

Architecture Review Team (ART) Framework

Guidelines for HedgeServ Inc. Technology platforms

Guidelines for HedgeServ Inc. Technology platforms 1

Version History 3

Architecture Review Overview 4

Technology Governance Principles 4

Development Team Prepartions 5

Full Architecture Review Process Details 7

Step 1: Validate Application Readiness 7

Activities (~60 minutes): 7

Artifacts to be produced: 7

Step 2: Customer Considerations 8

Activities (30-60 minutes): 8

Artifacts to be produced: 8

Step 3: Assess Technical Requirements 9

Functional & Non-Functional Requirements 9

Automation Requirements 10

Activities (4-8 hours): 10

Artifacts to be produced: 11

Step 4: Assess Security Requirements 12

Activities (30-60 minutes): 12

Artifacts to be produced: 12

Step 5: Assess Total Cost of Ownership (TCO) 13

Activities (8-16 hours): 13

Artifacts to be produced: 13

# Version History

|  |  |  |
| --- | --- | --- |
| Date | Author | Version |
|  |  |  |
| 03/01/2017 | Dawit Lessanu | 1.0 |

# Architecture Review Overview

The Architecture Review Template aspires to provide a rigorous, repeatable, fact-based methodology to determine if proposed technology solutions are properly designed for hosting within HedgeServ’s private or public cloud environments.

The Architecture Review Process focuses on 5 key areas:

* + Early application design readiness
  + Customer experience (as necessary)
  + Technical requirements
  + Security requirements
  + Total cost of ownership (TCO)

## Technology Governance Principles

To the extent possible, solutiosn will abide by the following principles.

* Implement resilient architectures to ensure high availability and business continuity.
* Implement security best practices with regard to transmission and persistence of sensitive client data and proper session management.
* Provide operational health integrity and consistently log information to accelerate discovery of problems or aid in root cause analysis.
* Solutions will remain vendor agnostic to minimize lock-in
* Solutions shall be fully automated with respect to:
  + Resource provisioning and testing
  + Software deployment and testing

# Development Team Preparations

This step is conducted between the development team, operations and architecture.

Initial High-Level Questions and Document for project team to answer ahead of the architecture review meeting. (***Duration 1-2 hours***)

|  |  |
| --- | --- |
| Criteria | Considerations |
| New Technology | * Is the solution introducing a new platform/3rd party component? (If YES, then this requires the input of Infrastructure and Application Engineering teams and a consideration of cost (licenses and support) with prior approval. |
| Solution can scale up & down | * Is the solution architected in such a way that it can be easily scaled horizontally? * Does the solution have tremendous elasticity demands? |
| Solution Logical Design | Are the logical components of the solution adhering to Service Oriented Architecture design?   * Is the solution leveraging **existing** technology, components or services adequately (*avoid duplicative work*)? * Loosely coupled logic that can be scaled independently (*scale, coupling and cohesion*)? * Reusable and Composable logic (*reuse*)? * Statelessness or fault tolerant state-management (*resilience, appropriate state-storage*)? * Adequate retry logic, application error and usage logging? * System attempts to address performance through efficient communication – avoid chatty calls, inappropriate protocols, large data volumes, etc.   + Leverages caching mechanisms appropriately. |
| Managed Services | * Will the solution adopt an 3rd Party platform, SaaS, PaaS (e.g. AWS-RDS, SQS, SNS, SWF)? * Are there proper support agreements available? |
| Analytics | * Have the analytics\usage-tracking requirements been gathered as part of this solution been well defined for all activities? |
| Security | * Is the solution employing proper authentication and authorization methods? * Is the solution incorporating software mitigations against the common OWASP Top 10? |
| Data Protection | * Data and operational security requirements must be considered (Encryption, SSL, Digital Signatures, RBAC, IAM, etc.) * Is any PIl data being stored in this solution? |

|  |  |  |  |
| --- | --- | --- | --- |
| Artifact | Description | Responsible | Provided |
| Architecture diagram | A Visio or PDF file reflecting the proven solution architecture, containing:   * Technology components/platforms * Communication paths * Data storage types * Server IPs w/ Ports (if available) | Architect | ☐ |

# Full Architecture Review Process Details

## Step 1: Validate Application Readiness

The goal here is to validate the information created in the first round with the development team.

|  |  |
| --- | --- |
| Criteria | Considerations |
| New Technology | If the solution is introducing new technology then a technology justification must be made and will require the participation of the Infrastructure and Application Engineering Support team. Cost for licenses, support and maintenance must be submitted by team in advance. |
| Solution is virtualized | * Is the solution currently virtualized?   *This may require a P2V exercise for some existing platforms. In some cases the P2V may not be automatic and must be carried out manually (e.g. Solaris to Linux migrations)*.   * Does the solution have prohibitions with the Hypervisor? |
| Solution can scale horizontally and vertically | * Is the solution architected in such a way that it can be easily scaled up or down in an automated fashion? * Does the solution have tremendous elasticity demands? |
| Solution Logical Design | Are the logical components of the solution adhering to SOA design principles such as:   * Loosely coupled logic that can be scaled independently? * Reusable and Composable logic? * Statelessness or fault tolerant state-management? * Proper retry logic? * Proper application error and usage logging? |
| Cloud Managed Services | Will the solution adopt SaaS, PaaS elements as part of the migration (e.g. AWS-RDS, SQS, SNS, SWF)? |
| Solution Resilience (HA) | Can the solution handle partial failures of infrastructure or dependent components/applications (*retry or multi-path logic is key in cloud*)   * Does the system **demonstrate HA**? * Does the solution architecture mitigate the need for downtime for maintenance or upgrades?   Is there a threshold of acceptable downtime? |
| Special Needs Assessment | Are there unique business or technical requirements for the solution that may make public cloud inappropriate? |

### Activities (~60 minutes):

* Architects and a (1) senior dev director/developer meet and discuss/documented solution design for public and private cloud considerations.

### Artifacts to be produced:

|  |  |  |  |
| --- | --- | --- | --- |
| Artifact | Description | Responsible | Provided |
| None | N/A | N/A | ☐ |

## Step 2: Customer Considerations

|  |  |
| --- | --- |
| Criteria | Considerations |
| Understanding the platform consumers | How many consumers are expected to use this solution and what is the anticipated growth? |
| Frameworks | Is the solution using supported and well documented Javascript Frameworks? (React, Ember, Angular) |
| High Performance (time-based) | Are there special performance needs for this application? (e.g. real time information feeds, etc) |
| Responsive UX | Is the system able to effectively render information across desktop, tablet and mobile device form factors? |
| Analytics | Has the business/technology defined the usage metrics the solution will be capturing and persisting? |
| Accessibility Requirements | Does the solution have any accessibility requirements (WCAG, ADA, etc)? |

### Activities (30-60 minutes):

* Architects and UX Manager will **validate** that the proposed solution architecture aligns to UX standards and addresses analytics, accessibility and responsive design requirements and that will provide a performant and responsive end-user experience.

### Artifacts to be produced:

|  |  |  |  |
| --- | --- | --- | --- |
| Artifact | Description | Responsible | Provided |
| None | None | None | ☐ |

## Step 3: Assess Technical Requirements

Every solution considered for placement would have specific implementation details based on the workload requirements. The following criteria address the technical considerations to measure.

### Functional & Non-Functional Requirements

|  |  |
| --- | --- |
| Criteria | Considerations |
| Compute Resources | * Private Cloud: What are our internal private cloud VM sizes? * Public Cloud: What are the IaaS specification requirements expressed in cloud configurations (e.g. m3.xlarge – RAM, CPU/vCPU)? This is a right-sizing exercise and may take some experimentation. |
| Network Resources (both public & private) | * Does either solution pose limits on network bandwidth and speed required to support ingress and egress of data rates? (Load Balancers, Routers, DNS, switches etc.) * Are there specific transport (ssl) or network protocols or port/firewall requirements? * Have load balancers, DNS routing, IP Address ranges been determined? Consider that Load Balancers can take as many IP address spaces as there are compute nodes under management. |
| State-Management (*Caching*) Resources | * Does the solution require durable data/state-management? * If so, what caching solution is being used? * What information being cached? * What is the cache management policy (TTL, FIFO, LRU, etc)? * What is the expected growth of the cache size needs? * Public Cloud: Have managed services such as AWS ElasticCache (memCacheD/Redis) been considered? |
| Storage Resources (Capacity Planning) | * How much data is expected to be stored? * How quickly will the data grow? * Does the solution require file/block/object storage services? * Are there specific performance/configuration requirements for storage? * Is there backend database relational or non-relational requirements? * Public Cloud: Have managed services such as AWS Relational Database services been considered? (e.g. AWS RDS) |
| Integration Considerations (*Search, IAM, Citrix, etc.)* | Is the solution properly connecting to common enterprise platforms (e.g. ForgeRock IAM, Centralized Logging, etc..) |
| Monitoring and Instrumentation | * How will operational tools monitor your workload either in the cloud or on-premise and what is the integration costs to instrument your workload? * Are there special considerations when monitoring this solution or any part of the solution?   **Be sure to include detailed monitoring as part of any AWScloud-based architecture.** |

The following activities and artifacts will inform the decision for technical requirements.

### Automation Requirements

Automation imposes greater maturity on the SDLC through a greater emphasis on automating the provisioning, deployment and testing of both software and infrastructure. Furthermore, environmental parity throughout the SDLC (DEV, QA, STAGING, PRODUCTION) is critical to realize the predictable outcome promised by automation. Finally, automation is to provide increased IT/Business Agility for deploying new applications, features or conducting POCs.

|  |  |
| --- | --- |
| Criteria | Considerations |
| Increasing IT agility to deploy new applications or POCs more quickly | * Does the current solution provide automation for deploying software, provisioning infrastructure and testing both? |
| Provisioning Infrastructure | * Is the solution automating provisioning of IT resources? This implies parity/partial-parity of provisioned resources across the SDLC environments (e.g. UAT, PROD). * Is resource provisioning approval being provided in an adequate fashion (private/public cloud)? |
| Environment Integrity | * Are provisioned resources being tested for proper security? (automated infrastructure testing to be performed as part of any deployment/provision process). |
| Software Deployment | * Are you automating software deployments? * Does the solution have unit-tests, lynting and measured code coverage standadards as quality gates? |
| Automated Testing | * Has QA been consulted for testing strategy completeness? * Are other forms of software testing able to be performed?   + Functional   + Load/Stress?   + Component Failure (e.g. Chaos Monkey)   + Failover-HA/DR (e.g. Chaos Gorilla) |

### Activities (4-8 hours):

* Determine solution functional and non-functional requirements for proper sizing and architecture design resilience.
* Measure performance, resource utilization.
* Validate capacity assumptions.
* Evaluate maturity of solution automation.

### Artifacts to be produced:

|  |  |  |  |
| --- | --- | --- | --- |
| Artifact | Description | Responsible | Provided |
| Infrastructure Diagram | A Visio or PDF file reflecting the solution infrastructure architecture in both private and public cloud. Should include number of instances and resource profile (CPU, RAM, STORAGE, etc). Network security borders (e.g. subnets) and routers/switches should also be captured. | Architect  Plat. Eng. SME | ☐ |
| Software Deployment Plan & Diagram | A detailed plan describing degree of automation achieved or expected for solution (including process and tools used). | Architect  DevOps Team Mgr. | ☐ |

## Step 4: Assess Security Requirements

The following criteria collectively provide quality of service to the business and solution.

|  |  |
| --- | --- |
| Criteria | Considerations |
| Security & Data Protection | Data and operational security requirements must be considered (Authentication, Authorization, Encryption, SSL, Digital Signatures, RBAC, IAM, etc.) |
| Hardening Virtual Machines | Are superfluous function locked down?   * Unused operating system services? * Leverage private/public IP subnets * Lock-down ports? * Security Groups (Root Privileges)? * Access Control lists? |

### Activities (30-60 minutes):

* Architect & development lead/director to schedule and complete security questionnaire with security team.
* Security Architect to schedule meeting to cover Hardening of VMs.

### Artifacts to be produced:

|  |  |  |  |
| --- | --- | --- | --- |
| Artifact | Description | Responsible | Provided |
| Security Questionairre | Filled out security questionnaire | * Architect * Platform Eng. SME * Security SME | ☐ |

## Step 5: Assess Total Cost of Ownership (TCO)

Architecture review will require a TCO analysis if, and only if:

1. One has not been performed for the existing private or public cloud architecture
2. Recommended changes are introducing new/additional public/private cloud resources

To execute an accurate cost analysis the cloud governance team needs:

* 1 Year cloud price estimates for the resources to be used for this architecture

|  |  |
| --- | --- |
| Criteria | Considerations |
| TCO | Has the application usage pattern, based on initial and projected business demand/growth been well understood?  Are there licensing costs associated with this solution stack?  Cost comparisons for TCO must be performed for a minimum of 1 year and incorporate planned growth. To be considered fully complete compute resource requirements across ***non-prod environments must be included***. |
| SLA | If we are leveraging a managed service/platform:  Are there published **SLAs**?  Are there costs related to different SLA levels? |

### Activities (8-16 hours):

Perform a 1-year cost analysis for solution including infrastructure, licensing and automation spend. Based on the solution and infrastructure architecture artifacts a pricing matrix should be developed to provide semi-accurate costing projections for solution.

### Artifacts to be produced:

|  |  |  |
| --- | --- | --- |
| Artifact | Description | Provided |
| 1-Year TCO | A spreadsheet detailing the 1-year CAPEX and OPEX spend for a solution. | ☐ |