

Architecture Review Team (ART) Framework: FOR DEVELOPERS

Guidelines for HedgeServ Inc. Technology platforms

Guidelines for HedgeServ Inc. Technology platforms 1

Architecture Review Overview 3

Technology Governance Principles 3

Development Team Preparations 4

Activities (3-4 hours): 5

# 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 4 key areas:

* + Early application design readiness
  + Customer experience (as necessary)
  + Technical requirements
  + Security requirements

## Technology Governance Principles

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

* Implement a solution that meets the business objectives and the technology vision
* Properly vet out the appropriate combinations of internal/external solutions. Where solution(s) exist in the marketplace and the function is non-strategic/non-market differentiate for HedgeServ, we should consider the buy option. We want to use our internal resources to build market-leading/competitive-differentiation products, not commodity functions.
* Identify potential targets for deprecations.
* 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.
* Ensure testability and measurability to support the development cycle and production cycle.
* 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 2 hours***)

Some foundational information should be collected:

* Deprecation or alternative/replacement potentials
* Product lifecycle and EOL triggers

|  |  |
| --- | --- |
| 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)? |
| Entitlements | *Are user entitlements well understood and documented?* This information has to be provided to logical access to enable them to properly provision access to users. |

|  |  |
| --- | --- |
| 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.   [Currently Supported Technology](https://confluence.hedgeserv.com/display/IO/Technology+Standards) |
| 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 Capacity & 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? * Data size, volume, accumulation velocity(how much in a month/year) * New data types   + Data retention/archiving |

### Activities (3-4 hours):

* Architects will work with stakeholders to **validate** that the proposed solution architecture aligns to I/O, Security and UX standards to provide a performant and responsive end-user experience.

|  |  |  |  |
| --- | --- | --- | --- |
| Artifact | Description | Responsible | Provided |
| Architecture diagram | A Visio or PDF file reflecting the proven solution architecture, containing:   * Technology components/platforms   + IDAM   + Front-end tech-stack   + Server-side tech-stack   + Data persistence (type/capacity)   + Orchestration/Messaging   + Monitoring/Instrumentation (logging, etc) * Secure Communication pathways * Server IPs w/ Ports (if available) | Architect | ☐ |