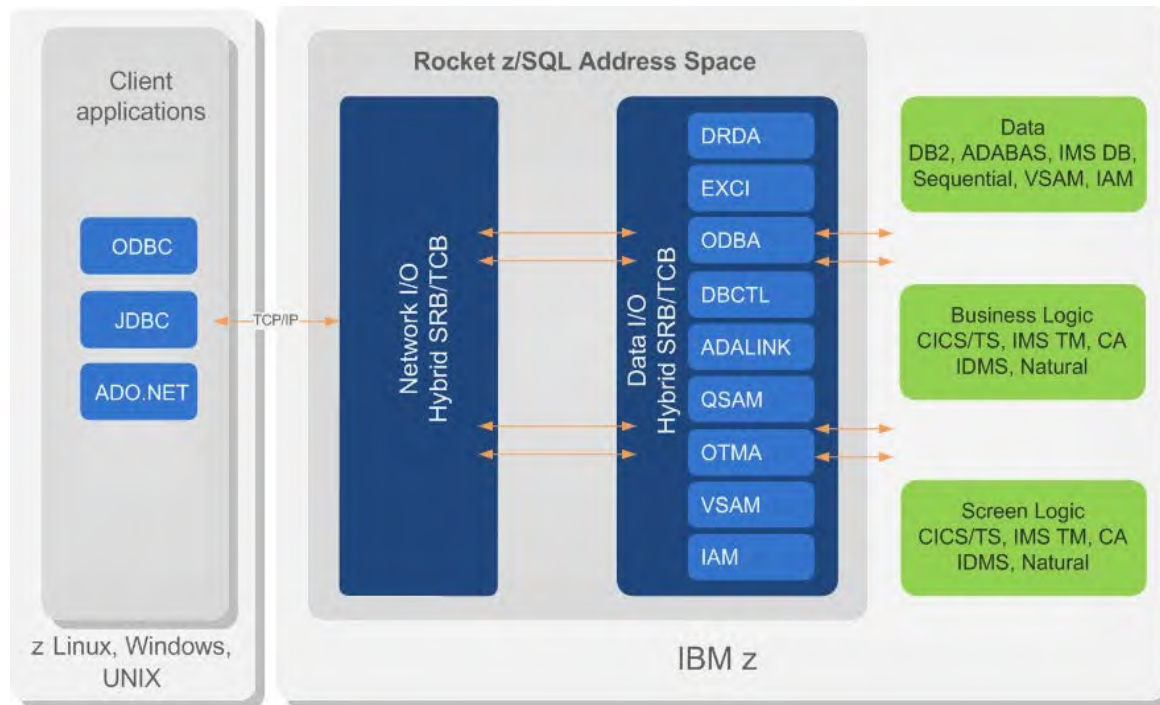
Leverages Specialty Engines for Reduced Cost

- Engineered for maximum TCO benefit
- Logical Dispatchable Unit (LDU)
 - hybrid SRB/TCB thread
- Unique Workload Manager enhancements
 - ability to prioritize zIIP workloads



Rocket z/SQL

Parallel I/O Architecture





Questions?

- www.rocketsoftware.com
 - Download a demo
 - <http://info.rocketsoftware.com/mainframe-data-access.html>

Contact

- Wayne Morton
 - wmorton@rocketsoftware.com
 - (713) 806-2321



Dank u wel
 Mahalo
 Juspaxar
 Spasibo
 Efcharisto
 Salamet
 Todara raba
 Cam on
 Dhanyawaad
 Köszi
 Grazie mille
 Kamsahamnida
 Dankon
 Pldamayado
 Tusen takk
 Terima kasih
 Doh je
 Maizika
 ekokaj
 Tawapach
 Shukria
 Shukria
 Iah
 Merastawhy
 Merastawhy
 rubun
 Biyan
 Bolzin
 Maake
 gozaimashita
 fakaan
 Spasibo
 Ekhmet
 Mehrbani
 Yaqhanyelay
 Efcharisto
 Arigato
 Dankscheen
 Komapsumnida
 Wabooja
 Medawage
 Mensi
 unichinash
 Tingki
 Shukuria
 Paidies
 Hatur
 arbu
 Denkaaja
 Agayji
 Naasobhya
 Baika
 Yopagaratan
 Memonchar
 Atto
 Garjiho
 suk sama
 Saemo
 Dala
 Dhanyabaad
 Yopagnsunki
 Oubla
 Obrigado
 Shukria
 Salamet

