UNIT - 3

INFRASTRUCTURE ARCHITECTURE BUILDING BLOCKS

ⓐ Network, Internetwork and communication protocols — DMZ, DNS, Load Balance, Firewall, Cluster

ⓑ IT Hardware and S/W.

ⓒ Middleware.

ⓓ Policies for Infrastructure Management

DNS : Unique Identity provide

Load Balance : Available both hardware and software.
- → Split the traffic.
- → Ensure scalability of EA.

Cluster : Ensure availability and performance of EA.

Firewall :
- Security policy.
- Application security.

DMZ (Demilitiarized Zone)
- → Provide access to external facing service.
- → Interoperability.

IT HARDWARE & S/W :

- Operating system
- Database server
- Web server
- Application server
- Directory server (or) Datastore

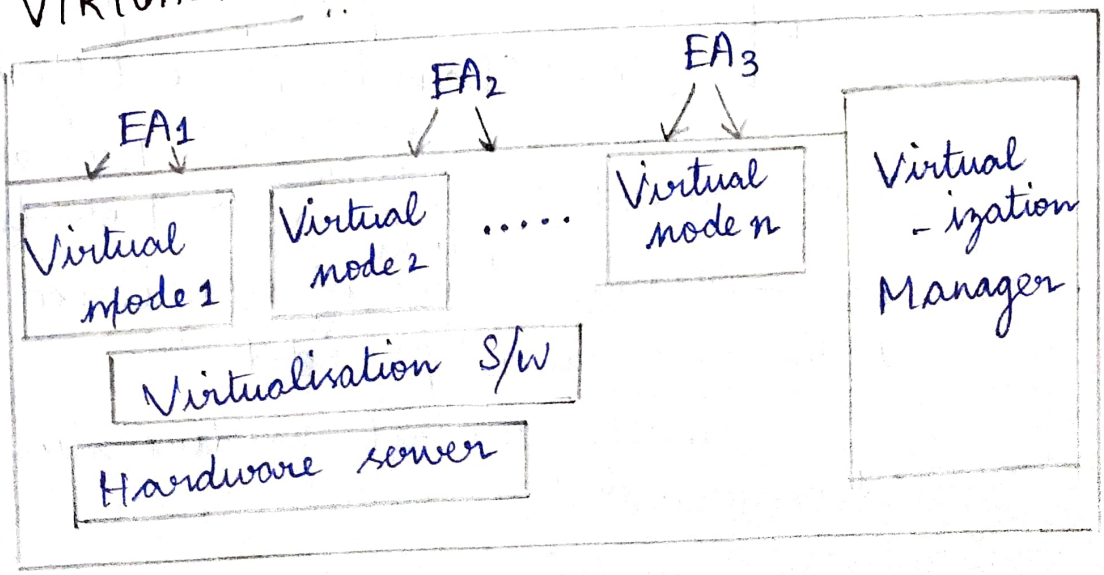
MIDDLEWARE : (S/W binds together software pieces of distributed application and enable integrate discrete EA and their component).

MOM — RPC

1. Message Oriented Protocol } — Client / Server Model. — Asynchronous communication.

2. RPC (Remote Procedure Call).

VIRTUALIZATION :



TECHNICAL SOLUTION:

- Bring all element together to provide holistic view of complete solution

DATA ARCHITECTURE AND DESIGN:

1. Structure data.
2. Unstructure data.

Structure Data
- Relational Data Model.
- XML.

Relational Data Model
- Conceptual modeling.
- Logical model.
- Physical model.
- Relationship
  - Recursive.
  - Super type and sub type.

Unstructure Data
- Blob (Binary Large Object)
- Clob (Character Large Object)

CODE REVIEW:

- Completeness
- Correctness
- Consistency
- Logic
- Traceability
- Robustness

- Measure degree of compilance with specification provided.
- Quality artifact produce during development.

COMPLETENESS:

- Ensure to meet all key requirement.

CORRECTNESS:

- Language specific idioms.
- Avoid hard coding.
- Eliminate unused variable.

CONSISTENCY:

- Ensure uniformity in coding.
- Commenting.
- Error and exception handling.

REVIEW OF CODE:

- Logical Perspective code.
- Ensure code result in expected behaviour.

MAINTAINABILITY:

- Review performed to ensure code is easily to understand from perspective maintainance.

TRACEABILITY:

- Find out any requiste functionality is missing.

REVIEW:

- Ensure code is able to handle error and exception handling.

STATIC CODE ANALYSIS:

*) Activity of analysing the code.

*) Identify the issue related to non-runtime aspect.

Static Code Analysis (STYLE)
- Code Analysis (STYLE)
- Logical Bugs
- Security Vulnerability
- Code Quality

CODE ANALYSIS:

- Focus on use of right language such as
  - Use of apt control construct
  - Appropriate language idioms
  - Right type of data structure
  - Whether code is correctly formulated for readability.
  - Identify very beginning of construction phase.

LOGICAL BUGS:

- Manual Code review activity
- Small subset of bugs maybe identify.

SECURITY VULNERABILITY:

1. Cross site Scripting (or) XSS
   - attacker pass malicious script through form field.

VULNERABLE CODE

```
response.write(request.getparameter("customer name"));
```

SECURE CODE

```
response.write(encoder(request.getparameter("customer name")));
```

2. SQL Injection

- attacked to change logic of underlying SQL Query by injecting malicious SQL from the form field.

MALICIOUS CODE:

```
Prepared Statement Pstmt = Conn. Prepared
                Statement ("Insert into
customer ( customed values"("+customer+"));
```

WITHOUT MALICIOUS CODE:

```
Prepared Statement } = Conn. Prepared Statement
Pstmt
("Insert customer (Cust name) values (?)")
String name = request. get Parameter
                ("Custname");

Pstmt. setstring (1, custname);
```

IMPROPER SESSION HANDLING:

- Whether session time out set
- Session User validate before him / her access

particular session.

CODE QUALITY:

Code being delivery Highest quality

- Modularity
- Extensibility
- Maintainability
- Reusability
- Testability
- Performance

Key metrics used to measure code quality

1. Code size - measure size of class in terms of lines of code
   - Big class more than 1000 codes
   - Poor readability and maintainability
   - Defects in maintainance.

2. Cyclomatic Complexity (Measure no. of linearly independent path)
   - Loop should be minimized
   - Eliminate Unnecessary Condition.

COMMENT TO CODE RATIO:

- Lower comment ratio result in difficulty in understanding.

NO. OF ATTRIBUTES:

- Measure no. of attribute in class.
- Impact maintainability.

COUPLING B/W OBJECTS:

DYNAMIC CODE ANALYSIS (Issue related with runtime code aspect):

Dynamic Code Analysis —(includes)→
- Code profiling
- Code Coverage

Code profiling
- Time consuming method.
- Call Sequence.
- Thread deadlock information
- Elasped Time.
- Live thread and thread stage

Code Coverage
- Path coverage
- Condition coverage
- Decision coverage
- Method coverage
- Statement coverage

CODE PROFILING

TIME CONSUMING METHOD:

- Performance Analysis of E.A.
- Capture runtime attribute of code and identify performance bottleneck.

Time consumed by method

- Metric capture time taken by CPU to complete of method
- Identify bottleneck of E.A and analysis performance.

Call Sequence

- Analysis of path traversed during code execution.
- List actual sequence of call.

Thread Deadlock

- Provide information about deadlock.

List of live thread and thread state

- List live thread of program.

Elapsed Time

Sleep
Program Pause
I/o Wait

- Provide information indication of impact of endured experience

CODE COVERAGE:

- Percentage to which source code EA is tested.
- Doesn't check performance of EA.

Statement Coverage

- Measure the degree in which stmt of EA under test have been covered during test execution

Method Coverage

- Measure function of application covered during test execution.

Decision Coverage

- Loop and all decision

Condition Coverage

- Condition of all Boolean expression

Path Coverage

- test all logical flows in given E.A.

UNIT-5

TYPES OF TESTING

Testing Categories
- What need to be test
- How to test
  - Black Box
  - White Box

What need to be test
- Functional Testing
- Non-Functional Testing

Functional Testing - Testing involves screenflow, data flow, business logic users and sfy validation.

Non-Functional Testing - Test all other thing except functionality

Utilute