

# Innovative Practices for Effective Performance Testing

*Enabling Systems for High Performance, Availability and Stability*

**Arunmozhiverman S**

**Polaris Application Certification Enterprise (PACE)**

# Agenda



- Performance Engineering - Trends & Challenges
- Performance Engineering - Misconceptions
- Performance Testing - Things to Address
- Performance Engineering Tools
- Performance Test Architecting & Execution Process
  - Complex Message Processing System
- Performance Testing - Engagement Models
  - Shared Services Model, OLTP Performance Testing
- Overall Performance Engineering Process

# Trends & Challenges



## Application Characteristics

- Compute intensive
- Platform independence
- Increased functionalities, flexibility
- Interfaces & integrations
- GUIs

## Technology Transformations

- Legacy migrations
- Infrastructure upgrades
- Software versions

## Business Trends

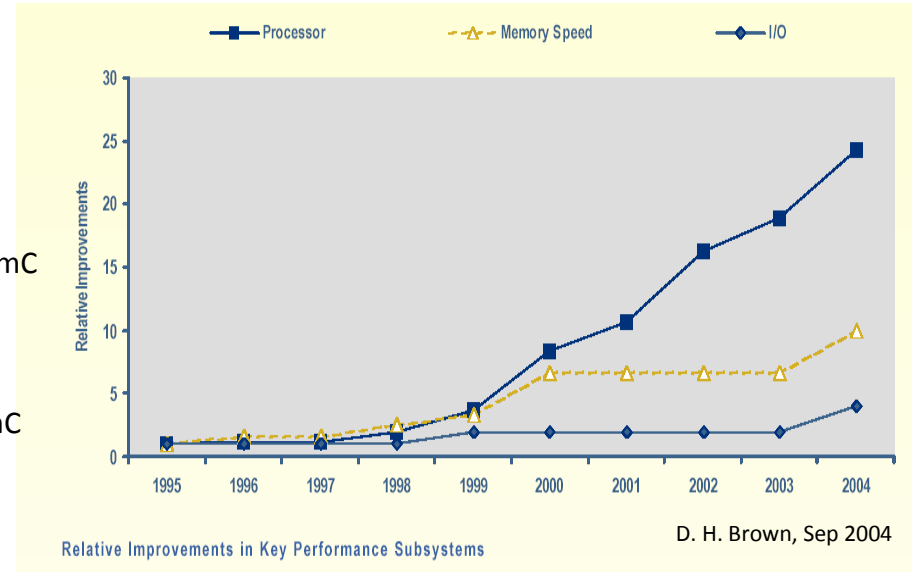
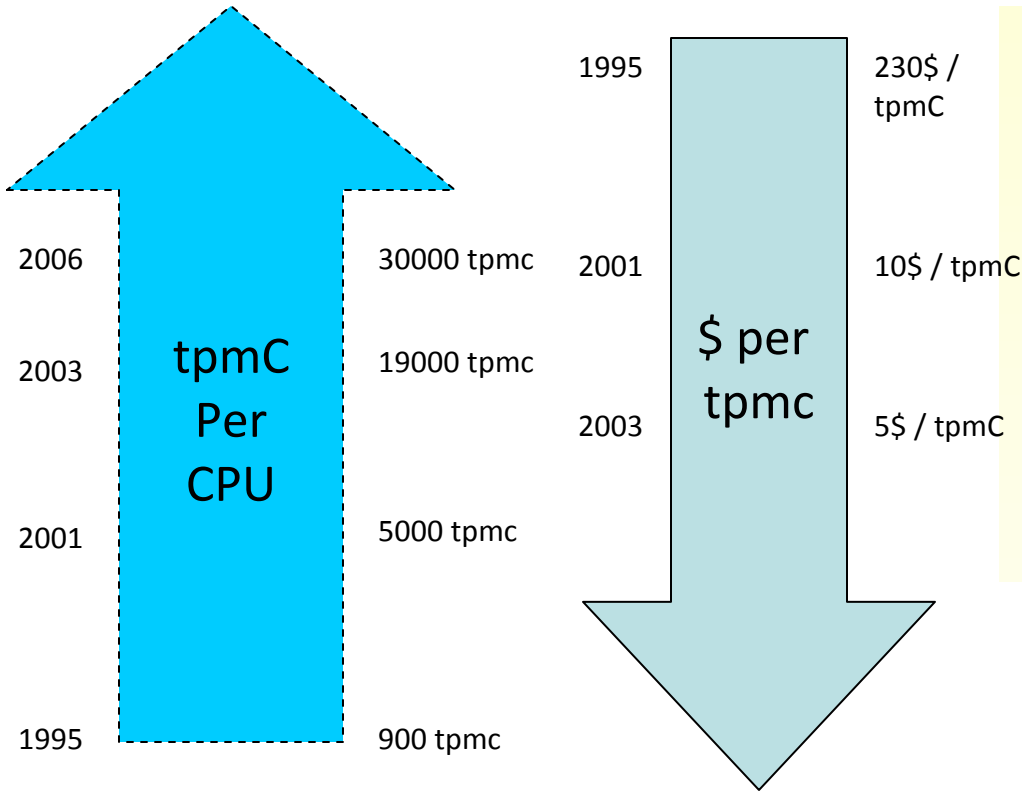
- Acquisitions
- Highs and lows
- Seasonal fluctuations

# Performance: Misconceptions

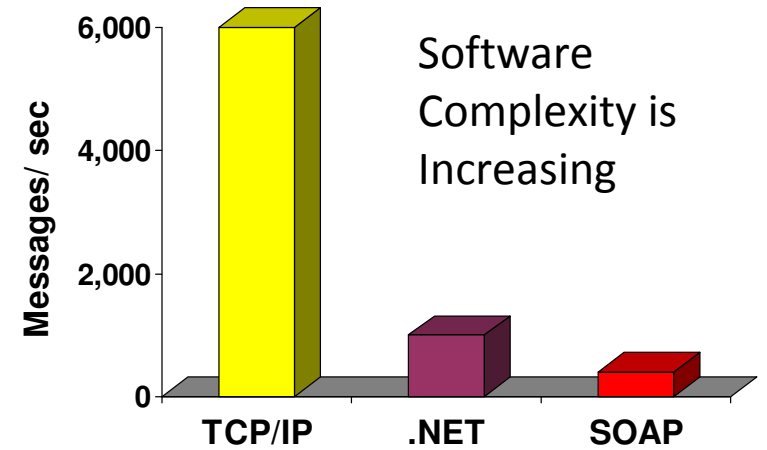


- Hardware upgrade is the easiest approach
- Performance is a Rocket Science, very hard to tackle
- 1 Day tuning when I face issues should be OK
- Is it Performance Problem – Hey, it is not mine then
- New technologies (J2EE, .NET, Oracle scalability) = Performance
- Performance Testing (LoadRunner, QA Load, etc) will solve all my problems
- Performance Optimization → Highly Complex
- Technically Strong People → High Performance
- Non availability of right hardware for tests = Easy Extrapolation

# Hardware & Performance Maturity



**RAM: MB to GB**      **Disk: GB to TB**  
**LAN: Mbps to Gbps**      **WAN: Kbps to Mbps**



# Performance Testing - Things to Address



- How do I define a scenario that is not known to begin with?
  - Is it only one or there many?
  - Do I need to take care of any abnormalities
  - How do the simulation scenarios differ in the following test types
    - Load
    - Stress / Endurance
    - Scalability
    - Volume
    - Etc
  - Are there any recovery conditions
  - Ultimately, Is it what I am observing in the production environment
- Pre Production:
    - Decision Trees
    - Parameter Identification thru evaluation of partition rules
  - Post Production Info: Operational distribution

# Performance Testing: Things to Address



- **Appropriate test duration - Transient vs. Steady State**
- **Choosing the right hardware / software configurations**
- **WAN and LAN simulations (Tools like Shunra)**
- **Populate the right amount of data (for volume testing, consider 2x / 3x growth)**
- **Results Validation Mechanism**
- **More importantly, right emphasis on effort estimation and schedules**

# Performance Engineering Tools



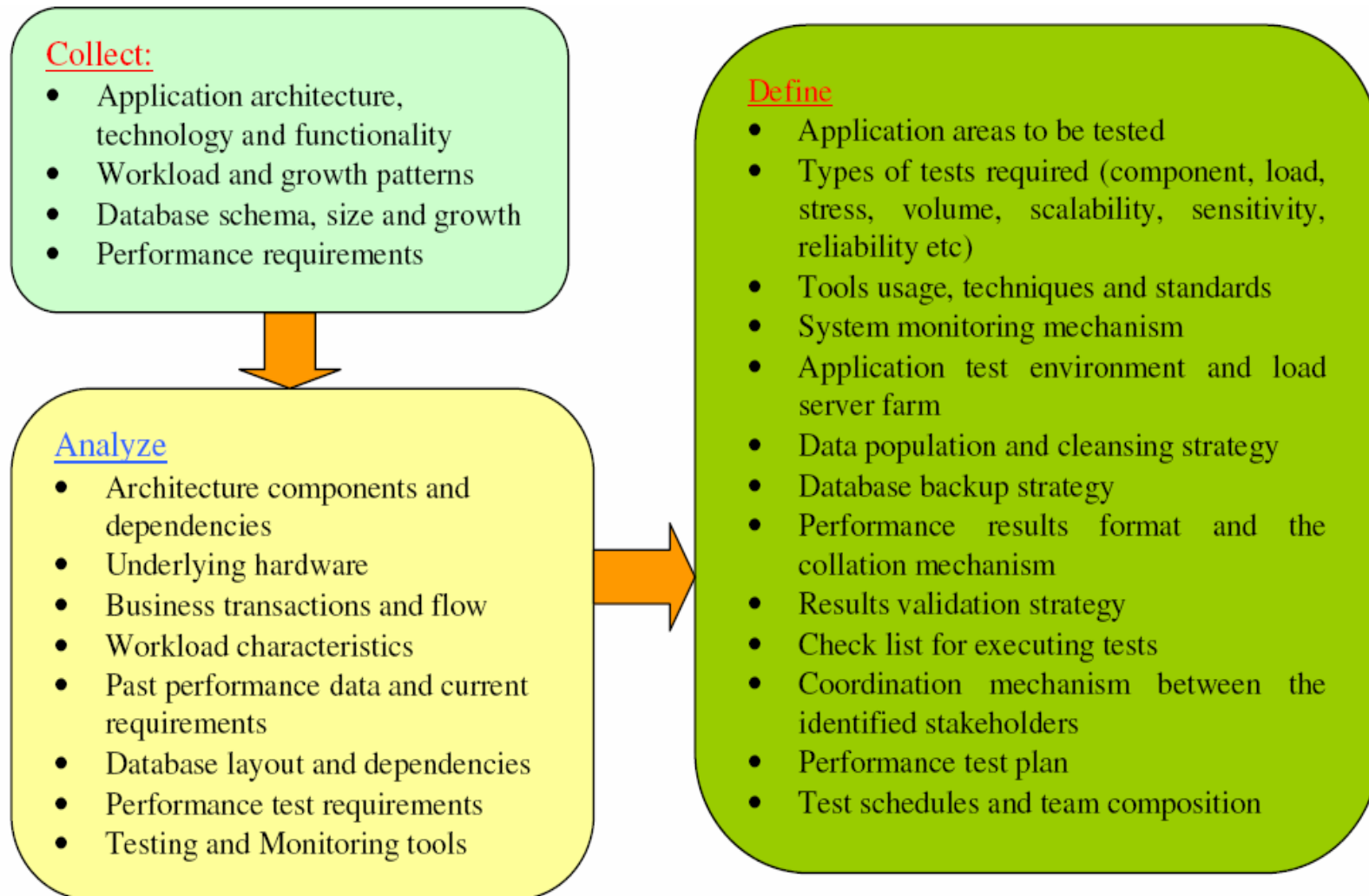
live your dream

Technologies		Perf Testing	Profiling	Monitoring
Web	Thin	LoadRunner, Rational, Silk, Compuware, Opensta	Mercury Deep Diagnostics, Wily, PerfMon, Veritas, Optimizelt, Log Analyzers, App Tools	BAC, Sitescope, MAM, HP OpenView Wily, IBM Tivoli, Acsera, Web trends Veritas, Statspack, OS tools
	Thick	LoadRunner		
Application	J2EE, .NET	LoadRunner, Compuware	Veritas, App Tools	
	Oracle Apps, Siebel	LoadRunner, Compuware		
	Custom	LoadRunner, Req. Injector		
Database	DB2, Oracle, SQL Server	LoadRunner, Custom	Statspack, OEM, Veritas, DB Artisan, Sitescope	
Messaging	MSMQ, IBM MQ, TIBCO etc	LoadRunner	Msg. Tools	

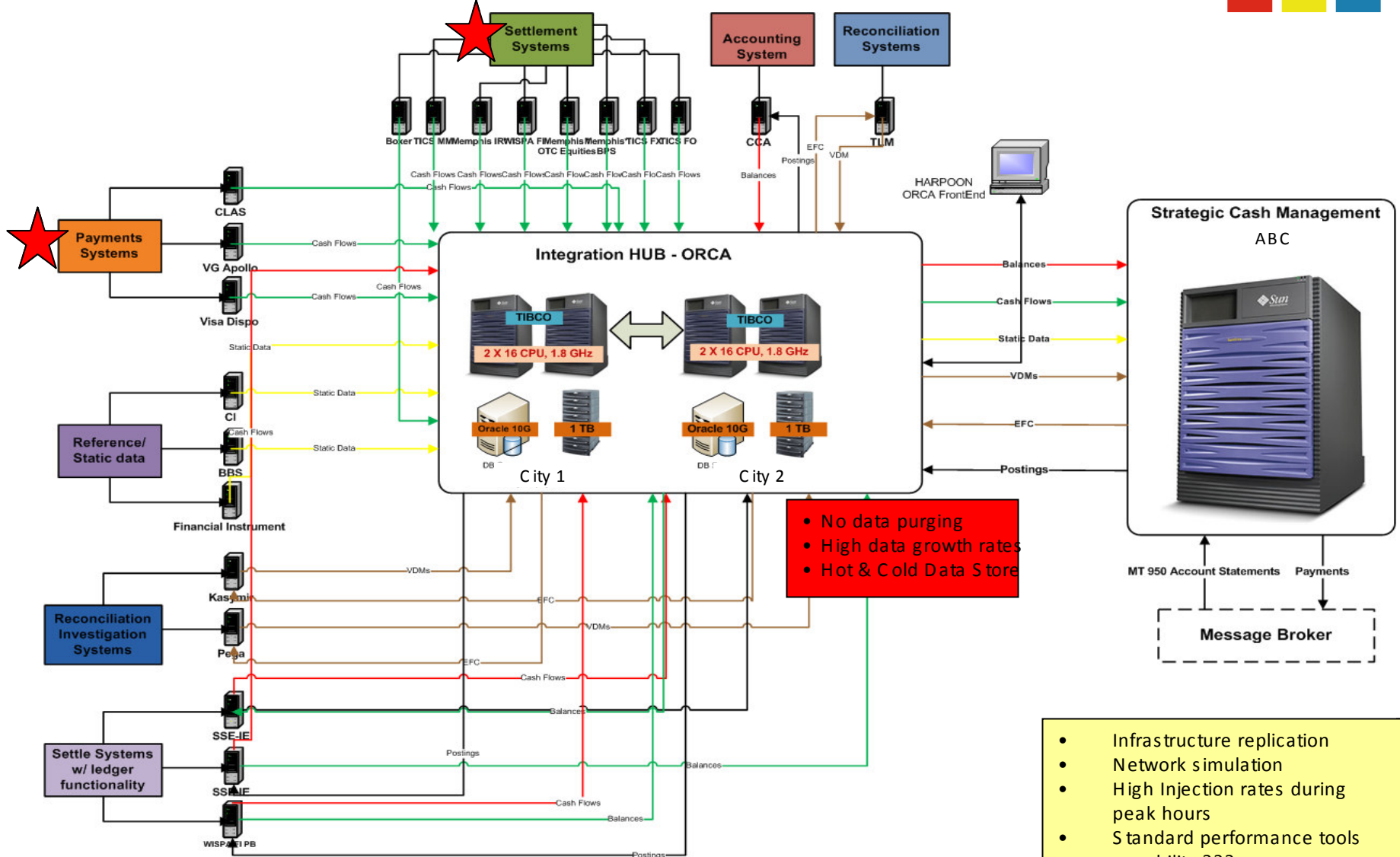
**Modelling: Hyperformix, IT Guru – OpNet**

**Management Tools: HP Mercury Performance Center**

# Performance Test Architecting Process...



# Strategic Cash Management System: Functional Architecture



• No data purging  
 • High data growth rates  
 • Hot & Cold Data Store

- Infrastructure replication
- Network simulation
- High Injection rates during peak hours
- Standard performance tools capability ???
- Upstream applications – mostly black box
- Legacy Craig to modern ORCA

4

<span style="color: red;">█</span> Balance Info	<span style="color: black;">█</span> Postings	<span style="color: green;">█</span> Cash Management Info
<span style="color: yellow;">█</span> Static Data Feed	<span style="color: brown;">█</span> Recon Process	

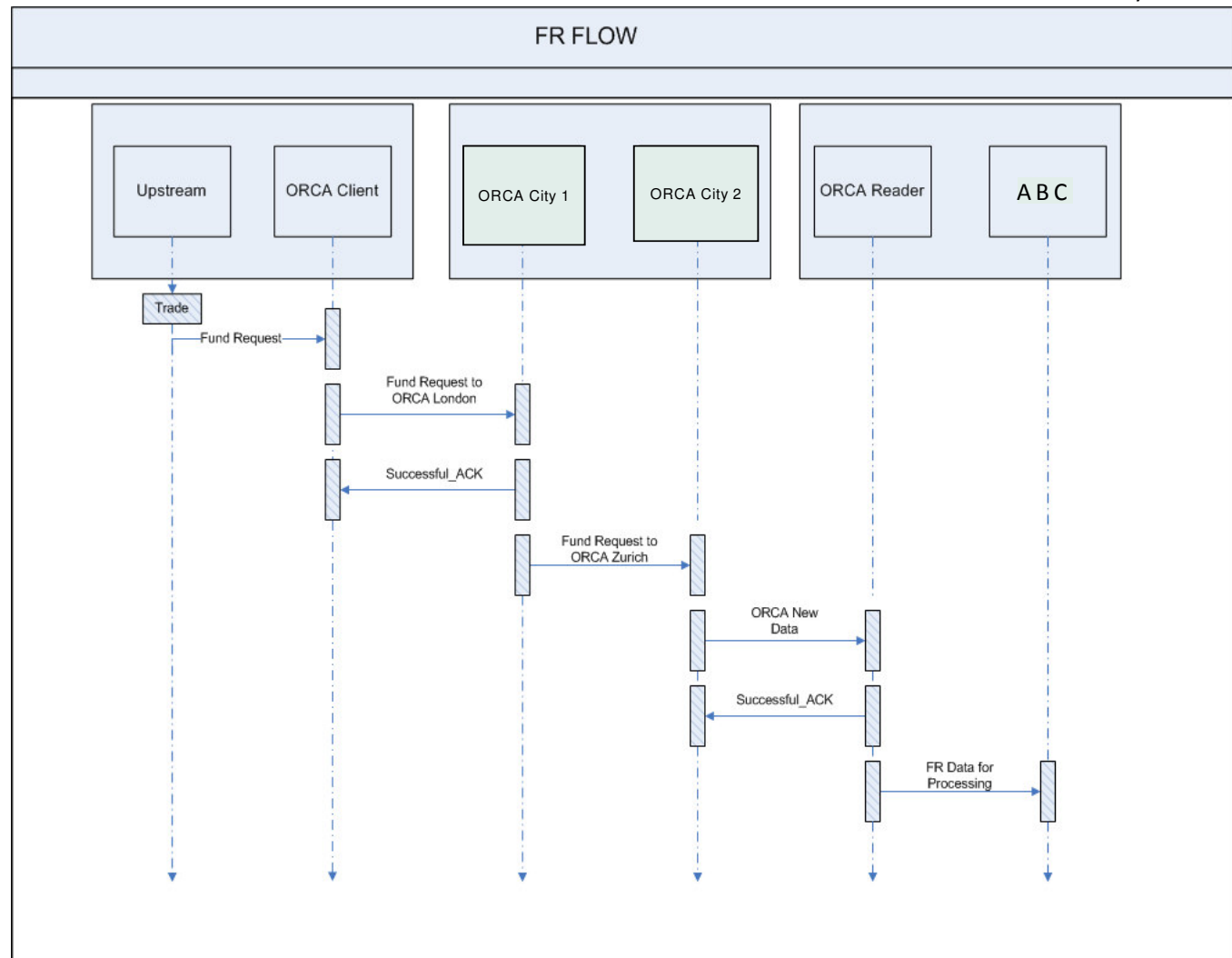
# Step 1: End-to-End (Asynchronous) Message Flow - Simplification



## Types of Messages

1. Create
2. Update
3. Replace
4. Cancel

- 7 different upstream types
- Each stream consisting of one or more applications (a total of 21)
- 21 different applications with 84 unique message formats
- Publish and subscribe
  - 7.78 million messages / day
  - Peak Volume—1.29 million/hr



## Step 2: Determine Asynchronous Workload Characteristics



Upstream	Application	Peak Volume	Daily Volume
Settlement Systems	Boxer	50000	300000
	TICS MM	25000	150000
	Memphis IR*	70000	420000
	WISPA FI	30000	180000
	Memphis * OTC Equities	45000	270000
	Memphis* BPS	100000	600000
	TICS FX	25000	150000
	TICS FO	25000	150000
Settle Systems w/ ledger functionality	SSE-IE	100000	600000
	SSE-NG	250,000	1500000
	WISPA FI PB	75000	450000
Reconciliation Systems	TLM	90000	540000
Payment System	CLAS	30000	180000
	VG?Apollo	80000	480000
	Visa Dispo	100000	600000
Reconciliation Investigation Systems	Kasymir	35000	210000
	Pega	40000	240000
Reference/Static data	CI	5000	30000
	BBS	10000	60000
	Financial Instrument	12000	72000
Accounting System	CCA	100000	600000
	<b>Total</b>	<b>1297000</b>	<b>7782000</b>

# Step 3: System Performance Requirements

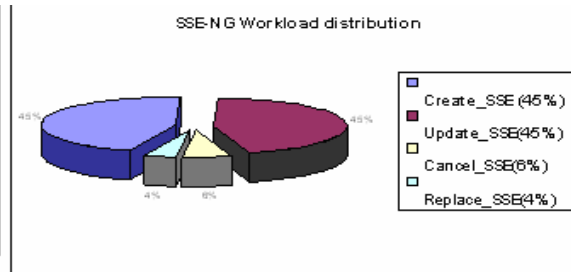


- Message Publishing Performance Requirements: The system involves the flow of Fund Request Messages from the upstream systems TICS FX, TICS MM, and SSE-NG through the message broker ORCA to Cash Management application. The performance requirements criteria for the E2E system are as follows:
  - Publish a peak load of 275,000 messages within one hour onto the system
  - In the event of a failure, Self recovery or BCP recovery within a duration of 30 Minutes
- Infrastructure utilization / performance requirements at ORCA
  - CPU - Less than 70% utilization
  - Memory – Less than 80% for non-memory intensive transactions and 80%-90% for memory intensive transactions
  - Disk – # of I/Os consumed, utilizations (values TBD - would depend on the transaction intensity ) etc

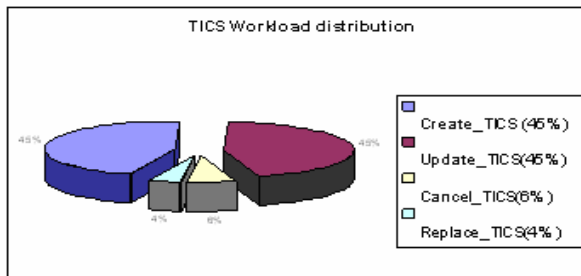
# Step 3: System Performance Requirements Contd...



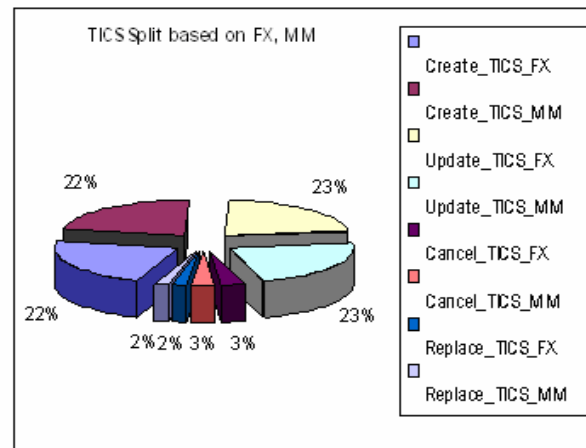
SSE-NG Individual Split based on message type	# of Messages
Create SSE (45%)	112500
Update SSE (45%)	112500
Cancel SSE (6%)	15000
Replace SSE (4%)	10000
Total	250000



TICS Individual Split based on message type	# of Messages
Create TICS (45%)	11250
Update TICS (45%)	11250
Cancel TICS (6%)	1500
Replace TICS (4%)	1000
Total	25000



TICS Split based on FX, MM	# of Messages
Create TICS FX	5625
Create TICS MM	5625
Update TICS FX	5625
Update TICS MM	5625
Cancel TICS FX	750
Cancel TICS MM	750
Replace TICS FX	500
Replace TICS MM	500
Total	25000



## Step 4: Infrastructure Analysis



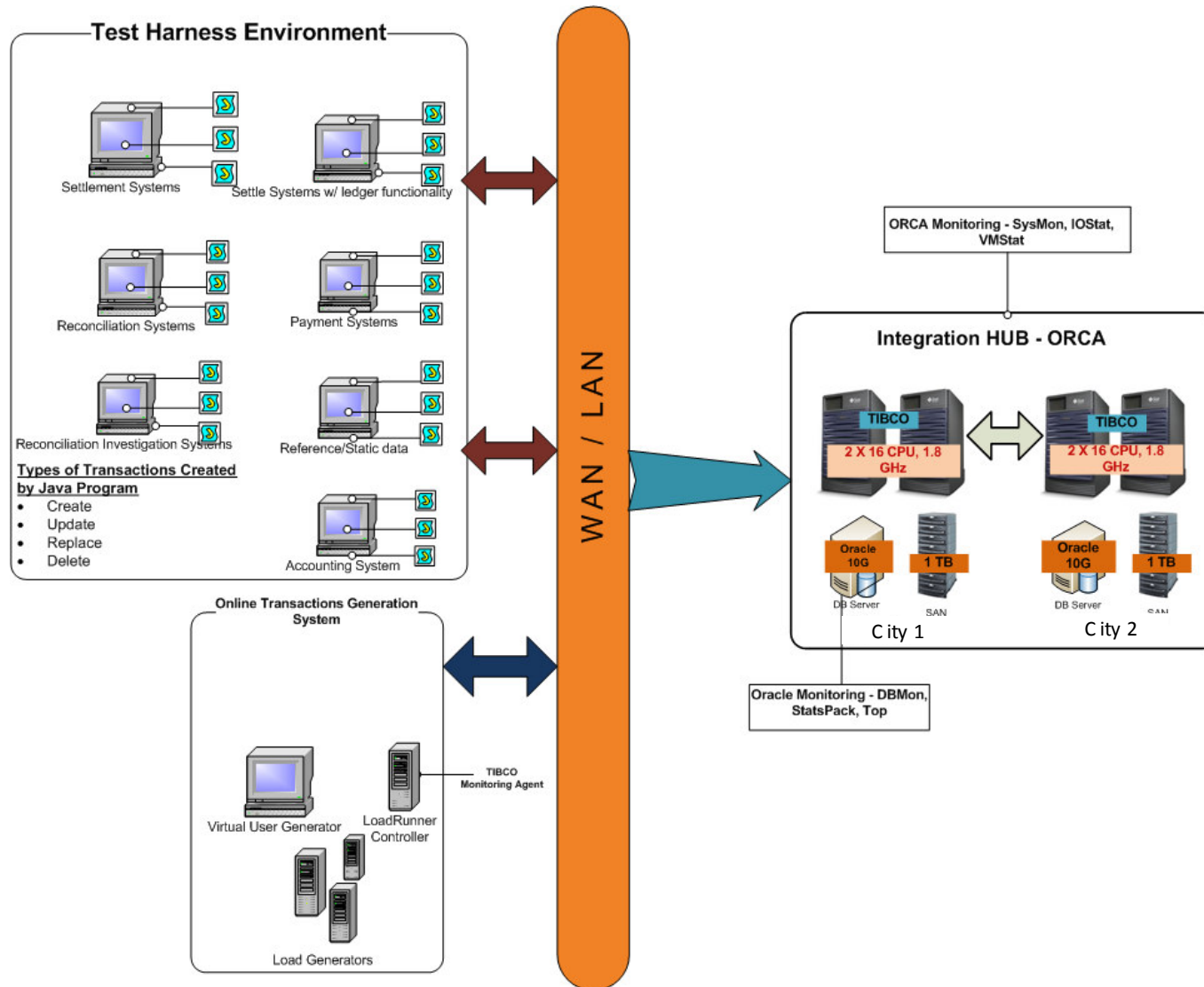
<b>Good</b>	<b>Better</b>	<b>Best</b>
50% of prod	DR site	Identical to prod

## Step 5: Identify the Performance Tests



- ✓ Load / Volume Test
- ✓ Soak Test
- ✓ BCP

# Step 6: Determine & Create the Test Harness Environment



# Step 7: Finalize the Test Data Generation Mechanism



Test Name	# of Funding Requests
Load Test	275,000
Benchmark (Stress) Test	50000
Soak Test	1500000 (One day load)
BCP and Recovery	405,000

- ✓ Java Program to read Test Data from a sample file
- ✓ Randomness in Funding Request achieved by modifying 16 digits in FRId
- ✓ Production data keeps growing → Data purging not considered
- ✓ Number of Funding Requests / file = 2000 ( In reality only one message is present in a file, but it is impossible to create 1.29 Mn files, 2000 messages / file was found to optimal for performance testing)
- ✓ Avg time for creating 2000 messages = 40 seconds per instance

# Step 8: Best Practices for Performance Test Execution



## Before Test Run

- Reboot servers
- Restore database
- Start database
- Start Application server
- Start Monitoring Tools
- Make sure enough disk space on load injectors
- Configure the test scenario

## During Test Run

- Monitor the test
- Look for errors
- Take Snapshots for database
- Inject the load gradually
- Monitor the resource utilizations
- Note down all the issues faced during the test run

## After Test Run

- Collate the results
- Save the results
- Backup all the logs
- Validate the test
- Take end snapshots for database
- Generate the report
- Analyze the results
- Identify bottlenecks
- Tune the system



live your dream

# Performance Testing - Engagement Models

# PTG @ A Large Insurance Firm



## Vision

Single end-to-end **Performance Testing Services Provider** for all the Application Development (AD) groups within the organization

## Mission

To enable AD's to **fulfill the business objectives** by ensuring **optimal performance and scalability** of IT applications with **minimized** software and hardware costs

## Objectives

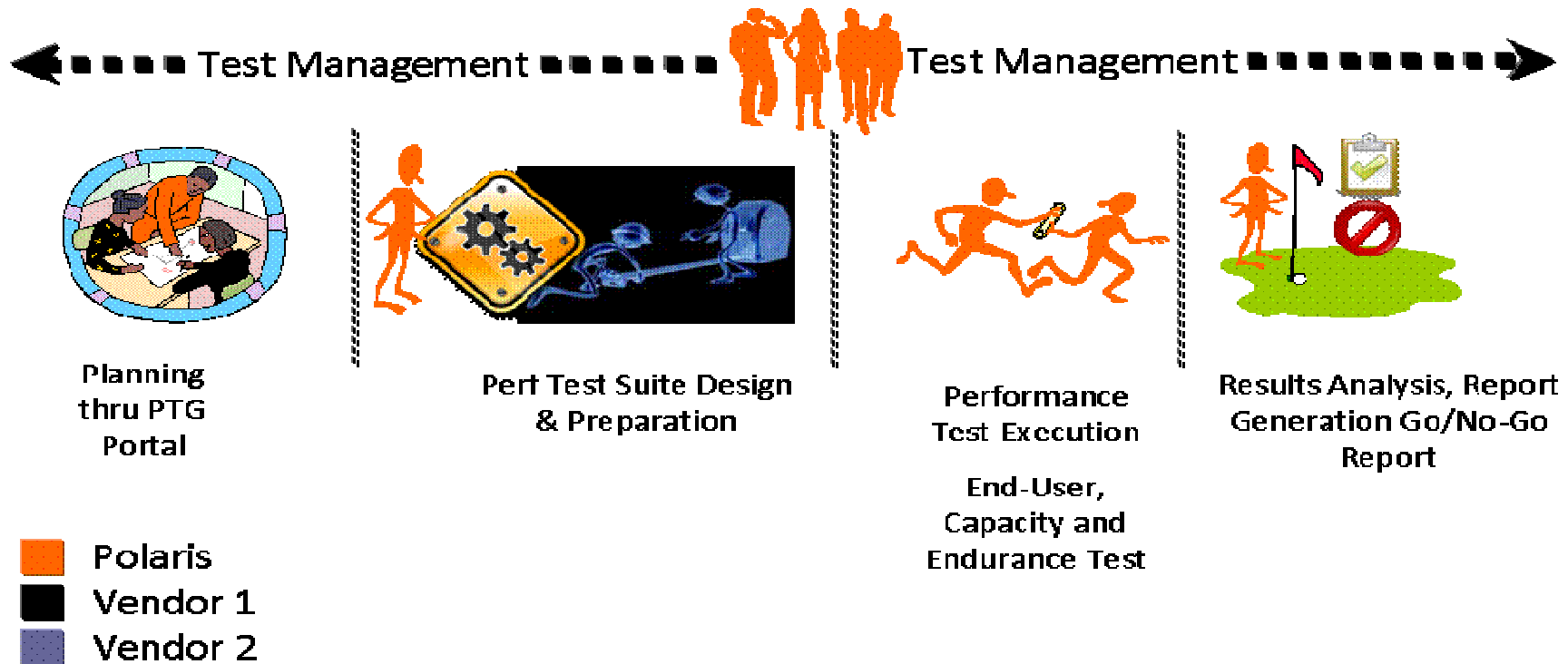
- Implement performance testing as a discipline into software development and maintenance life cycles
- Eliminate performance as a “Fix-it-Later” approach
- Prepare for worst case scenarios with high confidence levels
- Minimize the hardware costs of the production deployments
- Enforce best-in-class performance standards for design and code of applications
- Consolidate performance engineering services across the enterprise

# The Engagement Model

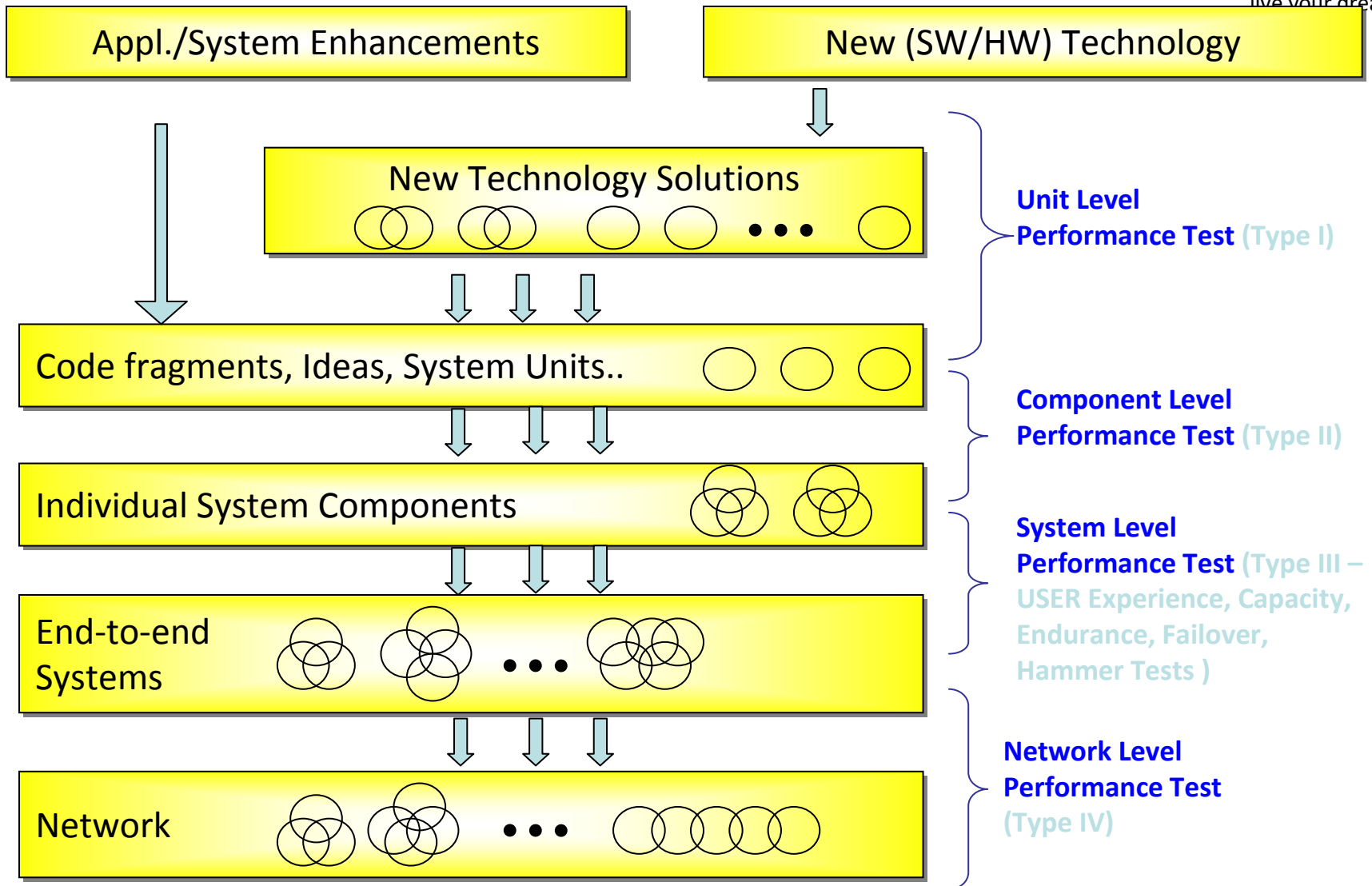


## Highlights'

- **Multivendor Environment**
- **Onsite – Nearshore – Offshore Delivery**
- **Proactive training of the resources for on demand ramp up**



# Types of Performance Tests

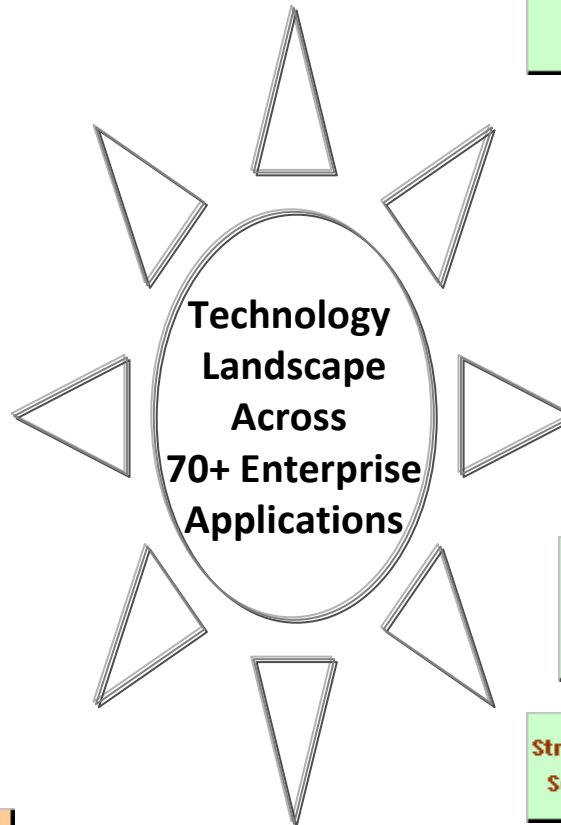


# Applications & Technologies Supported



**LoadRunner Protocols Used**  
 Web, HTTP/HTML, COM / DCOM, Citrix ICA

<b>Java, XML, WAS, Oracle 9.x</b>	Producer Website Producer Notification
<b>Java 1.4, JavaScript, Servlets, JSP, XML, WAS, Oracle 9.x</b>	FIV FIV Search Account Viewer
<b>Java/J2EE, Oracle, Web Sphere</b>	Unit Value History WINWeb 2.0 WINWeb 1.0 iTracker NEF Compensation Investment Tracker eForms eForms MegaStar
<b>J2EE, DB2, WebSphere</b>	DNSS 1.5 CIF DNSS Migration
<b>J2EE, XML, SQL Server, WebSphere</b>	BOSS I&W MIDA\$ 8.02 MIDA\$ 8.01 MIDA\$ 7.06 COMS & ECTR
<b>J2EE, DB2, CICS, WebSphere</b>	ENB-Spider ENB-CAS
<b>J2EE, Web service, MSSQL 2000, WebLogic</b>	NEF Compensation ADG Compensation Reports Website



<b>.Net, IIS, SQL Server/Oracle</b>	Electronic Funds Administration EGA Suite App
<b>BOSS Framework, LotusMail Server, STRUTS Framework</b>	SMRS - Sales Material and Compliance Review system
<b>Com Plus</b>	WIN 4.3
<b>JDBC, SOAP and HTTP based application</b>	eLeads
<b>Rational Application Developer 6, Hibernate 3.0, Struts 1.1 framework</b>	FPW - Financial Planning Website
<b>Relational Database using J2EE</b>	PMACS - Producer Management Accounting Compensation System
<b>Struts, Web services, SQL Server 2005, WebLogic</b>	Icomp
<b>VisualStudio VB.NET, 4. Transaction Logging will be done in a text file.</b>	bdCATS - Broker Dealer Commissions, Administration and Trading Support

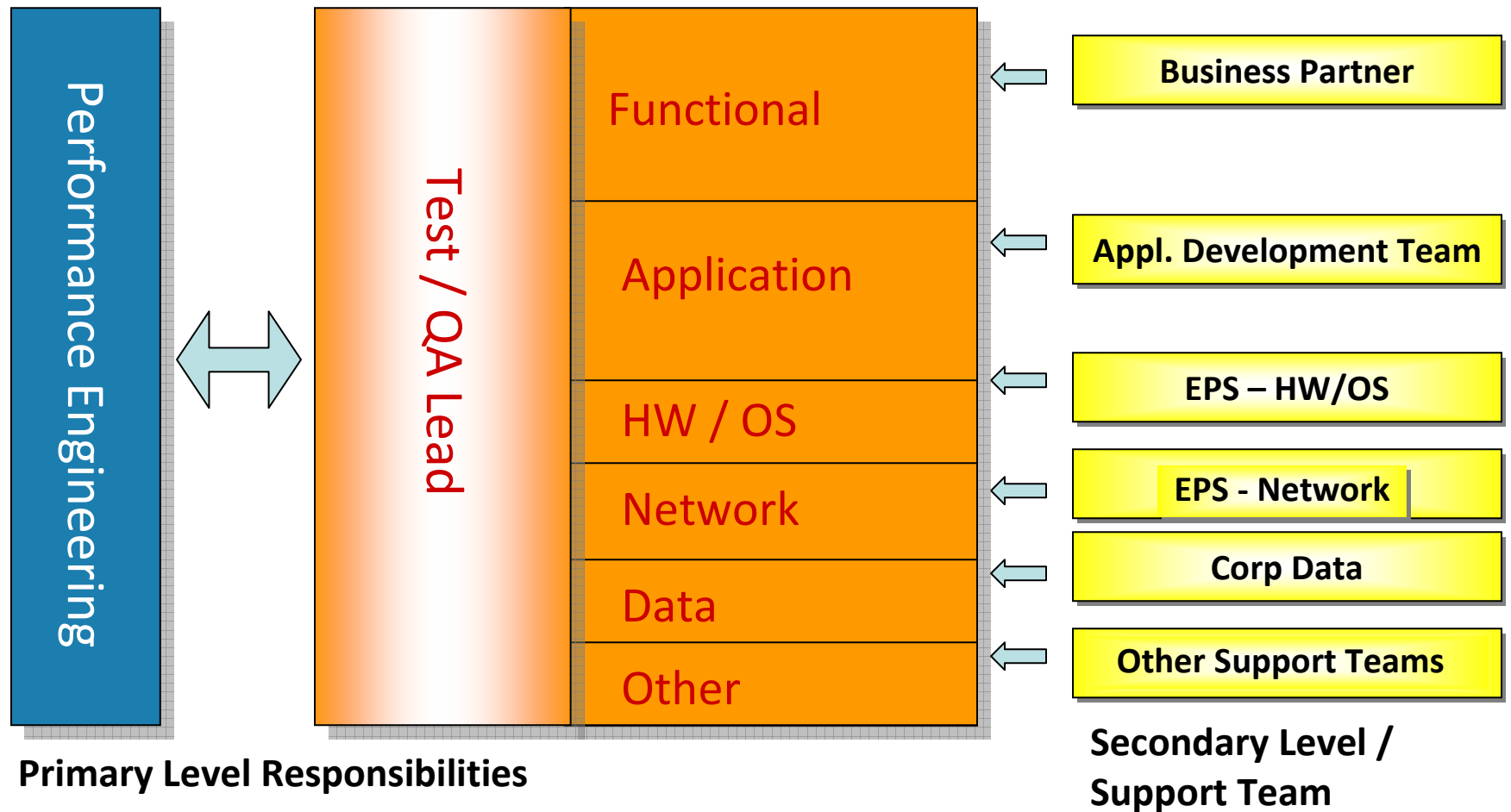
**Workload Simulation using LoadRunner & Performance Center and Custom PTG Portal**

# Operation Model



**Perf.  
Engineering**

**Application Under Test  
Project Team**



# SMRS - Sales Material & Compliance Review system



live your dream

- STRUTS Framework



- System supporting S&M personnel -> key to organizations growth
- 24/7 Availability globally to S&M personnel – key requirement
- Stringent & complex compliance rules streams in the system
- Multiple users with multiple access profiles across the globe

- Endurance test for 48 hours with 650 users
- Transaction mix test with 1500 concurrent users
  - Load simulated from diff geographies

- Websphere Application server
- UDB (DB2)
- BOSS Framework, Lotus Mail Server

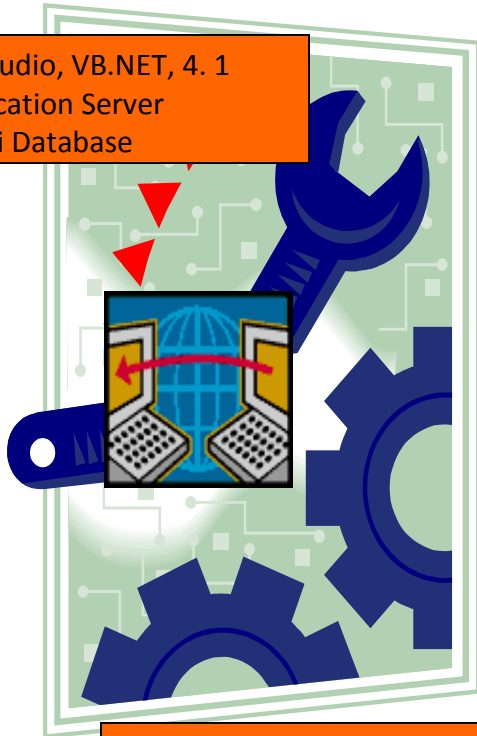
- Verified the system performance for 1500 concurrent users
- Transaction response times verified with Vuser load generated from various locations
- Successfully executed the endurance test for 48 hrs verifying the application availability for extended usage

- Will the system be available for longer duration of usage?
- Can the system handle users from different geographies?
- Verify application impact on other interfacing applications due to extensive changes
- Verify the system for any issues that might degrade the system performance

# bdCATS - Broker Dealer Commissions, Administration & Trading Support



- Visual Studio, VB.NET, 4. 1
- IIS Application Server
- Oracle 9i Database



- AJAX Framework

- Business critical suite of tools
- 250 concurrent user load expected
- Data to be prepared by Performance Testing team
- Querying transactions through multiple channels making the system complex
- New / additional features included in each release

- Peak load transaction mix test with 250 active users
- Data fill up runs for test data creation
- 48 hour endurance test with 80 active users
- Customized scripts to capture data queries

- Application performance baseline as reference point for future releases
- Data creation scripts significantly reducing the test data creation effort
- Bottom up approach, eliminating the bottlenecks in each stage of SDLC
- Reusable performance test suite (with minimal changes for each release)
- Querying transactions logged into text file to verify the successful completion
- System verified for various tests with varying user volumes up to a maximum of 250 users

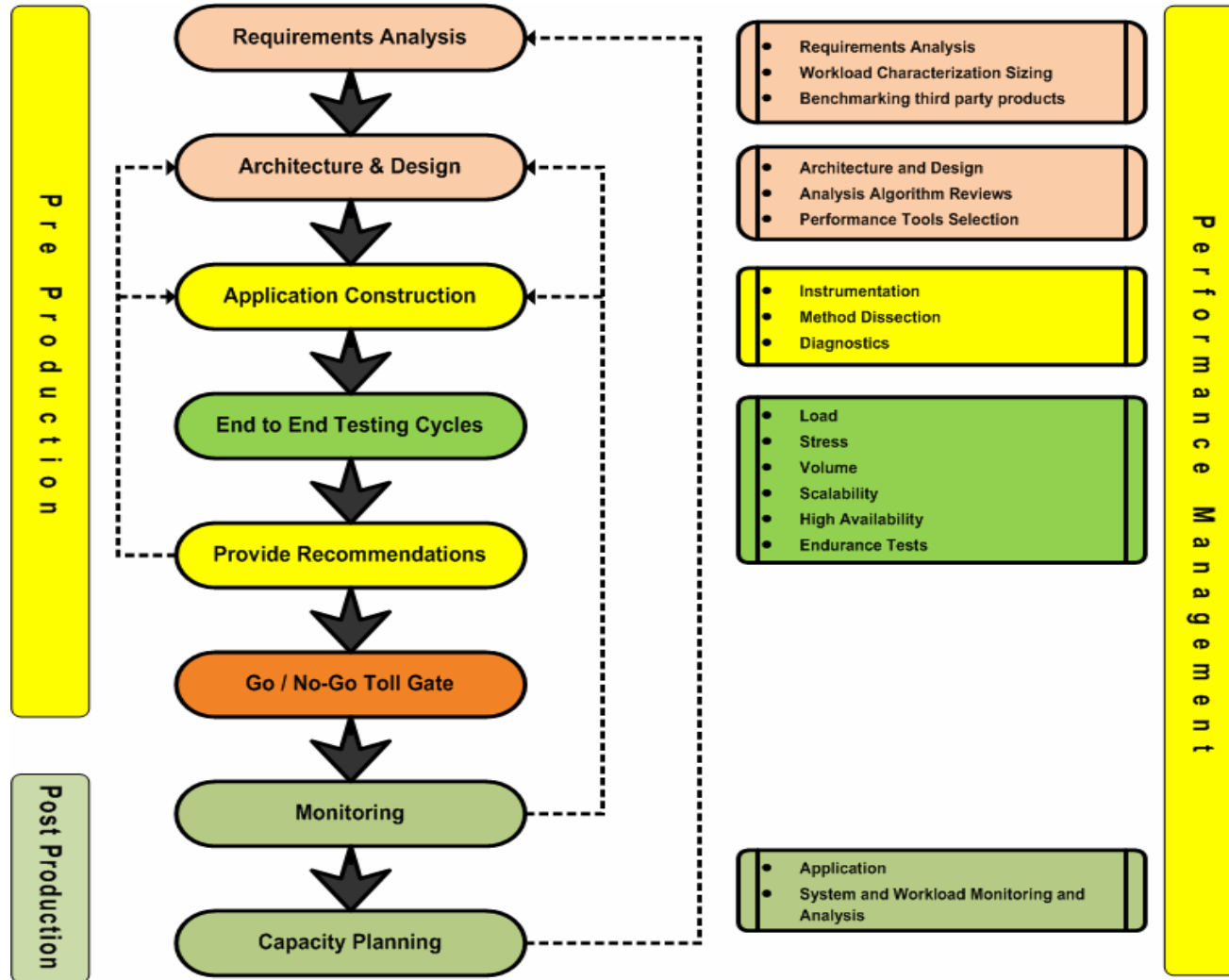




live your dream

# Overall Performance Engineering Process

# In-Process Performance Engineering





live your dream

# Questions ???



live your dream

# Thank You