Architecture Characteristics Worksheet

System/Project: **ClearView**

Domain/Quantum: **whole system**

Architect/Team: **mmest (Marco) /I2C**

Date: **2024-09-23** Next Review: _____

Top 3	Driving Characteristics	Implicit Characteristics
☐	**Data Integrity**	feasibility (cost/time)
☐	**Workflow**	security
☐	**Responsiveness**	maintainability
■	**Security – data privacy**	observability
■	**Data Consistency**	
■	**Interoperability**	
☐	**Scalability**	

Instructions

- Identify no more than 7 driving characteristics.
- Pick the top 3 characteristics (in any order).
- Implicit characteristics can become driving characteristics if they are *critical* concerns.
- Add additional characteristics identified that weren't deemed as important as the list of 7 to the *Others Considered* list.
- Definitions are on the following pages

Others Considered

extensibility

Common Architecture Characteristics

performance	data integrity	deployability
responsiveness	data consistency	testability
availability	adaptability	abstraction
fault tolerance	extensibility	workflow
scalability	interoperability	configurability
elasticity	concurrency	recoverability

Composite Architecture Characteristics

agility → maintainability, testability, deployability

reliability → availability, testability, data integrity, data consistency, fault tolerance

a / b denotes characteristics that are related; some systems only need one of these, other systems may need both

Architecture Characteristics Worksheet

performance
The amount of time it takes for the system to process a business request

responsiveness
The amount of time it takes to get a response to the user

availability
The amount of uptime of a system; usually measured in 9's (e.g., 99.9%)

fault tolerance
When fatal errors occur, other parts of the system continue to function

scalability
A function of system capacity and growth over time; as the number of users or requests increase in the system, responsiveness, performance, and error rates remain constant

elasticity
The system is able to expand and respond quickly to unexpected or anticipated extreme loads (e.g., going from 20 to 250,000 users instantly)

data integrity
The data across the system is correct and there is no data loss in the system

data consistency
The data across the system is in sync and consistent across databases and tables

adaptability
The ease in which a system can adapt to changes in environment and functionality

concurrency
The ability of the system to process simultaneous requests, in most cases in the same order in which they were received; implied when scalability and elasticity are supported

interoperability
The ability of the system to interface and interact with other systems to complete a business request

extensibility
The ease in which a system can be extended with additional features and functionality

deployability
The amount of ceremony involved with releasing the software, the frequency in which releases occur, and the overall risk of deployment

testability
The ease of and completeness of testing

abstraction
The level at which parts of the system are isolated from other parts of the system (both internal and external system interactions)

workflow
The ability of the system to manage complex workflows that require multiple parts (services) of the system to complete a business request

Created by Mark Richards, DeveloperToArchitect.com

Architecture Characteristics Worksheet

configurability

The ability of the system to support multiple configurations, as well as support custom on-demand configurations and configuration updates

recoverability

The ability of the system to start where it left off in the event of a system crash

feasibility (implicit)

Taking into account timeframes, budgets, and developer skills when making architectural choices; tight timeframes and budgets make this a driving architectural characteristic

security (implicit)

The ability of the system to restrict access to sensitive information or functionality

maintainability (implicit)

The level of effort required to locate and apply changes to the system

observability (implicit)

The ability of a system or a service to make available and stream metrics such as overall health, uptime, response times, performance, etc.