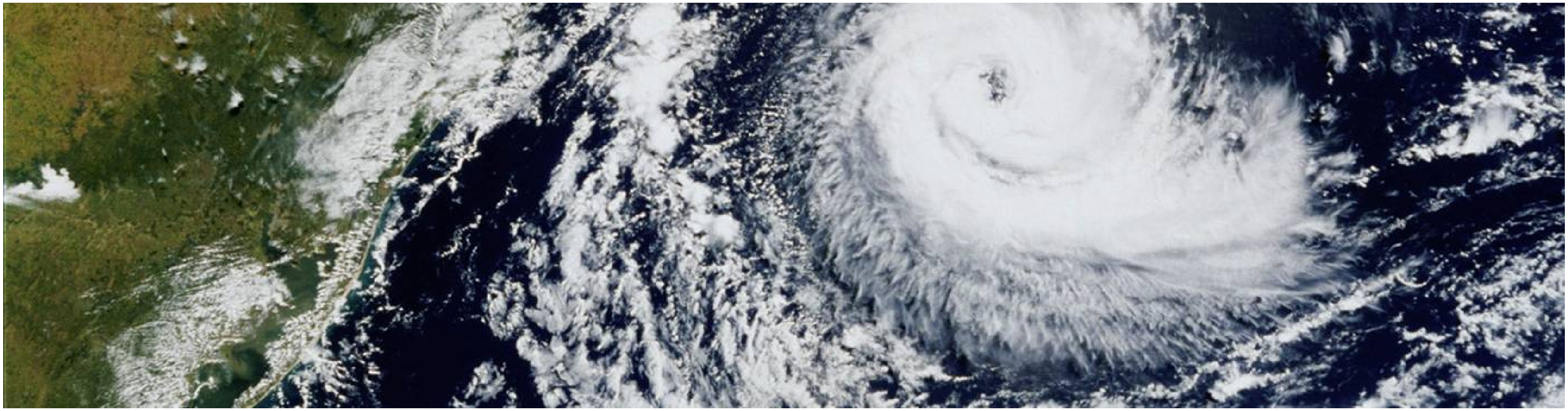
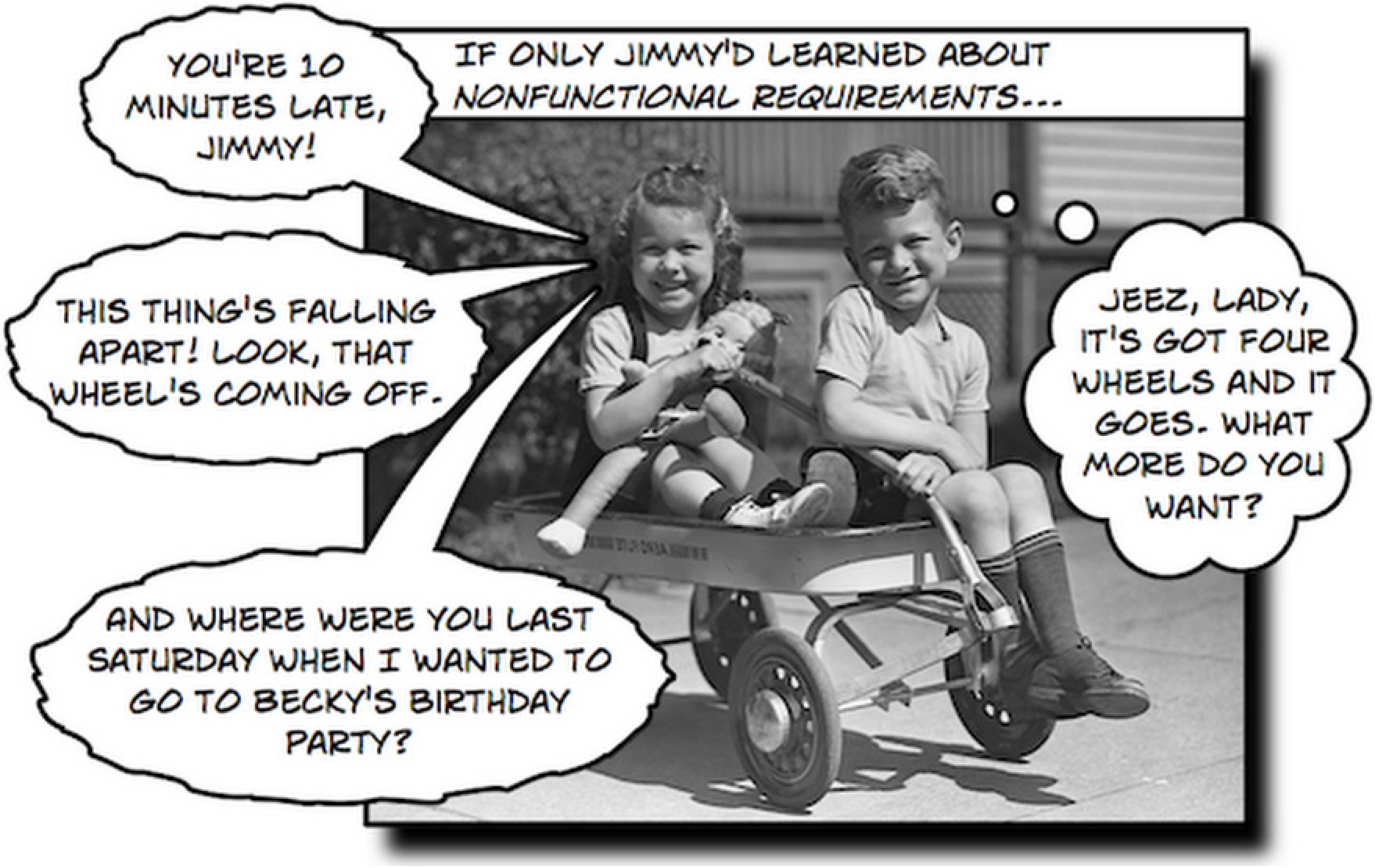
Best Practices in making production - grade applications

- A Performance Architect's View



Archanaa Panda, Bharathraj – IBM, HiPODS, India SW Labs

Quality Attributes or NFRs – A brief understanding


IF ONLY JIMMY'D LEARNED ABOUT NONFUNCTIONAL REQUIREMENTS...
YOU'RE 10 MINUTES LATE, JIMMY!
THIS THING'S FALLING APART! LOOK, THAT WHEEL'S COMING OFF.
AND WHERE WERE YOU LAST SATURDAY WHEN I WANTED TO GO TO BECKY'S BIRTHDAY PARTY?
JEEZ, LADY, IT'S GOT FOUR WHEELS AND IT GOES. WHAT MORE DO YOU WANT?

http://www.ibm.com/developerworks/websphere/zones/hipods

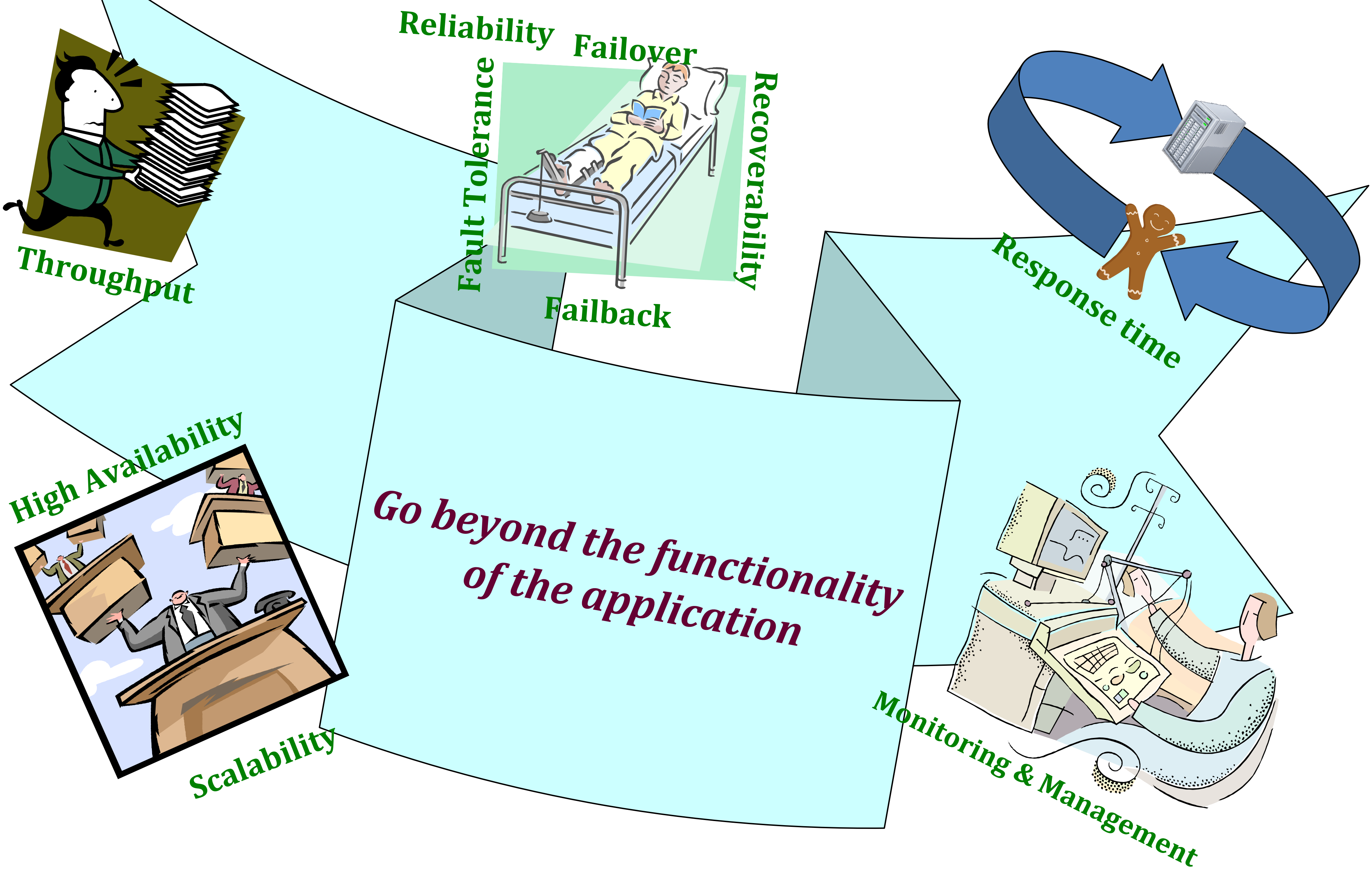
Why are NFRs important?

Neglecting NFRs can lead to series of software failures

- Systemic failure in Major European City's Ambulance System.
- System failure because of performance-scalability problems in the Department of Motor Vehicles Licensing system of a US state.
- European automaker recalled more than 50000 cars because of performance delay in airbags software
- System got severely delayed because of performance-scalability problems in a UK based major online retail chain.

Do you want your application to be in this list?

Quality Attributes or NFRs – A step further



Throughput

Reliability

Failover

Fault Tolerance

Recoverability

Failback

Response time

High Availability

Scalability

Go beyond the functionality of the application

Monitoring & Management

http://www.ibm.com/developerworks/websphere/zones/hipods

Build your system right!

Quantification of quality attributes

- Volumetric – Number of concurrent users, number of active users, estimated growth of users, estimated session duration
- Availability – Number of working hours, Available maintenance windows, How much time for system upgrades, SLAs
- Performance – Response time objective per use case, 85th percentile of response time, Throughput (no of transactions completed) per use case – time in hrs, minutes, secs

Performance metrics – Workload Model

- Build the right NFRs

- Computation mechanism – little's law:

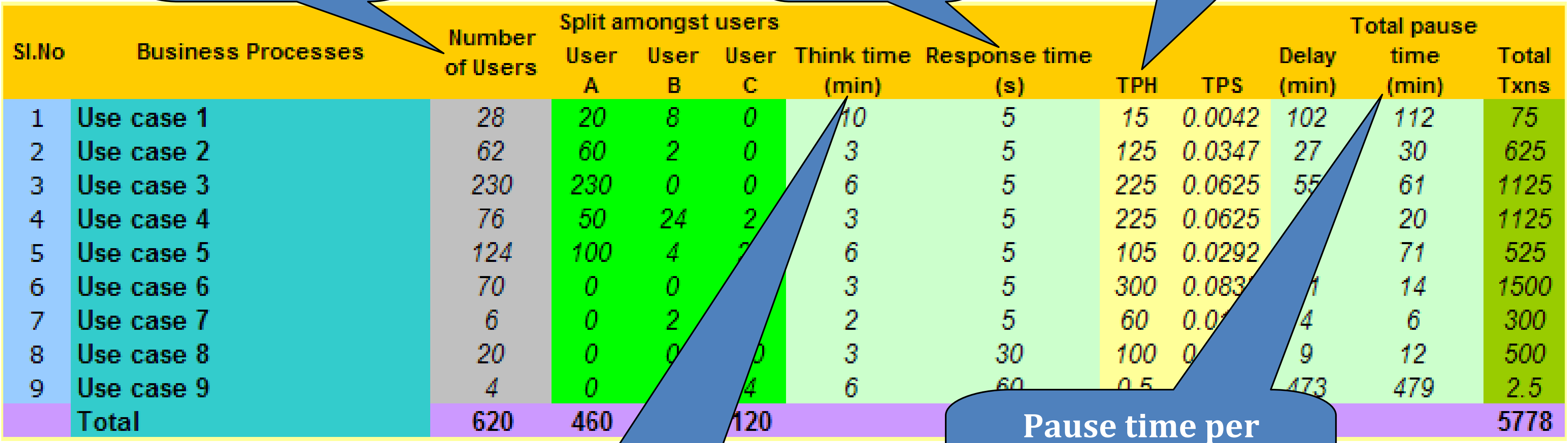
  – Number of concurrent users = Throughput of the system * (Response time + Total Pause Time)

- Sample [illegible]:

Number of users known

Response time known

Throughput known



Sl.No	Business Processes	Number of Users	Split amongst users User A	User B	User C	Think time (min)	Response time (s)	TPH	TPS	Delay (min)	Total pause time (min)	Total Txns
1	Use case 1	28	20	8	0	10	5	15	0.0042	102	112	75
2	Use case 2	62	60	2	0	3	5	125	0.0347	27	30	625
3	Use case 3	230	230	0	0	6	5	225	0.0625	55	61	1125
4	Use case 4	76	50	24	2	3	5	225	0.0625	[illegible]	20	1125
5	Use case 5	124	100	4	2	6	5	105	0.0292	[illegible]	71	525
6	Use case 6	70	0	0	[illegible]	3	5	300	0.083	1	14	1500
7	Use case 7	6	0	2	[illegible]	2	5	60	0.0	4	6	300
8	Use case 8	20	0	0	0	3	30	100	0	9	12	500
9	Use case 9	4	0	[illegible]	4	6	60	0.5	[illegible]	473	479	2.5
	Total	620	460	[illegible]	120							5778

Think time + Delay = Total pause time

Pause time per usecase calculated using little's law

http://www.ibm.com/developerworks/websphere/zones/hipods

It is all about balancing NFRs

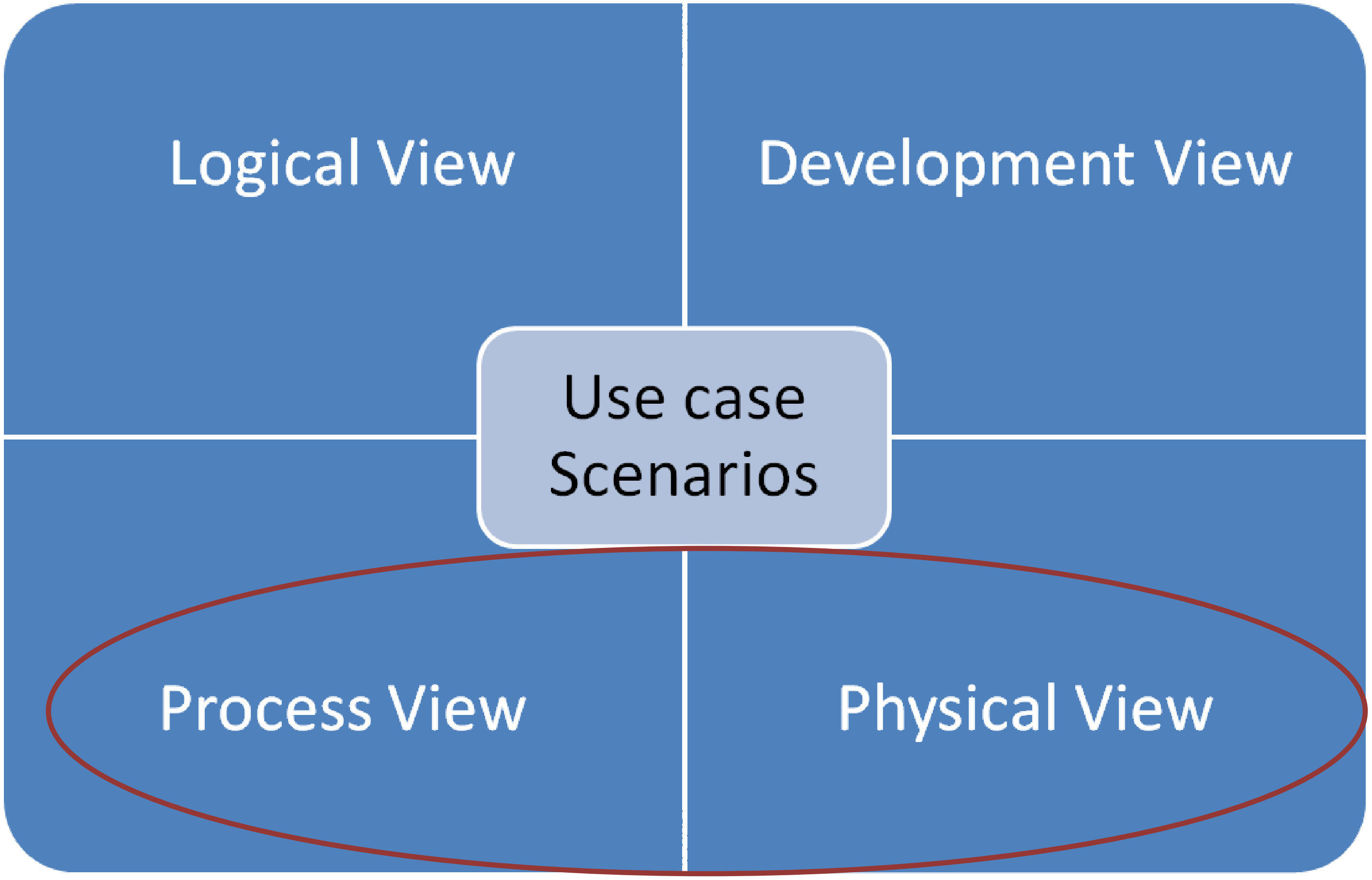
- Performance vs Reliability
- Performance vs Interoperability
- Performance vs Security
- Performance vs Manageability
- Manageability vs Scalability

http://www.ibm.com/developerworks/websphere/zones/hipods

Example Application domains and most relevant NFRs

Banking, finance, insurance	Reliability, security, performance, scalability
Telecom	Performance, scalability, maintainability, reliability
Government and military	Security, reliability
Transportation	Performance, scalability, accuracy, maintainability

The 4+1 Architecture View



http://www.ibm.com/developerworks/websphere/zones/hipods

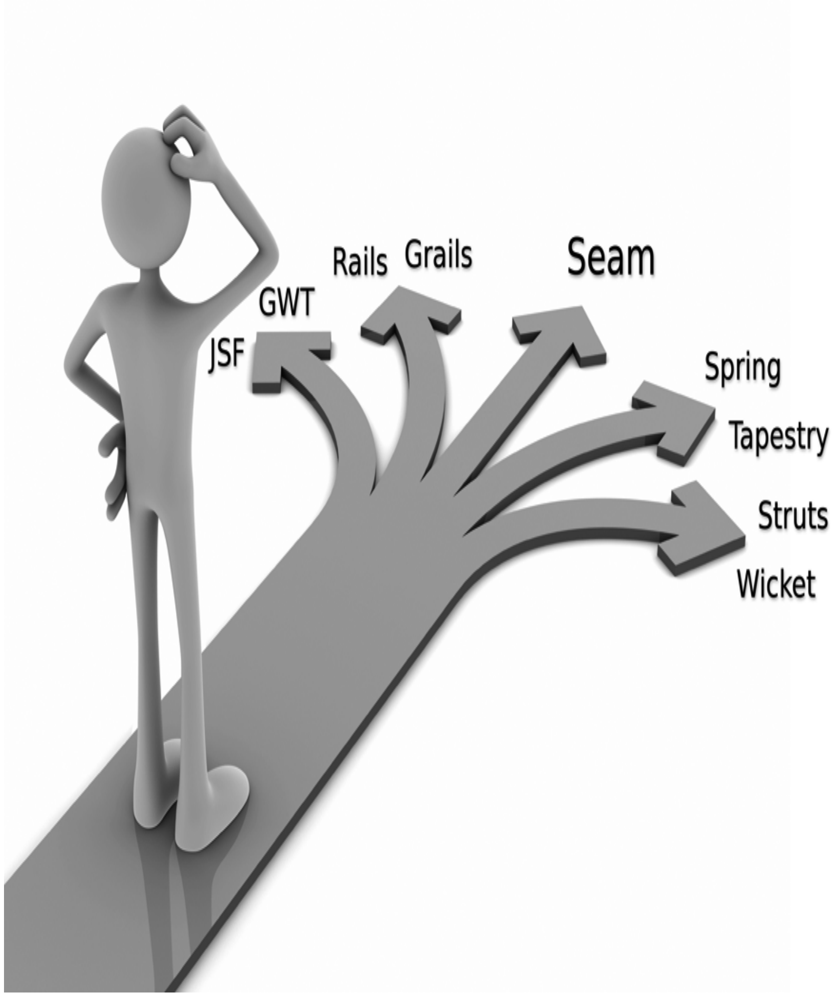
Have you used your framework properly for NFRs?

- Reading between the lines

- understanding lifecycle of framework components

- Make framework fit to application, not other way round.

- Evaluate framework for application NFRs

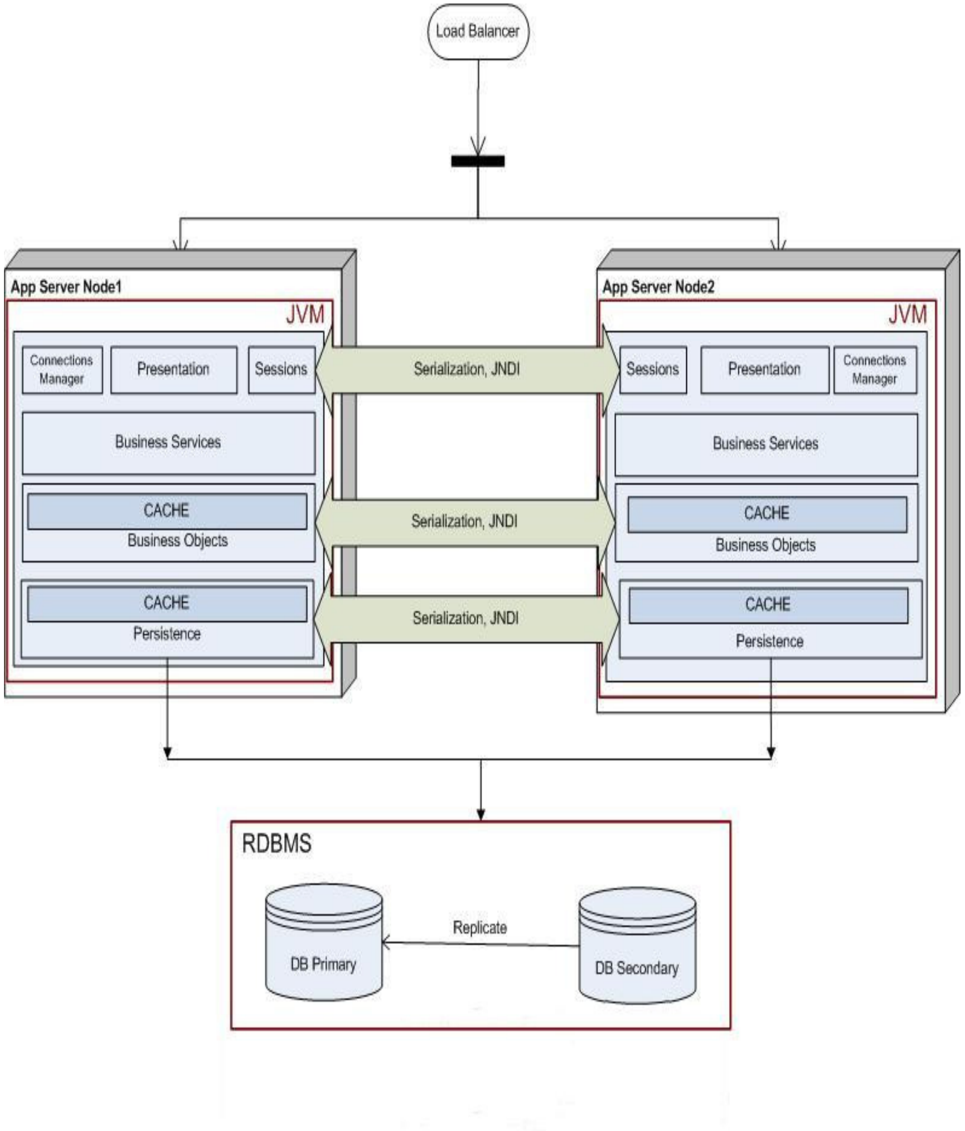


http://www.ibm.com/developerworks/websphere/zones/hipods

Deciding topology for application

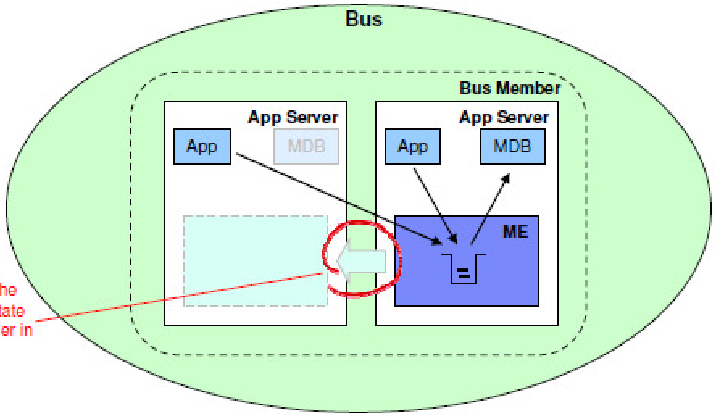
Looking beyond JavaPetStore or PlantsByWebSphere – typical 3-tier applications (default configurations)

- Monolithic vs Distributed
- Horizontal vs Vertical Scalability
- Clustering vs Farming
- Understanding clustering and availability features of application servers – servlet containers and sessions, EJBs, Message Queues

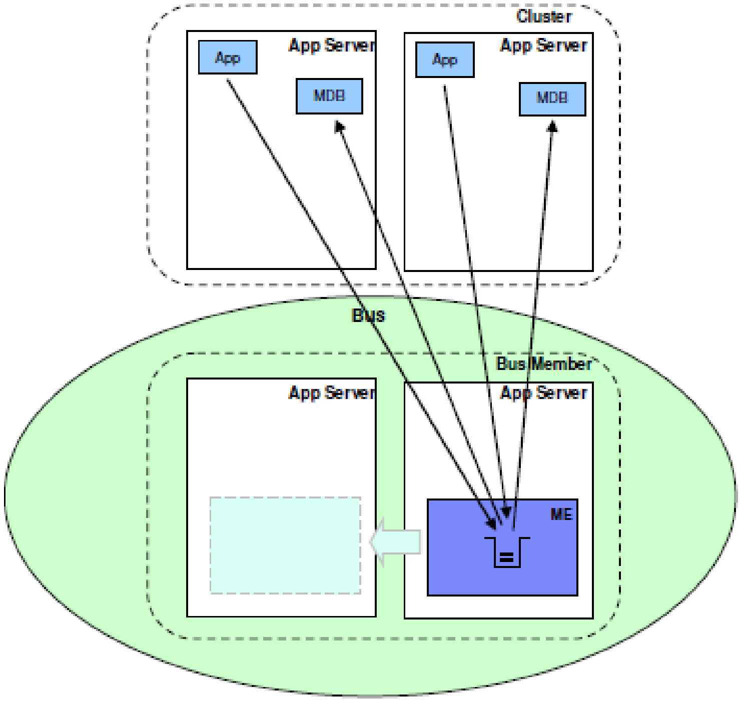


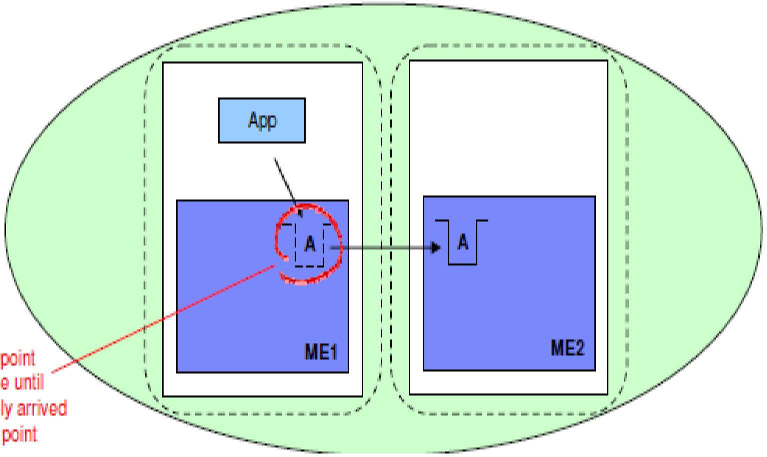
http://www.ibm.com/developerworks/websphere/zones/hipods

Deciding topology for application – eg JMS



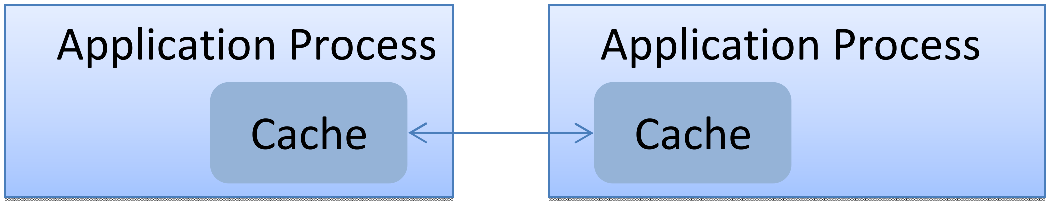
In the event of a failure, the ME and all its persistent state can failover to another server in the cluster.



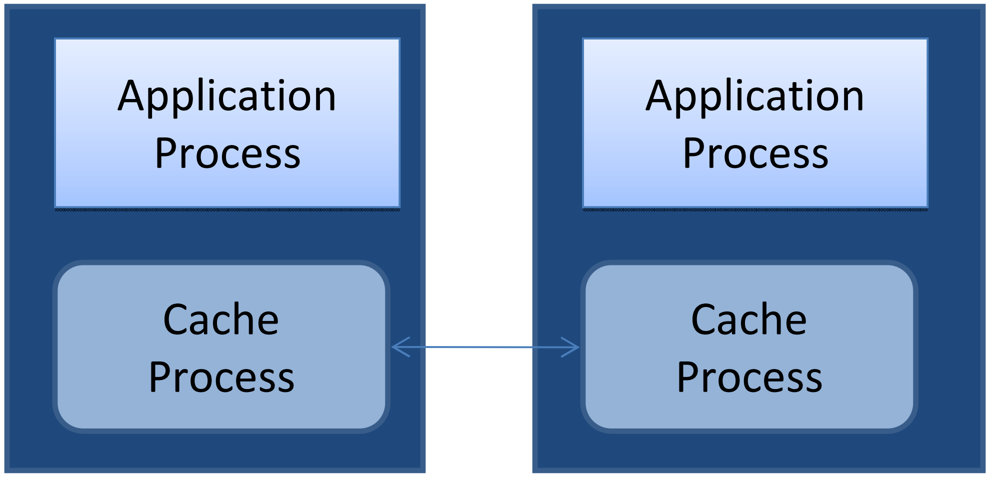


The remote queue point stores each message until it knows that it has safely arrived at the target queue point

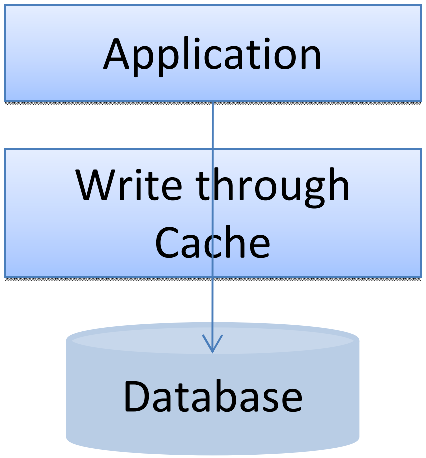
Deciding topology for application – eg Caching

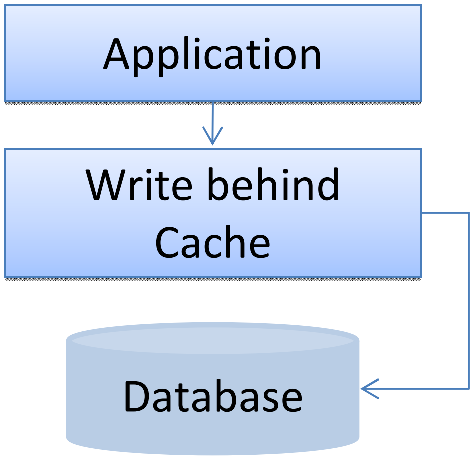


Co-located cache with Application Process (JVM)



Cache as separate process in same machine





http://www.ibm.com/developerworks/websphere/zones/hipods

Deciding topology for application – some guidelines

- Separation of business concerns or responsibilities like order capture and payment handling.
- Co-locate modules in same process/JVM when
  - Required to share memory frequently
  - When module1 and module2 are very inter-related or inter-dependent. Frequent communication and serialization is overhead
- Modules in different process
  - Memory limit - 32-bit OS
  - Fault tolerance and Availability

Deciding topology for application – some guidelines

- Modules in different processes (contd..)
  - Managing deployment of modules separately
  - Easier to isolate problems
- Modules in different machines
  - CPU, I/O and Memory requirements differ. Eg one module CPU intensive, other module I/O intensive.
  - Easier to isolate problems

Making monitoring-ready Production Grade Applications

- Logging not the only way to monitor.
- Build simple dashboards. Web Application with numerous pages can accommodate 1 simple monitoring page!
- Make manageability one of your requirements.
- Understand monitoring features of application servers and off-the-shelf solutions.
- GUI – simplest way to monitor.

http://www.ibm.com/developerworks/websphere/zones/hipods

http://www.ibm.com/developerworks/websphere/zones/hipods

CONTACT DETAILS

- Email ID: archanaa.panda@in.ibm.com
- Phone: +919818661064
- http://www.ibm.com/developerworks/websphere/zones/hipods