Architectural Implementation Analysis
A Comparative Methodology

Monday, March 02, 2015, 04:30PM-05:30PM

Sheraton Seattle, Greenwood

Monte Bauman

mbauman@us.ibm.com

IBM Columbus

Enterprise Server Technical Support

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

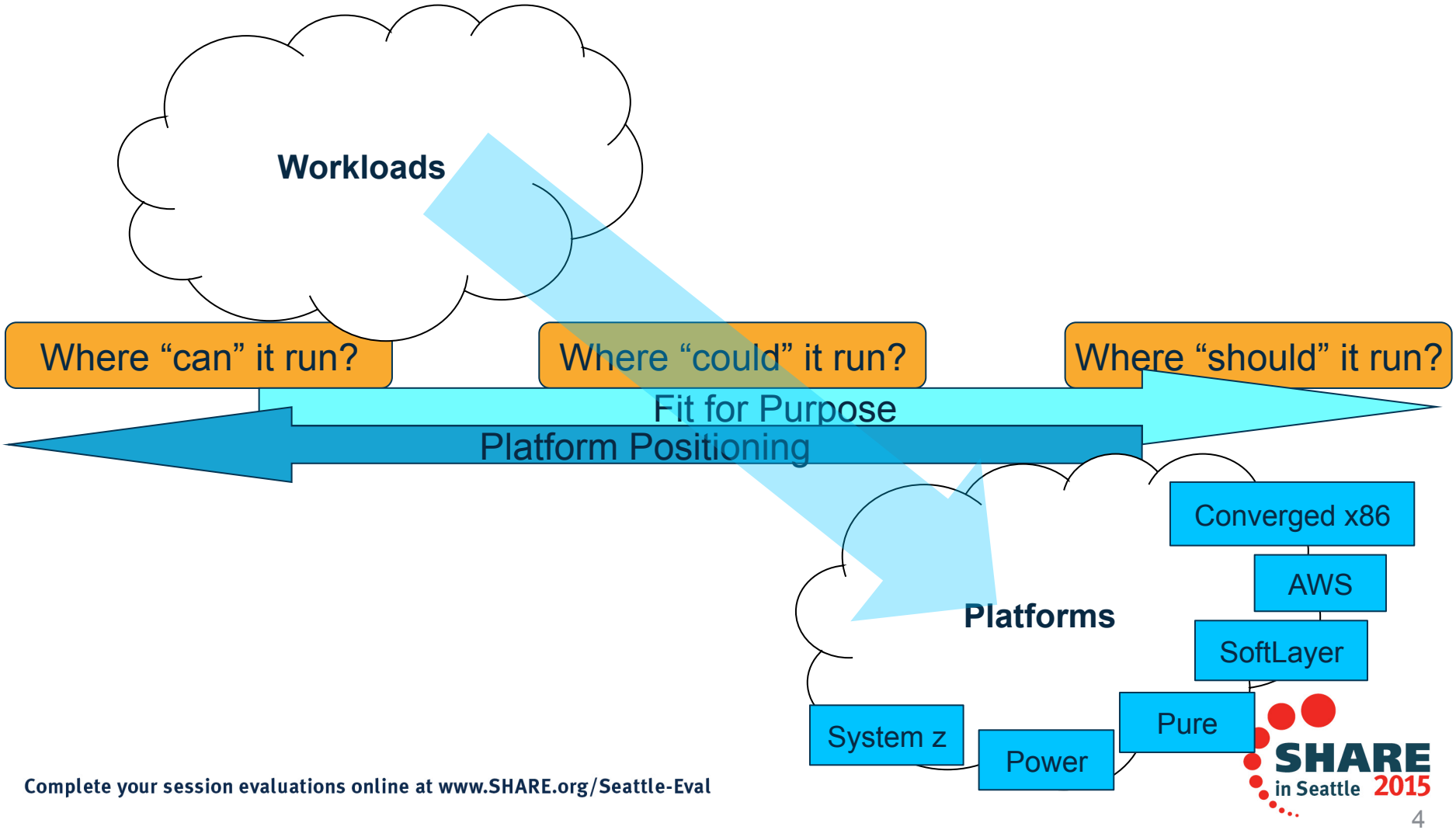
Presentation Plan

- The Challenge
- An Approach – IBM IT Optimization
- Architecture Analysis with RACEa
- Conclusion – I need your help

The Challenge

The Challenge

Run the right work on the right platform … optimizing cost time risk and capability



The Challenge
Making Everyone Happy

Architects

Developers

Engineers

Technicians

Call Center

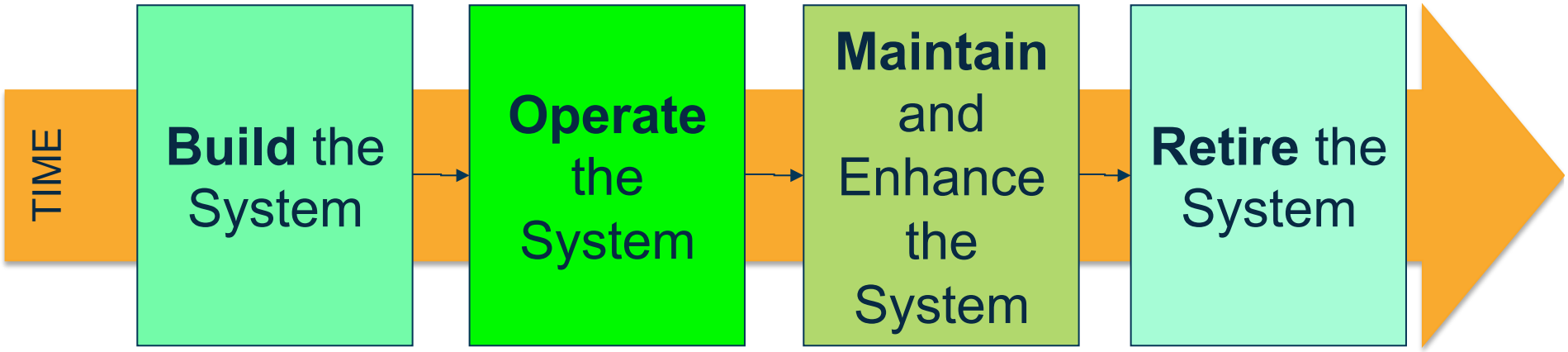
Business Analyst

Application Owner

The Challenge

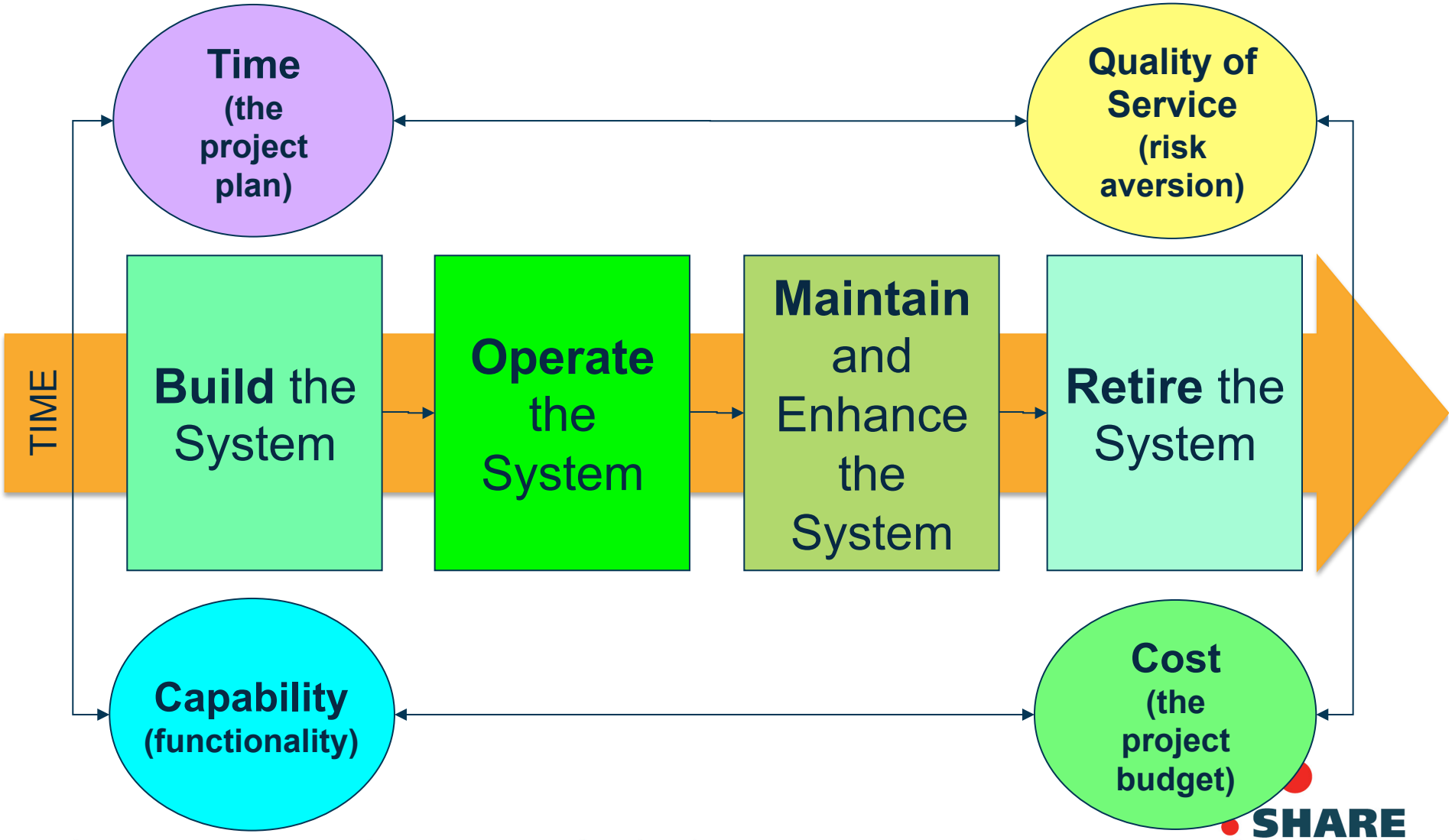
Making Everyone Happy…

Not Only Now … But Over Time …

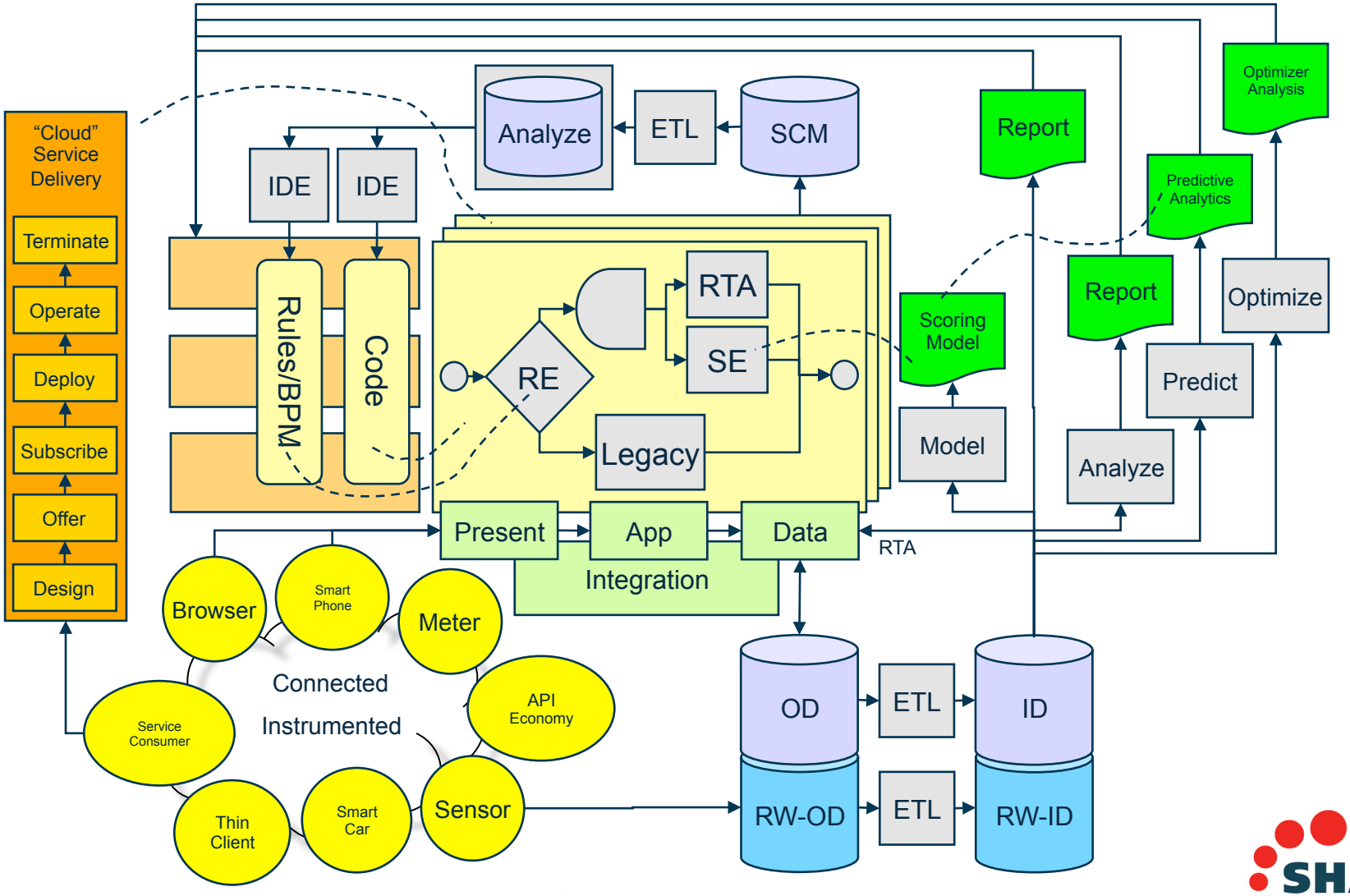


The Challenge

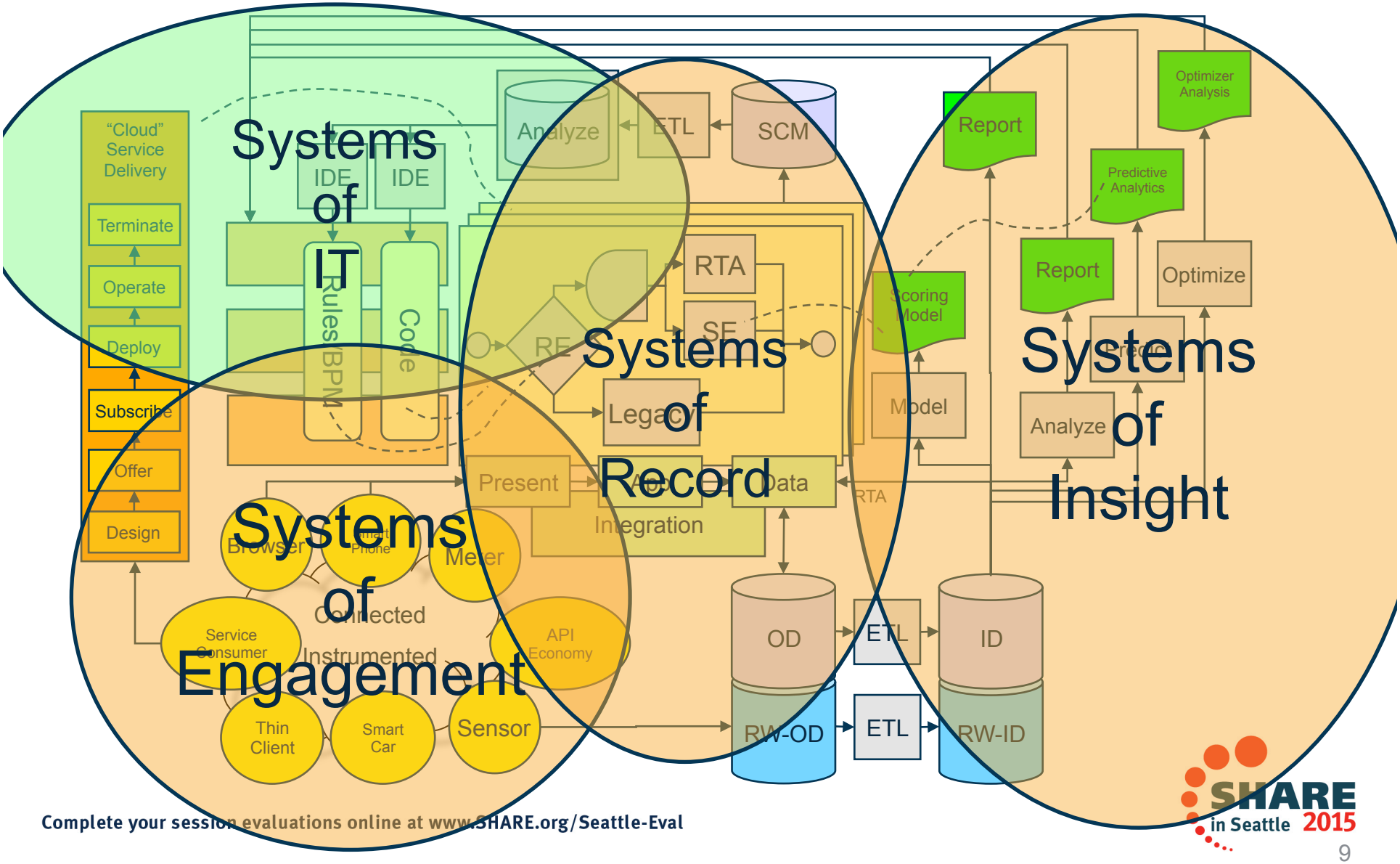
Decisions Considering All the Right Things…



Systems of IT: Cloud Service Delivery

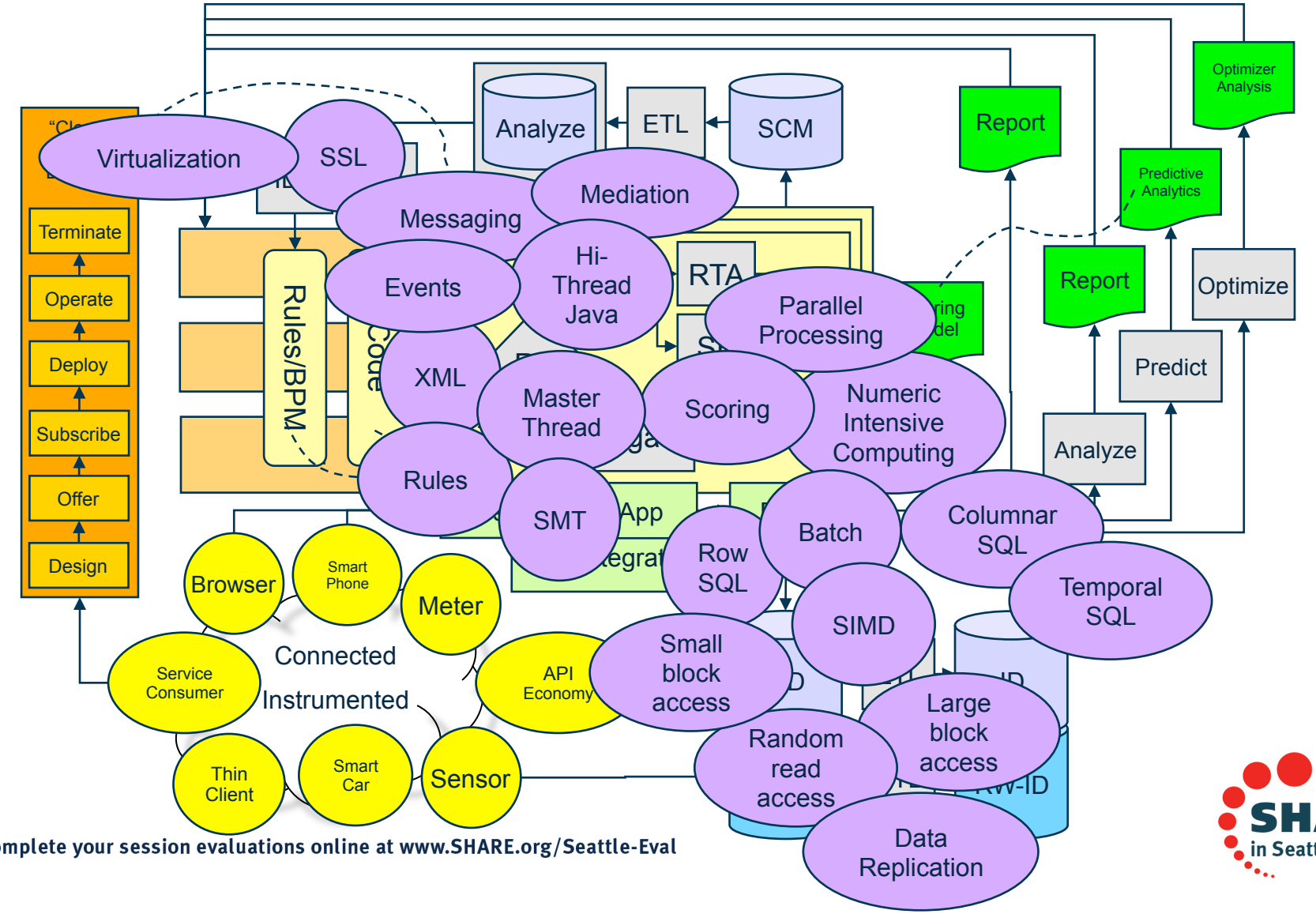


Systems of Record / Engagement / Insight / IT

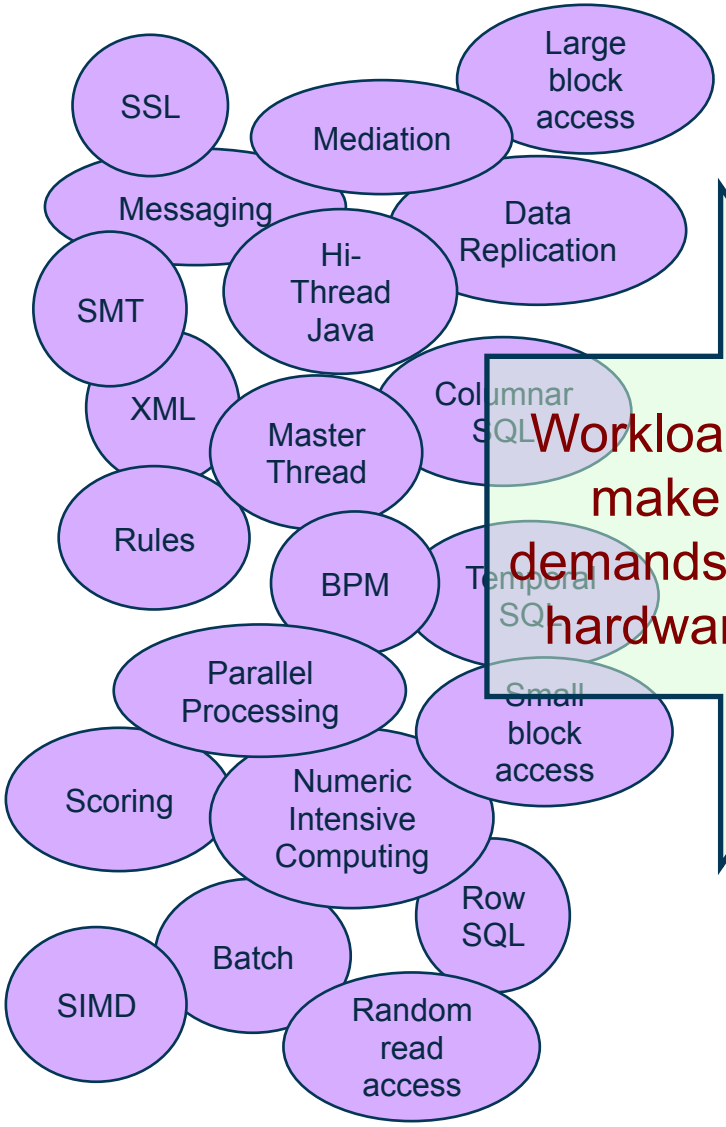


“Optimally” Executing All These Workloads is a Challenge…

No single platform can do it all (maybe that’s why we offer more then one!)



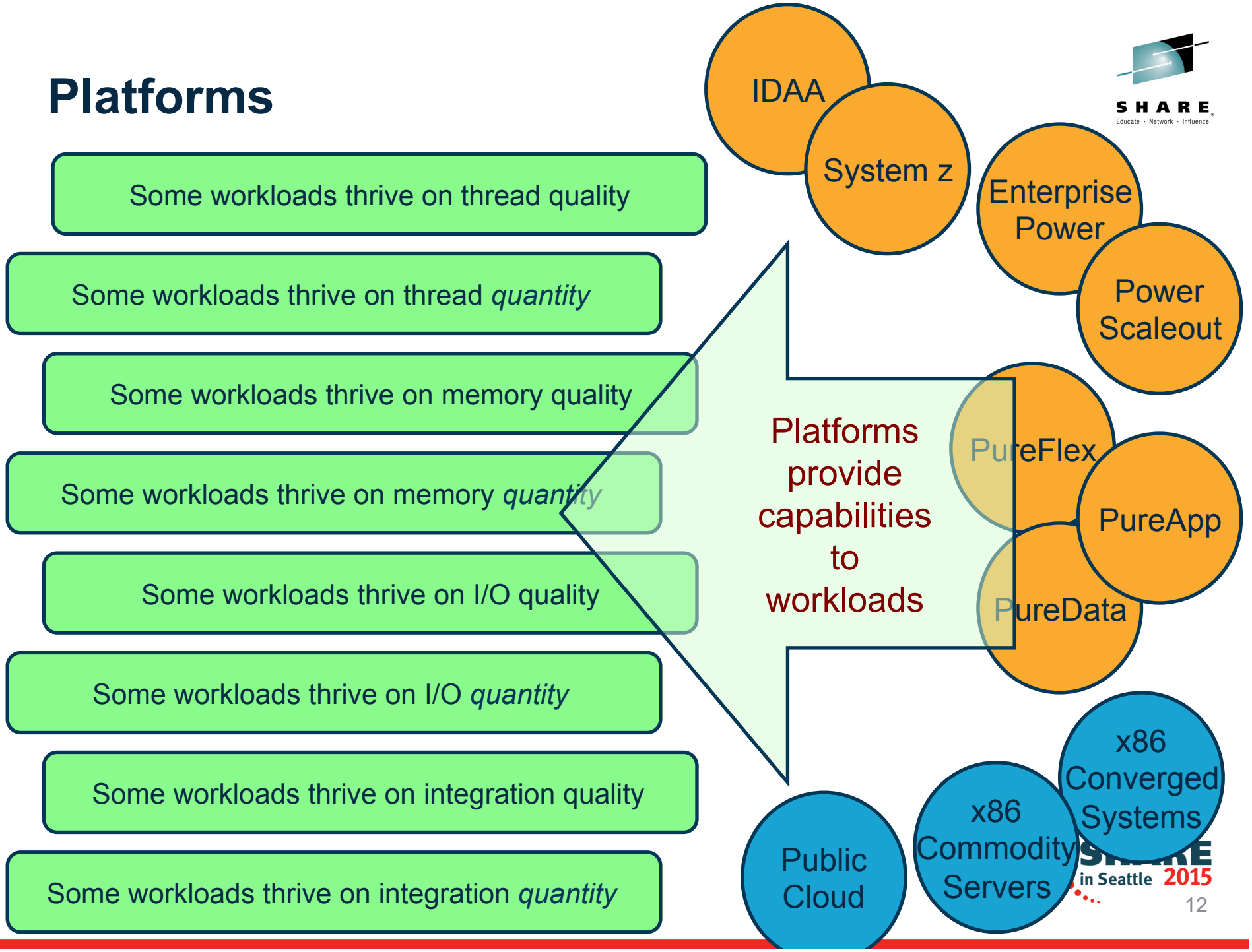
Workloads



Workloads make demands on hardware

- Some workloads thrive on thread quality
- Some workloads thrive on thread *quantity*
- Some workloads thrive on memory quality
- Some workloads thrive on memory *quantity*
- Some workloads thrive on I/O quality
- Some workloads thrive on I/O *quantity*
- Some workloads thrive on integration quality
- Some workloads thrive on integration *quantity*

Platforms


Some workloads thrive on thread quality
Some workloads thrive on thread *quantity*
Some workloads thrive on memory quality
Some workloads thrive on memory *quantity*
Some workloads thrive on I/O quality
Some workloads thrive on I/O *quantity*
Some workloads thrive on integration quality
Some workloads thrive on integration *quantity*
Platforms provide capabilities to workloads
IDAA
System z
Enterprise Power
Power Scaleout
PureFlex
PureApp
PureData
Public Cloud
x86 Commodity Servers
x86 Converged Systems

Choices Choices Choices

Making the Right Choice is…

(A) Hard

(B) Necessary

(C) Time Consuming

(D) Fit for Purpose

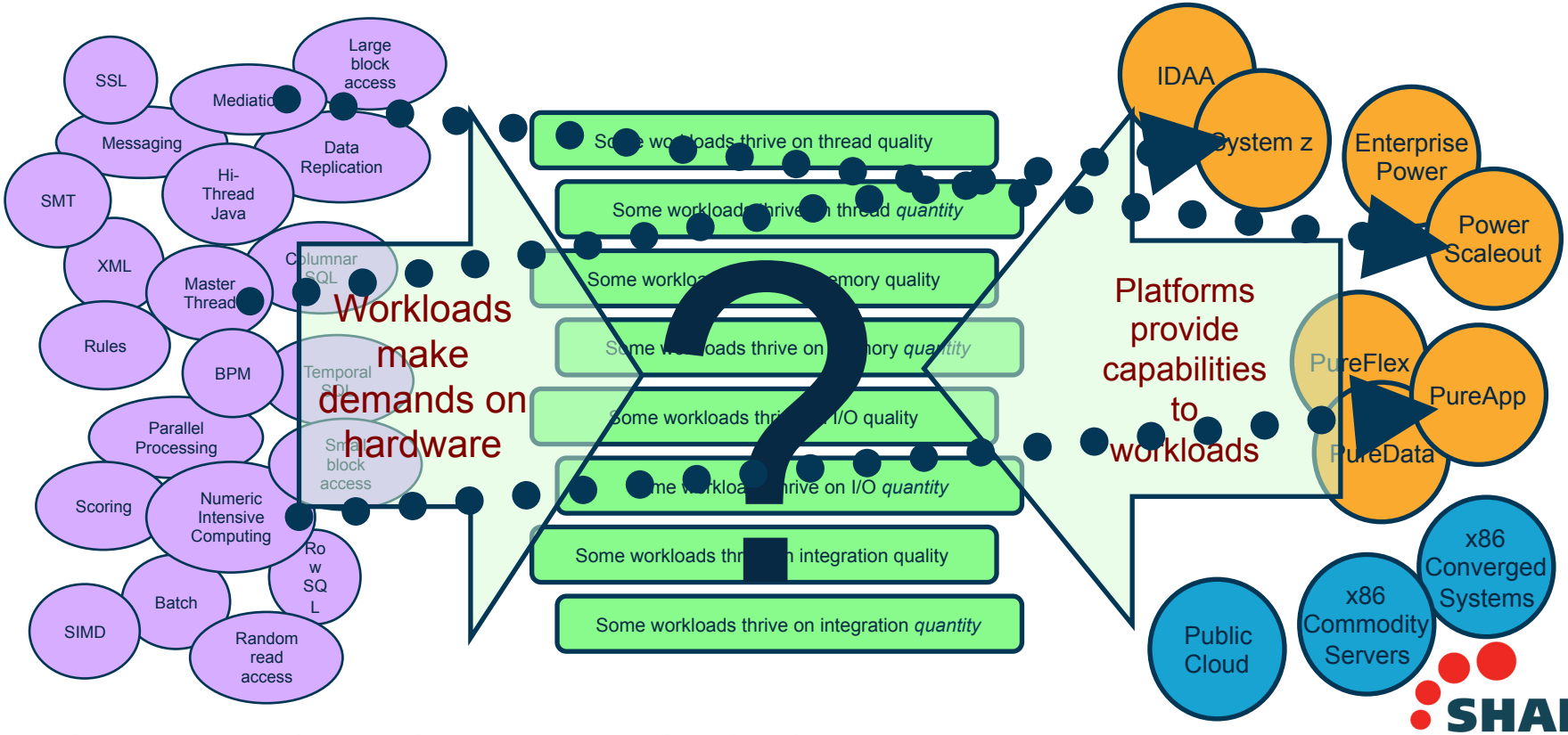
(E) IT Optimization

(F) A function of familiarity

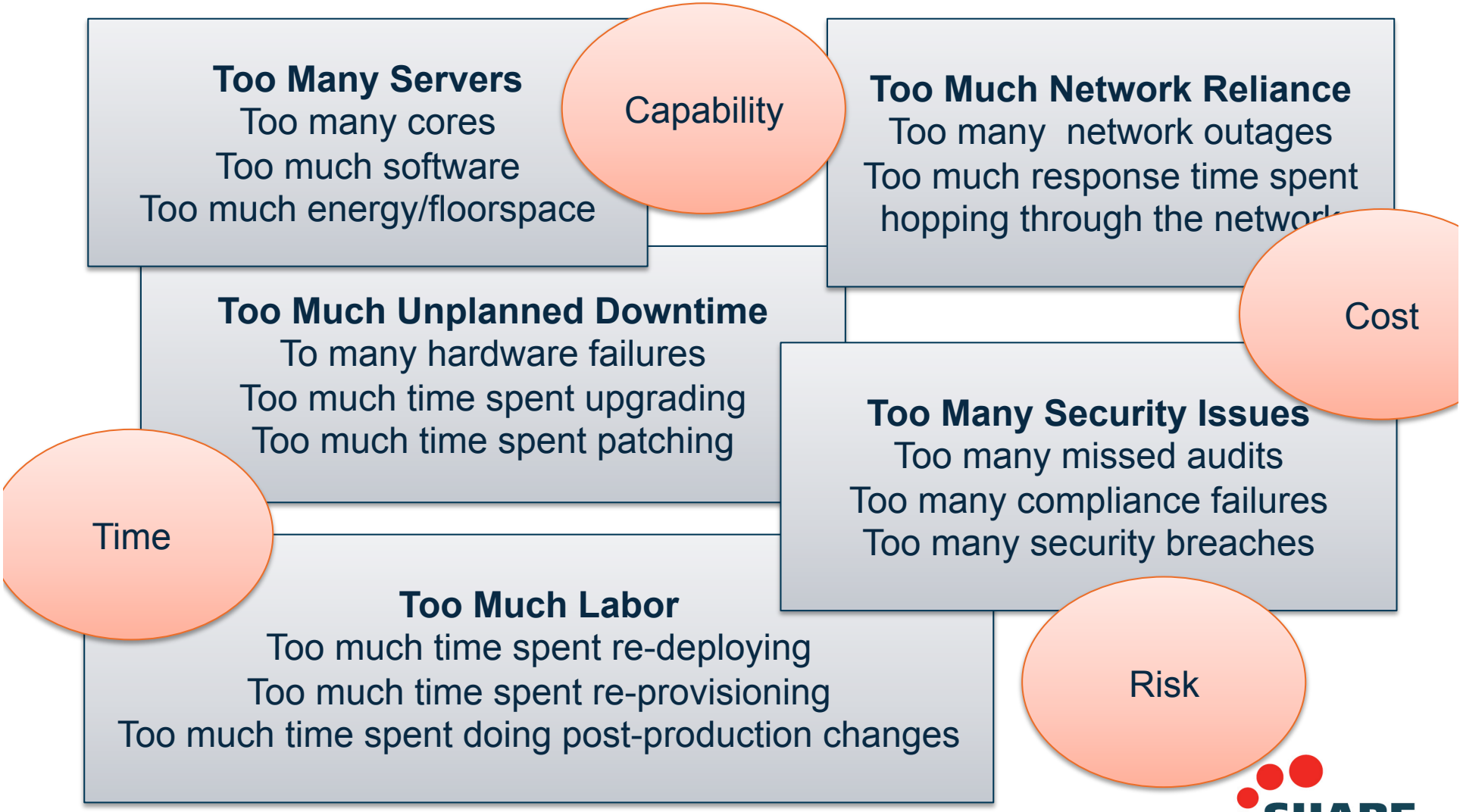
(G) Essential

(H) Important

(H) A wonderful thing!

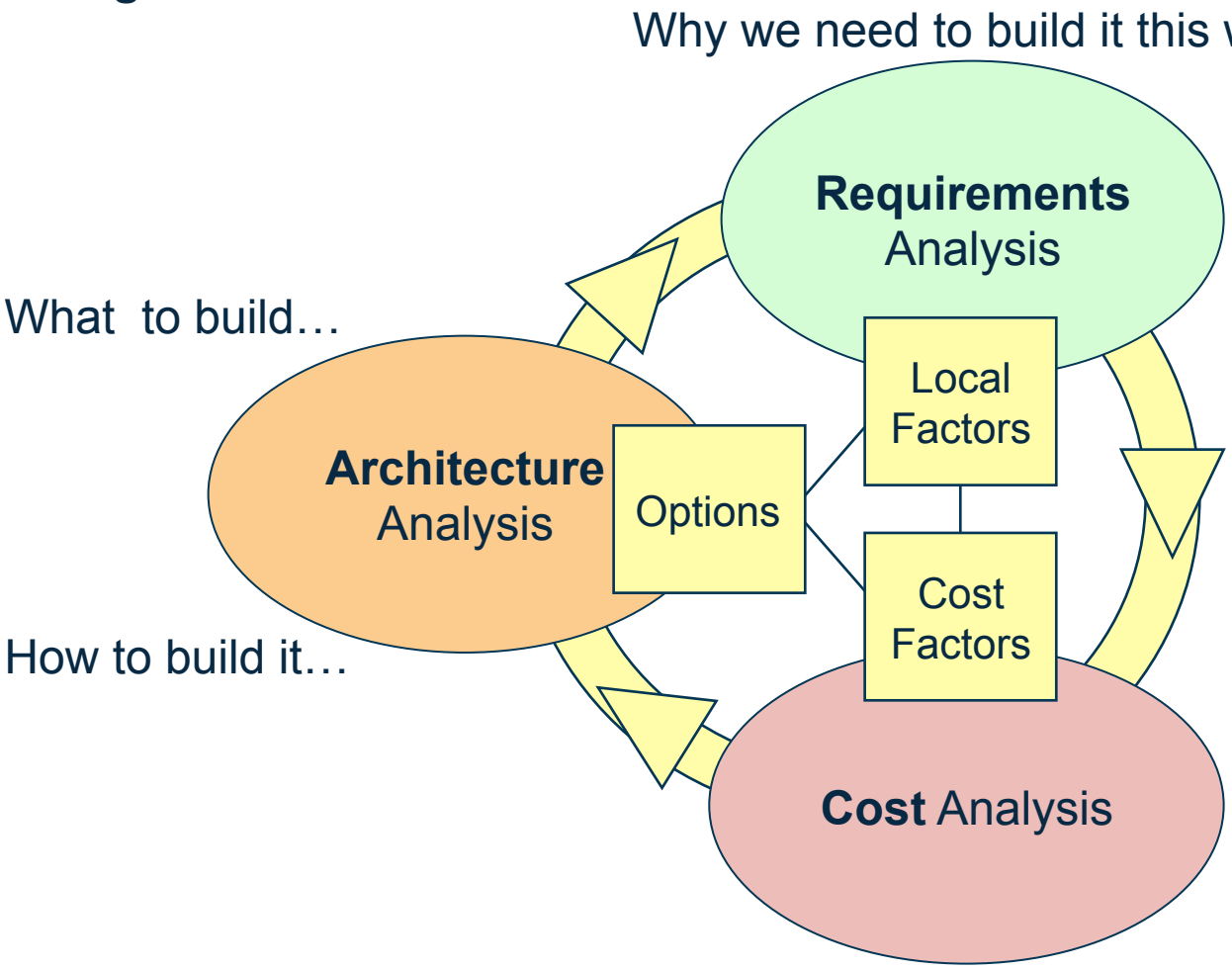


The Penalties of IT Un-Optimization



An Approach – IBM IT Optimization

IT Optimization Assessment Methodology
Three Stages

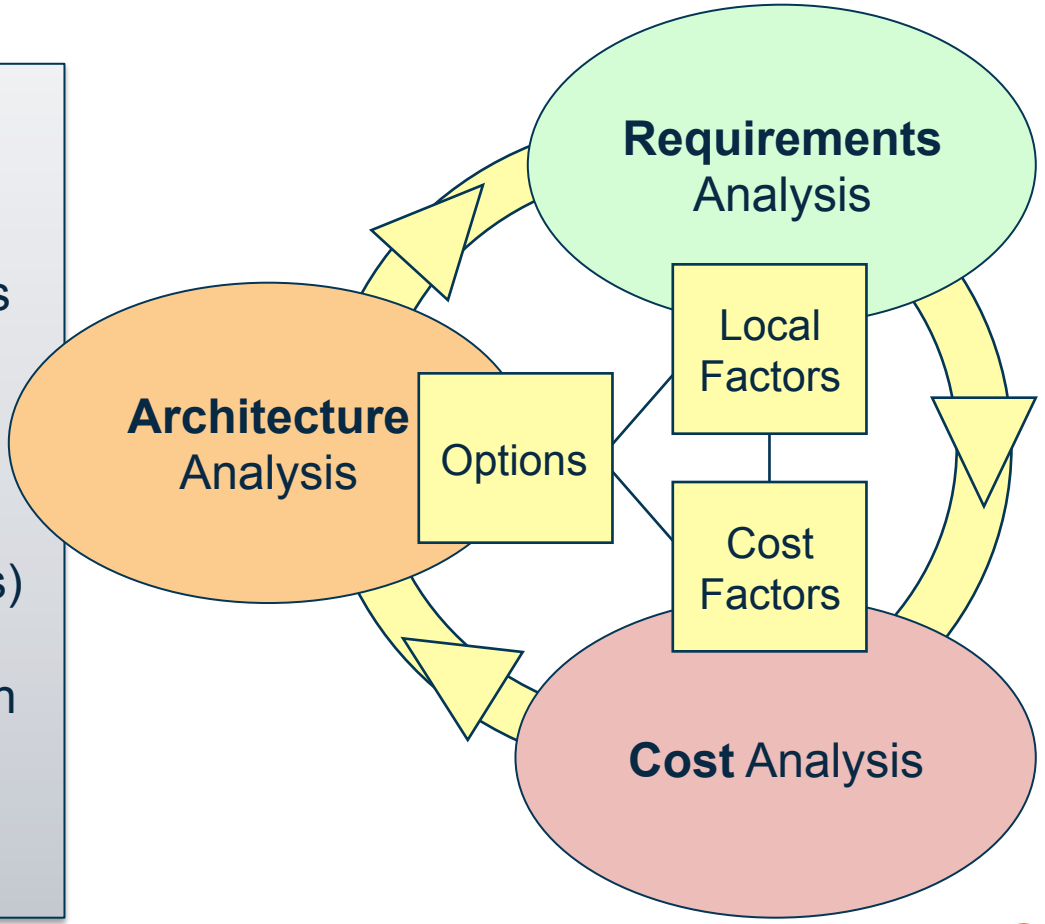


IT Optimization Assessment Methodology
Three Stages

Architecture Analysis

- Functional Requirements
  - Logical Architecture
    - Code and Data
      - Containers
    - Platforms (Clusters)
- Connections (Connectors)

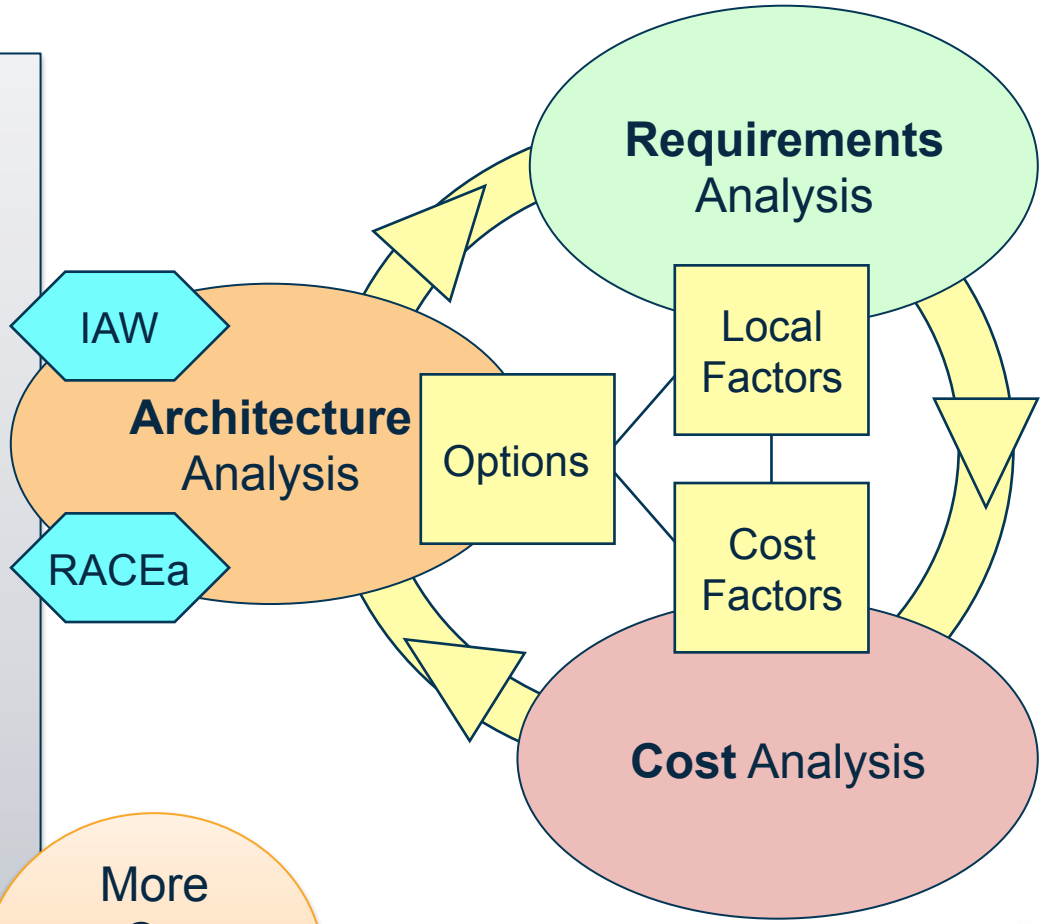
Sorted list of implementation options based upon logical architecture



IT Optimization Assessment Methodology
Three Stages

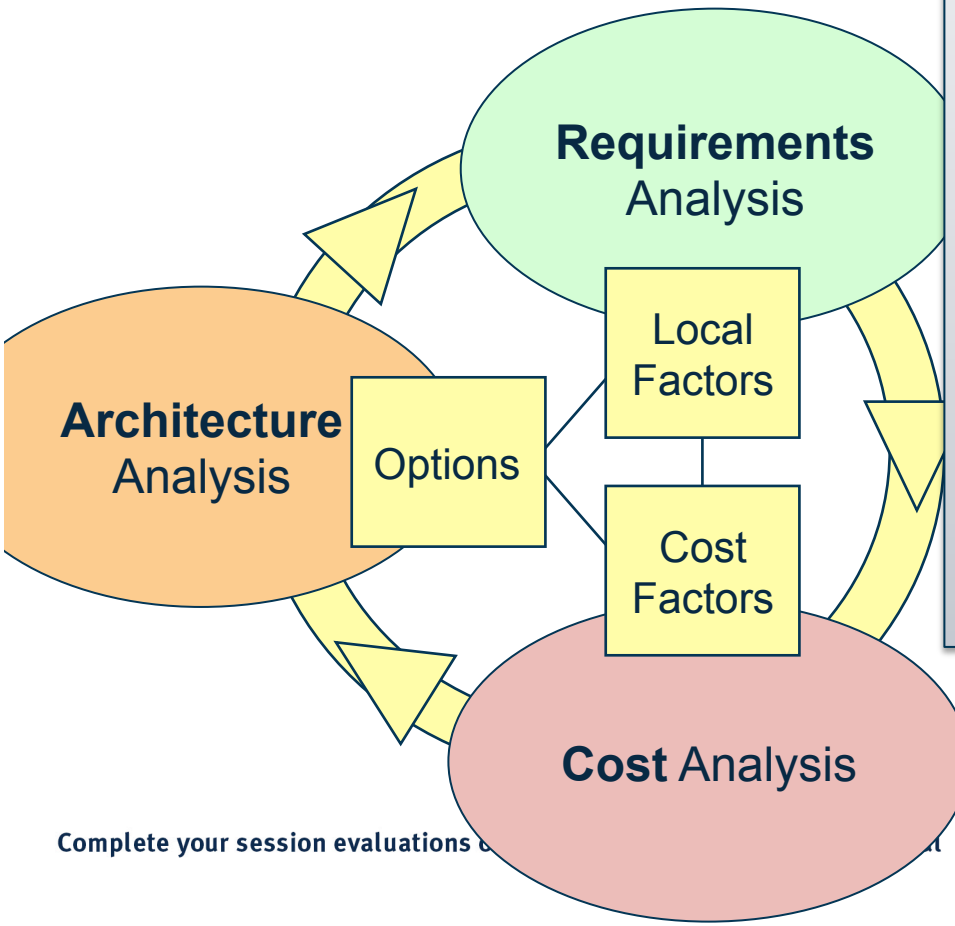
IAW
- IBM Infrastructure Architecture Workshop
- 1 or 2 day workshop
- IBM architects + app architects + enterprise architects + engineers
- Articulate and sort options

RACEa
- Comparative Architecture Analysis Tool
- Technical merit scoring
- Complexity scoring
- Bill of material scoring



More On RACEa Soon

IT Optimization Assessment Methodology
Three Stages



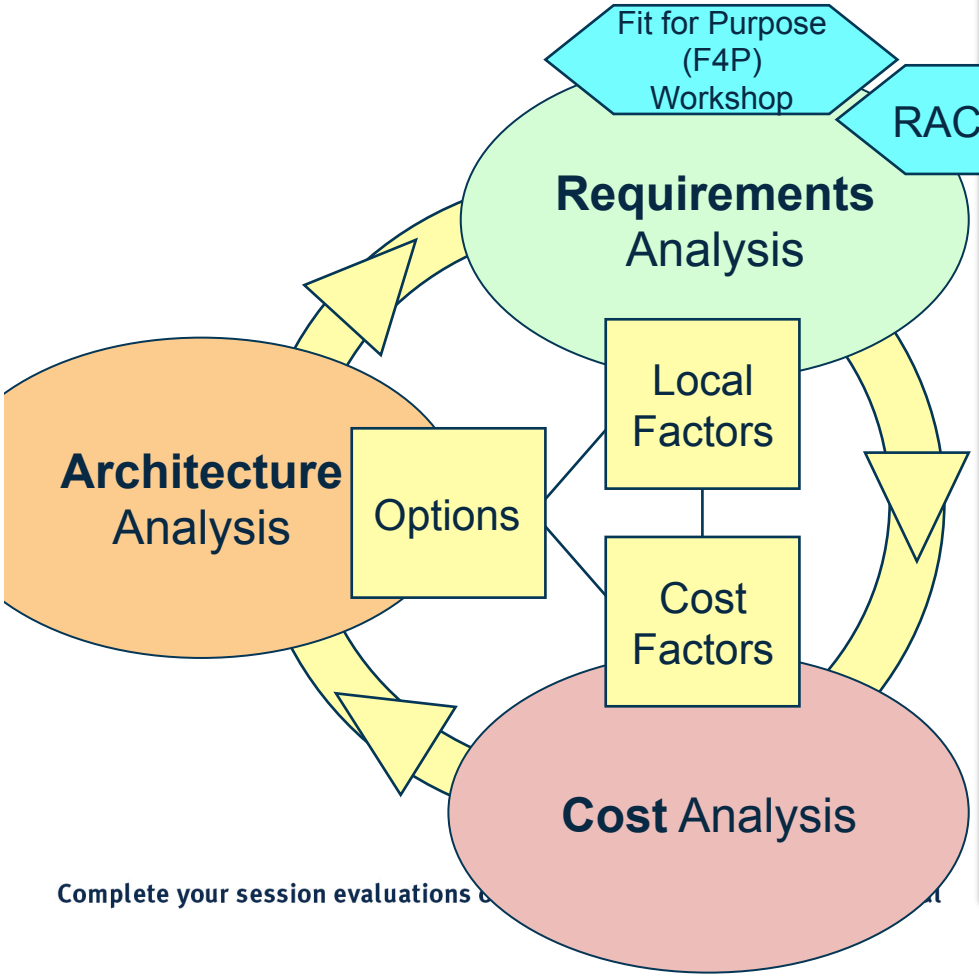
Requirements Analysis

- Non Functional Requirements
  - Quality of Service
  - Throughput and Scale
  - Resilience and DR
  - Security and Audit
  - Skills and Investments

Sorted list of implementation options based upon requirements fulfillment scoring

IT Optimization Assessment Methodology

Three Stages



Fit for Purpose Workshop

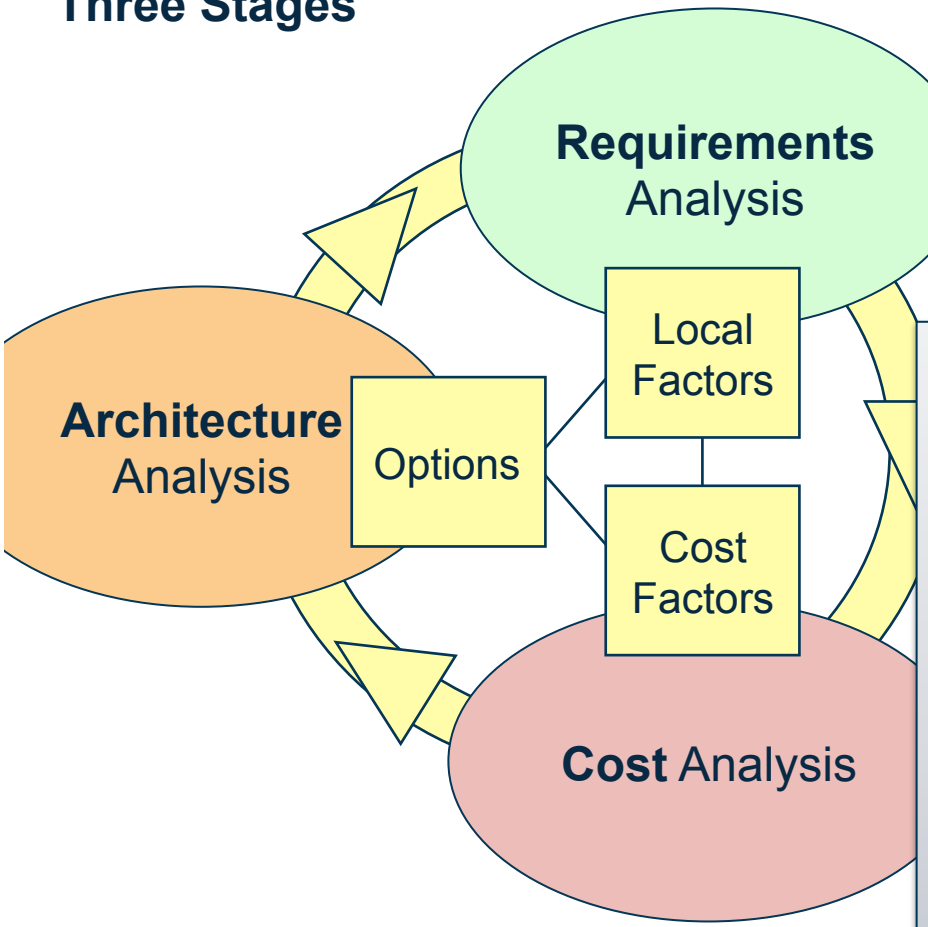
- 1 or 2 day workshop
- Project (or pattern) scope
- IBM moderated
- Architects + developers + engineers + IBM SMEs
- Structured debate
- Tool facilitated scoring
- Options requirements-fitness based sorting

RACEf

- Platform requirements analysis and filtering tool
- Once calibrated, creates customized enterprise platform positioning tool

IT Optimization Assessment Methodology

Three Stages



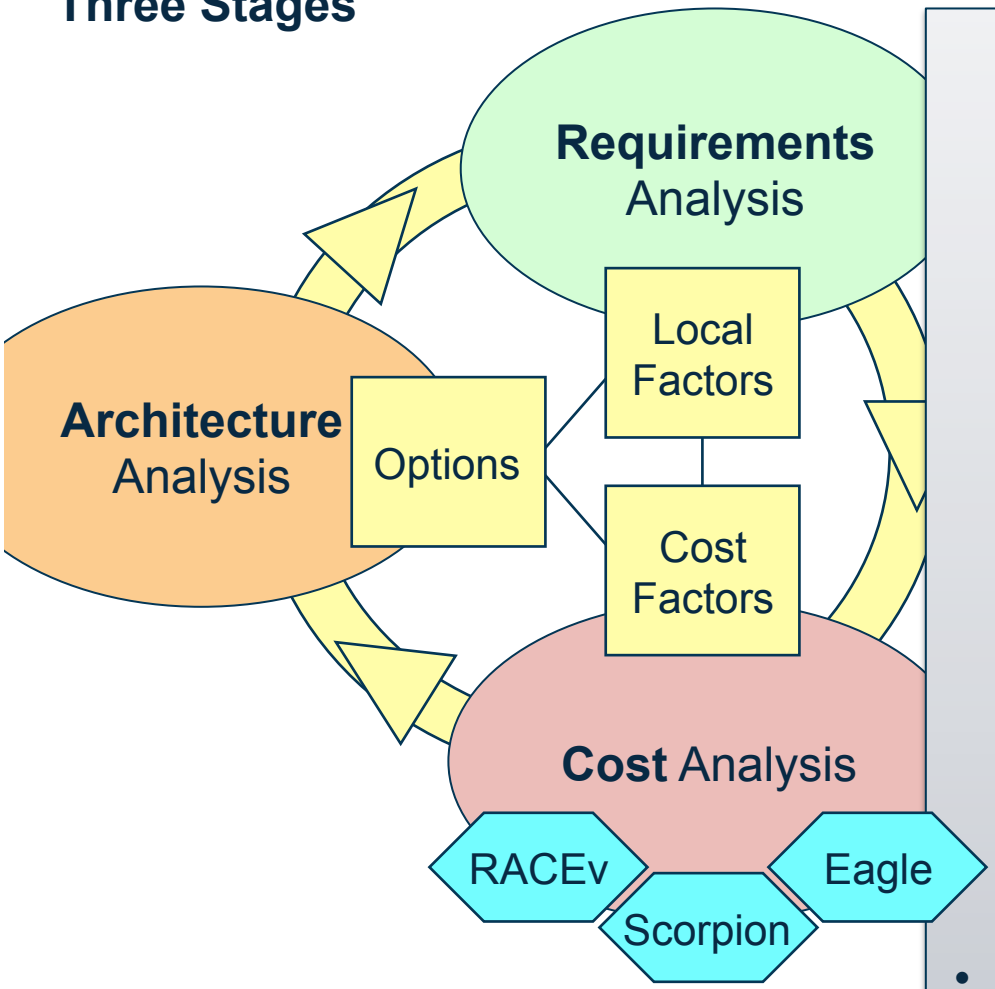
Cost Analysis

- Total cost of ownership (TCO)
- Complete Bill of Materials
- Hardware Software Storage Networks
- Labor Facilities DR
- Full Lifecycle
- Build Run Manage Retire

Sorted list of implementation options based upon TCO scoring

IT Optimization Assessment Methodology

Three Stages



RACEv

- Right-fitting applications into consolidated environments
- Spreadsheet-based tool
- Technical analysis
- Cost analysis
- TCO Scorecard

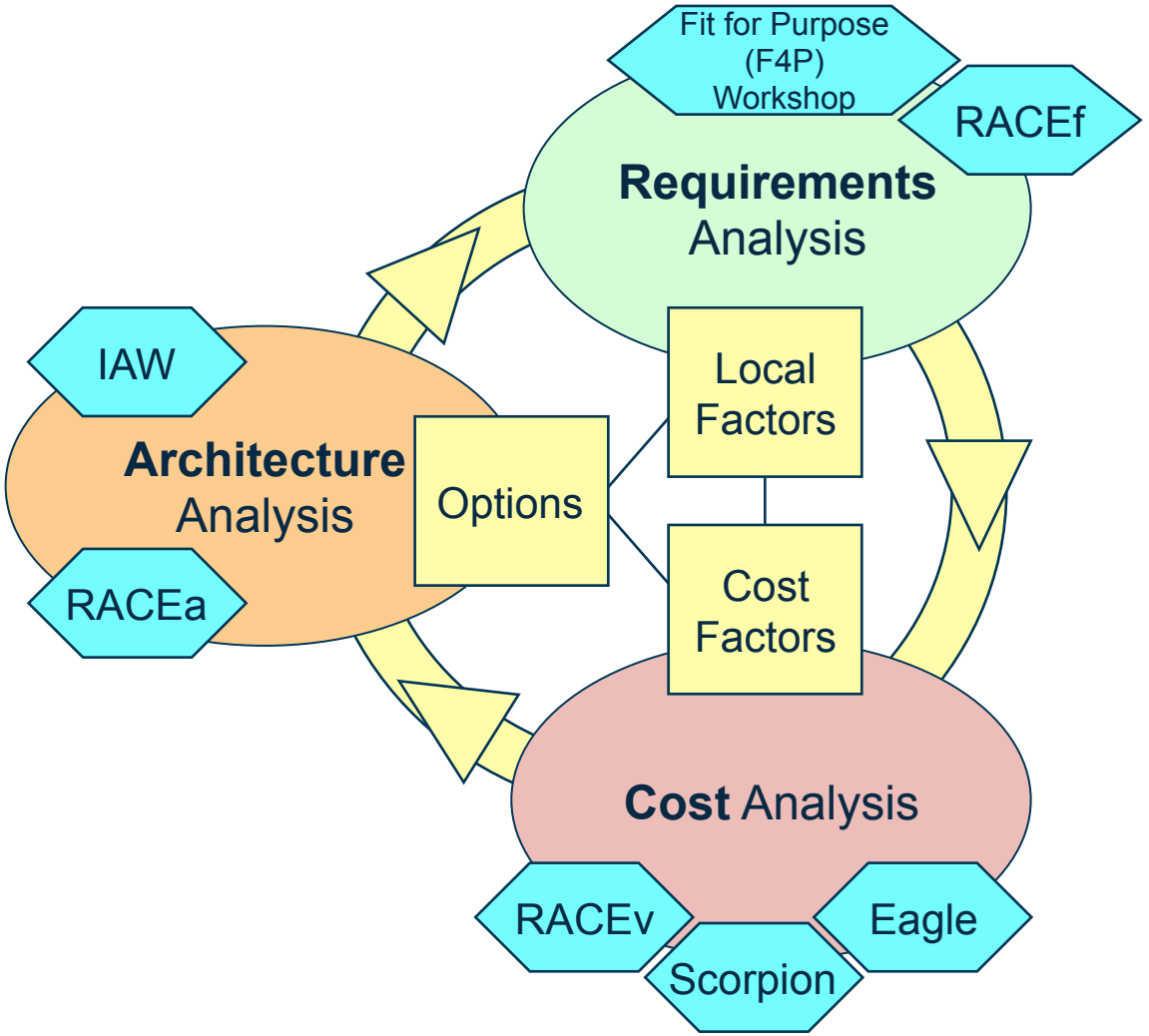
Scorpion

- IBM Global Business Service
- Consulting Engagement
- CIO/budget & down analysis
- Report for the CIO

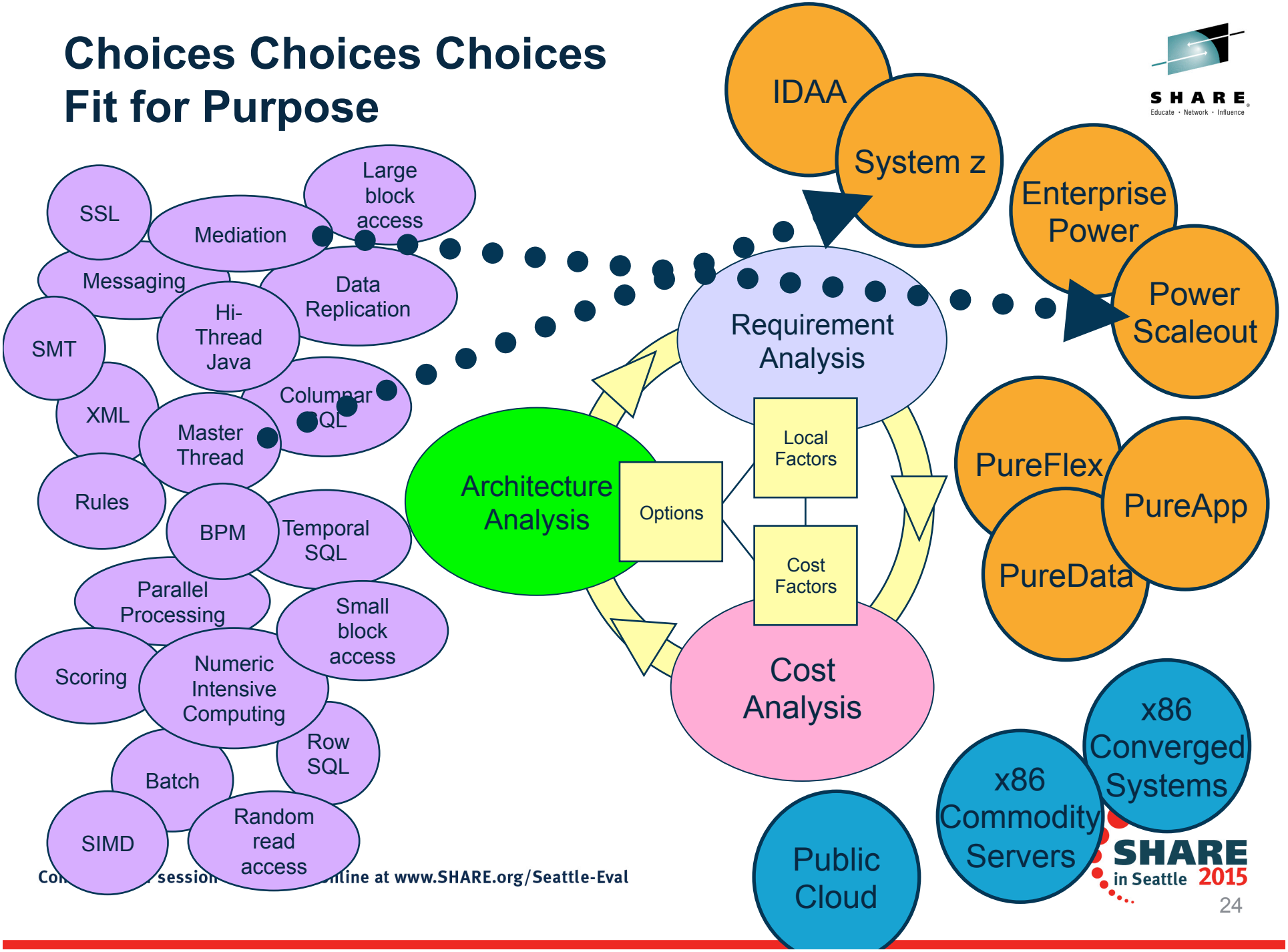
Eagle

- Consulting Engagement
- Bottom-up technical and TCO analysis
- TCO Scorecard

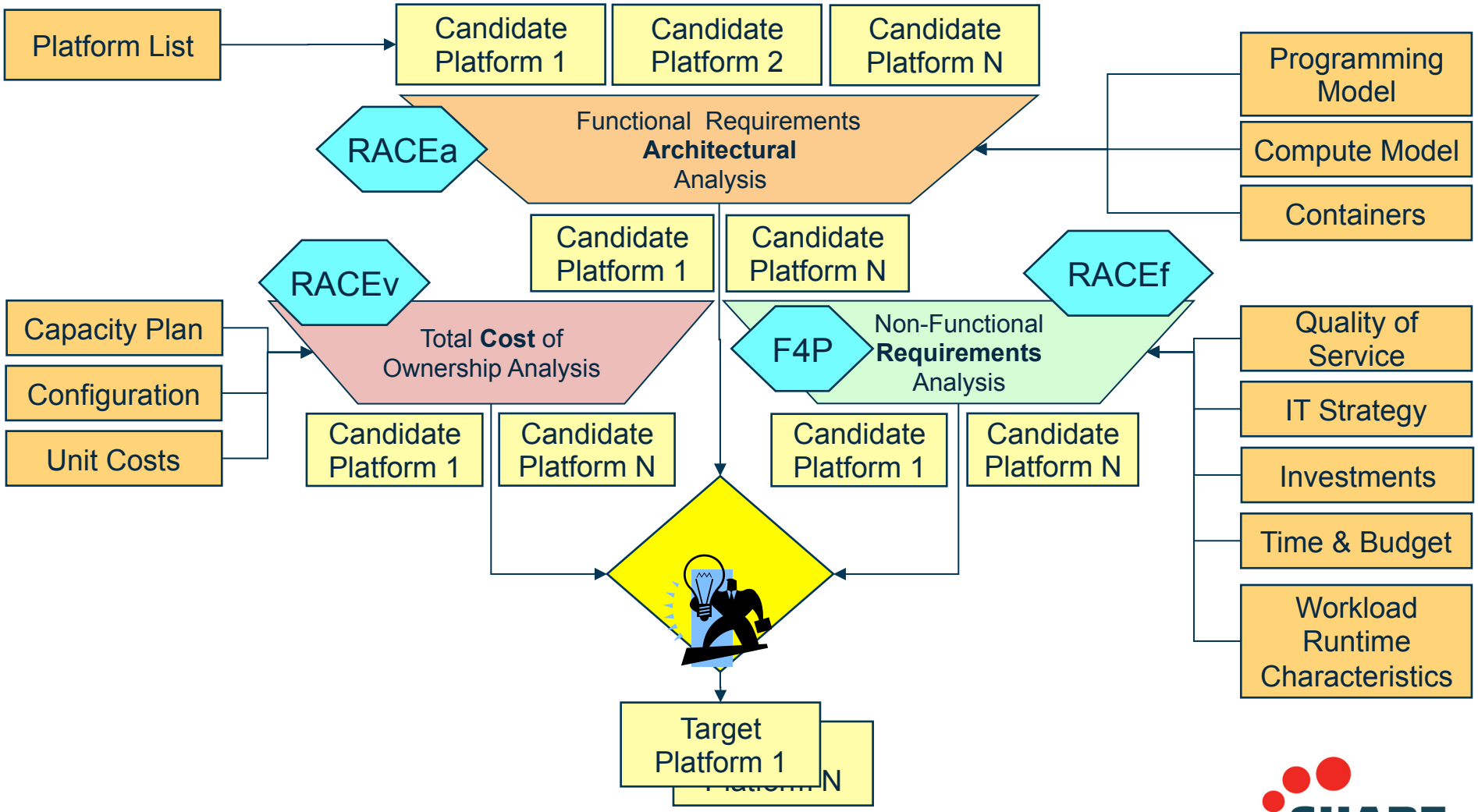
IT Optimization Assessment Methodology
Three Stages



Choices Choices Choices
Fit for Purpose



Workload Placement IT Optimization Process

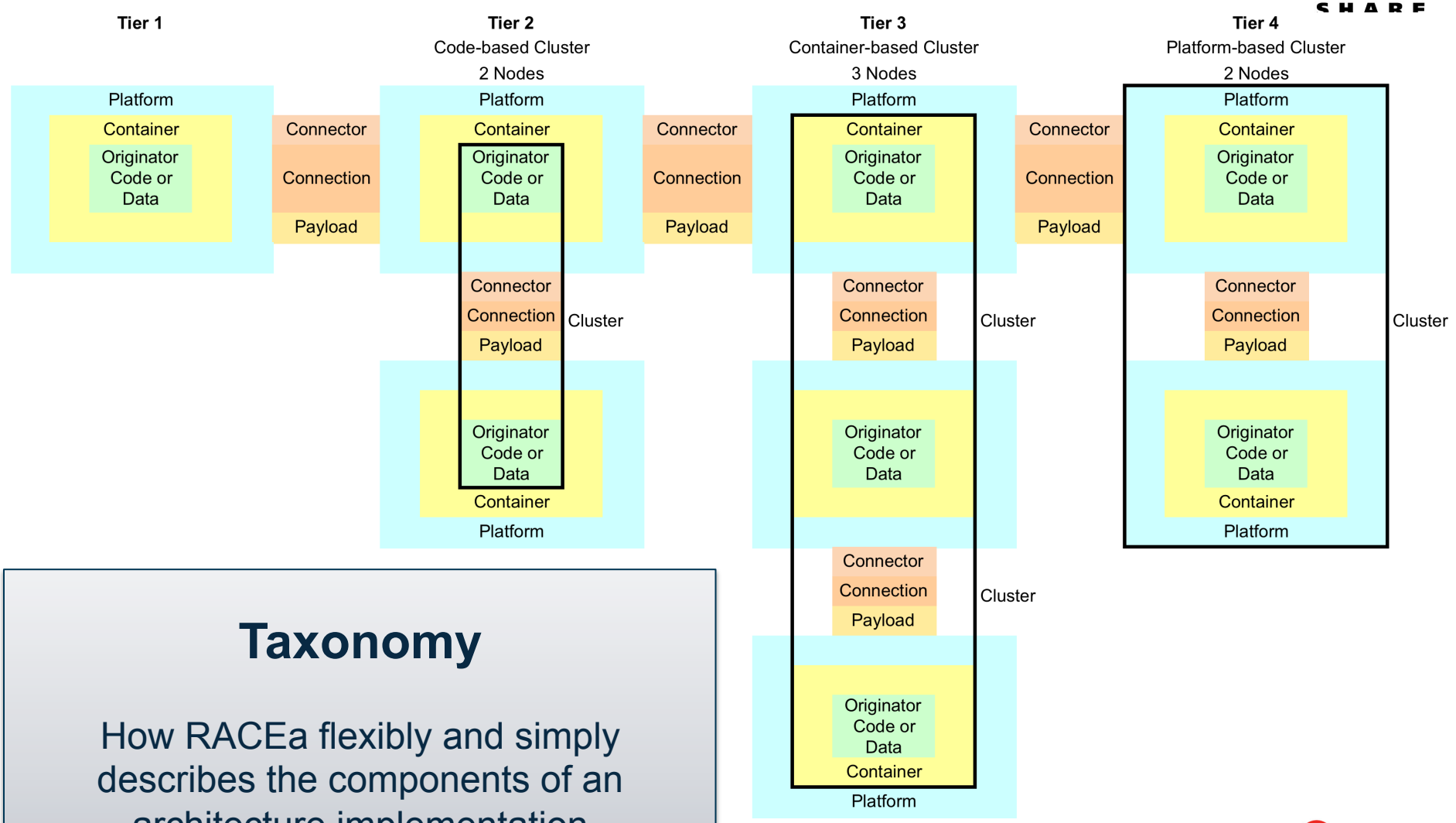


Architecture Analysis with RACEa

RACEa - Step by Step

1. Understand RACEa's architecture taxonomy
2. Document project description
3. Describe project attributes
4. Calibrate scoring attributes and tables
5. Setup architecture component relationships
6. Describe architecture implementation one
7. Describe architecture implementation two (three/four)
8. Review output reports
9. Implement the optimal architecture implementation

RACEa's Architecture Taxonomy



Taxonomy

How RACEa flexibly and simply describes the components of an architecture implementation

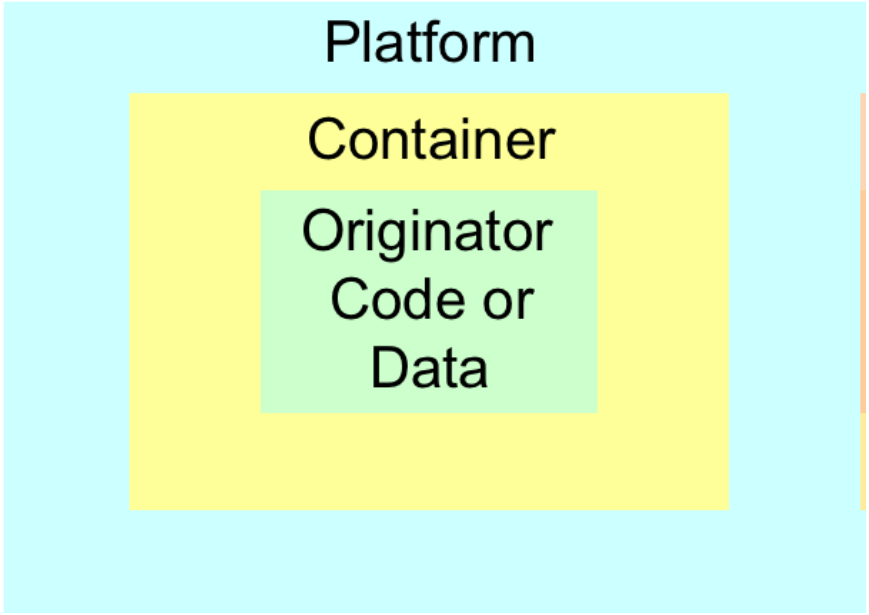
Taxonomy: A Node in the Architecture…

Node

- **Code** node contains code
- **Data** node contains data
- **Originator** node contains a device or sensor or other “internet of things” thing

- **Container** “holds” the code or data
- Usually middleware like WAS or DB2 or Apache

- **Platform** “holds” the container
- Usually a combo of hardware and hypervisor (optional) and operating system



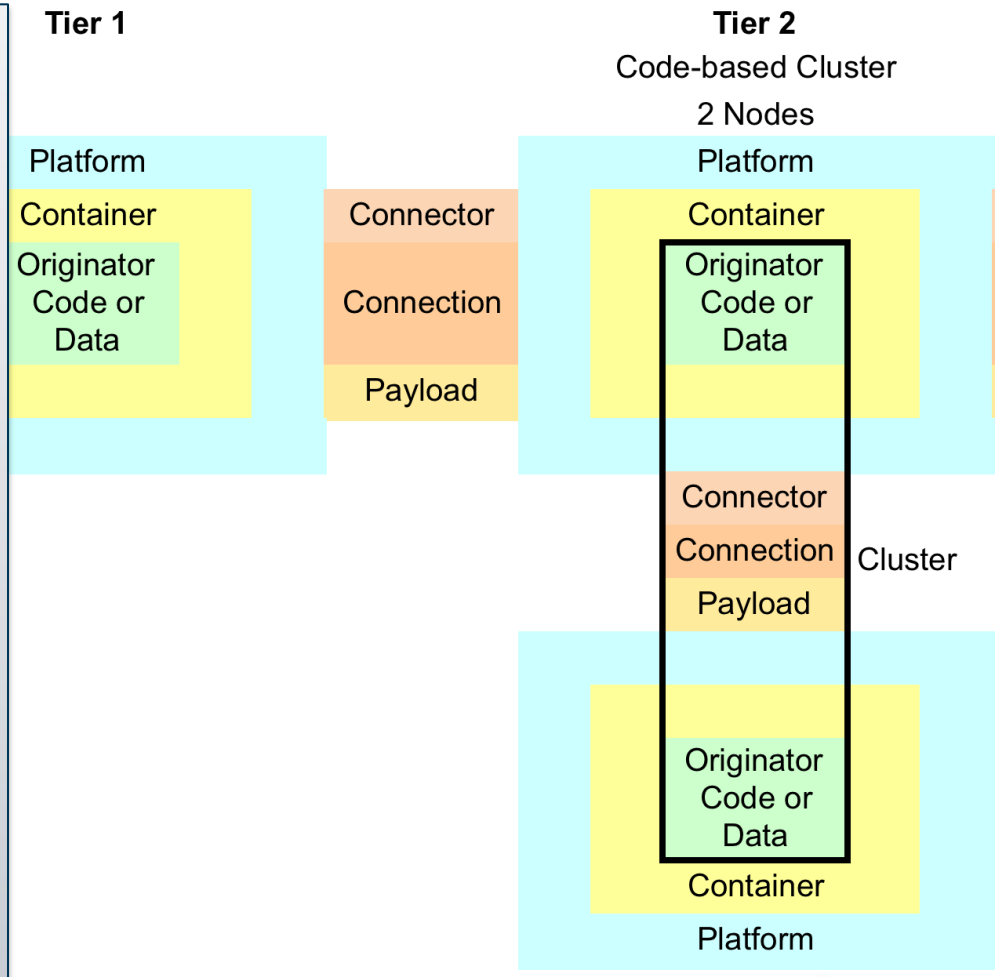
Taxonomy: Clusters and Connectors

Connections

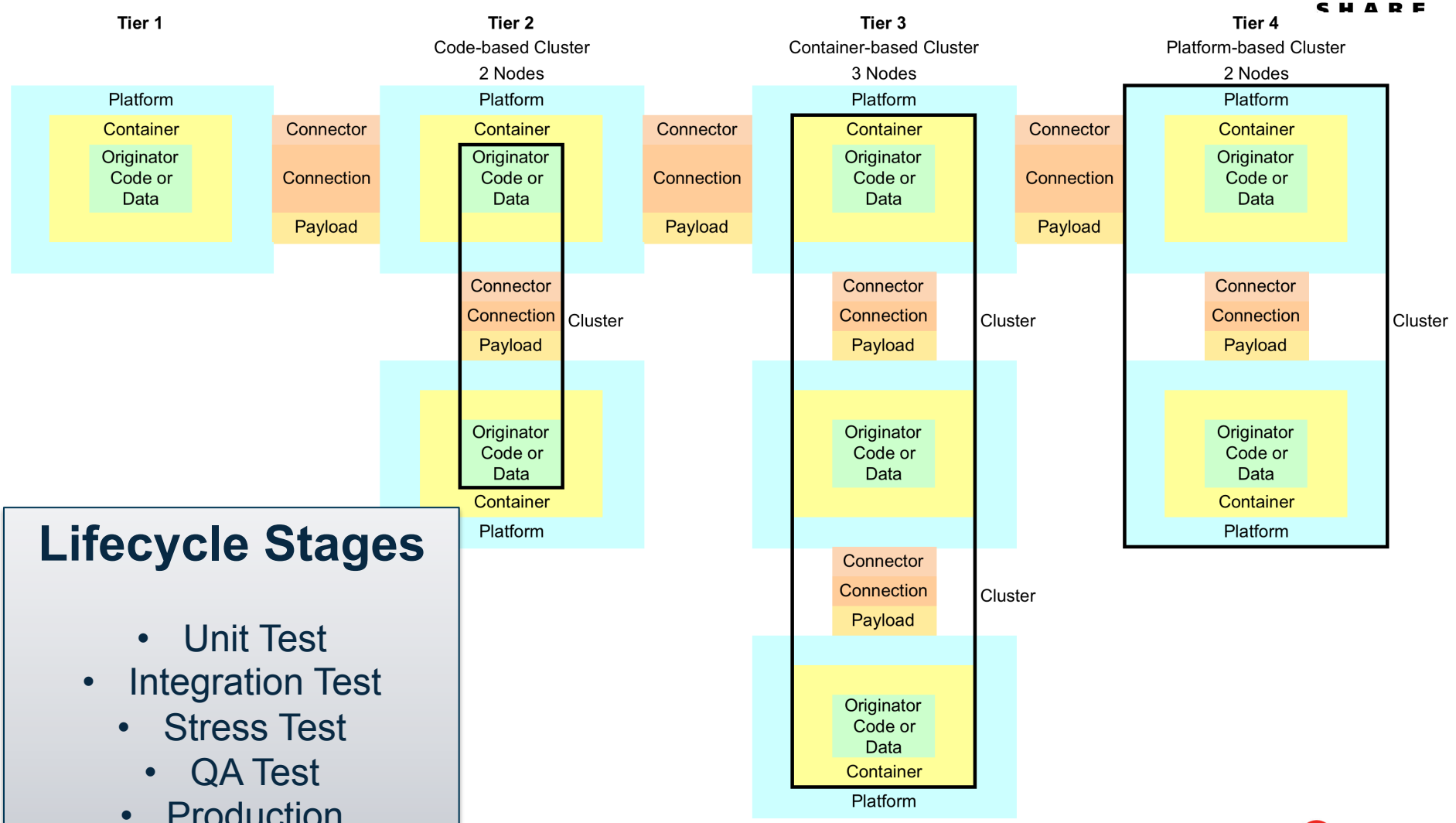
- Container interactions
- How nodes connect
- How fast / how distant
- How much data flows and how often

Clusters

- Type of cluster
- Code cluster
- Container cluster (like RAC)
- Platform cluster (like Sysplex)
- Type of cluster connection
- Local Area Network
- Coupling Links / Coupling Facility



Taxonomy: "Production" and …



Lifecycle Stages

- Unit Test
- Integration Test
- Stress Test
- QA Test
- Production
- DR

Taxonomy Elements

- **Originator**
  - Mobile device, internet of things thing, browser, etc.
- **Code**
  - HTML, Java, COBOL, config-files, etc.
- **Data**
  - Rows (tables), records (files), streams, etc.
- **Container**
  - Middleware
- **Platform**
  - Server, hypervisor (optional), and operating system (typically)
- Inbound and Outbound **Connector**
  - Payload and Inter-Node Invocation Frequency
- Inbound and Outbound **Connection**
  - Type and Distance
- **Cluster** Type
  - Code cluster
  - Container cluster
  - Platform cluster
  - Inter-Cluster Communication

Project Attributes

- Project description
  - Nature of project / purpose of analysis
- Custom one-off project
  - In which case automation (provisioning/orchestration) is not important
- Pattern-based highly replicated project
  - In which case automation is essential
- Something in between
  - In which case automation is important, but not essential
- Ample opportunities exist for localizing the tool's merit and complexity scoring system in simple weighted scoring tables

Scoring Attributes and Tables

- Element Ownership
- Element Disposition
- Element Provisioning Source
- Element Deployment Technique
- Quality of Service Confidence

	Merit Score Weights		
	Project Type		
	Highly Custom One-of-a-Kind Project	Project with both pattern based and one of a kind componentry	Highly automatable highly replicated pattern-based project
Assessment Attributes	**1**	2	**3**
Element Ownership	4	4	4
Element Disposition	7	7	7
Element Provisioning Source	5	6	7
Element Deployment Technique	4	7	9
Quality of Service Confidence	10	9	8

Scoring Attributes – Element Ownership

- Element Ownership
  - Corporate
  - Partner
  - Customer
  - Vendor/Supplier

Assessment Attribute 1: Element Ownership	Code	Data	Conta
Corporate	10	10	10
Partner	5	5	5
Customer	1	1	1
Vendor/Supplier	7	5	5

Scoring Attributes – Element Disposition

- Element Disposition
  - New
  - Extended
  - Existing / Shared

Assessment Attribute 2: Element Disposition	Code	Data	Container
New	5	5	5
Extended	7	7	7
Existing / Shared	10	10	10
n/a	0	0	0

Scoring Attributes – Element Provisioning Source

- Element Provisioning Source
  - Whitespace-Pool
  - Upgrade-Pool
  - New
  - Provisioning Not Required

Assessment Attribute 3: Element Provisioning Source	Code	Data	Containe
Whitespace-Pool	8	8	8
Upgrade-Pool	6	6	6
New	4	4	4
Provisioning Not Required	10	10	10

Scoring Attributes – Element Deployment Technique

- Element Deployment Technique
  - Custom
  - Pattern-Based
  - Orchestrated-Pattern
  - Deployment Action Not Required

Assessment Attribute 4: Element Deployment Technique	Code	Data	Container
Custom	4	4	4
Pattern-Based	6	6	6
Orchestrated-Pattern	8	8	8
Deployment Action Not Required	10	10	10

Scoring Attributes – QoS Confidence

- Quality of Service Confidence
  - High Confidence
  - Medium Confidence
  - Low Confidence
  - No Confidence

Assessment Attribute 5: Quality of Service Confidence	Code	Data	Container
High Confidence	10	10	10
Medium Confidence	7	7	7
Low Confidence	4	4	4

Architecture Component Relationships

- Customizable Configuration Tables
  - Use Case 1 – describe what your enterprise supports
    - For application development lifecycle support
  - Use Case 2 – describe what is possible
    - For enterprise architecture development
- Setup valid relationships between elements
  - Populate drop-down selection lists
  - The containers that can hold code
    - e.g. “WAS-ND” can hold “Java”
  - The containers that can hold data
    - e.g. “DB2” can hold “Row” (or “Table”, if you prefer)
  - The platforms that can hold containers
    - e.g. “z/OS on zEC12” can hold “DB2”
  - The connectors that connect containers
    - e.g. “WAS-ND” supports “JCA”
  - The connections that support connectors
    - e.g. “JCA” can flow over “Local_LAN”

Code, Data, or Originator Element	**Candidate Containers…**			
Java	WAS-ND	Liberty	CICS	Tomcat
COBOL	CICS	JES	IMS	DB2-SP

Implementation One …

- Map the logical architecture to a physical architecture
- For each node:
  - Choose code (or data) to use
  - Choose containers to use
  - Choose platforms to use
  - Choose connectors to use
  - Choose connections to use
  - Choose clustering and cluster connections to use
  - For each element … pick assessment attribute4s
  - Got another node? … add one! (add as many as you like!!!)
- Describe the physical architecture for production
  - Describe for test, QA, etc. (by adding more and more nodes)
- Concentrate on the things that vary between options
- **Build the 1st … then up to 3 more (total of 4) … one per sheet**

Implementation One …

	1	2
Tier Label	WAS Layer	CICS Layer
Description (Optional)	WAS	CICSPlex
Application Development Lifecycle Stage	Production	Production
Tier Quality of Service Requirement	High Quality of Service	High Quality of Service
Cluster Type	Not a Cluster	Container Cluster
Number of Nodes in Cluster	0	2
Intra-Cluster Communication	None	System z Coupling Facility
Tier Type (Originator/Code/Data)	Code	Code
Originator/Code/Data Selection	Java	COBOL
Container Selection	WAS-ND	CICS
Platform Selection	Windows-VM-x86	z/OS
Inbound Connector Selection		JCA
Inbound Connection Selection		Local_LAN
Inbound Average Payload Size		Small
Outbound Connector Selection	JCA	
Outbound Connection Selection	Local_LAN	
Outbound Average Payload Size	Small	
Average Processing Load Estimate	Medium-Light	Medium
Average Invocations of This Tier Per Transaction (Per Execution)	1	4

Analysis Output Report

- After two or more architecture implementations are described you can compare them on the “ScoreCard”

- ScoreCard Elements:
  - Relative Overall Merit Score
  - Relative Complexity Score
  - Relative Processing Burden Score
  - Relative Networking Burden Score
  - Relative Bill of Materials Report

Overall Merit Scoring

- Blended (weighted) score
  - Based upon each component's:
    - Ownership selection
    - Disposition selection
    - Provisioning Source selection
    - Deployment Technique selection
    - Quality of Service Confidence selection
- Provides "relative" indicator of merit
  - Indeed … all "scores" in this tool are "relative"
    - And only of value used within the tool to compare options

Complexity Scoring

Assessment Attribute 2: Element Disposition	Complexity
New	10
Extended	6
Existing / Shared	4
n/a	0

Assessment Attribute 3: Element Provisioning Source	Complexity
Whitespace-Pool	2
Upgrade-Pool	6
New	10
Provisioning Not Required	0

Assessment Attribute 4: Element Deployment Technique	Complexity
Custom	10
Pattern-Based	7
Orchestrated-Pattern	4
Deployment Action Not Required	0

Processing Burden Scoring

Processing Load Table	
Unknown	0
Light	1
Medium-Light	10
Medium	100
Medium-Heavy	1000
Heavy	10000

Network Burden Scoring

- Calibrating the “cost” of networking
- Remembering … the best networking is NO networking

Connections						
Type	**SubType**	**Distance**	**Parameter Passing**	**Short Description**	**Score**	**Network Load**
Shared Memory	Intra-Process Call/Return	n/a	Parameter Passing by Reference	Call_by_Reference	10	0
			Parameter Passing by Value	Call_by_Value	9.9	0
	Inter-Process Call/Return		Parameter Passing by Value	Inter-Process_Call_by_Value	9.5	0
	Message Queue Put/Get		Parameter Passing by Value	Local_Message_Queue	9	0
Network	LAN	Local_IP_Stack_Reflection	Parameter Passing by Value	Local_IP_Stack_Reflection	8	1
		Local_Virtual_LAN	Parameter Passing by Value	Local_Virtual_LAN	7	10
		SMC-R_Local_LAN	Parameter Passing by Value	SMC-R_Local_LAN	7.5	15
		Local_Shared_NIC	Parameter Passing by Value	Local_Shared_NIC	6	20
		Local_LAN	Parameter Passing by Value	Local_LAN	5	100
		Metro_LAN	Parameter Passing by Value	Metro_LAN	4	200
	WAN	Metro_WAN	Parameter Passing by Value	Metro_WAN	4	1000
		Regional_WAN	Parameter Passing by Value	Regional_WAN	3	1500
		National_WAN	Parameter Passing by Value	National_WAN	2	2000
		International_WAN	Parameter Passing by Value	International_WAN	1	2500

Network Burden Scoring (cont.)

Network Payload Size Table			
Size	Bytes	Kbytes	Mbytes
Very Small	1024	1	0.00
Small	4096	4	0.00
Medium	16384	16	0.02
Large	262144	256	0.25
Very Large	2097152	2048	2
Huge	16777216	16384	16

Bill of Materials

- List of all elements composing the architecture
- What's new list
- What's extended list
- What's reused (shared) list

ScoreCard

Architectural Analysis	Topology ONE	Topology TWO	Topology THREE	Topology FOUR
Name	WinTel	zLinux	AIX	z/OS
Description	WAS on Windows on x86	WAS on zLinux	WAS on AIX	WAS on z/OS
Cluster Scoring	303	303	303	303
Originator/Code/Data Scoring	548	548	548	548
Container Scoring	534	561	534	561
Platform Scoring	534	561	561	561
Inbound Connector Scoring	190	190	190	190
Outbound Connector Scoring	190	190	190	190
Inbound Connection Scoring	257	281	257	309
Outbound Connection Scoring	257	257	257	309
Total Architectural Score	**2813**	**2891**	**2840**	**2971**
Architectural Rank	**4**	**2**	**3**	**1**
Normalized Architectural Score	**0.947**	**0.973**	**0.956**	**1.000**

Complexity Analysis	Topology ONE	Topology TWO	Topology THREE	Topology FOUR
Cluster Scoring	28	28	28	28
Originator/Code/Data Scoring	24	24	24	24
Container Scoring	27	27	27	27
Platform Scoring	27	27	27	27
Inbound Connector Scoring	0	0	0	0
Outbound Connector Scoring	200	200	200	200
Inbound Connection Scoring	127	91	127	42
Outbound Connection Scoring	127	127	127	42
Total Complexity Score	**560**	**524**	**560**	**390**
Complexity Rank	**1**	**3**	**1**	**4**
Normalized Complexity Score	**0.188**	**0.176**	**0.188**	**0.131**

ScoreCard (cont.)

Load Analysis	Topology ONE	Topology TWO	Topology THREE	Topology FO
Networking Load	1638400	16384	1638400	0
Processing Load	410	410	410	410
Total Complexity Score	**1638810**	**16794**	**1638810**	**410**
Complexity Rank	**1**	**3**	**1**	**4**
Normalized Complexity Score	**551.602**	**5.653**	**551.602**	**0.138**

Bill of Materials Lists	Topology ONE	Topology TWO	Topology THREE	Topology FOU
New	Java + WAS-ND + Windows-VM-x86 + JCA	Java + WAS-ND + zLinux + JCA	Java + WAS-ND + AIX-Power-LPAR + JCA	Java + WAS-ND + z JCA
Unique New Elements Count	4	4	4	4
Total New Elements Count	5	5	5	5
Extended	* + Local_LAN	* + Local_IP_Stack_Reflection	* + Local_LAN	* + Inter-Process_Call_by_V
Unique Extended Elements Count	1	1	1	1
Total Extended Elements Count	2	2	2	2
Existing / Shared	* + COBOL + CICS + z/OS	* + CICS + z/OS	* + COBOL + CICS + z/OS	* + COBOL + CICS +
Unique Existing / Shared Elements Count	3	2	3	3
Total Existing / Shared Elements Count	3	2	3	3
Unique Number of Elements	8	7	8	8
Total Number of Elements	10	9	10	10

Conclusion

Keys to Success

- Look at the options (for projects (for patterns))
- Understand the options (what you have (what you could have)
- Pick the right option (For the right reason)
- Systematically
  - Adaptable
    - To technology as it evolves (change is constant)
    - To “local factors” as they evolve (change is constant)
  - Repeatable
  - Facilitate understanding, teaming, and learning

Executing a RACEa Workshop

- Architecture Analysis - RACEa Workshop
  - Typically one-half to one day on-site tooling-facilitated no-charge workshop
  - With application architects and platform architects & others
    - (1) define logical system architecture
    - (2) define rendition 1 architecture implementation
    - (3) define rendition 2 (3/4) architecture implementation(s)
    - (4) calibrate scoring tables
    - (5) examine results, discuss, calibrate, loop
    - (6) finalize results and report

Next Steps

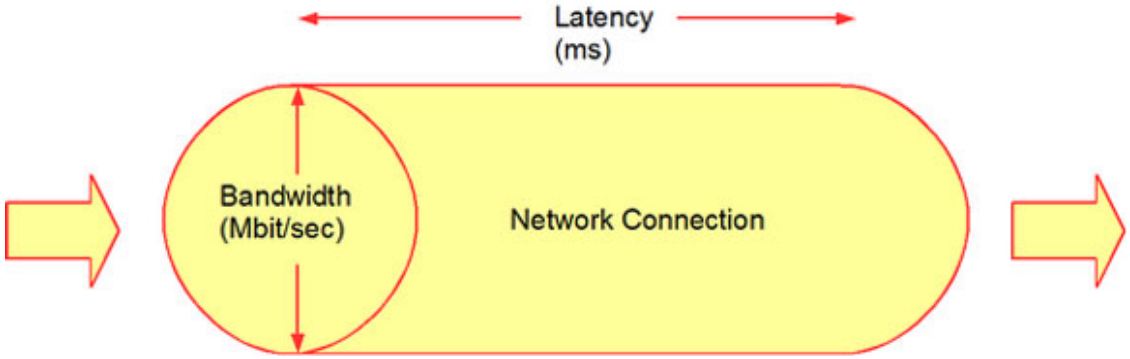
- RACEg
  - TCO Analysis Tool
  - beta testers needed
- RACEf
  - Requirements-Based Platform Selection Tool
  - beta testers needed
- RACEa
  - Architecture Analysis Tool
  - **beta testers needed**
  - **need calibration data / network & processing burden data**

The Final Chart

- Any questions?
- Any suggestions?
- Any way I can be of service?

  - Monte Bauman
  - Enterprise Server Technical Support
  - IBM Columbus
  - mbauman@us.ibm.com

Network Latency Matters



The Objective: Determine the actual latency incurred when making off-platform calls

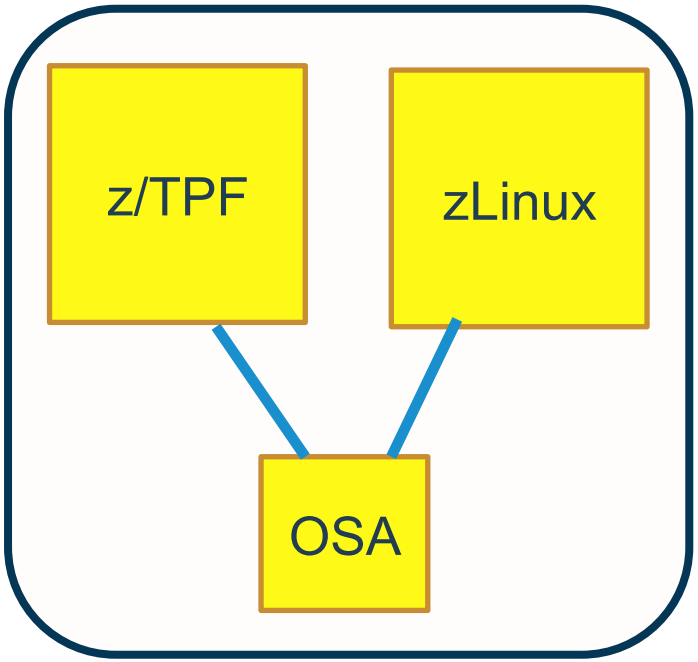
This study provides response time measurements for two simple TCP/IP configurations.

Two System z LPARs on the same zEC12 server share an OSA-Express adapter in the 1st measurement.

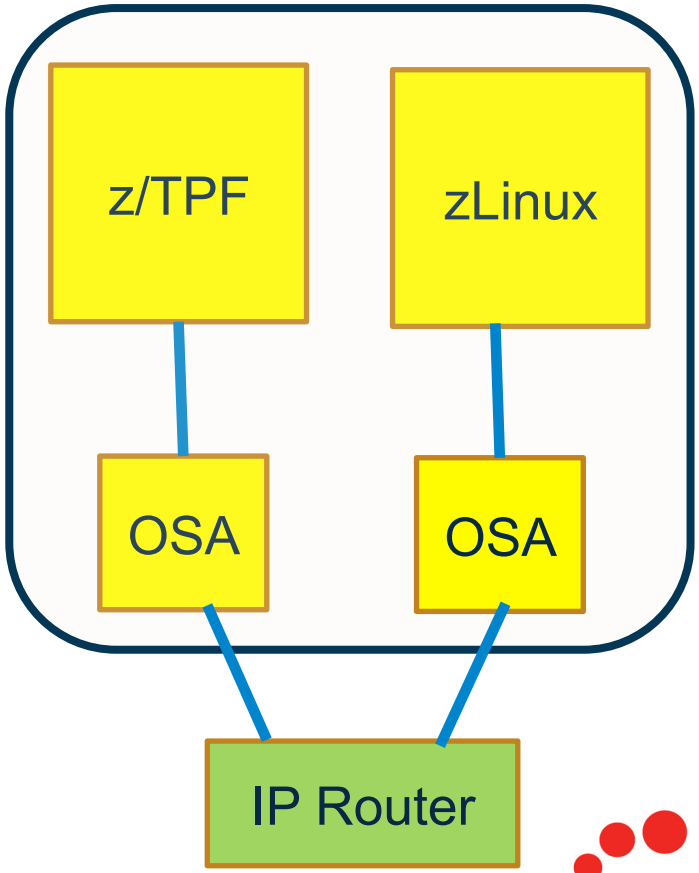
In the second, the same two LPARs use two different OSA cards connected to the same router (the LPARs are one network hop away)

Test Environment Comparison

Shared OSA Configuration



One-Hop Route Configuration



Controlled Test Environment for Apples to Apples Comparison

Server Constants

- Same zEC12 processor was used for all tests
  - z/TPF LPAR with one dedicated CP
  - zLinux LPAR with one dedicated IFL
- Same driver was used in all tests
  - Same number of driver instances was run for each comparison test
- The only difference in a given comparison test was the network path used

Message Driver Input

- The number of driver instances to start
- The message size (This is the amount of user data in each request message and each response message)
- The delay factor (which is how long to wait after receiving a response before sending the next request message)
- Number of messages to send before the driver exits

Each Instance of the Driver Does What

- Starts a long running TCP socket
- Loops N times doing the following:
  - Save current time (T1)
  - Issue socket send() API to send request message of size X
  - Issue socket read() API to read the response message
  - Get current time (T2) and calculate round trip time (RTT) for this request/response message pair (T2-T1) and then adjust the average RTT
    - **This is the RTT from the application perspective – this is not the network (TCP) RTT**
  - Sleep for a user specified amount of time
- Ends the socket

Round Trip Time - Test Results

100-Byte Message Test

Message Rate (Messages/Second)	Shared OSA RTT (microseconds)	1-Hop Route RTT (microseconds)	RTT Ratio
10,000	159	275	1.73
20,000	181	345	1.90
30,000	216	553	2.56
40,000	260	724	2.78

500-Byte Message Test

Message Rate (Messages/Second)	Shared OSA RTT (microseconds)	1-Hop Route RTT (microseconds)	RTT Ratio
5,000	154	439	2.85
10,000	161	443	2.75
15,000	170	448	2.63
20,000	181	455	2.51

1400-Byte Message Test

Message Rate (Messages/Second)	Shared OSA RTT (microseconds)	1-Hop Route RTT (microseconds)	RTT Ratio
2000	150	915	6.10
4000	153	920	6.01
6000	155	928	5.99
8000	159	934	5.87

5000-Byte Message Test

Message Rate (Messages/Second)	Shared OSA RTT (microseconds)	1-Hop Route RTT (microseconds)	RTT Ratio
500	550	2120	3.85
1000	551	2120	3.85
1500	550	2130	3.87
2000	552	2394	4.33

10,000-Byte Message Test

Message Rate (Messages/Second)	Shared OSA RTT (microseconds)	1-Hop Route RTT (microseconds)	RTT Ratio
250	401	2530	6.31
500	467	2547	5.45
750	390	2556	6.55
1000	390	2564	6.57

20,000-Byte Message Test

Message Rate (Messages/Second)	Shared OSA RTT (microseconds)	1-Hop Route RTT (microseconds)	RTT Ratio
125	362	4288	11.88
250	361	4344	12.00
375	367	4377	11.93
500	371	4390	11.83

Summary (Shared OSA vs 1-Hop Route)

Message Size	Average RTT Ratio	Average Extra Time Per Message (microseconds)
100	2.2	270
500	2.7	280
1400	6.0	292
5000	4.0	1640
10,000	6.2	2137
20,000	11.9	3984